UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

THIAGO MOTTA

Interação Gestual sem Dispositivos para Displays Públicos

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof[a]. Dr[a]. Luciana Nedel
Orientador

Porto Alegre, março de 2013

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Motta, Thiago

Interação Gestual sem Dispositivos
para Displays Públicos / Thiago Motta. – Porto Alegre: PPGC da UFRGS, 2013.

107 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2013. Orientador: Luciana Nedel.

1. IHC. 2. Interação natural. 3. Interação gestual. 4. Displays públicos. I. Nedel, Luciana. II. Título.

“O estudo em geral, a busca da verdade e da beleza
são domínios em que nos é consentido ficar crianças por toda a vida.
— ALBERT EINSTEIN

AGRADECIMENTOS

É sempre importante poder retribuir àqueles que nos ajudam com algumas palavras de agradecimento. São apenas palavras, mas certamente são sinceras.

Agradeço aos meus pais por terem me produzido, à minha mãe e às minhas avós por terem me fornecido uma excelente base educacional e me criado às custas de muito esforço – ah, que criança hiperativa eu era! – e à toda minha família, de perto e de longe, por sempre me apoiarem e contribuírem com sugestões e conselhos. Agradeço aos amigos que surgiram pelos caminhos, há muito ou pouco tempo, mas especialmente àqueles amigos que acabaram virando uma nova família, com novas mães, pais, avós, tios, primos e até alguns irmãos. E dentre os amigos nada mais justo do que agradecer a uma amiga especial, que há sete anos acumulou legalmente o cargo de namorada.

A vida vai seguindo seu rumo e cada vez encontramos mais pessoas a quem inevitavelmente precisaremos agradecer. E eis que acaba surgindo uma outra família, em pleno ambiente de trabalho! E nada melhor do que ter uma chefe que é também uma mãe. Por isso, agradeço a todos aqueles que, ainda que não queiram, precisam me ver todos os dias úteis da semana, em especial à líder dessa turma.

É claro que não poderia deixar de agradecer à minha excelentíssima orientadora, afinal esse trabalho não existiria se não fosse por ela. Assim agradeço desde à aceitação da minha inscrição, passando por todas as etapas de desenvolvimento desse trabalho (nas muitas vezes em que eu estava perdido e nas tantas outras em que eu era relapso para com as reuniões), até chegar agora à conclusão dessa etapa.

Tenho que agradecer também, é claro, aos voluntariosos testadores que foram imprescindíveis para a conclusão desse trabalho. A vocês, caros 37, dou em pífia retribuição a honra de terem seus nomes citados aqui: Abel, Affonso, Alessandro, André, Andressa, Artur, Augusto, Bernardo, Carolina, César, Daniel, Daniele, Devanir, Douglas, Éderson, Fabiana, Fernanda, Fernando, Frederico, Gabriel, Gabriella, Guilherme, Helier, Jerônimo, João, Juliana, Leandro, Márcio, Marcos, Pedro, Rafael, Raphael, Rosália, Victor, Vitor, Wagner e William. Muito obrigado por cederem cerca de 25 minutos de seu tempo em prol da ciência e de um mundo melhor!

Agradeço aqui também à magnífica banca avaliadora do meu trabalho, que atribuiu a ele conceito “A”. Muito obrigado pelo seu tempo, pela atenção desprendida e pelas sábias contribuições com as quais me brindaram.

Agradeço por fim às coisas simples da vida, que tornam melhor o viver. Aos meus gatos, aos livros – passagens para outros mundos –, aos filmes, aos jogos, às piadas e brincadeiras, às viagens, às flores que nascem tão belas e a tudo mais que, por mais simples que seja, faz parte da razão pela qual vale a pena estar aqui em mais uma jornada. Obrigado à vida. Obrigado ao amor. Obrigado!

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS 7

LISTA DE FIGURAS 8

LISTA DE TABELAS 13

RESUMO 14

ABSTRACT 15

1 INTRODUÇÃO 16

2 TRABALHOS RELACIONADOS 19
2.1 Interação em grandes telas 19
2.1.1 Reconhecimento de gestos em telas multi-toque 19
2.1.2 Emprego de dispositivos móveis 21
2.1.3 Dispositivos de captura de movimento 24
2.1.4 Interação sem dispositivos 27
2.2 Interação utilizando o Kinect 29

3 UM ESTUDO SOBRE INTERAÇÃO GESTUAL SEM DISPOSITIVOS . 33
3.1 Árvore genealógica acadêmica 34
3.2 Seleção e manipulação de objetos simples 37
3.3 Mapa de localização de um prédio 40

4 PROJETO E IMPLEMENTAÇÃO 41
4.1 Decisões de Projeto 41
4.1.1 Microsoft Kinect 42
4.1.2 Web 2.0 e HTML5 43
4.2 Capturando dados do Kinect 43
4.2.1 Isolando as mãos 45
4.2.2 Detectando o contorno das mãos 47
4.2.3 Descobrindo o estado das mãos 48
4.3 Integrando o Kinect ao navegador Web 50
4.4 Interpretando os dados na página Web 52

5 AVALIAÇÃO DO SISTEMA 54
5.1 Hipóteses avaliadas 54
5.1.1 Condições de iluminação 54
5.1.2 Presença de outras pessoas no ambiente 54
5.1.3 Tipo de local 54
5.1.4 Tipo de tarefa 55
5.1.5 Apresentação da informação 55
5.2 Testes conduzidos 55
5.2.1 Avaliação informal por especialistas 55
5.2.2 Avaliação formal com usuários 56
5.2.3 Estudos de observação 57

6 RESULTADOS 59
6.1 Condições de iluminação 62
6.2 Presença de outras pessoas no ambiente 63
6.3 Tipo de local 64
6.4 Tipo de tarefa 66
6.5 Apresentação da informação 67

7 CONSIDERAÇÕES FINAIS 70
7.1 Notas adicionais 71
7.2 Contribuições 73

REFERÊNCIAS 74

ANEXO I 78

ANEXO II 81

ANEXO III 84

ANEXO IV 86

ANEXO V 89

ANEXO VI 97

LISTA DE ABREVIATURAS E SIGLAS

ANOVA *Analysis of Variance* (Análise de Variância)

API *Application Programming Interface* (Interface de Programação de Aplicações)

CAVE *Cave Automatic Virtual Environment* (Ambiente Virtual Automático C̈averna¨)

CSS *Cascading Style Sheet* (Folha de Estilos em Cascata)

FPS *Frames Per Second* (Quadros Por Segundo)

HTML *Hypertext Markup Language* (Linguagem de Marcação de Hipertexto)

IHC Interação Humano-Computador

IR *Infrared* (Infravermelho)

PC *Personal Computer* (Computador Pessoal)

PPGC Programa de Pós-Graduação em Computação

VGA *Video Graphics Array* (Arranjo Gráfico de Vídeo)

SDK *Software Development Kit* (Kit pra Desenvolvimento de Software)

UFRGS Universidade Federal do Rio Grande do Sul

LISTA DE FIGURAS

Figura 1.1: Graus de Interação de um usuário com um display público. 18

Figura 2.1: Usuários jogando em um *large display*. No centro, um jogador interage com o Homeworld e, nas pontas, jogadores interagem com o Quake 3 Arena. 20

Figura 2.2: Dois usuários interagindo com o City Wall a partir do reconhecimento de gestos executados sobre a tela. 21

Figura 2.3: Câmera do celular filma a tela grande e, a partir da interpretação da posição dos itens, o usuário pode manipular os itens a partir da tela do celular. 22

Figura 2.4: LED traseiro do celular ilumina a tela semitransparente e a luminosidade é captada por uma câmera colocada atrás da tela. A partir disso, o sistema identifica a posição do celular e envia para ele informações para serem mostradas em sua tela. 23

Figura 2.5: O ARC-Pad em funcionamento: à esquerda o usuário clica em uma posição do celular e o cursor da tela grande é colocado na posição mapeada; à direita o usuário arrasta o dedo na tela do celular e o cursor da tela grande acompanha o movimento. 24

Figura 2.6: Usuário interagindo na tela grande com o auxílio de um mouse 3D. . 25

Figura 2.7: A Vision Wand: (a) dois modelos da Vision Wand, com pontas de cores diferentes; (b) disposição do sistema proposto, com duas câmeras para ler o movimento feito com a varinha, que interage com a tela grande. 26

Figura 2.8: Usuário utilizando a luva proposta por Vogel e Balakrishnan, que contém marcadores passivos que são lidos por uma câmera *Vicon Motion Tracking*, para interagir em um display grande. 26

Figura 2.9: A luva proposta por Möhring e Fröhlich: (a) o sistema de captura de movimento da luva, com os anéis em torno dos dedos, que fornecem retorno háptico; (b) usuário interagindo em uma CAVE com o protótipo proposto. 27

Figura 2.10: Usuário interagindo em uma tela de alta resolução utilizando um mouse sem fio e um chapéu com marcadores passivos capturados por uma câmera para descobrir a posição em que o usuário de encontra. . 28

Figura 2.11: Usuário interagindo com uma tela de alta resolução sem utilizar nenhum dispositivo diretamente (sem segurar ou tê-lo afixado em si). . . 29

Figura 2.12: Sistema com 3 Kinects colocados a uma curta distância do usuário de forma a um não interferir com outro. 30

Figura 2.13: Usuário realizando um gesto para transladar verticalmente uma imagem, erguendo o braço direito. 30
Figura 2.14: Usuário interagindo sobre uma superfície: à esquerda, a disposição do sistema, com o Kinect colocado a uma certa altura da superfície; à direita a imagem capturada pelo Kinect, que será interpretada pelo programa. 31

Figura 3.1: Gestos suportados pelo modelo proposto: à esquerda, com as duas mãos abertas; ao centro, com uma das mãos fechadas; à direita, com as duas mãos fechadas. 34
Figura 3.2: Máquina de estados das mãos conforme utilizada pela aplicação Web. De acordo com a configuração de cada mão, um estado diferente é detectado e a aplicação pode tomar uma ação específica. O estado inicial é o "inativo", indicado pelo círculo de tom alaranjado. As linhas tracejadas são assim desenhadas apenas para fins de deixar a imagem mais limpa. 35
Figura 3.3: Tela inicial da aplicação de visualização da árvore genealógica, com a tooltip sobre o nodo indicado pelo mouse mostrando o nome da pessoa a que este se refere e a barra lateral com outras informações sobre ela. 36
Figura 3.4: Árvore genealógica acadêmica do PPGC na forma de um grafo: nodos representam discentes e docentes e arestas representam relações de orientação. À direita, a legenda de como interpretar os dados do grafo. 37
Figura 3.5: Outro detalhe na visualização da árvore genealógica acadêmica do PPGC mostra os antigos alunos de doutorado (marcados com círculos verdes) que atualmente são co-orientadores do Programa. Nos círculos vermelhos, os antigos orientadores destes alunos, ligados por uma linha pontilhada no detalhe em amarelo, e, nos azuis, seus atuais co-orientandos. 38
Figura 3.6: Detalhe na visualização da árvore genealógica acadêmica do PPGC, mostrando as inter-relações que se formam entre os nodos do grafo, com os nomes de alguns professores conforme aparecem na tooltip do aplicativo. 39
Figura 3.7: Capturas de tela de uma execução da aplicação de selecionar e posicionar objetos simples: no topo, à esquerda a tela inicial da tarefa de seleção de objetos, ao centro um novo tamanho de quadrados e, à direita, uma aplicação de aumento de zoom pelo usuário; embaixo, à esquerda a tela inicial da tarefa de posicionamento de objetos, ao centro e à direita, o usuário movimenta um quadrado com a mão esquerda e direita, respectivamente. 39
Figura 3.8: Duas telas da aplicação de visualização do mapa do prédio do Instituto de Informática: à esquerda a tela inicial da aplicação, quando não há a presença de nenhum usuário em frente à tela – é exibida uma janela mostrando quais gestos podem ser executados no sistema; à direita a tela conforme aparece passados 9 segundos depois que há um usuário em frente à tela – em vermelho semi-transparente é exibido o esqueleto do usuário, com ícones das mãos nas pontas de cada braço. 40

Figura 4.1: O dispositivo Kinect, com indicações de seus componentes utilizados para interpretação de gestos e de voz. 42

Figura 4.2: Conjunto de juntas do esqueleto que o Kinect consegue reconhecer no SDK *Kinect for Windows* da Microsoft. 45

Figura 4.3: Gráfico mostrando o limite de profundidade que é testado para isolar a mão do usuário: a linha tracejada vermelha indica o índice de profundidade do ponto central da mão; as linhas tracejadas em verde indicam os limites de profundidade que são utilizados para fins de comparação. 46

Figura 4.4: Isolamento das mãos do usuário na imagem de profundidade: A) um quadrado (em vermelho) é definido com base nas juntas do pulso e do centro das mão (pontos azuis); B) o recorte da mão direita da imagem de profundidade; C) após processamento da imagem, utilizando a informação de profundidade associada a cada pixel, isola-se a mão do resto da imagem. 47

Figura 4.5: Quatro passos do processamento da imagem para extração do contorno: em cima, a imagem em processamento, com a máscara centralizada no pixel sendo lido (quadrado azul); em baixo, o resultado parcial do processamento, com os pixels já processados em branco e preto. 48

Figura 4.6: Processamento da imagem para localização dos pontos adjacentes do contorno das mãos: à esquerda, um ponto problemático para leitura; à direita, pixels em cinza representam pontos já inserido no *array* que armazena o contorno e alaranjados aqueles ainda não lidos. Em cima uma situação em que o próximo ponto só seria identificado com uma máscara 5x5 e, abaixo, situação em que seria necessária uma máscara 7x7. 49

Figura 4.7: Processamento do algoritmo K-curvature sobre o contorno de uma mão, com os pontos de descontinuidade detectados marcados por círculos brancos. Os pontos que representam pontas de dedos (picos) foram detectados pelo ângulo formado entre os vetores em laranja, ao passo que aqueles detectados pelo ângulo entre os vetores verdes representa uma junção entre dedos (vales). Nota-se que alguns pontos de descontinuidade não foram detectados. 51

Figura 4.8: Aplicação desenvolvida para testar a leitura dos dados do Kinect mostrando a detecção das pontas dos dedos, de acordo com o algoritmo K-curvature: ao centro a imagem de profundidade como obtida pelo Kinect e em cada lado uma das mãos isolada e com pontos de descontinuidade identificados. 52

Figura 4.9: A arquitetura do sistema proposto, descrevendo as comunicações envolvidas no processo de transpor os dados interpretados do Kinect para a aplicação Web no display público. 53

Figura 5.1: Local de aplicação dos testes com usuários. Destaque para a marcação do local onde o usuário deveria se posicionar. 56

Figura 6.1: Subterfúgios utilizados pelos desenvolvedores de jogos para Kinect para contornar os problemas inerentes ao dispositivo: A) tela do *Kinect Sports* mostra ícones grandes, sobre os quais o usuário consiga manter sua mão parada por alguns segundos; B) imagem do *Kinect Adventures* mostrando ícones nos cantos da tela, tipicamente locais de fácil acesso, e que "atraem" as mãos do usuário; C) mais uma do *Kinect Sports*, que mostra a técnica de indicar a posição em que o usuário deve ficar para que seus movimentos sejam reconhecidos; D) *Dance Central* e demais jogos de dança fazem o reconhecimento de padrões brutos, sem interpretação detalhada dos gestos do usuário. . . 60

Figura 6.2: Gráfico demostrando o resumo das respostas dos usuários acerca da execução das tarefas de acordo com uma escala *likert* de 1 a 5, na qual 1 é um valor baixo (e.g. baixa dificuldade). Respostas foram para responsividade do sistema em geral e das tarefas de *pan & zoom*; dificuldade em executar as tarefas de seleção e posicionamento; e grau de divertimento e exaustão ao executar o teste. 61

Figura 6.3: Gráficos mostrando as diferenças de dados obtidos nos testes com o ambiente iluminado por lâmpadas ou não: acima, em relação ao tempo médio de execução de cada tarefa pelos usuários e abaixo em relação ao número de erros médio ocorridos em cada tarefa. Em relação ao tamanho de quadrados, a disposição é sempre dos maiores para os menores. O ambiente iluminado é representado pelas barras em azul e sem iluminação pelas vermelhas. A barra em preto entrecortando cada barra marca o desvio padrão. 62

Figura 6.4: Gráficos mostrando as diferenças de dados obtidos nos testes com e sem a presença de outras pessoas interferindo no teste: acima, em relação ao tempo médio de execução de cada tarefa pelos usuários e abaixo em relação ao número de erros médio ocorridos em cada tarefa. Em relação ao tamanho de quadrados, a disposição é sempre dos maiores para os menores. A interação sem interferência é representada pelas barras em azul e com a presença de pessoas pelas vermelhas. A barra em preto entrecortando cada barra marca o desvio padrão. 63

Figura 6.5: À esquerda o sistema conforme instalado no saguão de entrada do prédio cujo mapa é visualizado na aplicação e à direita dois momentos com grupos interagindo com o display. 65

Figura 6.6: Gráfico comparativo dos tempos médios de finalização de cada uma das tarefas de seleção (azul) e posicionamento (vermelho). As barras pretas indicam o intervalo de desvio padrão. 66

Figura 6.7: Em cima, gráficos comparativo dos tempos de cada usuário para finalização das tarefas de seleção e posicionamento do maior quadrado (à esquerda) e do posicionamento do maior quadrado com a seleção do menor quadrado (à direita). Abaixo, gráfico comparativo do número de erros das tarefas de seleção e posicionamento. 67

Figura 6.8: Gráfico comparativo dos tempos médios de cada usuário nas tarefas de seleção (à esquerda) e posicionamento (à direita) dos quadrados em seus tamanhos diversos. 68

Figura 7.1: Duas fotos de um usuário realizando tarefas de seleção de quadrados – à esquerda na disposição inicial do tamanho de quadrados; à direita após os quadrados terem diminuído de tamanho e aumentado em quantidade uma vez – situações em que o sistema melhor se portou. 71

LISTA DE TABELAS

Tabela 6.1: Tabela com os tempos médios de seleção e posicionamento dos quadrados pelos usuários, em ordem de maior tamanho para menor. . . . 68

RESUMO

Com o constante crescimento tecnológico, é bastante comum deparar-se com um display público em lugares de grande concentração de pessoas, como aeroportos e cinemas. Apesar de possuírem informações úteis, esses displays poderiam ser melhor aproveitados se fossem interativos. Baseando-se em pesquisas sobre a interação com displays grandes e as características próprias de um display colocado em um espaço público, busca-se uma maneira de interação que seja adequada a esse tipo de situação.

O presente trabalho introduz um método de interação por gestos sem necessitar que o usuário interagente segure ou tenha nele acoplado qualquer dispositivo ao interagir com um display público. Para realizar as tarefas que deseja, o usuário só precisa posicionar-se frente ao display e interagir com as informações na tela com suas mãos. São suportados gestos para navegação, seleção e manipulação de objetos, bem como para transladar a tela de visualização e ampliá-la ou diminui-la.

O sistema proposto é construído de forma que possa funcionar em aplicações diferentes sem um grande custo de implantação. Para isso, é utilizado um sistema do tipo cliente-servidor que integra a aplicação que contém as informações de interesse do usuário e a que interpreta os seus gestos. É utilizado o Microsoft Kinect para a leitura dos movimentos do usuário e um pós-processamento de imagens é realizado de modo a detectar se as mãos do usuário se encontram abertas os fechadas. Após, essa informação é interpretada por uma máquina de estados que identifica o que o usuário está querendo executar na aplicação cliente.

Afim de avaliar o quão robusto o sistema se portaria em um ambiente público real, são avaliados critérios que poderiam interferir na tarefa interativa, como a diferença de luminosidade do ambiente e a presença de mais pessoas no mesmo local de interação. Foram desenvolvidas três aplicações a título de estudo de caso e cada uma delas foi avaliada de forma diferente, sendo uma delas utiliza para fins de avaliação formal com usuários.

Demonstrados os resultados da avaliação realizada, conclui-se que o sistema, apesar de não se portar corretamente em todas situações, tem potencial de uso desde que sejam contornadas suas deficiências, a maior parte das quais originária das próprias limitações inerentes ao Kinect. O sistema proposto funciona suficientemente bem para seleção e manipulação de objetos grandes e para aplicações baseadas em interação do tipo *pan & zoom*, como navegação em mapas, por exemplo, e não é influenciado por diferenças de iluminação ou presença de outras pessoas no ambiente.

Palavras-chave: IHC, interação natural, interação gestual, displays públicos.

Deviceless Gestural Interaction Aimed to Public Displays

ABSTRACT

With the constant technological growth, it is quite common to come across a public display in places with high concentration of people, such as airports and theaters. Although they provide useful information, these displays could be better employed if they were interactive. Based on research on topics of interaction with large displays and the characteristics of a display placed in a public space, a way of interaction that is suitable for this kind of situation is searched.

This paper introduces a method of interaction by gestures without requiring that the interacting user take hold or have to him attached any device to interact with a public display. To accomplish the tasks he wants, he needs just to position himself in front of the display and to interact with the information on the screen with his hands. Gestures supported provide navigation, selection and manipulation of objects as well as to pan and zoom at the screen.

The proposed system is constructed so that it works in different applications without a large installation cost. In order to achieve this, the system implements a client-server model application that is able to integrate the part that contains the useful information to the user and the one that interprets his gestures. The Microsoft Kinect is used for reading the user's movements and techniques of image processing are performed to detect if the user's hands are open or closed. After this information is obtained, it runs through a state machine that identifies what the user is trying to do in the application.

In order to evaluate how robust the system is in a real public environment, some criteria that could interfere with the interactive task are evaluated, as the difference in brightness in the environment and the presence of another people in the same place of interaction. Three applications were developed as a case study and each one was evaluated differently, one of them being used for formal user evaluation.

Given the results of the performed tasks, it is possible to conclude that the system, although not behaving correctly in all situations, has potential use if its difficulties are circumvented, most of which come from Kinect's own inherent limitations. The proposed system works well enough for selection and manipulation of large objects and for use in applications based on pan & zoom, like those that supports map navigation, for example, and difference of ilumination or the presence of other persons on the environment does not interfere with the interaction process.

Keywords: HCI, natural interaction, gestural interaction, public displays.

1 INTRODUÇÃO

Desde que foram criados, os computadores exigem algum tipo de interação humana para funcionar. No princípio, a conexão manual de fios, conectando entradas e saídas bastava, mas logo percebeu-se que aquele tipo de interação não era adequada. Surgiram o teclado e, anos mais tarde, o mouse. Porém, especialmente ao observar os últimos vinte anos, percebeu-se que esses métodos interativos – tidos como tradicionais; baseados em botões – estão se tornando obsoletos, sendo necessária uma reavaliação da forma com a qual o ser humano interage com o computador da atualidade.

Com o avanço tecnológico e a consequente miniaturização dos componentes eletrônicos que fazem um computador, surgiram os computadores portáteis – dentre esses não somente laptops e notebooks, mas também telefones celulares, videogames, relógios digitais e toda uma gama de dispositivos eletrônicos que possuem microprocessadores – e os grandes displays, com possibilidade de exibir imagens em altas resoluções. No que diz respeito à qualidade das telas, fica bastante claro o avanço gradual tecnológico, pois elas sempre foram exploradas comercialmente. Contudo, no que concerne à interatividade somente há muito pouco tempo surgiram dispositivos comerciais que empregam métodos de interação não tradicionais. Especificamente, o videogame Wii da Nintendo[1], lançado em 2006, e o telefone celular iPhone da Apple[2], lançado em 2007, que se utilizavam de acelerômetros e tela sensível a toque múltiplo como principais diferenciais interativos, respectivamente.

Desde então, percebeu-se que a metodologia de interação também poderia ser usada como diferencial comercial e diversos dispositivos surgiram para concorrer com os supracitados, como o PS-Move[3] e o Kinect[4], para o Sony PlayStation e o X-Box da Microsoft respectivamente. Hoje é comum ter um telefone celular que suporte multi-toque. A própria Microsoft tem investido nesse ramo, e a mais nova versão do sistema operacional Windows[5] possui amplo suporte não só ao reconhecimento de toques – tanto para PCs quanto para sua versão para *smartphones* –, mas também ao reconhecimento de gestos executados no ar, com o auxílio de seu Kinect, atualmente também em uma versão especialmente produzida para ser utilizado em computadores pessoais.

Com a constante evolução do hardware, os métodos de interação precisam ser adaptados para se adequarem à inovação tecnológica. Um exemplo claro disso é o crescente uso de *large-displays* ou *tiled-displays*, ou seja, telas grandes na qual o usuário pode ter acesso a uma grande quantidade de informações simultaneamente, e, especificamente, displays

[1] http://us.wii.com/hardware
[2] http://www.apple.com/br/iphone
[3] http://us.playstation.com/ps3/playstation-move
[4] http://www.xbox.com/pt-br/kinect
[5] http://windows.microsoft.com/pt-BR/windows-8/preview

públicos, que são aqueles disponibilizados ao público em geral em ambientes não controlados. Em todos os casos, o usuário não tem o conforto de sentar-se à uma mesa e utilizar os convencionais mouse e teclado para interagir, pois ele precisa se postar de pé e, muitas vezes, se deslocar fisicamente, para conseguir visualizar toda informação. (NI et al., 2006) apresentam um *survey* sobre o assunto, no qual destacam todos os cuidados que se deve ter ao trabalhar com esse tipo de displays.

Displays públicos podem ser encontrados facilmente em aeroportos, shoppings, praças, restaurantes, etc. Eles informam as próximas seções no cinema, o preço de algum produto em promoção, o prato do dia, notícias importantes e uma série de outras informações que podem ser úteis conforme a necessidade de quem o visualiza. Essas telas mencionadas são estáticas e tudo o que utiliza-se para interagir com elas são os olhos. No entanto, cada vez mais frequentemente telas interativas estão sendo dispostas ao público, como mostram os trabalhos de (PELTONEN et al., 2008) (MICHELIS; MüLLER, 2011) (COUTRIX et al., 2010) (JACUCCI et al., 2010), descrevendo situações nas quais frequentadores de um espaço público se depararam com um display interativo e se sentiram atraídos a experimentá-lo, e a tendência é que o número aumente (KUIKKANIEMI et al., 2011).

As possibilidades são muitas ao se avaliar displays públicos. Eles podem avaliar um usuário de acordo com sua idade ou origem por meio de reconhecimento de faces e/ou por meio de perguntas e respostas e então se adaptar de acordo, comunicando-se usando outro formalismo ou em outro idioma. Pode possibilitar que pessoas participem ativamente em um evento que está ocorrendo em um local completamente diferente, enviando comentários ou contribuições artísticas, entre outros. Para isso, entretanto, é preciso levar em conta alguns critérios, como observaram (VOGEL; BALAKRISHNAN, 2004), destacando os graus de interação que um display público tem com o usuário baseando-se em sua aproximação com o mesmo, como mostra a figura 1.1. Ao passar longe de um display público, um usuário não tem interesse em consultá-lo, mas esse panorama se altera conforme o usuário percebe o display e se aproxima dele. Ao detectar uma aproximação, o display deve fornecer uma resposta e, por fim, quando o usuário está imediatamente em frente a si, deve fornecer a ele um espaço de interação individualizado. Geralmente a IHC assume que o usuário tem conhecimento do computador em primeiro lugar, mas isso não é necessariamente verdade no caso de displays públicos (MüLLER et al., 2010).

Ainda que existam trabalhos que exploram as capacidades interativas de um display público, essa é uma área ainda pouco abordada e, consequentemente, não há muitos indicativos de qual seria o melhor método de interação a ser utilizado. Nos trabalhos mencionados acima, foram utilizados grandes telas sensitivas ao toque para possibilitar a interação dos usuários. Contudo, outras abordagens podem ser utilizadas, como será visto no capítulo 2. Dentre essas abordagens, uma que chama bastante atenção é aquela na qual o usuário não necessita de qualquer dispositivo e que pode ser utilizada em qualquer tipo de tela, ou seja, onde o usuário interage com as informações presentes em uma tela não sensitiva utilizando apenas o seu próprio corpo.

Essa abordagem é particularmente interessante, pois quando lidamos com displays de acesso público não é desejável o compartilhamento de dispositivos, uma vez que podem ser fonte de proliferação de vírus, especialmente nos tempos em que há controle de epidemias, como tem ocorrido nos últimos invernos, por exemplo. Além disso, o usuário a interagir deve ter um acesso rápido às informações, sem precisar aprender como manipular dispositivos. Em um cenário ideal, o usuário se posicionaria frente ao display e intuiria sobre o método de interagir com as informações ali presentes.

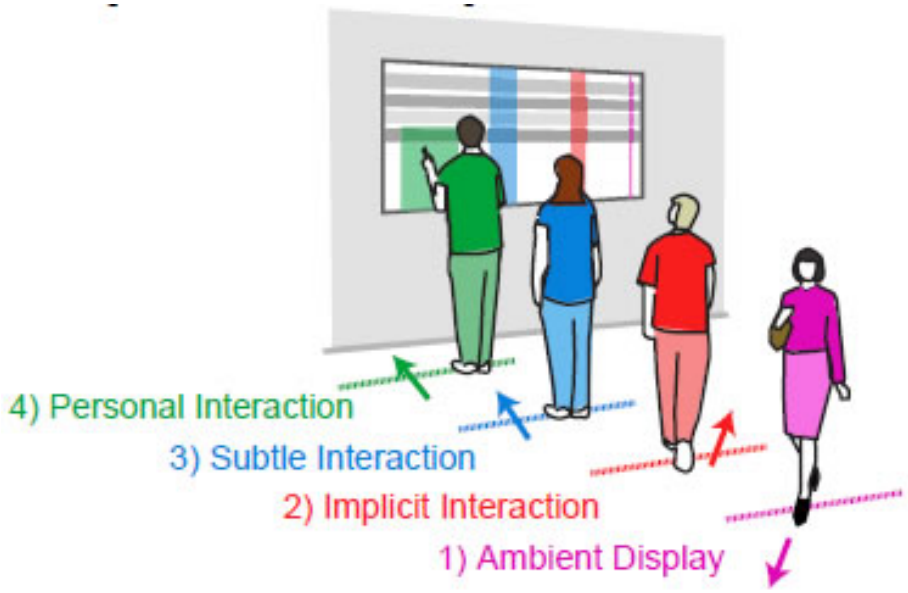

Figura 1.1: Graus de Interação de um usuário com um display público (VOGEL; BALAKRISHNAN, 2004).

Nesse trabalho, é proposto um modelo no qual o usuário utiliza-se unicamente de suas mãos para realizar tarefas interativas em um grande display. São apresentados estudos de casos que exploram técnicas de seleção e manipulação de objetos virtuais em duas dimensões e de *pan & zoom* sobre uma variada quantidade de informações. Visando sua aplicabilidade em displays públicos, são avaliados formalmente diversos critérios que poderiam interferir na interação do usuário em um ambiente não controlado, tal como a quantidade de iluminação do local e a presença de outras pessoas no mesmo ambiente. Além disso, a precisão da técnica proposta é avaliada em diferentes cenários e abordagens.

Após análise de resultados dos testes conduzidos, é possível constatar que o modelo proposto é suficientemente bom em determinados cenários, como a seleção e posicionamento de objetos grandes e o *pan & zoom* na tela, enquanto deixa a desejar em outros, como a seleção e posicionamento de objetos pequenos. Baseando-se no baixo número de pesquisas sobre o assunto, pode-se afirmar que ainda não há uma metodologia que permita uma interação mais precisa em um modelo sem dispositivos. Contudo, enquanto a tecnologia não avança no sentido de criar dispositivos robustos para o reconhecimento de gestos, a metodologia proposta nessa dissertação pode ser aplicada com desempenho aceitável e a um custo financeiro bastante baixo.

A seguir, no capítulo 2 são apresentados os trabalhos relacionados e o estado da arte no que se trata da interação com displays grandes. O capítulo 3 apresenta os estudos de caso realizados, explicando a metodologia escolhida e as aplicações desenvolvidas. Em seguida, no capítulo 4 encontram-se as decisões de projeto que foram tomadas e por quais motivos, bem como os detalhes de implementação conforme os dispositivos e ambientes de implantação definidos previamente. O capítulo 5 detalha os experimentos que foram conduzidos, em todos os aspectos estudados e o capítulo 6 expõe os resultados obtidos nesses experimentos, sob pontos de vista quantitativo e subjetivo, de acordo com avaliação dos usuários. No capítulo 7 o trabalho é concluído e são apresentadas considerações sobre os resultados atingidos e possíveis trabalhos futuros.

2 TRABALHOS RELACIONADOS

No que concerne esse trabalho, é necessário avaliar o estado da arte da interação em grandes displays, afim de identificar os diversos critérios que devem ser levados em conta e procurar construir um modelo que englobe os principais benefícios de cada técnica. Além disso, como almeja-se a construção de um modelo que empregue o reconhecimento de gestos do usuário sem a manipulação de dispositivos pelo mesmo, mostra-se pertinente examinar trabalhos que utilizam-se do Microsoft Kinect, atualmente o dispositivo com a melhor relação custo-benefício nesse sentido (TONG et al., 2012). De modo a organizar melhor as duas linhas de pesquisa, cada uma delas será apresentada em uma seção abaixo.

2.1 Interação em grandes telas

Ao se lidar com telas de alta resolução – e, muitas vezes, mesmo com telas normais – não há um consenso sobre qual a melhor forma de interação. Sabe-se que métodos tradicionais, como teclado e mouse, não são convenientes na maioria dos casos (NI et al., 2006), pois ao interagir com um display grande o usuário geralmente não pode fazê-lo sentado, ficando sem um apoio adequado para tais ferramentas. Procura-se, então, encontrar uma abordagem que se mostre mais interessante para este caso e a interação por gestos tem sido frequentemente utilizada em pesquisas recentes da área.

A interação gestual pode ser abordada de diversas formas, de acordo com a definição do projetista. Portanto, uma série de trabalhos têm sido publicados, apresentando maneiras distintas de se empregá-la. Afim de destacar os principais critérios das diferentes classes de interação por gestos e avaliar sua aplicabilidade em grandes displays, dividir-se-á a interação gestual em quatro categorias, conforme segue.

2.1.1 Reconhecimento de gestos em telas multi-toque

Telas sensíveis ao toque têm sido estudadas há bastante tempo. Porém, com o advento das telas sensitivas a múltiplos toques, sua utilização tem sido cada vez mais frequente, seja na Academia ou no Mercado. Com o passar do tempo, alguns gestos realizados na tela sensitiva já se tornaram praticamente padronizados, como o de abrir e fechar dois dedos junto a tela para fazer zoom e deslizar um dedo sobre a tela para alternar entre cenários. Essa padronização se deve especialmente ao surgimento de muitos produtos comerciais que empregam as mesmas técnicas, como os *smartphones* em geral e o Surface da Microsoft[1].

Há muitos trabalhos interessantes que utilizam essa tecnologia. A maioria deles apresenta uma forma alternativa de capturar os toques na tela, ao invés de utilizar um *tiled-*

[1] http://www.microsoft.com/surface/en/us/whatissurface.aspx

display de monitores sensitivos, que acabaria levando a um custo muito alto de projeto. O trabalho de Stødle *et al.* traz uma solução para esse problema com o emprego de câmeras (STØDLE et al., 2008). O trabalho se deu sobre um arranjo de telas projetadas, cuja interação é capturada por 16 câmeras e em cluster com 9 computadores. Para fins de teste do método interativo, sequências de toques na tela foram definidas para serem utilizados nos jogos de computador Quake 3 Arena[2] e Homeworld[3]. A tela foi utilizada simultaneamente por três jogadores, conforme pode ser visto na figura 2.1.

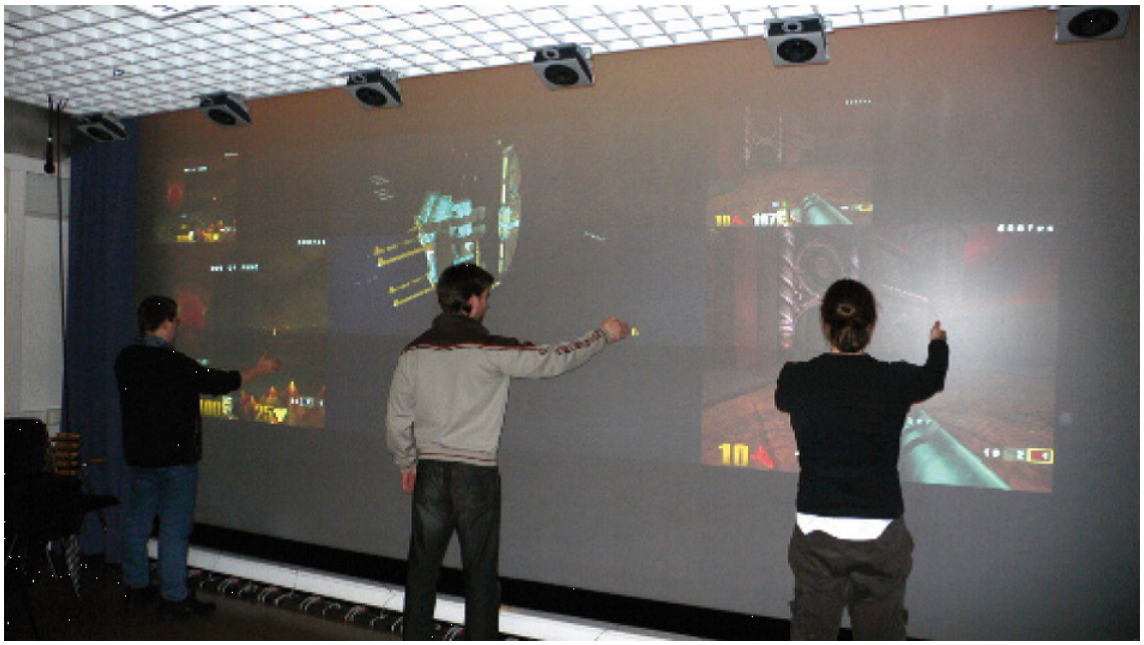

Figura 2.1: Usuários jogando em um *large display*. No centro, um jogador interage com o Homeworld e, nas pontas, jogadores interagem com o Quake 3 Arena (STØDLE et al., 2008).

O estudo inicial feito pelos autores indicou uma boa aceitação da nova metodologia, especialmente no que diz respeito ao Quake 3, ao contrário do que eles supunham. O maior impedimento do uso de seu sistema é o tempo de latência para a interpretação dos gestos, que figura em torno dos 204ms. Essa latência se deve, de acordo com os autores, principalmente por conta da tecnologia das câmeras e pelo sistema detector de toques, que precisa esperar por uma resposta de todas as câmeras para efetuar o processamento.

Trabalhos interessantes foram feitos com telas sensitivas em displays públicos, ou seja, utilizadas por usuários leigos e em um ambiente não totalmente controlado. Peltonen *et al.* e Jacucci *et al.* apresentam dois trabalhos realizados sobre o mesmo sistema de tela sensitiva ao toque do Helsinki Institute for Information Technology: City Wall (PELTONEN et al., 2008) e Worlds of Information (JACUCCI et al., 2010), respectivamente. A arquitetura do sistema proposto envolve uma tela retroprojetiva semitransparente, uma câmera sensitiva a emissões de infravermelho, posicionada próxima ao retroprojetor, e emissores de luz infravermelha sobre a tela para detectar os toques. De acordo com os autores, o sistema provê a captura de tantos toques quantos forem possíveis de serem feitos pela dimensão da tela.

[2]http://en.wikipedia.org/wiki/Quake_III_Arena
[3]http://www.relic.com/games/homeworld

Figura 2.2: Dois usuários interagindo com o City Wall a partir do reconhecimento de gestos executados sobre a tela (PELTONEN et al., 2008).

O trabalho apresenta como uma das vantagens do sistema a possibilidade de haver múltiplos usuários interagindo ao mesmo tempo com a tela, como acontece na figura 2.2 e o trabalho colaborativo que pode ser realizado por dois ou mais usuários ao buscar uma imagem, por exemplo. Entretanto, a possibilidade de mais de um interagente pode gerar problemas, como também é relatado no trabalho, como quando, por exemplo, um usuário invade o espaço que o outro está ocupando, sobrepondo sua visualização.

Embora telas sensitivas apresentem vantagens em relação a outros métodos interativos sobre displays grandes – o que se deve justamente à grande quantidade de dispositivos comerciais que empregam essa tecnologia e ao fato do usuário estar habituado a utilizá-la –, os sistemas que fazem uso dessa funcionalidade são em geral bastante custosos e/ou necessitam de um grande espaço para serem construídos.

