

**Superfície Mágica:  
criando superfícies interativas  
por meio de câmeras RGBD e projetores**

Alexandre Martins Ferreira de Sousa

DISSERTAÇÃO DE MESTRADO  
TEXTO SUBMETIDO PARA O  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
DA  
UNIVERSIDADE DE SÃO PAULO

Programa: Mestrado em Ciência da Computação

Orientador: Prof. Dr. Carlos Hitoshi Morimoto

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da CAPES

São Paulo, dezembro de 2015

**Superfície Mágica:  
criando superfícies interativas  
por meio de câmeras RGBD e projetores**

Esta versão da dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 23/06/2015. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. Carlos Hitoshi Morimoto (orientador) - IME-USP
- Prof. Dr. Flávio Soares Corrêa da Silva - IME-USP
- Prof<sup>a</sup>. Dr<sup>a</sup>. Veronica Teichrieb - UFPE

# Resumo

de Sousa, Alexandre M. F. **Superfície Mágica: criando superfícies interativas por meio de câmeras RGBD e projetores.** 2015. 73 f. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2015.

Em computação ubíqua, existe a ideia de tornar o computador onipresente, “invisível”, de modo a aproximar computadores e humanos. Com o avanço das tecnologias de hardware e de software, torna-se interessante investigar possibilidades inovadoras de interação com os computadores. Neste trabalho, exploramos novas formas de interação inspiradas nos atos de desenhar, agarrar e gesticular. Para testá-las, desenvolvemos novos algoritmos baseados em câmeras RGBD para detecção, classificação e rastreamento de objetos, o que permite a concepção de uma instalação interativa que utilize equipamentos portáteis e de baixo custo.

Para avaliar as formas de interação propostas, desenvolvemos a Superfície Mágica, um sistema que transforma uma superfície comum (como uma parede ou uma mesa) num espaço interativo multi-toque. A Superfície Mágica identifica toques de dedos de mãos, de canetas coloridas e de um apagador, oferecendo também suporte a uma varinha mágica para interação 3D. A Superfície Mágica suporta a execução de aplicativos, permitindo que uma superfície comum se transforme numa área interativa para desenho, num explorador de mapas, num simulador 3D para navegação em ambientes virtuais, entre outras possibilidades. As áreas de aplicação do sistema vão desde a educação até a arte interativa e o entretenimento. A instalação do protótipo envolve: um sensor Microsoft Kinect, um projetor de vídeo e um computador pessoal.

**Palavras-chave:** computação ubíqua, superfícies interativas, interfaces naturais, interatividade, design exploratório, Kinect.



# Abstract

de Sousa, Alexandre M. F. **Magic Surface: creating interactive surfaces using RGBD cameras and projectors.** 2015. 73 p. Master's Thesis - Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2015.

Ubiquitous computing is a concept where computing is thought to be omnipresent, effectively “invisible”, so that humans and computers are brought together in a seamless way. The progress of hardware and software technologies make it compelling to investigate innovative possibilities of interaction with computers. In this work, we explore novel ways of interaction that are inspired by the acts of drawing, grasping and gesturing. In order to test them, we have developed new RGBD camera-based algorithms for object detection, classification and tracking. This allows the conception of an interactive installation that uses portable and low cost equipment.

In order to evaluate the proposed ways of interaction, we have developed the Magic Surface, a system that transforms a regular surface (such as a wall or a tabletop) into a multitouch interactive space. The Magic Surface detects touch of hand fingers, colored pens and eraser. It also supports the usage of a magic wand for 3D interaction. The Magic Surface can run applications, allowing the transformation of a regular surface into an interactive drawing area, a map explorer, a 3D simulator for navigation in virtual environments, among other possibilities. Areas of application range from education to interactive art and entertainment. The setup of our prototype includes: a Microsoft Kinect sensor, a video projector and a personal computer.

**Keywords:** ubiquitous computing, interactive surfaces, natural user interfaces, interactivity, exploratory design, Kinect.



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	O projeto . . . . .	2
1.3	Contexto . . . . .	4
1.4	Contribuições . . . . .	4
1.5	Organização do Trabalho . . . . .	4
<b>2</b>	<b>Fundamentos</b>	<b>5</b>
2.1	Interação . . . . .	5
2.2	Interfaces naturais . . . . .	6
2.3	Interfaces tangíveis . . . . .	7
2.4	Interfaces baseadas em gestos . . . . .	8
2.5	Interfaces 3D . . . . .	9
2.6	Telas sensíveis ao toque . . . . .	9
2.7	Visão computacional e processamento de imagens . . . . .	11
2.8	Superfícies interativas . . . . .	12
<b>3</b>	<b>Trabalhos relacionados</b>	<b>15</b>
3.1	Trabalhos acadêmicos . . . . .	15
3.1.1	Quadro-branco multi-toque de baixo custo usando o Wiimote . . . . .	15
3.1.2	Usando o Kinect para emular um quadro-branco interativo . . . . .	16
3.1.3	WorldKit: criação rápida e fácil de aplicações interativas ad-hoc em superfícies do dia-a-dia . . . . .	17
3.1.4	The Office of the Future: uma abordagem unificada para modelagem baseada em imagens e para displays espacialmente imersivos . . . . .	18
3.1.5	The Everywhere Displays Projector: um dispositivo para criar interfaces gráficas ubíquas . . . . .	18
3.2	Trabalhos comerciais . . . . .	20
3.2.1	Ubi Interactive . . . . .	20
<b>4</b>	<b>Ferramentas de desenvolvimento</b>	<b>21</b>
4.1	Visão geral . . . . .	21
4.2	Microsoft Kinect . . . . .	23
4.3	OpenNI . . . . .	24
4.4	OpenCV . . . . .	25

4.5 openFrameworks . . . . .	25
4.6 HTML5 . . . . .	25
<b>5 Desenvolvimento do sistema . . . . .</b>	<b>27</b>
5.1 Visão geral . . . . .	27
5.2 Cenários de uso . . . . .	28
5.2.1 Explorando a superfície de desenho . . . . .	28
5.2.2 Explorando mapas . . . . .	29
5.2.3 Navegação em 3D . . . . .	30
5.3 Superfície multi-toque . . . . .	32
5.3.1 Visão geral . . . . .	32
5.3.2 Detecção de toques . . . . .	32
5.3.3 Extração de cor . . . . .	33
5.3.4 Classificação . . . . .	34
5.3.5 Rastreamento . . . . .	35
5.3.6 Apagador . . . . .	36
5.3.7 Instalação do protótipo . . . . .	37
5.4 Varinha mágica . . . . .	39
5.4.1 Visão geral . . . . .	39
5.4.2 Design . . . . .	40
5.4.3 Implementação . . . . .	41
5.4.4 Gestos . . . . .	44
5.4.5 Instalação do protótipo . . . . .	46
<b>6 Explorando formas de interação . . . . .</b>	<b>47</b>
6.1 Interação 2D usando canetas e apagador . . . . .	47
6.1.1 Lousa Mágica . . . . .	47
6.1.2 Mapa Mágico . . . . .	49
6.2 Interação 2D usando o próprio corpo . . . . .	50
6.2.1 Toque Mágico . . . . .	51
6.3 Interação 2D usando objetos tangíveis . . . . .	53
6.3.1 Tangible Beats . . . . .	53
6.4 Interação 3D usando uma varinha mágica . . . . .	61
6.4.1 Tapete Mágico . . . . .	61
<b>7 Conclusão . . . . .</b>	<b>67</b>
<b>Referências Bibliográficas . . . . .</b>	<b>69</b>

# Capítulo 1

## Introdução

### 1.1 Motivação

Comumente em computação pessoal, ao imaginarmos uma pessoa interagindo com algum computador, ocorre-nos o pensamento de alguém usando um dispositivo como: *desktop*, *laptop*, celular, *tablet*, video-game, *wearable*, etc. Seja como for, em tal situação o computador está confinado a uma determinada forma física, com a qual se dá o acesso aos programas requisitados pelo usuário. Em 1991, Mark Weiser, então pesquisador da Xerox PARC, observou:

*“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.”* [Wei91]

Tal visão revela a essência da computação ubíqua, paradigma de interação que prevê que o poder computacional esteja sempre presente no cotidiano do usuário humano. A intenção é que o computador se torne “invisível”, isto é, que permeie nossos arredores de tal modo que não mais o percebemos. Segundo Dix *et al.*, qualquer tecnologia de computação que permita interação humana fora de uma única estação de trabalho é uma tecnologia de computação ubíqua [DFAB04]. Telas interativas de grande escala, *pen-based interfaces* e tecnologias de voz e de visão são alguns exemplos.

Diferentemente da visão de Weiser, entretanto, hoje o acesso à informação se dá por meio de dispositivos específicos altamente personalizados - e a noção de “computadores em todo lugar” vem sendo materializada por meio da popularização dos dispositivos móveis [HWD10], pequenos por natureza. Por serem pequenos, eles inevitavelmente excluem certa gama de aplicações. Em contrapartida, o ambiente físico, naturalmente maior, pode ser equipado de modo a proporcionar interações mais confortáveis [XHH13] com aplicações que estejam, de algum modo, “misturadas” no espaço. Ao quebrarmos a concepção de que computador pessoal está exclusivamente associado a *desktop*, *mobile* e variações, abrimos espaço para entendê-lo também como um serviço proporcionado pelo ambiente no qual se encontra o ser humano. Em outras palavras: o computador perde sua forma física e o ambiente se torna mais interativo. Isto posto, percebemos que o campo da computação ubíqua apresenta oportunidades interessantes relacionadas ao design e à implementação de interfaces inovadoras.

Recentemente, tem havido grande popularização das telas sensíveis ao toque - algo já previsto há anos<sup>1</sup>. Hoje é possível encontrar, no varejo, tecnologia multi-toque (*multitouch*) amplamente disponível em *smartphones*, *tablets*, entre outros. Aplicativos existentes para esses dispositivos frequentemente usam manipulação direta, que - neste contexto - provê um mapeamento intuitivo entre pontos no espaço físico e pontos no espaço da tela [RDH09]. Entrada e saída se combinam e o acesso ao mundo digital passa a dispensar treinamento formal. Imaginamos que, futuramente, *displays* multi-toque estarão integrados, “misturados” em: paredes, chão, mesas e vidros - todos localizados em ambientes como: cafés, exposições de arte, salões de jogos, shoppings, etc.

---

<sup>1</sup>[http://usatoday30.usatoday.com/tech/products/2007-06-21-1895245927\\_x.htm](http://usatoday30.usatoday.com/tech/products/2007-06-21-1895245927_x.htm). Último acesso: 17 de maio de 2013.

Além de trazer o multi-toque para o ambiente, é possível utilizar, ainda, objetos físicos para acrescentar mais uma dimensão ao diálogo entre o usuário e o domínio digital. Dado que seres humanos frequentemente interagem entre si usando variadas formas de expressão (gestos, escrita e toques são alguns dos exemplos que ocorrem no cotidiano), podemos fazer uso desses aspectos no projeto de sistemas interativos.

Tecnologias de computação ubíqua extrapolam as fronteiras tradicionais de interação, trazendo algumas pistas de como serão os sistemas interativos futuros [DFAB04]. Neste trabalho, exploramos novas formas de interação em superfícies interativas. Usando equipamentos como projetores portáteis e câmeras RGBD<sup>2</sup>, todos encontrados atualmente no varejo, desenvolvemos um método para transformar superfícies comuns em espaços interativos multi-toque. Além de investigar questões técnicas, de natureza mais analítica, desejamos também sugerir, do ponto de vista da interface, uma visão relacionada ao que a tecnologia poderá proporcionar num futuro próximo.

*“We simply understand too little about the way the future will be, but can’t know unless we experiment more radically!”* [DFAB04]

## 1.2 O projeto

Objetivando explorar diferentes formas de interação no contexto da computação ubíqua (ver capítulo 6), neste projeto de mestrado criamos um protótipo que transforma uma superfície comum (rígida, opaca e não instrumentalizada) - como por exemplo uma parede ou uma mesa - num espaço interativo multi-toque (ver Figura 1.1). Denominada **Superfície Mágica**, ou simplesmente superfície interativa, a solução possibilita variadas formas de expressão e é construída com equipamentos portáteis e de baixo custo (vide Figura 1.2). O diálogo entre o espaço interativo e o usuário é inspirado nas ações de desenhar, gesticular e agarrar. Tal comunicação se dá por meio dos seguintes mecanismos de interação:

1. canetas coloridas;
2. varinha mágica;
3. apagador;
4. dedo.

Riscar a superfície com uma caneta colorida faz com que a região correspondente seja marcada digitalmente. As marcações podem ser removidas mediante a utilização de um apagador. Além disso, gestos com o dedo são utilizados para manipular (arrastar, redimensionar, etc.) elementos do espaço digital. Finalmente, nosso sistema reconhece gestos feitos com um bastão que, metaforicamente, faz o papel de uma “varinha mágica”.

Os artefatos físicos são objetos do cotidiano: marca-textos genéricos<sup>3</sup> (“canetas”) e uma “varinha mágica” montada a partir de um objeto de plástico. Ao contrário de uma lousa comum, cuja área de desenho é limitada, o espaço interativo aqui proposto é “infinito”: apesar de a superfície física ser limitada, o usuário pode arrastar/manipular o conteúdo desenhado.

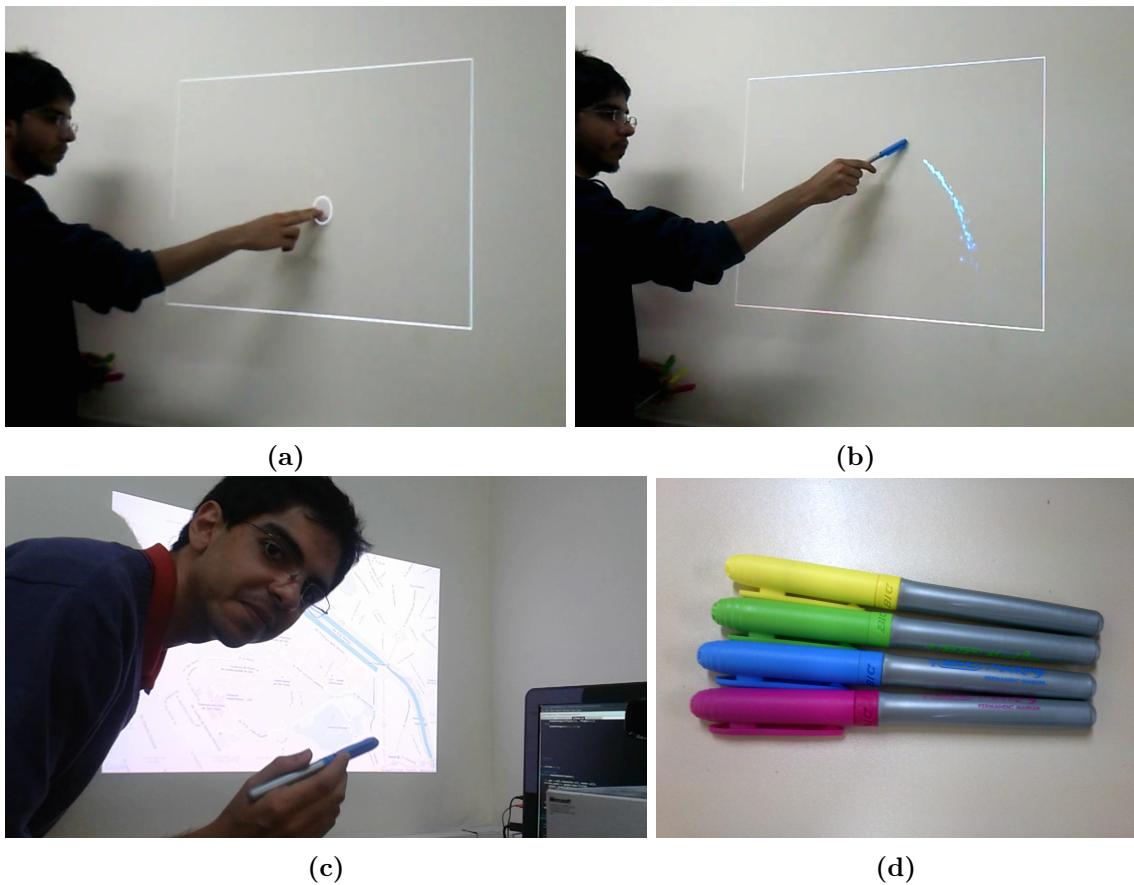
Superfícies interativas tais como a que desenvolvemos permitem explorar alternativas para se interagir com elementos virtuais. Baseada no estilo de manipulação direta, a forma de interação tem suas raízes nos atos de desenhar, gesticular e agarrar.

O site <https://github.com/alemart> inclui o código-fonte do projeto de mestrado. Adicionalmente, em <https://youtube.com/alemart88> é possível encontrar vídeos exibindo diversas aplicações interativas construídas durante a pós-graduação.

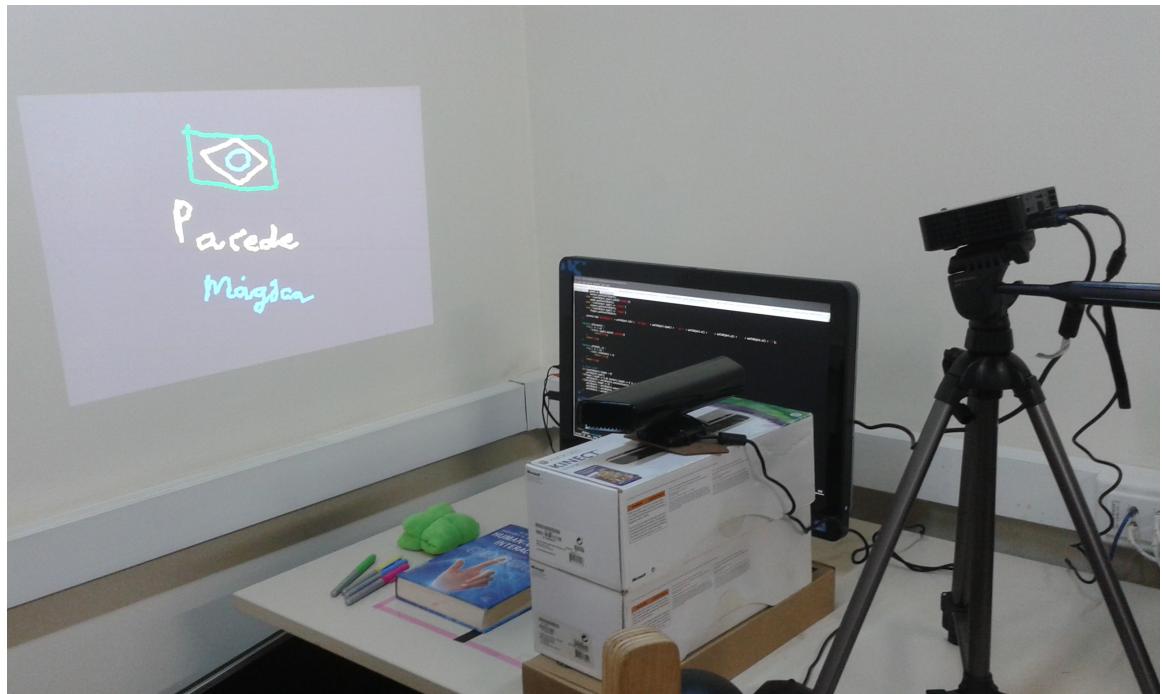
---

<sup>2</sup>Câmeras coloridas combinadas a sensores de profundidade.

<sup>3</sup>São encontrados em supermercados e em papelarias.



**Figura 1.1:** (a) Tocando uma parede com dedos das mãos. (b) Riscando uma parede com uma caneta colorida. (c) Um usuário e um mapa interativo. (d) Canetas coloridas.



**Figura 1.2:** Equipamentos utilizados na construção do protótipo.

### 1.3 Contexto

Um caso particular de superfície é uma parede. Veremos na seção 2.8 que paredes interativas podem estar imersas tanto em ambientes de trabalho quanto de entretenimento. Num contexto de trabalho, como um óbvio exemplo de uso de uma parede interativa citamos o planejamento colaborativo de projetos. Em administração de empresas, existe uma técnica denominada PERT<sup>4</sup>. Aplicando conceitos de teoria dos grafos, PERT ajuda o administrador a: definir o que deve ser feito, tomar decisões, visualizar incertezas e problemas potenciais, alocar tempo e recursos, etc. [DSUL68] Redes PERT são tipicamente representadas por meio de círculos (eventos) ligados por setas (atividades). Isto posto, uma parede interativa na qual se possa desenhar pode ser útil no planejamento de projetos, já que ela é capaz de mostrar, em tempo-real, potenciais gargalos e o(s) caminho(s) crítico(s) da rede.

Num contexto complementar ao de trabalho, notamos que uma superfície interativa também pode ser transformada numa plataforma de entretenimento digital. Sem treinamento prévio, pessoas poderiam usá-la como meio de expressão, permitindo brincadeiras diversas. Em espaços públicos, superfícies interativas podem ser usadas para captar a atenção das pessoas de modo a, por exemplo, anunciar uma marca.

### 1.4 Contribuições

As contribuições deste trabalho são três:

- O desenvolvimento de um sistema portátil e de baixo custo que transforma uma superfície comum num espaço interativo multi-toque;
- A ampliação do espaço interativo por meio de mecanismos que possibilitem interação 3D. Além de permitir toques na superfície (interação 2D), exploramos também o uso de gestos realizados fora da superfície, o que não é possível com telas sensíveis a toque convencionais;
- O design e a implementação de aplicações interativas como provas de conceito desta nova tecnologia.

### 1.5 Organização do Trabalho

No capítulo 2, discutimos conceitos relacionados ao projeto de mestrado. No capítulo 3, apresentamos alguns trabalhos relacionados ao tema desta dissertação. Já no capítulo 4, introduzimos as tecnologias envolvidas na parte prática do projeto. Depois, no capítulo 5, discutimos o design do sistema, bem como detalhes técnicos de implementação. No capítulo 6, exploramos diferentes formas de interação. Encerramos com a conclusão no capítulo 7.

---

<sup>4</sup>Program Evaluation and Review Technique.

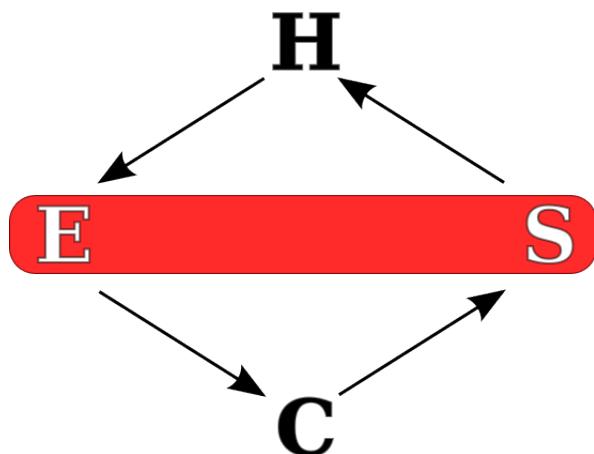
## Capítulo 2

# Fundamentos

### 2.1 Interação

Por volta dos anos 1980, à medida que o uso do computador se tornava mais difundido, passou a existir um crescente interesse em pesquisas envolvendo a interação entre pessoas e computadores [DFAB04]. Nascia então um campo interdisciplinar denominado Interação Humano Computador (IHC). IHC é uma disciplina que relaciona áreas como: ciência da computação, design, artes, psicologia, ergonomia, etc.

Quando falamos em interação, referimo-nos ao diálogo entre usuário e computador. Tradicionalmente, diz-se que um sistema interativo tem como propósito auxiliar um usuário humano a atingir objetivos [DFAB04]. No modelo proposto por Abowd e Beale [AB91], o processo de interação é quebrado em quatro vertentes: computador, humano, entrada (*input*) e saída (*output*). Aqui, o usuário se comunica com o sistema de computador por meio da interface, que é formada pelas componentes de entrada e de saída (ver região pintada na Figura 2.1).



**Figura 2.1:** *Interação Humano-Entrada-Computador-Saída.*

Fundamental ao modelo é a noção do ciclo interativo, representado na Figura 2.1 por meio de setas. Responsável por iniciar o diálogo, o usuário humano, que tem algum objetivo, concebe uma tarefa para atingí-lo. Posteriormente, tal usuário irá comunicar sua intenção ao sistema de computador por meio da entrada - e portanto deverá se expressar conforme as imposições dessa vertente. Em seguida, a intenção passada à entrada é repassada ao sistema na forma de operações a se desempenhar. O sistema de computador irá, então, transformar-se de modo a atender o que lhe foi pedido, informando seu novo estado à saída. Finalmente, é responsabilidade do usuário observar as alterações ocorridas na saída e avaliar, em relação a seu objetivo inicial, os resultados obtidos.

## 2.2 Interfaces naturais

O termo “interface natural” (*Natural User Interface*, ou NUI) aponta para uma área emergente em IHC. Focando-se em habilidades humanas como: olhar, toque, gestos corporais e voz, uma NUI busca utilizar canais de comunicação que afloram naturalmente nas atividades humanas, alavancando habilidades que as pessoas já desenvolveram durante suas interações no mundo físico<sup>1</sup>. Ao utilizar conhecimento prévio, tais interfaces abrem a porta para permitir a rápida transição de novato para *expert*, tornando-se efetivamente “invisíveis”<sup>2</sup>.

Nossa interação diária com outras pessoas e com o mundo é multissensorial: cada sentido fornece diferentes fragmentos de informação que nos ajudam a compor um todo. Como fazemos uso de diversos sentidos, parece razoável assumir que sistemas interativos que façam uso de mais de um canal sensorial também possam proporcionar uma experiência interativa mais rica ao usuário [DFAB04]. Segundo Dix *et al.*, o uso de múltiplos sentidos aumenta a largura de banda da interação entre o humano e o computador, tornando tal comunicação mais similar àquela desempenhada entre diferentes pessoas e entre pessoas e o ambiente - o que talvez torne o uso de tais sistemas mais “natural” [DFAB04].

Ao pensarmos em “interação natural”, lebramo-nos das interfaces multimodais, nas quais há o uso de várias modalidades de entrada para controlar um sistema [BKLJP04]. Na rotina diária, seres humanos naturalmente fazem uso simultâneo de diversos canais de comunicação. Similarmente, sistemas multimodais baseiam-se no uso concomitante de múltiplas modalidades de entrada [DFAB04]. Gestos, toques, olhar, fala, ondas cerebrais, batimentos cardíacos, movimentos das mãos, da cabeça e do corpo inteiro são alguns dos elementos que poderiam ser aproveitados, por um sistema computacional, para interação. Segundo Oviatt [Ovi03], um sistema multimodal processa, de maneira coordenada a um dispositivo multimídia de saída, duas ou mais modalidades de entrada. Nessa classe de interfaces, frequentemente tem-se como objetivo reconhecer expressões “naturais”, expressões que ocorrem no dia-a-dia durante a comunicação entre seres humanos, o que - em tese - possibilita um diálogo mais “intuitivo” com o computador. De fato: o desenvolvimento de sistemas multimodais inovadores tem sido facilitado pela crescente disponibilidade de grande variedade de tecnologias de *input* e *output* - e há um interesse crescente em tal área [Ovi03].

A indústria de video-games vem demonstrando a crescente importância das interfaces naturais. Em novembro de 2006, foi lançado o Nintendo Wii. Distribuído juntamente ao console, o controle Wii Remote (Wiimote) - dispositivo similar a um controle de televisão (vide Fig. 2.2a), dotado da capacidade de capturar certos movimentos do jogador [PB07] - foi caracterizado pela imprensa da época<sup>3</sup> como “intuitivo”, “tátil” e “inovador”, proporcionando ao Wii, com seus “jogos ridículamente divertidos” combinados a seu preço acessível, a posição de campeão de vendas entre os consoles de sua geração [Lee08].

Quatro anos mais tarde, em 2010, a Microsoft trouxe ao mercado o Kinect (ver Fig. 2.2b). Lançado inicialmente como um periférico para o console Xbox 360, tal dispositivo, composto essencialmente por uma câmera colorida, um sensor de profundidade, um motor e microfones, permite que o usuário interaja com jogos sem necessidade de qualquer *gamepad*: por meio de uma interface natural, o jogador pode simplesmente fazer gestos usando o próprio corpo, bem como emitir comandos de voz<sup>4</sup>. Em apenas 60 dias após seu lançamento, a Microsoft já havia vendido cerca de oito milhões de unidades<sup>5</sup>, o que resulta numa média de mais de 130 mil vendas por dia! Na época, a empresa considerou suas vendas como um “sucesso alucinante” e tal resultado foi parar no Livro Guinness dos Recordes como o aparelho eletrônico que vendeu mais unidades em menos tempo<sup>6</sup>.

<sup>1</sup><http://nuigroup.com/faq/>. Último acesso: 19 de maio de 2013.

<sup>2</sup>[https://en.wikipedia.org/wiki/Natural\\_user\\_interface](https://en.wikipedia.org/wiki/Natural_user_interface). Último acesso: 18 de junho de 2015/06/15.

<sup>3</sup><http://www.pcworld.idg.com.au/review/games/nintendo/wii/219764>. Último acesso: 19 de maio de 2013.

<sup>4</sup><https://en.wikipedia.org/wiki/Kinect>. Último acesso: 19 de maio de 2013.

<sup>5</sup>[http://gamasutra.com/view/news/32301/Microsoft\\_Ships\\_8\\_Million\\_Kinects\\_Introduces\\_Avatar\\_Kinect\\_Chat\\_Service.php](http://gamasutra.com/view/news/32301/Microsoft_Ships_8_Million_Kinects_Introduces_Avatar_Kinect_Chat_Service.php). Último acesso: 19 de maio de 2013.

<sup>6</sup>[http://www.gamasutra.com/view/news/33430/Microsoft\\_Kinect\\_Hits\\_10\\_Million\\_Units\\_10\\_Million\\_Games.php](http://www.gamasutra.com/view/news/33430/Microsoft_Kinect_Hits_10_Million_Units_10_Million_Games.php). Último acesso: 19 de maio de 2013.

Posteriormente, o mercado viu mais uma novidade no campo das interfaces naturais: o surgimento do Leap Motion<sup>7</sup>. Trata-se de um pequeno periférico projetado para rastrear, em 3D e com precisão sub-milimétrica, dedos da mão, canetas e outros objetos similares que entrem em seu campo de visão<sup>8</sup>. A empresa responsável pelo dispositivo, que já recebeu grandes investimentos e “centenas de milhares de pedidos”<sup>9</sup>, sugere que o aparelho encontra aplicações numa ampla variedade de áreas: jogos digitais, música, arte interativa, modelagem 3D, educação, finanças, medicina, etc<sup>10</sup>.