Além disso, telas sensitivas são mais adequadas quando o usuário está realizando a interação perto da tela, o que não é necessariamente verdade quando trata-se de displays públicos, pois uma das vantagens que essas telas oferecem é justamente poder visualizar um grande número de informações simultaneamente, situação em que o usuário precisa estar afastado da tela para que consiga visualizá-la por completo.

2.1.2 Emprego de dispositivos móveis

Outra técnica bastante utilizada para realizar a interação com grandes displays é o uso de dispositivos móveis como ferramentas interativas. Com o avanço tecnológico dos últimos anos, hoje é comum ver pessoas com *smartphones*, reprodutores de som, TVs portáteis e outros dispositivos do gênero, que contam com diversos recursos interativos, como acelerômetros e câmeras, por exemplo, e possuem um custo não proibitivo.

Os *smartphones*, em especial, vêm sendo muito utilizados em pesquisas na área de IHC, e são considerados essenciais na vida contemporânea. É o primeiro aparelho computacional que realmente conquistou a sociedade, independente de classe social (BALLAGAS et al., 2006). A maioria dos trabalhos que se utilizam de dispositivos móveis para interação o fazem por meios ópticos, seja por reconhecimento de padrões (BO-

RING et al., 2010; PEARS; JACKSON; OLIVIER, 2009) ou por rastreamento do dispositivo (OLWAL, 2006).

O *Touch Projector*, de Boring *et al.*, utiliza um *smartphone* para filmar uma tela grande e, a partir disso, identificá-la e receber seu conteúdo (vide figura 2.3). A partir disso, o usuário pode manipular elementos que aparecem na tela do *smartphone* e suas ações são refletidas na tela grande, além de poder mover elementos de uma tela para outra movendo o celular de modo que sua câmera aponte para a tela de destino (BORING et al., 2010).

Figura 2.3: Câmera do celular filma a tela grande e, a partir da interpretação da posição dos itens, o usuário pode manipular os itens a partir da tela do celular (BORING et al., 2010).

Inicialmente, o dispositivo móvel precisa enviar uma requisição para a tela grande, que, reconhecendo-o, lhe envia seu conteúdo pela rede. O sistema precisa, então, verificar a imagem lida pelo celular e determinar em que posição da tela do aparelho estão os itens que podem ser manipulados. O usuário pode, então, selecionar um item pressionando a tela e manipulá-lo transladando-o ou escalando-o. O trabalho aponta que o sistema funcionou bem quando os itens a serem manipulados apareciam grandes na tela do celular, enquanto que com os pequenos não teve um desempenho tão bom. A causa do problema, segundo os autores, é a qualidade da câmera usada - a câmera nativa do iPhone 3G -, que não tinha boa resolução e também só conseguia interagir com o sistema a 8 quadros por segundo.

O trabalho de Pears *et al.* é bastante semelhante, utilizando a câmera de um *smartphone* para localizar um objeto a ser manipulado em uma tela grande e utilizando o dispositivo móvel como um mouse 3D, capaz de prover 4 graus de liberdade (translação nos três eixos e rotação no eixo Z/no plano da tela). Uma diferença importante é que esse trabalho se utilizou de marcações nos itens para facilitar o reconhecimento pela câmera. Apesar de o protótipo apresentado ter funcionado bem, os autores destacam que os marcadores da imagens são um problema, pois agem como distratores para os usuários. Além disso,

mesmo com os marcadores, por vezes o usuário movia o celular rápido demais e o sistema perdia sua referência, causando o retorno do item manipulado à sua posição inicial (PEARS; JACKSON; OLIVIER, 2009).

O Light Sense (OLWAL, 2006), segue uma abordagem contrária a dos trabalhos anteriores. Ao invés de usar a tela grande como emissora de informações e o dispositivo móvel como captor, ele utiliza o LED – presente na maioria dos telefones celulares para melhorar a qualidade de fotos tiradas com baixa iluminação – como rastreador do aparelho, que é lido por uma câmera colocada atrás de uma tela semitransparente, responsável por projetar a imagem vista pelo usuário, conforme mostra a figura 2.4.

Figura 2.4: LED traseiro do celular ilumina a tela semitransparente e a luminosidade é captada por uma câmera colocada atrás da tela. A partir disso, o sistema identifica a posição do celular e envia para ele informações para serem mostradas em sua tela (OLWAL, 2006).

A câmera lê a posição do feixe de luz e envia essa informação para o computador, que a processa e, por sua vez, envia uma imagem para ser mostrada na tela do dispositivo móvel através de uma conexão *Bluetooth*. Isso permite, como no exemplo ilustrado, que o usuário navegue por uma tela (seja na vertical ou horizontal) que apresenta um mapa e receba informações sobre o local que está observando, como nomes de estações, informações de trânsito, etc.

Muito embora fazer o rastreamento da iluminação no espectro visível (no lugar de infravermelho, por exemplo) não tenha sofrido com interferências de outras fontes luminosas, o autor afirma que isso pode vir a ser um problema em espaços públicos, onde não se tem controle do ambiente. O autor ainda destaca que o sistema proposto pode ser usado concomitantemente com outras técnicas, como as que se utilizem da câmera do aparelho.

Contrário aos trabalhos supracitados, o ARC-Pad (MCCALLUM; IRANI, 2009) não utiliza meios ópticos para realizar a interação, mas, sim, faz um mapeamento absoluto ou

relativo da tela grande para a tela do dispositivo móvel da seguinte forma: ao pressionar rapidamente uma parte da tela do celular, o cursor da tela grande é posicionado na posição mapeada para a área pressionada; ao deslizar o dedo pela tela do celular, o cursor da tela grande segue o mesmo movimento, com uma leve aceleração adicional, como demostrado na figura 2.5.

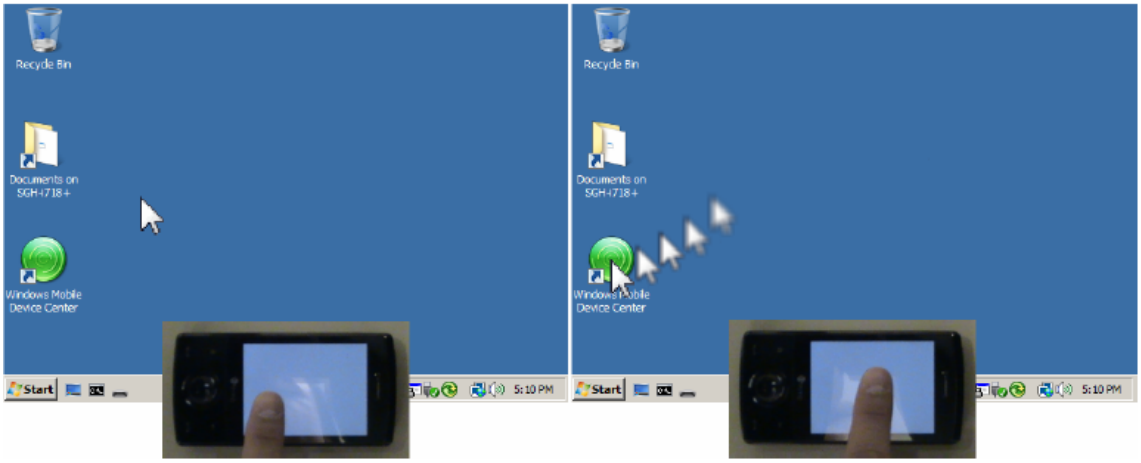

Figura 2.5: O ARC-Pad em funcionamento: à esquerda o usuário clica em uma posição do celular e o cursor da tela grande é colocado na posição mapeada; à direita o usuário arrasta o dedo na tela do celular e o cursor da tela grande acompanha o movimento (MCCALLUM; IRANI, 2009).

O uso de dispositivos móveis para interagir com displays grandes é bastante eficaz na maioria dos casos. Com o avanço tecnológico, eventuais problemas técnicos, como a duração da bateria e a sensitividade das telas, devem ser sanados, propiciando um aprimoramento natural das técnicas apresentadas. Contudo, apesar de estarem cada vez mais presente na sociedade, os *smartphones* ainda estão longe de se tornarem acessíveis a todos, especialmente ao se observar as diferenças sociais em países como o Brasil, por exemplo.

2.1.3 Dispositivos de captura de movimento

A abordagem mais utilizada ao projetar a interação com telas de alta resolução é a criação de novos dispositivos ou a adaptação de dispositivos existentes que não dispositivos móveis, como, por exemplo, o *Wiimote*, controle do Nintendo Wii, ou mouses 3D, especialmente em casos onde se quer obter um sistema cujo custo não seja proibitivo.

Nancel *et al.* fizeram um estudo detalhado sobre técnicas de interação com e sem dispositivos em um *tiled display* com 32 monitores de 30”. O objetivo principal do trabalho foi verificar o quanto o nível de direcionamento que o usuário tem a sua disposição ao interagir com um dispositivo é relevante para cumprir uma tarefa (NANCEL et al., 2011). O experimento utilizou um mouse 3D para realizar a primeira parte do estudo, que consistia em técnicas de translação e zoom para localizar determinada tela em uma espécie de labirinto, como pode ser visto na figura 2.6.

O direcionamento linear, provido pelo botão de scroll do mouse ao efetuar zoom foi decisivo para a performance dos usuários, resultado este levantado tanto na avaliação quantitativa quanto na subjetiva. O sistema, no entanto, apesar de funcionar bastante bem, é muito restritivo, permitindo apenas dois graus de liberdade. Testes mais aprofundados precisariam ser feitos para verificar se o nível de direcionamento é providencial quando

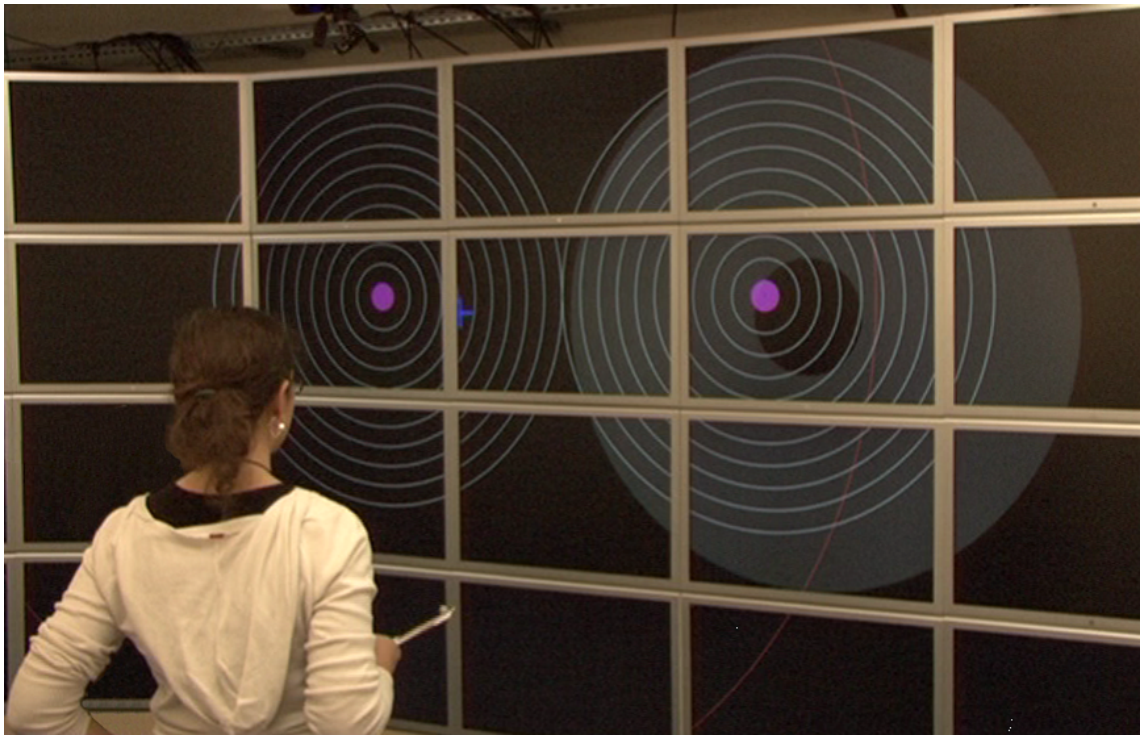

Figura 2.6: Usuário interagindo na tela grande com o auxílio de um mouse 3D (NANCEL et al., 2011).

em uma tarefa mais complexa. Ou autores também avaliaram o reconhecimento de gestos executados livremente no ar, sem o auxílio de dispositivos, e essa parte de seu trabalho será apresentada na próxima subseção.

A VisionWand (CAO; BALAKRISHNAN, 2003) é da classe de trabalhos que apresentam um novo dispositivo interativo. Ela consiste em uma varinha com pontas coloridas que são lidas por uma câmera e proporcionam ao usuário a seleção e manipulação de objetos virtuais em 5 graus de liberdade de acordo com a maneira que ele segura o dispositivo. O sistema funciona através do processamento de imagens lidas por duas câmeras que apontam para uma mesma região, onde os gestos com a varinha devem ser executados, como mostra a figura 2.7.

A VisionWand possui nove gestos distintos que podem ser reconhecidos e o sistema pode ainda diferenciá-los pela cor da ponta da varinha que está colocada para cima. Tal sistema permite uma infinidade de propriedades interativas, como as tradicionais seleção, manipulação e navegação e ainda comandos de desfazer e de consultar informações sobre determinado objeto. Os autores realizaram testes iniciais com o dispositivo e, pelas avaliações subjetivas dos usuários, concluíram que ele possui grande potencial de uso. Contudo, o posicionamento das câmeras precisa ser revisto, pois da forma como elas foram dispostas inicialmente é muito fácil haver oclusão.

Dado que as mãos são as partes do corpo humano mais utilizadas nas atividades interativas, é comum o uso de luvas de dados para capturar gestos. Entretanto, como luvas de dados comerciais possuem um custo muito alto, muitos pesquisadores investem na criação de luvas de dados mais acessíveis ou, ainda, que forneçam uma maior capacidade interativa (VOGEL; BALAKRISHNAN, 2005)(MOEHRING; FROEHLICH, 2011). O trabalho de Vogel e Balakrishnan apresenta uma luva de dados com pontos refletores passivos nas pontas dos dedos e uma *Vicon Motion Tracking* para identificar esses pontos,

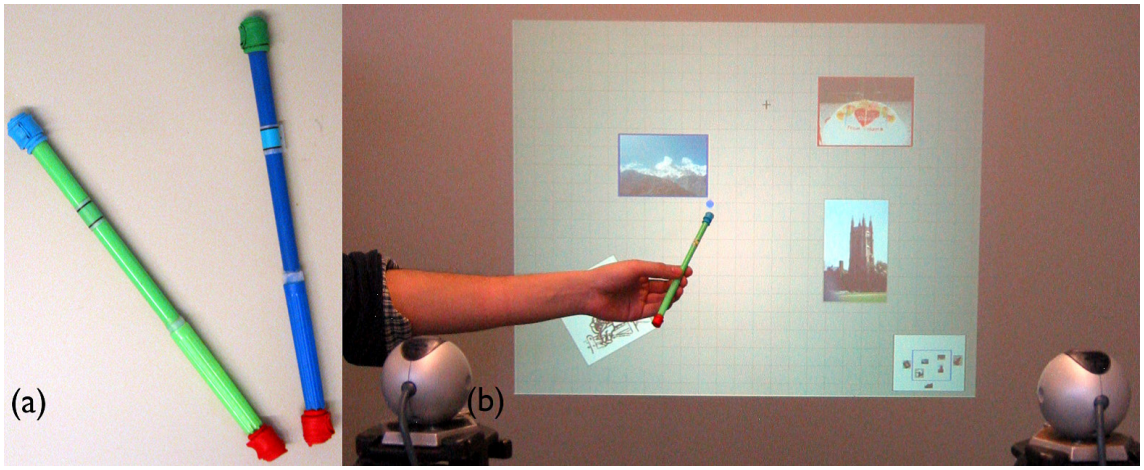

Figura 2.7: A Vision Wand: (a) dois modelos da Vision Wand, com pontas de cores diferentes; (b) disposição do sistema proposto, com duas câmeras para ler o movimento feito com a varinha, que interage com a tela grande (CAO; BALAKRISHNAN, 2003).

com a qual é possível manipular objetos virtuais em frente à tela, conforme apresentado na figura 2.8.

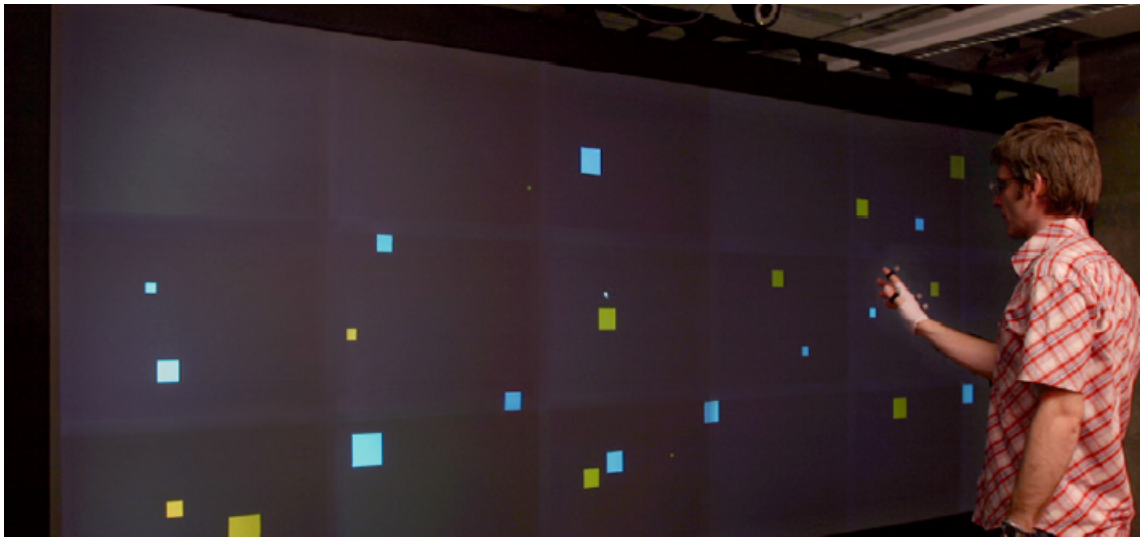

Figura 2.8: Usuário utilizando a luva proposta por Vogel e Balakrishnan, que contém marcadores passivos que são lidos por uma câmera *Vicon Motion Tracking*, para interagir em um display grande (VOGEL; BALAKRISHNAN, 2005).

O trabalho avaliou o desempenho dos usuários frente a tarefa de seleção de objetos. Uma abordagem interessante que os autores utilizaram foi a de introduzir falsas falhas de leitura da luva quando o usuário se posicionava de uma maneira desconfortável para ele, por exemplo ao esticar muito o braço. Isso foi feito para que o usuário procurasse interagir com o display da forma como interagiria se estivesse apenas utilizando o sistema casualmente e não se preocupasse somente com a velocidade de completude da tarefa, abrindo mão do próprio conforto momentaneamente para obter um melhor desempenho.

Já o trabalho de Möhring e Fröhlich também se utiliza de mecanismos ópticos para a captura da posição dos dedos e das mãos, em um sistema que, segundo os autores, oferece alta precisão. O diferencial dessa luva é que ela pode fornecer retorno háptico, que é uma

capacidade interativa bastante importante (vide luva na figura 2.9-a).

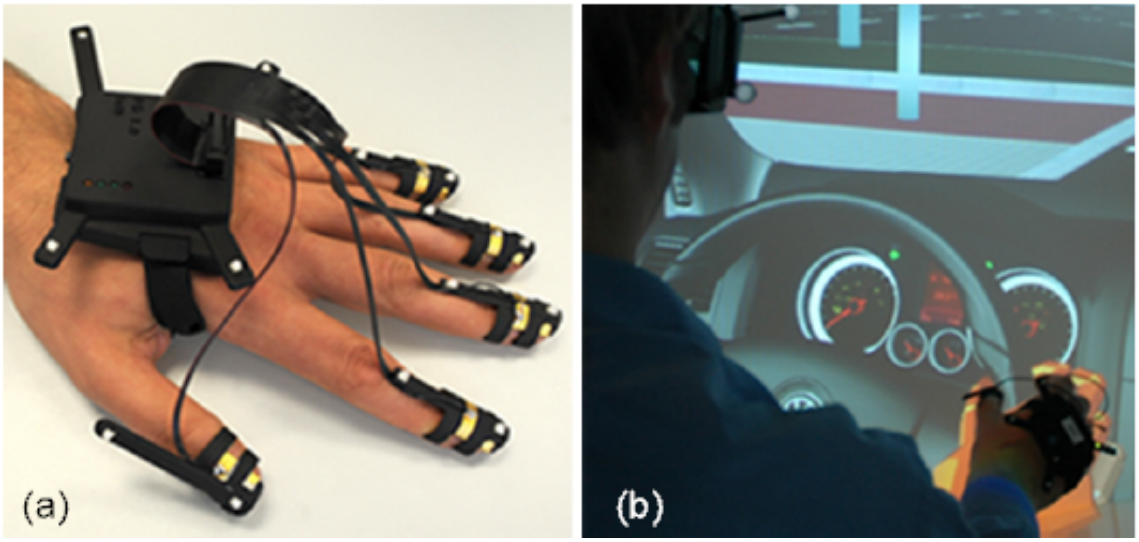

Figura 2.9: A luva proposta por Möhring e Fröhlich: (a) o sistema de captura de movimento da luva, com os anéis em torno dos dedos, que fornecem retorno háptico; (b) usuário interagindo em uma CAVE com o protótipo proposto (MOEHRING; FROEHLICH, 2011).

Os autores utilizaram as luvas para interagir com o interior de um carro virtual em uma CAVE (conforme mostra a figura 2.9-b), com o objetivo de encontrar uma maneira alternativa de apresentar um veículo a um cliente. Eles concluem que o dispositivo que criaram é o ideal para essa atividade a não ser que surja algum outro componente eletrônico no desenrolar da evolução tecnológica.

Outro trabalho interessante (BALL; NORTH; BOWMAN, 2007) tem como foco a quantidade de movimentação dos usuários frente a uma *tiled-display*. Eles utilizaram para a interação um mouse comum sem fio e um chapéu com marcadores ópticos para capturar a posição do usuário em relação à tela, como mostra a figura 2.10. As tarefas de estudo de caso foram executadas sobre um grande mapa e os usuários tiveram que realizar as tarefas em telas de tamanho variado.

O trabalho teve conclusões bastante interessantes: com uma maior tela, o usuário tende a se movimentar mais frente à tela em detrimento de translações virtuais do mapa; e, ao se movimentar frente à tela, os usuários finalizaram as tarefas em menos tempo. Esses resultados confirmam a utilidade de grandes displays, mas traz um questionamento interessante acerca do que leva o usuário a preferir a movimentação física à movimentação virtual quando essa opção existe.

A criação de dispositivos interativos é, certamente, uma boa abordagem na tentativa de resolver os problemas de interação em grandes displays. No entanto, por vezes seu uso não é factível, especialmente ao se utilizar captura óptica, quando problemas de oclusão são frequentes (CAO; BALAKRISHNAN, 2003).

2.1.4 Interação sem dispositivos

Em contrapartida à interação na qual os usuários utilizam dispositivos diretamente – sejam comerciais ou protótipos de pesquisa – existem pesquisas sendo feitas sobre interação sem dispositivos, ou seja, sem que qualquer componente, eletrônico ou não, seja

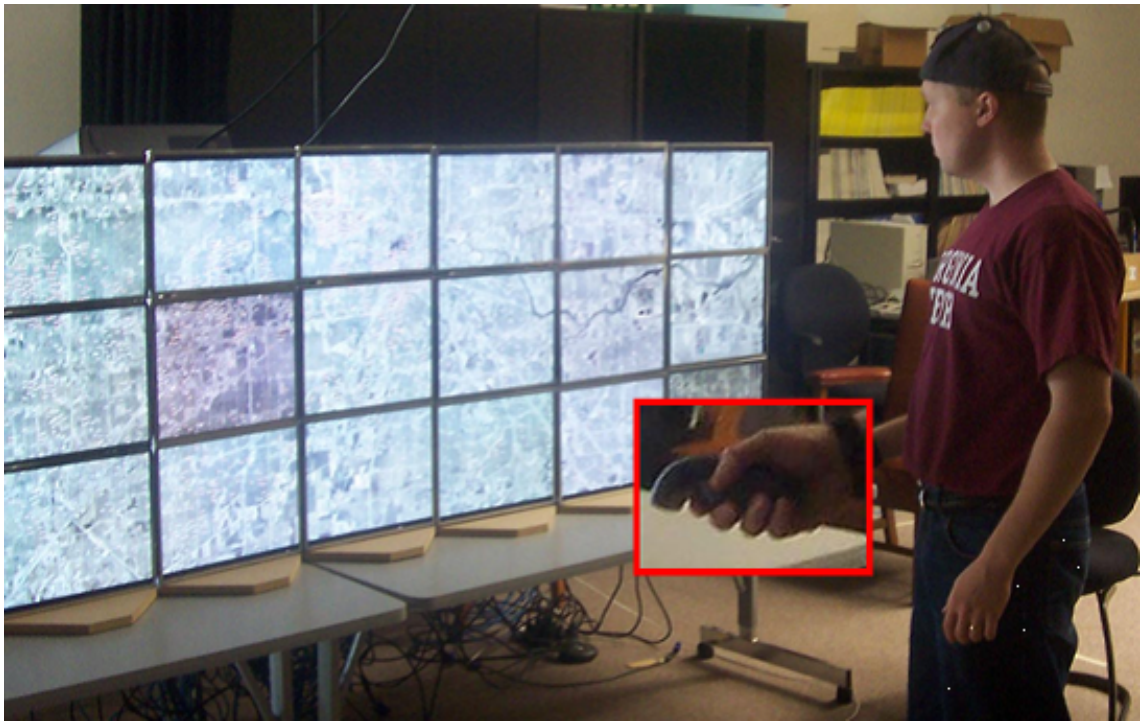

Figura 2.10: Usuário interagindo em uma tela de alta resolução utilizando um mouse sem fio e um chapéu com marcadores passivos capturados por uma câmera para descobrir a posição em que o usuário de encontra (BALL; NORTH; BOWMAN, 2007).

acoplado ou segurado pelo usuário. Essa é uma área de pesquisa ainda não muito desenvolvida, mas tem crescido atualmente, especialmente entre pesquisadores informais, graças ao surgimento de novas tecnologias de processamento e captura de imagens, como indica o sucesso do Microsoft Kinect.

Ao se lidar com o reconhecimento de gestos, no entanto, é preciso ter alguns cuidados (ASHBROOK; STARNER, 2010), de forma que os gestos definidos não conflitem com os gestos executados cotidianamente. Essa restrição é importante para que o usuário não execute algum comando indesejado ao coçar o queixo, por exemplo.

Complementando o que já foi apresentado sobre o trabalho de Nancel *et al.*, em contrapartida aos gestos executados com um mouse 3D, os autores também avaliaram os usuários ao realizar as mesmas tarefas sem o uso de qualquer dispositivo. Não é explicado o sistema utilizado para detectar os gestos, mas o usuário se utiliza das duas mãos para transladar e dar zoom em imagens em uma tela grande (NANCEL et al., 2011). Como pode ser visto na figura 2.11, provavelmente são utilizadas várias câmeras para fazer a captura dos movimentos.

Muito embora a interação sem dispositivos tenha tido desempenho inferior à da que utilizou o mouse 3D, os autores apontam que se trata de uma abordagem interessante, pois teve apenas uma pequena defasagem temporal em relação à execução com o mouse e chamou a atenção dos usuários, justamente pelo fato de não haver nenhum dispositivo diretamente envolvido.

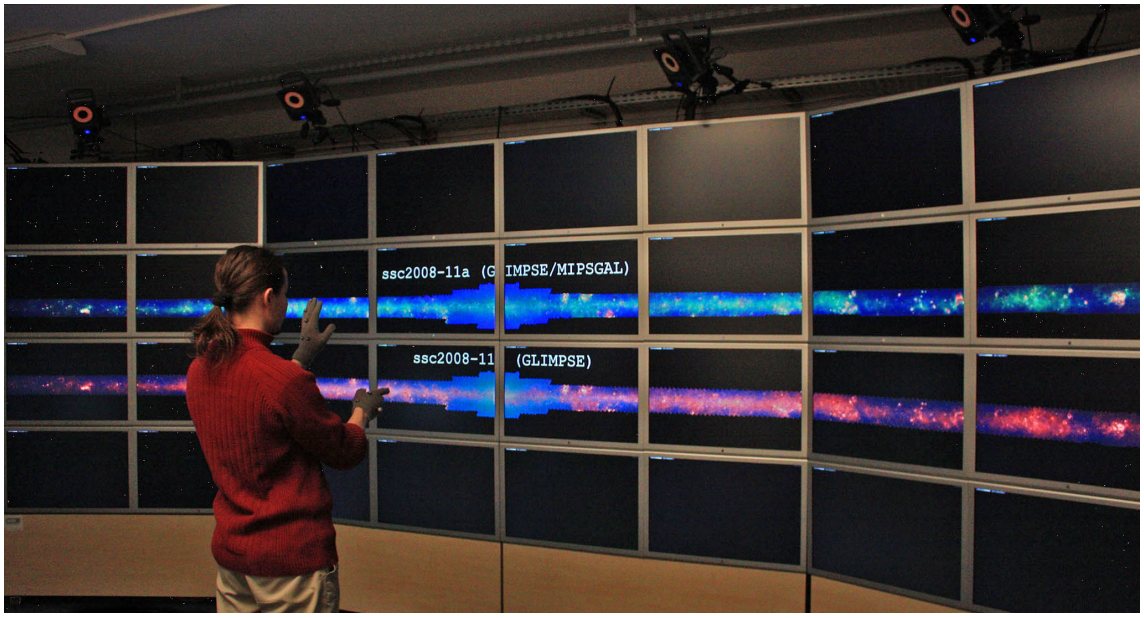

Figura 2.11: Usuário interagindo com uma tela de alta resolução sem utilizar nenhum dispositivo diretamente (sem segurar ou tê-lo afixado em si) (NANCEL et al., 2011).

2.2 Interação utilizando o Kinect

Ainda que possam ser encontradas na Web diversas aplicações que se utilizam do Microsoft Kinect para interação, poucas delas são trabalhos de pesquisa científica. Em sua maioria, essas aplicações são fruto da experiência de entusiastas em interação ou o resultado de um "passatempo de fim de semana", podendo ser conferidas apenas em vídeos no YouTube ou postagens em blogs. Apesar disso, alguns trabalhos científicos têm surgido nos dois últimos anos, que utilizam-se do Kinect para executar tarefas interativas ou, ainda, para o reconhecimento de humanoides e extração de modelos tridimensionais.

Um exemplo deste último é o trabalho de Tong *et al.*, que utilizou-se de 3 Kinects para gerar modelos tridimensionais de alguns usuários. Para que um Kinect não interferisse com os demais, o sistema proposto pelos autores utilizou um Kinect para o reconhecimento do torso do usuário, um para o reconhecimento das pernas e um terceiro para reconhecer a região pélvica, colocado no lado oposto dos dois primeiros, conforme mostra a figura 2.12. Em um tempo médio de 5,9 minutos, o método descrito reconhece o usuário, aplica diversos algoritmos para unir as imagens de profundidade geradas, resolver problemas de oclusão, alinhar as três partes do corpo e, por fim, gera uma malha de triângulos bastante interessante com um custo muito baixo em comparação com outras técnicas (TONG et al., 2012).

Apesar dos resultados obtidos, os autores destacam que a baixa resolução das câmeras do Kinect impede a geração de um modelo mais sofisticado, ainda que, dado o baixo custo para a criação do sistema (aproximadamente US$ 600,00), os resultados sejam bastante satisfatórios.

Outra abordagem para o reconhecimento de humanos (porém sem a geração de modelos 3D) é a apresentada por Xia e Balakrishnan, na qual a informação de profundidade lida por um Kinect é utilizada para reconhecer humanoides. Após o processamento da imagem de profundidade capturada, o sistema proposto é capaz de detectar a presença de humanos em 98,4% dos casos, sem falsos positivos. O não reconhecimento se deve,

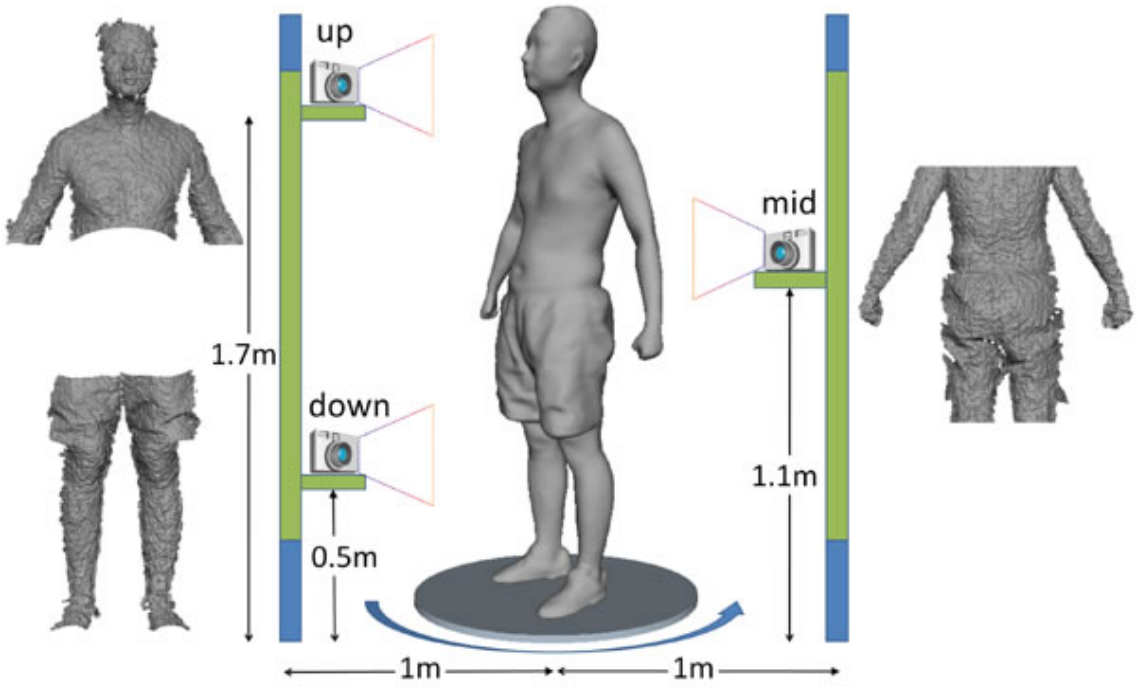

Figura 2.12: Sistema com 3 Kinects colocados a uma curta distância do usuário de forma a um não interferir com outro (TONG et al., 2012).

principalmente, ao fato de a metodologia utilizada depender muito da correta detecção da cabeça, não obtendo sucesso quando a cabeça do usuário está ocludida ou quando este está utilizando um chapéu ou um penteado extravagante (XIA; CHEN; AGGARWAL, 2011).

Voltando-se para uma aplicação interativa do Kinect, pode-se citar o trabalho de Bigdelou *et al.*, que utilizou o dispositivo para que usuários interagissem com um visualizador de imagens médicas. O sistema proposto reconhece até 16 gestos previamente gravados em uma fase de treino e calibragem do aplicativo. Os autores buscaram construir um módulo independente que poderia ser acoplado em qualquer aplicativo (BIGDELOU et al., 2012). A figura 2.13 mostra 3 passos de um usuário interagindo com um visualizador de imagens médicas, realizando uma translação vertical.

Figura 2.13: Usuário realizando um gesto para transladar verticalmente uma imagem, erguendo o braço direito (BIGDELOU et al., 2012).

Os autores realizaram testes com usuários e concluíram que é possível alcançar mais de 90% de precisão no reconhecimento de gestos quando há 8 gestos reconhecíveis diferentes. O sistema utiliza-se de comandos de voz para ativar ou desativar a leitura dos

gestos, evitando, assim, que gestos sejam reconhecidos sem que haja a intenção. Essa é uma preocupação pertinente nesses casos, onde gestos cotidianos podem ser confundidos com gestos reconhecidos pelo interpretador (ASHBROOK; STARNER, 2010).

Outro trabalho que utilizou o Kinect como instrumento de interação em um visualizador de imagens médicas é o de Gallo *et al.*. Em sua abordagem, o usuário passa por uma fase de calibração, na qual são extraídos dados como o comprimento do braço, a área da palma da mão, a área da mão cerrada e qual sua mão dominante. A partir de então, o usuário pode realizar gestos para fazer zoom, translação e rotação, entre outros, no aplicativo de visualização. Assim como o sistema anteriormente descrito, este também foi desenvolvido de forma a ser um módulo independente e foi integrado com a ferramenta de visualização via OpenCV [4] (GALLO; PLACITELLI; CIAMPI, 2011).

O trabalho de Wilson possui abordagem diferente, utilizando o Kinect para simular uma tela sensível ao toque, semelhante ao trabalho que analisou o uso de jogos em um grande display, já mencionado (STØDLE et al., 2008). Com o uso do Kinect o tempo de latência de leitura da câmera é reduzido, pois os dados já são interpretados pela SDK do aparelho, precisando ser feito apenas um pós-processamento desses dados. O sistema proposto pelo autor utiliza o Kinect afixado a uma certa altura de uma superfície não necessariamente plana (WILSON, 2010). O usuário interage, então, realizando gestos sobre a superfície, como visto na figura 2.14.

Figura 2.14: Usuário interagindo sobre uma superfície: à esquerda, a disposição do sistema, com o Kinect colocado a uma certa altura da superfície; à direita a imagem capturada pelo Kinect, que será interpretada pelo programa (WILSON, 2010).

Apesar de não apresentar detalhes quanto ao tempo de latência da interpretação do gesto, o autor reforça que o sistema proposto não é tão eficiente quanto telas sensitivas convencionais, mas destaca como pontos positivos da técnica a possibilidade de detectar gestos realizados sem haver contato direto com a superfície e também de verificar se dois ou mais toques simultâneos pertencem ao mesmo usuário.

Uma outra aplicação do Kinect foi apresentada por Oikonomidis *et al.*, na qual as imagens produzidas pela câmera de profundidade do Kinect são utilizadas para reconhecer a mão completa de um usuário, com todas as suas articulações. Para isso, o Kinect

[4] http://opencv.org

fica posicionado muito próximo à mão do usuário e são executados diversos algoritmos para processar as imagens lidas. O sistema proposto reconhece com bastante precisão a mão do usuário, chegando a 15Hz em processamento por GPU (IASON OIKONOMIDIS; ARGYROS, 2011).

Conforme pode ser visto, quase todos os trabalhos citados se utilizam da câmera de profundidade do Kinect para fazer o reconhecimento de padrões, mostrando que essa é uma abordagem promissora. Muito embora trabalhos supracitados (BIGDELOU et al., 2012) (GALLO; PLACITELLI; CIAMPI, 2011) sejam interessantes e apresentem boas técnicas interativas, passos de treino e calibragem são inconvenientes, especialmente em se tratando de displays públicos, quando o usuário não irá interagir caso o sistema não seja muito simples. Com base nisso e no que foi apresentado na primeira metade desse capítulo, um novo modelo pode ser definido.

3 UM ESTUDO SOBRE INTERAÇÃO GESTUAL SEM DISPOSITIVOS

Seja por chamar a atenção em filmes de ficção científica ou por traçarem um exato paralelo com a interação natural diária dos seres humanos, dentre todos os métodos de interação não convencional, a interação por reconhecimento de gestos sempre se mostrou atraente para os pesquisadores. A interação gestual é particularmente útil quando o usuário está interagindo de pé frente a displays grandes, em situações onde não se tem – e nem é desejável – dispositivos como mouse e teclado para realizar a interação, pois tais dispositivos não se adequam a esse tipo de situação.

O presente trabalho traz uma abordagem de interação gestual sem dispositivos, ou seja, sem que o usuário interagente precise segurar ou ter nele acoplado quaisquer dispositivos. Essa abordagem é desejável em uma interação com displays públicos, quando o usuário irá interagir de pé frente a um grande display e não se deseja um compartilhamento de dispositivos e nem de mecanismos complicados para possibilitar a interação. Pensa-se que em uma situação ideal, o usuário se colocaria em frente ao display e naturalmente intuiria sobre o modo correto de interagir com ele. A abordagem utilizada pretende propor uma alternativa que busque se aproximar de tal panorama.

No modelo proposto, o usuário interage com suas mãos (qualquer uma delas ou as duas simultaneamente) posicionadas em frente ao corpo para realizar translações, zoom, seleções e manipulações de elementos na tela, tudo em um ambiente 2D. Para fins de diferenciação entre gestos cotidianos e gestos a serem reconhecidos no sistema, o usuário deve fechar suas mãos ao interagir. Não é necessário qualquer tipo de calibragem e o sistema identifica automaticamente quando um usuário está em frente à tela para interagir, focando-se sempre no usuário que está mais próximo ao display em uma área na qual consiga enxergar facilmente seu conteúdo, ou seja, sem que esteja imediatamente em frente à tela.

O usuário pode apenas navegar pelas informações na tela, sem modificá-las, quando estiver com as mãos abertas, sendo possível consultar informações que estejam escondidas (como *tooltips*, por exemplo). Fechando as mãos, o usuário passa a manipular as informações da tela, podendo selecionar objetos e transladar a tela ou um determinado objeto, quando estiver apenas com uma mão fechada, ou aumentar e diminuir o zoom da tela, quando com as duas mãos fechadas. A figura 3.1 ilustra isso graficamente.

Esses gestos foram escolhidos por traçarem um exato paralelo com tarefas do cotidiano. Por exemplo, quando se consulta uma determinada entrada em uma lista, é comum que se passe o dedo pelos elementos da lista até parar no elemento desejado, fazendo assim uma navegação na lista. Ao escolher um determinado produto no supermercado, estende-se o braço e fecha-se a mão sobre o objeto escolhido, fazendo uma seleção entre

Figura 3.1: Gestos suportados pelo modelo proposto: à esquerda, com as duas mãos abertas; ao centro, com uma das mãos fechadas; à direita, com as duas mãos fechadas.

os produtos. Ao se movimentar um elemento sobre uma mesa – o próprio mouse, por exemplo –, fecha-se a mão sobre ele e movimenta-se a mão fechada até a posição em que se deseja soltá-lo.

Somente as tarefas de translação e zoom (chamadas habitualmente de *pan & zoom*) não possuem paralelo em situações cotidianas. No entanto, é possível buscar embasamento em técnicas já consolidadas para essas atividades em dispositivos sensíveis ao toque, como *smartphones* e *tablets*. Para fazer zoom em uma foto utilizam-se dois dedos na tela, cujas posições se afastam para aumentar o zoom e se aproximam para diminuí-lo. Quando a foto passou por uma aplicação de zoom e não se enquadra totalmente na tela, utiliza-se um dedo cuja posição se altera de acordo com a translação que se quer realizar sobre ela. Sendo assim, os gestos propostos parecem bastante intuitivos e adequam-se aos gestos que o usuário já está acostumado a realizar, diminuindo, assim, o tempo de treinamento necessário à técnica proposta. A partir do fechamento e abertura das mãos, é possivel descobrir o que o usuário pretende através de uma máquina de estados, demonstrada na figura 3.2.

Afim de avaliar o modelo proposto, foram desenvolvidas três aplicações, de modo que fosse possível examinar com detalhe cada uma das funcionalidades providas. Cada uma delas será descrita em maiores detalhes nas três seções seguintes.

3.1 Árvore genealógica acadêmica

Essa foi a primeira aplicação desenvolvida, e apresenta um visualizador de grafos que exibe a árvore genealógica acadêmica do Programa de Pós-Graduação em Computação (PPGC) da UFRGS na forma de um grafo. Nesse grafo os nodos representam alunos e docentes e as arestas representam a relação de orientação entre eles. O sistema possui uma *tooltip* que informa o nome da pessoa a que se refere um nodo quando uma mão está sobre ele e, adicionalmente, uma barra lateral que apresenta informações adicionais, como quando a pessoa ingressou na Universidade, como pode ser visto na figura 3.3.

Os nodos e arestas possuem cores e formatos diferentes, que trazem informações agregadas: nodos quadrados representam homens, enquanto redondos representam mulheres; nodos vermelhos representam docentes e azuis discentes, exceto quando representa a pessoa acessando o sistema, caso em que aparece em verde; nodos que representam alunos de mestrado são sólidos, enquanto aqueles que representam doutorandos possuem um quadrado (ou círculo) branco em seu centro; arestas em cinza representam orientação, ao

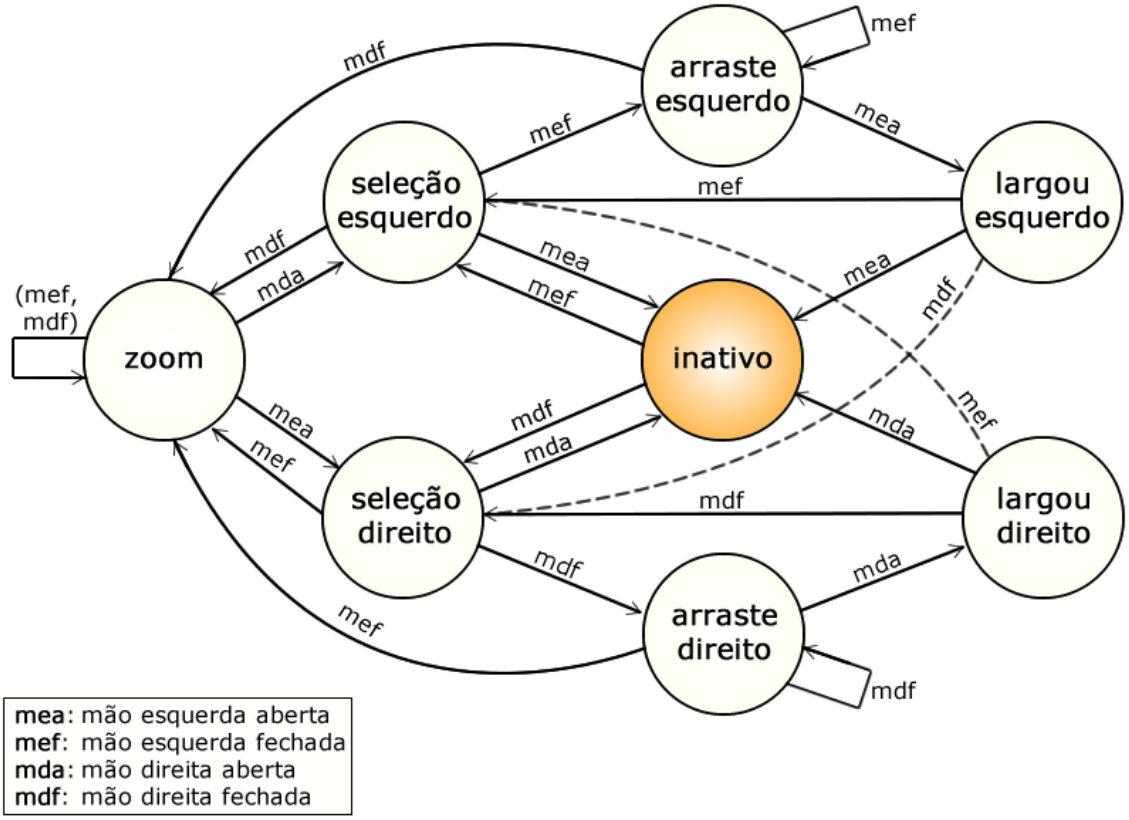

Figura 3.2: Máquina de estados das mãos conforme utilizada pela aplicação Web. De acordo com a configuração de cada mão, um estado diferente é detectado e a aplicação pode tomar uma ação específica. O estado inicial é o "inativo", indicado pelo círculo de tom alaranjado. As linhas tracejadas são assim desenhadas apenas para fins de deixar a imagem mais limpa.

passo que as em marrom indicam co-orientação; e, por fim, arestas contínuas representam orientações ou co-orientações ativas e as arestas tracejadas representam as concluídas. A figura 3.4 mostra a árvore genealógica completa do PPGC, juntamente com uma legenda para interpretar os dados exibidos.

Para construir essa árvore, os dados necessários foram obtidos diretamente da base institucional da Universidade, que contém registros de orientações de seus Programas de Pós-Graduação desde 2001. O grafo completo do PPGC possui 799 nodos e 836 arestas e é construído na aplicação de acordo com um modelo de forças potenciais. A finalidade desse modelo é posicionar os nodos de um grafo no espaço de modo que todas as arestas tenham comprimento aproximado (além de minimizar os cruzamentos entre elas). Isso é feito através da atribuição de forças entre os conjuntos de arestas e nodos baseando-se em suas posições relativas, e usando estas forças tanto para simular o movimento dos nodos e arestas ou minimizar a sua energia (KOBOUROV, 2012).

Nessa aplicação, o usuário pode consultar as informações agregadas aos nodos do grafo (nome da pessoas na *tooltip* e informações adicionais na barra lateral) ao passar com a mão aberta sobre eles. Pode também fazer zoom no grafo, para observar com mais detalhes alguma região de interesse, dessa forma podendo observar com mais clareza as ligações que acabam se formando entre as pessoas do grafo, graças às relações de orientação e co-orientação que conectam os docentes. Além disso, é possível transladar a tela e reposicionar um nodo específico.

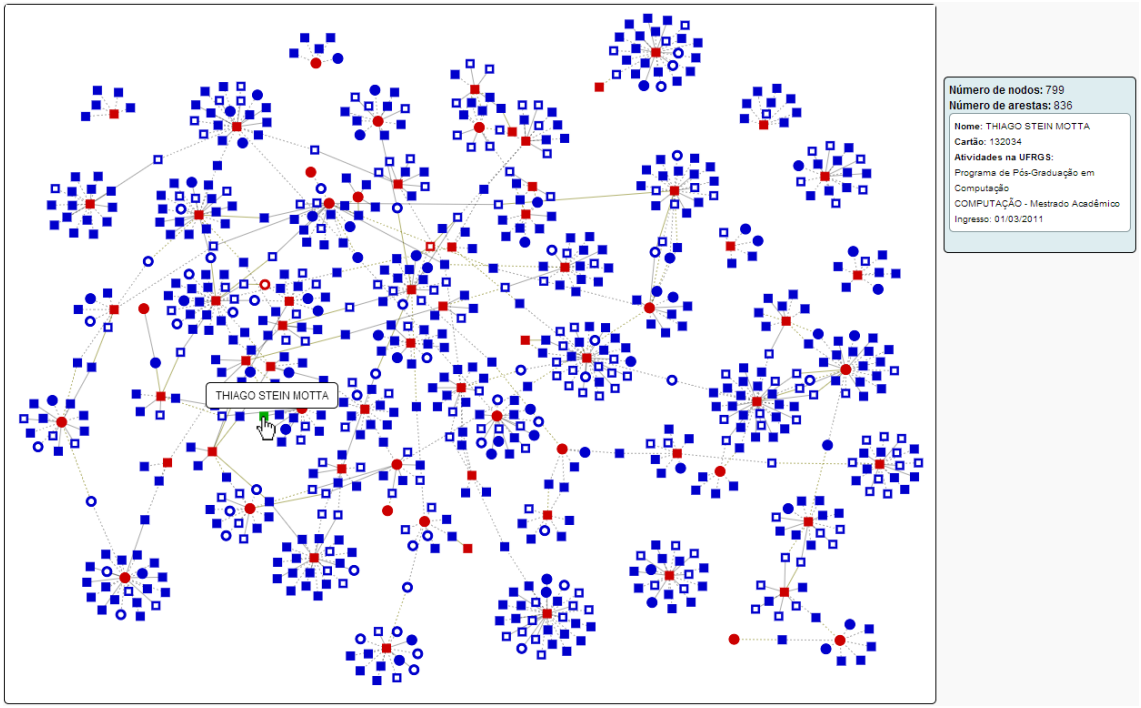

Figura 3.3: Tela inicial da aplicação de visualização da árvore genealógica, com a tooltip sobre o nodo indicado pelo mouse mostrando o nome da pessoa a que este se refere e a barra lateral com outras informações sobre ela.

Também é interessante observar na aplicação os ex-alunos de doutorado do Programa que passaram a ser orientadores. Na figura 3.5 é possível ver o caso de dois alunos: os ex-alunos estão indicados por um círculo verde; foram orientados por professores do PPGC, indicados por círculos vermelhos, conforme pode ser visto pela linha pontilhada cinza, destacada em amarelo; e agora co-orientam outros alunos, marcados com círculos azuis.

A figura 3.6, por sua vez, mostra uma aplicação de zoom no grafo, onde é possível ver, por exemplo, que a Profª Luciana Nedel está relacionada com o Prof. Anderson Maciel por uma orientação inativa sua que era uma co-orientação do professor. Por sua vez, o Prof. Anderson Maciel se relaciona com o Prof. Marcelo Walter por uma co-orientação ativa sua que é, ao mesmo tempo, uma orientação ativa do Prof. Marcelo. Logo, é possível afirmar que a Profª Luciana Nedel se relaciona com o Prof. Marcelo Walter através de dois discentes e um docente nesse contexto.

Todos os gestos descritos anteriormente podem ser utilizados nessa aplicação: mãos abertas para navegar pela tela e consultar informações a respeito dos nodos; seleção e manipulação dos nodos do grafo; e *pan & zoom* no grafo. Para que o usuário tenha certeza que o sistema está interpretando seus movimentos corretamente, novamente são exibidos os ícones das mãos abertas ou fechadas.

Essa aplicação foi construída com o intuito de ser utilizada em um display público colocado em algum local relevante da Universidade, como o saguão do prédio da Reitoria, por exemplo. A aplicação, caso disponibilizada ao público, seria capaz de identificar a qual Programa de Pós-Graduação pertence ou pertenceu o usuário através da leitura de seu cartão de identificação da UFRGS. A partir disso, montaria o grafo correspondente à árvore genealógica acadêmica deste Programa.

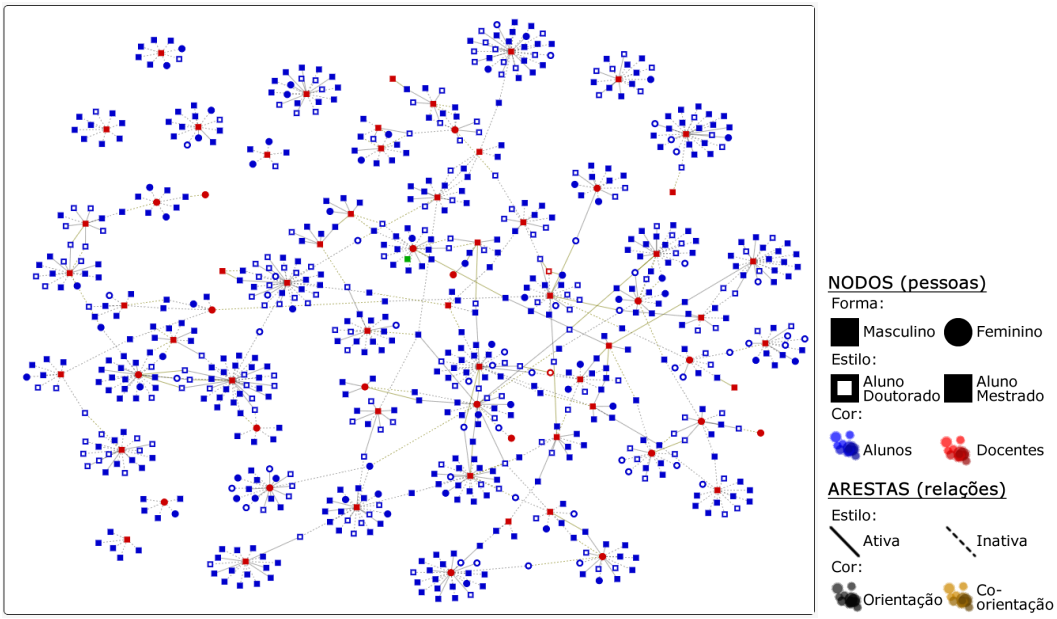

Figura 3.4: Árvore genealógica acadêmica do PPGC na forma de um grafo: nodos representam discentes e docentes e arestas representam relações de orientação. À direita, a legenda de como interpretar os dados do grafo.

3.2 Seleção e manipulação de objetos simples

Essa aplicação foi desenvolvida para a condução de experimentos com usuários. Ela apresenta uma série de pequenas tarefas que devem ser cumpridas pelo usuário, compreendendo especialmente seleção e manipulação de objetos. Primeiramente são exibidos seis quadrados na tela, sendo um deles de cor verde, indicando que ele deve ser selecionado pelo usuário. Após sua seleção, um outro quadrado dentre os seis fica verde e esse será o próximo que o usuário terá de selecionar. Esse procedimento se repete por cinco vezes e, então, o tamanho dos quadrados diminui pela metade, o que possibilita que eles apareçam em maior número. Após cinco novas seleções, os quadrados voltam a diminuir de tamanho e, após mais cinco seleções, diminuem uma terceira vez.

Ao final dessas últimas cinco seleções, surgem na tela um quadrado verde do tamanho original (grande) e um quadrado vazado em preto, levemente maior que o verde. A partir de então, o usuário não deve apenas selecionar o quadrado verde, mas também posicioná-lo de forma que ele fique dentro do quadrado vazado preto. O tamanho desses quadrados também diminui a cada cinco posicionamentos, mas somente por duas vezes, e não três como na tarefa de seleção. Após o último posicionamento, uma mensagem é exibida, informando que a tarefa terminou.

Também é possível fazer *pan & zoom* em qualquer momento dessa aplicação, bastando, para isso, que nenhuma mão se feche sobre o quadrado que aparece em verde. Essa aplicação exibe o esqueleto do usuário em semi-transparência, além de ícones das mãos esquerda e direita, que indicam a posição em que as mãos do usuário se encontram, bem como se elas estão abertas ou fechadas. Na figura 3.7 é possível ver que o esqueleto do usuário é desenhado atrás dos quadrados, enquanto os ícones das mãos ficam sempre à frente. A figura mostra 6 etapas de uma execução dessa aplicação: na parte de cima, as tarefas de seleção – da esquerda para a direita, a tela inicial da aplicação, o tamanho diminuído dos quadrados após as primeiras cinco seleções e uma aplicação de zoom na

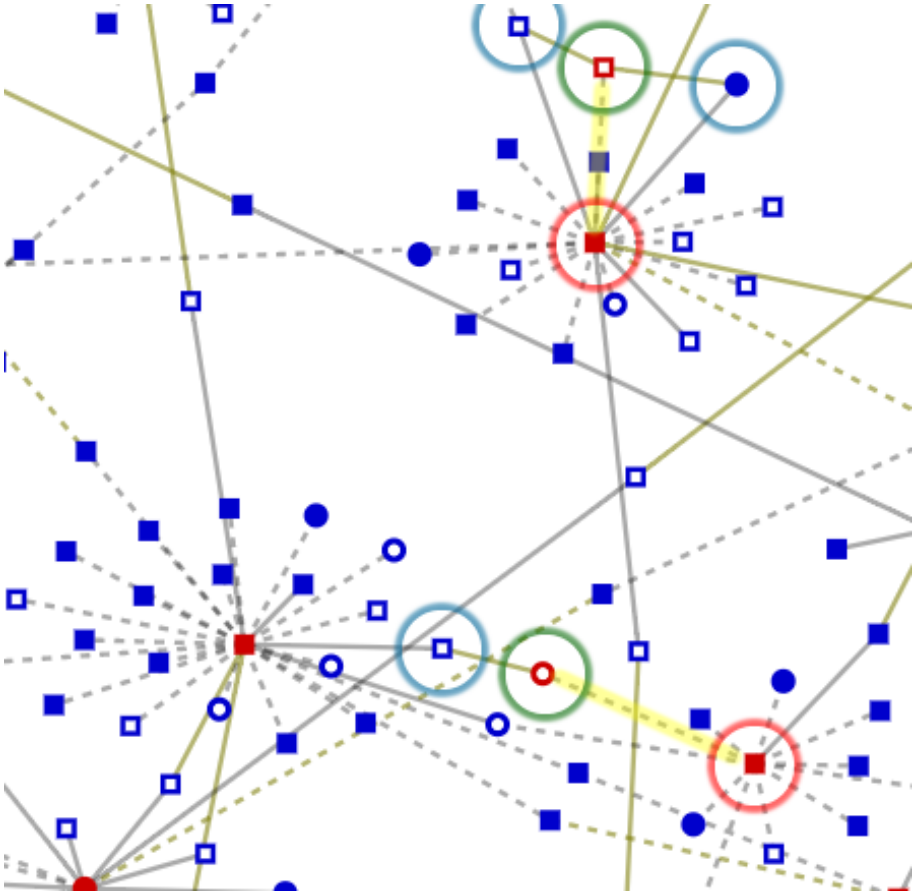

Figura 3.5: Outro detalhe na visualização da árvore genealógica acadêmica do PPGC mostra os antigos alunos de doutorado (marcados com círculos verdes) que atualmente são co-orientadores do Programa. Nos círculos vermelhos, os antigos orientadores destes alunos, ligados por uma linha pontilhada no detalhe em amarelo, e, nos azuis, seus atuais co-orientandos.

tela; abaixo, as tarefas de posicionamento – da esquerda para a direita, a tela inicial dessa etapa, o usuário manipulando um quadrado com sua mão esquerda e, por fim, fazendo o mesmo com a mão direita.

A título de responsividade para o usuário, os quadrados mudam de cor quando alguma das mãos passa por cima deles, passando para um vermelho claro, como pode ser visto na figura 3.7, no topo à direita. Além disso, quando o usuário seleciona o quadrado verde e permanece segurando-o, este muda para um tom vermelho escuro, como mostra a figura 3.7, no centro e na direita, em baixo. O usuário pode fazer zoom e transladar a tela o quanto desejar, mas a tela retorna para a posição central e em escala original cada vez que se inicia uma tarefa de posicionamento dos objetos.

Essa aplicação suporta os gestos de seleção, manipulação, translação da tela e aplicação de zoom na tela. O gesto para navegação também está presente, mas a única informação que exibe é a resposta visual acerca de qual quadrado está abaixo de cada mão, quando é o caso. A aplicação foi desenvolvida de modo que o grau de dificuldade das tarefas fosse aumentando a medida que o usuário fosse adquirindo prática com o sistema, o que foi comprovado após a aplicação de testes com usuário, como será visto no capítulo 6.

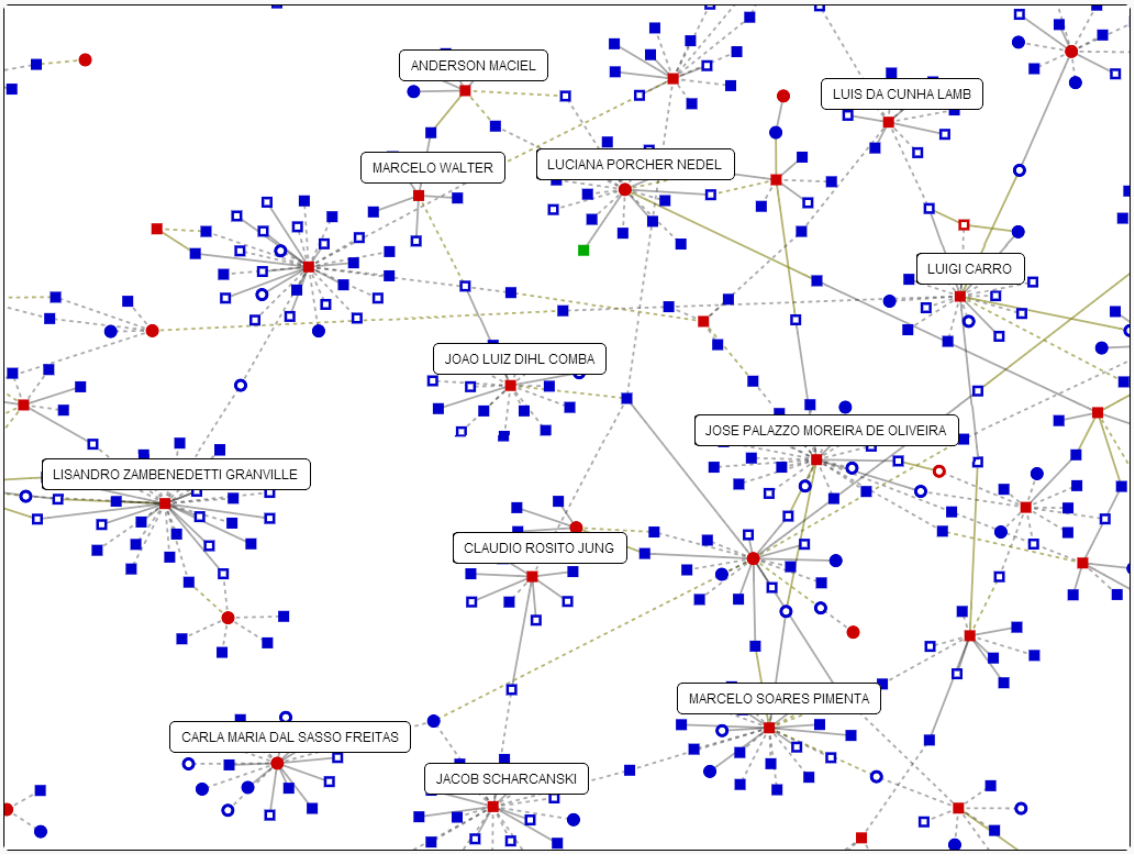

Figura 3.6: Detalhe na visualização da árvore genealógica acadêmica do PPGC, mostrando as inter-relações que se formam entre os nodos do grafo, com os nomes de alguns professores conforme aparecem na tooltip do aplicativo.

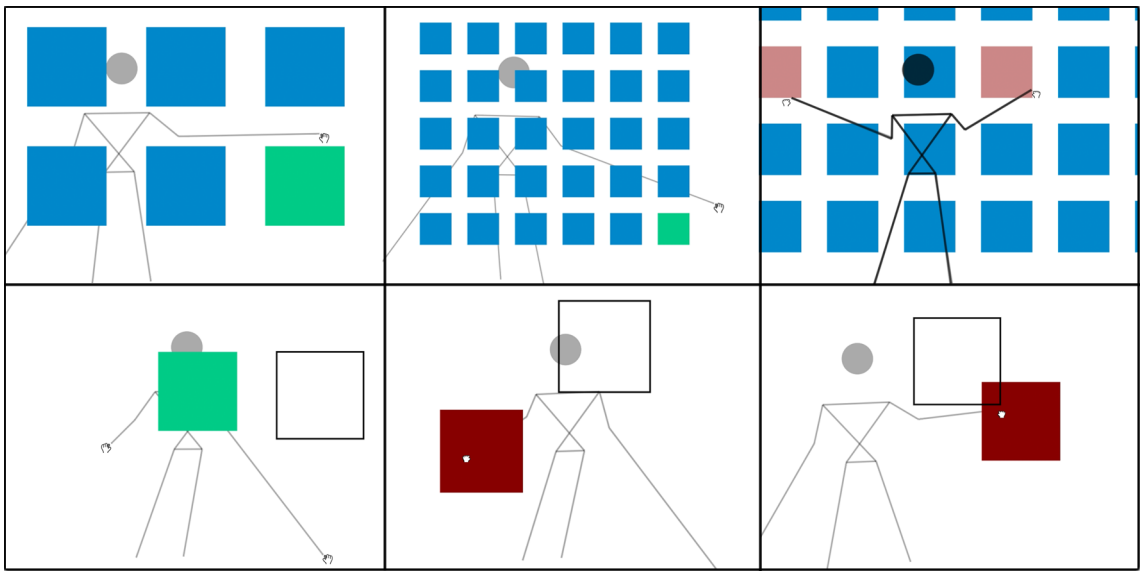

Figura 3.7: Capturas de tela de uma execução da aplicação de selecionar e posicionar objetos simples: no topo, à esquerda a tela inicial da tarefa de seleção de objetos, ao centro um novo tamanho de quadrados e, à direita, uma aplicação de aumento de zoom pelo usuário; embaixo, à esquerda a tela inicial da tarefa de posicionamento de objetos, ao centro e à direita, o usuário movimenta um quadrado com a mão esquerda e direita, respectivamente.

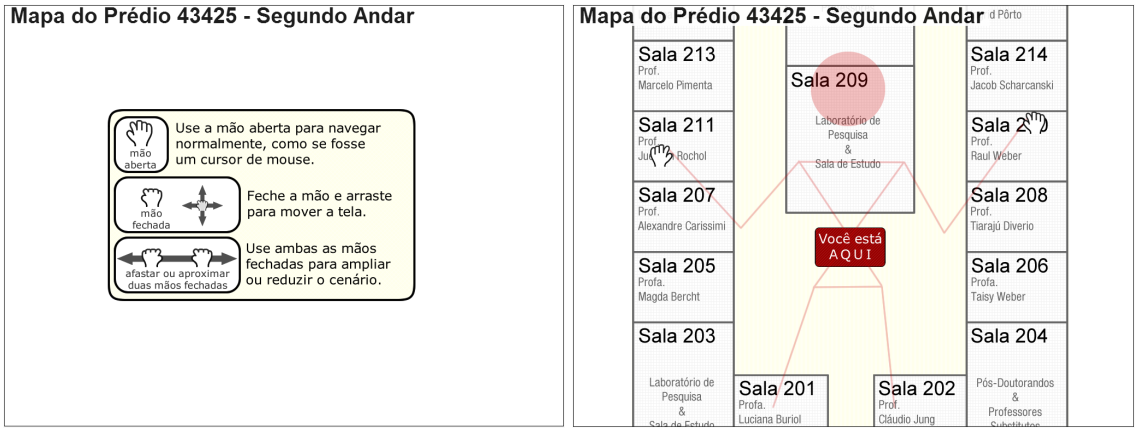

Figura 3.8: Duas telas da aplicação de visualização do mapa do prédio do Instituto de Informática: à esquerda a tela inicial da aplicação, quando não há a presença de nenhum usuário em frente à tela – é exibida uma janela mostrando quais gestos podem ser executados no sistema; à direita a tela conforme aparece passados 9 segundos depois que há um usuário em frente à tela – em vermelho semi-transparente é exibido o esqueleto do usuário, com ícones das mãos nas pontas de cada braço.

3.3 Mapa de localização de um prédio

Essa aplicação foi construída para analisar como as pessoas reagem a um display público interativo. Ela apresenta o mapa do prédio do Instituto de Informática da UFRGS onde estão localizados gabinetes de alguns professores e laboratórios de pesquisa. O mapa apresenta todas as salas e corredores do prédio, com indicativo dos números das salas e os nomes dos docentes que nela mantém seus gabinetes, quando é o caso. Possui também um indicativo do local onde se encontra o display público que exibe a aplicação, servindo para o usuário como um localizador.

O mapa só é carregado quando é detectada a presença de um usuário em frente ao display. Enquanto um usuário não é detectado, uma janela com instruções de uso do sistema é exibida, conforme pode ser visto na figura 3.8 à esquerda. Quando um usuário é detectado, o sistema continua exibindo a janela de instruções por 9 segundos, mas também passa a exibir o mapa do prédio e o esqueleto semi-transparente do usuário atrás dessa janela. Ao final dos 9 segundos, a janela desaparece e o usuário fica livre para interagir com a aplicação, conforme mostra a figura 3.8 à direita.

Essa aplicação mostra o esqueleto do usuário para que este possa se localizar junto ao sistema, bem como os mesmos ícones das mãos já utilizados na aplicação anterior, que aparecem ao final dos braços do esqueleto. A aplicação reconhece os gestos definidos para aumentar e diminuir o zoom e para transladar o mapa, sendo possível, desta forma, percorrer o mapa do prédio na tela e diminui-lo e aumentá-lo de acordo com o desejo do usuário.

4 PROJETO E IMPLEMENTAÇÃO

A construção desse trabalho passou por uma longa etapa de planejamento, na qual decisões precisaram ser tomadas. Em virtude dessas decisões, o modelo proposto foi desenvolvido em um ambiente Web e utilizando o Kinect como dispositivo interativo. Entretanto, não existe uma forma direta de integrar o Kinect com uma página de Internet, pois os navegadores, para fins de segurança, não possuem acesso às portas de entrada e saída de um computador. Assim, para que tal integração fosse possível, o sistema proposto foi definido em duas partes, seguindo os moldes de um sistema cliente-servidor, que se comunica por troca de mensagens.

As decisões de projeto são apresentadas na seção 4.1, bem como detalhes sobre as tecnologias escolhidas para a construção do sistema proposto. Em seguida, para fins de clareza, os detalhes de implementação de cada uma das partes do sistema serão apresentados em seções distintas, com uma terceira seção descrevendo a maneira como foi feita a integração do Kinect ao browser.

4.1 Decisões de Projeto

Desde o princípio da elaboração desse trabalho, desejava-se criar um modelo de interação que pudesse ser facilmente implantado em qualquer local. Para isso, ele precisaria ser de baixo custo financeiro e o mais portável possível. Além disso, como já foi frisado, era desejável que a interação não necessitasse de qualquer acoplamento ou manipulação de dispositivos, dando ao usuário a liberdade de se mover conforme desejasse e sem problemas de compartilhamento de objetos.

Tendo isso em mente, o Microsoft Kinect se mostrou a melhor alternativa, pois possui um custo bastante baixo e, com a ajuda de um SDK, fornece dados úteis de uma maneira bastante simples. Conforme visto nos trabalhos relacionados, o dispositivo tem bastante potencial para prover uma interação sem dispositivos, desde que suas limitações sejam contornadas. Sendo um dispositivo pequeno, sua colocação acima de uma tela grande não representaria um problema. Na subseção 4.1.1, as capacidades do Kinect são apresentadas em maiores detalhes.

Definido o hardware necessário para a interpretação dos gestos, restava a definição da interface com o usuário, que teria de ser desenvolvida de acordo com os critérios de portabilidade desejados. Com a evolução das funcionalidades de aplicativos para Internet, especialmente após o surgimento do padrão HTML5, uma abordagem Web mostrou-se interessante. É bem estabelecido – especialmente com o aumento das aplicações "em nuvem" – que sistemas para Internet possuem fácil acesso e são portáveis para diversas plataformas, portanto, a escolha dessa abordagem pareceu apropriada. Ademais, com uma abordagem Web, a integração com a base de dados Institucional para obtenção dos

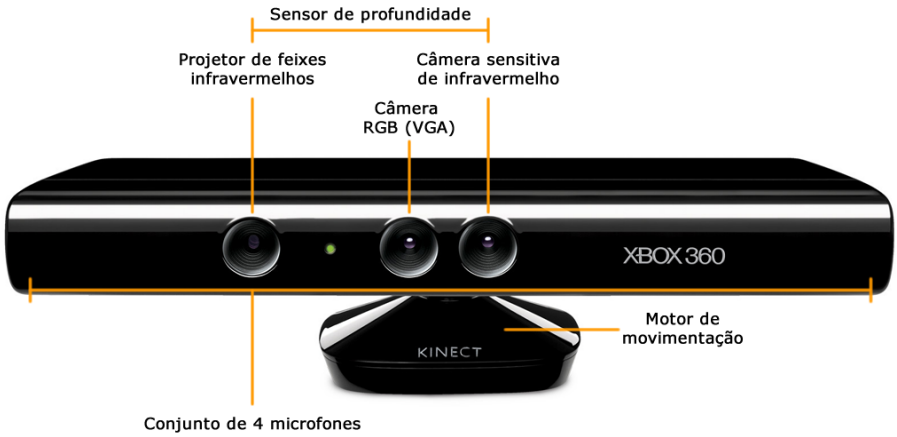

Figura 4.1: O dispositivo Kinect, com indicações de seus componentes utilizados para interpretação de gestos e de voz.

dados necessários à construção das árvores genealógicas acadêmicas fica facilitada, já que a UFRGS mantém seus dados gerenciáveis e acessíveis através da Internet. A subseção 4.1.2 apresenta maiores detalhes sobre essa tecnologia.

4.1.1 Microsoft Kinect

Lançado no final de 2010, o Kinect é um dispositivo de captura de movimento que permite aos usuários interagir com jogos e outras aplicações sem a necessidade de ter uma ligação física com qualquer dispositivo, como mouse, teclado ou *joystick*. Isto é obtido através do rastreamento do corpo do usuário, do reconhecimento de gestos e de voz. O dispositivo é composto por uma câmera normal, com resolução VGA, um projetor de feixes infravermelhos e uma câmera sensitiva de radiação infravermelha, bem como um conjunto de quatro microfones distribuídos pela extensão do dispositivo, que são capazes de suprimir ruídos, conforme mostra a figura 4.1. O Kinect entrou para o livro dos recordes como o dispositivo eletrônico mais rapidamente vendido, depois de vender oito milhões de unidades em apenas 60 dias[1].

A fim de adquirir informações de profundidade da cena, o Kinect projeta padrões de luz IR e calcula a profundidade de cada pixel da imagem produzida pela sua câmera sensitiva utilizando a deformação destes padrões na cena captada. Devido à utilização de luz infravermelha, o Kinect funciona em todas condições de iluminação em locais cobertos, seja em completa escuridão ou em um local bem iluminado, mas não funciona em um local exposto ao sol ou iluminado por lâmpadas incandescentes muito potentes. Este é um fator importante, pois em locais públicos não se possui controle sobre a iluminação. O dispositivo ainda é capaz de rastrear e extrair dados do esqueleto para dois usuários ativos, simultaneamente.

O campo de visão horizontal do Kinect é de 57 graus, enquanto que o vertical é de 43 graus. O sensor de profundidade funciona corretamente entre distâncias entre 1,2m e 3,5m. As imagens produzidas pelo sensor de profundidade possuem resolução de 320x240 pixels a 30fps. A câmera RGB produz imagens com resolução de 640x480 pixels com 32 bits (padrão VGA) e funciona também a 30fps. Por fim, os microfones captam audio a 16 bits em uma frequência de 16 kHz. O dispositivo ainda possui um

[1] http://en.wikipedia.org/wiki/Kinect

pequeno motor em sua base para a movimentar as câmeras no sentido vertical de modo a captar com maior precisão o usuário, conforme sua altura e posição.

4.1.2 Web 2.0 e HTML5

A Internet é um excelente ambiente para suportar aplicações, especialmente porque, graças à evolução de tecnologias para Web – como a linguagem JavaScript e, mais recentemente, com a introdução e popularização do padrão HTML5 –, é possível desenvolver para a Internet quase qualquer aplicação que antes só era possível em sistemas locais. Além disso, na rede o acesso a um mesmo aplicativo e a uma mesma fonte de dados pode ser realizado simultaneamente em locais completamente distintos e sem qualquer ligação adicional entre eles, desde que possuam uma conexão à Internet.

A Web 2.0 é marcada pela colaboração entre os usuários (BENKLER, 2011), em um ambiente onde as aplicações procuram ser tão interativas quanto possível. Nessa constante busca por inovação, surgem novas tecnologias quase que diariamente, com grandes empresas da rede disputando entre si para atrair mais usuários, que, cada vez mais, anseiam por novidades. Foi assim que surgiram os círculos do Google+[2] e a linha do tempo do Facebook[3], por exemplo, que mostram o quão dinâmica pode ser uma página de Internet.

O padrão HTML5, por sua vez, foi construído com a ideia de integrar o conteúdo exibido nos browsers em plataformas diferentes, como um computador e um *tablet*, por exemplo. Em particular, o HTML5 adiciona várias novas funções sintáticas, incluindo marcações para áudio e vídeo e, em especial, elementos <*canvas*>, destinados a delimitar uma área para renderização dinâmica de gráficos bi ou tridimensionais, bem como a integração de conteúdos que substituem o uso de marcações <*object*>. Estas funções são projetadas para tornar mais fácil a inclusão e a manipulação de conteúdo gráfico e multimídia na Web, sem ser necessário recorrer a *plugins* proprietários e APIs[4].

Com o uso do elemento <*canvas*>, é possível criar modelos gráficos unicamente com HTML e JavaScript, possibilitando que aplicações desenvolvidas em OpenGL, por exemplo, sejam transpostas para um ambiente Web, agregando, assim, portabilidade, já que qualquer computador com um navegador atualizado instalado poderia executá-la.

4.2 Capturando dados do Kinect

Antes de o Kinect ser lançado, em 2010, a Microsoft rechaçava qualquer suposição de que seu dispositivo poderia funcionar em um computador pessoal, tendo inclusive ameaçado processar quem quebrasse a encriptação da transferência de dados do Kinect, tornando possível sua utilização no PC. Paralelamente a isso, a empresa Adafruit[5] oferecia uma recompensa para quem produzisse um driver para o dispositivo antes mesmo de seu lançamento. Um dia após o lançamento, o driver foi construído e, futuramente, originou o primeiro SDK para desenvolvimento com o Kinect no PC: a OpenKinect[6]. Tempos após, descobriu-se que o financiador do prêmio para a quebra da encriptação do Kinect era a própria Microsoft[7] e que o dispositivo, afinal, possui sua conexão USB aberta.