Na área de interfaces vestíveis (*wearable interfaces*), outros produtos comerciais recentes incluem: o Google Glass<sup>11</sup>, óculos com acesso à Internet e que fazem uso de realidade aumentada, o MYO<sup>12</sup>, dispositivo que, vestido num antebraço do usuário, interpreta sinais elétricos advindos da atividade muscular e o Apple Watch<sup>13</sup>, um relógio computadorizado.

Enfim, conforme afirmam Perani e Bressan, a evolução da área de IHC revela a importância crescente da valorização do uso do corpo no processo de interação, o que contribui para um maior envolvimento sensorial do ser humano durante sua comunicação com o computador [PB07].



(a) *Wii Remote*<sup>14</sup>.



(b) *Microsoft Kinect*.

## 2.3 Interfaces tangíveis

Em nossa rotina diária, sentimos e manipulamos o ambiente físico que nos circunda. Interagimos constantemente com o mundo material e, nesse processo, o sentido do tato se faz muito presente. Ao girar a maçaneta de uma porta, agarrar um copo de água ou mesmo pegar uma fruta, fazemos uso de complexas habilidades motoras. Atualmente, entretanto, todo esse complexo de habilidades é frequentemente deixado de lado quando interagimos com o domínio digital. Por exemplo: ao interagirmos com uma GUI<sup>15</sup>, na qual informações são retratadas por meio de pixels numa superfície física, geralmente utilizamos dispositivos de entrada como: teclado, *mouse*, tela sensível ao toque. Apesar de as GUIs serem muito populares e flexíveis, ao ficarmos limitados a elas acabamos não aproveitando um rico conjunto de informações motoras advindas do ser humano. Como resposta a essa falta, entram em cena as interfaces tangíveis (TUIs<sup>16</sup>). Na visão de Ishii e de Ullmer, as interfaces tangíveis permitem que conteúdo digital seja acoplado a objetos físicos, possibilitando que os usuários “agarrem e manipulem” informação [IU97].

As TUIs revelam uma maneira de trazer à realidade a visão de computação ubíqua proposta por Weiser: entrelaçar a tecnologia digital no mundo físico, tornando o computador “invisível” [Wei91]. Em vez de fazer com que, numa tela, pixels se traduzam em diversos *widgets*, as TUIs aproveitam as formas físicas dos objetos numa tentativa de reduzir, tanto quanto possível, a descontinuidade entre os mundos digital e real [Ish08]. A Figura 2.3 mostra o reacTable, uma popular interface tangível relacionada à área musical [JGAK07].

<sup>7</sup><https://www.leapmotion.com/>. Último acesso: 20 de maio de 2013.

<sup>8</sup>[http://en.wikipedia.org/wiki/Leap\\_Motion](http://en.wikipedia.org/wiki/Leap_Motion). Último acesso: 20 de maio de 2013.

<sup>9</sup><http://techland.time.com/2013/02/27/leap-motion-pushes-ship-date-to-may-13-announces-app-store/>. Último acesso: 20 de maio de 2013.

<sup>10</sup><https://www.leapmotion.com/apps>. Último acesso: 20 de maio de 2013.

<sup>11</sup><http://www.google.com/glass>. Último acesso: 28 de junho de 2013.

<sup>12</sup><https://www.thalmic.com/myo/>. Último acesso: 28 de junho de 2013.

<sup>13</sup><https://www.apple.com/br/watch/>. Último acesso: 6 de junho de 2015.

<sup>14</sup>Autor: Yeray Hernández. CC-BY 2.0. <https://www.flickr.com/photos/yerahg/551627536>. Último acesso: 18 de junho de 2015.

<sup>15</sup>Do inglês, *Graphical User Interface*.

<sup>16</sup>Do inglês, *Tangible User Interfaces*.



**Figura 2.3:** *reacTable*<sup>17</sup> [JGAK07].

## 2.4 Interfaces baseadas em gestos

Segundo Mitra e Acharya, gestos são movimentos corporais significativos que envolvem dedos, braços, cabeça ou o corpo inteiro, tendo como objetivo transmitir informação ou interagir com o ambiente [MA07]. Ghirotti e Morimoto apontaram duas principais razões para utilizar gestos para interagir com computadores [GM10]:

1. As pessoas usam ampla variedade de gestos no dia-a-dia. Novos gestos podem ser aprendidos fácil e rapidamente ao observá-los sendo desempenhados por outros;
2. Interfaces baseadas em gestos permitem o uso de “frases gestuais”. Tais frases são formas de se dividir o diálogo homem-máquina em partes. Cada parte tem um significado simples, podendo ser facilmente interpretada por computadores. Por exemplo, a ação de deslocar um objeto num ambiente virtual pode ser decomposta em três partes: pegá-lo, movê-lo e soltá-lo.

Conforme mostrado na Figura 2.4, gestos podem ser estáticos (a pessoa assume uma pose específica), dinâmicos (há movimento) ou uma mistura dos dois (usado, por exemplo, em linguagens de sinais). O reconhecimento automático de gestos dinâmicos requer segmentação temporal. Frequentemente, o usuário precisa especificar o início e o final de um gesto no espaço e no tempo [MA07].

O advento de reconhecedores de gestos leves e robustos possibilita a prototipagem rápida de interfaces baseadas em gestos [WWL07, Li10]. Após normalizar os gestos, frequentemente adicionando invariância de posição, escala e rotação, eles empregam técnicas de *template matching* para o reconhecimento. Tais reconhecedores normalmente se baseiam em trigonometria e geometria elementar e apresentam ótimo desempenho com apenas poucas amostras de treinamento, o que faz deles uma escolha atraente para os propósitos deste trabalho.

<sup>17</sup>Autor: Daniel Williams. CC-BY-SA 2.0. [https://commons.wikimedia.org/wiki/File:Reactable\\_Multitouch.jpg](https://commons.wikimedia.org/wiki/File:Reactable_Multitouch.jpg). Último acesso: 18 de junho de 2015.

(a) Gesto dinâmico<sup>18</sup>.

(b) Gesto estático.

**Figura 2.4:** Tipos de gestos.

## 2.5 Interfaces 3D

À medida que nos distanciamos de ambientes de escritório, interagir com os computadores da forma como interagimos com ambientes desktop não é mais o suficiente. Avanços em software e hardware estimularam o desenvolvimento de uma nova geração de interfaces de usuário denominadas pós-WIMP. Interfaces pós-WIMP contêm pelo menos uma técnica de interação que não é baseada em *widgets* bidimensionais tais como menus e ícones [VD97]. A motivação que rege essa nova geração de interfaces é empregar o conhecimento mundano, pré-existente dos usuários de modo a reduzir a distância entre os objetivos dos humanos e as ações necessárias para alcançá-los [JGH<sup>+</sup>08].

Aplicações desenvolvidas para ambientes não-desktop frequentemente requerem novas abordagens para interação: novos dispositivos, novas técnicas e novas metáforas. Por exemplo: uma aplicação de realidade virtual pode permitir que um usuário coloque um objeto em qualquer posição num espaço 3D, uma tarefa para o qual um *mouse* 2D é inadequado [BKLJP04]. Neste caso, o uso de uma interface 3D poderia ser mais imediato, isto é, apresentar menor distância cognitiva entre a ação do usuário e o retorno do sistema computacional.

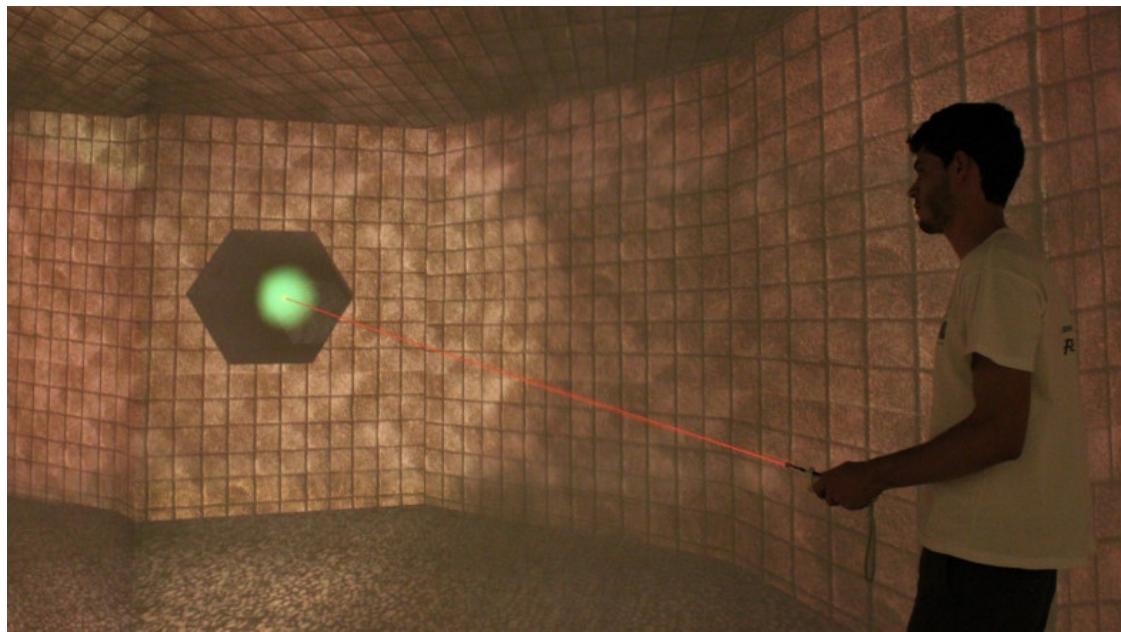
De acordo com Bowman *et al.* [BKLJP04], interação 3D é toda interação na qual as tarefas do usuário são desempenhadas num contexto tridimensional. Sistemas interativos que exibam gráficos 3D não necessariamente apresentam interação 3D (exemplo: não há interação 3D quando um usuário seleciona, através de um menu convencional, diferentes pontos de vista de uma maquete virtual). Por outro lado, pode haver interação 3D sem um dispositivo de entrada tridimensional (exemplo: se um usuário, como estratégia de navegação, clica num objeto-alvo numa maquete virtual para deslocar-se até ele, então um dispositivo de entrada bidimensional foi mapeado para uma localização tridimensional, e portanto houve interação 3D). Uma interface de usuário 3D (3D UI<sup>19</sup>) é uma interface de usuário que envolve interação 3D. A Figura 2.5 mostra um exemplo de interface 3D.

## 2.6 Telas sensíveis ao toque

Quando temos uma gama complexa de informações gráficas sendo mostradas numa tela (ex: em ferramentas de desenho), torna-se interessante manipular diretamente os itens exibidos (apontar, selecionar, etc.). Em comparação com a abordagem de digitar códigos num teclado, o uso de mecanismos de manipulação direta reduz a chance de erros de digitação, mantém a atenção do usuário no *display* e evita que ele tenha que aprender comandos [SP05], favorecendo o ato de reconhecer em

<sup>18</sup>Autor: Andrew Lee Neer. CC-BY-SA 3.0. [https://commons.wikimedia.org/wiki/File:Neer\\_Conducting,\\_CAA\\_\(Detroit,\\_MI\)\\_2011\\_\(1\).jpg](https://commons.wikimedia.org/wiki/File:Neer_Conducting,_CAA_(Detroit,_MI)_2011_(1).jpg). Último acesso: 18 de junho de 2015.

<sup>19</sup>Do inglês, *3D user interface*.



**Figura 2.5:** Point and go [CRdS<sup>+</sup> 12].

detrimento do de relembrar. Nesse contexto, entram em cena os dispositivos de apontamento, que podem oferecer tanto controle direto (*touchscreen, stylus*<sup>20</sup>) quanto indireto (*mouse, touchpad*).

Telas sensíveis ao toque, também conhecidas como *touchscreens*, possibilitam ao usuário apontar e selecionar objetos mostrados na tela. Contudo, ao contrário do *mouse*, as *touchscreens* oferecem controle direto, já que detectam eventos de toque na própria tela. Como exibem características tanto de entrada quanto de saída, o uso de tais dispositivos se torna apropriado em “ambientes hostis”, já que não há hardware separado que possa ser danificado [DFAB04]. Vale acrescentar que designers de sistemas posicionados em espaços públicos valorizam as telas sensíveis ao toque [SP05], já que a durabilidade desses dispositivos em ambientes muito movimentados é alta.

Hoje em dia, as telas sensíveis ao toque estão disponíveis em: caixas eletrônicos, *smartphones*, *tablets* e até mesmo em *notebooks*. *Touchscreens* são muito relevantes no mercado: além da presença dessa tecnologia no conhecido *iPad*, observamos também que a venda de *notebooks* com telas sensíveis ao toque vem crescendo substancialmente<sup>21</sup>. A própria Microsoft, recentemente, procurou favorecer o uso do toque na interface Metro do Windows 8<sup>22 23</sup>.

Atualmente, em celulares e *tablets*, encontra-se normalmente telas sensíveis ao toque construídas com tecnologia capacitiva ou resistiva<sup>24</sup>. Numa tela capacitiva, há alterações das características elétricas do sensor quando os dedos das pessoas entram em contato com a superfície. Em contrapartida, uma tela resistiva reage à pressão nela realizada: há a detecção de toques graças à colisão de duas placas que ficam sobrepostas. Uma limitação dessas telas reside no fato de que elas não são facilmente escaláveis.

Um termo frequentemente utilizado na literatura é o multi-toque (*multitouch*), que é uma tecnologia de interação capaz de detectar dois ou mais contatos sobre uma superfície (normalmente, dedos humanos). Trabalhos acadêmicos envolvendo multi-toque já vêm sido desenvolvidos pelo menos desde a década de 1980 [LBS85]. Entretanto, foi a partir de 2005, quando Jeff Han demonstrou

<sup>20</sup>pequena peça similar a uma caneta - porém, sem tinta.

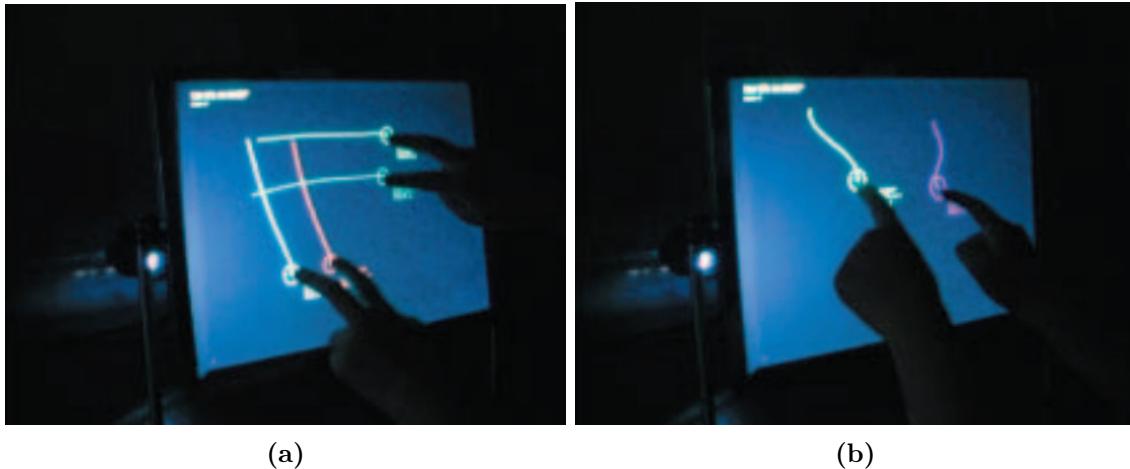
<sup>21</sup>[http://olhardigital.uol.com.br/produtos/digital\\_news/noticias/venda-de-notebooks-com-touchscreen-aumenta-51,8](http://olhardigital.uol.com.br/produtos/digital_news/noticias/venda-de-notebooks-com-touchscreen-aumenta-51,8). Último acesso: 28 de maio de 2013.

<sup>22</sup>[http://en.wikipedia.org/wiki/Windows\\_8](http://en.wikipedia.org/wiki/Windows_8). Último acesso: 28 de maio de 2013.

<sup>23</sup>Convém ressaltar: existe um fenômeno vulgarmente conhecido como “gorilla arm”, que se refere ao fato de que é ergonomicamente ruim manter o braço na horizontal e sem apoio por muito tempo. Interagir com uma tela sensível ao toque disposta na vertical pode causar esse problema.

<sup>24</sup><http://www.zoom.com.br/celular/deumzoom/qual-a-diferenca-entre-tela-capacitiva-e-tela-resistiva>. Último acesso: 11 de junho de 2013.

a FTIR<sup>25</sup> - uma técnica simples, de baixo custo e facilmente escalável que pode ser usada para se criar telas multi-toque [Han05] - que um grande interesse na tecnologia se tornou evidente<sup>26</sup>. De lá para cá, surgiram diversos projetos envolvendo aplicações da tecnologia multi-toque. Em 2009, por exemplo, Taylor *et al.* propuseram o VPlay [TIK<sup>+</sup>09]: uma mesa interativa multi-toque que permitia que os usuários, de maneira colaborativa, misturassem e manipulassem múltiplos vídeos em tempo real. No mesmo ano, Hochenbaum e Vallis publicaram o Bricktable [HV09]: um sistema voltado à autoria musical que combinava multi-toque e interfaces tangíveis [Ish08]. Comercialmente, citamos o Microsoft PixelSense<sup>27</sup>: uma superfície horizontal interativa multi-toque que também é capaz de reconhecer objetos nela colocados.



**Figura 2.6:** Interação multi-toque utilizando a técnica FTIR [Han05].

## 2.7 Visão computacional e processamento de imagens

De acordo com Gomes e Velho [GV08], ao conjunto de técnicas e métodos relativos à manipulação de dados ou imagens no computador denomina-se genericamente de Computação Gráfica. A computação gráfica pode ser dividida em quatro sub-áreas, que se diferenciam pela natureza dos objetos manipulados em cada uma delas: Modelagem Geométrica, Síntese de Imagens, Processamento de Imagens e Visão Computacional (ver Fig. 2.7).

Na Modelagem Geométrica, descreve-se dados geométricos no computador. Na Síntese de Imagens (ou Visualização), utiliza-se dados advindos de um sistema de modelagem geométrica para a geração de uma imagem que pode ser exibida por meio de algum dispositivo gráfico de saída.

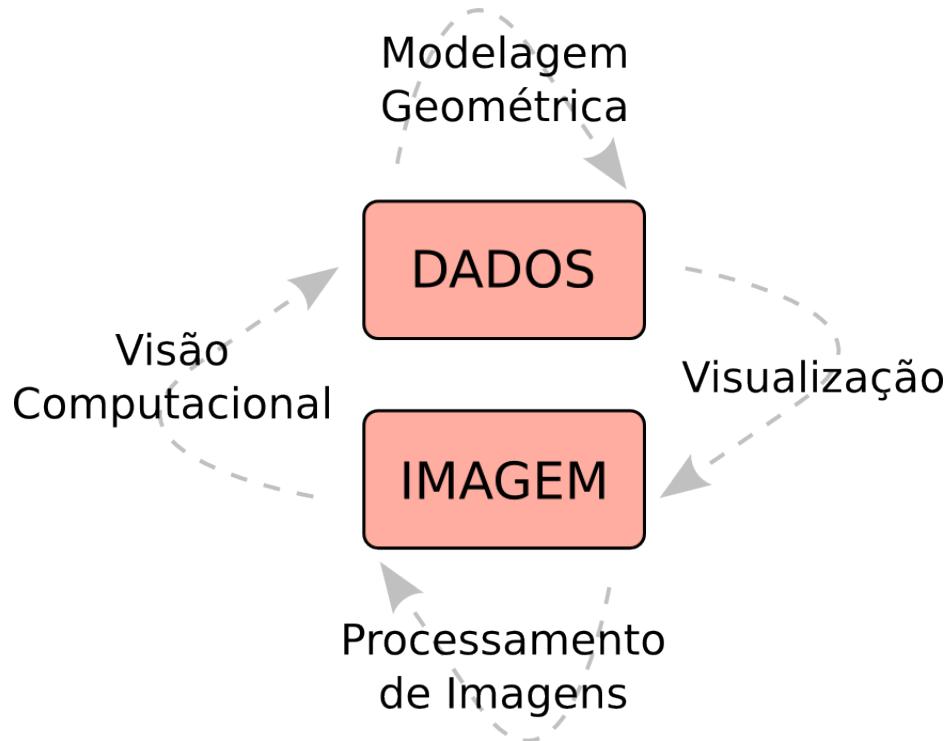
Quando falamos em Processamento de Imagens, referimo-nos a uma forma de processamento de dados na qual um sistema recebe uma imagem como entrada e, após processá-la, devolve outra imagem como saída. Frequentemente, o processamento de imagens é uma etapa que antecede o desempenhar de novos processamentos, como por exemplo o reconhecimento de padrões. Operações comuns na área incluem: segmentação de objetos, transformações geométricas (rotação, escala, *shearing*, etc.), correção de cor, realce de características, entre outras.

A área de Visão Computacional (ou Análise de Imagens) trabalha para a obtenção, a partir de imagens de entrada, informações sobre os dados que as originaram. Em outras palavras, ao contrário da área de visualização, que trata da geração de imagens, a de visão se ocupa da interpretação das imagens. Técnicas de visão computacional são de grande utilidade em diversas áreas, tais como: robótica [RBTH10], realidade aumentada [BF13] e interação humano-computador [Han05].

<sup>25</sup>Frustrated Total Internal Reflection.

<sup>26</sup>Jeff Han demonstrou seu protótipo na conferência TED: <http://www.youtube.com/watch?v=ac0E6deG4AU>. Último acesso: 11 de junho de 2013.

<sup>27</sup><http://www.pixelseNSE.com/>. Último acesso: 11 de junho de 2013.



**Figura 2.7:** *Computação Gráfica, adaptado de [GV08].*

## 2.8 Superfícies interativas

Segundo o *Cambridge English Dictionary*, uma superfície é a “parte ou camada superior ou externa de alguma coisa”, “a parte plana superior de uma mesa, de um armário, etc.”<sup>28</sup>. Ao introduzirmos elementos que envolvem: o multi-toque, o apelo aos sentidos, a computação ubíqua, a interatividade e elementos multimídia, ampliamos o entendimento convencional do que vem a ser uma superfície. Isto posto, neste trabalho entendemos que uma superfície interativa é uma superfície com computação e interação nela embutidas.

Um exemplo particular de superfície é uma parede. Segundo o Michaelis Moderno Dicionário da Língua Portuguesa, parede é uma “obra geralmente de tijolo e argamassa, ou outro material apropriado, com que se fecham externamente os edifícios, bem como se formam suas divisões internas”<sup>29</sup>. Neste trabalho, entretanto, entenderemos parede como uma superfície material, plana e perpendicular ao chão.

O termo parede comumente transmite uma ideia ligada à noção de obstáculo, barreira. Podemos, entretanto, estender esse conceito. Segundo Moutinho [Mou10], as paredes interativas<sup>30</sup> “pretendem ser aplicações introduzidas na estrutura arquitetônica dos edifícios” (ver Fig. 2.8), sendo que há um diálogo entre: ser humano, parede interativa (sistema) e ambiente envolvente. Tal diálogo pode ser decomposto em:

- **Diálogo sistema - ambiente envolvente:** consiste na noção de que a parede interativa, por meio de sensores nela embutidos, pode responder a circunstâncias e estímulos advindos do espaço físico no qual ela se localiza. Por exemplo: a parede pode reagir à temperatura ambiente, à intensidade sonora presente nos arredores, à luminosidade existente, ou mesmo à quantidade de pessoas passando pelo espaço. A reação da parede pode se dar de múltiplas maneiras: mostrar anúncios, trocar cores, exibir alguma animação, etc.

<sup>28</sup><http://dictionary.cambridge.org/dictionary/british/surface>. Último acesso: 18 de junho de 2015.

<sup>29</sup><http://michaelis.uol.com.br/moderno/portugues/index.php?lingua=portugues-portugues&palavra=parede>. Último acesso: 10 de junho de 2013.

<sup>30</sup>Moutinho usa o termo “paredes inteligentes”, que é equivalente.

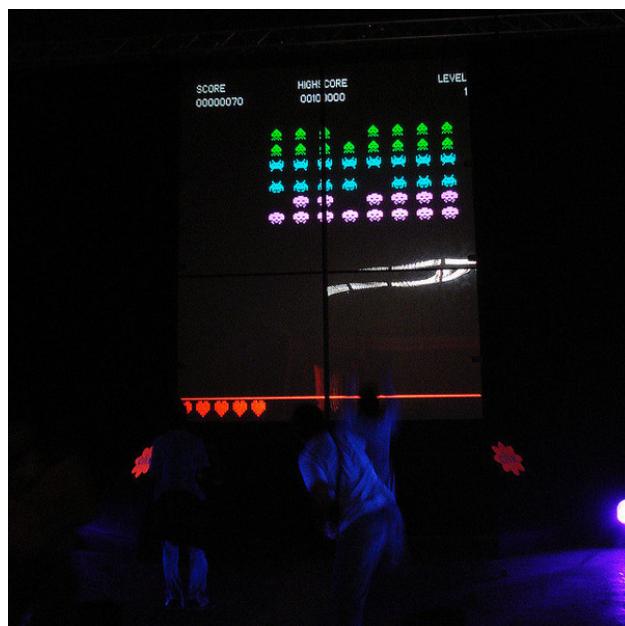
- **Diálogo usuário - sistema:** ocorre quando a parede responde a um estímulo intencional originado por um usuário humano. Por exemplo: a detecção de toques, o rastreamento do corpo ou mesmo o reconhecimento da fala podem ser usados, pela parede, para dialogar com o usuário. Um mecanismo de multi-toque poderia ser traduzido na realização de um desenho, o que daria origem a uma funcionalidade artística presente no ambiente.
- **Diálogo sistema - sistema:** neste caso, por exemplo, duas ou mais paredes interativas podem se comunicar e cruzar dados, mesmo remotamente. Como um caso de uso, pode-se imaginar soluções de monitoramento e segurança capazes de identificar pessoas e registrar seus comportamentos numa base de dados central. Moutinho alerta que a incorporação desse tipo de “inteligência” na infra-estrutura presente no ambiente é como “um lobo em pele de cordeiro”.

Além disso, os ambientes envolventes podem ser classificados em:

- **Ambientes de trabalho:** inclui grande variedade de espaços, tais como escritórios, ambientes dedicados a atividades educacionais, hospitais, etc. Paredes interativas podem permitir que os espaços sejam adaptados de acordo com as funções neles desenvolvidas;
- **Ambientes de entretenimento:** apresentam amplo contexto de utilização, cobrindo residências e espaços públicos (museus, shoppings, aeroportos, etc.). Paredes interativas podem ser utilizadas para captar a atenção do público, de modo a oferecer serviços diversos, conteúdo educacional, ou meramente diversão.

Moutinho ainda sugere que as paredes interativas poderiam oferecer a possibilidade de sofrer mutações geométricas. Além disso, como no caso da Smart Glass<sup>31</sup>, tais paredes também poderiam controlar a passagem de luz - o que, na prática, permitiria a reconfiguração dos espaços de modo a surpreender, por exemplo, necessidades de privacidade visual.

Complementarmente, citamos aqui as Powerwalls: matrizes de telas separadas que funcionam conjuntamente de modo a prover um *display* unificado. Powerwalls podem ser construídas, por exemplo, por meio de sistemas de multi-projeção tais como o FastFusion, utilizado numa CAVE<sup>32</sup> [TKCZ12].



**Figura 2.8:** *Space Invaders XL*<sup>33</sup>.

<sup>31</sup><http://www.smartglassinternational.com>. Último acesso: 11 de junho de 2013.

<sup>32</sup>*Cave Automatic Virtual Environment*.

<sup>33</sup>Autor: Multitouch Barcelona (foto de Sandra Moreira). CC-BY-NC-ND 3.0. <https://www.behance.net/gallery/4276719/Multitouch-Space-Invaders>. Último acesso: 16 de dezembro de 2015.



# Capítulo 3

## Trabalhos relacionados

Pesquisas envolvendo superfícies interativas existem há décadas: já na década de 1960, Sutherland construiu o Sketchpad [Sut64], que incluia um *display* CRT, uma *light-pen* e a noção de que o ser humano deveria ser capaz de desenhar diretamente na tela do computador. Na época, tal trabalho demonstrou que não era necessário forçar o usuário a se comunicar como um computador: ao permitir que a memória da máquina fosse manipulada de maneira visual, era possível ao computador transformar abstrações em formas perceptuais (e vice-versa). Anos mais tarde, em 1992, nos laboratórios da Xerox PARC foi desenvolvido o Liveboard [EBG<sup>+</sup>92]: uma larga superfície interativa voltada a reuniões e trabalhos colaborativos. O protótipo era envolto num gabinete de madeira e permitia que, com uma caneta, o usuário desenhasse diretamente sobre a tela e executasse programas através de uma interface do usuário estilo WIMP<sup>1</sup>. O Liveboard trazia consigo a visão das *computer walls*, que se encaixa no contexto da computação ubíqua.

Com o avanço em áreas como: placas gráficas, processadores, projetores e interfaces naturais, começaram a surgir, na área de computação ubíqua, trabalhos interativos de fácil instalação e/ou “futuristas”. Veremos, nesta seção, alguns deles.

### 3.1 Trabalhos acadêmicos

#### 3.1.1 Quadro-branco multi-toque de baixo custo usando o Wiimote

Usando um controle Wiimote (ver 2.2), J. C. Lee demonstrou como transformar uma superfície plana, como uma parede ou uma mesa, num quadro-branco interativo [Lee08]. Dado que o Wiimote - que pode ser conectado a um PC via Bluetooth - possui, entre outros recursos, um sensor infra-vermelho de alta velocidade (100 Hz) capaz de rastrear simultaneamente até quatro fontes luminosas, Lee observou que o periférico de video-game é uma “plataforma de impressionante custo-benefício” viável para pesquisa em interação. Em seu popular vídeo na Internet<sup>2</sup>, Lee exibiu um protótipo no qual uma caneta dotada de um LED<sup>3</sup> infra-vermelho era utilizada para riscar, digitalmente, uma superfície (ver Fig. 3.1). Um botão na caneta acendia/apagava o LED.