O SDK da OpenKinect, apesar de cumprir com o que se propõe, é bastante limitado,

[2] http://plus.google.com
[3] http://facebook.com
[4] http://pt.wikipedia.org/wiki/HTML5
[5] http://www.adafruit.com/
[6] http://openkinect.org
[7] http://en.wikipedia.org/wiki/Kinect#Open_source_drivers

não possuindo uma leitura do esqueleto nativa. O primeiro SDK a suportar essa funcionalidade foi o desenvolvido pela OpenNI[8], utilizando drivers da própria Prime Sense[9], a empresa que efetivamente desenvolveu a tecnologia por trás do Kinect. Este SDK é bastante completo e estável, funcionando tanto em ambientes Linux quanto Windows ou OS X, e é o mais utilizado por entusiastas em projetos isolados. Ele permite acesso às imagens dos sensores de profundidade, às imagens da câmera RGB e ao esqueleto de até dois usuários, compostos por 20 juntas cada.

Tenha sido premeditadamente ou para explorar uma nova plataforma para o Kinect, a Microsoft lançou seu próprio SDK, o *Kinect for Windows*, em junho de 2011, bem como anunciou que lançaria uma nova versão de seu dispositivo específica para uso em PCs, homônimo do SDK, futuramente lançado em fevereiro de 2012 nos Estados Unidos. O SDK da Microsoft, hoje em sua versão 1.6, possui uma série de vantagens em relação a seus concorrentes. Além de fornecer acesso às imagens RGB e de profundidade e o esqueleto de 20 juntas (ver figura 4.2) de até dois jogadores, também possibilita o acesso ao sistema de reconhecimento de voz do dispositivo e ao seu motor de movimento[10] e, em especial, não necessita de uma pose de calibração do usuário para reconhecer seu esqueleto, como o SDK da OpenNI.

O único problema ao utilizar o SDK da Microsoft é que o mesmo foi desenvolvido para funcionar unicamente em ambientes Windows, portanto, afim de contar com todas suas vantagens, o sistema interpretador do Kinect foi desenvolvido para Windows, no Visual Studio 2010, em linguagem C#, a recomendada pela Microsoft para desenvolver com o Kinect. Essa escolha precisou ser feita para que fosse possível eliminar o passo de calibração, que teria de ser feito com o outro SDK. Conforme já foi afirmado, passos de calibração ou treino são enfadonhos para um usuário casual do sistema e devem ser evitados em aplicações para displays públicos, que têm como público alvo um grupo variado de usuários.

Para esse trabalho, foram utilizadas as informações de profundidade e do esqueleto do usuário. Inicialmente, o sistema deve detectar se existe algum usuário presente para interagir. Isso é facilmente detectado ao utilizar o evento do SDK que é disparado quando as informações do esqueleto estão prontas, porque, naturalmente, só haverá informações sobre o esqueleto do usuário se houver um usuário em primeiro lugar. Com as informações de juntas do esqueleto, descobre-se também em que posição se encontram as mãos do usuário. Para isso utilizam-se as juntas *HandRight* e *HandLeft*, que podem ser identificadas na figura 4.2. As coordenadas são interpretadas em 3D (x,y,z), então a informação de profundidade é descartada e o ponto obtido para cada mão (x,y) é escalonado para uma resolução base de 1024x768 pixels, utilizando-se, para isso, um método do próprio SDK.

Inicialmente, foi experimentado um método de interação que utilizasse apenas as informações de juntas e os ângulos que elas formam entre si. Dessa forma, diferenciava-se uma seleção de uma navegação quando o usuário tinha seus braços estendidos ou próximos ao corpo, respectivamente. Apesar de ter funcionado, a técnica era pouco precisa e exigia muito treinamento do usuário interagente, especialmente ao se realizar seleções de objetos pequenos.

Como nenhum SDK atualmente é capaz de reconhecer nativamente quando a mão do usuário está aberta ou fechada, um pós-processamento das informações lidas pelo Kinect precisa ser feito para obter essa informação. A ideia simples de que a área ocupada pela

[8] http://www.openni.org

[9] http://www.primesense.com/

[10] http://labs.vectorform.com/2011/06/windows-kinect-sdk-vs-openni-2

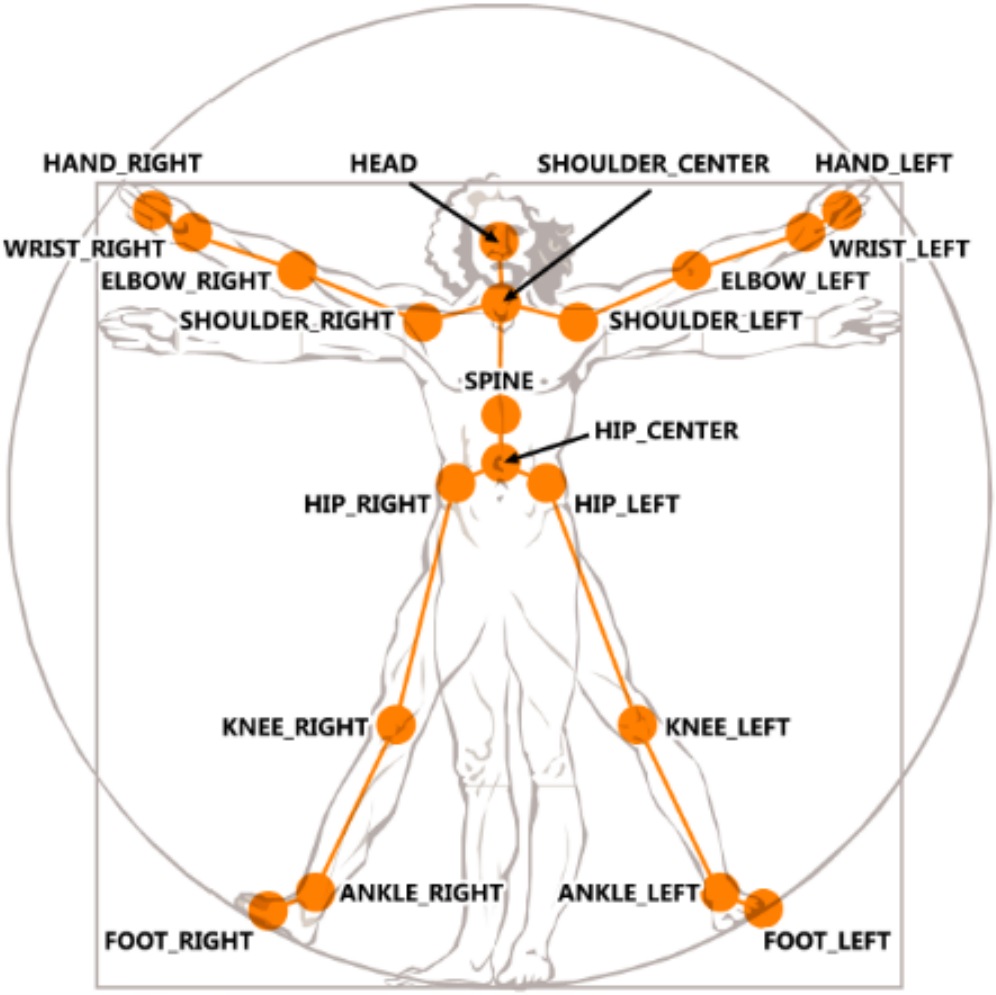

Figura 4.2: Conjunto de juntas do esqueleto que o Kinect consegue reconhecer no SDK *Kinect for Windows* da Microsoft.

mão aberta é maior que aquela da mão fechada (GALLO; PLACITELLI; CIAMPI, 2011) parece promissora, mas, como lida-se com um grupo muito heterogêneo de usuários, que possuem mãos de tamanhos diferentes, seria necessária uma calibração inicial do sistema, o que, conforme já frisado mais de uma vez, não é desejado. Para que o sistema funcione com qualquer usuário e sem um passo de calibragem, é preciso empregar algoritmos de processamento de imagens sobre a imagem RGB ou de profundidade obtida. Como a imagem de profundidade possui mais informações (ainda que tenha uma resolução menor), foi sobre ela que foram realizados os processamentos.

A interpretação das mãos fechadas é feita em três passos: primeiro, localiza-se e isola-se as mãos do usuário; segundo, identifica-se o contorno das mesmas; por fim, faz-se uma leitura do contorno seguindo o algoritmo K-curvature (SHAKER; ABOU ZLIEKHA, 2007), buscando por pontas de dedos. Se, ao final do processamento, alguma ponta de dedo foi encontrada, conclui-se que a mão em questão está aberta. Caso contrário, está fechada. Cada um dos passos será descrito em maiores detalhes abaixo.

4.2.1 Isolando as mãos

O primeiro passo necessário para descobrir o estado das mãos do usuário é descobrir, na imagem de profundidade, onde elas se localizam. Para isso, são utilizadas, para cada

mão, duas juntas do esqueleto obtido pelo Kinect: do pulso e da mão (na figura 4.2, *WristLeft*, *HandLeft*, *WristRight* e *HandRight*). Com a utilização de um método do próprio SDK, para cada uma das mãos, faz-se um mapeamento das coordenadas das juntas para coordenadas da imagem de profundidade, obtendo-se os pontos W(x,y) e H(x,y), representando o pulso e o centro da mão. Calcula-se a distância d entre esses pontos e define-se um quadrado de lado $2d$ centralizado no ponto H, que engloba toda a região ocupada pela mão na imagem de profundidade, conforme mostra a figura 4.4-A.

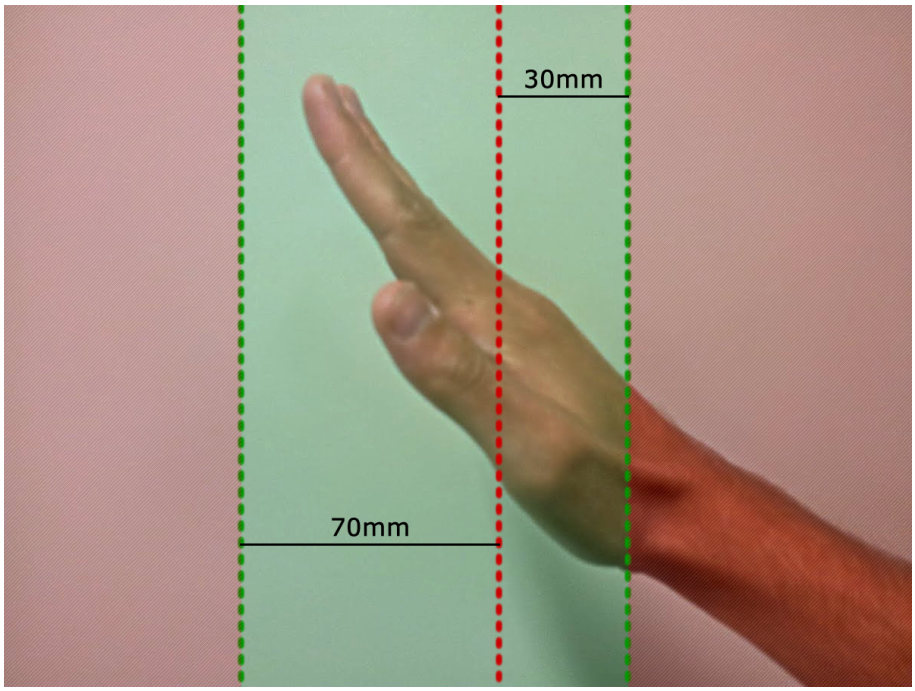

Figura 4.3: Gráfico mostrando o limite de profundidade que é testado para isolar a mão do usuário: a linha tracejada vermelha indica o índice de profundidade do ponto central da mão; as linhas tracejadas em verde indicam os limites de profundidade que são utilizados para fins de comparação.

Obtendo-se uma imagem quadrada que engloba toda a mão do usuário, conforme mostra a figura 4.4-B, utiliza-se a informação de profundidade, em milímetros, associada ao ponto H, do centro da mão, para comparar com todos os demais pixels da imagem. Como a mão se encontra a uma profundidade muito próxima do pulso e, mesmo, do braço, é preciso estabelecer um limite pequeno para que somente a mão seja selecionada de fato. Em contrapartida, à frente da mão não há nenhum outro membro e o corpo humano não permite uma inclinação de 90^{o} entre a mão e o braço, de forma que os dedos sempre ficarão mais próximos do Kinect do que a palma da mão, onde o centro se encontra, conforme pode ser observado na figura 4.3. Dessa forma, dois limites são definidos ao se fazer a comparação entre os pixels: *handDepthLimitFront*, para comparar com distâncias maiores que a mão, com valor 70mm; e *handDepthLimitBack*, para comparar com distâncias menores que a mão, com valor 30mm. Ao fazer a comparação de todos os pixels da imagem, cria-se uma nova imagem, na qual os pontos que se encontram dentro do limite estipulado ficam em branco e os demais em preto, como mostra a figura 4.4-C.

Figura 4.4: Isolamento das mãos do usuário na imagem de profundidade: A) um quadrado (em vermelho) é definido com base nas juntas do pulso e do centro das mão (pontos azuis); B) o recorte da mão direita da imagem de profundidade; C) após processamento da imagem, utilizando a informação de profundidade associada a cada pixel, isola-se a mão do resto da imagem.

4.2.2 Detectando o contorno das mãos

O próximo passo é detectar o contorno da mão que foi isolada. Esse procedimento é feito em dois passos específicos: primeiramente, a imagem é processada para que reste apenas o contorno da mão; após, a imagem passa por um novo processamento para que seja detectado quais pixels do contorno são adjacentes entre si. O segundo passo é necessário para o processamento do algoritmo K-curvature, que percorre o contorno da imagem partindo de um ponto e seguindo sempre para um ponto adjacente, não processando a imagem inteira.

Processar a imagem deixando apenas o contorno da mão visível é bastante simples. Para isso, basta percorrer a imagem pixel a pixel e analisar cada um deles com uma máscara de convolução 3x3, centralizada no pixel sendo lido. Se o pixel em questão for branco e qualquer dos outros pixels ao seu redor for preto, este é considerado um pixel de contorno, assim é mantido em branco. Caso qualquer um dos testes seja falso, o pixel é colorido de preto, como demonstra a figura 4.5. Ao final do processamento, somente o contorno da imagem permanece em branco.

A partir da nova imagem gerada, contendo apenas o contorno da mão, é preciso definir quais pontos desse contorno são adjacentes, para que seja possível executar o algoritmo para detecção das pontas dos dedos. Isso é feito com uma nova comparação de máscaras na imagem, porém analisando apenas os pixels de contorno da mão. Primeiro, define-se um *array* de pontos do contorno A. Em seguida, iniciando-se de um ponto qualquer do contorno, é feita uma comparação dos pontos vizinhos dentro de uma máscara 3x3. Se algum dos pixels vizinhos é branco e ainda não se encontra em A, adiciona-se o pixel ao *array* e centraliza-se a máscara nesse pixel, repetindo-se o processo. Caso nenhum dos vizinhos do pixel analisado seja um ponto branco que ainda não está em A, amplia-se a

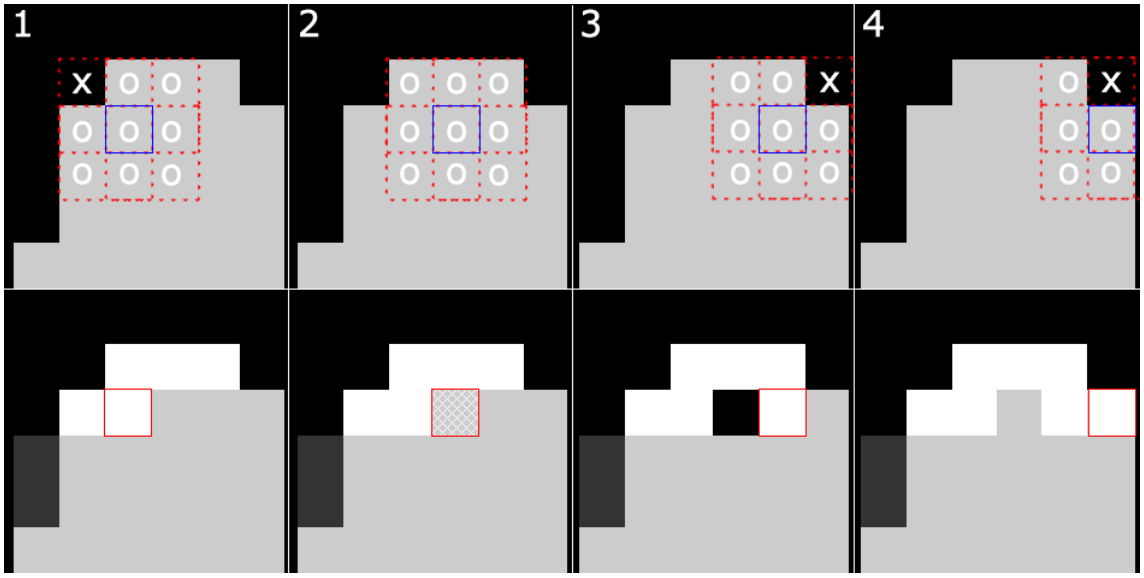

Figura 4.5: Quatro passos do processamento da imagem para extração do contorno: em cima, a imagem em processamento, com a máscara centralizada no pixel sendo lido (quadrado azul); em baixo, o resultado parcial do processamento, com os pixels já processados em branco e preto.

máscara de comparações para 5x5. Se, mesmo assim, não for encontrado um vizinho que se enquadre no teste, a máscara é ampliada mais uma vez para 7x7. Caso o teste falhe mais uma vez, conclui-se que todo o contorno foi percorrido e o algoritmo encerra.

É necessário testar o contorno com uma máscara de dimensão até 7x7 porque a resolução da imagem gerada pelo Kinect é muito baixa e pontos do contorno podem ficar espaçados de tal forma que uma máscara menor seja incapaz de percorrer todo o contorno da mão. A figura 4.6 mostra os diversos tamanhos de máscaras sobrepostas e identifica duas situações em que a leitura necessitaria de máscaras de tamanho 5x5 (à direita, em cima) e 7x7 (à direita, em baixo). Casos em que seria necessária uma máscara ainda maior podem ocorrer, mas considera-se que o tamanho máximo de 7x7 representa a melhor alternativa no que diz respeito a custos de processamento e à correta identificação do contorno.

Ao final dessa etapa, o *array A* contém, ordenadamente, os pontos de contorno da imagem e pode ser executado sobre ele o algoritmo para detecção de pontos convexos.

4.2.3 Descobrindo o estado das mãos

Finalmente, o último passo para definir se a mão lida está fechada ou aberta é procurar por pontas dos dedos, que representam descontinuidades no contorno da mão. Para isso, é utilizado o algoritmo K-curvature (SHAKER; ABOU ZLIEKHA, 2007), que é executado sobre o contorno de uma figura. Para execução do algoritmo é necessária a definição de duas constantes: *pointsInterval*, que define o intervalo em que os pontos de contorno serão lidos; e *limitAngleToBeFingertip*, que define o ângulo entre vetores que representará uma descontinuidade no contorno. No modelo proposto, esses valores foram definidos em 6 e 50, respectivamente, após uma série de testes com valores diversos.

Ao executar, o algoritmo recebe como entrada o *array A* que contém os pontos do contorno. O algoritmo inicia o processamento na posição *pointsInterval* do array e segue a intervalos definidos até que chegue ao final do array menos *pointsInterval* posições.

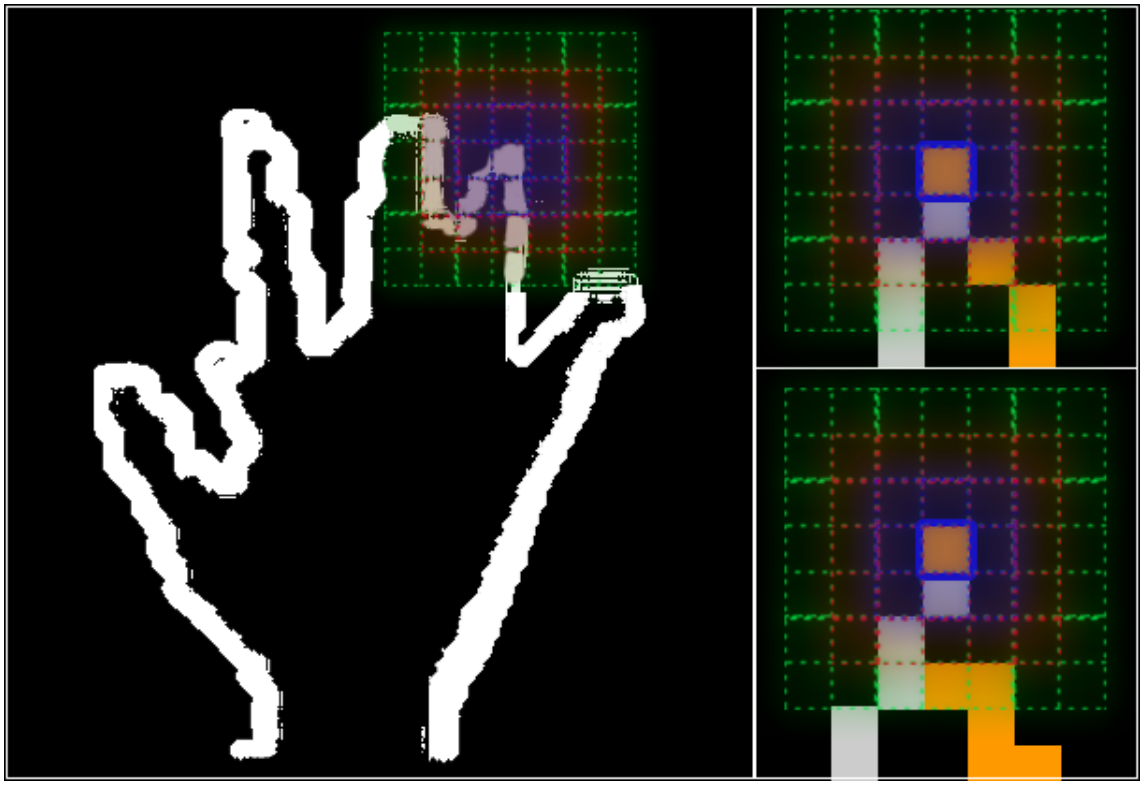

Figura 4.6: Processamento da imagem para localização dos pontos adjacentes do contorno das mãos: à esquerda, um ponto problemático para leitura; à direita, pixels em cinza representam pontos já inserido no *array* que armazena o contorno e alaranjados aqueles ainda não lidos. Em cima uma situação em que o próximo ponto só seria identificado com uma máscara 5x5 e, abaixo, situação em que seria necessária uma máscara 7x7.

Para cada posição *i* sendo analisada, é lida a posição *i-pointsInterval* e a *i+pointsInterval*. Então é calculado o ângulo que se forma entre os vetores formados por <*i-pointsInterval*, *i*> e <*i*, *i+pointsInterval*>. Se esse ângulo for menor que *limitAngleToBeFingertip*, a posição *i* corresponde a um ponto de descontinuidade. Abaixo um pseudo-código que realiza o processamento do algoritmo descrito, onde *contourPoints* é o *array* que contém o contorno da imagem avaliada.

Se ao final do processamento, tiver sido lido algum ponto de descontinuidade – que pode ser uma ponta de dedos ou a região entre dois dedos – constata-se que a mão está aberta. Um teste extra poderia ser realizado afim de determinar se o ponto de descontinuidade é realmente uma ponta de dedo, mas como esse ponto em si não é uma informação relevante, não é necessário fazê-lo. A figura 4.7 mostra uma execução do algoritmo sobre uma imagem. Pontos vermelhos circundados de branco são os pontos de descontinuidade detectados, sendo os formados pelos vetores em laranja os que representam pontas dos dedos e os em verde aqueles que marcam regiões entre dois dedos. Observa-se que nem todos os pontos de descontinuidade foram detectados, mas a detecção de apenas um deles é o suficiente para essa aplicação.

Afim de testar a funcionalidade do algoritmo, bem como dos passos anteriores necessários para identificação do estado das mãos, um aplicativo foi desenvolvido em C#, que mostra ao centro a imagem de profundidade sendo produzida pelo Kinect, com eventuais pessoas em amarelo, e em cada lado imagens de cada uma das mãos, com a identificação dos pontos de descontinuidade detectados, conforme mostra a figura 4.8.

Algorithm 1 Algoritmo K-curvature conforme usado neste trabalho.

```
numeroDeDescontinuidades = 0
for i = 0 to tamanhoArray(A)-pointsInterval do
    p1 = contourPoints[i - pointsInterval]
    p2 = contourPoints[i]
    p3 = contourPoints[i + pointsInterval]
    angulo = anguloEntreOsPontos(p1, p2, p3)
    if (angulo <= limitAngleToBeFingertip) then
        numeroDeDescontinuidades += 1
        i = i + pointsInterval
    end if
end for
```

Como movimentos rápidos das mãos podem fazer com que o sistema interprete incorretamente seus estados (por exemplo, uma mão aberta pode ser reconhecida como fechada, pois ao movimentá-la rapidamente, ela se torna um "borrão" na leitura da câmera), um pequeno *buffer* de 10 posições foi construído para armazenar os estados da mão como em uma fila, o qual é consultado e transmite o estado que se encontra em maior número entre suas posições. Isso gera um pequeno atraso na interpretação, mas compensa com uma estabilidade muito maior.

4.3 Integrando o Kinect ao navegador Web

Conforme já frisado, o Kinect não pode se comunicar diretamente com o navegador, pois este não possui acesso às portas do sistema computacional, por onde trafegam os dados do dispositivo. Para que uma integração seja possível, é preciso utilizar uma arquitetura do tipo cliente-servidor, na qual a aplicação que lê e interpreta os dados do Kinect age como servidor e se comunica via troca de mensagens com a aplicação cliente desenvolvida para a plataforma Web.

Para que essa comunicação ocorra, o mais indicado é a utilização de *WebSockets*, que são canais de comunicação *full-duplex* definidos e padronizados para troca de mensagens em servidores e navegadores Web, embora possa ser utilizado para qualquer tipo de aplicação. *WebSockets* possuem protocolos leves e de fácil utilização, trafegam em um canal de conexão TCP e utilizam, por padrão, a porta 80, sendo suportados em todos os navegadores modernos.

A aplicação que interpreta o Kinect deve abrir um *WebSocket* e aguardar conexões. Por sua vez, a aplicação Web deve se conectar ao *WebSocket* aberto e informar que está pronta para receber mensagens. A partir de então, a aplicação servidora deve enviar mensagens contendo dados úteis que foram interpretados a partir do processamento dos dados do Kinect. A mensagem é transmitida em formato JSON, que é um formato de dados específico de JavaScript e utilizado amplamente em páginas de Internet, por sua simplicidade e robustez. É transmitido, basicamente, um *array* contendo as seguintes informações: posição das juntas correspondentes à cabeça, a ambos os ombros, cotovelos, pulsos, pernas e mãos, duas juntas correspondentes à cintura e, em especial, dois inteiros que indicam se as mãos estão abertas (0) ou fechadas (1). Em posse desses dados, a aplicação Web pode interpretá-los e utilizá-los para fins diversos, empregando apenas os dados que lhe forem necessários. A figura 4.9 ilustra as comunicações envolvidas no sistema desenvolvido.

Essa abordagem é particularmente interessante porque torna a aplicação Web quase

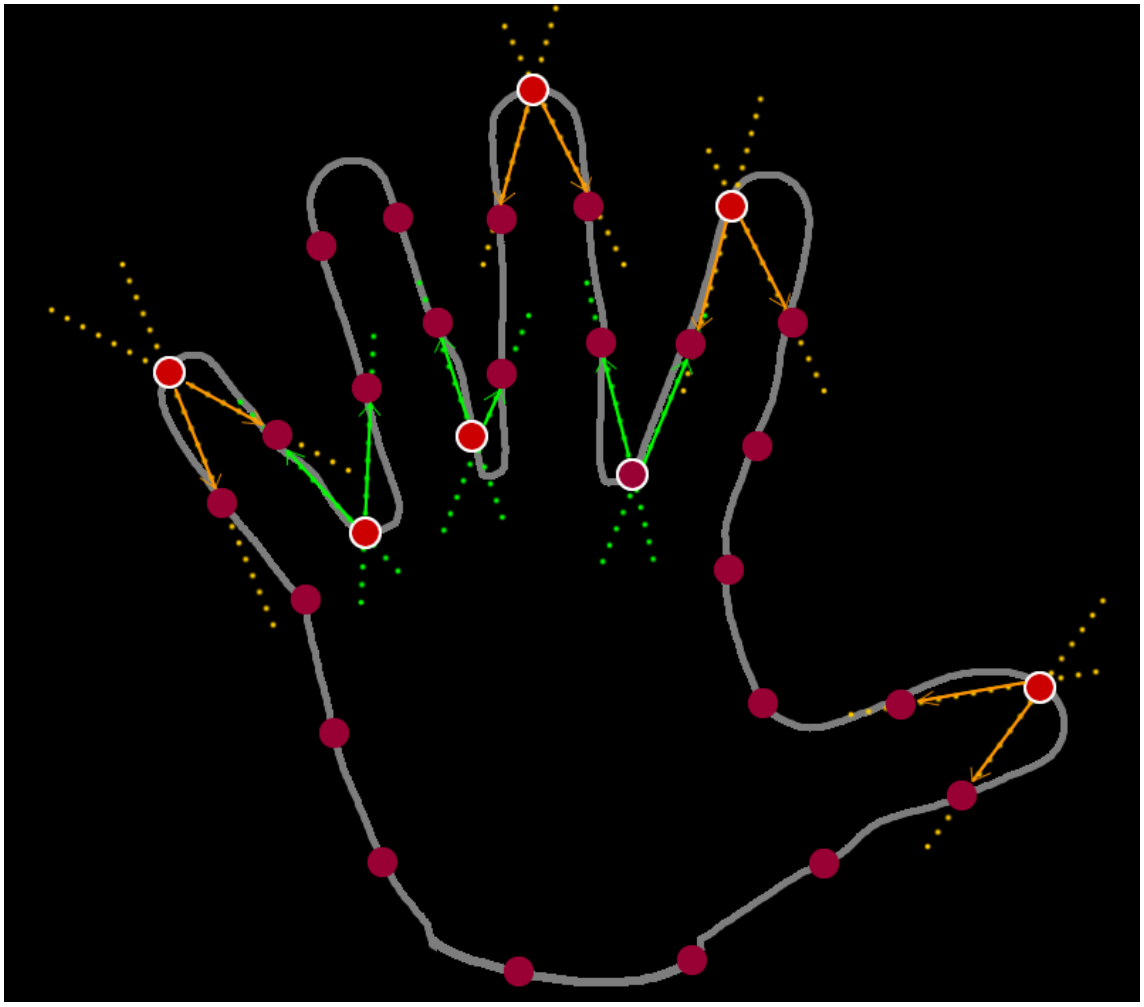

Figura 4.7: Processamento do algoritmo K-curvature (SHAKER; ABOU ZLIEKHA, 2007) sobre o contorno de uma mão, com os pontos de descontinuidade detectados marcados por círculos brancos. Os pontos que representam pontas de dedos (picos) foram detectados pelo ângulo formado entre os vetores em laranja, ao passo que aqueles detectados pelo ângulo entre os vetores verdes representa uma junção entre dedos (vales). Nota-se que alguns pontos de descontinuidade não foram detectados.

completamente independente da aplicação servidora, tendo apenas que se conectar ao *WebSocket* aberto por esta. Ambas aplicações devem rodar localmente no mesmo computador – muito embora essa não seja uma restrição necessária –, já que o processamento da linguagem JavaScript é feito de forma local pelo navegador. No entanto, a aplicação Web pode estar hospedada em qualquer servidor do mundo, podendo ser acessada de qualquer computador. Dessa forma, para que o sistema seja utilizado em um computador específico, basta instalar nele o SDK *Kinect For Windows*, conectar nele um Kinect, rodar a aplicação servidora e acessar a aplicação Web pelo site correspondente.

Para a aplicação servidora, foi utilizada a biblioteca Fleck[11], específica para troca de mensagens via *WebSocket* em C#. A conexão é aberta localmente na porta 8181, de modo a não interferir com o serviço de servidor Web, que utiliza a porta 80. Para a aplicação Web, é utilizada a classe nativa *WebSocket*, da linguagem JavaScript.

[11] https://github.com/statianzo/Fleck

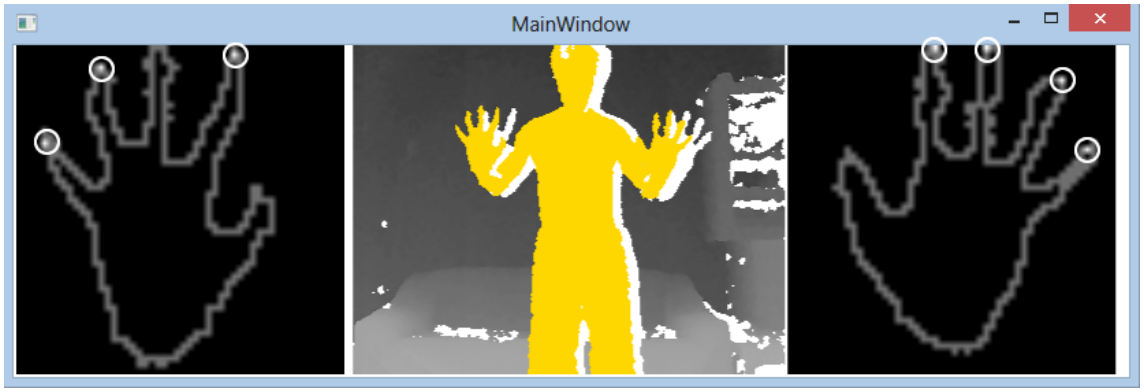

Figura 4.8: Aplicação desenvolvida para testar a leitura dos dados do Kinect mostrando a detecção das pontas dos dedos, de acordo com o algoritmo K-curvature (SHAKER; ABOU ZLIEKHA, 2007): ao centro a imagem de profundidade como obtida pelo Kinect e em cada lado uma das mãos isolada e com pontos de descontinuidade identificados.

4.4 Interpretando os dados na página Web

As aplicações Web utilizadas nesse estudo foram construídas utilizando as linguagens HTML5, JavaScript e PHP. Toda a parte visual foi implementada em HTML5, com uso de CSS para detalhes gráficos. A interatividade das páginas se dá por intermédio de linguagem JavaScript, com uso da biblioteca jQuery, que facilita a manipulação de elementos na tela. Linguagem PHP foi utilizada quando se fizeram necessárias interações com a base de dados ou com arquivos de texto.

Para interpretar os dados recebidos via mensagem pelo WebSocket, as aplicações possuem uma máquina de estados – vide figura 3.2 – que se atualiza de acordo com a informação de estado de cada uma das mãos. Os estados possíveis são: inativo, seleção com a mão direita, arraste com a mão direita, soltar com a mão direita, seleção com a mão esquerda, arraste com a mão esquerda, soltar com a mão esquerda e zoom.

A posição das mãos é atualizada independentemente dos estados, sendo indicada por ícones que mostram mãos abertas e fechadas. Adicionalmente, um esqueleto formado por segmentos de reta é construído com base nas informações de juntas recebidas via mensagem, conforme pode ser visto na figura 3.7, nas aplicações de mapa e de seleção e manipulação de quadrados descritas no capítulo 3. Essas atualizações são feitas a cada recebimento de mensagem, bem como a atualização da máquina de estados.

Seleções são determinadas de acordo com a posição da mão que a realizou, bem como os arrastes/translações. Caso exista um elemento na posição da mão, este será o elemento afetado, caso contrário será a tela de desenho, que é construída com o uso de um elemento <*canvas*> de HTML5. A ação de soltar é utilizada para liberar o elemento ou a tela da ação de posicionamento. O zoom é realizado com base na posição das duas mãos, focando-se no ponto-médio entre as duas: quando inicia, é armazenada a distância entre os pontos que representam as posições das mãos, d_0; a partir de então, nas atualizações posteriores, a distância entre esses pontos, d_i, é novamente calculada. Caso d_i seja maior que d_0, considera-se que foi realizada uma ampliação, ou seja, um aumento de zoom. Do contrário, ou seja, se d_i é menor que d_0, é considerado que foi feita uma redução, ou seja, uma diminuição do zoom.

Adicionalmente, para a aplicação que mostra a árvore genealógica foi utilizada a bi-

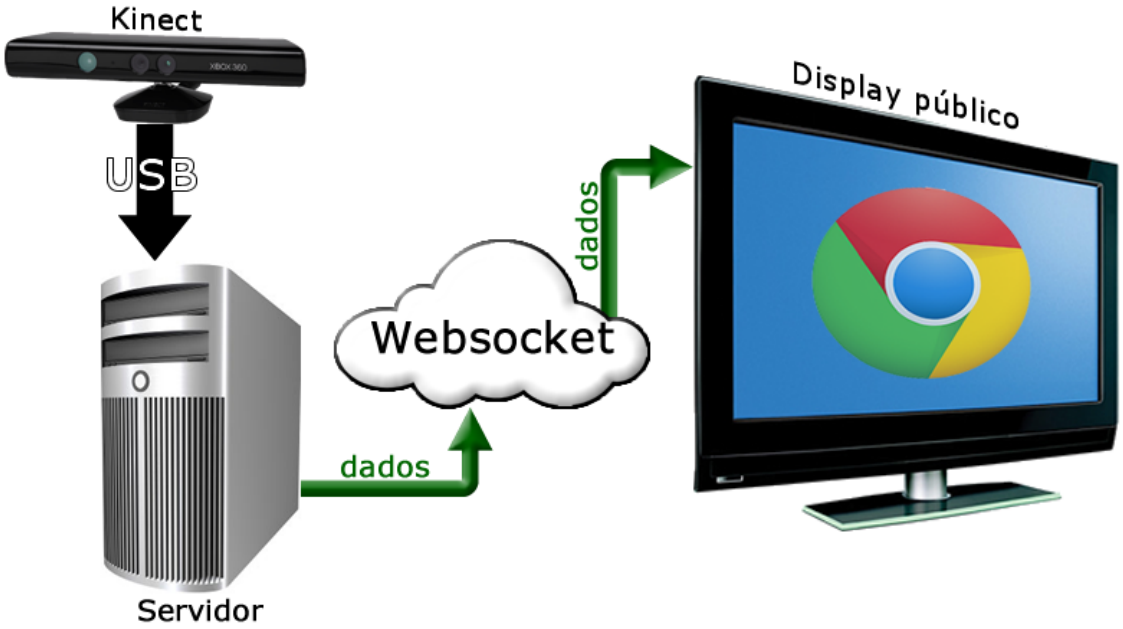

Figura 4.9: A arquitetura do sistema proposto, descrevendo as comunicações envolvidas no processo de transpor os dados interpretados do Kinect para a aplicação Web no display público.

blioteca de JavaScript *arbor.js*[12]. Essa biblioteca tem a função de posicionar os nodos do grafo de acordo com a interpretação de forças potenciais (*force-oriented*), mas deixa livre seu desenho de acordo com a escolha do desenvolvedor. Como afirmam os responsáveis pela biblioteca no website da mesma, "o que você desenvolve deve estar focado nas características que fazem o seu projeto único – os dados do grafo e o estilo visual – ao invés de gastar tempo nos cálculos matemáticos necessários à criação de seu layout".

[12]http://arborjs.org

5 AVALIAÇÃO DO SISTEMA

Após desenvolvido e testado, o sistema proposto foi instalado em um computador com processador Intel Core 2 Quad modelo q9550 de 2.83 GHz, com 4 GiB de memória RAM e placa de vídeo nVidia GeForce GTS 250 com 2287 MiB, conectado a um televisor de 55" por um cabo padrão VGA. O sistema operacional da máquina era um Windows 7 de 64 bits. As três aplicações desenvolvidas foram, então, avaliadas de acordo com os critérios relevantes para seu uso em um display público, sendo eles: condições de iluminação do ambiente, presença de outras pessoas, tipo do local, tipo da tarefa executada e apresentação da informação. Cada um deles será explicado com maiores detalhes mais adiante, juntamente com as hipóteses consideradas em cada caso. Ao final do capítulo, na seção 5.2, serão apresentados os testes de interação que foram realizados de forma a avaliar o sistema proposto.

5.1 Hipóteses avaliadas

Baseando-se nos critérios de avaliação considerados, elaborou-se uma hipótese para cada situação que poderia influenciar na interação em display público, de forma a buscar sua confirmação nos testes a serem conduzidos. Para fins de clareza, cada uma das hipóteses será descrita separadamente a seguir.

5.1.1 Condições de iluminação

Como o display de 55" é grande o suficiente para fornecer uma boa iluminação do usuário a sua frente e, de acordo com a descrição técnica do Kinect, este funciona até mesmo em um ambiente com ausência total de luz, a hipótese avaliada foi de que a iluminação não representa fator relevante na execução das tarefas.

5.1.2 Presença de outras pessoas no ambiente

Como a aplicação foi desenvolvida para sempre detectar a pessoa que está mais próxima do display, bem como notadamente as pessoas se sentem desconfortáveis quando observadas ao fazer algo não convencional, a hipótese levantada foi de que a presença de outras pessoas é fator relevante na execução das tarefas, especialmente quando transitam entre o usuário interagente e o display.

5.1.3 Tipo de local

O local de instalação do display público pode ser de três tipos: salas fechadas e controladas; locais fechados por onde passa um fluxo grande de pessoas, sem controle do ambiente, como shoppings, aeroportos e saguões de prédios; e espaços abertos, expostos

ao sol, como parques, praças e calçadas. O tipo de local pode ser relevante na interação se interferir no desempenho do sistema ou do usuário. Como dificilmente encontram-se displays públicos em espaços abertos, bem como a utilização do Kinect não ser recomendada nesse tipo de ambiente, foram avaliados apenas os dois primeiros tipos de locais. A hipótese levantada é de que o tipo de local não é fator relevante para a execução das tarefas interativas, desde que o usuário esteja sozinho no ambiente. Caso contrário, essa hipótese conflitaria com aquela levantada acerca da presença de outras pessoas no mesmo local.

5.1.4 Tipo de tarefa

Como notadamente a tarefa de manipulação de quadrados se mostra mais difícil e mais propensa a erros, já que o usuário precisa manter um objeto selecionado por mais tempo, a hipótese avaliada foi de que a tarefa de seleção é mais fácil que a de manipulação. Como as tarefas de *pan & zoom* servem apenas de suporte para a realização das tarefas, apenas foi avaliado se e o quanto os usuários fazem uso delas, bem como a forma como o realizam.

5.1.5 Apresentação da informação

Como a grande maioria das pessoas tem dificuldade em manter suas mãos perfeitamente paradas quando com os braços esticados e como as distâncias entre os quadrados a serem selecionados/manipulados aumentam conforme diminuem-se os tamanhos, a hipótese levantada é de que objetos maiores são mais fáceis de selecionar e manipular do que objetos menores.

5.2 Testes conduzidos

Existem diversas formas de se avaliar um projeto de interação humano-computador. Dentre elas, a mais aceita é a avaliação formal com usuários, uma avaliação quantitativa, na qual dados são obtidos na realização de testes com usuários utilizando o sistema proposto. Entretanto, uma avaliação cognitiva por especialistas pode ser feita para determinar se o sistema está apto para ser avaliado formalmente. Complementarmente, uma observação de campo pode resultar em indicadores qualitativos sobre o uso do sistema. Durante a construção desse trabalho, esses três tipos de avaliações foram conduzidas e cada uma delas foi determinante para a sua conclusão. Cada etapa de avaliação está descrita em maiores detalhes abaixo.

5.2.1 Avaliação informal por especialistas

Tendo sido a primeira aplicação desenvolvida, a aplicação de visualização da árvore genealógica foi exposta para três especialistas em interação, de modo que pudesse ser avaliado o método de interação proposto. Os avaliadores foram a Profª Luciana Nedel, a Profª Silvia Olabarriaga, da *University of Amsterdam*, e o Prof. Derek Reilly, da *Dalhousie University*.

O objetivo inicial desse trabalho era implantar o reconhecimento de gestos unicamente sobre a aplicação de visualização da árvore genealógica. Contudo, após a avaliação informal pelos especialistas supracitados, chegou-se à conclusão de que o sistema não era robusto o suficiente para isso e que avaliações mais elaboradas precisariam ser realizadas. Em especial, verificou-se que o sistema era muito instável quando na seleção de

elementos pequenos. Porém, ao aumentar o zoom na tela, a seleção ficava facilitada.

Em consequência das avaliações realizadas, decidiu-se por desenvolver as outras duas aplicações citadas no capítulo 3, que serviriam para avaliar o sistema formalmente e em um espaço público de fato. Dentre as modificações sugeridas pelos especialistas que foram implementadas nas duas aplicações seguintes foi a criação de um esqueleto semi-transparente que ajudasse o usuário a se localizar no sistema. Outra modificação importante foi a estabilização do reconhecimento do estado das mãos, com a introdução de um pequeno *buffer* para armazenar os estados, como explicado no capítulo 4.

5.2.2 Avaliação formal com usuários

A aplicação de seleção e manipulação de objetos simples foi utilizada para fins de avaliação formal com usuários (PREECE; ROGERS; SHARP, 2005). Para isso, 38 usuários foram convidados a realizar os testes propostos e foi medido o tempo com que eles executaram as tarefas e o número de erros ocorridos, bem como quanto utilizaram das funcionalidades de *pan & zoom*. Todos testadores executaram as tarefas no mínimo duas vezes, sendo a primeira vez para que eles se habituassem com o meio de interação. Os usuários também responderam a dois questionários: o primeiro antes de realizar os testes, de modo a caracterizá-los e o segundo sobre suas opiniões a respeito da execução das tarefas, aplicado após os testes para fins de avaliação subjetiva. Os questionários se encontram anexados a esse trabalho.

Figura 5.1: Local de aplicação dos testes com usuários. Destaque para a marcação do local onde o usuário deveria se posicionar.

O experimento foi conduzido em uma sala com iluminação artificial por lâmpadas fluorescentes com duas variações de luminosidade, bem como com e sem a presença de

outras pessoas. Três usuários tiveram de ser descartados devido a erros na execução da aplicação, totalizando 35 usuários com dados úteis. Havia uma marcação no chão indicando a posição ideal em que o usuário deveria se posicionar, que era ampla de modo que possibilitasse movimentação horizontal. Foi produzido um vídeo explicativo[1] sobre as tarefas e os modos de interação, que era exibido a todos os testadores após eles responderem o primeiro questionário. Com a explicação do teste feita em vídeo, todos usuários receberam exatamente as mesmas instruções, não sendo induzidos por algo que o condutor do teste tenha passado individualmente. A figura 5.1 mostra o ambiente de execução dos testes e em anexo a este trabalho está o Script de Interação, detalhando os passos de interação a ser cumpridos pelos usuários no teste.

Os testes tiveram como variáveis independentes, ou seja, que não dependem do usuário, o tipo da tarefa (selecionar/posicionar) e o tamanho dos quadrados. As variáveis dependentes, que alteram de acordo com o usuário foram o tempo de execução de cada tarefa e o número de erros ocorridos. Considerou-se como *erro* uma seleção de outro quadrado que não o em verde na tarefa de seleção e o largar do quadrado a ser posicionado em um local que não o destino correto na tarefa de posicionamento.

Afim de avaliar a diferença na quantidade de iluminação no ambiente, foram selecionados 8 usuários, sem nenhum critério específico, para realizar o teste na aplicação de seleção e manipulação de quadrados duas vezes (além da primeira, de treinamento) com duas variações de luminosidade medidas por um luxímetro: com iluminação normal, de $731lux$ e sem nenhuma iluminação que não a do display, de $221lux$. Para não ser fator determinante na avaliação, os usuários efetuaram os testes de forma alternada entre as duas intensidades de luminosidade.

Para avaliar se a presença de outras pessoas no ambiente é fator determinante para a interação em um display público, 10 pessoas que não as que avaliaram as diferenças de iluminação foram selecionadas de forma aleatória para executar as tarefas duas vezes: uma com a presença de outra pessoa transitando atrás e à frente delas, se prostrando ao seu lado e tentando interagir simultaneamente com o display, de forma a propositalmente atrapalhar o teste, e outra sem qualquer interferência. Novamente, para que a ordem de avaliação das modalidades não interferisse nos resultados da avaliação, a ordem delas foi alternada entre os usuários.

Na aplicação de seleção e manipulação de quadrados, estes têm seu tamanho diminuído no decorrer da execução das tarefas, bem como são apresentados em maior quantidade, tanto nas tarefas de seleção quanto nas de manipulação. Utilizando-se disso, foi avaliado se o tamanho e quantidade de objetos interfere no tempo de realização das tarefas, comparando-se o resultado de todos os 35 usuários em um ambiente com iluminação por lâmpadas e sem a presença de outras pessoas, bem como para avaliar o tipo de tarefa executada.

5.2.3 Estudos de observação

A aplicação que mostra o mapa do prédio foi executada primeiramente em uma sala fechada e, após, no saguão de entrada do prédio ao qual o mapa pertence, junto com um recipiente com balas e pirulitos para atrair usuários. Foi feita uma observação do ambiente durante 12 horas, dispostas em dois dias em duas semanas distintas, buscando avaliar a forma com que as pessoas se portam na presença de um display público interativo. A observação foi feita em um local distante do display de modo a não interferir no ambiente

[1] http://www.youtube.com/watch?v=tXKwPY9-X5c

observado. O prédio em que o sistema foi implantado é transitado por diversos professores e bolsistas, pois há, além de gabinetes de professores, laboratórios de ensino no prédio. Estimadamente, cerca de 100 pessoas passaram pelo display em cada um dos dois dias de observação.

Foram tomadas notas de como as pessoas alteram seu comportamento quando há algo diferente em um local em que elas estão habituadas a transitar, bem como o modo com que essas pessoas interagiram com o display. Diversas pessoas passaram pelo local, sozinhas ou em grupo e nenhuma foi indiferente ao aplicativo colocado na entrada do prédio, especialmente porque o esqueleto de uma pessoa no ambiente era exibido no display no instante em que ela passava pela frente do mesmo, não importando se estivesse a uma grande distância.

6 RESULTADOS

Em uma análise inicial, utilizando o sistema para interagir com a visualização de árvore genealógica (a primeira aplicação desenvolvida), ficou evidente que o sistema não era suficientemente robusto. Erros de interpretação ocorriam quando eram necessários movimentos finos para interação, como selecionar um nodo específico do grafo e posicioná-lo em algum lugar. Apesar disso, o sistema se mostrou eficiente em tarefas mais simples, que exigissem gestos mais amplos, como uma translação na tela ou a realização de zoom.

O sistema desenvolvido foi apresentado a três especialistas em interação, que o utilizaram e fizeram sugestões para sua melhoria. Ainda que a maioria de tais aprimoramentos tenha sido implementado, percebeu-se que há uma barreira tecnológica que impede a construção de um sistema totalmente robusto, representada pelo Microsoft Kinect. O dispositivo é suficientemente bom para reconhecer o esqueleto do usuário, mas a baixa resolução de suas câmeras – especialmente daquela sensitiva de infravermelho – dificulta a interpretação correta da abertura e fechamento das mãos em alguns casos. O Kinect também apresenta certa instabilidade na montagem do esqueleto, que se deve especialmente à predição incorreta da posição de juntas que ele não é capaz de localizar.

Esses problemas inerentes ao dispositivo são bastante conhecidos pela Microsoft e demais desenvolvedoras de jogos para X-Box que utilizam-no. Isso fica visível ao analisar os subterfúgios que são utilizados para contornar os problemas nos jogos. Em quase todos os jogos existem botões grandes, sobre os quais a mão deve ficar parada durante alguns instantes para fazer uma seleção (CHOUMANE; CASIEZ; GRISONI, 2010), de modo que o jogador consiga deixar seu braço e mão parados sobre a figura durante alguns segundos, como mostra a figura 6.1-A. Outra técnica bastante utilizada são botões nos cantos da tela, uma técnica utilizada por interfaces convencionais que utilizam o mouse, com um aditivo que essas posições funcionam como imãs para as mãos, atraindo-as quando estão perto delas, como exemplificado na figura 6.1-B. Como há uma região ideal para o reconhecimento correto do usuário, alguns jogos indicam a posição em que o jogador deve ficar, como pode ser visto na figura 6.1-C. Por fim, jogos para Kinect costumam utilizar gestos amplos para interação, de modo que seja mais difícil não reconhecê-los. Por exemplo, em jogos em que é preciso pular ou correr, o jogo pede que o usuário erga bem os joelhos para um reconhecimento correto, uma interação não muito natural quando se corre ou pula normalmente. Gestos amplos podem ser reconhecidos via interpretação de padrões, uma técnica amplamente utilizada por jogos de dança, como mostra a figura 6.1-D.

No entanto, mesmo procurando contornar as deficiências do Kinect, os próprios jogos para o dispositivo não encontram um sucesso completo. Se há outro usuário transitando no ambiente é bastante comum que o jogo perca o jogador, demorando para reconhecê-lo novamente. Jogos que tentam reconhecer movimentos finos, como o jogo de dardos do *Kinect Sports* ou o *Kinect Star Wars* em suas lutas de sabre, falham miseravelmente na

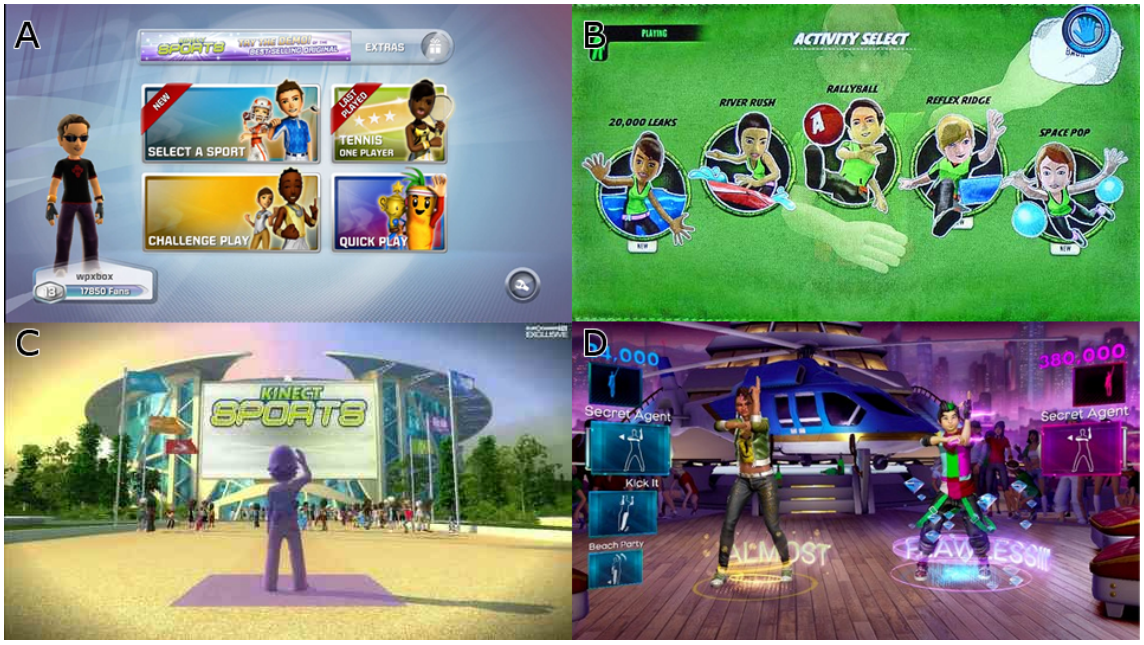

Figura 6.1: Subterfúgios utilizados pelos desenvolvedores de jogos para Kinect para contornar os problemas inerentes ao dispositivo: A) tela do *Kinect Sports* mostra ícones grandes, sobre os quais o usuário consiga manter sua mão parada por alguns segundos; B) imagem do *Kinect Adventures* mostrando ícones nos cantos da tela, tipicamente locais de fácil acesso, e que "atraem" as mãos do usuário; C) mais uma do *Kinect Sports*, que mostra a técnica de indicar a posição em que o usuário deve ficar para que seus movimentos sejam reconhecidos; D) *Dance Central* e demais jogos de dança fazem o reconhecimento de padrões brutos, sem interpretação detalhada dos gestos do usuário.

grande maioria das vezes, algo totalmente inaceitável para um produto comercial.

Muito embora tenha seus problemas, o modelo proposto nesse trabalho possui grande potencial de aplicação quando utilizado para tarefas interativas mais simples. Na avaliação formal com usuários, utilizando a aplicação de seleção e manipulação de quadrados, os testadores responderam a diversas perguntas sobre a execução das tarefas e suas respostas indicam que o sistema proposto teve bom desempenho em determinadas situações. A figura 6.2 apresenta um gráfico que resume algumas dessas respostas, onde é possível perceber, em especial, que a tarefa de seleção foi avaliada como "muito pouco" ou "pouco" difícil por 20 usuários. Em contrapartida, a tarefa de posicionamento foi avaliada como "muito difícil" ou "difícil" por 16 usuários, um número bastante elevado. Finalmente, outro ponto interessante a ser reparado é que 23 usuários consideram que a execução do teste foi "divertida" ou "muito divertida".

Em um campo para comentários gerais no segundo questionário, muitos usuários destacaram que o sistema se mostra cansativo em determinadas situações, o que pode ter sido gerado pela dificuldade em posicionar os quadrados menores quando a posição de destino estava muito distante. Um dos usuários destaca que o sistema é mais responsivo quando os objetos a serem selecionados ou manipulados estão na altura de seu ombro ou acima. Tal fato, verificado ao observar a execução dos testes, pode indicar que um deslocamento vertical da posição virtual das mãos do usuário poderia aumentar a eficiência do sistema, mas também acarretaria em um aumento da exaustão, já que o usuário teria que manter sua mão erguida durante toda a tarefa interativa.

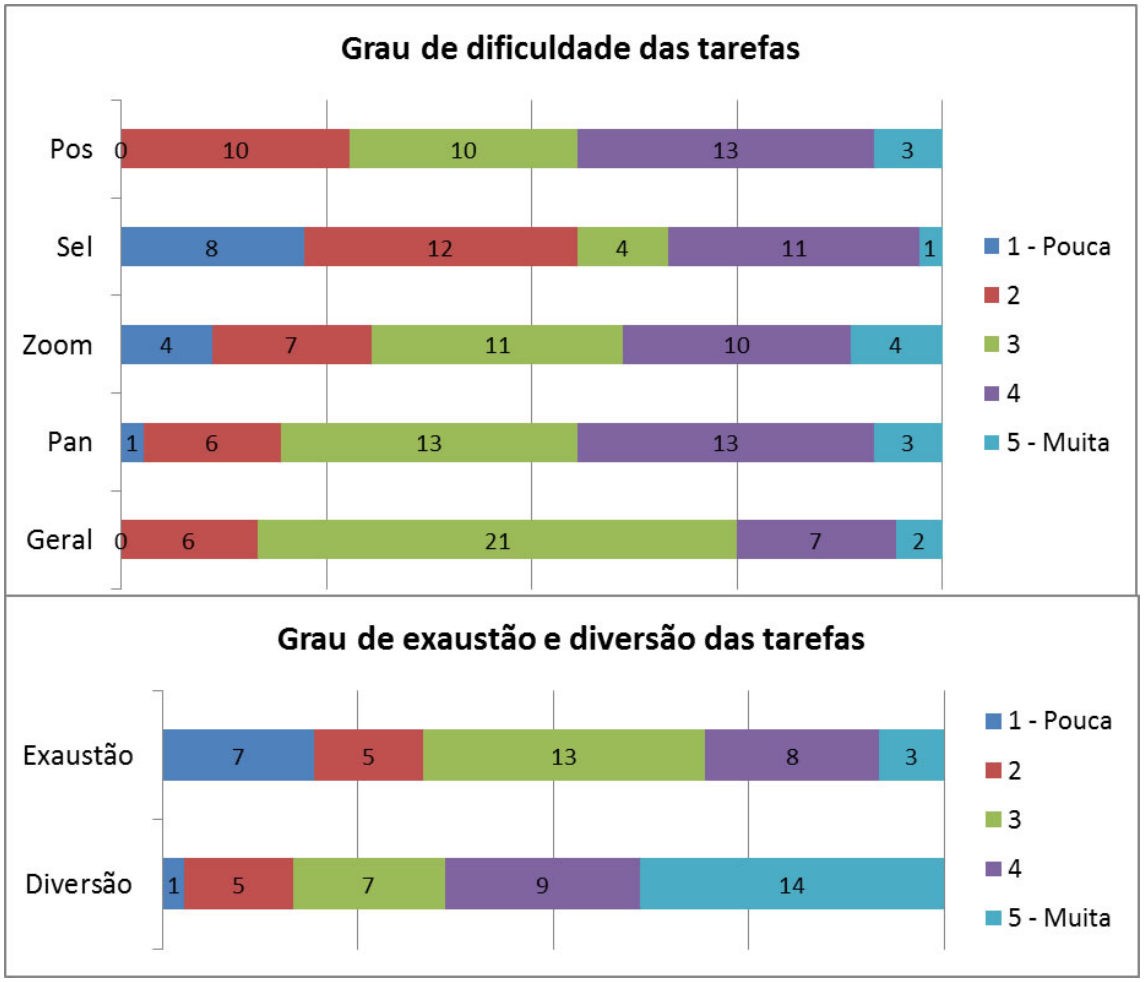

Figura 6.2: Gráfico demostrando o resumo das respostas dos usuários acerca da execução das tarefas de acordo com uma escala *likert* de 1 a 5, na qual 1 é um valor baixo (e.g. baixa dificuldade). Respostas foram para responsividade do sistema em geral e das tarefas de *pan & zoom*; dificuldade em executar as tarefas de seleção e posicionamento; e grau de divertimento e exaustão ao executar o teste.

Outra característica interessante que foi observada pelos usuários é que a prática dos gestos interativos melhora muito o desempenho do usuário. Citando um comentário: "com um curto tempo de uso/interação da ferramenta, o domínio de suas funcionalidades já é razoável". Graças à execução da tarefa inteira uma primeira vez apenas como treinamento, o usuário já realiza os testes com velocidade e precisão maiores do que o fez no treino.

Os usuários que realizaram os testes são divididos entre 30 homens e 8 mulheres, possuem idade média de 23,82 anos e mediana de 24, todos habituados ao uso diário de computador. Dezoito deles já haviam interagido com o Kinect e quatorze já interagiram frente a uma TV interativa, mas essas características não influenciaram nos testes de acordo com uma análise de variância (*p-values* 0.4534 e 0.2297 respectivamente). Além disso, 25 dos usuários interagem diariamente ou quase diariamente com dispositivos com telas sensíveis ao toque, dessa forma tendo conhecimento dos gestos utilizados para alterar a escala da tela.

Abaixo, em seções específicas, são apresentados os resultados de todos os critérios avaliados nesse trabalho para uma interação gestual sem dispositivos em um display pú-

blico.

6.1 Condições de iluminação

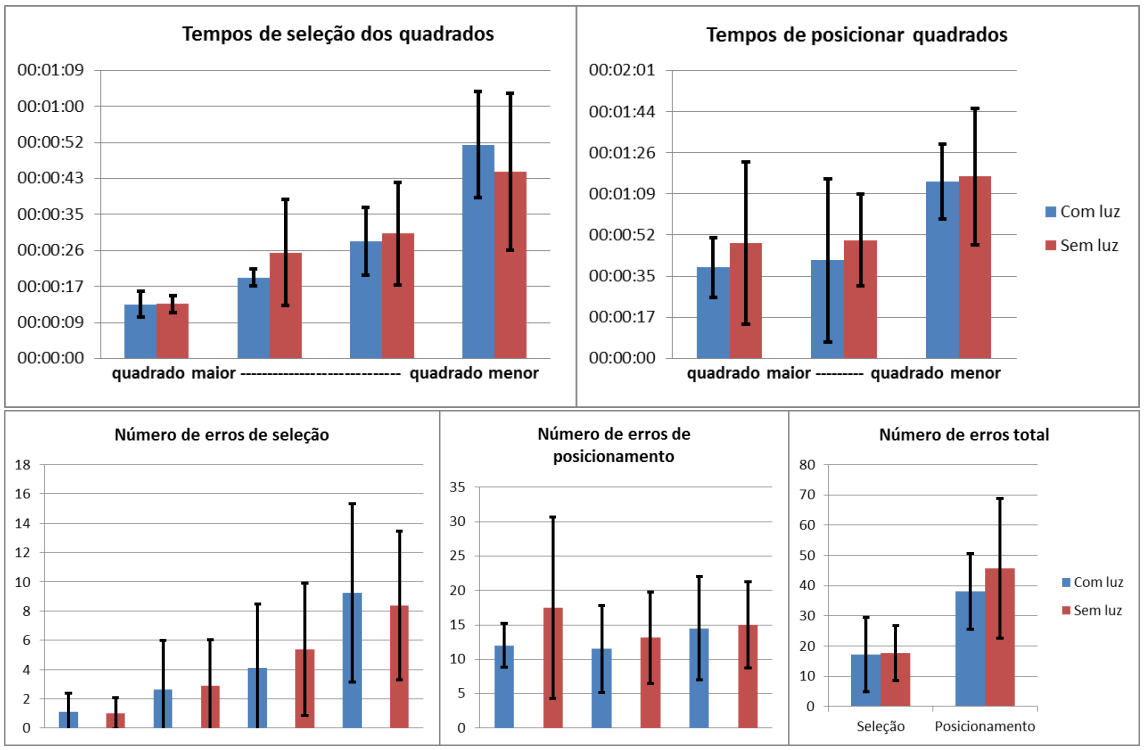

Figura 6.3: Gráficos mostrando as diferenças de dados obtidos nos testes com o ambiente iluminado por lâmpadas ou não: acima, em relação ao tempo médio de execução de cada tarefa pelos usuários e abaixo em relação ao número de erros médio ocorridos em cada tarefa. Em relação ao tamanho de quadrados, a disposição é sempre dos maiores para os menores. O ambiente iluminado é representado pelas barras em azul e sem iluminação pelas vermelhas. A barra em preto entrecortando cada barra marca o desvio padrão.

Para provar a hipótese proposta, a hipótese nula, ou seja, que a iluminação **é** um fator relevante no uso do sistema, foi avaliada. Conforme esperado, a condição de iluminação do ambiente não se mostrou fator relevante na execução das tarefas interativas. Segundo a análise de variância (ANOVA), a seleção de quadrados não teve seu tempo determinado pela quantidade de iluminação. Em ordem decrescente de tamanho de quadrados obteve-se: tempo médio de 0'13" com e sem iluminação, com *p-value* 0.9256; tempo médio de 0'19" com iluminação e 0'25" sem, com *p-value* 0.2822; tempo médio de 0'28" com iluminação e 0'30" sem, com *p-value* 0.7816; e tempo médio de 0'51" com iluminação e 0'45" sem, com *p-value* 0.6303.

O posicionamento também não mostrou correlação, obtendo-se médias de tempo com e sem iluminação e *p-values* de: 0'38", 0'49" e 0.4394 para o maior quadrado; 0'41", 0'50" e 0.3514 para o quadrado de tamanho médio; e 1'14", 1'16" e 0.8991 para o menor quadrado. Mesmo que o tempo de posicionamento seja levemente menor nos casos com iluminação, a análise de variância mostra que tais resultados não são conclusivos.

O número de erros em cada tarefa também não se mostrou estatisticamente relevante, apresentando médias de 17.12 e 17.62 para seleção e 38.00 e 45.62 para posicionamento,

com e sem iluminação respectivamente, e *p-values* 0.9274 e 0.4267 para erros de seleção e posicionamento respectivamente. O menor número de erros nas tarefas foi 4 e 22 com iluminação para seleção e posicionamento, respectivamente, enquanto que sem ilumina-ção foram 9 e 21. O maior número de erros foi 38 e 57 para seleção e posicionamento, respectivamente, com iluminação, e 32 e 82 sem iluminação.

Assim, de acordo com análises de variância, não é possível afirmar que a iluminação é um fator relevante ao aplicar-se o sistema proposto em um display público. Logo, a hipótese nula é provavelmente falsa e a hipótese provavelmente é verdadeira, ou seja, a iluminação não deve ser fator relevante para a execução das tarefas. A figura 6.3 apresenta os gráficos de tempos e número de erros das tarefas de seleção e posicionamento com e sem a presença de iluminação no ambiente por lâmpadas. Ao analisar os gráficos levando em conta o desvio padrão indicado há um forte indicativo de que não existe relação entre a variação de iluminação com o desempenho dos usuários nos testes.

6.2 Presença de outras pessoas no ambiente

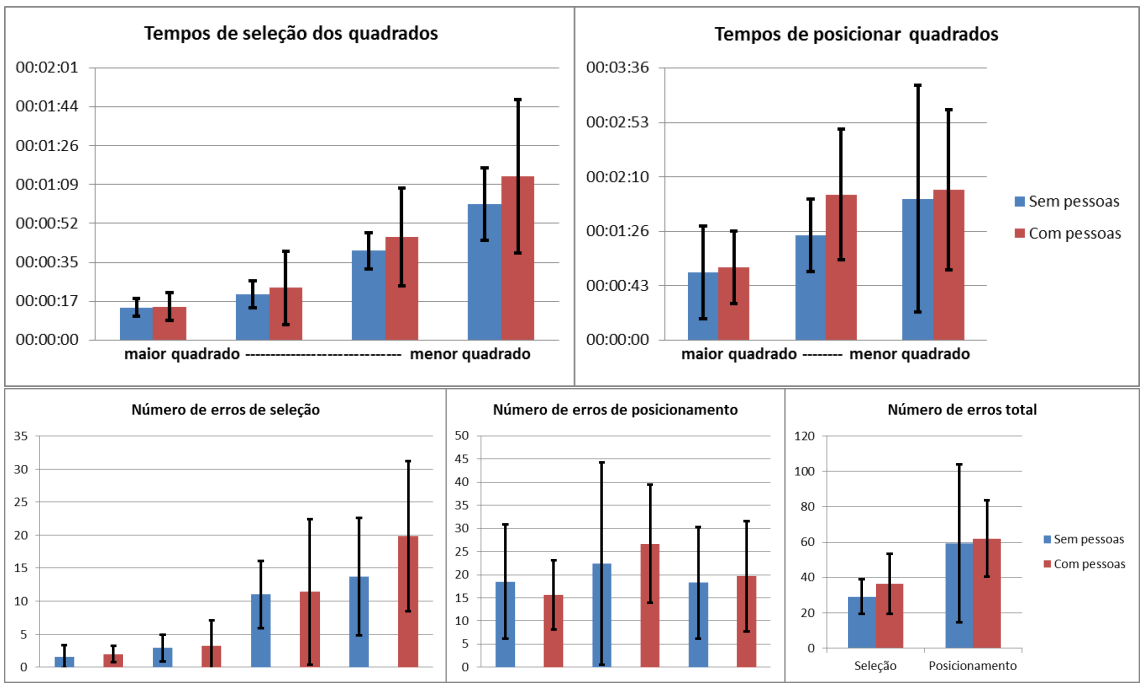

Figura 6.4: Gráficos mostrando as diferenças de dados obtidos nos testes com e sem a presença de outras pessoas interferindo no teste: acima, em relação ao tempo médio de execução de cada tarefa pelos usuários e abaixo em relação ao número de erros médio ocorridos em cada tarefa. Em relação ao tamanho de quadrados, a disposição é sempre dos maiores para os menores. A interação sem interferência é representada pelas barras em azul e com a presença de pessoas pelas vermelhas. A barra em preto entrecortando cada barra marca o desvio padrão.

Ao contrário do que havia sido previsto, de acordo com uma análise de variância dos tempos de execução das tarefas e do número de erros cometidos, não foi possível provar que a presença de outras pessoas transitando no mesmo ambiente que o interagente representa fator relevante para o cumprimento das tarefas.

A seleção de quadrados teve médias de tempo sem e com a presença de outras pessoas

de: 0’15” nos dois casos para o maior deles; 0’20” e 0’23” para o segundo maior; 0’40” e 0’46” para o terceiro; e 1’01” e 1’40” para o menor quadrado. Os *p-values* obtidos na ANOVA foram 0.8566, 0.6293, 0.5106 e 0.4404 respectivamente. Por sua vez, a análise do posicionamento dos quadrados teve médias de tempo sem e com pessoas e *p-values* nos três tamanhos de quadrado também em ordem decrescente de: 0’54”, 0’58” e 0.7939; 1’23”, 1’56” e 0.3363; e 1’52”, 1’59” e 0.8034. A análise de variância do número de erros apresentou *p-values* 0.2637 e 0.8600 em relação à seleção e ao posicionamento respectivamente.

As médias do número de erros foram de 29.20 e 36.40 para seleção e 59.20 e 62.00 para posicionamento, sem e com a presença de outras pessoas respectivamente. O menor número de erros nas tarefas foi 19 e 29 sem outras pessoas para seleção e posicionamento, respectivamente, enquanto que com outras pessoas foram 17 e 31. O maior número de erros foi 50 e 69 para seleção e posicionamento, respectivamente, sem outras pessoas, e 63 e 98 com outras pessoas.

Muito embora tanto os tempos de execução como o número de erros ocorridos indiquem que a presença de outras pessoas é prejudicial ao desempenho do usuário, não é possível afirmar isso estatisticamente. A não confirmação da hipótese é um ponto muito positivo para o sistema proposto, indicando que o mesmo é capaz de identificar a pessoa que está interagindo e manter-se coerente durante toda a tarefa interativa, recuperando-se rapidamente de eventuais erros. A figura 6.4 apresenta gráficos que mostram o quão próximos ficam os tempos e o número de erros nas diversas tarefas com e sem a presença de outras pessoas no ambiente. Novamente, a análise dos limites impostos pelas barras de desvio padrão indica que ambas configurações do ambiente são semelhantes.

6.3 Tipo de local

O local onde se dará a interação em um display público pode influenciar a execução das tarefas quando há mudança de iluminação ou presença de mais pessoas no ambiente. Esses critérios específicos foram avaliados e os resultados apresentados acima, indicando que a hipótese levantada poderia ser confirmada. No entanto, alguns fatores psicológicos também podem influenciar o resultado e tais fatores não podem ser medidos quantitativamente. Procurando avaliar o sistema em um ambiente de uso real, observações foram feitas quando o sistema foi implantado na entrada de um prédio com grande circulação de pessoas e foi possível perceber a existência de certos padrões de comportamento.

No primeiro dia em especial foi possível perceber que as pessoas se mostram interessadas ao passar por um display público interativo, olhando interrogativamente para a tela quando seu esqueleto aparece desenhado nela. Porém, apesar de parecerem interessadas, as pessoas parecem ter receio de experimentar o sistema, decidindo seguir seu caminho após alguns instantes de hesitação.

Esse padrão ocorre com menos frequência quando as pessoas estão em grupo, caso em que parece se sentirem encorajadas a interagir com o sistema como se para mostrarem-se corajosas. Um dos elementos de um grupo sempre toma a frente e começa a interagir com a tela e os demais membros se aproximam e tentam fazer o mesmo, como que procurando atrapalhar seu companheiro, mas sem obter sucesso já que o sistema é focado em apenas um usuário. Após alguns momentos de exploração, o grupo segue seu caminho, comentando sobre a experiência que tiveram. A figura 6.5 mostra o display em seu estado inicial no local onde foi colocado e duas situações em que grupos interagiram com o sistema.

O receio em interagir com o sistema também desaparece quando o usuário acha que

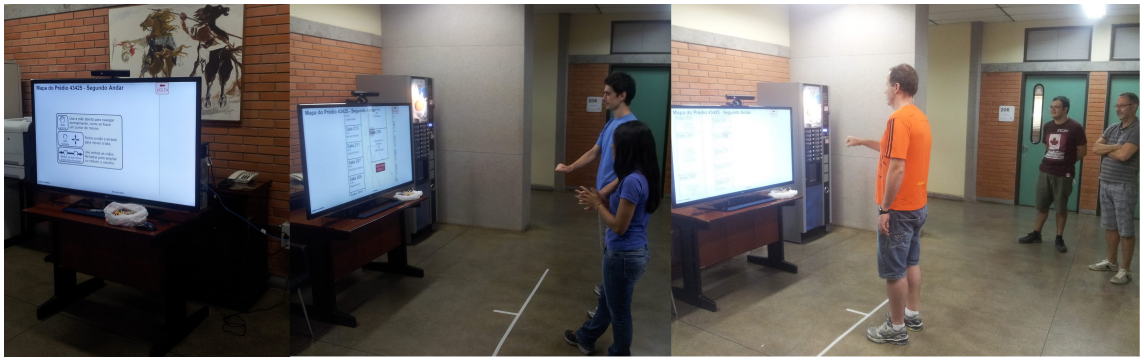

Figura 6.5: À esquerda o sistema conforme instalado no saguão de entrada do prédio cujo mapa é visualizado na aplicação e à direita dois momentos com grupos interagindo com o display.

está sozinho no ambiente. Isso ficou bastante claro ao observar um usuário que ia frequentemente buscar folhas na impressora que ficava ao lado da tela e sempre observava o display, mas, como havia outras pessoas passando na volta, não se atrevia a interagir. Porém, em uma de suas visitas à impressora, olhou para os lados e não viu ninguém e então tentou utilizar o sistema por alguns minutos, mas logo voltou a seguir seu caminho (especialmente porque como ele carregava papéis em uma mão, o sistema não reconhecia se ela estava aberta ou fechada e não se comportava corretamente). Importante ressaltar que o observador estava durante todo o tempo em um local mais à frente do display, a cerca de 6 metros, mas o sistema instalado prende tanto a atenção dos usuários que o observador nem foi notado.

A presença de balas sobre a mesa onde se encontrava o display serviu para atrair alguns usuários, mas o próprio sistema parecia ser suficientemente atrativo. No segundo dia de observação, mais pessoas pararam para interagir com o sistema. Um usuário reparou no observador e pediu uma "demonstração" para ver como o sistema funcionava e depois tentou interagir um pouco. Muitas pessoas utilizaram o sistema apenas para ver como funcionava, sem dar muita atenção para a informação, talvez por já saberem a localização e ocupação de todas as salas do prédio.

A aplicação de visualização do mapa fica exibindo instruções de como interagir até que surja algum usuário em frente à tela, ainda mantendo-as na tela por alguns segundos após isso. Alguns usuários que leram as instruções ao lado e já se puseram em frente ao display para interagir ficaram com a impressão de que o sistema não funcionava corretamente, pois seus movimentos só eram interpretados após as instruções terem desaparecido. Isso por vezes fazia com que o usuário perdesse o interesse e fosse embora.

O sistema se portou de maneira idêntica tanto na sala fechada quanto no saguão do prédio, muito embora não se possa dizer o mesmo das pessoas que interagiram com ele, especialmente as que o fizeram sozinhas. Parece haver algo que restringe as pessoas quando devem executar gestos não usuais em ambientes públicos, como já foi observado no trabalho de (RICO; BREWSTER, 2010). Em geral, o sistema pareceu ter sido bem aceito pelos usuários que se atreveram a utilizá-lo e obtido sucesso. Um dos usuários comentou que para interagir corretamente "é só questão de se acostumar com os gestos utilizados". Entretanto, não foi possível confirmar a hipótese de que o tipo de local não influencia o desempenho dos usuários, pois entende-se que há fatores que necessitariam de um estudo mais detalhado para que o cenário pudesse ser melhor avaliado.