A instalação física do quadro-branco interativo requer: um projetor, um Wiimote, um computador e pelo menos uma caneta LED (alternativamente, o projetor pode ser substituído por um monitor LCD). O sensor infra-vermelho do Wiimote é posicionado em direção à superfície interativa de tal modo que não exista oclusão do(s) LED(s) da(s) caneta(s). Ao restringir a trajetória da(s) caneta(s) a um plano, torna-se possível mapear o sistema de coordenadas do sensor para o sistema de coordenadas do quadro-branco digital. Por meio de matrizes de homografia, calculadas num procedimento de calibração de quatro pontos, qualquer ponto luminoso detectado pela câmera do Wiimote pode ser mapeado para o pixel correspondente no *display*. Como o controle do Wii detecta até quatro pontos luminosos ao mesmo tempo, Lee observa que a mesma abordagem possibilita a

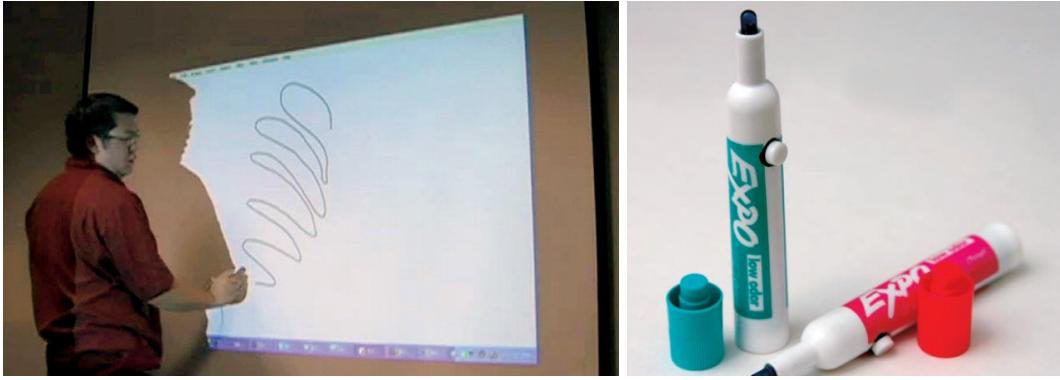
<sup>1</sup> Windows, Icons, Menus, Pointers.

<sup>2</sup><http://www.youtube.com/watch?v=5s5EvhHy7eQ>. Último acesso: 2 de junho de 2013.

<sup>3</sup>Light-emitting diode.

criação de superfícies interativas multi-toque.

A solução proposta por Lee apresenta algumas limitações: a autonomia do controle é limitada, a sensibilidade da caneta é afetada pela luminosidade do ambiente e um mau posicionamento do sensor pode originar oclusão, atrapalhando a experiência do usuário. Apesar dos problemas, a tecnologia foi bem recebida em instalações educacionais, tendo sido considerada “fácil de usar” [ST09].



**Figura 3.1:** Display interativo [Lee08].

### 3.1.2 Usando o Kinect para emular um quadro-branco interativo

Em 2011, Avancini e Ronchetti propuseram uma solução de baixo custo que utilizava um Microsoft Kinect (ver 4.2) para transformar uma parede comum num quadro-branco interativo [AR11]. No protótipo desenvolvido naquele projeto de mestrado, um Kinect e um projetor eram posicionados de forma a apontar para uma parede. Com o dedo indicador da mão direita, o usuário podia controlar a posição do cursor do *mouse*. O sistema - desenvolvido usando o arcabouço OpenNI e o middleware NITE<sup>4</sup> (*Natural Interaction Technology for End-users*), ambos encontrados gratuitamente na Internet - não suporta multi-toque. Por outro lado, o protótipo também não requer canetas LED.

Um processo de calibração de três etapas revela a ideia central da solução computacional proposta no projeto. São elas: *skeleton calibration*, *point calibration* e *finger calibration*. Primeiramente, o usuário se posiciona à frente do Kinect e faz uma determinada pose corporal denominada Psi. Assim, OpenNI e NITE são usados, como “caixas pretas”, para detectar o humano. Isto possibilita o rastreamento do esqueleto e a remoção do fundo. Em seguida, por meio de uma nova calibração, é encontrado um mapeamento entre o sistema de coordenadas do sensor de profundidade do Kinect e o sistema de coordenadas do *display* projetado na parede. Finalmente, uma última calibração, na qual se calcula um deslocamento entre a palma da mão direita do usuário e a ponta do dedo indicador correspondente, é efetuada. Nessa etapa, a pessoa indica, interativamente, qual é o deslocamento. Este último passo é realizado porque o esqueleto obtido pelo NITE não é suficientemente detalhado a ponto de incluir os dedos das mãos.

Da forma que o sistema foi concebido, o usuário não pode girar o punho, pois uma vez calculado o deslocamento entre a palma da mão direita e a ponta do dedo indicador, não há qualquer procedimento de atualização desse dado. Além disso, os pesquisadores reconhecem outra limitação importante: ao contrário do que seria de se esperar, para efetuar um clique a pessoa deve utilizar um *mouse* comum, que é segurado na outra mão.

<sup>4</sup>O NITE transforma, usando algoritmos de visão computacional, mapas de profundidade em “dados significativos”. Por exemplo: o sensor Kinect observa a cena, o OpenNI fornece o mapa de profundidade e o NITE faz o rastreamento do esqueleto do usuário. Tal esqueleto pode ser prontamente utilizado pelo programador.

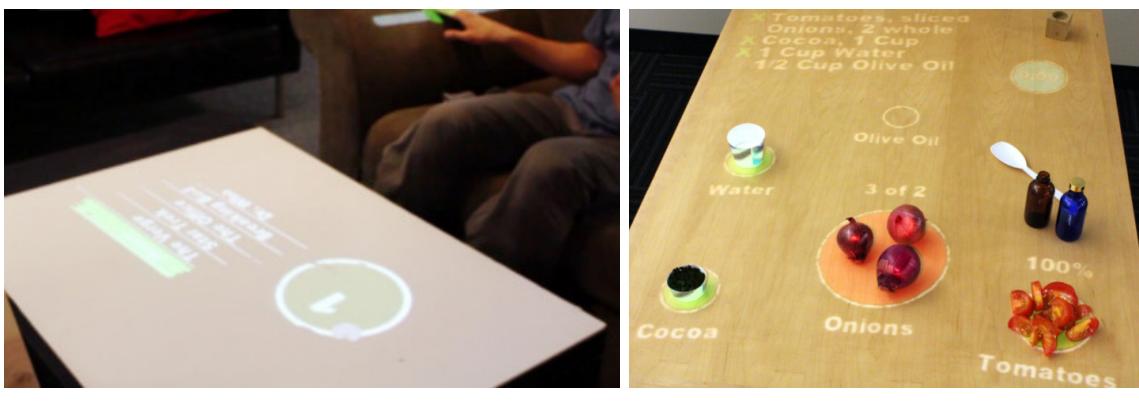
### 3.1.3 WorldKit: criação rápida e fácil de aplicações interativas ad-hoc em superfícies do dia-a-dia

Inspirado na noção de computação ubíqua, o WorldKit [XHH13] permite que o usuário use superfícies comuns, do dia-a-dia (mesas, chão, sofás, etc.), para interagir com aplicativos. Usando um projetor e um sensor de profundidade (Kinect), o WorldKit torna possível “misturar” programas ao ambiente físico: as superfícies, que se tornam sensíveis ao toque, possibilitam fazer uso de recursos computacionais por meio de um espaço que já é familiar ao usuário (vide Figura 3.2). No contexto do lar, por exemplo, há a possibilidade de se desenvolver um sistema de entretenimento<sup>5</sup> que, integrado ao ambiente, dê a impressão de que o computador perdeu sua forma física - “desapareceu”.

A ideia central do trabalho reside no fato de que o usuário pode, ao esfregar a palma da mão em alguma superfície do ambiente, “pintar” um *interactor*<sup>6</sup> na localização especificada. Desta maneira, numa sala de estar, é possível instanciar - por exemplo - um controle de televisão num dos braços de um sofá e um controle de iluminação da sala numa parede. Ao combinar vários *interactors*, que são desenvolvidos em linguagem Java, o usuário acaba “pintando” um aplicativo imerso no espaço físico.

O software que controla o sistema roda, num *laptop*, a 30 quadros por segundo (*framerate* do Kinect). A detecção de toques do usuário nas superfícies, realizada por meio da utilização do sensor de profundidade do Kinect, serve a dois propósitos: criar *interactors* e, posteriormente, interagir com eles. Diversos tipos de *interactors* foram implementados pelos autores do trabalho: *binary contact* (diz se a região na qual o *interactor* reside está ou não sendo tocada), *linear axis touch* (similar a barras de rolagem), *contact counting* (conta quantos objetos estão em contato com a superfície), *multitouch input*, etc.

Tecnicamente, a funcionalidade de detecção de toques consiste, inicialmente, em capturar 50 quadros consecutivos da imagem de profundidade do Kinect. Calcula-se então a média e o desvio padrão de cada pixel, de modo a criar um perfil de fundo e um de ruído, respectivamente. É claro que, nesta etapa, o ambiente não deve apresentar atividade (cena estacionária). Posteriormente, numa cena onde existe movimento, cada pixel advindo do sensor de profundidade é testado: se seu valor estiver distante da média em três ou mais desvios-padrão, ele é considerado significativo. Pixels significativos que estejam entre 3 e 50 milímetros de distância em relação ao perfil de fundo são considerados pixels de contato. Finalmente, por meio de um algoritmo de computação de componentes conexas, é feita a detecção dos *blobs* resultantes. Ruídos remanecentes são filtrados nesta etapa.



(a) Sala de estar.

(b) Cozinha.

**Figura 3.2:** Realidade aumentada em diferentes ambientes [XHH13].

<sup>5</sup>home entertainment system.

<sup>6</sup>Interactor, termo utilizado no artigo, corresponde aproximadamente ao conceito de *widgets*, que são as componentes de uma GUI: botões, barras de rolagem, etc.

### 3.1.4 The Office of the Future: uma abordagem unificada para modelagem baseada em imagens e para displays espacialmente imersivos

Raskar *et al.* introduziram ideias, tecnologias e resultados para “o escritório do futuro”, um projeto de *display* semi-imersivo alocado num ambiente de escritório (ver Fig. 3.3) onde se combina aquisição e renderização de cenas [RWC<sup>+</sup>98]. Os autores combinam técnicas de visão computacional e de computação gráfica num sistema que utiliza noções de CAVEs, *tiled display systems* e modelagem baseada em imagens. Uma das motivações citadas no trabalho é a suposição de que sistemas de telecolaboração que usam “cabeças falantes” 2D e lousas compartilhadas frequentemente inibem muito da interação que ocorreria caso os colaboradores estivessem de fato na mesma sala. Os autores entendem que, para uma comunicação interpessoal efetiva, é interessante ver e interagir com os colaboradores em 3D, “naturalmente” - isto é, usando todas as nuances sutis da comunicação verbal e não-verbal: apontar, gesticular, andar, etc. Para isso, eles propuseram o uso de um SID<sup>7</sup>, um *display* que circunda o usuário com um panorama de imagens.

Um dos principais módulos do sistema proposto é o que captura, continuamente e em temporal, modelos baseados em imagens do ambiente físico do escritório. Os autores utilizam câmeras e projetores, em pares e calibrados entre si, para obter a profundidade e a refletância de cada pixel da cena. O sistema emprega a projeção de padrões binários no ambiente. A câmera observa  $n$  imagens sucessivas, cria imagens binárias correspondentes usando uma técnica de limiarização adaptativa (i.e., uma técnica de limiarização que leva em conta variações na intensidade do fundo) e gera, para cada pixel, códigos de  $n$  bits correspondentes ao padrão projetado. Uma tabela esparsa de *lookup*, baseada em dados de calibração, permite a execução de uma interpolação trilinear para determinar as coordenadas tridimensionais de cada pixel da cena. Combinando-se tais coordenadas com os dados de cor e de refletância de cada pixel, tem-se o modelo. Vale ressaltar que a função de refletância, computada por meio da técnica *imperceptible structured light*, é utilizada para limiarizar as imagens projetadas de padrões binários de luz estruturada. Tal técnica permite que se projete padrões imperceptíveis ao olho humano, mas visíveis por uma câmera sincronizada ao projetor.

Outro objetivo do trabalho proposto é gerar imagens adaptadas à posição do usuário e que pareçam corretas mesmo quando projetadas em superfícies potencialmente irregulares. Como o usuário pode se mover no escritório, os autores empregaram um rastreador magnético para determinar sua localização 3D. O algoritmo de renderização, baseado numa abordagem de duas etapas, recebe como entrada: um modelo da superfície, os parâmetros intrínsecos e extrínsecos do projetor, a posição do usuário e uma imagem, tipicamente gerada por uma renderização 3D convencional. Como saída, ele devolve uma imagem correspondente à imagem de entrada que, quando exibida pelo projetor, aparece alinhada com a localização do humano no espaço tridimensional. No artigo, os autores também discutem a combinação geométrica das câmeras, para obter um modelo unificado da cena, e as funções de *blending* aplicadas aos diversos projetores de vídeo, para levar em conta a sobreposição das imagens projetadas.

### 3.1.5 The Everywhere Displays Projector: um dispositivo para criar interfaces gráficas ubíquas

No contexto da computação ubíqua, uma possibilidade é criar interfaces gráficas pervasivas. Com isso em mente, Pinhanez apresentou o *Everywhere Displays projector* [Pin01], um dispositivo que acopla um projetor de vídeo a um espelho rotatório, permitindo a renderização de imagens geradas por computador em locais como: paredes, mesas, móveis, geladeiras, portas de entrada, etc. O projeto, exibido na Figura 3.4, é descrito como um dos primeiros dispositivos que permite que um sistema de computador atue transparentemente no mundo físico. Metaforicamente, ele é descrito como um “braço robótico” de luz.

Pode-se deixar o equipamento num canto de uma sala, de modo a minimizar, na projeção, a presença de sombras decorrentes da atividade humana. Quando a projeção não é ortogonal à superfície, a imagem projetada é distorcida. Para corrigir a distorção ocasionada pela projeção

---

<sup>7</sup> *spatially immersive display*.

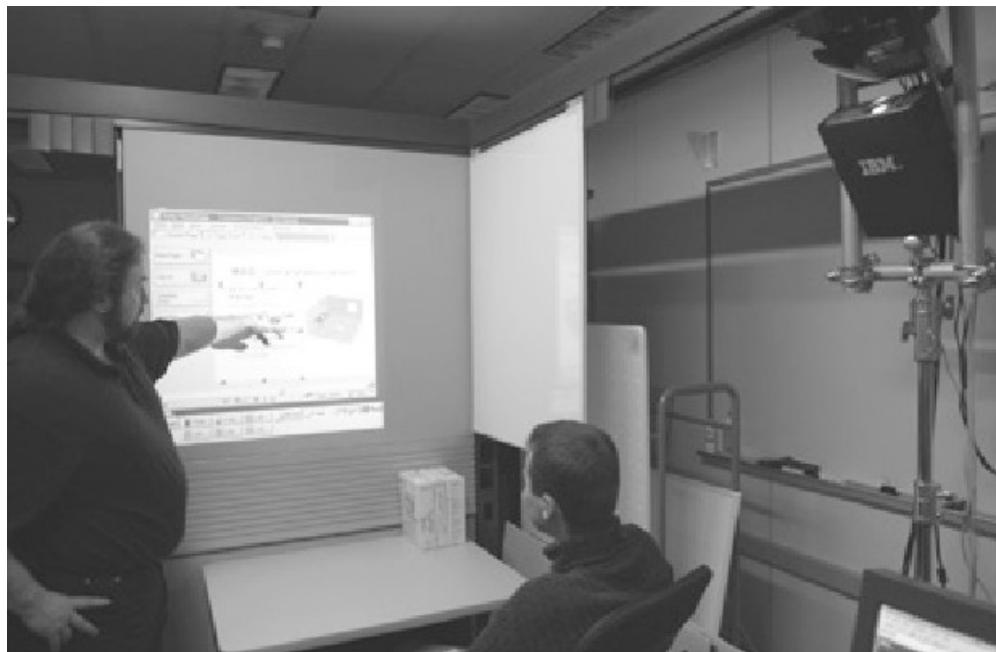


**Figura 3.3:** Esboço conceitual do “escritório do futuro” [RWC<sup>+</sup> 98].

oblíqua, o sistema desempenha uma distorção inversa antes da projeção. Ele faz uma simulação prévia num mundo virtual 3D, de tal forma que a imagem seja primeiramente projetada numa superfície virtual equivalente à real. Se a câmera virtual possuir as mesmas características focais do projetor e se a superfície virtual estiver posicionada de modo equivalente à real, então a cena capturada pela câmera virtual corresponde à cena “vista” pelo projetor. Segundo Pinhanez, como os projetores fazem o inverso das câmeras (i.e., eles projetam luz ao invés de capturá-la), o resultado é uma projeção livre de distorção.

Para desempenhar a projeção em diferentes superfícies do ambiente físico, o sistema deve ser recalibrado. Os parâmetros da calibração da superfície virtual são determinados manualmente por meio da projeção de um certo padrão virtual e do ajuste fino de escala, rotação e posição da superfície no mundo virtual 3D, bem como do campo de visão da câmera. O processo leva entre 10 a 20 minutos. O autor cita a possibilidade de automatizar tal processo.

O artigo também comenta a possibilidade de tornar as superfícies sensíveis ao toque. Embora algoritmos específicos não tenham sido detalhados, é discutido que isso se daria por meio do emprego de uma câmera rotatória e de técnicas de visão computacional que usam, por exemplo, o comprimento da sombra das mãos do usuário para inferir eventos de toque. Finalmente, o trabalho lista algumas possibilidades criativas de aplicação: a criação de *displays* interativos em superfícies não instrumentalizadas, o acesso à computação em espaços públicos e em ambientes hostis, o uso de realidade aumentada para trazer informação ao ambiente físico (ex: propaganda e entretenimento), entre outras.



**Figura 3.4:** Trabalho colaborativo com o Everywhere Displays [Pin01].

## 3.2 Trabalhos comerciais

### 3.2.1 Ubi Interactive

Com o mote “*interaction everywhere*”, Ubi Interactive<sup>8</sup> é uma empresa que é apoiada e financiada<sup>9</sup> através do *Kinect Accelerator Program*, programa fruto da colaboração entre a Microsoft e a TechStars<sup>10</sup>, uma incubadora de *startups*. Fundada por Chathoth e Zhang, a empresa vende um produto capaz de transformar paredes e mesas comuns em telas interativas multi-toque. Para atingir tal objetivo, um projetor convencional é combinado a um Microsoft Kinect. Por meio de um software desenvolvido pelos fundadores, pode-se interagir com programas Windows (mapas, jogos, apresentações, entre outros) na parede ou mesa. A empresa nasceu na Universidade Técnica de Munique (Alemanha).

A *startup* recebeu ampla cobertura da mídia<sup>11 12</sup>. Um dos pontos de venda do produto é o fato de as telas sensíveis a toque de grandes dimensões - aquelas apropriadas a salas de reuniões e estabelecimentos comerciais - serem muito caras. Como tais ambientes normalmente já contam com projetores ou televisões largas, os fundadores notaram que poderiam torná-los sensíveis a toque de uma forma barata. Numa entrevista<sup>13</sup>, um dos empresários especulou que, no futuro, os ambientes serão mais “inteligentes”, detectando a presença dos usuários e interagindo com eles de forma mais natural.

O produto, demonstrado em diversos vídeos que circulam na Internet<sup>14 15</sup>, requer o sistema operacional Windows 8 e é vendido no site oficial.

<sup>8</sup> <http://www.ubi-interactive.com/>. Último acesso: 14 de julho de 2013.

<sup>9</sup> <http://www.electronista.com/articles/12/05/28/startup.uses.kinect.to.make.existing.displays.into.3d.touchescreens/>. Último acesso: 14 de julho de 2013.

<sup>10</sup> <http://www.techstars.com/>. Último acesso: 14 de julho de 2013.

<sup>11</sup> <http://www.geekwire.com/2012/startups-building-businesses-microsoft-kinect/>. Último acesso: 14 de julho de 2013.

<sup>12</sup> [http://www.abcactionnews.com/dpp/news/science\\_tech/microsoft-startup-ubi-interactive-kinect-technology-to-develop-any/](http://www.abcactionnews.com/dpp/news/science_tech/microsoft-startup-ubi-interactive-kinect-technology-to-develop-any/). Último acesso: 14 de julho de 2013.

<sup>13</sup> <http://www.geekwire.com/2012/qa-ubi-interactives-chao-zhang-humans-interact-computers-future/>. Último acesso: 14 de julho de 2013.

<sup>14</sup> <http://www.youtube.com/watch?v=2dMpjBeY4z0>. Último acesso: 14 de julho de 2013.

<sup>15</sup> <http://www.youtube.com/watch?v=n8s6mpFohDQ>. Último acesso: 14 de julho de 2013.

# Capítulo 4

## Ferramentas de desenvolvimento

Nesta seção, discutimos brevemente algumas abordagens para se criar uma superfície interativa. Em seguida, apresentamos tecnologias que são usadas neste projeto de mestrado.

### 4.1 Visão geral

Há várias maneiras de se construir e obter uma superfície interativa. Uma delas é instalar uma lousa digital. Como exemplo, citamos as SMART Boards<sup>1</sup>, desenvolvidas pela empresa Smart Technologies. Tais lousas, que já incluem projetor, são utilizadas em contextos educacionais e estão disponíveis em diversos modelos. Os usuários podem interagir por meio de canetas ou de dedos da mão. De acordo com a Wikipedia<sup>2</sup>, nos modelos mais recentes a detecção dos toques é feita por meio de uma tecnologia proprietária denominada DViT™(*Digital Vision Touch*): câmeras localizadas nos quatro cantos da lousa enviam dados de posição a um módulo que determina a localização exata de contato (modelos mais antigos utilizam tecnologia resistiva e não aceitam *multitouch*). Em vários produtos da linha, canetas são guardadas numa bandeja acoplada à lousa. Ao tirar uma caneta da bandeja, um sensor detecta tal evento e faz com que a lousa passe a interpretar todos os toques com a tela como advindos dessa caneta.

Modelos de SMART Boards têm “áreas ativas da tela” de dimensões diversas: alguns modelos apresentam diagonal de 77” (proporção 4 : 3), enquanto que outros exibem 87” (16 : 10) ou mesmo 94 $\frac{1}{4}$ ” (16 : 9). Como resultado, tem-se áreas de desenho bastante extensas. De acordo com os manuais de usuário, tais produtos pesam entre 20 e 30 kg. Há também um SDK<sup>3</sup> disponível para desenvolvedores cadastrados no SMART Ecosystem Network (SEN). Apesar das diversas vantagens desses produtos, eles não são facilmente transportados. Além disso, os modelos mais novos (como a linha 800, que possui *multitouch*) têm preço salgado: num site de Internet<sup>4</sup>, encontramos uma SMART Board 800 usada custando oito mil e quinhentos reais. Diversos revendedores não divulgam preço e pedem para o cliente fazer orçamento. Do ponto de vista de desenvolvimento de software, o uso das SMART Boards apresenta desvantagens: o SDK tem licença de uso proprietária na qual a Smart Technologies reserva direitos exclusivos de distribuição dos aplicativos criados para a plataforma. Ademais, o SDK tem documentação deficiente, o fórum do SEN é pouco movimentado e alguns usuários-desenvolvedores reclamam impiedosamente do suporte técnico<sup>5</sup>.

Outra alternativa para se criar uma superfície interativa é usar um HP Pocket Whiteboard. Útil para salas de aula e de reunião, trata-se de uma ferramenta portátil, sem fio, que permite que se integre a qualquer superfície a funcionalidade de um quadro-branco. Diferentemente das SMART Boards, o HP Pocket Whiteboard não inclui projetor: ele inclui, entre outros ítems, um

<sup>1</sup><http://smarttech.com/smartboard>. Último acesso: 31 de julho de 2013.

<sup>2</sup>[https://en.wikipedia.org/wiki/Smart\\_Board](https://en.wikipedia.org/wiki/Smart_Board). Último acesso: 31 de julho de 2013.

<sup>3</sup>Software Development Kit.

<sup>4</sup>[http://produto.mercadolivre.com.br/MLB-498146708-lousa-smartboard-800-completa-imperdivel-\\_JM](http://produto.mercadolivre.com.br/MLB-498146708-lousa-smartboard-800-completa-imperdivel-_JM). Último acesso: 31 de julho de 2013.

<sup>5</sup><http://sen.smarttech.com/Collaborate/Support-forums.aspx?g=posts&t=103>. Último acesso: 31 de janeiro de 2013.

receptor magnético e uma *stylus* com três botões que funciona como *mouse*. Apesar de não termos encontrado um SDK específico, acreditamos que a atividade do aparelho é convertida em eventos de *mouse*. O produto requer o sistema operacional Windows e não permite *multitouch*. Lançado em 2011, ele parece ter sido descontinuado pela HP. Pelo menos o preço é menos salgado: cerca de mil dólares<sup>6</sup> <sup>7</sup> <sup>8</sup>.

Podemos, ainda, utilizar tecnologias de video-games para construir uma superfície interativa. De fato, o advento de sensores de profundidade de baixo custo possibilitou a detecção de eventos de toque em superfícies comuns. Usando um Microsoft Kinect, Wilson [Wil10] descreveu uma técnica de processamento de imagens que detecta toques numa mesa não instrumentalizada. Apesar de os resultados não serem tão acurados quanto os de tecnologias *multitouch* capacitivas, a abordagem é satisfatória para variadas aplicações (com o benefício adicional de que o espaço físico fora da superfície também pode ser explorado). Xiao, Harrison e Hudson aplicaram uma técnica similar no WorldKit [XHH13], uma solução que transforma superfícies do dia-a-dia em espaços interativos.

Neste projeto de mestrado, decidimos usar um sensor Kinect para construir a superfície interativa. Como o aparelho inclui um sensor de profundidade e uma câmera colorida (ver Fig. 4.1), podemos combinar esses recursos e propor soluções computacionais para reconhecer quais são os objetos que tocam a superfície. Vantagens do uso do Kinect incluem:

- facilidade de transporte e de instalação;
- compatibilidade com Windows, GNU/Linux e MacOS;
- disponibilidade de bibliotecas livres que fornecem dados do sensor;
- ampla disponibilidade no varejo, não sendo restrito a pesquisadores ou empresas;
- baixo custo: cerca de 350 a 500 reais no comércio local (no 1º semestre de 2015);
- possibilidade de explorar a área fora da superfície, permitindo interações em 3D.

Como desvantagens, citamos:

- problemas de oclusão (o usuário deve evitar entrar na frente do sensor);
- acurácia inferior se comparado a tecnologias multi-toque capacitivas;
- *framerate* pequeno, prejudicando interações mais “velozes”;
- área interativa menor que as oferecidas por alternativas comerciais.

Quanto à área interativa, dado o campo de visão do Kinect (ver 4.2), se colocarmos o sensor a uma distância  $d = 80\text{cm}$  da superfície, perpendicularmente, teremos uma área interativa teórica de  $(2dtg(57^\circ/2)) \times (2dtg(43^\circ/2)) \approx 86,87\text{cm} \times 63,03\text{cm}$ . Adicionalmente, se forçarmos uma proporção de tela *widescreen* de 16 : 10, teremos uma diagonal  $D$  teórica medindo pouco mais de 40'':

$$D = \sqrt{(2dtg(57^\circ/2))^2 + \left(\frac{10}{16}(2dtg(57^\circ/2))\right)^2} \approx 102,44\text{cm} > 40''$$

Na prática, como também dependemos de posicionamento do projetor, do Kinect, e de área livre no laboratório, teremos uma área interativa menor. Entretanto, apesar das limitações, consideramos o Kinect uma alternativa adequada para fins de prototipação.

<sup>6</sup><http://www.amazon.com/HP-A0X37AA-ABA-Pocket-Whiteboard/dp/B006F1VM28/>. Último acesso: 31 de julho de 2013.

<sup>7</sup><http://www.cdw.com/shop/products/HP-Pocket-Whiteboard-interactive-whiteboard/3045600.aspx>. Último acesso: 31 de julho de 2013.

<sup>8</sup><http://bestpriceinusa.com/HP-Pocket-Whiteboard-interactive-white-A0X37AA-ABA/p?p=235749>. Último acesso: 31 de julho de 2013.

## 4.2 Microsoft Kinect

Lançado em 2010 pela Microsoft, o Kinect é um sensor de movimentos originalmente projetado para servir de controle para jogos do console Xbox 360. Tendo sido um enorme sucesso de vendas<sup>9</sup>, o dispositivo trouxe ao universo dos games o mote “você é o controle”<sup>10</sup>. Por meio de movimentos corporais, gestos e voz, tornou-se possível ao usuário desfrutar de seus jogos sem a necessidade de qualquer controle físico. Além disso, o impacto do periférico não se limitou ao console da empresa. Como ele pode ser conectado a um PC via porta USB, suas características inovadoras aliadas a seu baixo custo atraíram pesquisadores de áreas como: robótica [RBTH10], jogos digitais [DP11], realidade aumentada [XHH13] e IHC [Wil10].

O Kinect inclui: microfones, um motor, uma câmera RGB, uma câmera infravermelha e um projetor de luz também infravermelha<sup>11</sup>. Uma das qualidades importantes do dispositivo é a habilidade de detectar profundidade: o emissor de luz projeta um padrão estruturado de pontos no ambiente. Tal padrão, cujo modelo é conhecido, é observado pela câmera infravermelha. Com isso, uma tecnologia proprietária da empresa israelense PrimeSense<sup>12</sup> é capaz de gerar um mapa de profundidade da cena. De acordo com Zhang [Zha12], apesar de os passos exatos envolvidos no processo não terem sido disponibilizados ao público, notamos que se soubermos cruzar um ponto na imagem capturada com um ponto no padrão projetado, podemos reconstruí-lo em 3D usando triangulação [Zha12]. Adicionalmente, por usar infravermelho, observamos que o sensor não funciona bem em ambientes abertos e não detecta corretamente a profundidade de objetos brilhantes.

O website msdn (Microsoft Developer Network) inclui algumas especificações do sensor de profundidade<sup>13 14</sup>:

Característica	Especificação
Campo de visão	57° horizontal x 43° vertical
Resolução	640x480
Frame rate	30 Hz
Depth range	de 80cm até 4m, aproximadamente

**Tabela 4.1:** especificações do sensor de profundidade.

Apesar de a resolução nominal do sensor de profundidade ser 640x480, sua resolução efetiva é menor. Segundo Kreylos<sup>15</sup>, a câmera infravermelha, ao capturar o padrão de pontos que foi projetado, só pode calcular a profundidade dos pixels que correspondam a algum dos pontos infravermelhos. Como tais pontos precisam ser espaçados (para que sejam distinguidos pelo sistema), apenas uma fração dos pixels da câmera infravermelha verá os pontos luminosos. Assim, a profundidade dos outros pixels deverá ser estimada via interpolação.

Ainda que o *depth range* nominal do Kinect comece em 80cm, na prática verificamos que o sensor continua funcionando a distâncias um pouco menores (por volta de 70cm). Segundo a Microsoft, o Kinect pode detectar a profundidade de artefatos cuja distância esteja no intervalo especificado na tabela 4.1. O Kinect for Windows, versão alternativa projetada para PCs, inclui o *near mode*: em tal modo, o intervalo válido é de 40cm a 3m, aproximadamente.

Cada pixel do sensor de profundidade, representado por um valor de 11 bits (disparidade), não é função linear/afim da distância. No trabalho de Herrera *et al.* [HKh11], a relação entre a distância real  $z_d$  e a disparidade  $d$  foi modelada de acordo com a equação  $z_d = \frac{1}{\alpha \cdot d + \beta}$ , onde  $\alpha$  e  $\beta$  são constantes determinadas via calibração.