6.4 Tipo de tarefa

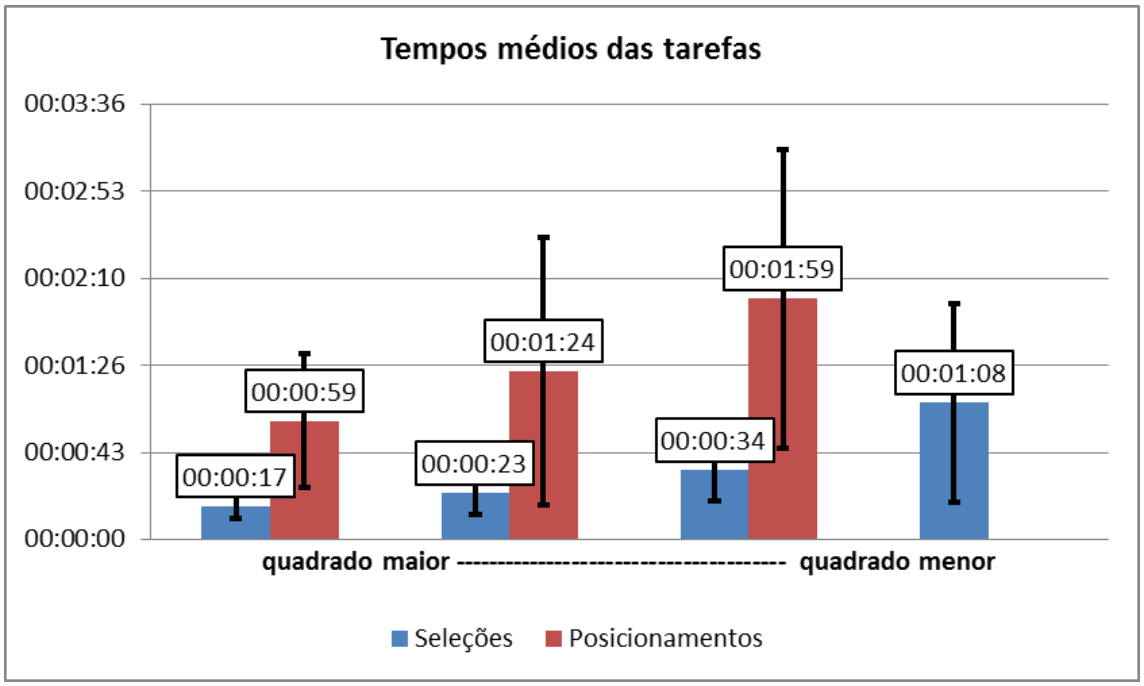

Figura 6.6: Gráfico comparativo dos tempos médios de finalização de cada uma das tarefas de seleção (azul) e posicionamento (vermelho). As barras pretas indicam o intervalo de desvio padrão.

Conforme supunha-se, após a análise de variância, ficou bastante claro que a tarefa de seleção é mais fácil de ser executada que a de posicionamento. Os três tamanhos de quadrados que são comuns às duas tarefas, em ordem de tamanho decrescente, tiveram médias de tempo: 0'17" para seleção e 0'59" para posicionamento, com *p-value* $2.4523*10^{-10}$; 0'23" para seleção e 1'24" para posicionamento, com *p-value* $1.0989*10^{-6}$; e 0'34" para seleção e 1'59" para posicionamento, com *p-value* $6.4350*10^{-9}$. Assim, com uma probabilidade muito elevada, a tarefa de seleção deve ser mais fácil que a de posicionamento, portanto a hipótese está provavelmente correta. Ao analisar o gráfico da figura 6.6 essa afirmativa fica bastante clara também.

Em uma análise mais aprofundada, conclui-se que mesmo a seleção do segundo maior quadrado deve ser mais simples que o posicionamento do maior quadrado, que é o mais fácil de de posicionar, obtendo-se um *p-value* $4.9689*10^{-8}$ em uma análise de variância, com médias de tempo 0'23" e 0'59" respectivamente. O mesmo ocorre ao se comparar a seleção do segundo *menor* quadrado com o posicionamento do maior quadrado, caso em que se obtém um *p-value* 0.00019 (tempos médios 0'34" e 0'59"), número bastante mais alto que os demais, porém ainda com significância estatística. A tabela 6.1 apresenta os tempos médios dos usuários nas tarefas de seleção e posicionamento dos quadrados, com seus respectivos desvios padrão.

Entretanto, a consideração mais importante é não há uma correlação estatisticamente relevante entre a tarefa de selecionar o menor quadrado (tempo médio de 1'08") e posicionar o maior quadrado (0'59"), caso em que obtém-se um *p-value* 0.3794. Isso pode ser verificado graficamente na figura 6.7-topo, que mostra a comparação dos tempos médios de cada usuário para a seleção e o posicionamento do maior quadrado e a seleção do menor quadrado. Esse panorama não ocorre ao analisar o número de erros, quando o

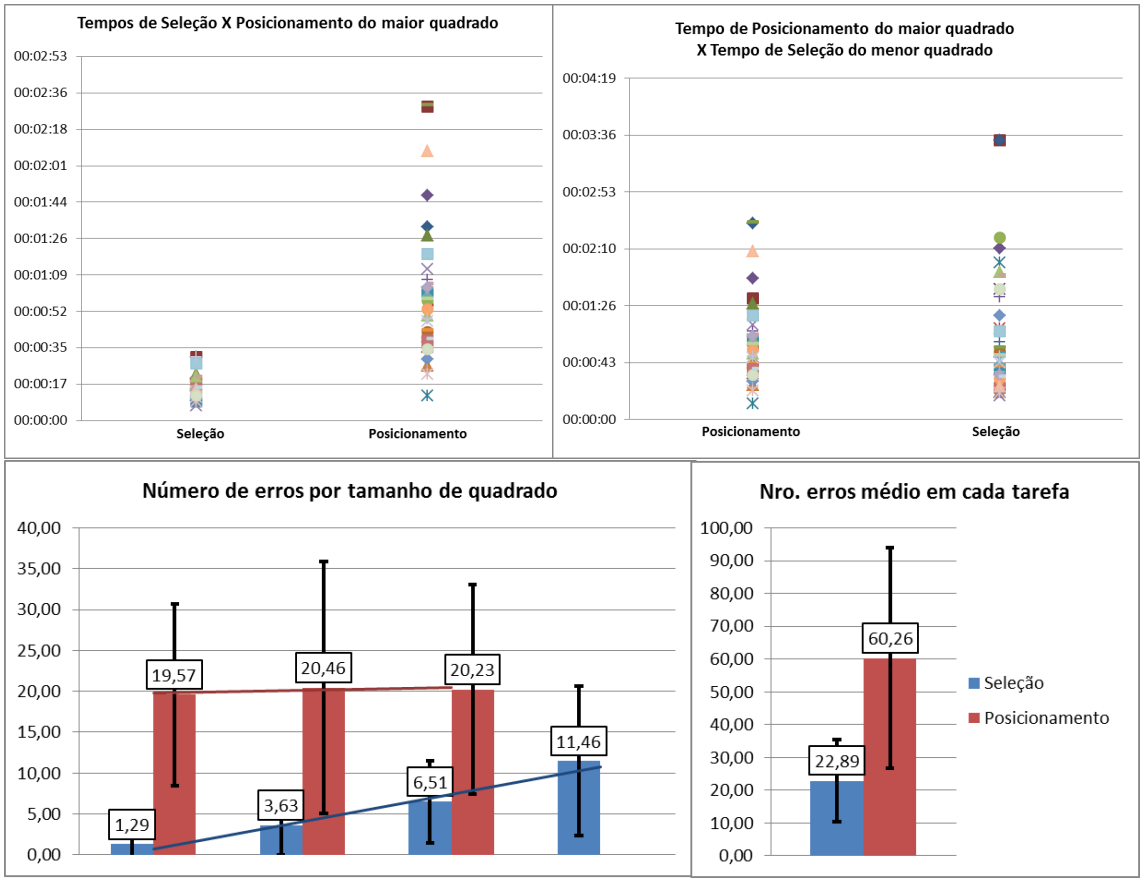

Figura 6.7: Em cima, gráficos comparativo dos tempos de cada usuário para finalização das tarefas de seleção e posicionamento do maior quadrado (à esquerda) e do posicionamento do maior quadrado com a seleção do menor quadrado (à direita). Abaixo, gráfico comparativo do número de erros das tarefas de seleção e posicionamento.

p-value obtido é 0.0013, indicando que na tarefa de selecionar o menor quadrado (com número médio de 11.45 erros) devem ocorrer menos erros que na de posicionar o maior quadrado (que possui número médio de 19.57 erros). Ao analisar o gráfico da parte de baixo da figura 6.7 é possível ver que o número médio de erros é muito maior nas tarefas de posicionamento do que nas de seleção.

Em relação aos gestos para translação da tela e alteração do nível de zoom, todos os usuários os utilizaram ao menos uma vez. A translação foi efetuada com frequência, em algumas vezes por engano, o que atrapalhou um pouco a tarefa interativa especialmente na de posicionamento. Tanto a translação quanto o zoom foram efetuados nos dois tipos de tarefas. Muitos usuários utilizaram a translação para levar o quadrado a ser selecionado ou posicionado para uma posição acima dos ombros, área em que a interpretação dos gestos era mais precisa. Além disso, muitos testadores utilizaram o zoom quando os quadrados se apresentavam em seu menor tamanho, seja na tarefa de seleção ou na de posicionamento dos quadrados.

6.5 Apresentação da informação

No que diz respeito ao tamanho e quantidade dos quadrados, as hipóteses se confirmaram, mostrando que quadrados maiores são mais fáceis de selecionar (a exemplo dos

	Quadrado maior			Quadrado menor
Seleção				
Média	00:17	00:23	00:34	01:08
Desvio Padrão	00:06	00:10	00:16	00:49
Posicionamento				
Média	00:59	01:24	01:59	(inexistente)
Desvio Padrão	00:33	01:06	01:14	(inexistente)

Tabela 6.1: Tabela com os tempos médios de seleção e posicionamento dos quadrados pelos usuários, em ordem de maior tamanho para menor.

grandes botões utilizados no jogos para X-Box) e posicionar. Considerando-se os tamanhos hipotéticos de quadrados 8, 4, 2 e 1 para a seleção, o quadrado de tamanho 8 deve ser mais fácil de selecionar do que um de tamanho 4 de acordo com uma análise de variância com *p-value* 0.0025 e tempos médios de 0'17" e 0'23" respectivamente. Seguindo-se a regra, um quadrado de tamanho 4 deve ser mais fácil de selecionar que um de tamanho 2 com um *p-value* 0.0004 (médias de tempo 0'23" e 0'34") e de tamanho 2 em relação ao de tamanho 1 com um *p-value* 0.0002 (médias de tempo 0'34" e 1'08").

Os resultados se repetem em relação à tarefa de posicionamento. Considerando-se os tamanhos hipotéticos de quadrados 8, 4 e 2, os quadrados de tamanho 8, com tempo médio de posicionamento de 0'59", devem ser mais fáceis de posicionar que os de tamanho 4, que levaram em média 1'24" para serem posicionados, segundo uma análise de variância com *p-value* 0.0529. Em relação aos de tamanho 2, com média de tempo 1'59", o *p-value* passa a ser $3.8358*10^{-5}$. Por sua vez, os quadrados de tamanho 4 devem ser mais fáceis de posicionar que os de tamanho 2 de acordo com uma análise de variância com *p-value* 0.0369. Os gráficos da figura 6.8 mostram as diferenças de tempos em que os usuários cumpriram cada um dos 7 conjuntos de tarefas (4 de seleção e 3 de posicionamento), com os tamanhos de quadrado variando de maior a menor.

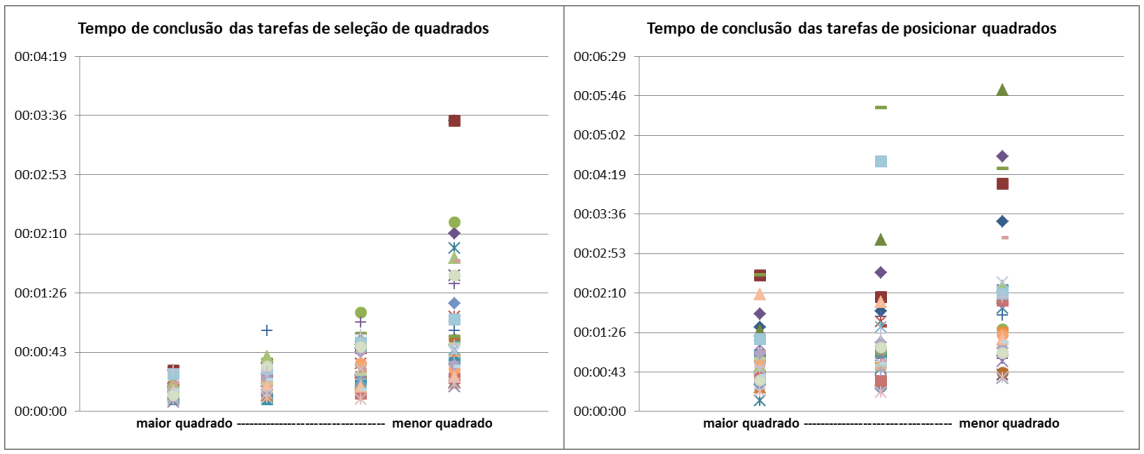

Figura 6.8: Gráfico comparativo dos tempos médios de cada usuário nas tarefas de seleção (à esquerda) e posicionamento (à direita) dos quadrados em seus tamanhos diversos.

É interessante reparar na figura 6.7-baixo que o número médio de erros na tarefa de seleção cresce quase linearmente conforme diminui o tamanho do quadrado, como indica

a linha azul escura ao passo que o número médio de erros de posicionamento mantêm-se praticamente constante. Isso é coerente com o resultado obtido em uma análise de variância do número de erros, na qual as seleções com tamanhos de quadrado decrescendo devem ter número de erros crescente (*p-values* 0.0008, 0.0075 e 0.0065 em comparação com os tamanhos de exemplo 8-4, 4-2 e 2-1 respectivamente) e os posicionamentos não devem apresentar uma relação entre os tamanhos de quadrados e número de erros (*p-values* 0.7834, 0.9463 e 0.8193 para as correlações de tamanhos de exemplo 8-4, 4-2 e 8-2 respectivamente).

Exposto isso, é possível afirmar que objetos de tamanhos grandes são fáceis de selecionar e com baixa ocorrência de erros de seleção. Objetos grandes são mais fáceis de posicionar do que objetos pequenos, mas, conforme visto na seção 6.4, tem maior dificuldade que a seleção de objetos grandes e médios. De qualquer forma, a hipótese de que quadrados maiores são mais fáceis de selecionar e posicionar foi confirmada, o que vai ao encontro do indicado pela *Fitts' Law* (FITTS, 1954).

7 CONSIDERAÇÕES FINAIS

Foi apresentado um estudo sobre um sistema interativo por gestos para um display público, que não exige do usuário a utilização de qualquer dispositivo. No sistema proposto, o usuário fica em frente a um display público e utiliza suas mãos para interagir com as informações presentes na tela, utilizando, para isso, gestos análogos aos quais utilizaria em outras situações cotidianas, como fechar a mão para selecionar um objeto e afastar dois pontos de contato para ampliar a tela.

O estudo de caso foi conduzido com o auxílio de três aplicações com funcionalidades diversas, de modo que fosse possível avaliar o método de interação proposto em cenários diferentes, com a exploração de todos os gestos previstos. De modo a avaliar como o sistema se comportaria em um ambiente público normal, onde o controle das possíveis situações é mínimo, foram conduzidos testes e observações acerca dos principais aspectos que interferem na execução de uma tarefa interativa em público, sendo eles: a diferença de iluminação do ambiente, a presença de outras pessoas junto ao interagente, o tipo do local onde o display pode ser utilizado, o tipo de tarefa a ser executada e a forma de apresentação das informações.

Baseando-se nos resultados obtidos, é possível afirmar que o sistema, apesar de possuir deficiências em determinados aspectos, se portou suficientemente bem para interagir com objetos grandes, em tarefas de seleção (como mostra a figura 7.1) ou translação à curta distância. Também fica claro que sistemas que empregam funcionalidades de *pan & zoom* podem se valer do método interativo proposto, já que usuários em um local púbico real foram capazes de interagir com as informações do display em uma aplicação desse estilo.

Ainda que não apresente uma solução definitiva para a interação em displays públicos, o trabalho serve para indicar as principais dificuldades ao se buscar uma solução para esse problema interativo. A proposta introduzida pelo sistema proposto, com o uso do fechamento das mãos para diferenciar a seleção da navegação parece bastante intuitiva para os usuários, bem como a utilização das duas mãos para modificar a intensidade de zoom na tela.

Os problemas que ficam são inerentes à tecnologia escolhida, que, apesar de ser de baixo custo financeiro e de fácil instalação e uso, ainda apresenta problemas típicos de tecnologias novas. Não há um consenso sobre a melhor forma de interpretar gestos realizados no ar, mas o Kinect se mostra uma boa solução com um baixo custo financeiro e de implantação, já que é portátil e não necessita de qualquer estrutura adicional para que possa ser utilizado. Espera-se que com o desenrolar dos avanços tecnológicos, uma segunda versão do dispositivo seja mais robusta e, em especial, utilize-se de câmeras de maior resolução para produzir as imagens de profundidade. Com o iminente lançamento do novo console de video-game da Microsoft, resta esperança de que realmente seja lan-

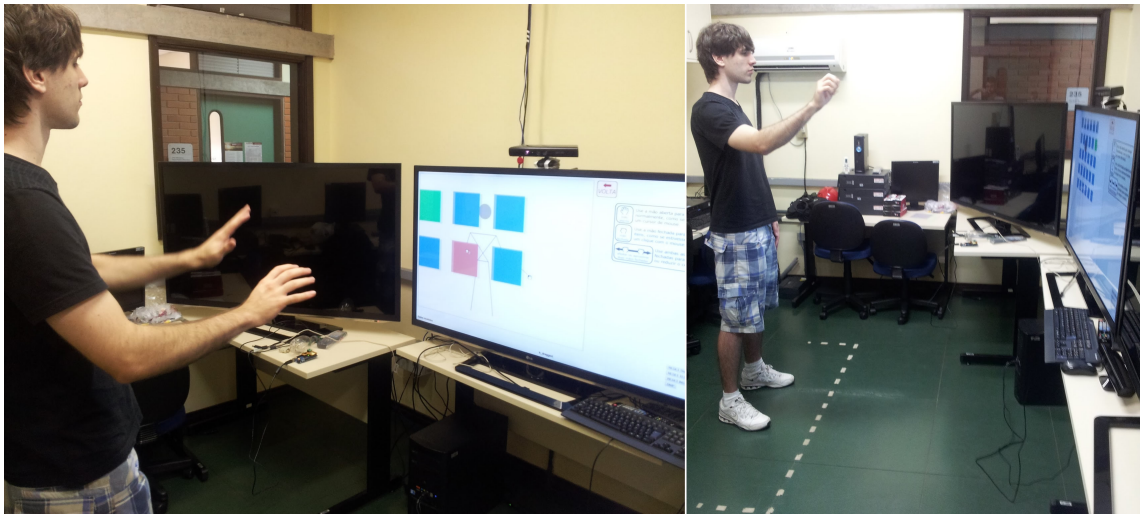

Figura 7.1: Duas fotos de um usuário realizando tarefas de seleção de quadrados – à esquerda na disposição inicial do tamanho de quadrados; à direita após os quadrados terem diminuído de tamanho e aumentado em quantidade uma vez – situações em que o sistema melhor se portou.

çada uma versão aperfeiçoada de seu periférico.

Por fim, cabe salientar novamente que a solução proposta nesse trabalho para prover uma interação gestual sem dispositivos para um display público é bastante interessante, pois torna independentes as aplicações de interpretação dos gestos e de apresentação da informação. Logo, qualquer desenvolvedor poderia se valer das informações obtidas pela aplicação que interpreta o Kinect e construir sua própria aplicação Web que faça o que ele desejar, desde que ela se conecte a um *WebSocket* e utilize a máquina de estados apresentada para decifrar as informações recebidas e obter a intenção do usuário. E tudo isso com um custo financeiro baixo, que compreende os valores necessários para a aquisição de um Microsoft Kinect e um grande display, além de um computador.

7.1 Notas adicionais

A partir dos resultados obtidos nas avaliações, é possível perceber que o sistema proposto pode ser utilizado em displays públicos desde que algumas restrições seja respeitadas. Aplicações baseadas em *pan & zoom*, como a do mapa do prédio, são potenciais usufruidoras do método de interação introduzido no trabalho. Aplicativos mais complexos, mas que possuam botões grandes e tarefas simples também são factíveis para o aproveitamento do método. Fica bastante claro que há deficiências no sistema proposto, mas cabe salientar novamente que os próprios fabricantes de jogos para o Kinect, que têm total acesso às capacidades do dispositivo e suporte da Microsoft, não conseguem produzir um software que utilize com robustez o reconhecimento de gestos em tarefas elaboradas.

O sistema proposto foi além do que o Kinect pode prover, possibilitando uma maneira de diferenciar as mãos fechadas e abertas, o que amplia as capacidades interativas da aplicação. Além disso, com a sua construção independente e o suporte a aplicações Web, a aplicabilidade do sistema é bastante extensa, já que é possível construir praticamente qualquer tipo de aplicação para Internet, especialmente quando são utilizadas as novas técnicas introduzidas pelo HTML5.

Durante o período de desenvolvimento do trabalho apresentado, muitos problemas e

dificuldades tiveram de ser superados e frequentemente esbarrou-se em limitações de tecnologia. Em um primeiro momento, a aplicação que interpreta os dados do Kinect havia sido construída sobre o SDK da OpenNI. Porém, ao buscar por bibliotecas que facilitassem a comunicação do aplicativo com *WebSockets* só foram encontrados problemas, ora de compilação ora de execução. A descoberta da biblioteca Fleck impulsionou a escolha do SDK da Microsoft, cuja utilização é recomendada com linguagem C#, justamente a contemplada pela biblioteca. Após a conversão do que já havia sido feito para o novo ambiente, percebeu-se que o SDK Kinect for Windows possuía outras tantas vantagens em relação ao anteriormente utilizado e que sua escolha foi um grande acerto de projeto.

Como a primeira implementação do sistema, que utilizava apenas as juntas do esqueleto e seus respectivos ângulos para diferenciar as tarefas interativas não funcionou bem o suficiente, passou-se à procura de uma forma de identificar os estados das mãos do usuário. Ao fazer uma pesquisa no Google sobre o assunto, diversas implementações aparecem, bem como alguns vídeos demonstrando que elas funcionam. Porém, buscar por suporte sobre os detalhes de implementação mostra-se uma tarefa bastante complicada, pois as poucas informações existentes não são claras. Fazer o download de exemplos ou, mesmo, de código-fonte, disponibilizado em alguns casos, não foi de grande ajuda, pois nenhuma das aplicações compilou ou sequer executou.

Felizmente, após uma procura de algumas semanas, uma das implementações mencionava o uso do algoritmo K-curvature e uma nova pesquisa no Google trouxe resultados interessante sobre seu funcionamento e suas aplicações, muitas das quais eram justamente para identificar as pontas dos dedos. Como o algoritmo é simples e as demais técnicas de processamento de imagem utilizadas no trabalho são bastante triviais, partiu-se para uma implementação própria, o que se mostrou extremamente mais eficiente do que todas as semanas investidas na pesquisa e teste de aplicações prontas.

Por fim, com os dados do Kinect sendo identificados com sucesso e com a integração das aplicações cliente e servidora funcionando, a construção, uso e aprimoramento da aplicação de visualização de grafos para utilização dos gestos propostos foi uma tarefa muito animadora. Ainda que não tenha se portado suficientemente bem para ser implantada em um ambiente real, a interação gestual sobre a aplicação mostrou-se melhor do que era esperado após um longo período de insucessos de implementação.

Ao inciar esse projeto, o objetivo final era a implantação do sistema em um prédio central da Universidade para que qualquer aluno ou ex-aluno de Pós-Graduação Stricto Sensu da UFRGS pudesse utilizá-lo e consultar sua árvore genealógica acadêmica. Esperava-se que tal realização atraísse os antigos alunos de volta à Universidade, encorajando-os a atualizarem seus dados junto à base de dados institucional, o que sempre é um grande interesse da Instituição. Infelizmente, no estágio atual parece que uma aplicação complexa como um visualizador de grafos não seja indicada para uso no sistema proposto – ao menos não com todas as funcionalidades desejadas. Contudo, abre-se margem para um aprofundamento de pesquisa, agora com um norteamento em relação aos problemas existentes.

Um futuro trabalho interessante a ser feito é integrar o sistema introduzido nesse trabalho com outro método interativo, como aquele provido por dispositivos móveis. Nesse panorama, o usuário visualizaria e interagiria com informações em um display público, mas a aplicação permitiria que ele utilizasse o seu próprio celular, por exemplo, para inserir informações, realizar interações finas – como apresentado no trabalho de (DEBARBA; NEDEL; MACIEL, 2012), no qual o usuário é capaz de realizar gestos precisos após uma seleção prévia dos elementos a serem manipulados – ou, ainda, receber uma cópia dos da-

dos. Dessa forma o usuário não interagiria diretamente com nenhum dispositivo que não o seu próprio, evitando eventuais problemas de compartilhamento e curvas de aprendizado.

Será interessante também experimentar uma versão do sistema que somente reconheça as mãos do usuário quando elas estão na área compreendida entre seus tórax e topo da cabeça, região em que o sistema se mostrou mais responsivo. Embora já seja possível adiantar que haverá perda de precisão nos gestos executados, cabe avaliar se tal perda será compensada pela maior robustez do reconhecimento dos gestos. Supõe-se que esse problema da região mais responsiva seja devido ao posicionamento do Kinect sobre o display, já que seu deslocamento para a parte de baixo da tela modificou a região de maior eficiência, mas resta um teste mais elaborado para confirmar essa hipótese. Se confirmada, ela indica que a posição ideal para a instalação do Kinect é no centro da tela, o que é inviável.

7.2 Contribuições

O desenvolvimento desse trabalho rendeu discussões e descobertas interessantes, algumas das quais se mostraram contribuições importantes para a pesquisa científica. A primeira versão do sistema desenvolvido utilizava a extensão do braço como diferencial interativo ao utilizar o Kinect para interagir com um display grande. Ela demonstrou que o dispositivo poderia ser utilizado nesse cenário e introduziu uma forma de integrar o reconhecimento de gestos provido por ele a uma aplicação Web. Em seu estágio intermediário, o trabalho foi publicado na seção de *Work in Progress* do SIBGRAPI 2012 - *Conference on Graphics, Patterns and Images*, ocorrido em Ouro Preto - MG, em agosto de 2012, sob o título *Deviceless Gestural Interaction in Public Displays*.

Após o desenvolvimento e aprimoramento da aplicação de visualização de árvore genealógica, mas ainda utilizando o primeiro modelo de interação, o trabalho voltou a ser publicado, dessa vez na CLEI 2012 - *XXXVIII Conferencia Latinoamericana en Informática*, com o título *Gestural Interaction for Manipulating Graphs in a Large Screen Using the Kinect Integrated to the Browser*. A Conferência ocorreu em Medellìn - Colômbia, em outubro de 2012.

Finalmente, após desenvolvida a nova versão de interpretação de gestos, que identifica o estado das mãos do usuário, e realizada a avaliação formal com usuários, foi submetido e aceito um novo artigo, na categoria de “artigos completos” do XV SVR - Simpósio de Realidade Virtual e Aumentada, em maio de 2013, em Cuiabá - MG. Os artigos aceitos e publicados no SIBGRAPI 2012, na CLEI 2012 e no SVR 2013 se encontram anexados a esse trabalho.

REFERÊNCIAS

ASHBROOK, D.; STARNER, T. MAGIC: a motion gesture design tool. In: HUMAN FACTORS IN COMPUTING SYSTEMS, 28., New York, NY, USA. **Proceedings...** ACM, 2010. p.2159–2168. (CHI '10).

BALL, R.; NORTH, C.; BOWMAN, D. A. Move to improve: promoting physical navigation to increase user performance with large displays. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, New York, NY, USA. **Proceedings...** ACM, 2007. p.191–200. (CHI '07).

BALLAGAS, R. et al. The Smart Phone: a ubiquitous input device. **IEEE Pervasive Computing**, Piscataway, NJ, USA, v.5, p.70–, January 2006.

BENKLER, Y. **The Penguin and the Leviathan**: how cooperation triumphs over self-interest. [S.l.]: Crown Publishing Group, 2011.

BIGDELOU, A. et al. Simultaneous categorical and spatio-temporal 3D gestures using Kinect. In: D USER INTERFACES (3DUI), 2012 IEEE SYMPOSIUM ON, 3. **Anais...** [S.l.: s.n.], 2012. p.53 –60.

BORING, S. et al. Touch projector: mobile interaction through video. In: HUMAN FACTORS IN COMPUTING SYSTEMS, 28., New York, NY, USA. **Proceedings...** ACM, 2010. p.2287–2296. (CHI '10).

CAO, X.; BALAKRISHNAN, R. VisionWand: interaction techniques for large displays using a passive wand tracked in 3d. In: ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, 16., New York, NY, USA. **Proceedings...** ACM, 2003. p.173–182. (UIST '03).

CHOUMANE, A.; CASIEZ, G.; GRISONI, L. Buttonless clicking: intuitive select and pick-release through gesture analysis. In: VIRTUAL REALITY CONFERENCE (VR), 2010 IEEE. **Anais...** [S.l.: s.n.], 2010. p.67 –70.

COUTRIX, C. et al. Engaging spect-actors with multimodal digital puppetry. In: NORDIC CONFERENCE ON HUMAN-COMPUTER INTERACTION: EXTENDING BOUNDARIES, 6., New York, NY, USA. **Proceedings...** ACM, 2010. p.138–147. (NordiCHI '10).

DEBARBA, H.; NEDEL, L.; MACIEL, A. LOP-cursor: fast and precise interaction with tiled displays using one hand and levels of precision. In: D USER INTERFACES (3DUI), 2012 IEEE SYMPOSIUM ON, 3. **Anais...** [S.l.: s.n.], 2012. p.125 –132.

FITTS, P. M. The information capacity of the human motor system in controlling the amplitude of movement. **Journal of Experimental Psychology**, [S.l.], v.47, n.6, p.381–391, 1954.

GALLO, L.; PLACITELLI, A.; CIAMPI, M. Controller-free exploration of medical image data: experiencing the kinect. In: COMPUTER-BASED MEDICAL SYSTEMS (CBMS), 2011 24TH INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2011. p.1 –6.

IASON OIKONOMIDIS, N. K.; ARGYROS, A. Efficient model-based 3D tracking of hand articulations using Kinect. In: BRITISH MACHINE VISION CONFERENCE. **Proceedings...** BMVA Press, 2011. p.101.1–101.11. http://dx.doi.org/10.5244/C.25.101.

JACUCCI, G. et al. Worlds of information: designing for engagement at a public multi-touch display. In: HUMAN FACTORS IN COMPUTING SYSTEMS, 28., New York, NY, USA. **Proceedings...** ACM, 2010. p.2267–2276. (CHI '10).

KOBOUROV, S. G. Spring Embedders and Force Directed Graph Drawing Algorithms. **CoRR**, [S.l.], v.abs/1201.3011, 2012.

KUIKKANIEMI, K. et al. From Space to Stage: how interactive screens will change urban life. **Computer**, [S.l.], v.44, n.6, p.40 –47, june 2011.

MCCALLUM, D. C.; IRANI, P. ARC-Pad: absolute+relative cursor positioning for large displays with a mobile touchscreen. In: ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, 22., New York, NY, USA. **Proceedings...** ACM, 2009. p.153–156. (UIST '09).

MICHELIS, D.; MüLLER, J. The Audience Funnel: observations of gesture based interaction with multiple large displays in a city center. **Int. J. Hum. Comput. Interaction**, [S.l.], v.27, n.6, p.562–579, 2011.

MOEHRING, M.; FROEHLICH, B. Effective manipulation of virtual objects within arm's reach. In: VIRTUAL REALITY CONFERENCE (VR), 2011 IEEE. **Anais...** [S.l.: s.n.], 2011. p.131 –138.

MOTTA, T.; NEDEL, L. Deviceless Gestural Interaction in Public Displays. In: WORKSHOP OF WORKS IN PROGRESS (WIP) IN SIBGRAPI 2012 (XXV CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES), Ouro Preto, MG, Brazil. **Anais...** [S.l.: s.n.], 2012.

MOTTA, T.; NEDEL, L. Gestural interaction for manipulating graphs in a large screen using the Kinect integrated to the Browser. In: INFORMATICA (CLEI), 2012 XXXVIII CONFERENCIA LATINOAMERICANA EN. **Anais...** [S.l.: s.n.], 2012. p.1–7.

MOTTA, T.; NEDEL, L. Deviceless Gestural Interaction for Public Displays. In: VIRTUAL AND AUGMENTED REALITY (SVR), 2013 15TH SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2013.

MüLLER, J. et al. Requirements and design space for interactive public displays. In: MULTIMEDIA, New York, NY, USA. **Proceedings...** ACM, 2010. p.1285–1294. (MM '10).

NANCEL, M. et al. Mid-air pan-and-zoom on wall-sized displays. In: HUMAN FACTORS IN COMPUTING SYSTEMS, 2011., New York, NY, USA. **Proceedings...** ACM, 2011. p.177–186. (CHI '11).

NI, T. et al. A Survey of Large High-Resolution Display Technologies, Techniques, and Applications. In: VIRTUAL REALITY CONFERENCE, 2006. **Anais...** [S.l.: s.n.], 2006. p.223 – 236.

OLWAL, A. LightSense: enabling spatially aware handheld interaction devices. In: IEEE AND ACM INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY, 5., Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2006. p.119–122. (ISMAR '06).

PEARS, N.; JACKSON, D. G.; OLIVIER, P. Smart Phone Interaction with Registered Displays. **IEEE Pervasive Computing**, Piscataway, NJ, USA, v.8, p.14–21, April 2009.

PELTONEN, P. et al. It's Mine, Don't Touch!: interactions at a large multi-touch display in a city centre. In: PROCEEDING OF THE TWENTY-SIXTH ANNUAL SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, New York, NY, USA. **Anais...** ACM, 2008. p.1285–1294. (CHI '08).

PREECE, J.; ROGERS, Y.; SHARP, H. **Design de Interacao**. [S.l.]: Bookman, 2005.

RICO, J.; BREWSTER, S. Usable gestures for mobile interfaces: evaluating social acceptability. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, New York, NY, USA. **Proceedings...** ACM, 2010. p.887–896. (CHI '10).

SHAKER, N.; ABOU ZLIEKHA, M. Real-time Finger Tracking for Interaction. In: IMAGE AND SIGNAL PROCESSING AND ANALYSIS, 2007. ISPA 2007. 5TH INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2007. p.141 –145.

STØDLE, D. et al. Gesture-Based, Touch-Free Multi-User Gaming on Wall-Sized, High-Resolution Tiled Displays. **Journal of Virtual Reality and Broadcasting**, [S.l.], v.5, n.10, Nov. 2008. urn:nbn:de:0009-6-15001,, ISSN 1860-2037.

TONG, J. et al. Scanning 3D Full Human Bodies Using Kinects. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, CA, USA, v.18, p.643–650, 2012.

VOGEL, D.; BALAKRISHNAN, R. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In: ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, 17., New York, NY, USA. **Proceedings...** ACM, 2004. p.137–146. (UIST '04).

VOGEL, D.; BALAKRISHNAN, R. Distant freehand pointing and clicking on very large, high resolution displays. In: ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, 18., New York, NY, USA. **Proceedings...** ACM, 2005. p.33–42. (UIST '05).

WILSON, A. D. Using a depth camera as a touch sensor. In: ACM INTERNATIONAL CONFERENCE ON INTERACTIVE TABLETOPS AND SURFACES, New York, NY, USA. **Anais...** ACM, 2010. p.69–72. (ITS '10).

XIA, L.; CHEN, C.-C.; AGGARWAL, J. Human detection using depth information by Kinect. In: COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS (CVPRW), 2011 IEEE COMPUTER SOCIETY CONFERENCE ON. **Anais...** [S.l.: s.n.], 2011. p.15 –22.

ANEXO I

Questionário de Pré-Avaliação

Pré-Questionário de Avaliação

* Required

Nome: *

Idade *

Sexo *

- M
- F

Lado motor predominante *

- Destro
- Canhoto

Nível de escolaridade *

Ensino Médio completo/incompleto

Já utilizou alguma TV interativa por gestos ou algum outro tipo de interação com telas grandes? *

- Sim
- Não

Já jogou jogos com o Kinect? *

- Sim
- Não

Se sim, como você considera seu grau de experiência?

	1	2	3	4	5	
Pouco experiente	○	○	○	○	○	Muito experiente

Como você considera seu grau de experiência com interação 3D em geral? *

Jogos de Wii, PS Move, TVs interativas, testes de IHC anteriores e demais situações interativas

	1	2	3	4	5	
Pouco experiente	○	○	○	○	○	Muito experiente

Com que frequência você costuma utilizar dispositivos sensíveis ao toque? *
Smartphones, tablets etc.

	1	2	3	4	5	
Pouco frequente (nunca ou quase nunca)	○	○	○	○	○	Muito frequente (todos os dias)

Submit

Never submit passwords through Google Forms.

ANEXO II

Questionário de Pós-Avaliação

Pós-Questionário de Avaliação

* Required

Nome *
Utilize o mesmo nome que no outro questionário

Como você considera a responsividade do sistema? *
O sistema interpretou corretamente o que você queria?

	1	2	3	4	5	
Pouco responsivo	○	○	○	○	○	Muito responsivo

Como você considera a tarefa de translação/movimentação da tela? *

	1	2	3	4	5	
Pouco responsivo	○	○	○	○	○	Muito responsivo

Como você considera a tarefa de zoom na tela? *

	1	2	3	4	5	
Pouco responsivo	○	○	○	○	○	Muito responsivo

Você completou a tarefa de SELECIONAR os quadrados? *

- ○ Sim
- ○ Não

Como você considera a tarefa de SELEÇÃO dos quadrados? *

	1	2	3	4	5	
Fácil	○	○	○	○	○	Dificil

Você completou a tarefa de MOVIMENTAR os quadrados? *

- ○ Sim
- ○ Não

Como você considera a tarefa de MOVIMENTAÇÃO dos quadrados? *

	1	2	3	4	5	
Fácil	○	○	○	○	○	Dificil

Qual tarefa foi mais complicada na sua opinião? *

- SELECIONAR os quadrados
- MOVIMENTAR os quadrados

Quão rápida, na sua opinião, foi a realização da tarefa? *

	1	2	3	4	5	
Muito devagar						Muito rápida

Quão divertida foi a realização da tarefa? *

	1	2	3	4	5	
Pouco divertida						Muito divertida

Quão exaustiva foi a realização da tarefa? *

	1	2	3	4	5	
Pouco cansativa						Muito cansativa

Use esse espaço para compartilhar suas opiniões sobre o sistema e a tarefa, caso desejar.

Submit

Never submit passwords through Google Forms.

ANEXO III

Script de Interação

Script de Interação

1 – Posicione-se em frente à tela;

2 – Selecione o quadrado verde, dentre os demais quadrados, azuis, fechando a mão sobre o quadrado em questão;

3 – Após a seleção do quadrado verde, outro ficará verde e deverá ser selecionado;

4 – Repita os passos 2 e 3 por 20 vezes;

5 – Posicione o quadrado verde dentro do contorno quadrangular em preto. Para isso, feche a mão sobre o quadrado verde e, sem abri-la, leve a mão até o contorno quadrado em preto;

6 – Após posicionar o quadrado verde, outro quadrado verde aparecerá em algum lugar da tela e deverá ser posicionado;

7 – Repita os passos 5 e 6 por 15 vezes;

8 – Durante qualquer momento na tarefa interativa é possível fazer *pan & zoom*. Para o *pan*, feche a mão sobre qualquer lugar da tela que não o do quadrado verde; para o zoom, feche as duas mãos e as afaste (*zoom-in*) ou aproxime (*zoom-out*);

9 – Após a seleção e posicionamento dos 35 quadrados, o teste se encerra.

ANEXO IV

Artigo do SIBGRAPI 2012

(MOTTA; NEDEL, 2012a)

Deviceless Gestural Interaction in Public Displays

Thiago Motta, Luciana Nedel
Institute of Informatics – UFRGS, Porto Alegre, Brazil
Email: {tsmotta,nedel}@inf.ufrgs.br

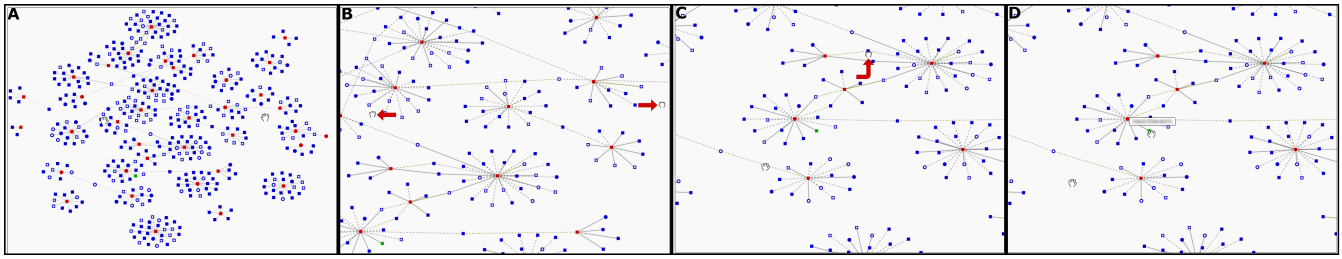

Fig. 1. Screenshots of a user performing a task in the system: (A) the initial screen of the application; (B) zooming into the graph to find a node; (C) translating the graph to put the green node in the center; (D) searching information of the green node.