<sup>9</sup> [http://www.huffingtonpost.com/2011/03/09/microsoft-kinect-fastest-selling-consumer-electronics\\_n\\_833706.html](http://www.huffingtonpost.com/2011/03/09/microsoft-kinect-fastest-selling-consumer-electronics_n_833706.html). Último acesso: 3 de junho de 2013.

<sup>10</sup> <http://www.xbox.com/pt-BR/Kinect/Home-new>. Último acesso: 3 de junho de 2013.

<sup>11</sup> Trata-se de um emissor NIR: *near infrared*.

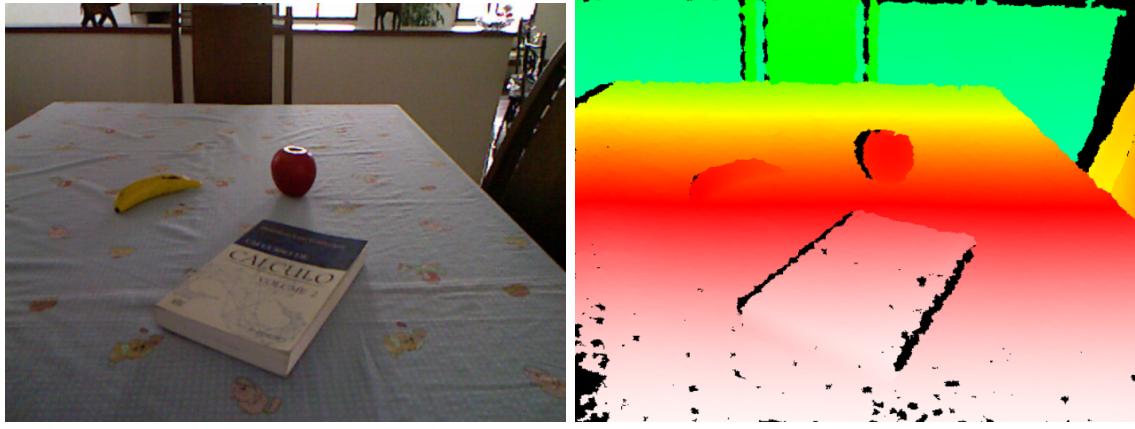
<sup>12</sup> <http://www.primesense.com/>. Último acesso: 3 de junho de 2013.

<sup>13</sup> <http://msdn.microsoft.com/en-us/library/jj131033.aspx>. Último acesso: 3 de junho de 2013.

<sup>14</sup> [http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth\\_Ranges](http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth_Ranges). Último acesso: 3 de junho de 2013.

<sup>15</sup> <http://doc-ok.org/?p=584>. Último acesso: 4 de junho de 2013.

A câmera colorida do Kinect apresenta a resolução 640x480, fornecendo 30 quadros por segundo. É possível atingir resoluções maiores, embora isso provoque queda brusca no *frame rate*. Cada pixel é representado por 32 bits, dos quais apenas 24 são utilizados. Além disso, como a geometria entre as duas câmeras é fixada, pode-se, via software, alinhar suas respectivas imagens por meio de calibração [HKh11].



**Figura 4.1:** Exemplos de imagens fornecidas pelo Kinect.

Apesar de a resolução do Kinect ser um tanto limitada para nossos propósitos, notamos que o Kinect for Xbox One, nova versão do aparelho, inclui uma câmera *time-of-flight* e uma câmera colorida de alta resolução (1080p). O novo Kinect também é mais acurado do que seu predecessor<sup>16</sup>.

Adicionalmente, temos o motor do Kinect, capaz de girar o aparelho na vertical (*tilt*) em  $\pm 27^\circ$ . Os microfones, que são quatro, estão distribuídos na parte inferior do aparelho e permitem interação por voz.

Finalmente, observamos que o Kinect é capaz de rastrear pessoas (*skeleton tracking*), fornecendo ao programador a posição tridimensional de certas partes do corpo. O sensor também pode rastrear a pose da cabeça e ler expressões faciais. Tais funcionalidades, discutidas por Zhang [Zha12], não são exploradas neste trabalho.

### 4.3 OpenNI

Segundo o site oficial<sup>17</sup>, OpenNI (*Open Natural Interaction*) é uma organização sem fins lucrativos formada em 2010 para “promover e padronizar a compatibilidade e interoperabilidade de *Natural Interaction* (NI), dispositivos, aplicações e middleware”. Um dos produtos disponibilizados pela organização é o arcabouço OpenNI, que fornece uma API<sup>18</sup> voltada à criação de programas envolvendo interação natural. Tal arcabouço é software livre (licença Apache) e multiplataforma: há versões para Windows, GNU/Linux e MacOS. Para utilizar o Kinect, deve-se instalar um *driver* disponibilizado pela empresa PrimeSense<sup>19</sup>.

Além do OpenNI, há outras tecnologias que permitem acessar o Kinect via código: pode-se usar tanto o SDK da Microsoft, dotado de grande variedade de recursos sofisticados, quanto o libfreenect<sup>20</sup>, criado por entusiastas na Internet. Dentre essas três opções, o SDK da Microsoft é o mais completo, porém é restrito a Windows. Como neste trabalho só estamos interessados nas imagens de cor e de profundidade, acabamos optando pelo OpenNI, que também roda em Linux.

<sup>16</sup>[http://en.wikipedia.org/wiki/Kinect#Kinect\\_for\\_Xbox\\_One](http://en.wikipedia.org/wiki/Kinect#Kinect_for_Xbox_One). Último acesso: 4 de junho de 2013.

<sup>17</sup><http://www.openni.org/about/>. Último acesso: 5 de junho de 2013.

<sup>18</sup>*Application Programming Interface*.

<sup>19</sup><https://github.com/avin2/SensorKinect>. Último acesso: 5 de junho de 2013.

<sup>20</sup>[http://openkinect.org/wiki/Main\\_Page](http://openkinect.org/wiki/Main_Page). Último acesso: 5 de junho de 2013.

## 4.4 OpenCV

Em 1999, na Intel, uma equipe de desenvolvedores liderada pelo pesquisador Gary Bradski iniciou o projeto OpenCV (*Open Source Computer Vision Library*). Trata-se de uma biblioteca livre, disponibilizada sob licença BSD, que traz funcionalidades relacionadas a visão e aprendizagem computacional. Lançada para o público no final de 2006, a versão 1.0 do OpenCV, que foi escrita em linguagem C, incluía rotinas de: processamento de imagens, cálculos envolvendo matrizes e álgebra linear, estruturas de dados, algoritmos diversos de visão e aprendizagem computacional, etc. Em 2009 foi liberada a versão 2.0, desenvolvida em C++. De acordo com a Wikipedia, o OpenCV é usado em diversas áreas de aplicação: realidade mista, IHC, robótica, etc<sup>21</sup>. Adicionalmente, segundo o site oficial<sup>22</sup>, a biblioteca traz interfaces para C, C++, Python e Java, podendo ser utilizada em ambientes Windows, GNU/Linux, MacOS, iOS e Android.

## 4.5 openFrameworks

openFrameworks é um arcabouço C++ que se propõe a facilitar a atividade de experimentação nas áreas de interatividade e multimídia. Desenvolvido inicialmente por Zach Lieberman, Theo Watson, Arturo Castro e Chris O’Shea, o arcabouço conta com colaboradores da Parsons School of Design, do Hangar Center for the Arts e do MediaLabMadrid, bem como com a ajuda de entusiastas da Internet [Nob12]. O desenvolvimento do projeto, iniciado em 2005 no programa de design e arte da Parsons School of Design, veio como resposta a uma deficiência que os fundadores encontraram: havia muitas bibliotecas C++, completamente separadas umas das outras, que poderiam ser usadas como “peças” em projetos interativos. Tratava-se de bibliotecas relativas a: gráficos, áudio, *playback* de vídeos, fontes, etc. Tal situação criava uma barreira de entrada significativa para quem quisesse rapidamente “experimentar”, isto é, usar o computador como forma de expressão. De acordo com o site oficial<sup>23</sup>, o openFrameworks funciona como uma “cola de propósito geral”, agrupando várias tecnologias conhecidas (OpenGL, GLUT, FreeImage, FreeType, OpenAL, OpenCV, etc.) e fornecendo ao usuário uma maneira intuitiva e prática para a criação de projetos interativos.

## 4.6 HTML5

HTML5 é a quinta versão da Linguagem de Marcação de Hipertexto (HTML). Ao fornecer uma plataforma abrangente de desenvolvimento para páginas da Web, ela promete eliminar a necessidade de instalação de *plug-ins* de terceiros tais como Java ou Flash nos navegadores. Além de introduzir elementos semânticos, o HTML5 traz consigo um rico conjunto de funcionalidades multimídia (áudio, vídeo, gráficos 2D e 3D) que permite o desenvolvimento de *web apps*<sup>24</sup> complexos e de fácil portabilidade. As capacidades interativas do HTML5 receberam atenção massiva da mídia quando, em 2010, Steve Jobs - o então CEO da Apple - concluiu que o “Flash não é mais necessário para assistir vídeos ou consumir qualquer tipo de conteúdo para a Web” e que novos padrões, tais como o HTML5, dominariam o mercado<sup>25</sup>.

As ricas funcionalidades multimídia da linguagem associadas à ampla disponibilidade de bibliotecas JavaScript e de APIs para o desenvolvimento de *web apps* diversas fazem dela uma escolha interessante para se prototipar, com facilidade, aplicações para a Superfície Mágica. Pode-se comunicar dados entre um programa C++ e uma aplicação HTML5 por meio de WebSockets, tecnologia que permite comunicação *full-duplex* por meio de uma única conexão TCP<sup>26</sup>.

<sup>21</sup><http://pt.wikipedia.org/wiki/OpenCV>. Último acesso: 4 de junho de 2013.

<sup>22</sup><http://opencv.org>. Último acesso: 4 de junho de 2013.

<sup>23</sup><http://www.openframeworks.cc/about/>. Último acesso: 4 de junho de 2013.

<sup>24</sup>Programas de computador que rodam num navegador de internet.

<sup>25</sup><http://en.wikipedia.org/wiki/HTML5>. Último acesso: 30 de maio de 2015.

<sup>26</sup><http://en.wikipedia.org/wiki/WebSocket>. Último acesso: 30 de maio de 2015.



# Capítulo 5

## Desenvolvimento do sistema

### 5.1 Visão geral

Neste capítulo, descrevemos em maiores detalhes o desenvolvimento do projeto proposto. Como explicado na seção 1.2, o protótipo do sistema **Superfície Mágica** permite interação por meio dos seguintes mecanismos:

1. canetas coloridas;
2. varinha mágica;
3. apagador;
4. dedo.

Do ponto de vista de implementação, temos duas partes principais:

1. a transformação de uma superfície comum num espaço interativo multi-toque (ver Figura 5.1a) no qual se possa diferenciar entre toques de: dedos de mão, canetas coloridas e apagador;
2. o desenvolvimento de uma “varinha mágica” para interação em três dimensões (vide Figura 5.1b).



**Figura 5.1:** (a) Uma parede se torna interativa. (b) Um usuário portando uma varinha mágica.

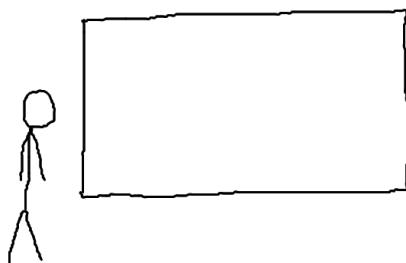
Uma vez criada a infraestrutura supracitada, podemos utilizar a superfície e suas ferramentas para explorar diferentes formas de diálogo entre o ser humano e o mundo digital. Antes dos detalhes técnicos, veremos alguns cenários de uso.

## 5.2 Cenários de uso

Por meio de quadrinhos, os personagens Pedro e Bruno nos ajudarão a mostrar alguns cenários de uso do sistema.

### 5.2.1 Explorando a superfície de desenho

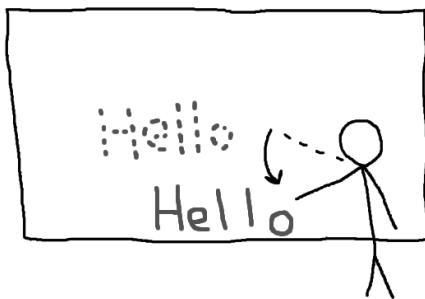
Uma parede, tornada interativa, aceita desenhos à mão livre. Por ser digital, ela também oferece um espaço “infinito” para expressão.



Pedro vai testar sua novíssima Superfície Mágica!



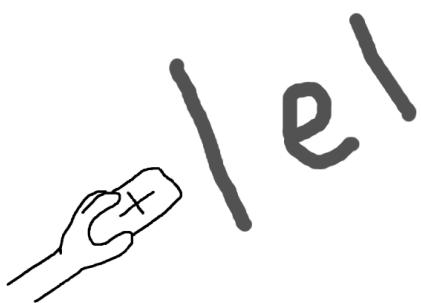
Com a caneta, ele escreve algo



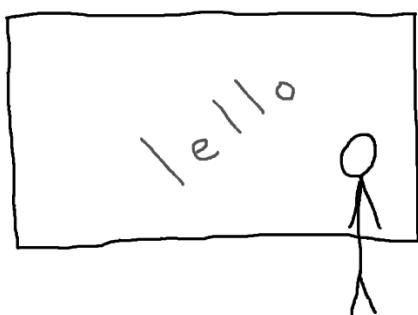
e, com o dedo indicador, arrasta todo o conteúdo.



Também é possível redimensionar (escalar) o conteúdo.



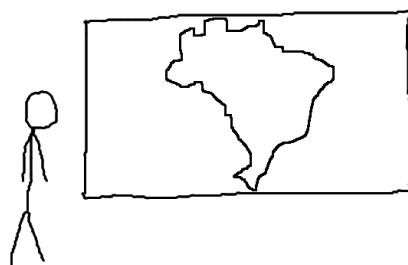
Com o apagador, Pedro exclui parte do conteúdo



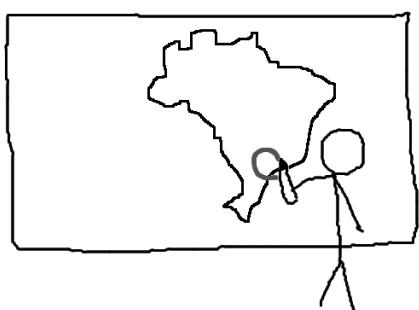
e vê o resultado.

### 5.2.2 Explorando mapas

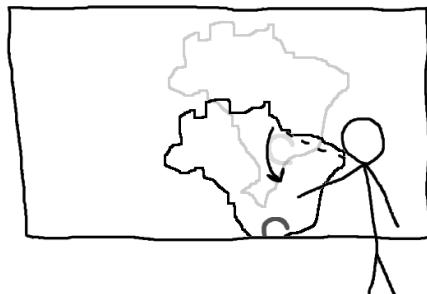
Pedro está estudando para uma prova de Geografia. Entretanto, ele está com dificuldades para associar diferentes regiões do território brasileiro a seus respectivos climas (subtropical, tropical de altitude, tropical, equatorial, semi-árido). Ele decide então recorrer à Superfície Mágica.



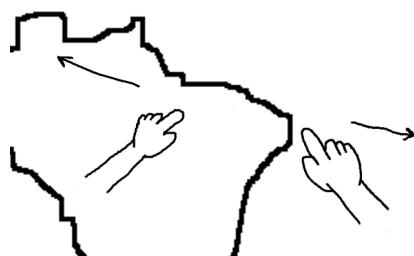
Pedro lança o aplicativo de mapas.



Com a caneta, ele destaca uma região



e, com o dedo indicador, arrasta o mapa.



Pedro aumenta o *zoom* na região do sertão nordestino,



pede para o sistema exibir informações climáticas

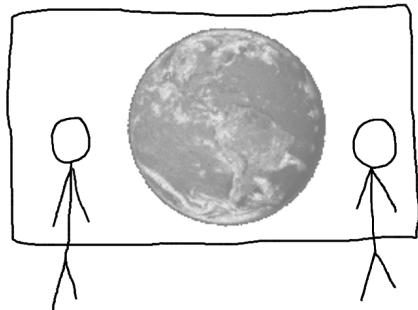


e vê o resultado.

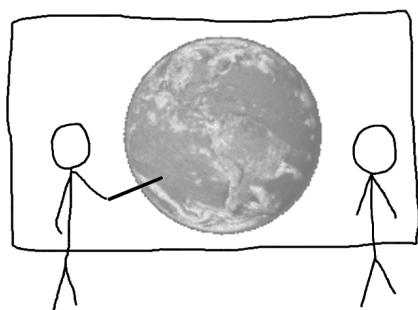
### 5.2.3 Navegação em 3D

Pedro e Bruno estão estudando sobre a floresta Amazônica, cuja vegetação é densa, latifoliada, higrófita<sup>1</sup> e bastante diversificada. Ambos estão entediados. “*Falta interatividade!*”, exclama Bruno. É quando Pedro sugere fazer um *tour* virtual pela Amazônia, sobrevoando a região. Bruno topa a ideia.

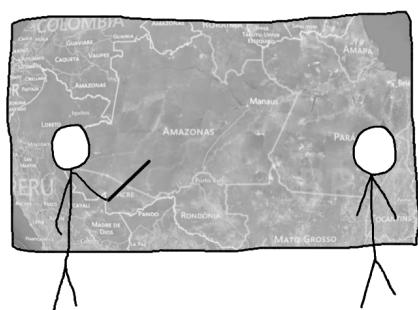
<sup>1</sup>Uma planta higrófita está adaptada a ambientes com muita umidade.



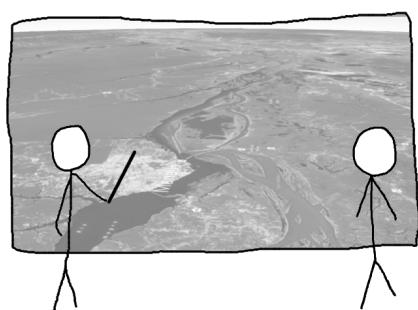
Pedro lança o simulador 3D de exploração do globo terrestre.



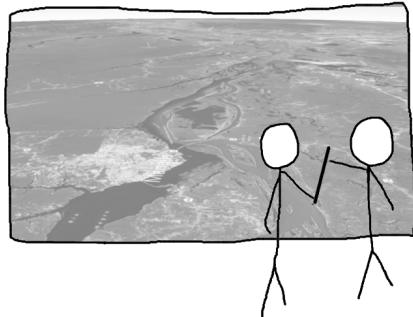
Com a varinha mágica, ele aponta para a região da Amazônia



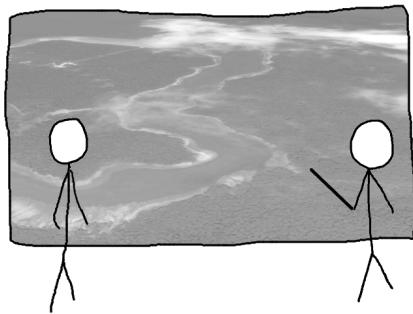
e dá um comando de zoom.



Pedro instrui o simulador a sobrevoar a região



e cede a varinha mágica a Bruno.



Bruno utiliza a varinha para navegar pela Amazônia.

## 5.3 Superfície multi-toque

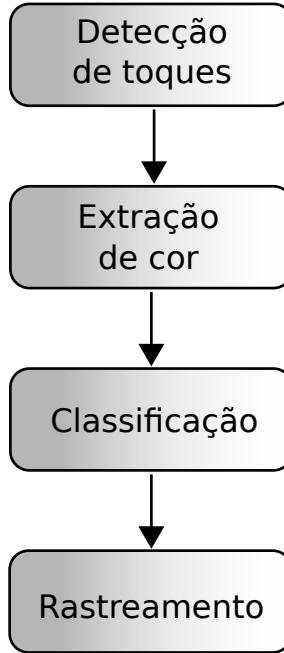
### 5.3.1 Visão geral

Como visto no capítulo 4, um sensor Kinect é usado para transformar uma superfície comum, não instrumentalizada, num espaço interativo multi-toque. A superfície também é capaz de reconhecer quais objetos a tocam (dedos da mão, canetas ou apagador). Conforme mostrado no diagrama 5.2, nossa abordagem é dividida nas seguintes etapas: detecção dos toques, extração de cor, classificação e rastreamento. Exploraremos cada uma delas em detalhes, para em seguida falarmos sobre a instalação física da Superfície Mágica.

### 5.3.2 Detecção de toques

A técnica usada na etapa de detecção de toques é baseada nos trabalhos de Wilson [Wil10] e Xiao, Harrison e Hudson [XHH13], com a diferença que aplicamos uma máscara de modo a restringir a detecção de eventos de toque a uma superfície plana (como uma parede, uma mesa, ou mesmo uma tela LCD comum). O método é resumido da seguinte maneira:

1. Para representar a área interativa na superfície, o usuário escolhe manualmente quatro pontos na imagem de profundidade. Esse processo define uma máscara, que será aplicada a imagens de profundidade subsequentes;
2. Pesquisas mostraram que as estimativas de profundidade fornecidas pelo Kinect são estáveis [Wil10]. Isto posto, quando a superfície estiver vazia, coletamos alguns *snapshots*  $\{S_1, \dots, S_n\}$  consecutivos da imagem de profundidade (novo sistema usa  $n = 90$ ). Denotando o valor da profundidade do  $j$ -ésimo pixel de  $S_i$  por  $d_{ij}$ , sejam  $\mu_j$  e  $\sigma_j$  a média e o desvio-padrão de  $\{d_{1j}, \dots, d_{nj}\}$ , respectivamente. Diremos que o modelo de fundo é a coleção de todos os  $\mu_j$ . O conjunto de todos os  $\sigma_j$  é usado como um perfil de ruído;



**Figura 5.2:** Superfície multi-toque: etapas.

3. Uma subtração de fundo entre o modelo e a imagem de profundidade mais recente fornecida pelo Kinect é efetuada. Como resultado, teremos uma imagem contendo os objetos que estão acima da superfície;
4. Apenas os pixels cuja profundidade estejam entre os valores  $d_{min}$  e  $d_{max}$ , determinados experimentalmente, são considerados para processamento posterior;
5. Pixels cuja profundidade esteja a uma distância menor do que três desvios-padrão da média são descartados. Adicionalmente, operações morfológicas de erosão e dilatação são efetuadas, de modo a remover quaisquer artefatos indesejados que ainda restem;
6. Finalmente, os *blobs* da imagem resultante são identificados por meio de análise de componentes conexas. *Blobs* cuja área esteja fora de um intervalo pré-determinado são descartados.

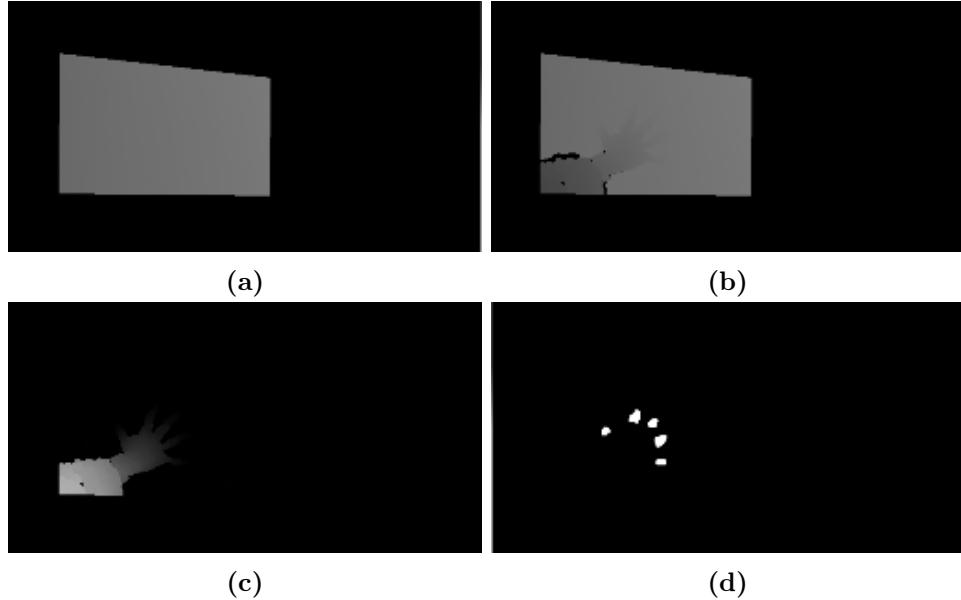
O centro de massa das componentes conexas resultantes indicará um ponto de toque. Além disso, suas coordenadas  $(x, y)$  na imagem de profundidade serão transformadas para um espaço  $[0, 1] \times [0, 1]$  por meio de uma matriz de homografia, computada usando um procedimento padrão de calibração de quatro pontos [BK08].

### 5.3.3 Extração de cor

Apesar de podermos usar o sensor de profundidade para simular um dispositivo multi-toque, nosso sistema também precisa reconhecer quais exatamente são os objetos que tocam a superfície. Para isso, podemos coletar dados de cor, aproveitando a câmera RGB já embutida no Kinect. Na implementação atual, antes de extrairmos quaisquer dados de cor, as imagens RGB e de profundidade são alinhadas.

Dado um ponto de toque  $s' = [s'_x \ s'_y]^t$  na imagem de profundidade, usamos a abordagem descrita por Kramer *et al.* [KPH<sup>+</sup>12] para computar sua posição no mundo físico  $s = [s_x \ s_y \ s_z]^t$ . Então, para um raio  $r$  pré-definido, consideramos a seguinte esfera:

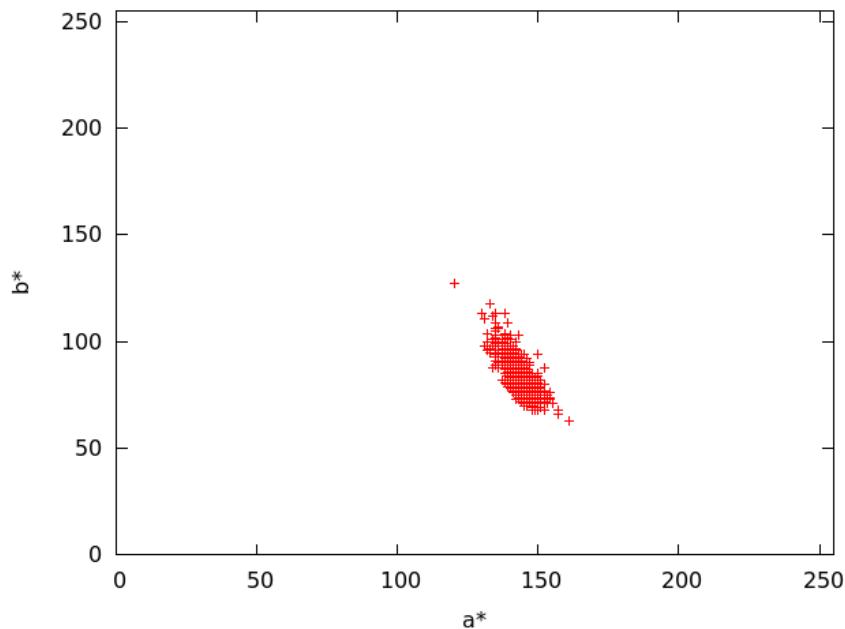
$$\gamma : (x - s_x)^2 + (y - s_y)^2 + (z - s_z)^2 = r^2$$



**Figura 5.3:** (a) Modelo de fundo. (b) Uma imagem de profundidade mostrando uma mão. (c) Artefatos sobre a superfície. (d) Blobs resultantes.

Primeiramente, coletamos as cores de uma amostra de pontos em  $\gamma$ . Então, os valores de cromacidade de cada ponto são computados no espaço CIELAB, gerando um conjunto  $C$  de pares de números. Tal conjunto representa, aproximadamente, a ponta do artefato físico que toca a superfície.

Para ilustrar nossa abordagem, coletamos essas amostras durante alguns quadros de um vídeo no qual uma caneta riscava a superfície. A Figura 5.4 mostra o *cluster* cromático dessa caneta.



**Figura 5.4:** Cluster cromático.

### 5.3.4 Classificação

Para classificarmos  $C$ , numa primeira etapa cada par  $c_i \in C$  de valores cromáticos é analisado individualmente. Baseado na observação de que o *cluster* cromático tem aproximadamente o formato de uma elipse, decidimos estender o *elliptical boundary model* de Lee e Yoo [LY02]. Tal modelo,

proposto inicialmente como um classificador binário de pele, foi por nós ampliado de modo a abrigar mais categorias.

Para cada possível categoria  $j$ , o método classifica o par  $c_i$  como pertencente à categoria  $j$  se, e somente se,

$$(c_i - \phi_j)^t \Lambda_j^{-1} (c_i - \phi_j) < \theta$$

onde  $\theta$  é um escalar especificado empiricamente e  $\phi_j \in \mathbb{R}^2$  e  $\Lambda_j \in \mathbb{R}^{2 \times 2}$  são os parâmetros do modelo  $j$ , computados de acordo com [LY02].

Para efetuar o treinamento dos modelos, em primeiro lugar filtramos as amostras de treinamento de modo a remover 5% dos pares cromáticos de baixa frequência. Em seguida, para cada categoria  $j$ , dadas as sequências filtradas de treinamento  $P_j = \{p_{j1}, p_{j2}, \dots, p_{jn_j}\}$  computamos o conjunto  $Q_j = \{q_{j1}, q_{j2}, \dots, q_{jm_j}\}$  como tendo os mesmos pontos que  $P_j$ , mas sem repetição de elementos. Finalmente, estimamos os parâmetros dos modelos de acordo com as equações:

$$\begin{aligned} \phi_j &= \frac{1}{m_j} \sum_{i=1}^{m_j} q_{ji} \\ \Lambda_j &= \frac{1}{n_j} \sum_{i=1}^{m_j} f_{ji} (q_{ji} - \mu_j) (q_{ji} - \mu_j)^t \end{aligned}$$

onde:

- $m_j = |Q_j|$  é a quantidade de pares de treinamento  $p_{ji}$  distintos;
- $f_{ji}$  é a frequência absoluta de amostras de treinamento em  $P_j$  cuja cromacidade é  $q_{ji}$ ;
- $n_j = \sum_{i=1}^{m_j} f_{ji}$  é o tamanho de  $P_j$ ;
- $\mu_j = \frac{1}{n_j} \sum_{i=1}^{m_j} (f_{ji} \cdot q_{ji})$  é a média dos pontos em  $P_j$ .

Diferentemente de uma gaussiana multivariável, o *elliptical boundary model* leva em conta a possibilidade de o *cluster* cromático ser assimétrico em relação ao pico de densidade.

Definindo  $M_j$  como o conjunto de pares cromáticos  $c_i \in C$  pertencentes à categoria  $j$ , bem como o volume com sinal  $v_j$  como sendo o determinante  $|\Lambda_j|$ , classificamos o *cluster*  $C$  de acordo com a densidade:

$$\text{classe}(C) = \operatorname{argmax}_j \frac{|M_j|}{|v_j|}$$

Se  $|M_j| = 0$  para todo  $j$ , então  $C$  é classificado como dedo de mão (pois cor de pele apresenta maior variabilidade que os *clusters* correspondentes às poucas canetas coloridas que usamos).

### 5.3.5 Rastreamento

A última etapa consiste em rastrear pontos de toque que se movem pela superfície. O algoritmo de rastreamento adota uma estratégia gulosa para associar pontos de toque do quadro atual e do anterior. Como discutido em Kramer *et al.* [KPH<sup>+</sup>12], a cada quadro da aplicação, novos IDs são associados a todos os pontos de toque. Então, a nova localização de cada ponto de toque do quadro anterior é estimada usando posição e velocidade anteriores. A nova localização é encontrada nas proximidades, dentro de um raio pré-fixado, recebendo o mesmo ID e não sendo mais considerada como candidata a ser associada a outros pontos. Na prática, o algoritmo termina atribuindo novos IDs a todos os novos pontos de toque.

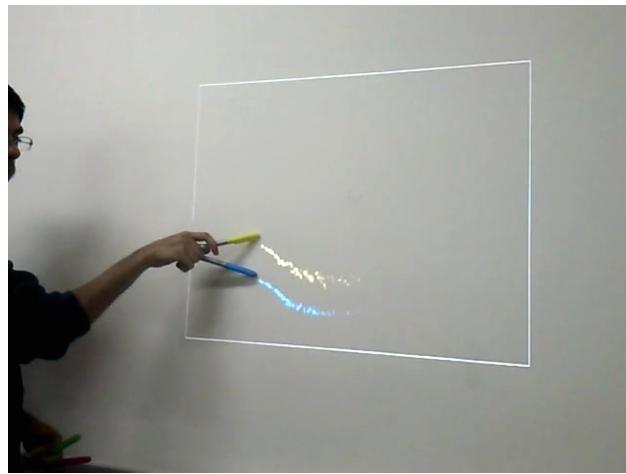
Adicionalmente, fizemos um filtro no qual pontos de toque são removidos da memória apenas se estiverem ausentes por pelo menos 6 quadros.

É importante ressaltar que, apesar de robusto na prática, o classificador descrito na etapa anterior pode errar. Em nossos testes com o Kinect, observamos um pequeno atraso entre as imagens

RGB e de profundidade. Se, com uma caneta, o usuário riscar a superfície interativa de maneira brusca, o sensor de profundidade poderá capturar o toque, mas a etapa de extração de cores fornecerá pontos da superfície (ou uma mistura entre superfície e caneta, devido a um efeito “blur”). Como resultado, teremos uma classificação incorreta. Acreditamos que este fato está relacionado ao *framerate* do sensor: 30 quadros por segundo.

Para contornar a situação de possíveis classificações incorretas atrapalhando a interação, adotamos a seguinte heurística: se um ponto de toque  $s$  for classificado como  $j$  durante pelo menos  $k$  quadros consecutivos,  $s$  passará a ser sempre classificado como  $j$ . Em nosso sistema, adotamos  $k = 30$ .

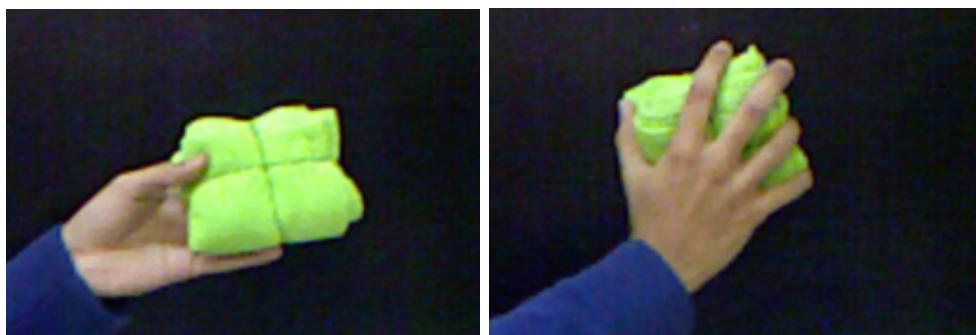
A Figura 5.5 mostra o sistema reconhecendo toques de canetas de cores distintas. Notamos que, uma vez treinados os modelos de cores, a precisão do reconhecimento depende de fatores como: iluminação ambiente, brilho do projetor, etc. No laboratório, controlamos essas variáveis e o protótipo acaba funcionando satisfatoriamente. Entretanto, idealmente as canetas teriam um mecanismo de iluminação ativa.



**Figura 5.5:** Reconhecendo canetas de cores diferentes.

### 5.3.6 Apagador

Além de detectar toques de mão e de canetas coloridas, o sistema também permite que o usuário apague o que desenhou. O ato de apagar tem sua forma de interação inspirada no contexto da limpeza doméstica. Assim como uma dona de casa usa panos para limpar pias ou vidros, o usuário também utiliza um pano para “limpar” marcações feitas na superfície (ver Fig. 5.6).



(a) Um pano preso com elásticos. (b) Apagador em utilização.

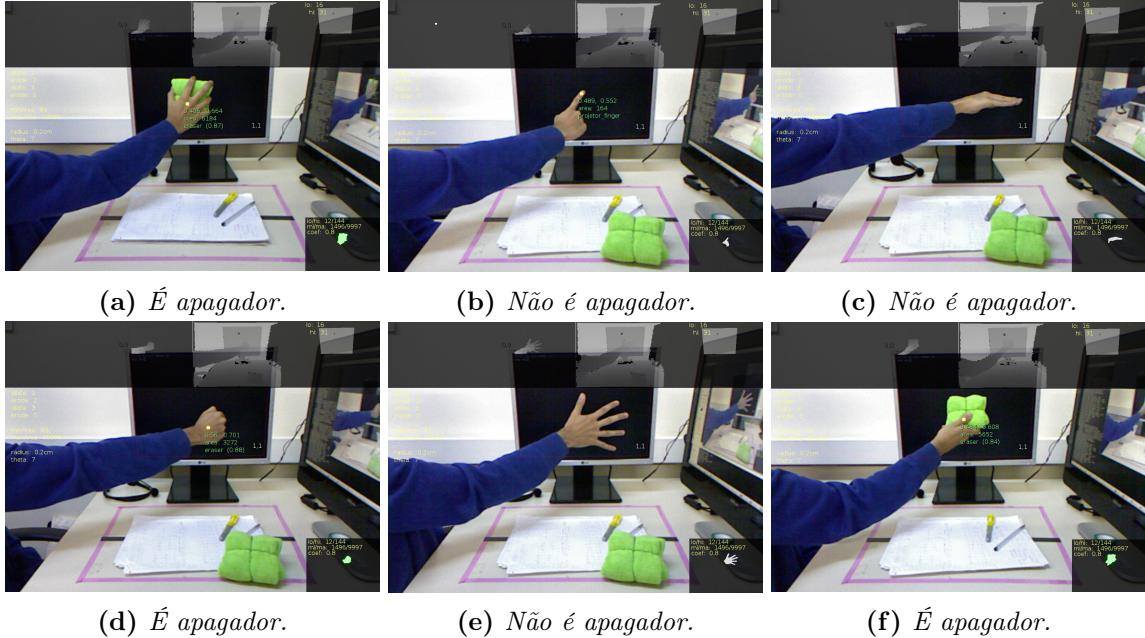
**Figura 5.6:** Apagador.

Para identificar a presença de um apagador, desempenhamos os passos descritos em 5.3.2, terminando com um conjunto de *blobs* que representam artefatos sobrepostos à superfície interativa.

Sejam  $M$  qualquer um desses *blobs* e  $c_M = \frac{1}{|M|} \sum_{p \in M} p$  seu centro de massa. Diremos que  $M$  é um candidato a apagador se, e somente se,

$$\frac{|\{p \in M \mid \|p - c_M\| \leq \sqrt{\frac{|M|}{\pi}}\}|}{|M|} \geq \alpha$$

onde  $\alpha \in [0, 1]$  é um parâmetro empírico. Se  $M$  for candidato a apagador e se não houver detecção de dedos de mão ou de toques de caneta na superfície<sup>2</sup>, então diremos que  $M$  é um apagador.



**Figura 5.7:** Protótipo.

Informalmente,  $\alpha$  codifica a proporção mínima de pontos do *blob* que devem pertencer ao círculo de área  $|M|$  centrado em  $c_M$ . Em nosso sistema, utilizamos  $\alpha$  entre 0,7 e 0,8. A proporção da expressão anterior é computada por meio de um algoritmo de *floodfill*.

Na Figura 5.7, exibimos nosso protótipo reconhecendo (ou não) a presença de um apagador. Por conveniência, nesses testes fizemos um monitor assumir o papel de superfície interativa.

### 5.3.7 Instalação do protótipo

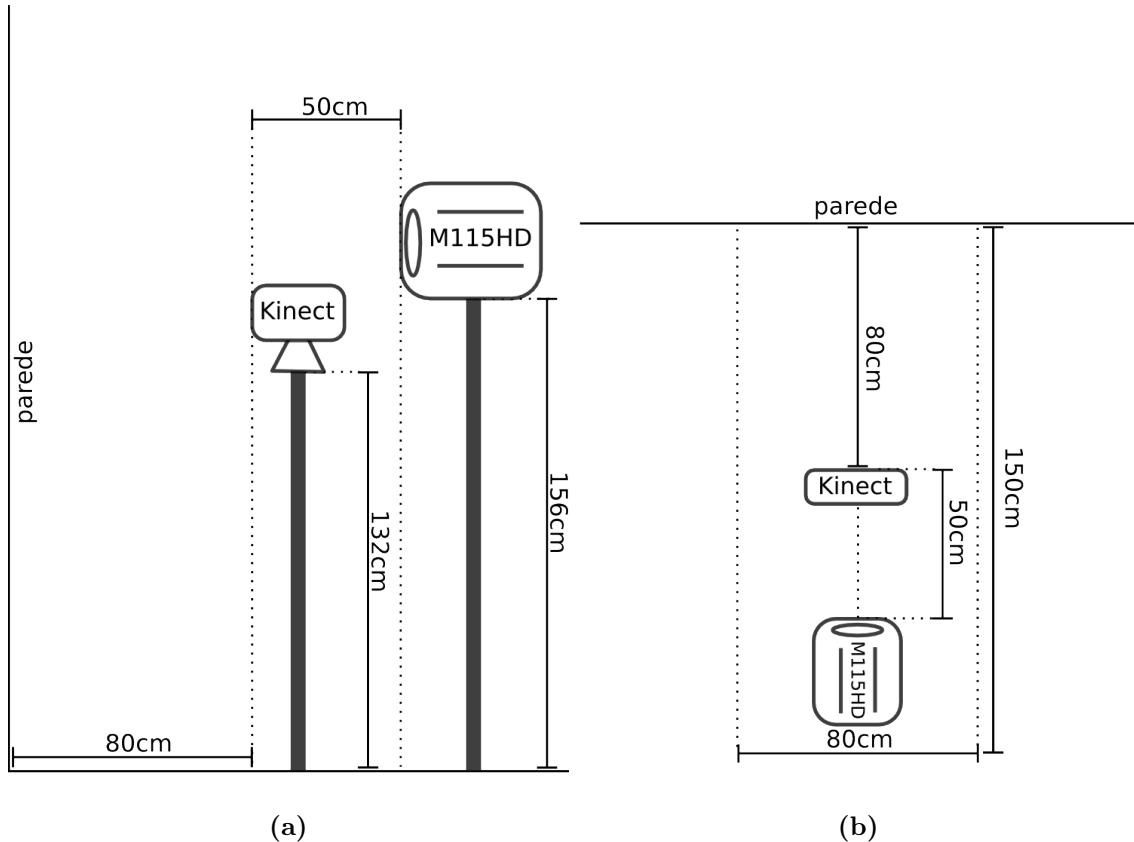
Um dos requisitos do trabalho é a criação de uma superfície interativa portátil e de baixo custo. Idealmente, o aparato responsável por transformar uma superfície comum em interativa deve ser tal que a instalação possa ser feita não apenas em laboratórios de pesquisa, mas também - por exemplo - em ambientes domésticos ou artísticos. Conforme mostrado na Figura 5.8, os seguintes ítems são necessários:

- Um computador munido do nosso software. Em nosso protótipo, é utilizado um PC Dell All-in-one Inspiron 2330. Já utilizamos também um *laptop* Dell Vostro 4350. Ambos executam Ubuntu GNU/Linux;
- Um sensor RGBD. Nossa protótipo utiliza um Microsoft Kinect que, conectado a uma porta USB do computador, permanece apontado para a superfície;
- Um projetor de vídeo. Em nosso protótipo, utilizamos um projetor portátil Dell M115HD.

<sup>2</sup>ou, se houver apenas marcações verdes, removemos todas elas.



**Figura 5.8:** Instalação do protótipo.



**Figura 5.9:** Planta sugerida.

Todos esses equipamentos podem ser encontrados hoje em lojas locais de informática, ou mesmo comprados via Internet. Devido à sua resolução não muito alta (vide 4.2), o Kinect deve ficar próximo à superfície - a cerca de oitenta centímetros de distância - pois queremos detectar até mesmo toques de dedos da mão.

Na Figura 5.9, exibimos uma planta sugerida para a montagem do protótipo<sup>3</sup>. Tal planta se refere à transformação de uma parede num espaço interativo. Devido à presença de mecanismos de calibração, o sistema acomoda variações nas medidas. Em nossos testes, uma instalação similar

<sup>3</sup>omitimos o computador.

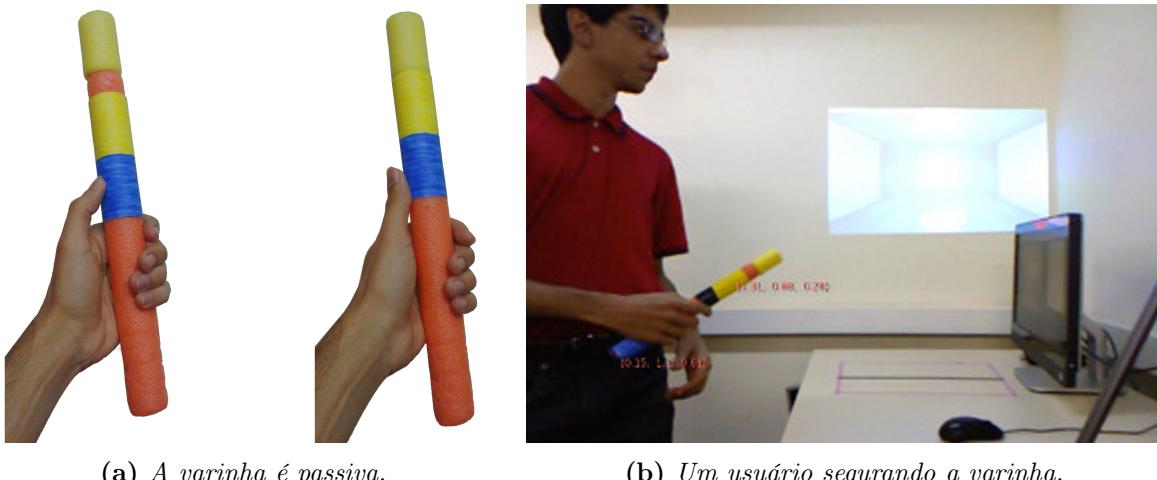
proporcionou uma área interativa *widescreen* de aproximadamente  $80\text{cm} \times 50\text{cm}$  (gerando uma diagonal de  $37''$  em relação a uma proporção de 16 : 10). Ressaltamos aqui a portabilidade da instalação: com um espaço de  $1\text{m} \times 2\text{m} = 2\text{m}^2$ , já se pode montar o protótipo.

Como última observação, notamos que, devido a restrições de implementação, o protótipo desenvolvido neste trabalho de mestrado funcionará apenas em ambientes fechados, livres de luz solar excessiva. Conforme exposto na seção 4.2, o sensor de profundidade do Kinect tem seu funcionamento baseado em luz infravermelha.

## 5.4 Varinha mágica

### 5.4.1 Visão geral

Mágica, como descrita em contos de fantasia, é normalmente codificada em “feitiços” e traduzida em ferramentas específicas [Bin00]. Varinhas, por outro lado, são utilizadas em contextos do mundo real como uma representação de poder [Hil06]. O conceito de varinha mágica, já arraigado no imaginário popular, inspira diversas formas de se interagir com o computador. Ao adotar o papel de “feiticeiro”, o usuário pode usar uma varinha mágica para apontar para lugares, executar gestos e lançar “feitiços” para interagir com o mundo virtual.



(a) A varinha é passiva.

(b) Um usuário segurando a varinha.

**Figura 5.10:** A varinha mágica.

Varinhas encontram aplicações em diversas áreas, tais como: navegação em ambientes virtuais [KPD12, CGVT03, GKR07, CRdS<sup>+</sup>12], casas inteligentes [WS03], educação e arte interativa. Uma varinha mágica poderia ser utilizada num ambiente doméstico para ligar ou desligar uma luz, controlar uma televisão e assim por diante. Um professor poderia usar uma varinha mágica em sala de aula para apontar e manipular um objeto em 3D (ex: uma célula, uma molécula ou uma máquina), de modo a complementar uma explicação. Similarmente, para propósitos de *storytelling*, um artista poderia manipular uma varinha mágica como parte de uma *performance* artística, misturando os mundos real e virtual.

As câmeras RGBD estão se tornando cada vez mais baratas e difundidas. Dentre as alternativas mais recentes, citamos o RealSense<sup>4</sup> da Intel, um sensor RGBD que pode ser embarcado em dispositivos como *laptops*. A potencial ubiquidade de tais sensores no futuro faz deles uma escolha atraente para implementar uma varinha mágica para interação em três dimensões.

A varinha mágica desenvolvida neste trabalho é uma interface de usuário 3D que, além de ser construída de forma barata, utiliza conhecimentos prévios do usuário obtidos na infância (ex: filmes de fantasia). Ao contrário de trabalhos similares, nossa varinha inclui um *switch liga/desliga* para ampliar as capacidades de interação e é rastreada usando apenas uma câmera RGBD, o que a torna

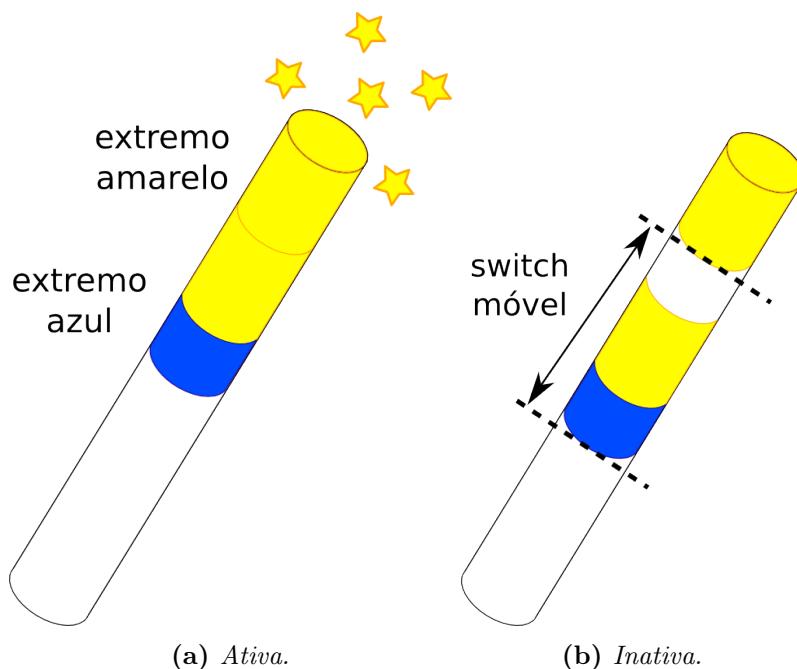
<sup>4</sup><https://software.intel.com/en-us/realsense/>. Último acesso: 11 de março de 2015.

resiliente contra ruído de fundo. Uma técnica de visão computacional para a detecção da varinha também é descrita.

### 5.4.2 Design

Os objetivos de design da varinha mágica são:

1. tanto sua posição como sua orientação devem ser rastreadas no espaço tridimensional;
2. ela deve permitir que usuários apontem para elementos numa tela;
3. ela deve trazer um mecanismo para habilitar a execução de “feitiços” (gestos), dando à varinha seu teor “mágico”;
4. ela deve ser construída facilmente e de forma barata;
5. ela deve ser sem fio.



**Figura 5.11:** Varinha mágica: dois modos de operação.

O design que propusemos para a varinha traz características que permitem que ela seja rastreada por meio de uma câmera RGBD. Começamos com uma haste de plástico de aproximadamente 30cm de comprimento. Marcações coloridas inseridas nos extremos da varinha (veja Figura 5.11a) facilitam seu rastreamento por um sistema de visão computacional. Tal configuração permite a estimativa da pose com 5 graus de liberdade (posição no espaço 3D e dois ângulos de rotação: empinamento e cabeceio). Isto posto, pode-se definir um segmento orientado 3D que parte da cabeça da varinha e tem a mesma direção e orientação da ferramenta. Uma vez que as coordenadas da tela sejam conhecidas, é fácil usar a varinha como um dispositivo de apontamento.

Também projetamos um comutador (*switch*): uma parte móvel colocada próximo a um dos extremos da varinha, útil para indicar quando um feitiço começa e quando ele termina (vide Fig. 5.11b). O comutador pode ser movimentado usando o dedo polegar e funciona como um elemento liga-desliga. Quando ele está sobreposto ao extremo correspondente da varinha (isto é, o extremo que compartilha a mesma cor do comutador), dizemos que a varinha está ativa. Caso contrário, dizemos que a varinha está inativa. Neste trabalho, definimos um feitiço (ou encantamento) como a trajetória desempenhada pela varinha enquanto ela estiver ativa.

O marcador azul ajuda a determinar a orientação da varinha. Ele pode ser colocado próximo ao *switch* (Fig. 5.10a) ou num dos extremos da varinha (Fig. 5.10b). Após testar ambas as configurações, avaliamos que a primeira escolha é melhor tanto para reduzir oclusões quanto para ajudar o usuário a deslocar o comutador.

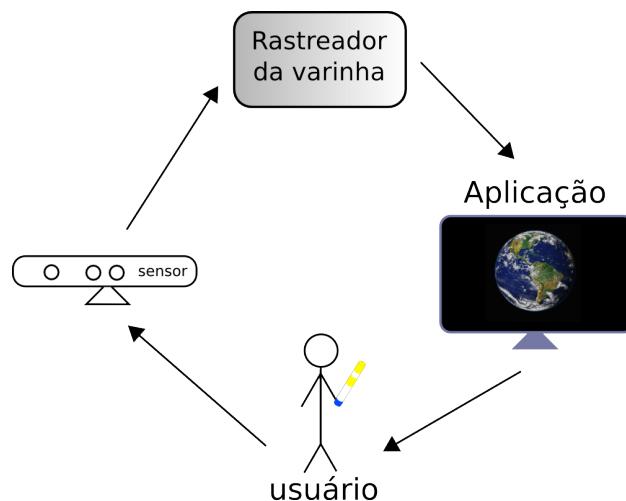
O design da nossa varinha passiva (i.e., sem componentes eletrônicos) torna sua construção fácil, barata e faz dela um dispositivo de interação sem fio. A Figura 5.10a exibe um protótipo.

### 5.4.3 Implementação

#### Visão geral

A implementação computacional da varinha mágica é dividida em duas camadas:

- **Camada sensorial:** uma câmera RGBD captura dados de cor e profundidade do ambiente;
- **Camada de rastreamento:** determina a posição, orientação e estado (ativo ou inativo) da varinha.



**Figura 5.12:** Ciclo envolvendo a varinha mágica.

A Figura 5.12 mostra um ciclo: a camada sensorial (simbolizada pelo sensor RGBD), após captar a interação do usuário, comunica-se com camada de rastreamento. Tal camada passa os dados da varinha para uma aplicação, que responde às ações do usuário e dá a ele o retorno correspondente.

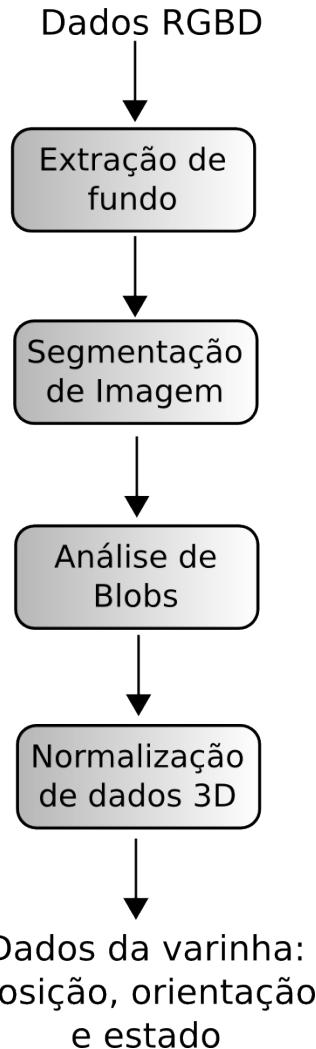
Em relação à camada sensorial, uma câmera Kinect é utilizada. Sua disponibilidade e custo acessível fazem dela uma escolha atraente para implementar a varinha. A camada de rastreamento apresenta desafios técnicos que serão apresentados em mais detalhes.

Como mostrado na Figura 5.13, a partir dos dados RGBD obtidos da camada sensorial, uma sequência de passos é desempenhada de modo a extrair a posição, orientação e estado da varinha.

#### Extração de fundo

Ao contrário de trabalhos similares baseados em rastreamento óptico, rastrear a varinha com um sensor RGBD faz com que ela seja resiliente contra ruído de fundo, já que o fundo pode ser extraído utilizando dados de profundidade. Adicionalmente, Wilson mostrou que as estimativas de profundidade fornecidas pelo Kinect são estáveis [Wil10].

Assim como na técnica mostrada na seção 5.3.2, um modelo de fundo é criado tomando-se  $n$  snapshots  $\{S_1, S_2, \dots, S_n\}$  da imagem de profundidade. Denotando o valor de profundidade do  $i$ -ésimo pixel de  $S_j$  por  $d_{ij}$ , sejam  $\mu_j$  e  $\sigma_j$ , respectivamente, a média e o desvio-padrão de  $\{d_{1j}, \dots, d_{nj}\}$ . Dizemos que o modelo de fundo é a coleção de todos os  $\mu_j$ . O conjunto de todos os  $\sigma_j$  é utilizado como um perfil de ruído.



**Figura 5.13:** Passos executados pela camada de rastreamento.

A extração de fundo é desempenhada subtraindo-se o modelo de fundo das imagens de fundo que são continuamente adquiridas.

### Segmentação de imagem

Após o passo anterior, a imagem colorida fornecida pelo sensor RGBD possuirá os artefatos pertencentes ao *foreground*<sup>5</sup>. As marcações coloridas da varinha são segmentadas por meio da conversão da imagem colorida para o espaço HSV e pelo emprego de um método de limiarização no canal da matiz.

### Análise de blobs

Os *blobs* resultantes, computados utilizando análise de componentes conexas, indicam as marcações coloridas. Adicionalmente à marcação azul, a varinha apresenta duas marcações amarelas: uma corresponde ao comutador liga-desliga e a outra, a um dos extremos da varinha. Se o comutador estiver ligado, as marcações se sobrepõem, o que significa que apenas um *blob* amarelo está visível na imagem colorida. Neste caso, o estado da varinha é definido como “ativo”. Se houver dois *blobs* amarelos, o estado da varinha é “inativo”.

---

<sup>5</sup>primeiro plano.

### Normalização de dados 3D

Um sensor RGBD como o Kinect é capaz de fornecer a posição 3D, em seu próprio sistema de coordenadas, de qualquer pixel [KPH<sup>+</sup>12]. Isto posto, a marcação amarela é utilizada para determinar a posição da varinha no espaço 3D. Tal posição é definida como equivalente à posição, no espaço tridimensional, do centro de massa do *blob* amarelo (ou da média dos centros de massa se houver mais de um *blob* amarelo, isto é, se a varinha estiver inativa).

Para que a varinha seja usável pelas aplicações da Superfície Mágica, é necessário normalizar sua posição e orientação, de modo a abrigar diferentes localizações da câmera. Seja  $R$  a região cúbica descrita por:

$$R : \lim_{t \rightarrow \infty} \sqrt{t \left| x - \frac{1}{2} \right|^t + t \left| y - \frac{1}{2} \right|^t + t \left| z - \frac{1}{2} \right|^t} \leq \frac{1}{2}$$

Criamos uma transformação  $T$  que mapeia uma posição do sistema de coordenadas da câmera para o espaço normalizado descrito por  $R$ . Sejam  $(x_k, y_k, z_k)$  as coordenadas da câmera no sistema da câmera e  $(x, y, z)$  a posição normalizada correspondente. Dado um modelo de parâmetros  $(a, b, c, d, e, f, g, h, i, j, k, l)$ , o mapeamento  $T$  é definido como:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ z_k \\ 1 \end{bmatrix}$$

Expandindo a equação matricial acima, temos:

$$\begin{cases} x = a \cdot x_k + b \cdot y_k + c \cdot z_k + d \\ y = e \cdot x_k + f \cdot y_k + g \cdot z_k + h \\ z = i \cdot x_k + j \cdot y_k + k \cdot z_k + l \end{cases}$$

ou seja,

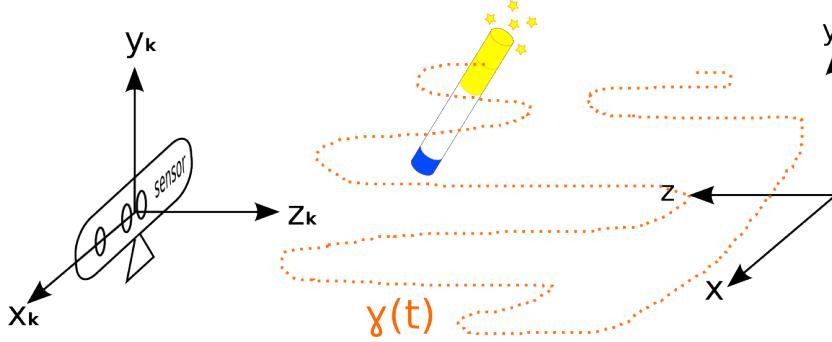
$$\begin{cases} x = x_k \cdot a + y_k \cdot b + z_k \cdot c + 1 \cdot d + 0 \cdot e + 0 \cdot f + 0 \cdot g + 0 \cdot h + 0 \cdot i + 0 \cdot j + 0 \cdot k + 0 \cdot l \\ y = 0 \cdot a + 0 \cdot b + 0 \cdot c + 0 \cdot d + x_k \cdot e + y_k \cdot f + z_k \cdot g + 1 \cdot h + 0 \cdot i + 0 \cdot j + 0 \cdot k + 0 \cdot l \\ z = 0 \cdot a + 0 \cdot b + 0 \cdot c + 0 \cdot d + 0 \cdot e + 0 \cdot f + 0 \cdot g + 0 \cdot h + x_k \cdot i + y_k \cdot j + z_k \cdot k + 1 \cdot l \end{cases}$$

Um procedimento de calibração é empregado para estimar os parâmetros do modelo. Dado um conjunto de  $n \geq 4$  correspondências, denotadas por  $(x^{(j)}, y^{(j)}, z^{(j)}) \leftrightarrow (x_k^{(j)}, y_k^{(j)}, z_k^{(j)})$  para  $1 \leq j \leq n$ , as equações acima podem ser reescritas como:

$$\begin{bmatrix} x^{(1)} \\ y^{(1)} \\ z^{(1)} \\ x^{(2)} \\ y^{(2)} \\ z^{(2)} \\ \vdots \\ x^{(n)} \\ y^{(n)} \\ z^{(n)} \end{bmatrix} = \begin{bmatrix} x_k^{(1)} & y_k^{(1)} & z_k^{(1)} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_k^{(1)} & y_k^{(1)} & z_k^{(1)} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_k^{(1)} & y_k^{(1)} & z_k^{(1)} \\ x_k^{(2)} & y_k^{(2)} & z_k^{(2)} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_k^{(2)} & y_k^{(2)} & z_k^{(2)} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_k^{(2)} & y_k^{(2)} & z_k^{(2)} \\ \ddots & & & & & & & & & & \\ x_k^{(n)} & y_k^{(n)} & z_k^{(n)} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_k^{(n)} & y_k^{(n)} & z_k^{(n)} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_k^{(n)} & y_k^{(n)} & z_k^{(n)} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \\ k \\ l \end{bmatrix}$$

Uma solução de quadrados mínimos para os parâmetros do modelo pode ser encontrada através da aplicação de uma matriz pseudo-inversa, computada utilizando métodos como decomposição em valores singulares ou fatoração QR [TBI97].