***Abstract*—This paper presents a technique to provide natural interaction with public displays. The Microsoft Kinect is used to capture information about the user and an interpretation is made to identify what kind of interaction the user is trying to do. Gestures supported provide navigation, panning, and zooming, without any coupling devices on the user.**

***Keywords*-Gestural Interaction; Natural Interaction; HCI.**

I. Introduction

The use of large screens in public spaces such as airports, malls, bars and even squares in the streets informing upcoming events, promotions and other information that may be useful to users around it is becoming each day more common. Usually, these displays are not interactive and the user can only be updated with information. However, by providing interaction capabilities to the users, public displays have great potential for use, as has been seen in some recent works.

Traditional interaction devices are generally not suitable to interact with this kind of displays, especially when they are in public places, where there is no control over the environment. In these situations, it is suitable to provide interaction without any device additional.

Aiming to solve this problem, this paper presents a technique that uses Microsoft Kinect to recognize gestures of a user, providing a natural method of interaction without any direct use of devices. As a case study, an application that allows the interactive visualization of a social network was implemented and tested. This application runs in a browser, in which the Kinect was integrated to allow for user interaction.

II. Related Work

Researchers make use of various approaches to interact with large displays and the most common involves the use of touch-sensitive screens and devices that simulate such functionality [1]. Other strategies make use of mobile devices to interact with large screens [2], provide the development of new devices [3], or expand the functionality of existing devices, such as data gloves [4] for instance.

Even though all the strategies employed are able to solve the problem of interaction with large displays, there is scant research that does it without the use of any device by the user, as made by [5]. It is suitable especially when one wants a system capable of being conducted by different users in public areas.

This work wants to provide an intuitive interaction method that is low cost, robust, and capable of being used in actual uncontrolled environments.

III. Gestural Interaction

By developing an interaction method based on gestures in public areas, some problems must be taken into account, such as lighting, user identification, correct interpretation of user gestures, etc. Especially when it comes in a gestural interaction deviceless, to identify what the user is trying to do is not an easy task. In the approach used employing the Kinect to identify the user, the focus on the user is handled by the device, but other problems need to be addressed. In this work we try to identify some of these problems through the implementation of a case study.

A. Case Study

In the approach presented in this work, the user interacts with an information system using his hands and arms to browse and select virtual objects in a public display. The case study used is an application that runs in a Web browser, and shows a graph representing the academic social network of the Computer Science Graduate Program (PPGC) at UFRGS (see Figure 1). In this graph, nodes represent persons (professors and students), and edges indicate supervising relationships. In other words, the graph represents the genealogy of the PPGC.

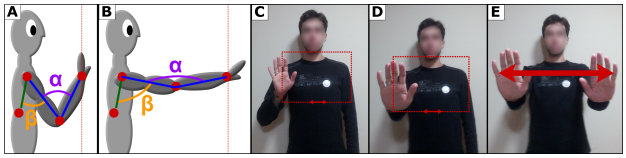

Fig. 2. Gestures in the system: (A) user with the arm close to their body indicates navigation; (B) user with outstretched arms indicates selection; (C) user performing navigation in the system; (D) user performing manipulation in the system; (E) user performing zoom-in and zoom-out, in a movement similar to those used in touch-screens.

To interpret the users' gestures, the system uses joints of their skeleton, easily identified by Kinect. For navigation, the system read the joints of the hands of the user and a selection is made when the user stretch his arm.

The depth information provided by Kinect is not accurated enough to identify this movement and, because of this, the response of the system became unstable. To solve this problem, the angles formed by the joints of the user's arm were used: between the arm and forearm (α), and between the arm and the side of the user (β). Thus, if α is greater than 110° and β is greater than 70°, the system interprets that the user is making a selection (Figure 2-B). Otherwise, the user is navigating (Figure 2-A).

In order to make these calculations, wrist, elbow and shoulder joints were used both on left and right sides of the skeleton, while the spine joints were used once. To provide robustness to the system, when the user reaches out completely toward the Kinect – in which case the device cannot correctly detect the skeletal joints – the depth information is used. This arrangement brings stability: when the user's hand is at the maximum distance that can be reached regarding the rest of his body (the dashed line in Figure 2-A,B).

The features developed in the case study are: nodes and graph manipulation, and a tooltip that shows the name of the person represented by the node being pointed by the cursor. The gestures used are: free movement of the hands along the body (Figure 2-C), for navigation; free movement with hands away from the body (Figure 2-D) for selection and manipulation; and approaching and moving apart both hands with the arms outstretched, to zoom in and out (Figure 2-E).

To integrate the Kinect with the browser, the client-server model was applied. The server is responsible for reading and interpreting the data obtained by Kinect and sends the information to the test application client. Two 3D coordinates are transmitted, corresponding to the positions of both hands, and two integers indicating that both hands are in navigation (0) or selection (1) state. With these data, the Web application can interpret which action is being taken: just moving the cursor around the graph to view information of the nodes; moving the entire graph or each individual node in the X and Y axes; or zooming the graph using both hands.

B. Implementation Details

The Microsoft official SDK was used to interpret data acquired with Kinect. The test application was developed in HTML5, JavaScript and PHP and runs on Web browsers. The library arbor.js (arborjs.org) was used to build the graph.

IV. Preliminary Results

Although the proposed solution does not work with extreme accuracy, using the zoom it is possible to manipulate individual nodes quite easily, even that some problems occur when the user stretches the whole arm and the system does not identify the desired selection. The solution presented for the selection mode, using angles of joints aided by depth information worked well enough, but still needs to be improved.

The solution presented behaved better than solutions that use only the depth information for selection, or those that use the absence of movement in which the user leaves their hand for a few seconds. However, it remains to be implemented a version that interprets the opening and closing movement of the hands to select objects. Details for this implementation are now being analyzed and the question is whether such identification does not adversely affect system performance.

V. Final Considerations

In this paper we present the first steps towards an implementation of a gestures-based interface for public displays. The Microsoft Kinect was used to identify the user and from this information, the selection and manipulation tasks were identified. The approach used runs in real-time and opens room for future improvements. A deep user study will be done to evaluate the robustness and usability of the system.

Acknowledgment

This work was partially supported by CNPq-Brazil under the project 311547/2011-7 and by Microsoft Brazil Interop Lab at UFRGS.

References

[1] G. Cohn, D. Morris, S. N. Patel, and D. S. Tan, "Your noise is my command: sensing gestures using the body as an antenna," in *Proceedings of the 2011 annual conference on Human factors in computing systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 791–800. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979058

[2] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch, "Touch projector: mobile interaction through video," in *Proceedings of the 28th international conference on Human factors in computing systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 2287–2296. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753671

[3] L. Aguerreche, T. Duval, and A. Lécuyer, "Reconfigurable tangible devices for 3d virtual object manipulation by single or multiple users," in *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '10. New York, NY, USA: ACM, 2010, pp. 227–230. [Online]. Available: http://doi.acm.org/10.1145/1889863.1889913

[4] M. Moehring and B. Froehlich, "Effective manipulation of virtual objects within arm's reach," in *Virtual Reality Conference (VR), 2011 IEEE*, march 2011, pp. 131 –138.

[5] M. Nancel, J. Wagner, E. Pietriga, O. Chapuis, and W. Mackay, "Mid-air pan-and-zoom on wall-sized displays," in *Proceedings of the 2011 annual conference on Human factors in computing systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 177–186. [Online]. Available: http://doi.acm.org/10.1145/1978942.1978969

ANEXO V

Artigo da CLEI 2012

(MOTTA; NEDEL, 2012b)

Gestural Interaction for Manipulating Graphs in a Large Screen Using the Kinect Integrated to the Browser

Thiago Motta, Luciana Nedel
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, Brazil
Email: {tsmotta,nedel}@inf.ufrgs.br

***Abstract*—Gestural interaction has gained prominence in recent years and various devices and methods for providing it have been developed in both the Academy and the Market. Microsoft's Kinect, which is able to identify the skeleton of the interacting user in real time, is the latest example of this fact, and has been the subject of recent research. This paper presents a way to integrate Kinect on the Browser, so that the user can interact face to a high-resolution screen without having to directly manipulate any device and have all the portability and availability that the Internet provides. In the case study presented, the user interacts with a graph representing a social network and is able to identify and move nodes, in addition to pan and zoom the graph as a whole. The paper concludes that such a mechanism has great potential for use and presents future work.**

***Index Terms*—HCI, Gestural Interaction, Public Displays.**

I. Introduction

When thinking about natural interaction, what usually comes to mind are methods introduced in science fiction movies as *The Matrix* and *Minority Report*, in which the user uses their own hands to perform the desired tasks on the computer. However, the farther we are to achieve the high technology portrayed in the movies, gestural interaction is increasingly taking charge of the devices with which we interact daily.

In recent years, the study of Human-Computer Interaction gained prominence and even the Market began to see the interaction as an important feature of Computer Science. Once the research were focused on hardware or software performance, in several variations, but the interaction with the machine was always done conventionally by keyboard and/or mouse or some other device with buttons and never was a relevant factor of the work. However, this scenario began to change gradually in recent decades and became clear that conventional interaction devices did not have good performance on some tasks, beginning thus studies to design more adequate methods, especially when interacting in virtual environments of Computer Graphics.

With the launch of the Nintendo Wii[1] in 2006, and the first iPhone[2] in 2007, non-conventional interaction started to be used by the masses and, with the consequent explosion of commercial devices employing differentiated interaction, academic research gained even more strength (and adepts).

Whether for appearing in science fiction movies or for presenting an exact parallel with the natural daily interaction of human beings, among all non-conventional methods of interaction, interaction through gesture recognition has always been attractive to researchers. With the advent of Microsoft's Kinect[3], which provides recognition of the skeleton with a low computational and financial cost and acceptable performance, it soon became the object of research in the Academy. An example is the work of Tong *et al.* [1], which mentions that the Kinect device is "compact, inexpensive and easy to use" device.

The gestural interaction is particularly useful when the user is interacting standing in front of large displays, situations in which there are no mouse and keyboard to perform the interaction – and they are not desirable –, since such devices are not suitable for this situation. This is precisely the problem that this paper aims to solve, with the introduction of a method that integrates gesture recognition provided by Kinect with the versatility of Web-based applications. In other words, this paper presents a way to send the data identified by the Kinect to the user's web browser and thus utilize them to perform tasks available on the interactive application.

As a case study, we present an application that displays a graph representing the academic social network of the *Programa de Pós-Graduação em Computação* (PPGC) of UFRGS, in which nodes represent students and teachers and edges represent the relations of orientation and co-orientation between them. The user is able to manipulate the graph as a whole, panning and zooming, and the individual nodes, querying information about the person they represent and repositioning them.

Following, Section II presents the related work, which also

[1] http://us.wii.com/hardware

[2] http://www.apple.com/br/iphone

[3] http://www.xbox.com/pt-br/kinect

978–1–4673–0793–2/12/$31.00

seek ways to interact in front of a large display. Section III describes the model proposed by this work, with its specific techniques and how to identify gestures. Implementation details are presented in Section IV and Section V presents the preliminary results obtained by the system. The work is concluded in Section VI, which also describes the work to be developed in future.

II. Related Work

When dealing with the interaction in high-resolution screens, the interaction paradigm should be treated differently from the conventional model adopted, because as well as change the visualization paradigm, so does the interaction one [2]. However, despite the large number of research being done, the ideal way to interact with large displays has not been discovered yet. Ni *et al.* [3] present a Survey which addresses some criteria that must be met when dealing with this kind of display.

In the search of the best way to interact in this kind of scenario, the researches on gestural interaction tend to follow very different approaches. They can be grouped as follows: interaction with touch sensitive screens, interaction via mobile devices, interaction provided by various devices, and interaction performed without the aid of any device directly. Each type has advantages and disadvantages and the choice of the designer must be in accordance with the respective interactive task. However, whatever the task, there is still no consensus on what is the most suitable device to conduct this type of interaction.

The works of Stødle *et al.* [4] and Wilson [5] presents systems based on multiple cameras to capture the user's taps on a surface. In particular, Peltonen *et al.* [6] and Jacucci *et al.* [7] presents two works based on the same system of touch sensitive screens of the Helsinki Institute for Information Technology: City Wall and Worlds of Information, respectively, in which users interact with large screens at public places. Although the touch recognition in general is quite efficient, the systems proposed for providing this functionality have high implementation cost, and in cases using cameras, also have problems of occlusion.

Another widely used technique to perform interaction with large displays is the use of mobile devices as interactive tools. Smartphones, in particular, have been widely used in HCI literature, and are considered essential in contemporary life and "the first computational device that actually won the society, despite of social class" [8]. Most studies that use mobile devices do the interaction by optical means, being pattern recognition, as the works of Boring *et al.* [9] and Pears *et al.* [10], or tracking the device, as the work of Olwal [11], although there are alternatives that maps the mobile device's screen to the big screen, as the work of McCallum and Irani [12]. However, the economic reality of Latin America is inconsistent with the developed countries and it is still a small percentage of people who owns smartphones - and even less than that of those who would know how to use them in interactive tasks not strictly related to the devices.

In the development of new devices or interactive techniques, Nancel *et al.* [13], Cao and Balakrishnan [14], Vogel and Balakrishnan [15], Moehring and Froehlich [16], Aguerreche *et al.* [17] and Ball *et al.* [18] present works in which various devices are used to perform the tasks, such as wireless mice and recognition of reflective surfaces (objects made of a material that reflects infrared emissions, becoming thus identifiable by specific devices). In some cases, the recognition of gestures is quite robust, especially with those techniques that use reflective surfaces. However, again these works are uneconomical to be available for the masses and moreover bring the drawback of forcing the users to hold or have attached to them a specific device, which is undesirable in many cases.

Finally, regarding to deviceless interaction the literature is still with low volume and depth. The continuity of the work of Nancel *et al.* [13] shows users interacting freely in front of a tiled-display, but do not detail the techniques used for motion capture. The work of Cohn *et al.* [19] search a very original approach, making the interpretation of electromagnetic noise to identify user interaction. Although there are not a large amount of papers about the interaction without devices, this scenario should change soon, especially with the academic exploitation of the Kinect, which provides a series of pre-interpreted data via the SDK (Software Development Kit) and with an excellent cost-effective relation. The present work is an example of this.

III. Proposed Model

With the constant evolution of hardware, the interaction methods must be adapted to suit the technological innovation. A clear example of this is the growing use of large-displays or tiled-displays, i.e. large screens where the user can access a great amount of information simultaneously. Facing these screens, the user does not have the comfort of sitting at a table and use the conventional mouse and keyboard to interact, because he needs to stand up and often move physically to access all the information and manipulate the data as he wants.

Aiming to solve this problem, we want to have a system that allows the user to interact freely in front of a high-resolution screen without any coupling devices such as bracelets, hats etc.. In an ideal environment, the user would stand in front of the screen and naturally "discover" what he need to do in order to interact with the system, because the interaction would be as natural as possible. However, finding the ideal method of interaction is not a trivial task.

In the present approach, the user interacts with the system with his hands and arms, performing navigation and selection in the system. The case study used is an application that runs in the browser, which shows a graph representing the academic social network of the *Programa de Pós-Graduação em Computação* (PPGC) of the Federal University of Rio Grande do Sul, in which the nodes are teachers and students and the edges are relations of orientation and co-orientation within the scope of the Courses in recent years.

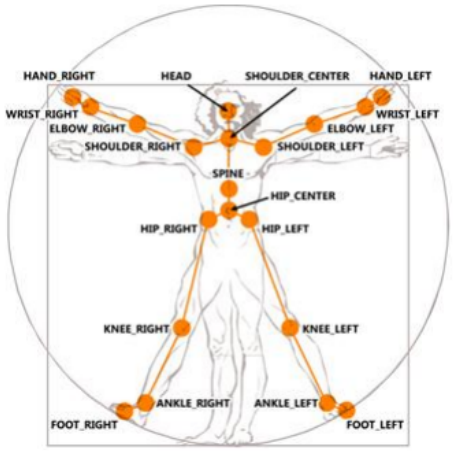

Fig. 1. The skeleton as identified by Kinect, with the 20 detectable joints named.

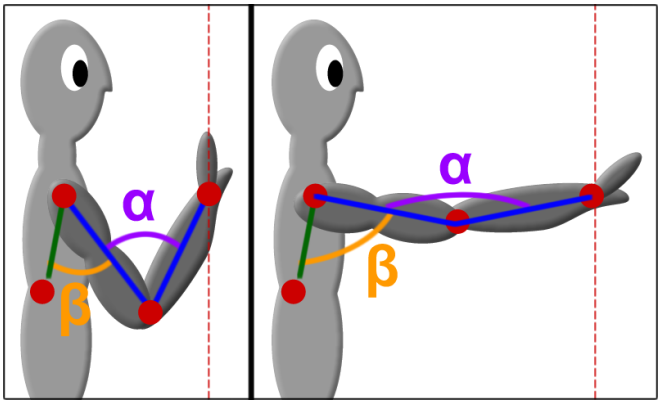

Fig. 2. The system of angles used to identify gestures for navigation and selection: A) user with the arm close to his body indicates navigation; B) user with an outstretched arm indicates a selection.

To better describe the proposed system, each of the parts of the system is presented individually: first, the program that reads and interprets the Kinect's data; secondly the application used as a case study; and finally the integration between these two subsystems.

A. The Kinect

Microsoft's Kinect, launched in 2010, entered the Guinness Records Book as the fastest-selling electronic device, having sold 8 million units in just 60 days[4]. The device features a depth camera, an RGB camera and a microphone array, which allows 3D motion capture, facial recognition and voice recognition (still without support for Portuguese). However, the main feature is that, through a combination of information obtained by their cameras, it can identify a skeleton of the user in front of it, composed of 20 joints, as shown in Figure 1.

In possession of these joints' data, it is necessary to interpret the gestures of the interacting user to determine whether he is performing a movement of navigation or selection. Making an analogy with the mouse, the system must identify the position of the mouse cursor as the position of a given hand, and the click of a button as the way this hand is positioned.

To identify the "click", i.e. the selection movement of hands, it was initially thought to use only the depth information provided by Kinect. However, it became clear that this information was not accurate enough and proved quite unstable in some configurations of the user's hand. To solve this problem, we identified the angles formed by the joints of the user's arm: between the arm and forearm (α), and between the arm and the side of the user (β). Thus, if α is greater than 110 degrees and β is greater than 70 degrees, the system interprets that the user is making a selection (Figure 2-B). Otherwise, he is in navigation mode (Figure 2-A).

In order to make these calculations we used *wrist*, *elbow* and *shoulder* joints, both on left and right side of the skeleton, and the joint of the *spine* in common for both sides. This approach performed quite well, except in the specific case where the user's arms are completely stretched in the direction of the Kinect. In this configuration, the device could not identify the joints of the skeleton correctly, because they present themselves almost one in front of another in this situation, and in such cases the system does not detect a user selection correctly. Fortunately, this configuration is exactly when the depth information is more reliable, because the user's hands are at the maximum distance they can be from the rest of his body (marked by the dashed line in Figure 2) . So if the difference between the depth of the joint of the wrist and the joint of the spine of the user is greater than a *X* value, it is also considered that he is performing a selection, complementing the interpretation of the angles.

B. Case Study

In order to probe the proposed interaction techniques, it was decided to create a real use application, i.e., that represented a product that really could be found commercially. With the growing interest in the research on social networks, both in areas of Computer Science and Communication, it was decided to create a tool to visualize the academic social network of the Graduate Programs in Computer Science of the Federal University of Rio Grande do Sul. The view would be made in the form of a graph in which nodes are students and teachers and edges are the relationships of orientation and co-orientation between them. The application should have at least the information of name, course and type of orientation represented by the elements of the graph.

The features developed are: whole graph and individual nodes manipulation, the display of a tooltip with the name of the person represented by the node that the cursor is pointing, and the display of academic information in a sidebar, such as date of entry into the University, Course (if students) or Department (if professors) etc.. Nodes are displayed in red when the person represented by him is a professor and blue if it is a student, except if the node represents the person who

[4]http://en.wikipedia.org/wiki/Kinect

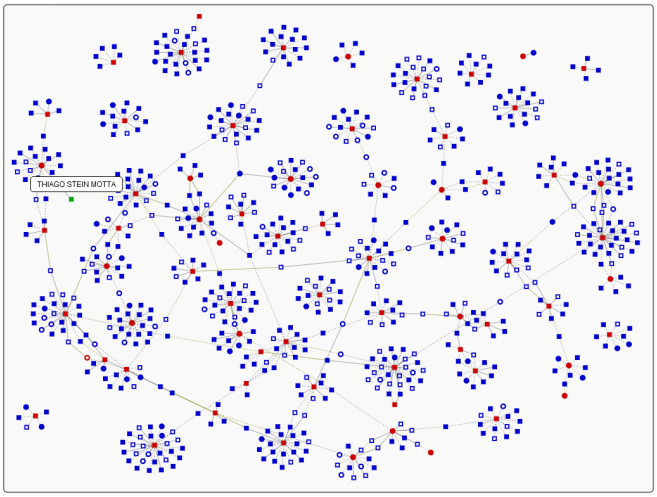

Fig. 3. The social network of the *Programa de Pós-Graduação em Computação* of UFRGS displayed as a graph in the application. Nodes are students (blue) and professors (red) and edges are relations of orientation (gray) and co-orientation (brown) between them.

is interacting with the system, in which case it is drawn in green.

Figure 3 shows the complete social network of the PPGC as displayed in the created application. The square nodes represent male students or teachers, while the round representing the female ones. Nodes with white dots are PHD students, while the filled are Masters. Gray edges represent orientation relations and brown co-orientation. Dotted edges are already finished relationships and solid ones are relationships that are on course. As can be seen in Figure 3, people are interconnected with each other through shared guidance between teachers or through the exchange of counselors by the students. Yet, as is possible to note, some counselors do not relate to the others, representing "islands" in the graph.

C. Connecting the Kinect to the Web

Since the browser is not able to directly recognize devices connected to the computer, there is no way to interpret the Kinect's data directly in programming languages for the Internet. Therefore, we used the Client-Server Model to make this integration possible. The Server is responsible for the reading and interpretation of data obtained by the Kinect and transmits the interpreted information for the Client application.

Two 3D points (x, y, z) corresponding to the positions of both hands and two integers indicating that each hand is in navigation (0) or selection (1) state are transmitted. With these data, the web application can interpret which action is in course: just moving the cursor in the graph to view information of the nodes; translating the entire graph or an individual node in the X and Y axes; or zooming the graph using both hands in a motion similar to that used to zoom in touch sensitive screens. Figure 4 shows the types of gestures recognized by the system, used to perform the features available in the application: navigation (Figure 4-A), selection (Figure 4-B) and zoom (Figure 4-C).

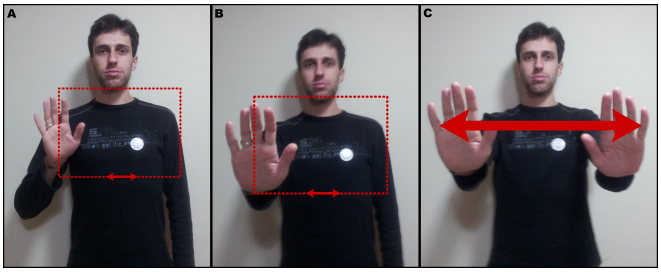

Fig. 4. The actions available in the application: A) navigation in the graph, B) manipulation of the graph or of a specific node, C) zooming in the graph.

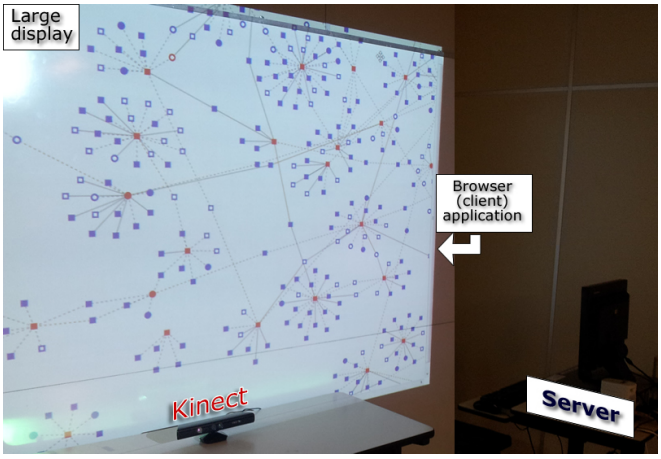

Fig. 5. The layout of the system in a room that uses a projected screen.

Thus, the system is ready to be installed anywhere, just needing to run the Server program and open the Client website in a modern Web browser. Figure 5 shows the proposed system being used in a room with a projective screen, a typical example where interaction with keyboard and mouse is not appropriate.

IV. Implementation Details

This section describes the technical details of the construction of this system, showing the choices that had to be taken and the main difficulties that presented themselves during the development process. As the system is designed in two very distinct parts, for clarity, once again each of the subsystems is presented in a separate subsection.

A. Reading and Interpreting the Kinect

When they launched the Kinect, Microsoft has ruled out any possibility of supporting the connection of their device to Personal Computers and threatened to legally sue anyone who could break the encryption used by Kinect to send the captured data. However, shortly after launch, the first library that could identify the Kinect and read its data in a Personal Computer was released: the libfreenect[5]. Later, the author of the library and other coders joined to create the first unofficial SDK (Software Development Kit) to support the use

[5]https://github.com/OpenKinect/libfreenect

of Kinect on a PC, called OpenKinect[6]. This SDK allows the developer to access the images provided by Kinect's cameras and its motor mechanisms, but not automatically identifies the skeleton mentioned above.

The OpenNI[7], used in conjunction with the Prime Sense NITE[8] is another unofficial SDK, which appeared sometime after the first. As the Prime Sense is the company that developed the camera's system used by Kinect, this SDK has a much broader support and is generally chosen to develop applications that use the Kinect in the Academy or by HCI enthusiasts. With a specific pose for user calibration, the SDK provides the position of the 20 joints of the skeleton in 3D space and is also able to identify specific gestures made with hands, such as *wave* and *push*.

However, seeing that a number of developers and researchers were interested in working with the Kinect, Microsoft decided to support this practice and, in June of 2011, released their own SDK[9], that could be downloaded and used freely for noncommercial purposes. Shortly afterwards, the SDK has been improved and a new version was released, but still both versions are available for download. The Microsoft's official SDK also has recognition of the skeleton, with the important difference that it is not necessary a calibration pose by the user. The recognition of joints is also more accurate than that of the OpenNI and the manner of use it is much simpler. The only drawback is that it only supports the use of the Microsoft's .NET Framework.

As already described, the Server subsystem was built to read and interpret the Kinect's data and transmit them to the browser. This is done via *WebSockets*. This is the best alternative for communication between web systems, because WebSockets are lightweight and are supported in all modern browsers.

As the OpenNI/NITE's SDK is the most widely used and provides the data that would be needed for this subsystem - besides having previously been used for studies of the Kinect by the authors of this paper - it was the choose one at first. However, due to consecutive failures in using libraries to deal with WebSockets in C++, it was decided to use the official SDK instead. Besides the ease with which it is manipulated via C#, the Microsoft's SDK has greater robustness in recognition of the user, not having as much instability as its competitor.

While libraries that manage WebSocket's connection in C++ were complicated both to understand and to use, in C#, running on .NET Framework, was found a very robust and practical library to handle the message exchanges with this type of socket: the Fleck[10]. Therefore, after repeated unsuccessful attempts to use the OpenNI/NITE's SDK, the quick results that the approach using the Microsoft's SDK facilitated the work of integrating the two applications afterwards.

The development of the application that identifies, reads and interprets the data provided by Kinect was developed in C#, in Visual Studio 2010. When preliminary tests on the subsystem developed were made, it was found that the information of hand's position and of which function they were being used for – navigation or selection – were being captured with sufficient stability. Therefore, it was decided that it was ready to be integrated into the Client application of the case study.

B. Building the Test Application

Once defined that the application of the case study would be implemented in programming languages for the Internet, frameworks and libraries that would aid in the construction of graphs in Web environments were surveyed. After being discarded technologies that use Adobe Flash[11] as well as the creation of a Java Applet, we came upon the library arbor.js[12]. This library helps in creating and managing a force-oriented graph, but leaves the user to freely choose the technology and the way in which the graph will be drawn. That is, as reported by the library website: "the code you write with it can be focused on the things that make your project unique – the graph data and your visual style – rather than spending time on the physics math that makes the layouts possible".

With the permission of the Department, the data related to orientation and co-orientation in the Computer Science Graduate courses were searched directly in the institutional database. The application was developed as a website, which is composed of two main areas: the graph drawing area and a sidebar information area. The website was built in PHP, JavaScript and HTML languages, using HTML5 techniques for drawing 2D objects on a *canvas* element. The structure of the graph is controlled by potential forces with the aid of the library arbor.js, mentioned above. The pilot application runs locally on an Apache Server and displays with better performance by the Google Chrome browser[13], which has the better management of threads between modern browsers, and, consequently, the greatest processing power.

When started, the application loads the data from a CSV (Comma Separated Values) file and builds the graph by adding the nodes and then the edges. The library decides the initial position of each node and proceeds to seek an equilibrium in which all of them are visible. The application, as a Client, must connect to a WebSocket and receive information from the Server subsystem by message exchange. The data is received in text format and converted to JSON, a format that is simple and easy to use with the JavaScript language.

V. Preliminary Results

In a preliminary evaluation of the system, it is possible to see that the use of the Kinect for applications on large screens has great potential. Without the need to directly use any device to carry out interactive tasks and without even having to do any calibration by the user, users interacted freely with the system without receiving prior instructions about its use. The first

[6]http://openkinect.org
[7]http://openni.org
[8]http://www.primesense.com/nite
[9]http://www.kinectforwindows.org
[10]https://github.com/statianzo/Fleck
[11]http://www.adobe.com/en/products/flash.html
[12]http://arborjs.org
[13]http://www.google.com/chrome

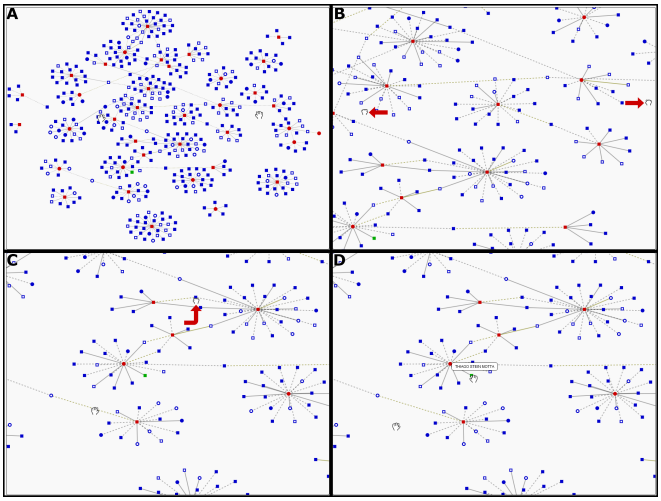

Fig. 6. Manipulation of the graph in order to see the information of the green node: A) the initial screen of the system; B) user performs a zoom in the graph; C) user moves the graph in X and Y axes in order to put the green node more centralized on the screen as possible; D) user places the cursor corresponding to the right hand above the green node, consulting its information in the tooltip and in the sidebar (omitted in this view).

performed tests were all informal, made by only four different users. Still, it was possible to glimpse a range of possibilities for use of the Kinect on the Web, because once the integration process has been defined, future applications may benefit from the work already done. Figure 6 displays four screens of the system that were captured in the course of a task in which the user located and centralized the node displayed in green. First, it was done a zooming in the graph, which then was moved so as to leave the green node in the center of the screen. Finally, the cursor corresponding the user's right hand was positioned on top of the node to display its information.

Unfortunately, for performance reasons, it was not possible to run tests on the complete graph. As all the processing for drawing the graph and the routines needed to interpret the data received by the Server program are interpreted by the browser in JavaScript, the test application was very slow when displaying all nodes and edges. On a computer with more processing power this problem could be solved, but there was no possibility to test this hypothesis yet.

Although the proposed solution does not work with extreme accuracy, with the help of the zoom is possible to manipulate individual nodes with ease, even when some problems occur when the user stretches the whole arm and the system does not identify the desired selection. The solution presented for the selection mode, using angles of joints aided by depth information worked well enough, but still needs to be improved. It is possible that an initial calibration for each user was able to solve some problems, however this need is not desirable. Another problem to be analyzed is the slight loss of position when the user performs the selection. In general, the system identifies as if the user had lowered his hand slightly, besides being making the selection. With continued use of the system, the user realize he need to make the selection of nodes with a slight "fix" of the hand position, but this solution is far from ideal.

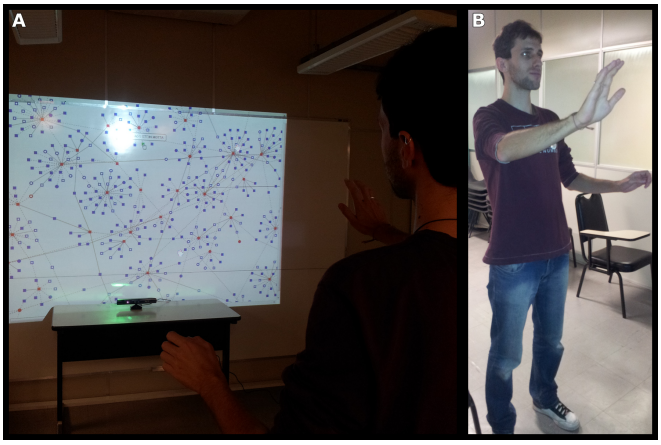

Fig. 7. User interacting with the system: A) view of the user and screen; B) front view of the user, where you can see that there is no support that would enable the use of mouse or keyboard. In this case, the proposed system presents a good alternative.

Although the proposed system has a number of qualities, many improvements still need to be made. The issue of engine performance when interacting with the complete graph need to be examined with caution, since optimizations in the construction of the software may be possible and a simple code refactoring can be enough to solve it.

The selection method also needs to be revised, because the inconveniences of the user's virtual hand suffer a slight shift when the selection is carried out must be avoided. Researches will be made in order to capture the state of the user's hands for replacing the system of joint angles, i.e. identifying when the user is with his hand open or closed. However, this challenge has no trivial solution, since any of the existing SDKs has this feature already integrated. We need to do a pre-processing of the image read by the Kinect to identify the position of the user's hands and thus discover whether they are open or closed. However, this calculation needs to be done in real time, so that no delay is introduced in the system, what can become a great challenge.

VI. Conclusions & Future Work

The paper presented a methodology for integrating Kinect to the browser to provide the possibility to interact by gestures with the application being used. This approach is especially useful when the interacting user is standing in front of a big screen with a large amount of data, as shown in Figure 7. In such a situation, mouse and keyboard are not appropriate and sometimes are even undesirable – for example, when interacting with public displays, without any security or support for the use of specific devices.

Even though it has been shown not to be ideal, the presented solution is a model for future applications that seek to improve this technique. Especially when interacting with a graph visualization system – very useful when you want to visually

study social networks – the proposed method worked well enough, providing the user with the most common interactive possibilities in this type of system.

Although it has not been stable enough, the presented solution for selection by gesture recognition is an alternative approach for the commonly used selection by absence of movement, when the user needs to keep his hand still in a certain place to make the selection. This approach would not work with the application of graph visualization or any proposal that allows pan and zoom, because a selection cannot be done associated with motion, as it depends on the absence of it to work. The angles+depth selection was also more effective and efficient than the approach that only uses the depth information, because this information is often unstable (particularly in situations in which the two hands are too close together or close to the body).

When the system is improved and robust enough, formal evaluation tests should be performed to verify the feasibility of it. In order to achieve this, the definition of goals to be pursued by testers will be needed, for example, find a specific node or find the shortest path of interconnections between a node and another. Only after a formal evaluation with users we will be able to say with certainty that the system can be used by the general public, as desired.

Finally, when the system is already settled and tested, if it is concluded that the interaction model is robust and effective enough, an exhibition of the work for the academic community of the University will be organized. Thus, students and former students of UFRGS will be able to check where they stand in the social network of their Graduate course, interacting freely in front of a large public display placed in a central building of the University. And, of course, without having to hold any device for this.

ACKNOWLEDGMENT

This work was partially supported by CNPq-Brazil under the project 311547/2011-7, and by Microsoft Brazil Interop Lab at UFRGS. We also thanks CPD-UFRGS that supports Thiago Motta.

REFERENCES

[1] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, pp. 643–650, 2012.

[2] D. Keefe, "Integrating visualization and interaction research to improve scientific workflows," *Computer Graphics and Applications, IEEE*, vol. 30, no. 2, pp. 8 –13, march-april 2010.

[3] T. Ni, G. Schmidt, O. Staadt, M. Livingston, R. Ball, and R. May, "A survey of large high-resolution display technologies, techniques, and applications," in *Virtual Reality Conference, 2006*, march 2006, pp. 223 – 236.

[4] D. Stødle, T.-M. S. Hagen, J. M. Bjørndalen, , and O. J. Anshus, "Gesture-based, touch-free multi-user gaming on wall-sized, high-resolution tiled displays," *Journal of Virtual Reality and Broadcasting*, vol. 5, no. 10, Nov. 2008, urn:nbn:de:0009-6-15001,, ISSN 1860-2037.

[5] A. D. Wilson, "Using a depth camera as a touch sensor," in *ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '10. New York, NY, USA: ACM, 2010, pp. 69–72. [Online]. Available: http://doi.acm.org/10.1145/1936652.1936665

[6] P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, and P. Saarikko, "It's mine, don't touch!: interactions at a large multi-touch display in a city centre," in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 1285–1294. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357255

[7] G. Jacucci, A. Morrison, G. T. Richard, J. Kleimola, P. Peltonen, L. Parisi, and T. Laitinen, "Worlds of information: designing for engagement at a public multi-touch display," in *Proceedings of the 28th international conference on Human factors in computing systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 2267–2276. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753669

[8] R. Ballagas, J. Borchers, M. Rohs, and J. G. Sheridan, "The smart phone: A ubiquitous input device," *IEEE Pervasive Computing*, vol. 5, pp. 70–, January 2006. [Online]. Available: http://dx.doi.org/10.1109/MPRV.2006.18

[9] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch, "Touch projector: mobile interaction through video," in *Proceedings of the 28th international conference on Human factors in computing systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 2287–2296. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753671

[10] N. Pears, D. G. Jackson, and P. Olivier, "Smart phone interaction with registered displays," *IEEE Pervasive Computing*, vol. 8, pp. 14–21, April 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1550400.1550527

[11] A. Olwal, "Lightsense: enabling spatially aware handheld interaction devices," in *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ser. ISMAR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 119–122. [Online]. Available: http://dx.doi.org/10.1109/ISMAR.2006.297802

[12] D. C. McCallum and P. Irani, "Arc-pad: absolute+relative cursor positioning for large displays with a mobile touchscreen," in *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, ser. UIST '09. New York, NY, USA: ACM, 2009, pp. 153–156. [Online]. Available: http://doi.acm.org/10.1145/1622176.1622205

[13] M. Nancel, J. Wagner, E. Pietriga, O. Chapuis, and W. Mackay, "Mid-air pan-and-zoom on wall-sized displays," in *Proceedings of the 2011 annual conference on Human factors in computing systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 177–186. [Online]. Available: http://doi.acm.org/10.1145/1978942.1978969

[14] X. Cao and R. Balakrishnan, "Visionwand: interaction techniques for large displays using a passive wand tracked in 3d," in *Proceedings of the 16th annual ACM symposium on User interface software and technology*, ser. UIST '03. New York, NY, USA: ACM, 2003, pp. 173–182. [Online]. Available: http://doi.acm.org/10.1145/964696.964716

[15] D. Vogel and R. Balakrishnan, "Distant freehand pointing and clicking on very large, high resolution displays," in *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ser. UIST '05. New York, NY, USA: ACM, 2005, pp. 33–42. [Online]. Available: http://doi.acm.org/10.1145/1095034.1095041

[16] M. Moehring and B. Froehlich, "Effective manipulation of virtual objects within arm's reach," in *Virtual Reality Conference (VR), 2011 IEEE*, march 2011, pp. 131 –138.