Do ponto de vista do usuário final, é desnecessário descrever manualmente as correspondências. O procedimento de calibração requer, simplesmente, que o usuário movimente a varinha próximo à câmera (vide Figura 5.14). Seja  $\gamma(t) : (x_k(t), y_k(t), z_k(t))$  a curva que descreve tal movimento, onde  $t \geq 0$  denota o tempo.



**Figura 5.14:** Procedimento de calibração.

Procedemos definindo o que denominamos funções de caixa:

$$\begin{aligned} x_k^\#(d) &= d \cdot \max_{t \geq 0} x_k(t) + (1 - d) \cdot \min_{t \geq 0} x_k(t) \\ y_k^\#(d) &= d \cdot \max_{t \geq 0} y_k(t) + (1 - d) \cdot \min_{t \geq 0} y_k(t) \\ z_k^\#(d) &= (1 - d) \cdot \max_{t \geq 0} z_k(t) + d \cdot \min_{t \geq 0} z_k(t) \end{aligned}$$

Agora, para  $i \in \mathbb{N}^*$ , considere a seguinte família de funções:

$$J_i(j) = \left\lfloor \frac{j-1}{2^{(i-1)}} \right\rfloor - 2 \left\lfloor \frac{j-1}{2^i} \right\rfloor$$

Então, para  $j = 1, 2, \dots, 8$ , definimos as correspondências:

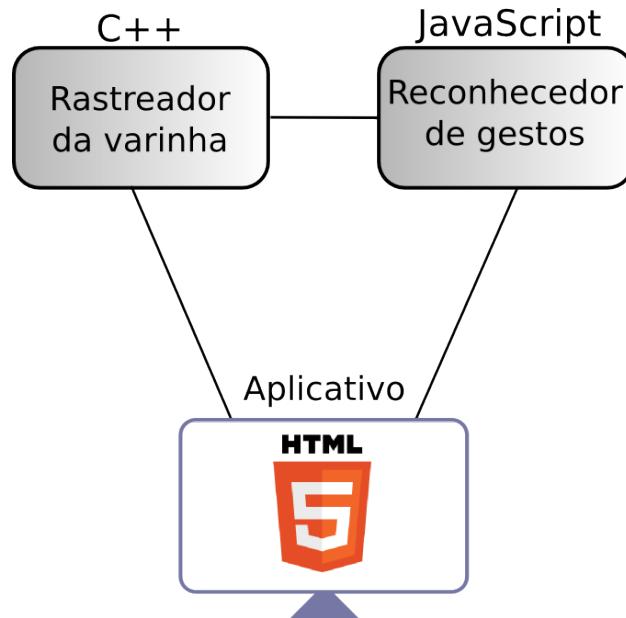
$$\begin{aligned} x_k^{(j)} &= J_1(j) \leftrightarrow x_k^{(j)} = x_k^\#(J_1(j)) \\ y_k^{(j)} &= J_2(j) \leftrightarrow y_k^{(j)} = y_k^\#(J_2(j)) \\ z_k^{(j)} &= J_3(j) \leftrightarrow z_k^{(j)} = z_k^\#(J_3(j)) \end{aligned}$$

Uma vez que os modelos do parâmetro tenham sido estimados, a posição 3D dos dois extremos da varinha, no espaço normalizado, determinará sua orientação. Esta fase termina emitindo o estado (ativo ou inativo) e a estimação da pose com 5 graus de liberdade: a posição 3D da varinha e dois ângulos de rotação.

#### 5.4.4 Gestos

Aplicativos HTML5 têm acesso aos dados de posição, orientação e estado da varinha. Isto se dá por meio do uso do protocolo TUIO [KBBC05] via WebSockets. Adicionalmente, um módulo de reconhecimento de gestos amplia as capacidades interativas da varinha, tornando-a efetivamente “mágica” (ver Fig. 5.15).

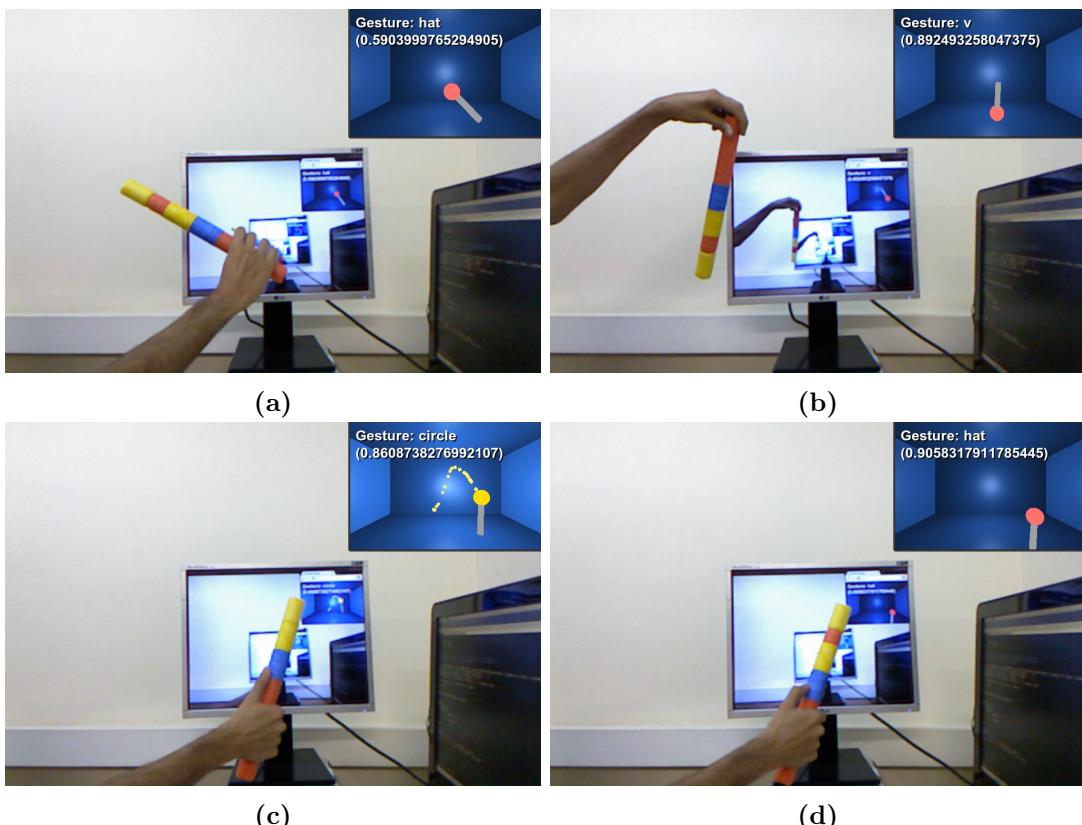
O reconhecedor de gestos recebe um feitiço do rastreador de varinha e atribui a ele um rótulo ( $\wedge$ ,  $\vee$  ou  $\bigcirc$ ). Mais especificamente, vale destacar que os feitiços realizados pelo usuário são projetados



**Figura 5.15:** A varinha pode controlar uma aplicação HTML5.

num plano e depois passados ao reconhecedor. O reconhecedor recebe os gestos, para em seguida normalizá-los e compará-los com um conjunto de gestos conhecidos, conforme descrito em [WWL07].

Para fins de prototipação, utilizamos o \$1 Gesture Recognizer, um classificador leve e robusto que faz uma busca de vizinhos mais próximos utilizando uma métrica baseada na geometria dos gestos [WWL07]. A Figura 5.16 mostra uma implementação do protótipo rodando a aproximadamente 28 quadros per segundo num PC Dell All-in-One Inspiron 2330 (com 6 GB de memória RAM e um CPU Intel Core i5 rodando a 2,30 GHz).

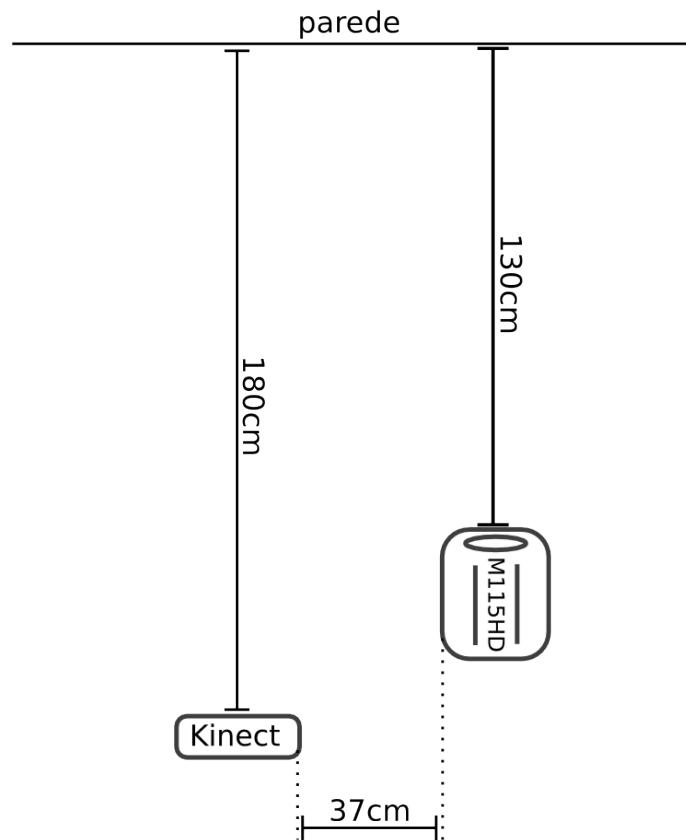


**Figura 5.16:** Protótipo em funcionamento.

Em nossos testes, a distância entre a varinha e a câmera variava de  $50\text{cm}$  a  $90\text{cm}$ , aproximadamente. O ajuste automático de brilho do Kinect causava, às vezes, dificuldades no rastreamento. Acreditamos que o uso de iluminação ativa possa mitigar tais dificuldades. Apesar disso, conseguimos usar a varinha mágica para interação.

#### 5.4.5 Instalação do protótipo

Estritamente falando, o uso da varinha mágica não requer um projetor de vídeo. Quando um for usado, entretanto, sugerimos uma instalação física ligeiramente diferente da mostrada em 5.3.7. O Kinect é colocado mais distante da parede que o projetor, e ambos deixam de ficar centralizados. A Figura 5.17 exibe as adaptações sugeridas.



**Figura 5.17:** Adaptações da planta para o uso da varinha mágica.

# Capítulo 6

## Explorando formas de interação

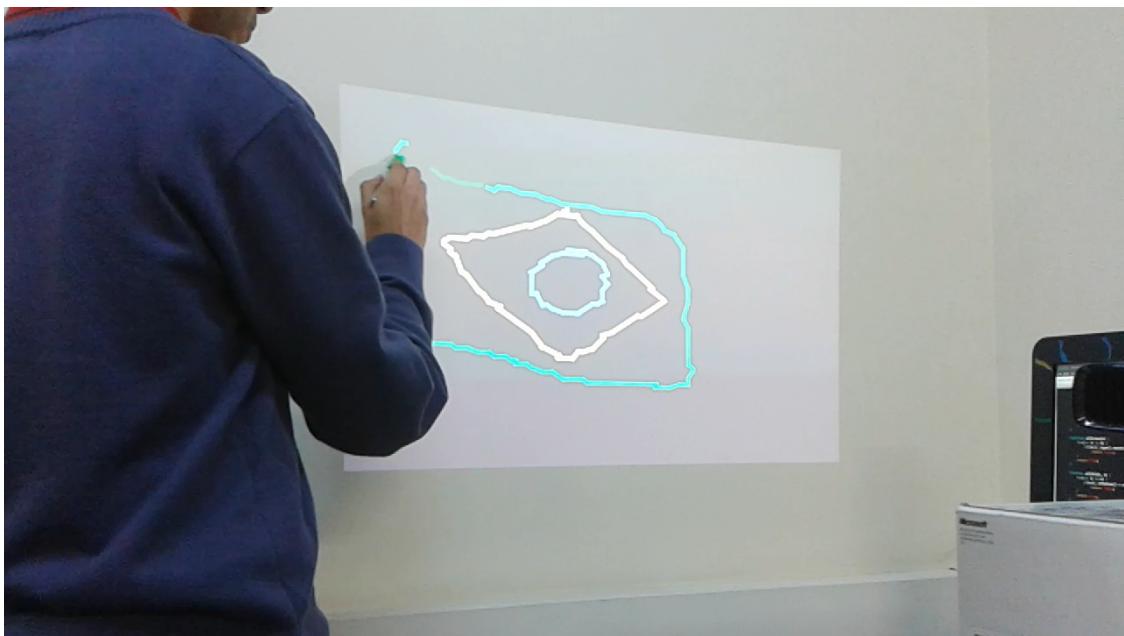
### 6.1 Interação 2D usando canetas e apagador

Uma maneira de se explorar formas de interação com uma superfície interativa é imaginá-la como uma lousa digital. Lousas permitem que o utilizador escreva, desenhe, apague e, de modo geral, manipule conteúdo. Neste trabalho, propusemos duas aplicações que enfatizam tais características: Lousa Mágica e Mapa Mágico.

#### 6.1.1 Lousa Mágica

##### Visão geral

O aplicativo Lousa Mágica transforma a superfície interativa num espaço para desenho que pode ser usado como meio de colaboração, estudos ou arte. A área de desenho é ilimitada, no sentido de que o usuário, além de riscar a superfície com cores diversas, pode arrastar e redimensionar o conteúdo. A Figura 6.1 mostra um exemplo de uso do programa.



**Figura 6.1:** Um desenho numa parede tornada interativa.

##### Interação

Assim como num quadro-branco convencional, por meio de canetas coloridas o usuário pode inserir marcações na superfície de desenho. Com o apagador, pode-se remover conteúdo. Ao contrário

de uma solução analógica, entretanto, com a Superfície Mágica o usuário tem a possibilidade utilizar gestos para efetuar translação e escala do conteúdo. Tais gestos, amplamente usados em soluções *multitouch* [KPH<sup>+</sup>12], são como mostrados na Figura 6.2:

- Encostar e arrastar dedo para mover o conteúdo;
- Encostar e arrastar dois dedos separados para redimensionar o conteúdo.

(a) *Translação*.(b) *Escala*.**Figura 6.2:** *Gestos*.

Sejam  $x_i, x_f, x'_i, x'_f \in [0, 1]^2$ . Se o usuário encosta um dedo (ou, genericamente, um par de dedos) na posição  $x_i$  e o arrasta para  $x_f$ , então o conteúdo desenhado será deslocado por:

$$\Delta x = x_f - x_i$$

Por outro lado, se o usuário encosta um dedo em  $x_i$  e outro em  $x'_i$ , arrastando-os para  $x_f$  e  $x'_f$ , respectivamente, então o conteúdo será ampliado (ou reduzido) pelo fator:

$$\Delta s = \frac{\|x'_f - x_f\|}{\|x'_i - x_i\|}$$

A implementação da Lousa Mágica foi realizada com a biblioteca HTML5 paperjs<sup>1</sup>. A Figura 6.3 mostra interações diversas com a aplicação.

---

<sup>1</sup><http://paperjs.org/>. Último acesso: 30 de maio de 2015.



**Figura 6.3:** Interações diversas na Lousa Mágica.

### 6.1.2 Mapa Mágico

#### Visão geral

O Mapa Mágico, que pode ser visto como uma extensão da Lousa Mágica, é uma aplicação de mapas. Seu objetivo é mostrar como a superfície interativa pode ser transformada num mapa cuja abrangência varie desde o mundo inteiro até o nível de ruas e avenidas. Pode-se pensar em tal aplicação num contexto educacional ou de reunião de negócios. A Figura 6.4 mostra um exemplo de uso.



**Figura 6.4:** Um mapa interativo projetado numa parede.

## Interação

Assim como na Lousa Mágica, o usuário pode utilizar canetas coloridas para desenhar sobre o mapa, o apagador para limpar a marcação das canetas e gestos com os dedos para arrastar e redimensionar o conteúdo. Uma interessante característica de se trazer uma solução de mapas para uma superfície interativa é a possibilidade de se fazer anotações que respeitam a escala do mapa.

A implementação deste aplicativo foi realizada mediante utilização da API do Google Maps<sup>2</sup>. Ao contrário do paperjs utilizado na Lousa Mágica, a API do Google Maps atualmente suporta apenas níveis discretos de *zoom*. Isto requer, portanto, uma adaptação no mecanismo de escala.

Sejam  $x_i, x_f, x'_i, x'_f \in [0, 1]^2$ . Suponha que o usuário encoste um dedo na posição  $x_i$  e outro na posição  $x'_i$ , arrastando-os para  $x_f$  e  $x'_f$ , respectivamente. Então, o nível de *zoom* do mapa será incrementado por:

$$\Delta s = \left\lfloor \log_2 \left( \frac{\|x'_f - x_f\|}{\|x'_i - x_i\|} \right) + \frac{1}{2} \right\rfloor$$

Intuitivamente, a expressão acima codifica: se douro a distância entre os dedos, aumento o *zoom* em 1 unidade. Se diminuo-a pela metade, reduzo o *zoom* também em 1 unidade.

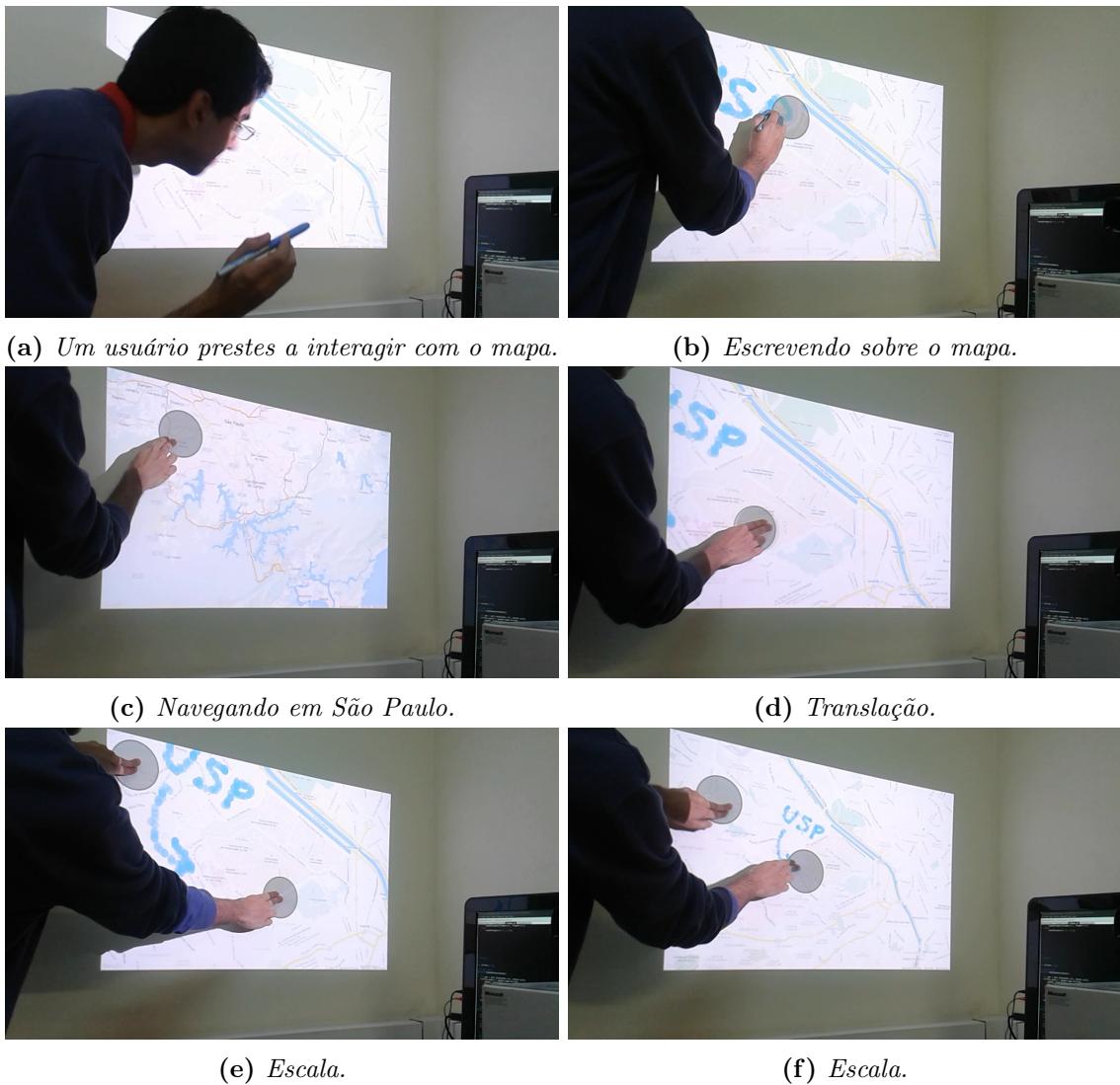
A Figura 6.5 mostra interações diversas com a aplicação de mapas.

## 6.2 Interação 2D usando o próprio corpo

Na linha da computação ubíqua, além de introduzir a computação no ambiente, podemos utilizar o corpo do próprio usuário para interação. A funcionalidade de rastreamento de pessoas do Kinect já é um passo nessa direção (ver 4.2). Neste trabalho, entretanto, exploramos o uso de uma parte do corpo humano como superfície interativa.

---

<sup>2</sup><https://developers.google.com/maps/>. Último acesso: 30 de maio de 2015.



**Figura 6.5:** Interações diversas no Mapa Mágico.

### 6.2.1 Toque Mágico

#### Visão geral

O aplicativo Toque Mágico transforma uma das mãos do usuário num controle deslizante (*slider*). Por meio de um gesto de toque feito com dedos da outra mão, controla-se o *slider*. Tal funcionalidade poderia ser utilizada num sistema de computação sensível ao contexto. Por exemplo: em ambientes de casas inteligentes, esse mecanismo poderia ser utilizado para controlar a intensidade luminosa de uma lâmpada num cômodo ou para controlar a passagem de luz de uma janela/parede inteligente translúcida (ver 2.8), dependendo da direção do olhar do usuário. Em nossa prova de conceito, utilizamos o *slider* para girar um objeto tridimensional.

#### Interação

Em nosso protótipo, o usuário utiliza uma das mãos como *slider* horizontal e dedos da outra mão como cursor. À medida que os dedos são deslocados na superfície da mão tornada interativa, o objeto tridimensional é rotacionado.

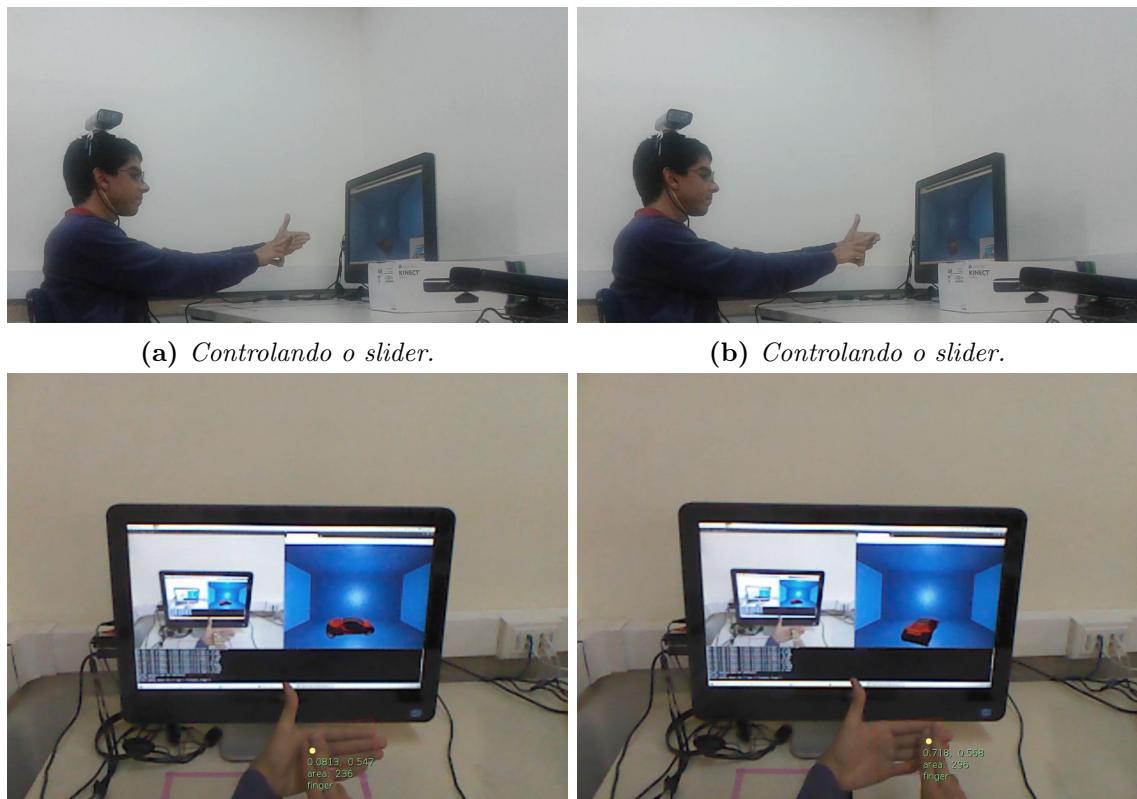
Sejam  $x_i, x_f \in [0, 1]^2$ . Suponha que o usuário encoste um dedo na posição  $x_i$ , deslocando-o para  $x_f$ . Neste caso, o deslocamento angular do objeto em torno do eixo vertical é dado por:

$$\Delta\theta = \frac{k \pi}{180} \left( (x_f - x_i)^t \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)$$

onde  $k \in \mathbb{Z}^*$  é uma constante empírica.



**Figura 6.6:** Um Kinect preso a um usuário.



(a) Controlando o slider.

(b) Controlando o slider.

(c) Visão da câmera de cena.

(d) Visão da câmera de cena.

**Figura 6.7:** Toque Mágico.

Para demonstrar o conceito, prendemos um Kinect ao corpo do usuário (Fig. 6.6), de modo que o sensor seja utilizado como câmera de cena. A Figura 6.7 exibe a prova de conceito.

Notamos que uma câmera RealSense seria mais apropriada que um Kinect para esta demonstração. Além de ser menor e mais leve, o intervalo de funcionamento do sensor de profundidade vai de 0,2m a 1,2m<sup>3</sup> - ou seja, é mais próximo que o do Kinect (ver 4.2).

O Toque Mágico foi implementado utilizando a biblioteca HTML5 three.js.<sup>4</sup> O aplicativo dispensa o uso de projetores de vídeo.

## 6.3 Interação 2D usando objetos tangíveis

Seres humanos desenvolveram habilidades complexas para interagir com o ambiente físico. Agarrar objetos é uma ação prevalente em nosso dia-a-dia. Atualmente, entretanto, muitas dessas habilidades não são exploradas quando se interage com conteúdo digital. Isto posto, nos campos da arte interativa e do entretenimento, as interfaces tangíveis - discutidas por Hiroshi Ishii em Tangible Bits [Ish08] - dão origem a muitas possibilidades criativas. Um exemplo é o Music Bottles, um projeto que explora o uso garrafas, dispostas sobre uma mesa especial, como interfaces para o controle de música. [IML01]

### 6.3.1 Tangible Beats

#### Visão geral

Uma caixa de música é um instrumento musical mecânico. Um pente de metal com dentes que emitem sons é tocado por pinos espalhados na superfície de um cilindro giratório (Fig. 6.8). Esse mecanismo de produção de sons inspira formas inovadoras de ajudar músicos amadores a compor trabalhos em tempo real e de maneira colaborativa. Em particular, pode-se desenvolver uma interface tangível que se baseia no conceito da caixa de música, introduzindo também elementos de entretenimento, interatividade e colaboração.



**Figura 6.8:** Caixa de música<sup>5</sup>.

Design exploratório no campo das interfaces musicais assistidas por computador tem sido um tema de pesquisa nos últimos anos. Usando um sensor Nintendo Wiimote, Fornari propôs uma inter-

<sup>3</sup>[http://en.wikipedia.org/wiki/Intel\\_RealSense](http://en.wikipedia.org/wiki/Intel_RealSense). Último acesso: 7 de junho de 2015.

<sup>4</sup><http://threejs.org/>. Último acesso: 7 de junho de 2015.