[17] L. Aguerreche, T. Duval, and A. Lécuyer, "Reconfigurable tangible devices for 3d virtual object manipulation by single or multiple users," in *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '10. New York, NY, USA: ACM, 2010, pp. 227–230. [Online]. Available: http://doi.acm.org/10.1145/1889863.1889913

[18] R. Ball, C. North, and D. A. Bowman, "Move to improve: promoting physical navigation to increase user performance with large displays," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 191–200. [Online]. Available: http://doi.acm.org/10.1145/1240624.1240656

[19] G. Cohn, D. Morris, S. N. Patel, and D. S. Tan, "Your noise is my command: sensing gestures using the body as an antenna," in *Proceedings of the 2011 annual conference on Human factors in computing systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 791–800. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979058

ANEXO VI

Artigo do SVR 2013

(MOTTA; NEDEL, 2013)

Deviceless Gestural Interaction for Public Displays

Thiago Motta, Luciana Nedel
Institute of Informatics
Federal University of Rio Grande do Sul (UFRGS)
Porto Alegre, Brazil
E-mail: {tsmotta,nedel}@inf.ufrgs.br

***Abstract*—This paper introduces a gestures based interaction method to interact with public displays that avoids the use of any device attached to the user. To accomplish interactive tasks, the user just need to position himself in front of the display and interact with the information presented on the screen using his hands. Supported gestures provide navigation, selection and manipulation of objects, as well as panning and zooming at the screen. In order to evaluate how robust the system is in a real public scenario, some criteria that could interfere on the interactive task are evaluated, as the amount of brightness in the environment, and the presence of other persons in the same interaction place. Given the results of the performed tasks, it is possible to conclude that the system, although not behaving correctly in all situations, has potential use if its difficulties are circumvented, most of which come from the inherent limitations of the Kinect, the device used.**

***Keywords*-interactive computing; interactive systems; large-screen displays; natural language interfaces**

I. Introduction

Public displays are displays that are available to the public, usually in uncontrolled environments. They can be easily found in airports, shopping malls, parks, restaurants, etc. They are used to present the following sessions in the cinema, the price of some product, the dish of the day, important news, and a lot of other information that may be useful to those who see it. These mentioned screens are static and, normally, non-interactive. However, increasingly interactive displays are been presented to the public [1], as shown by a couple of works [2][3] that describe situations in which people faced an interactive display on a public space. It is important to notice that, in these situations, the user does not have the comfort of sitting at a desk and use the conventional mouse and keyboard to interact. He needs to stand up and often move physically to be able to explore all the information.

Although there are works that explore the capabilities of an interactive public display, this is an area still poorly addressed, and consequently there are few indications of which would be the best methods of interaction to be used. In the works mentioned above, large touch-screens were used. However, other approaches can be taken, as will be seen in Section II. Among these approaches, one that draws attention is the one that can be used in any type of screen, and where the user does not need to hold any device to interact with the information displayed.

Figure 1. Snapshot of two users interacting with a public display looking for informations on a map.

In our work, a system is proposed in which the user only uses his hands to perform tasks on an interactive display (see Figure 1. We implemented case studies that explore natural gestures for selection and manipulation of virtual objects in 2-D, pan and zoom on an assorted amount of information. Aiming its applicability to public displays, several criteria that could interfere on the user interaction on uncontrolled environments were formally evaluated, such as the amount of illumination in a room and the presence of other people in the same space. Furthermore, the accuracy of the proposed technique was evaluated for different scenarios and goals.

After the analysis of the results obtained on the tests conducted, it is clear that the proposed model is good enough in certain scenarios, such as the selection and manipulation of large objects and panning and zooming the screen, but it lacks in others, such as the selection and manipulation of small objects. Based on the low number of studies on the subject, one can say that there is not a method that allows user interaction without devices more accurately. However, while the technology does not advance to create robust

devices for gesture recognition, the methodology proposed in this paper can be applied with acceptable performance and at a very low cost.

The remaining of this paper is organized as follows. Section II present related works on interaction with large displays, and gestures recognition using the Kinect, the hardware we are using for gestures capture. Section III explains our strategy for device less gestural interaction and Section IV details the design and implementation of this project. Section V details the user studies conducted and Section VI presents the results obtained. Conclusions and future works are presented in Section VII.

II. Related Work

In order to identify the criteria that must be taken into account, in this work, we covered the state of the art on interaction with large displays. Moreover, as the goal is to build a model that employs gesture recognition without any device manipulation by the user, it is also important to examine studies that employ the Microsoft Kinect, since it is currently the device with the better cost-benefit relation [4].

A. Interaction on large displays

Touchscreens have been studied for a long time. However, with the advent of the multi-touch sensitive screens, its use has been increasingly common. Two studies were conducted on the same touchscreen system at the Helsinki Institute for Information Technology [2][3]. The architecture of the system proposed involves a semitransparent back-projective screen and a camera sensitive to infrared emissions, positioned next to the projector. Touches are detected emitting infrared light on the screen. According to the authors, the system provides the capture of as many taps as possible to be made by the dimension of the screen.

A common technique used to perform the interaction with large displays is the use of mobile devices as interactive tools. The *Touch Projector* [5] uses a smartphone to record a big screen, identify it and receive its content. The work of Pears et al. [6] is very similar, using the camera of a smartphone to locate an object to be manipulated on a large screen and using the mobile device as a 3D mouse, being able to provide 4 DOF interaction.

The most employed approach is to create new devices or to adapt an existing device, such as the Nintendo Wii or a 3D mouse, especially in cases where a system whose cost is not prohibitive is needed. The VisionWand [7] is an example. It consists of a wand with colored tips that are read by a couple of cameras and provides the user the selection and manipulation of virtual objects in 5 DOF interaction. Differently, the LOP-cursor [8] uses the accelerometers embedded on a smartphone to interact with objects in a display wall very precisely.

The interaction through data gloves is also a common practice. Vogel and Balakrishnan [9] present a data glove with passive reflector points in the fingertips and a Vicon Motion Tracking to identify these points with which you can manipulate virtual objects in front of the screen. Mohering and Froehlich [10] use optical mechanisms for capturing the position of the fingers and hands, in a system that, according to the authors, provides high accuracy, and has also haptic feedback.

B. Interacting with the Kinect

Although we can easily find in the Web many applications using the Microsoft Kinect to interact, there are few scientific research papers on the subject. An interesting research use 3 Kinects to generate 3-D models of the users [4]. Despite the good results, the authors emphasize that the low resolution of Kinect cameras prevent the generation of more sophisticated models, although, given the low cost for setting up the system (approximately US$ 600.00), the results are quite satisfactory. Another interesting work using Kinect to recognize humanoid [11] is able to detect the presence of humans with an accuracy of 98.4%.

Turning to an interactive application of the Kinect, the works of Bidgelou et al. [12] and Gallo et al. [13] can be mentioned. They use the device to allow interaction with medical data. However, both jobs require an initial calibration by the user.

All works above uses the Kinect's depth camera to recognize patterns, showing that this is a promising approach. Although some works [12][13] are interesting and present good interactive techniques, training and calibration steps are inconvenient, especially when it comes to public displays, where the user will not be willing to interact if the system is not very simple. Based on that and in what was presented in the first half of this section, a new model should be proposed.

III. Deviceless Gestural Interaction

This paper presents an approach for gestural interaction without devices, i.e., without any device attached to the user body or in his hands. This approach is desirable for interaction with public displays, where the user interact standing in front of a large screen, does not want to share any devices, and also does not want any complicated mechanism to perform the interaction. We believe that in an ideal situation, the user faces the display and automatically discover what he need to do in order to interact with it. Our approach seeks such panorama.

In the proposed system, the user interacts with his hands (either one or both) positioned in front of the body to perform translations, zoom, selection and manipulation of elements on the screen, all of this in a 2-D environment. In order to differ between everyday gestures and gestures to be recognized by the system, the user must close his hands to interact. The system does not need any calibration and automatically identifies when a user is in front of the screen to interact. It always focuses on the user closest to

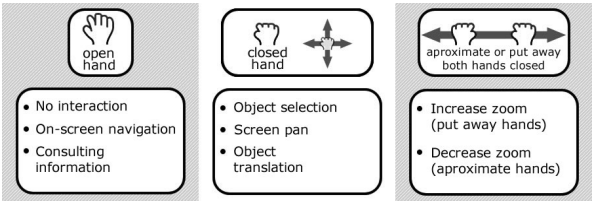

Figure 2. Gestures supported by the proposed model: at left, with both hands opened, at the center with one hand closed, and at right with both hands closed.

the display in an area in which he can easily see its contents, i.e., without being immediately in front of the screen.

With both hands opened, the user can browse the information on the screen without changing them, seeing information that is hidden (e.g. in tooltips). Closing one hand, the user can select and manipulate the information on the screen. Closing both, it is possible to zooming. Figure 2 illustrates this graphically.

These gestures were chosen in order to trace a precise parallel with daily tasks. For example, when you query a particular entry in a list, it is common to swipe the items on the list until stop in to the desired information. When choosing a product in the supermarket, the user extends his arm and closes his hand on the object choosen, making a selection between products. When a user moves an element on a table – e.g. the mouse itself – he closes his hand over it and moves his wrist to the position where he wants to drop it.

The gestures for pan and zoom were based on techniques already established for these activities on touch devices. To zoom into a picture, we normally use two fingers on the screen, whose positions deviate to zoom in and get closer to zoom out. When a picture has undergone a zooming and does not fit completely on the screen, the user uses a finger whose position changes according to where he wants to put it. Therefore, the proposed gestures seem quite intuitive and adequate to mimic the gestures that the user use to perform, reducing the training time required for the proposed technique.

In order to evaluate the proposed model, two applications were developed, so that it is possible to examine in detail each of the functionalities provided. Each one will be described in greater detail below.

A. *Selection and manipulation of simple objects*

This application was developed for conducting experiments with users. It presents a series of small tasks that must be accomplished by the users, comprising especially selection and manipulation of objects. Initially, six squares are displayed on the screen, one being green, indicating that it should be selected by the user. Upon selection, another of the six square becomes green and, thus, will be the next that the user will have to select. This procedure is repeated

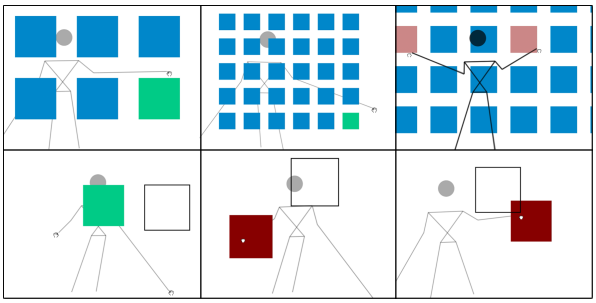

Figure 3. Screenshots of the application to select and position simple objects.

five times, and then the size of the square decreases by half, which enables them to appear in greater numbers. After five new selections, the squares decrease in size again and, after five more selections, a third time.

At the end of these last five selections, a green square of the original size (large) appears on the screen and as well as a hollow square in black, slightly larger than the green one. From then on, the user must not only select the green square, but also position it so that it is within the hollow black square. The size of these squares also decreases after five positionings, but only twice, not three as in the selection task. After the last positioning, a message is displayed stating that the task finished.

The user can pan and zoom on the screen at any time by just closing his hands on any screen space that does not contain the green square. The user skeleton is displayed in semi-transparency, and also icons indicating the position and status of the user's hands. In Figure 3 is possible to see that the skeleton of the user is drawn behind the squares, while the icons of hands are always on top. Figure 3 shows execution steps of this application: at the top, the tasks of selection – from left to right: the home screen of the application, the decreased size of the square after the first five selections and zooming in screen; below, the positioning tasks – from left to right: the initial screen of this task, the user manipulating a square with his left hand, and finally, doing the same with his right hand.

This application supports gestures for selection, manipulation, panning and zooming on the screen. The gesture for navigation is also present, but the only information that it displays is a visual feedback about the square which is below each hand. The application was developed so that the level of difficulty of the tasks increases as users gets practice with the system, which was confirmed after the application of tests with users.

B. *Localization map of a building*

This application was built to analyze how people react to a public interactive display. It shows the map of the building of the Institute of Informatics at UFRGS where professor's

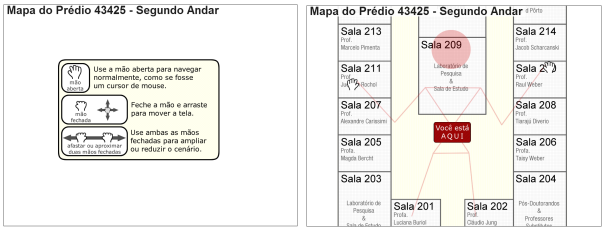

Figure 4. Two screenshots of the application that displays the map of the building.

offices are located as well as some research laboratories. The map shows all the rooms and corridors of the building, with indications of room numbers and names of professors that own the room, if that is the case. It has also an indicative of the location where the public display is on the building, serving to the user as a locator.

The map is only loaded when the presence of a user in front of the display is detected. While a user is not detected, a window with instructions about the system is displayed (Figure 4-left). When a user is detected, the system continues displaying the instructions for 9 seconds, but also starts to display the map of the building and the semi-transparent skeleton of the user behind that window. After 9 seconds, the window disappears and the user is free to interact with the application (Figure 4-right).

This application also shows the user skeleton so that he can sense his presence in the system, and also the same hand icons already used in the previous application. The application recognizes gestures to pan and zoom on the map.

IV. Design and Implementation

We wanted to build an interaction model that could be easily implanted anywhere. In order to achieve this, the system would need to have low financial cost and be as portable as possible. Moreover, as already observed, it was desirable that the interaction did not require any manipulation or coupling of devices, giving users freedom to move as he wanted and without the need to share objects. With that in mind, Microsoft Kinect has proven the best alternative because it has a very low cost and with the help of an SDK, provides useful data in a quite simple way. As seen in related work, the device has enough potential to provide deviceless interaction, provided that its limitations are circumvented.

Having the hardware necessary for the interpretation of gestures, a definition of the user interface was needed, which would have to be developed according to the criteria of portability desired. With the evolution of Internet applications features, especially after the arrival of the HTML5 standard, a Web approach proved to be interesting. It is well established – especially with the increase of applications "on the cloud" – that Web systems are easily portable, therefore, the choice of this approach seemed appropriate. Thus, the proposed model was developed in a Web environment and using the Kinect as interactive device.

A. Capturing Kinect data

The Kinect interpretation is through the Microsoft's *Kinect For Windows* SDK, which has a number of advantages over its competitors, especially not requiring a user calibration pose to recognize its skeleton, as the OpenNI SDK. For this work, the depth information and user's skeleton were used.

The first step is to detect if there is a user to interact. That is, when the SDK informs that the skeleton data are ready. Then, with the joints information from the skeleton it is discovered the position of the hands of the user, using the 2D points corresponding to the joints *HandRight* and *HandLeft*. Finally, it identifies the status of the user's hands.

As currently no SDK can recognize natively if the user's hand is open or closed, a post-processing of the Kinect information need to be done to get this information. For the system to work with any user without a step of calibration, image processing algorithms are employed on the obtained depth image. To identify the status of each hand the system takes three steps, better described below.

1) Isolating the hand: The first step is to locate the user's hand. To this, we use two joints of the skeleton obtained by the Kinect: wrist and hand. The coordinates of the skeleton joints are mapped to coordinates on the depth image, obtaining the points W(x, y) of the wrist, and H(x, y), of the center of hand. Calculating the distance d between these points it is possible to set up a square of side $2d$ centered at point H, which encompasses the entire region occupied by the hand on the depth image.

From this square image obtained, the depth information P, in millimeters, of the point H is used to be compared with all other image pixels. Pixels are colored in white when its depth information is inside the limit [P(H)-30mm, P(H)+70mm]. Otherwise in black. At the end of this process, it is obtained an image of the user's hand in white over a black background.

2) Detecting the contour of the hand: The next step is to detect the contour of the hand isolated. This procedure is done in two specific steps: first, it is applied a high-pass filter to the image that lefts only the outline of the hand; after the image goes trough a new processing to detect contour pixels that are adjacent to each other.

To detect the adjacency between contour points, first is defined an array A to store them. Then, starting from any point, a comparison is made between the neighboring points within a 3x3 mask. If any of the neighboring pixels is white and is not yet in A, the pixel is added to the array and the mask is centered in that pixel, repeating the process. If none of the neighbors of the pixel is a white dot that is not yet in A, the mask is expanded to 5x5 comparisons. If even so is not found a neighbor that fits the test, the mask is

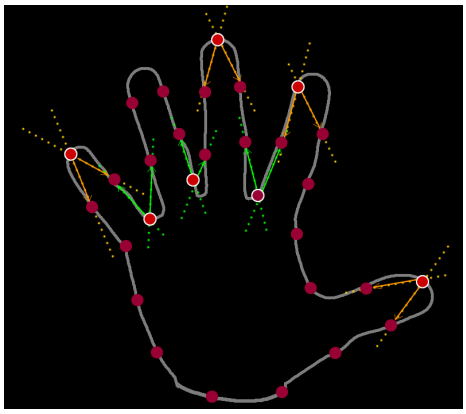

Figure 5. Running the algorithm K-curvature [14] on the outline of a hand, with the points of discontinuity detected marked by white circles.

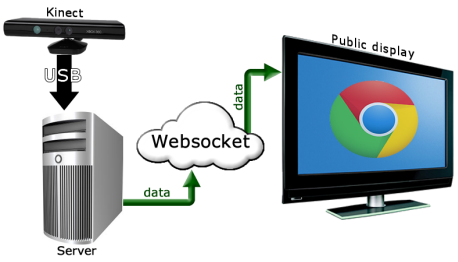

Figure 6. The architecture of the proposed system, describing the communications involved in the process of transposing the data interpreted from the Kinect to the Web application in a public display.

enlarged again to 7x7. If the test fails again, it is concluded that the entire contour has been traversed and the algorithm terminates. At the end of this step, the array *A* contains, in order, the contour points of the image.

3) Discovering the state of the hand: Finally, the last step to discover if the hand is open or closed deals is to search for the fingertips, which represent discontinuities in the contour of the image. To achieve this, we used the K-curvature algorithm [14], which runs on a contour. To implement the algorithm is necessary to define two constants: *pointsInterval*, which defines the range in which the contour points will be read; and *limitAngleToBeFingertip*, which defines the angle between the vectors that represent a discontinuity in the contour. In the proposed model, these values were set to 6 and 50, respectively, after a series of tests with different values.

When running, the algorithm takes as input the *A* array containing the points of the contour. The algorithm starts processing at the position *pointsInterval* of the array and continues at intervals until reaching the end of the array minus *pointsInterval* positions. For each position *i* being analyzed, the positions *i-pointsInterval* and *i+pointsInterval* are read. Then the angle formed between the vectors (*i-pointsInterval*, *i*) and (*i*, *i+pointsInterval*) is calculated. If this angle is less than *limitAngleToBeFingertip*, the position *i* corresponds to a point of discontinuity.

If at the end of process there has been obtained any discontinuity point (a fingertip or a point between two fingers) it is considered that the hand is open. Figure 5 shows an implementation of the algorithm on an image. Red dots circled in white are the points of discontinuity detected, being formed by the orange vectors (for fingertips) or green ones (for regions between two fingers). It can be observed that not every point of discontinuity were detected, but detecting only one of them is enough for this work.

As fast movements of the hands can cause the system to misinterprets their states (e.g. an open hand can be recognized as closed as moving it too quickly, it becomes a "blur" in the camera), a small buffer of 10 positions was built to store the states of the hand as in a queue, which is consulted and transmits the state that is in greater number between its positions. This causes a slight delay in interpretation, but compensates with a greater stability.

B. Integrating the Kinect to the browser

For this integration be possible, it must be used a client-server architecture, where the application that reads and interprets data from the Kinect acts as a server and communicates via message exchange with the client application, developed for Web platform. The exchange of messages is made using a WebSocket. The application that interprets the Kinect should open a WebSocket and wait for connections. In turn, the Web application should connect to the opened WebSocket and inform that it is ready to receive messages. After that, the server application must send messages that contain useful data that were obtained from data processing of the Kinect. It is transmitted basically an array containing the following information: location of the joints corresponding to the head, both shoulders, elbows, wrists, legs and hands, two joints corresponding to the waist and, in particular, two integers that indicate if the hands are open (0) or closed (1). Received these data, the Web application can interpret them and use them for different purposes, using only the data that it needs. Figure 6 illustrates the communications involved in the developed system.

C. Interpreting the data on the Web

The Web applications used in this study were constructed using languages HTML5, JavaScript and PHP. The entire visual part was implemented in HTML5, using CSS for graphic details. The interactivity of the pages takes place by means of JavaScript language, using the library jQuery, which facilitates the manipulation of screen elements.

To interpret the data received via WebSocket's message, the applications have a state machine that is updated according to the state information of each hand. The possible states are *idle, selection with right hand, drag with right*

Figure 7. Place of the application of user testing. It is possible to see the location where the user should position himself.

hand, hold with right hand, selection with left hand, drag with left hand, hold with left hand, and zoom.

The hand position is updated regardless of the states, and is indicated by icons that show open and closed hands. Additionally, a skeleton formed by line segments is constructed on based on the joints information received via message, as can be seen in Figure 3. These updates are made every incoming message, as well as updating the state machine.

Selections and the drags/translations are determined according to the position of the hand. If there is an element in the position of the hand, this element will be affected, otherwise will be the whole canvas, which is built with the use of an HTML5 element *¡canvas¿*. The action release is used to release the element or screen for positioning action. The zoom is performed based on the position of both hands, focusing on mid-point between them: when it starts, it is stored the distance between the points that represent the positions of the hands, d_0; then, in later updates, the distance between these points, d_i, is recalculated. If d_i is greater than d_0, it is considered that an expansion was performed, i.e. an increase of zoom. Otherwise, i.e. if d_i is smaller than d_0 is considered that a reduction was made, namely a decrease in zoom.

V. User evaluation

Once the system was developed and working, the two case study applications were evaluated according to the relevant criteria to its installation in a public display, namely: ambient lighting, presence of other people, type of location, task type performed and presentation of information. Each will be explained in more detail in separate subsections below.

The application that shows the map of the building was installed at the entrance of the building to which it's map refers. There it was monitored for a period of 12 hours spread over two days in order to observe people's reactions before a public interactive display. The application of selection and manipulation of simple objects was used for formal evaluation with users. To this, 38 users were asked to perform the tests proposed and the time spended to execute the tasks and the number of errors were recorded, as well as the use of pan and zoom. All testers performed the tasks at least twice, the first one being for they become accustomed to the system. The users also answered two questionnaires: the first before carrying out the tests in order to characterize them and the second with their opinions regarding the tasks, applied after testing, for subjective evaluation purposes.

Users who performed the tests are divided among 30 men and 8 women, have an average age of 23.82 years and a median of 24, all accustomed to the daily use of a computer. The experiment was conducted in a room with artificial lighting by fluorescent lamps with two variations of brightness, as well as with and without the presence of other people. Three users had to be discarded due to errors in the execution of the application, totaling 35 users with useful data. There was a mark on the floor indicating the optimum position where the user should position himself, which was ample in order to allow horizontal movement. It was produced a video explaining about the tasks and gestures that was shown to all the testers after they answer the first questionnaire. With the explanation of the test done on video, all users received exactly the same instructions, not being induced by something that the conductor has passed individually. Figure 7 shows the environment of the tests.

The tests had as independent variables, i.e. which does not depend on the user, the job type (selecting/positioning) and the size of the squares. The dependent variables, which change according to the user, were the time of each task and the number of errors occurred. It was considered as *error* a selection of other than the green square on the task of selecting and the dropping of the square to be positioned in a location other than the correct destination on the positioning task.

VI. Results

Although it has its problems, the proposed model in this study has great application potential when used for simple interactive tasks. In formal evaluation with users, testers responded to several questions about the tasks and their responses indicate that the proposed system had good performance in certain situations.

In a field for general comments on the second questionnaire, many users noted that the system is tiresome on certain situations, which may have been generated by the difficulty in positioning the smaller squares when the target position was too far away. Another interesting feature that was observed by users is that the practice of the interactive gestures greatly improves user performance. Quoting a comment: "with a short time of use/interaction, the domain of the application functionality is already reasonable". Thanks to the implementation of the entire task only as training at

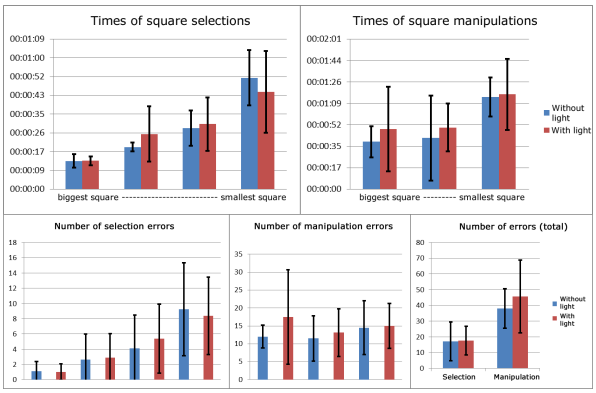

Figure 8. Graphics showing the differences in the data obtained in tests with the illuminated by lamps or not: above in relation to the average execution time of each task and below in relation to the average number of errors occurring in each task. Regarding the size of squares, the arrangement is always from the biggest to the smallest. The illuminated environment is represented by the blue bars and without illumination by red. The black line on each bar marks the standard deviation.

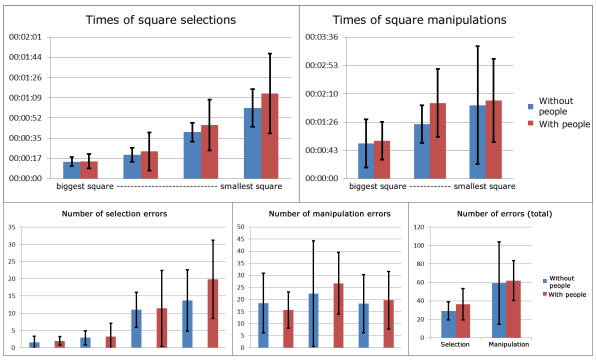

Figure 9. Graphics showing the differences in the data obtained in tests with and without the presence of other persons: above in relation to the average execution time of each task by users and below in relation to the average number of errors occurred in each task. Regarding the size of squares, the arrangement is always from the biggest to the smallest. The interaction without interference is represented by the bars in blue and with the presence of people the ones in red. The black line on each bar marks the standard deviation.

first, the user already performs the tests with greater speed and accuracy than did in training session.

Below, in specific subsections, the results of all criteria evaluated in this work for a gestural deviceless interaction for a public display are described.

A. *Illumination conditions*

In order to evaluate if the difference in illumination was determinant, 8 users were selected, without any specific criteria, for testing the application twice (in addition to the first, the training one) with two light variations measured by a luximeter: with normal lighting, of 731 *lux*, and with no lighting other than the display, of 221 *lux*. To not be a determining factor in the assessment, users performed tests alternating between the two light intensities.

As expected, the lighting was not a relevant factor in interactive tasks. According to an analysis of variance (ANOVA), selecting squares did not had its time determined by the amount of illumination, obtaining p-values 0.9256, 0.2822, 0.7816 and 0.6303 for square sizes ranging in sizes of four major the lowest, in order. The positioning also had the same response, obtaining p-values 0.4394, 0.3514 and 0.8991, ranging in sizes of squares large, medium and small, respectively. The number of errors in each task also was not significant, with p-values 0.9274 and 0.4267 for selection and positioning errors respectively. Thus, according to ANOVA, the illumination should not be a relevant factor when applying the proposed system in a public display, as can be seen in the graph of Figure 8.

B. *Presence of other persons in the same place*

To evaluate whether the presence of other people in the environment is an important factor to interact in a public display, 10 people, other than those who evaluated the differences in lighting, were randomly selected to complete the tasks twice: once with the presence of people passing behind and ahead of them, bowing at his side and trying to simultaneously interact with the display, and a second without any interference. Again, avoiding the interference in the tests, the order of tasks was alternated between users.

Contrary to what was anticipated, according to an analysis of variance of the execution times of the tasks and the number of errors, it was not possible to conclude that the presence of other people in the same environment that the interacting user impact on the completeness of the tasks. The analysis of the selection of squares had p-values 0.8566, 0.6293, 0.5106 and 0.4404 with square sizes varying in descending order. In turn, the positioning of the squares analysis had p-values 0.7939, 0.3363 and 0.8034 from square in three sizes in descending order too. The analysis of variance of the number of errors produced p-values 0.2637 and 0.8600 in relation to the selection and manipulation respectively.

Not confirming the hypothesis is a very positive point for the proposed system, indicating that it is able to identify the person who is interacting and remain consistent throughout the interactive task, recovering quickly from errors, even though the average time of completion of tasks is greater when there were other people. Figure 9 presents graphics showing how close are the times and number of errors in the various tasks with and without the presence of other people in the environment. The analysis of the limits imposed by the lines of standard deviation indicates that both environment settings are similar.

C. *Sort of place*

The place where the interaction in a public display take place can influence the execution of tasks due to lightness

influence or due the presence of another people in the place. These specific criteria were evaluated and the results presented above. However, some psychological factors can also influence the outcome and these factors can not be measured quantitatively. Aiming to evaluate the system in a real environment of use, observations were made when the system was installed at the entrance of a building with great traffic of people and it was possible to perceive the existence of certain behavior patterns.

On the first day in particular, it was possible to see that people demonstrates interest passing by a public interactive display, looking quizzically at the screen when their skeleton appears on it. However, despite the interested, people seem to be afraid to try out the system, deciding to follow his path after a some hesitation.

This pattern differs when people are in a group, in which case people seems to feel encouraged to interact with the system as if to appears bold. One of the elements of a group always takes the lead and begins to interact with the screen, then the other members approximate and try to do the same, as if to disturb his companion, but usually without success, since the system focuses only on one user. After a few moments of exploration, the group moves on, commenting about their experience. Figure 10 shows the display in its initial state where it was placed and two situations in which groups interacted with the system.

The fear in interacting with the system also disappears when the user thinks he is alone in the environment. This became clear when observing a user who would often come to take papers from the printer next to the screen and always watched the display, but as there were always other people going by, he did not dare to interact. However, in one of his visits to the printer, he looked around and saw no one and then tried to use the system for a few minutes, but soon returned to his business (especially because as he carried papers in one hand, the system did not recognize if it was open or closed and did not behave properly). The observer was always in a location far ahead of the display, about 6 feet, but the system holds the users' attention with such intensity that the observer was not noticed.

The system behaved identically both in the closed room and in the lobby of the building, although one can not say the same of those who interacted with it, especially those who did it alone. There seems to be something that restricts people when they must perform unusual gestures in public. In general, the system seemed to be well accepted by users who dared to use it and succeeded. One user commented that "to interact correctly is just a matter of getting used to the gestures".

D. Sort of task

As explained in Section III, the developed applications explore activities of navigation, selection and manipulation of objects and pan and zoom on the screen. The application

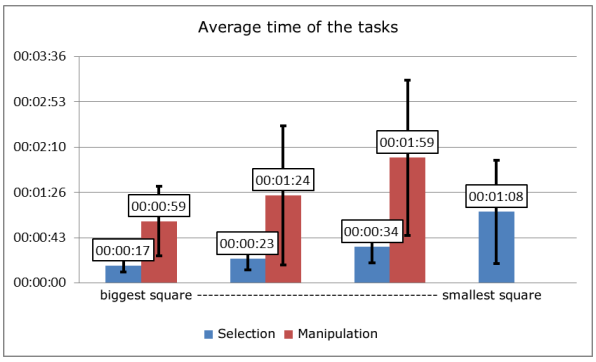

Figure 11. Graphic comparing the average time of completion of each task of selection (blue) and manipulation (red). The black lines indicate the standard deviation range.

of selection and manipulation of squares was especially developed to evaluate the type of tasks, varying in difficulty and type and allowing the user to always make translation and zoom on the screen, trying to present the tasks with a level of increasing difficulty. For this evaluation, we compared the results of every 35 users with maximum illumination and without the presence of people.

As assumed, after an analysis of variance, it become pretty clear that the selection task is easier to perform than the manipulation. The three sizes of squares that are common to both tasks resulted in the analysis p-values $2.4523*10^{-10}$, $1.0989*10^{-6}$ and $10*6.4350^{-9}$ with sizes in descending order. Thus, with a very high probability, the selection task should be easier than the positioning, as shown by the graphic of Figure 11.

On further analysis, it is concluded that even the selection of the second largest square should be easier than positioning the larger square (the easiest to position), resulting a p-value $4.9689*10^{-8}$ in an analysis of variance. The same is true when comparing the selection of the second smaller square with the position of the largest square, in which case results a p-value 0.00019, a number quite higher than the others, but still statistically significant.

However, no statistically significant correlation was found between the task of selecting the smallest square and of manipulating the larger square, in which case ANOVA results p-value 0.3794. This scenario does not occur when analyzing the number of errors when the p-value obtained is 0.0013, indicating that the task of selecting the lowest square produces less errors that positioning the larger square.

Regarding gestures to translate the screen and change the zoom level, all users used them at least once. The translation was performed frequently by mistake, which disturbed the interactive task especially in positioning. Both the translation and the zoom were performed on both types of tasks. Many users used the translation to bring the square to be selected to a position above their shoulders, an area where the interpretation of gestures was more accurate. Furthermore,

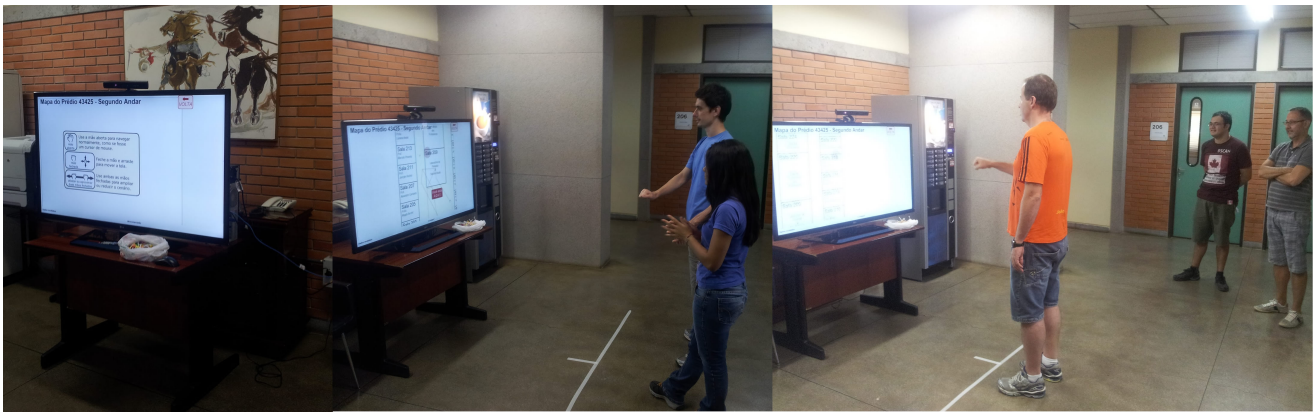

Figure 10. At left the system as installed in the entrance hall of the building whose map is displayed in the application. At right two pictures showing groups interacting with the display.

many testers used the zoom when the squares were presented in its smaller size, both in selection or positioning tasks.

E. Information presentation

In the application of selection and manipulation, the squares have its size decreased during the execution of tasks, and are presented in greater number in the screen, both in the selection and manipulation tasks. Using this, we evaluate whether the size and number of objects interfere in time to accomplish the tasks, comparing the results of all 35 users.

Concerning the size and quantity of the square, the hypothesis that larger squares are easier to select and position was confirmed. Considering the hypothetical square sizes 8, 4, 2 and 1 for the selection, the squares of size 8 should be easier to select than those of size 4, according to an analysis of variance with p-value 0.0025. Following the rule, a square of size 4 should be easier to select than one with size 2, with a p-value 0.0004, and of size 2 relative to size 1 with a p-value 0.0002.

The results repeats in the positioning task. Considering the hypothetical sizes of squares 8, 4 and 2, the squares of size 8 should be easier to position that those of size 4, according to an analysis of variance with p-value 0.0529, and those of size 2 with p-value $3.8358*10^{-5}$. In turn, squares of size 4 should be easier to position than squares of size 2, according to an analysis of variance with p-value 0.0369.

VII. Conclusion and Future Work

Was presented a study on a gestural interactive system for public displays that does not require the user to hold any device. In the proposed system, the user stands in front of a public display and use his hands to interact with the information presented on the screen, using for that gestures similar to the ones used in everyday situations, as close the hand to select an object and put away two points of contact to enlarge the screen.

Based on the evaluations, it is possible to say that the system, despite having deficiencies in certain aspects, behaved well enough to interact with large objects in selection and manipulation at short distance tasks. It is also clear that systems employing features of pan and zoom can take advantage of the interactive method proposed, since users in a real pubic place were able to interact with the display in an application of this kind.

Although it does not provide a definitive solution to the interaction in public displays, the work serves to indicate the main difficulties when seeking a solution to this interactive problem. The proposal introduced by the system, using the closing hand to differentiate the selection from navigation seems pretty intuitive for users, as well as the use of both hands to modify the intensity of zoom on the screen.

An interesting future work to be done is to integrate the system introduced in this paper with another interactive method, like those provided by mobile devices. In this scenario, the user would see and interact with information in a public display, but the application would allow him to use his own mobile phone to insert information or perform delicate interactions or also receive a copy of the data. In that way the user does not interact directly with any device that is not his own, avoiding any problems of sharing or learning curves.

Finally, it should be noted again that the solution proposed in this work to provide a free gestural interaction for public displays is interesting because it makes independent the applications of gesture interpretation and of information presentation. Thus, any developer could take advantage of the information obtained by the application that interprets Kinect and build his own Web application that does what he wants, provided it connects to a WebSocket and use the state machine to decipher the information received, thus obtaining the user's intent. And all this with a low financial cost, which comprises the values needed for the purchase of a Microsoft

Kinect, a large display and a computer.

ACKNOWLEDGMENT

We would like to thank all the users who kindly tested the system. This work was partially supported by Microsoft Brazil Interop Lab at UFRGS, CNPq-Brazil through projects 311547/2011-7 and 485820/2012-9, and CPD-UFRGS that supports Thiago Motta.

REFERENCES

[1] U. Hinrichs, S. Carpendale, N. Valkanova, K. Kuikkaniemi, G. Jacucci, and A. V. Moere, "Interactive public displays," *IEEE Computer Graphics and Applications*, vol. 33, no. 2, pp. 25–27, 2013.

[2] P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, and P. Saarikko, "It's mine, don't touch!: interactions at a large multi-touch display in a city centre," in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 1285–1294.

[3] G. Jacucci, A. Morrison, G. T. Richard, J. Kleimola, P. Peltonen, L. Parisi, and T. Laitinen, "Worlds of information: designing for engagement at a public multi-touch display," in *Proceedings of the 28th international conference on Human factors in computing systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 2267–2276.

[4] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, pp. 643–650, 2012.

[5] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch, "Touch projector: mobile interaction through video," in *Proceedings of the 28th international conference on Human factors in computing systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 2287–2296.

[6] N. Pears, D. G. Jackson, and P. Olivier, "Smart phone interaction with registered displays," *IEEE Pervasive Computing*, vol. 8, pp. 14–21, April 2009.

[7] X. Cao and R. Balakrishnan, "Visionwand: interaction techniques for large displays using a passive wand tracked in 3d," in *Proceedings of the 16th annual ACM symposium on User interface software and technology*, ser. UIST '03. New York, NY, USA: ACM, 2003, pp. 173–182.

[8] H. Debarba, L. Nedel, and A. Maciel, "Lop-cursor: Fast and precise interaction with tiled displays using one hand and levels of precision," in *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, 2012, pp. 125–132.

[9] D. Vogel and R. Balakrishnan, "Distant freehand pointing and clicking on very large, high resolution displays," in *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ser. UIST '05. New York, NY, USA: ACM, 2005, pp. 33–42.

[10] M. Moehring and B. Froehlich, "Effective manipulation of virtual objects within arm's reach," in *Virtual Reality Conference (VR), 2011 IEEE*, march 2011, pp. 131 –138.

[11] L. Xia, C.-C. Chen, and J. Aggarwal, "Human detection using depth information by kinect," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, june 2011, pp. 15 –22.

[12] A. Bigdelou, T. Benz, L. Schwarz, and N. Navab, "Simultaneous categorical and spatio-temporal 3d gestures using kinect," in *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, march 2012, pp. 53 –60.

[13] L. Gallo, A. Placitelli, and M. Ciampi, "Controller-free exploration of medical image data: Experiencing the kinect," in *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, june 2011, pp. 1 –6.

[14] N. Shaker and M. Abou Zliekha, "Real-time finger tracking for interaction," in *Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on*, sept. 2007, pp. 141 –145.