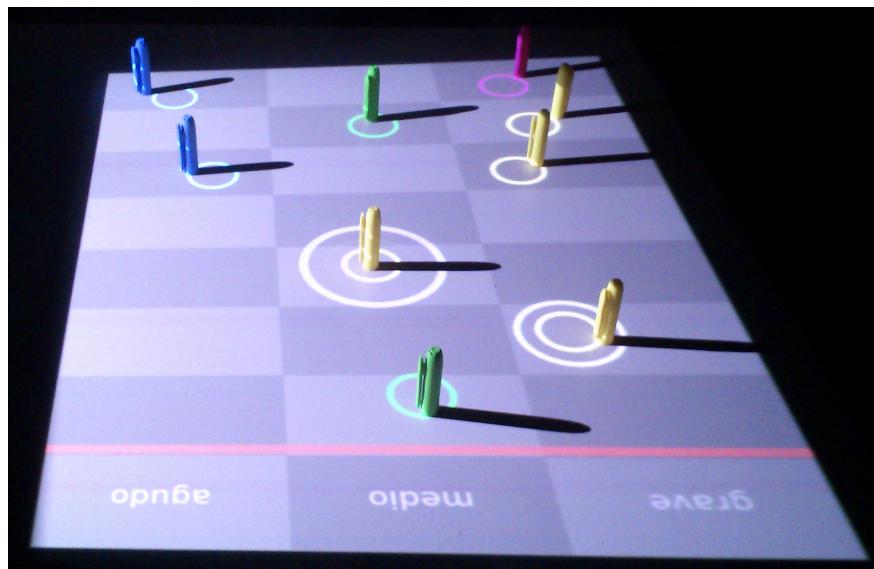
<sup>5</sup>Autor: Simon A. Eugster. CC-BY-SA 3.0. [http://commons.wikimedia.org/wiki/File:Musical\\_box\\_open.jpg](http://commons.wikimedia.org/wiki/File:Musical_box_open.jpg). Último acesso: 22 de julho de 2014.

face baseada em gestos que utiliza movimentos corporais para expressão musical [For10]. Os dados do acelerômetro são mapeados para mecanismos de síntese sonora. O popular reacTable [JGAK07] introduziu um design inovador que, por meio de uma interface tangível, possibilita que *performers* componham ritmos musicais complexos de maneira colaborativa e em tempo real. Tendo atingido popularidade massiva fora do mundo acadêmico, ele eventualmente se tornou um produto comercial.

Focando-se em músicos amadores, The Music Table [BMHS03] apresenta uma maneira de compor padrões musicais por meio do arranjo, numa mesa, de cartões impressos em papel. Cada cartão representa um som. Ao mover os cartões horizontal ou verticalmente, um usuário pode, respectivamente, modificar a altura dos sons, ou sua posição numa sequência tocada repetidamente. Uma câmera rastreia a posição dos cartões. *Feedback* em tempo-real é dado na forma de áudio e de imagens exibidas numa tela colocada próxima à mesa. Um cartão especial, denominado *instrument card*, permite que se mude o instrumento de outros cartões por meio de uma “palheta” de instrumentos pré-programados. Elementos da cultura japonesa inspiram a estética visual: criaturas tridimensionais são renderizadas no topo dos cartões, lembrando protótipos de realidade aumentada. Apesar de o sistema trazer uma representação física da música, o retorno visual é desacoplado da mesa (os espaços de entrada e de saída são distintos). Utilizar diferentes instrumentos musicais pode ser incômodo, exigindo muitas etapas para modificar cada cartão.

Ainda na área de interação tangível, BeatBearing [BO08] explora a “natural” habilidade dos usuários de manipular objetos físicos, fazendo uso de restrições e *affordances* físicas. Objetivando prover uma experiência de “flow”, BeatBearing traz uma interface auto-contida. O usuário se vê diante de uma superfície física de controle, de uma tela e de uma fonte sonora que compartilham o mesmo espaço de interação. No topo de um monitor de tela plana posicionado horizontalmente, há componentes eletrônicos que tomam forma de grade. O ritmo musical é criado ao se colocar bolas em buracos espalhados pela grade. Modificar a posição da bola nos eixos horizontal e vertical muda, respectivamente, a posição e o instrumento da nota. Uma linha de varredura é mostrada no monitor. Ao passar por uma bola, a nota apropriada é tocada. Apesar de não ser possível mudar a altura de uma nota, BeatBearing permite a manipulação “hands-on” do ritmo musical.

Tangible Beats é uma interface de usuário que permite a criação de ritmos musicais por meio da manipulação de *tokens* coloridos<sup>6</sup> numa superfície plana. O projeto provê tanto uma representação tática quanto uma visual dos ritmos, o que permite que os usuários intuitivamente compreendam a essência abstrata da música (ver Figura 6.9). Ao manipular os *tokens*, usuários novatos criam e escutam novos padrões musicais de forma colaborativa e em tempo real.



**Figura 6.9:** *Tangible Beats*.

---

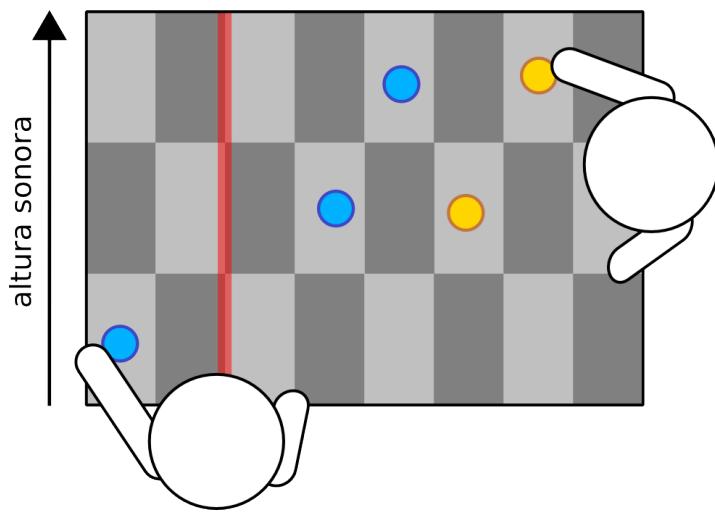
<sup>6</sup>Na prática, esses *tokens* são tampas de canetas coloridas, daquelas que se usa para escrever em CDs e DVDs.

## Interação

O objetivo de design desta aplicação é permitir que leigos brinquem com música de forma colaborativa e intuitiva. Tangible Beats, um caso de design exploratório, é construído sobre o conceito da caixa de música. O simples mecanismo desse dispositivo pode ser entendido por pessoas sem qualquer *background* musical, fazendo dele a inspiração de nossa escolha. Devido às restrições físicas e às *affordances* dos *tokens*, a colaboração entre vários usuários é uma consequência “natural” do design. Os usuários simplesmente movem os *tokens* numa mesa e escutam os ritmos musicais resultantes.

Este experimento de design conta com:

- uma grade retangular de  $8 \times 3$  células, que é projetada numa mesa;
- 4 *tokens* coloridos de 4 cores diferentes (amarelo, azul, verde, magenta), totalizando 16 *tokens*;
- uma linha vermelha vertical de escaneamento que se move horizontalmente, repetidamente desaparecendo num lado da grade e reaparecendo no lado oposto.

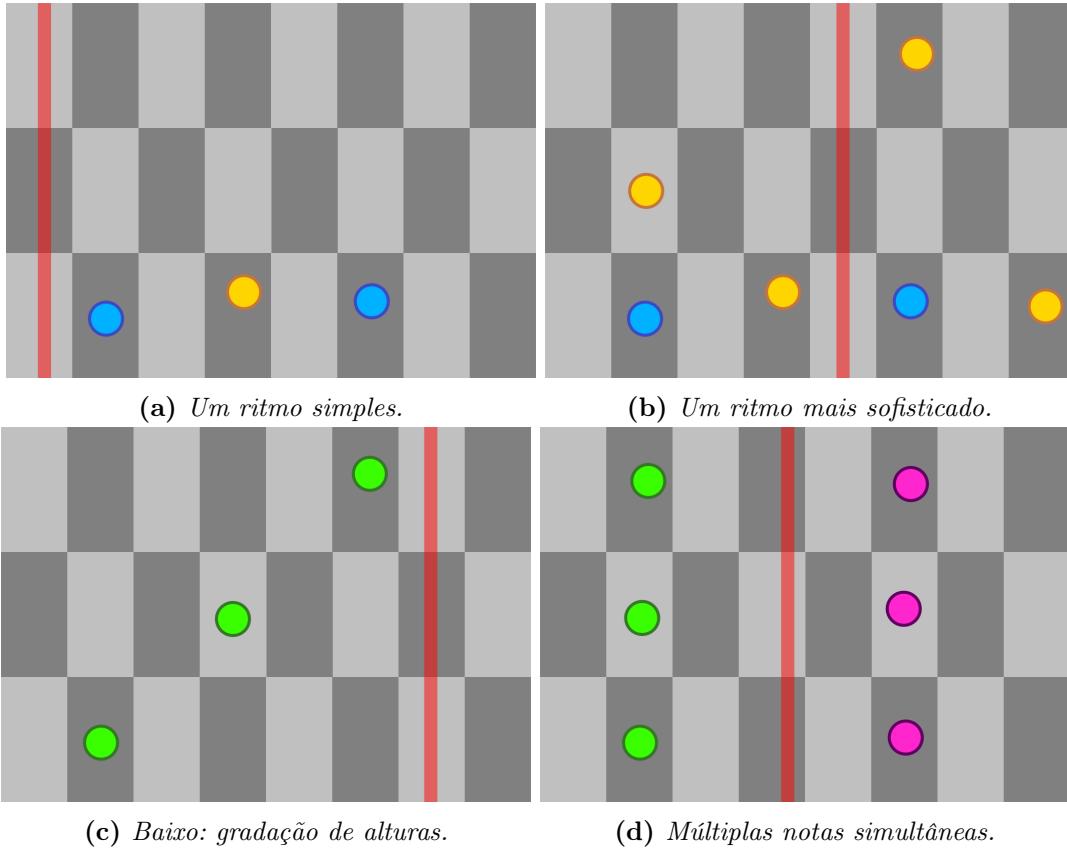


**Figura 6.10:** Composição colaborativa.

Como mostrado na Figura 6.10, os usuários compõem seus *beats* interativa e colaborativamente colocando *tokens* nas células da grade. O modelo físico da música funciona da seguinte forma:

- cada *token* físico representa um som;
- *tokens* de diferentes cores são mapeados para diferentes instrumentos;
  - Os *tokens* amarelo, azul, verde e magenta são mapeados para chimbau, tambor, “bater de palmas” e “retrô”, respectivamente.
- cada célula da grade pode armazenar até um som;
- o eixo horizontal da grade representa o tempo e o vertical, a altura de um som.

Quanto mais próximo um *token* estiver da parte “inferior” da grade, menor a altura do som correspondente. Inversamente, *tokens* colocados no “topo” da grade são mapeados para sons de maior altura. Sempre que a linha de escaneamento atravessa um *token*, o som correspondente é tocado. Nesse momento, um anel luminoso é mostrado temporariamente de cima para baixo para prover *feedback* visual adicional. Combinar as dimensões do tempo e da altura sonora possibilita um mecanismo fácil de aprender para a composição dos ritmos musicais. Como os *tokens* podem ser colocados em qualquer lugar num espaço físico contínuo, a grade dá uma característica discreta aos sons, o que



**Figura 6.11:** Ritmos de exemplo.

faz com que tenhamos algo similar a um compasso simples. As Figuras 6.11a, 6.11b, 6.11c e 6.11d mostram diferentes ritmos que podem ser criados com o sistema.

Quanto mais rápida a linha de escaneamento, mais *beats* por segundo serão ouvidos pelo usuário. Pode-se modificar a velocidade da linha de escaneamento usando um gesto bimanual. Quando um dedo for colocado na grade (digamos, dedo *A*), um modo de configuração de velocidade é ativado. Tocar a grade com outro dedo (digamos, *B*) permite a modificação da velocidade da seguinte forma: mover *B* na direção da linha de escaneamento aumenta sua velocidade, mas movê-lo na direção oposta diminui sua velocidade. Ao retirar o dedo *A* da mesa, o usuário desativa o modo de configuração de velocidade.

A tabela 6.1 resume a conexão entre os recursos oferecidos pela interface e o retorno auditivo.

Retorno sonoro	Recurso da interface
Instrumento	<i>Tokens</i> de diferentes cores
Localização dos <i>beats</i> no tempo	Posição horizontal dos <i>tokens</i>
Altura sonora	Posição vertical dos <i>tokens</i>
Tempo	Gesto bimanual

**Tabela 6.1:** Como os recursos da interface são mapeados para feedback sonoro.

Anéis coloridos são projetados ao redor dos *tokens* como uma forma de retorno visual para informar que o sistema reconheceu os artefatos físicos. Apesar de a linha de escaneamento ser uma entidade virtual, um som é tocado sempre que ela ultrapassa um *token*. A mesa então se torna um dispositivo de interação bidimensional, já que o usuário comunica suas intenções e recebe o *feedback* visual correspondente na própria mesa. Em outras palavras: a entrada e a saída de dados ocorre no mesmo espaço interativo.

Assim como mostrado na Figura 6.12, a instalação física do protótipo inclui um sensor Kinect, um projetor de vídeo e alto-falantes. Do ponto de vista de implementação, o módulo de visão



**Figura 6.12:** Configuração típica de uma instalação do Tangible Beats, mostrando: mesa, Kinect, projetor, caixa de som e alguns objetos coloridos.

computacional descrito em 5.3 provê tuplas  $(x_i, y_i, c_i)$  que indicam a posição e a cor dos *tokens* físicos. Esses dados são convertidos para saída visual e sonora. Como a linha de escaneamento varre a superfície interativa no eixo-x, faz-se necessário discretizar as posições dos *tokens*, dado que, na prática, elas residem num espaço contínuo  $[0, 1] \times [0, 1]$ . Foi utilizada a seguinte função de discretização:

$$D_w(x) = \frac{\lfloor wx \rfloor}{w} + \frac{1}{2w}$$

onde  $w \in \mathbb{N}^*$  é o número de células da grade no eixo horizontal. Sempre que a linha de escaneamento, caracterizada por uma variável  $s \in [0, 1]$ , ultrapassa um *token*, o som correspondente é tocado. A grade  $8 \times 3$  é renderizada usando cores neutras, garantindo que a matiz dos *tokens* coloridos seja mantida na presença do projetor de vídeo.

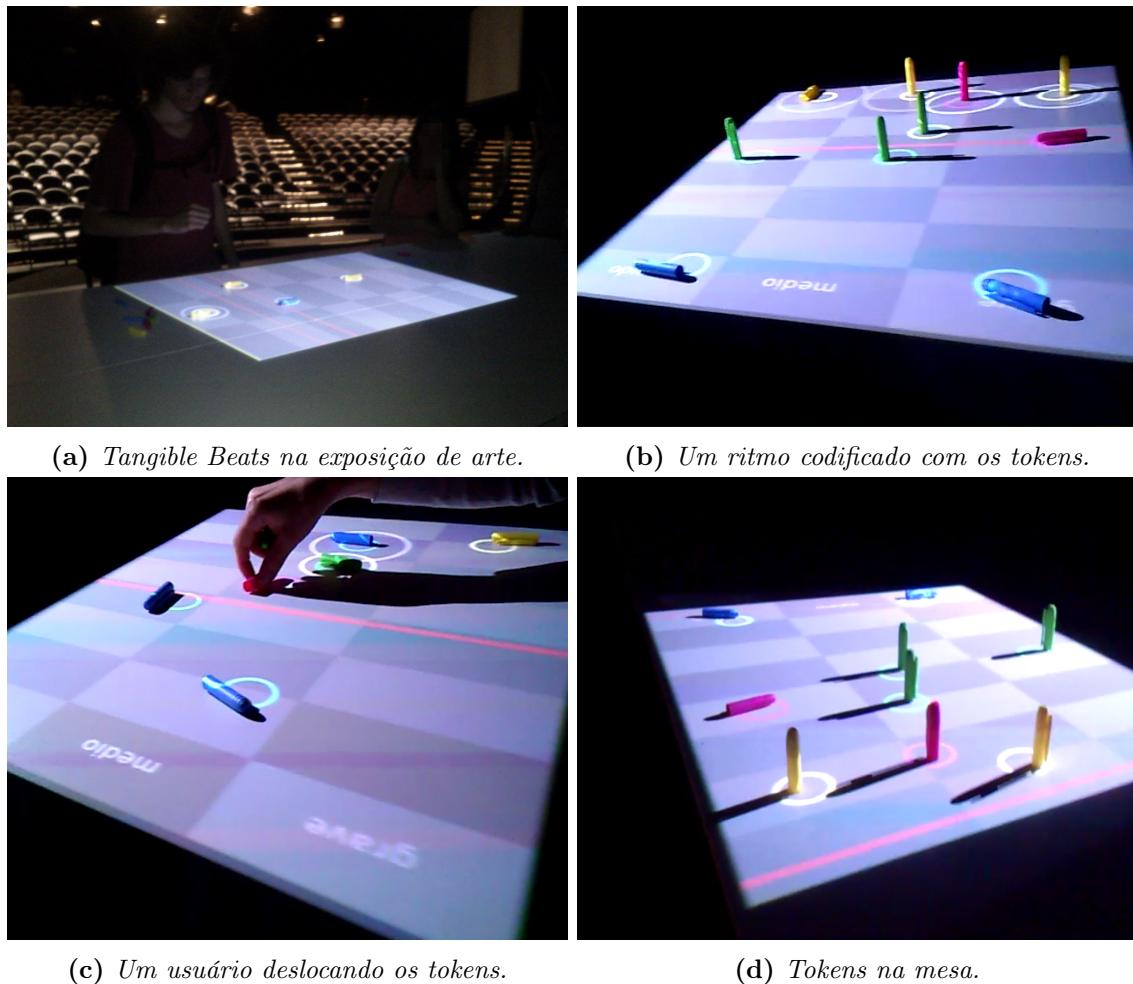
## Testes

Em 2014, a Tenda Cultural Ortega y Gasset<sup>7</sup>, então instalada na Praça do Relógio da Universidade de São Paulo, publicou um edital para que alunos enviassem, para exibição pública, projetos artísticos envolvendo: música, teatro, cinema, dança, fotografia, etc. Decidimos que essa seria uma boa oportunidade para exibir um projeto de arte interativa ao público geral. Enviamos a proposta do Tangible Beats e o projeto foi aprovado com entusiasmo. Contando com a ajuda de vários funcionários da tenda (desde a parte logística até a instalação do protótipo no palco de teatro da tenda), conseguimos exibir o sistema por dois dias. Também ganhamos um camarim.

Durante a exibição, visitantes brincavam livremente com o sistema. O perfil dos usuários variava desde crianças pequenas até pessoas mais maduras. O público foi composto majoritariamente por universitários, visitantes do ensino médio e funcionários locais. Em nossos testes, verificamos que

<sup>7</sup><http://prceu.usp.br/tendaculturalortegaygasset/>. Último acesso: 1 de junho de 2015.

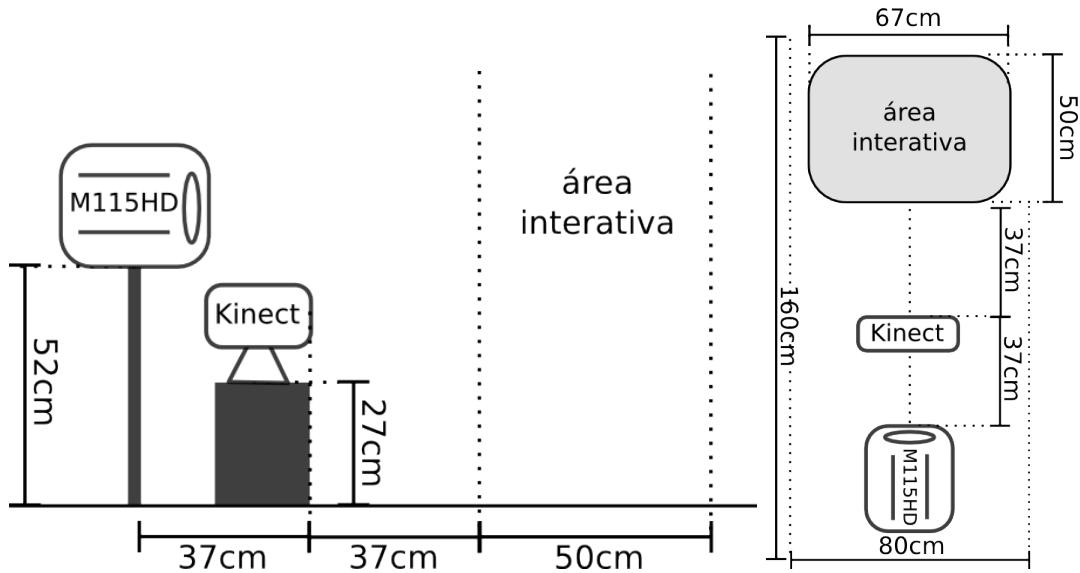
as pessoas geralmente “descobriam” os ritmos interativamente (ver Figura 6.13). Elas manipulavam os *tokens* até que escutassem algo que “fizesse sentido” para elas. Apesar de termos dado uma explicação básica do que o sistema fazia, observamos que a maioria dos usuários preferiam descobrir por si mesmos como utilizá-lo (provavelmente devido à natureza lúdica do protótipo). Quando surgiam dúvidas, os próprios usuários perguntavam o que queriam saber. A colaboração ocorria espontaneamente, particularmente entre casais de namorados e grupos de amigos.



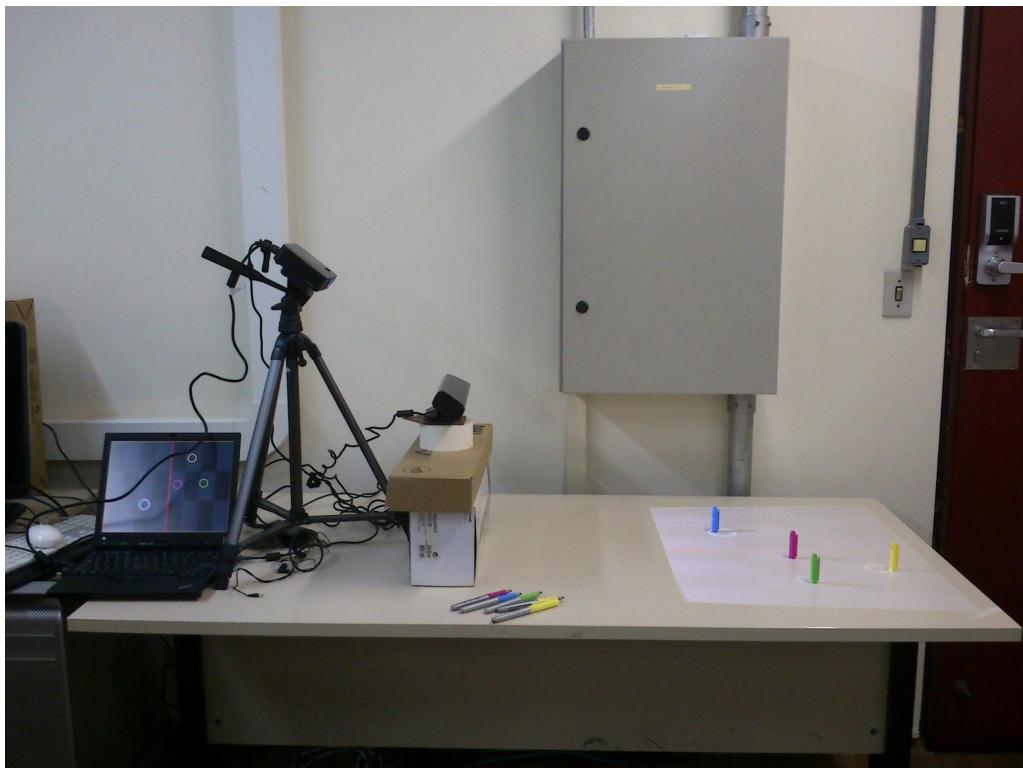
**Figura 6.13:** O protótipo em exibição na tenda.

O feedback dos usuários tendeu ao entusiasmo: “*Fiquei viciada nisso, não consigo parar! É muito interativo!*”, “*Forma nova, divertida e incrível de tocar um instrumento*”, “*Cara, é muito mágico isso! Como você fez?*”, “*Eu gostei porque pude criar um som que fosse repetitivo, que parecia música*”. Tal feedback sugere que o design introduzido por este trabalho é apropriado para um contexto de entretenimento.

A Figura 6.14 mostra uma planta sugerida para a instalação do sistema. Apesar de o protótipo funcionar bem num ambiente controlado de laboratório (ver Fig. 6.15), houve algumas questões técnicas durante a exibição de arte. A mesa foi colocada num palco de teatro com chão um tanto trémulo. Isso introduziu, de tempos em tempos, ruído no sistema, exigindo recalibração. Adicionalmente, devido a condições adversas de iluminação, o módulo de reconhecimento de cores falhou algumas vezes. Como ele não reconhecia cor de pele corretamente, houve problemas para testar o gesto bimanual para controlar a velocidade da linha de escaneamento. De acordo com um músico local, tal controle seria um “*imenso recurso para expressão criativa*”. Possíveis *workarounds* para essa questão incluem: treinar os modelos de cor em condições adversas de iluminação ou introduzir uma barra de rolagem virtual na mesa.



**Figura 6.14:** Planta sugerida para a instalação do Tangible Beats.



**Figura 6.15:** Protótipo montado num laboratório.

Após a defesa deste mestrado, também apresentamos o Tangible Beats na Feira de Profissões da USP, realizada em agosto de 2015 no Parque de Ciência e Tecnologia da USP (ver Fig. 6.16a). O evento é gratuito e voltado a alunos do ensino médio e a vestibulandos em geral. Os visitantes são recebidos por professores, alunos de graduação e de pós-graduação e funcionários da USP, obtendo informações relativas a: unidades de ensino, cursos oferecidos, vestibular, formação acadêmica, grade curricular, especializações, etc. O Tangible Beats foi apresentado principalmente para auxiliar na divulgação do curso de Ciência da Computação do IME-USP. Assim como na Tenda Cultural, o projeto foi recebido pelos usuários com grande entusiasmo e curiosidade (Figs. 6.16b e 6.16c).



(a) Feira de Profissões da USP.



(b) Os alunos Deusany Júnior, da Computação (à esquerda) e Gabriel Capella, da Estatística (à direita), ajudando na instalação do protótipo.



(c) Tangible Beats sendo apresentado ao público.

**Figura 6.16:** Tangible Beats na Feira de Profissões da USP.

Finalmente, notamos que o Tangible Beats também rendeu uma reportagem para a Agência Universitária de Notícias<sup>89</sup> e uma citação pelo Núcleo de Empreendedorismo da USP<sup>10</sup>.

<sup>8</sup><http://www.usp.br/aun/exibir.php?id=6484&edicao=1141>. Último acesso: 1 de junho de 2015.

<sup>9</sup><https://youtu.be/h6EtCzf02dA>. Último acesso: 1 de junho de 2015.

<sup>10</sup><https://pt-br.facebook.com/uspempreende/posts/841598645913883>. Último acesso: 5 de junho de 2015.

## 6.4 Interação 3D usando uma varinha mágica

Uma importante aplicação das interfaces pós-WIMP é a interação em ambientes virtuais. As interfaces de usuário que possibilitam tal atividade idealmente devem ser fáceis de usar e de aprender, de tal modo que seu uso se torne “habitual”, “instintivo”<sup>11</sup>. À medida que nossa interação com os computadores de expande além da configuração do *desktop* tradicional, torna-se desejável explorar outras formas de *input*. Uma possível abordagem é utilizar uma varinha.

As *affordances* de uma varinha podem ser exploradas de várias maneiras, já que ela pode ser mapeada para diferentes tarefas de interação em ambientes virtuais. Usar a noção de uma varinha mágica para interação é interessante, pois além de ser não obstrutiva, ela reduz a curva de aprendizado de dispositivos menos intuitivos como por exemplo o *mouse* 3D [CGVT03]. Num estudo recente, Khan *et al.* [KPD12] mostraram que adotar uma varinha é uma maneira eficiente de se deslocar num ambiente virtual, comparado com depender da posição do usuário no espaço físico. Adicionalmente, em comparação com luvas interativas ou *mouse* 3D, os usuários de varinhas tendem a adotar posturas corporais mais confortáveis (eles tendem a ficar em posturas que gostem em vez de posturas que eles acreditam ser necessárias para o funcionamento do sistema) [HGJ<sup>+</sup>13].

Veremos a seguir um exemplo de utilização de uma varinha para navegação no ambiente terrestre.

### 6.4.1 Tapete Mágico

#### Visão geral

Diferentes tipos de varinhas já foram utilizados para navegação em ambientes virtuais, que é uma tarefa muito comum [BKLJP04]. Usando tecnologia magnética, Ciger *et al.* [CGVT03] propuseram “*The Magic Wand*”, uma interface de usuário para interação com um ambiente virtual mostrado numa grande tela de projeção. O usuário voa num “tapete mágico”, explorando o cenário. A varinha é usada para apontar para regiões da cena, e um pequeno vocabulário de comandos de voz (denominados “feitiços” no trabalho) é utilizado para dar instruções ao simulador. O sistema reconhece quatro posturas (orientações) básicas da varinha. De acordo com os autores, ao contrário dos dados de orientação, a posição informada pelo rastreador é muito inaccurada (e portanto é descartada).

Usando um par de câmeras estéreo calibradas, Guo *et al.* [GKR07] projetaram a *Featured Wand*, uma varinha passiva com marcadores coloridos nos extremos e uma marcação em espiral no meio. Um sistema de visão computacional detecta a posição e a orientação da varinha no espaço tridimensional a 9 quadros por segundo. Os autores utilizaram a varinha para navegação em ambientes virtuais: incliná-la para cima ou para baixo controla o empinamento do *viewpoint*, e rotacioná-la em torno do próprio eixo controla o balanceio. Elevá-la ou rebaixá-la controla a elevação, e movimentos laterais controlam o cabeceio.

A XWand [WS03] é um objeto portátil em forma de varinha usado em ambientes inteligentes. A ideia é controlar dispositivos apontando para eles e executando gestos simples. O hardware proposto inclui, entre outros componentes: sensores, um LED infravermelho e um microcontrolador. Os autores combinam os dados de um acelerômetro e de um magnetômetro para determinar a orientação 3D da varinha. Um par de câmeras calibradas e equipadas com filtros infravermelhos é utilizada para estimar, por triangulação, a posição 3D da varinha. Apesar de a solução proposta fornecer uma saída com 6 graus de liberdade, uma limitação do trabalho é a sua alta complexidade.

Cabral *et al.* [CRdS<sup>+</sup>12] desenvolveram uma aplicação colaborativa que funciona numa CAVE equipada com um sistema Optitrack de quatro câmeras infravermelhas. Um dos usuários da aplicação, denominado explorador, utiliza um *Wiimote* acoplado a um marcador reflexivo para apontar para onde ele quer navegar (tanto a posição quanto a orientação do *Wiimote* são rastreadas). Então, ele aperta um botão para deslocar-se para o local apontado. Essa metáfora de interação, chamada de *Point and Go*, foi utilizada num ambiente virtual fechado (quer dizer, sem céu).

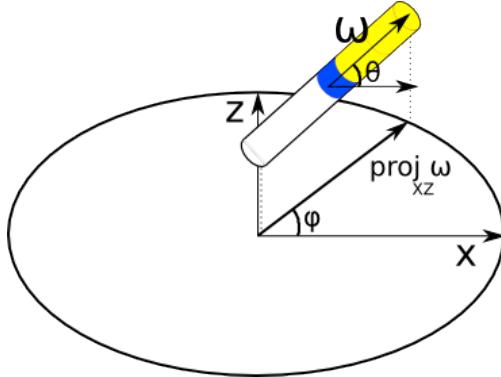
---

<sup>11</sup>Preferimos a expressão “second nature” do inglês.

## Interação

Aplicamos a varinha mágica como dispositivo de entrada para a atividade de navegação em ambientes virtuais. Como demonstração, construímos o Tapete Mágico, um simulador para navegação no globo terrestre. Objetivamos que, por meio do simulador, o usuário possa sobrevoar qualquer lugar do planeta. Isto posto, desejamos que:

1. O tapete permita que o usuário viaje pela Terra (Figs. 6.18a e 6.18b);
2. A orientação da varinha controle o *viewpoint* (Figs. 6.19a e 6.19b);
3. Encantamentos (feitiços) regulem a altitude do tapete (Figs. 6.20a e 6.20b).



**Figura 6.17:** Orientação da varinha.

A orientação da varinha controla duas características do *viewpoint*: o cabeceio e a movimentação (ver Fig. 6.17). Mais precisamente, seja  $\omega = [\omega_x \omega_y \omega_z]^t$  a orientação da varinha, com  $\|\omega\| = 1$ . Então, a cada *framestep* o simulador:

1. Atribui o valor  $\varphi + \varphi_0^{12}$  ao cabeceio, onde  $\varphi = \text{atan}2(\omega_z, \omega_x)$ ;
2. Avança o *viewpoint* se  $-\tan(\theta_{lim}) \leq \tan(\theta) \leq \tan(\theta_{lim})$ , onde  $\theta = \sin^{-1}(\omega_y)$  e  $\theta_{lim}$  é empírico<sup>13</sup>.

Importante característica a ser considerada quando se projeta este tipo de interface de usuário é a posição de descanso. A posição de descanso é importante para estabelecer uma situação na qual não há interação com o sistema, de modo que o usuário não se canse após um curto período de interação [GM10]. Para descansar, o usuário pode deixar a varinha “em pé” (i.e.,  $\theta \approx 90^\circ$ ), instruindo o sistema a parar o vôo. Se a varinha estiver “deitada” (i.e.,  $\theta \approx 0^\circ$ ), o *viewpoint* avançará.

Três encantamentos são suportados pelo simulador:  $\wedge$ ,  $\vee$  e  $\bigcirc$ . A tabela 6.2 associa os encantamentos com as ações desempenhadas pelo simulador.

Encantamento	Ação desempenhada
$\wedge$	Aumenta altitude.
$\vee$	Diminui altitude.
$\bigcirc$	Gira cabeceio em $180^\circ$ .

**Tabela 6.2:** Como os encantamentos são mapeados para ações do simulador.

O simulador foi construído utilizando Cesium, um *Virtual Globe & Map Engine* de código aberto feito em WebGL<sup>14</sup>.

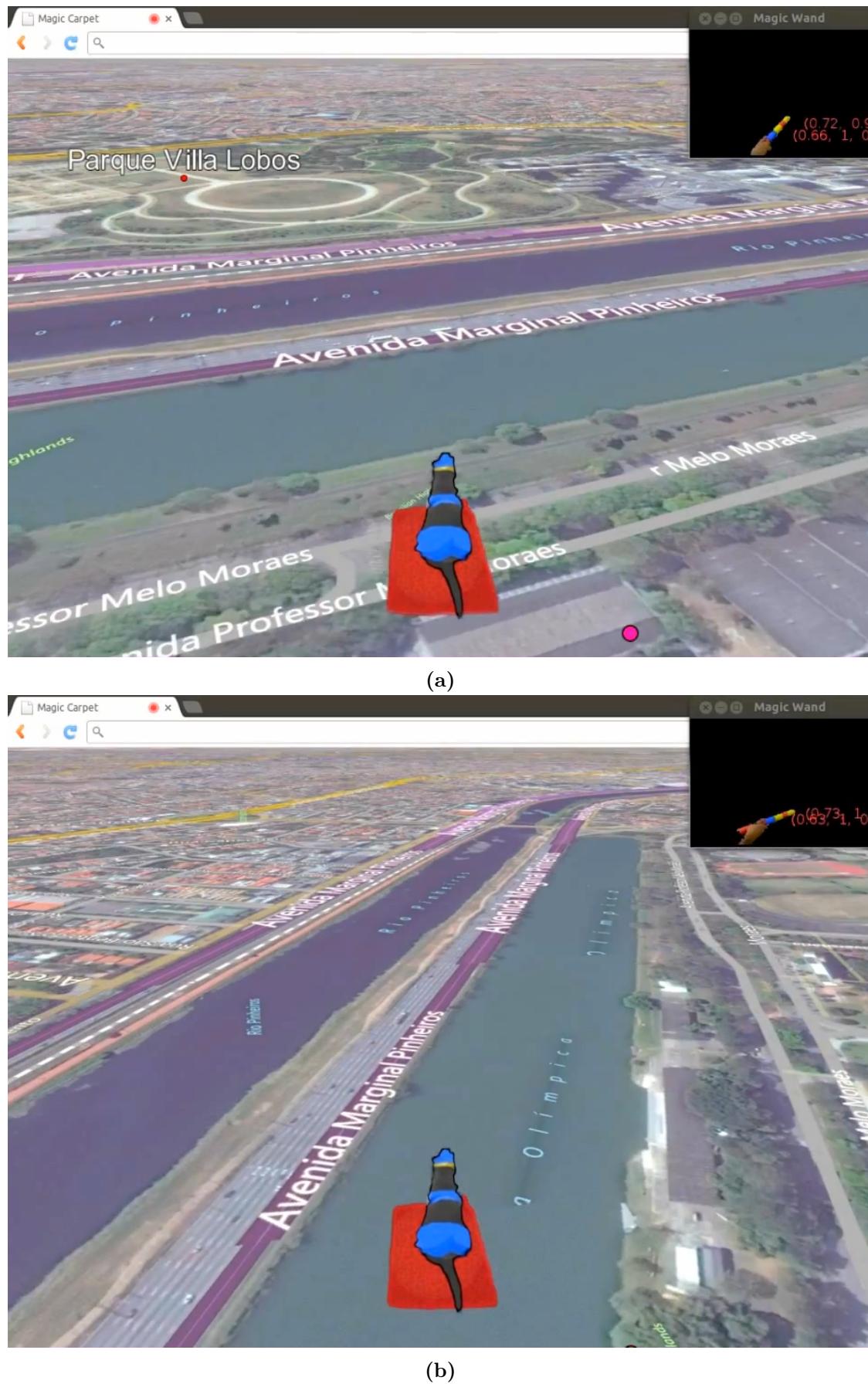
<sup>12</sup> $\varphi_0 = 1,23\pi$ .

<sup>13</sup> $\theta_{lim} = 45^\circ$ .

<sup>14</sup><https://cesiumjs.org/>. Último acesso: 15 de março de 2015.



Figura 6.18: Navegação num ambiente virtual.



**Figura 6.19:** Usando a varinha para controlar o viewpoint.



**Figura 6.20:** Um encantamento.



# Capítulo 7

## Conclusão

Neste trabalho de mestrado, apresentamos a **Superfície Mágica**: um sistema que transforma uma superfície comum num espaço interativo multi-toque. Inserimos o mundo digital no ambiente físico, explorando formas de interação inspiradas nos atos de desenhar, gesticular e agarrar. Procuramos extrapolar as fronteiras tradicionais de interação, imaginando e concebendo possibilidades para sistemas interativos futuros.

Do ponto de vista prático, desenvolvemos um protótipo que identifica, numa superfície, toques de dedos de mãos, de canetas coloridas e de um apagador, além de suportar o uso de uma varinha mágica. Nosso protótipo executa diversos aplicativos, cujas áreas de aplicação variam desde a educação até a arte interativa e o entretenimento. A instalação do sistema depende apenas de equipamentos portáteis, de baixo custo e amplamente disponíveis no varejo: um sensor Microsoft Kinect, um projetor de vídeo e um computador pessoal. Isto posto, foi necessário desenvolver novos algoritmos baseados em câmeras RGBD para detecção, classificação e rastreamento de objetos.

Nesta dissertação, a discussão de fundamentos das áreas de interação humano-computador, computação ubíqua, visão computacional e processamento de imagens acompanha a apresentação de trabalhos técnicos relacionados. A discussão de algumas tecnologias precede a descrição das principais contribuições deste projeto de mestrado.

As principais contribuições deste trabalho podem ser divididas em três esferas: o desenvolvimento de um sistema portátil que transforma uma superfície comum num espaço interativo multi-toque, a ampliação de tal espaço por meio de mecanismos que possibilitem interação 3D e o design e a implementação de aplicações interativas como provas de conceito da tecnologia.

Transformar uma superfície comum, como uma parede ou mesa, num ambiente multi-toque envolveu a pesquisa, implementação e adaptação de algoritmos de visão computacional e processamento de imagens presentes na literatura. Simulamos uma superfície multi-toque usando técnicas que trabalham com uma imagem de profundidade. Com base em trabalhos anteriores de Wilson [Wil10] e de Xiao, Harrison e Hudson [XHH13], computamos um modelo de fundo, capturamos os artefatos dispostos num certo intervalo acima do fundo e fazemos uma filtragem dos *blobs* correspondentes. A classificação dos objetos é feita mediante um cálculo de cores. Após extrair as cores de pixels no entorno e convertê-las para o espaço L\*a\*b\*, obtemos um *cluster cromático* que codifica a distribuição de cores do artefato. Estendemos o método de Lee e Yoo [LY02], originalmente pensado como um classificador binário de cor de pele, e classificamos o artefato com base em uma métrica de densidade que propusemos - o que nos permitiu abrigar variadas classes. O rastreamento dos objetos é feito de forma gulosa, conforme descrito em Kramer *et al.* [KPH<sup>+</sup>12]. Finalmente, a normalização das coordenadas dos objetos é realizada por meio de matrizes de homografia [BK08].

A segunda contribuição deste mestrado abrange a interação 3D. Introduzimos a “varinha mágica”, um dispositivo de apontamento para interação 3D que é fácil de ser construído e que dispensa componentes eletrônicos. O trabalho envolveu a concepção e a implementação de um algoritmo de visão computacional que permite o rastreamento da varinha com 5 graus de liberdade. Dada uma imagem RGBD, extraímos o fundo usando os dados de profundidade e segmentamos os marcadores da varinha usando a imagem colorida. A segmentação se dá por meio da conversão da imagem

colorida do espaço RGB para o HSV e por uma limiarização no canal da matiz. Uma análise de componentes conexas é realizada em seguida, indicando as regiões de interesse. Para que a varinha seja usável em aplicações interativas, faz-se necessário normalizar a posição de seus marcadores. Tal normalização envolve a transformação dos dados do sistema de coordenadas 3D da câmera para um sistema 3D normalizado. Introduzimos um modelo matemático inspirado numa transformação afim e propusemos um procedimento de calibração que, pedindo apenas que o usuário faça movimentos com a varinha em frente à camera, permite a posterior determinação dos parâmetros do modelo. Em nossos testes, o algoritmo rodou em tempo-real, mostrando-se hábil para o uso em aplicações interativas. Por fim, além de apontar, vale destacar que a varinha também permite que o usuário realize “feitiços” - gestos capturados e reconhecidos pelo sistema.

A terceira contribuição deste trabalho foi explorar o uso, em variados contextos, das tecnologias aqui desenvolvidas. Buscamos alavancar o conhecimento mundano dos usuários na interação com o computador, seguindo a linha de raciocínio por trás das interfaces pós-WIMP [VD97]. Como prova de conceito, mostramos uma parede interativa, com as aplicações Lousa e Mapa Mágicos. Também tornamos o corpo do usuário num espaço interativo, prototipando para isso o Toque Mágico. Prosseguimos apresentando o Tangible Beats, uma divertida interface tangível na qual os usuários criam seus próprios ritmos musicais deslocando objetos coloridos numa mesa interativa. O projeto foi exibido ao público na Tenda Cultural e na Feira de Profissões da USP, tendo sido recebido com grande entusiasmo e curiosidade em ambas as ocasiões. Propusemos também o Tapete Mágico, uma aplicação 3D que possibilita o uso da varinha mágica para navegação num ambiente de realidade virtual. Contribuições à parte, é importante frisar que com a iniciativa de levar protótipos do laboratório para o público geral, tivemos a oportunidade de, além da pesquisa, impactar também o pilar de extensão da universidade.

Durante as atividades deste mestrado, publicamos cinco artigos:

1. Alexandre M. F. de Sousa, Anderson C. M. Tavares e Carlos H. Morimoto. Penspective: a Perspective-Aware Game Map Editor with Natural Interaction. Em *Proceedings of the 12th Brazilian Symposium on Games and Digital Entertainment*, SBGames 2013, páginas 17–20. [[dSTM13](#)]
2. Alexandre M. F. de Sousa e Carlos H. Morimoto. Tangible Beats: a collaborative user interface for creating musical beats. Em *Proceedings of the 13th Brazilian Symposium on Games and Digital Entertainment*, SBGames 2014, páginas 304–307. **FINALISTA<sup>1</sup>**. [[dSM14a](#)]
3. Alexandre M. F. de Sousa e Carlos H. Morimoto. Tangible Beats: uma interface tangível para expressão e exploração musical. Em *Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems*, IHC 2014, páginas 385–388. [[dSM14b](#)]
4. Alexandre M. F. de Sousa e Carlos H. Morimoto. 5\* Magic Wand: a RGBD camera-based 5 DoF user interface for 3D interaction. Em *Proceedings of the 17th Symposium on Virtual and Augmented Reality*, SVR 2015, páginas 15–22. **BEST PAPER<sup>2</sup>**. [[dSM15b](#)]
5. Alexandre M. F. de Sousa e Carlos H. Morimoto. 5\* Magic Wand: a RGBD camera-based 5 DoF pointing device for 3D interaction. *SBC Journal on Interactive Systems*, JIS, 6(2):22–29, 2015. [[dSM15a](#)]

Dado o caráter exploratório deste projeto, há amplo espaço para trabalhos futuros. Tecnicamente, ressaltamos a utilização de sensores RGBD mais recentes para a captura da interação. Em relação às interfaces desenvolvidas, destacamos a possibilidade de usá-las para controlar jogos eletrônicos e sistemas de realidade aumentada, além de validar e comparar o desempenho e a usabilidade das variadas formas de interação. Finalmente, do ponto de vista de criatividade e de experimentação, citamos a possibilidade de colaboração com pesquisadores das áreas de arte e design, bem como a exposição de trabalhos em eventos de arte e tecnologia.

<sup>1</sup>Ficou entre os três melhores artigos resumidos da trilha de arte & design.

<sup>2</sup>Ficou em 1º lugar entre os artigos completos da conferência.

# Referências Bibliográficas

- [AB91] Gregory D Abowd e Russell Beale. Users, systems and interfaces: A unifying framework for interaction. Em *HCI*, volume 91, páginas 73–87, 1991. [5](#)
- [AR11] Mattia Avancini e Marco Ronchetti. Using kinect to emulate an interactive whiteboard. Dissertação de Mestrado, University of Trento, 2011. [16](#)
- [AW10] Lisa Anthony e Jacob O Wobbrock. A lightweight multistroke recognizer for user interface prototypes. Em *Proceedings of Graphics Interface 2010*, páginas 245–252. Canadian Information Processing Society, 2010.
- [BF13] Matthias Baldauf e Peter Fröhlich. The augmented video wall: multi-user ar interaction with public displays. Em *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, páginas 3015–3018. ACM, 2013. [11](#)
- [Bin00] Kim Binsted. Sufficiently advanced technology: using magic to control the world. Em *CHI'00 Extended Abstracts on Human Factors in Computing Systems*, páginas 205–206. ACM, 2000. [39](#)
- [BK08] Gary Bradski e Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'reilly, 2008. [33](#), [67](#)
- [BKLJP04] Doug A Bowman, Ernst Kruijff, Joseph J LaViola Jr e Ivan Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2004. [6](#), [9](#), [61](#)
- [BMHS03] Rodney Berry, Mao Makino, Naoto Hikawa e Masumi Suzuki. The augmented composer project: The music table. Em *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, página 338. IEEE Computer Society, 2003. [54](#)
- [BO08] Peter Bennett e S O'Modhrain. The beatbearing: a tangible rhythm sequencer. Em *Proc. of NordiCHI*, volume 2008, 2008. [54](#)
- [CB03] Xiang Cao e Ravin Balakrishnan. Visionwand: interaction techniques for large displays using a passive wand tracked in 3d. Em *Proceedings of the 16th annual ACM symposium on User interface software and technology*, páginas 173–182. ACM, 2003.
- [CGVT03] Jan Ciger, Mario Gutierrez, Frederic Vexo e Daniel Thalmann. The magic wand. Em *Proceedings of the 19th spring conference on Computer graphics*, páginas 119–124. ACM, 2003. [39](#), [61](#)
- [CRdS<sup>+</sup>12] Marcio Cabral, Gabriel Roque, Douglas dos Santos, Luiz Paulucci e Marcelo Zuffo. Point and go: Exploring 3d virtual environments. Em *3D User Interfaces (3DUI), 2012 IEEE Symposium*, páginas 183–184. IEEE, 2012. [10](#), [39](#), [61](#)
- [DFAB04] Alan Dix, Janet Finlay, Gregory D. Abowd e Russell Beale. *Human-Computer Interaction*. Pearson Education Limited, 3a edição, 2004. [1](#), [2](#), [5](#), [6](#), [10](#)

- [DP11] Bruno Campagnolo De Paula. Adaptando e desenvolvendo jogos para uso com o microsoft kinect. *Proceedings of SBGames*, 2011. [23](#)
- [dSJ12] Wedson Gomes da Silveira Junior. Manipulação de objetos 3d em ambientes colaborativos com o uso do dispositivo kinect. Dissertação de Mestrado, Universidade Federal de Uberlândia, 2012.
- [dSM14a] Alexandre M. F. de Sousa e Carlos H. Morimoto. Tangible beats: a collaborative user interface for creating musical beats. Em *Proceedings of the 13th Brazilian Symposium on Games and Digital Entertainment, SBGames 2014*, páginas 304–307. Sociedade Brasileira de Computação, 2014. [68](#)
- [dSM14b] Alexandre M. F. de Sousa e Carlos H. Morimoto. Tangible beats: uma interface tangível para expressão e exploração musical. Em *Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems, IHC 2014*, páginas 385–388. Sociedade Brasileira de Computação, 2014. [68](#)
- [dSM15a] Alexandre M. F. de Sousa e Carlos H. Morimoto. 5\* magic wand: a rgbd camera-based 5 dof pointing device for 3d interaction. *SBC Journal on Interactive Systems, JIS*, 6(2):22–29, 2015. [68](#)
- [dSM15b] Alexandre M. F. de Sousa e Carlos H. Morimoto. 5\* magic wand: a rgbd camera-based 5 dof user interface for 3d interaction. Em *Proceedings of the 17th Symposium on Virtual and Augmented Reality, SVR 2015*, páginas 15–22. IEEE, 2015. [68](#)
- [dSTM13] Alexandre M. F. de Sousa, Anderson C. M. Tavares e Carlos H. Morimoto. Penspective: a perspective-aware game map editor with natural interaction. Em *Proceedings of the 12th Brazilian Symposium on Games and Digital Entertainment, SBGames 2013*, páginas 17–20. Sociedade Brasileira de Computação, 2013. [68](#)
- [DSUL68] David M. Dresdner, John A. Spiech, Gerald M. Usland e C. G. Latterner. *Uma Introdução Programada ao PERT*. Livraria Pioneira Editora, 3a edição, 1968. [4](#)
- [EBG<sup>+</sup>92] Scott Elrod, Richard Bruce, Rich Gold, David Goldberg, Frank Halasz, William Janssen, David Lee, Kim McCall, Elin Pedersen, Ken Pier et al. Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. Em *Proceedings of the SIGCHI conference on Human factors in computing systems*, páginas 599–607. ACM, 1992. [15](#)
- [For10] J Fornari. Gestural controlled virtual percussion instruments. Em *IX Brazilian Symposium on Games and Digital Entertainment (SBGames 2010), Florianópolis, SC-Brazil*, 2010. [54](#)
- [GKR07] Feng Guo, Don Kimber e Eleanor G Rieffel. Featured wand for 3d interaction. Em *ICME*, páginas 2230–2233, 2007. [39](#), [61](#)
- [GM10] Silvia E Ghirotti e Carlos H Morimoto. Um sistema de interação baseado em gestos manuais tridimensionais para ambientes virtuais. Em *Proceedings of the IX Symposium on Human Factors in Computing Systems*, páginas 159–168. Brazilian Computer Society, 2010. [8](#), [62](#)
- [GV08] Jonas Gomes e Luiz Velho. *Fundamentos da computação gráfica*. IMPA, 2008. [11](#), [12](#)
- [Han05] Jefferson Y Han. Low-cost multi-touch sensing through frustrated total internal reflection. Em *Proceedings of the 18th annual ACM symposium on User interface software and technology*, páginas 115–118. ACM, 2005. [11](#)

- [HGJ<sup>+</sup>13] Martin Henschke, Tom Gedeon, Richard Jones, Sabrina Caldwell e Dingyun Zhu. Wands are magic: a comparison of devices used in 3d pointing interfaces. Em *Human-Computer Interaction-INTERACT 2013*, páginas 512–519. Springer, 2013. [61](#)
- [Hil06] Idar Douglas Hillgaard. The magic wand. Dissertação de Mestrado, Trinity College, 2006. [39](#)
- [HKH11] Daniel Herrera, Juho Kannala e Janne Heikkilä. Accurate and practical calibration of a depth and color camera pair. Em *Computer Analysis of Images and Patterns*, páginas 437–445. Springer, 2011. [23](#), [24](#)
- [HV09] Jordan Hochenbaum e Owen Vallis. Bricktable: A musical tangible multi-touch interface. Em *Proceedings of Berlin Open Conference &09. Berlin, Germany*, 2009. [11](#)
- [HWD10] Chris Harrison, Jason Wiese e Anind K Dey. Achieving ubiquity: The new third wave. *MultiMedia, IEEE*, 17(3):8–12, 2010. [1](#)
- [IML01] Hiroshi Ishii, Ali Mazalek e Jay Lee. Bottles as a minimal interface to access digital information. Em *CHI'01 extended abstracts on Human factors in computing systems*, páginas 187–188. ACM, 2001. [53](#)
- [Ish08] Hiroshi Ishii. Tangible bits: beyond pixels. Em *Proceedings of the 2nd international conference on Tangible and embedded interaction*, páginas xv–xxv. ACM, 2008. [7](#), [11](#), [53](#)
- [IU97] Hiroshi Ishii e Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. Em *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, páginas 234–241. ACM, 1997. [7](#)
- [JGAK07] Sergi Jordà, Günter Geiger, Marcos Alonso e Martin Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. Em *Proceedings of the 1st international conference on Tangible and embedded interaction*, páginas 139–146. ACM, 2007. [7](#), [8](#), [54](#)
- [JGH<sup>+</sup>08] Robert JK Jacob, Audrey Girouard, Leanne M Hirshfield, Michael S Horn, Orit Shaer, Erin Treacy Solovey e Jamie Zigelbaum. Reality-based interaction: a framework for post-wimp interfaces. Em *Proceedings of the SIGCHI conference on Human factors in computing systems*, páginas 201–210. ACM, 2008. [9](#)
- [KBBC05] Martin Kaltenbrunner, Till Bovermann, Ross Bencina e Enrico Costanza. Tuio: A protocol for table-top tangible user interfaces. Em *Proc. of the The 6th Int'l Workshop on Gesture in Human-Computer Interaction and Simulation*, páginas 1–5, 2005. [44](#)
- [KOR<sup>+</sup>02] David M Krum, Olugbenga Omoteso, William Ribarsky, Thad Starner e Larry F Hodges. Speech and gesture multimodal control of a whole earth 3d visualization environment. Em *ACM International Conference Proceeding Series*, volume 22, páginas 195–200, 2002.
- [KPD12] V Khan, Marije Pekelharing e Nils Desle. Efficient navigation in virtual environments: A comparative study of two interaction techniques: The magic wand vs. the human joystick. Em *Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on*, páginas 1–5. IEEE, 2012. [39](#), [61](#)
- [KPH<sup>+</sup>12] Jeff Kramer, Matt Parker, Daniel Herrera, Nicolas Burrus e Florian Echtler. *Hacking the Kinect*. Apress, 2012. [33](#), [35](#), [43](#), [48](#), [67](#)
- [LBS85] SK Lee, William Buxton e KC Smith. A multi-touch three dimensional touch-sensitive tablet. Em *ACM SIGCHI Bulletin*, volume 16, páginas 21–25. ACM, 1985. [10](#)

- [Lee08] Johnny Chung Lee. Hacking the nintendo wii remote. *Pervasive Computing, IEEE*, 7(3):39–45, 2008. [6](#), [15](#), [16](#)
- [Li10] Yang Li. Protractor: a fast and accurate gesture recognizer. Em *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, páginas 2169–2172. ACM, 2010. [8](#)
- [LY02] Jae-Young Lee e Suk I Yoo. An elliptical boundary model for skin color detection. Em *Proc. of the 2002 International Conference on Imaging Science, Systems, and Technology*. Citeseer, 2002. [34](#), [35](#), [67](#)
- [MA07] Sushmita Mitra e Tinku Acharya. Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3):311–324, 2007. [8](#)
- [Mou10] Ana Maria Moutinho. Inteligência ambiente: contributo para a conceptualização de “parede inteligente”. Dissertação de Mestrado, Universidade de Lisboa, Faculdade de Belas-Artes, 2010. [12](#)
- [MR97] Nobuyuki Matsushita e Jun Rekimoto. Holowall: designing a finger, hand, body, and object sensitive wall. Em *Proceedings of the 10th annual ACM symposium on User interface software and technology*, páginas 209–210. ACM, 1997.
- [Nob12] Joshua Noble. *Programming Interactivity*. O'Reilly Media, 2012. [25](#)
- [Ovi03] Sharon Oviatt. Multimodal interfaces. *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications*, páginas 286–304, 2003. [6](#)
- [PB07] Letícia Perani e Renato Teixeira Bressan. Wii will rock you: Nintendo wii e as relações entre interatividade e corpo nos videogames. Em *VI Brazilian Symposium on Computer Games and Digital Entertainment SBGames, São Leopoldo, RS*, 2007. [6](#), [7](#)
- [PdBBM05] Alexandre Souza Perucia, Antônio Córdova de Berthêm, Guilherme Lage Bertschinger e Roberto Ribeiro Castro Menezes. *Desenvolvimento de jogos eletrônicos*. Novatec, 2005.
- [Pin01] Claudio Pinhanez. The everywhere displays projector: A device to create ubiquitous graphical interfaces. Em *Ubicomp 2001: Ubiquitous Computing*, páginas 315–331. Springer, 2001. [18](#), [20](#)
- [Pin03] Claudio Pinhanez. Creating ubiquitous interactive games using everywhere displays projectors. Em *Entertainment Computing*, páginas 149–156. Springer, 2003.
- [RBTH10] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux e John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. Em *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, páginas 2155–2162. IEEE, 2010. [11](#), [23](#)
- [RDH09] Jason L. Reisman, Philip L. Davidson e Jefferson Y. Han. Generalizing multi-touch direct manipulation. Em *SIGGRAPH 2009: Talks*, SIGGRAPH '09, páginas 12:1–12:1, New York, NY, USA, 2009. ACM. [1](#)
- [RSP07] Yvonne Rogers, Helen Sharp e Jenny Preece. *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 2007.

- [RWC<sup>+</sup>98] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin e Henry Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. Em *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, páginas 179–188. ACM, 1998. [18](#), [19](#)
- [SJ07] Andrew Sears e Julie A Jacko. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. CRC Press, 2007.
- [SP05] Ben Shneiderman e Catherine Plaisant. Designing the user interface 4 th edition, 2005. [9](#), [10](#)
- [ST09] Filipe Vieira da Silva e José Manuel Torres. Avaliação da utilização em sala de aula de um quadro digital interactivo baseado no wiimote. Em *Revista da Faculdade de Ciência e Tecnologia. Porto: Edições Universidade Fernando Pessoa*, 2009. [16](#)
- [Sut64] Ivan E Sutherland. Sketch pad a man-machine graphical communication system. Em *Proceedings of the SHARE design automation workshop*, páginas 6–329. ACM, 1964. [15](#)
- [TBI97] Lloyd N Trefethen e David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997. [44](#)
- [TIK<sup>+</sup>09] Stuart Taylor, Shahram Izadi, David Kirk, Richard Harper e Armando Garcia-Mendoza. Turning the tables: an interactive surface for vjing. Em *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, páginas 1251–1254. ACM, 2009. [11](#)
- [TKCZ12] Fernando Teubl, Celso Kurashima, Marcio Cabral e Marcelo Zuffo. Fastfusion: A scalable multi-projector system. Em *Virtual and Augmented Reality (SVR), 2012 14th Symposium on*, páginas 26–35. IEEE, 2012. [13](#)
- [VD97] Andries Van Dam. Post-wimp user interfaces. *Communications of the ACM*, 40(2):63–67, 1997. [9](#), [68](#)
- [Wei91] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991. [1](#), [7](#)
- [Wil10] Andrew D Wilson. Using a depth camera as a touch sensor. Em *ACM international conference on interactive tabletops and surfaces*, páginas 69–72. ACM, 2010. [22](#), [23](#), [32](#), [41](#), [67](#)
- [WS03] Andrew D Wilson e Steven Shafer. Xwand: Ui for intelligent spaces. Em *Proceedings of the SIGCHI conference on Human factors in computing systems*, páginas 545–552. ACM, 2003. [39](#), [61](#)
- [WWL07] Jacob O Wobbrock, Andrew D Wilson e Yang Li. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. Em *Proceedings of the 20th annual ACM symposium on User interface software and technology*, páginas 159–168. ACM, 2007. [8](#), [45](#)
- [XHH13] Robert Xiao, Chris Harrison e Scott E Hudson. Worldkit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. Em *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, páginas 879–888. ACM, 2013. [1](#), [17](#), [22](#), [23](#), [32](#), [67](#)
- [Zha12] Zhengyou Zhang. Microsoft kinect sensor and its effect. *Multimedia, IEEE*, 19(2):4–10, 2012. [23](#), [24](#)