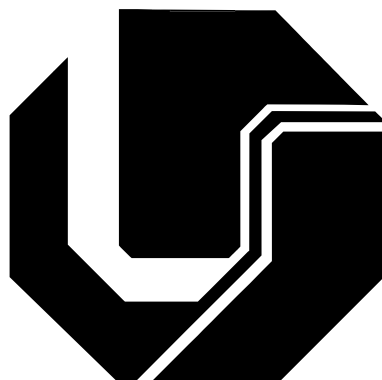


**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



Uso de Interfaces Naturais na modelagem de objetos virtuais

Fábio Henrique Monteiro Oliveira

Uberlândia

2013

Fábio Henrique Monteiro Oliveira

Uso de Interfaces Naturais na modelagem de objetos virtuais

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do grau de Mestre em Ciências.

Área de Concentração: Processamento da Informação

Prof. Alexandre Cardoso, Dr. - Orientador (UFU)

Prof. Edgard A. Lamounier Jr, Dr. - Co-orientador (UFU)

Uberlândia

2013

Fábio Henrique Monteiro Oliveira

Uso de Interfaces Naturais na modelagem de objetos virtuais

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do grau de Mestre em Ciências.

Área de Concentração: Processamento da Informação

Uberlândia, 05 de agosto de 2013.

Banca examinadora:

Prof. Alexandre Cardoso, Dr.
Orientador - UFU

Prof. Edgard A. Lamounier Jr, Dr.
Co-orientador - UFU

Prof.^a Ana Claudia Patrocínio, Dra
Examinadora - UFU

Prof. Claudio Kirner, Dr.
Examinador - UNIFEI

Agradecimentos

Agradeço a Deus, pai misericordioso, pela agraciada oportunidade, de por meio deste trabalho estar concretizando mais um passo nessa infinita caminhada que possui n vertentes.

Rendo ainda singelos agradecimentos a família (Ângela, Jahudehir, Guilherme, Gabrielle) que me acolheu, visto que o apoio e o incentivo nunca me faltou. E não menos a minha companheira (Rafaela) que está sempre presente a somar por meio do amor.

Aproveito a ocasião para expressar a minha gratidão pelos inestimáveis amigos e amigas que diariamente me apoiam.

Aos professores Alexandre e Edgard, pela confiança, orientação e amizade.

Aos colegas e amigos do Programa de Pós-graduação em Engenharia Elétrica e do Laboratório de Computação Gráfica pelo apoio e pelos incontáveis momentos de alegria.

Ao professor Claudio e a professora Ana pela participação na banca avaliadora, da qual resultou em agregadoras observações e considerações.

Ao CNPq e a FAPEMIG por fomentar a pesquisa.

E a todos aqueles que diretamente ou indiretamente contribuíram para a realização deste trabalho.

Meus sinceros agradecimentos a todos vocês!

"...quando estou fraco então sou forte."

Paulo de Tarso

Resumo

OLIVEIRA, F. H. M. *Uso de Interfaces Naturais na modelagem de objetos virtuais*. Dissertação (Mestrado) - Universidade Federal de Uberlândia, Agosto 2013.

As pesquisas na área de interfaces gestuais vêm crescendo significativamente. Em especial, após o desenvolvimento de sensores que podem capturar movimentos corporais com precisão. Como consequência, surgem diversos campos para a aplicação destas tecnologias. Dentre eles, destaca-se o setor de modelagem 3D, o qual é marcado por possuir programas robustos. Entretanto, estes são muitas vezes ausentes de uma interface homem-computador facilitadora. Isto porque a interação, normalmente, é viabilizada pelo mouse com 2 graus de liberdade. Devido a estas limitações, tarefas frequentes como rotacionar o ponto de vista da cena e transladar um objeto são assimiladas pelo usuário com dificuldade. Este desconforto perante a usual e complexa interface dos programas para modelagem 3D é um dos fatores que culminam na desistência de seu uso. Neste contexto, *Natural User Interfaces* se destacam, por melhor explorar os gestos naturais humanos, a fim de promover uma interface mais intuitiva. Neste trabalho é apresentado um sistema que permite ao usuário realizar a modelagem 3D por meio de poses e gestos com as mãos provendo uma interface com 3 graus de liberdade. Uma avaliação foi conduzida com 10 pessoas, a fim de validar a estratégia e a aplicação proposta. Nesta avaliação os participantes reportaram que o sistema tem potencial para se tornar uma interface inovadora, a despeito de suas limitações. Em geral, a abordagem de rastreamento das mãos para modelagem 3D parece ser promissora e merece uma investigação mais aprofundada.

Palavras-chave: Interface Natural. Computação Ubíqua. Sensor de Profundidade. Modelagem 3D.

Abstract

OLIVEIRA, F. H. M. *A 3D Natural User Interface on virtual object modeling*. Master's thesis - Federal University of Uberlândia, August 2013.

The researches about gestural interfaces have been grown significantly. In particular, after the development of sensors that can accurately capture bodily movements. Consequently, several fields arise for the application of these technologies. Among them stands the 3D modeling industry, which is characterized by having robust software. However, these software often lack a facilitator human-computer interface. This happens since the interaction is usually enabled by 2 degrees of freedom mouse. Due to these limitations, common tasks such as rotate the scene viewpoint or move an object are hardly assimilated by users. This discomfort with the usual and complex 3D modeling software interface is one of the reasons that lead to quit its use. In this context, Natural User Interfaces stand out by exploring the natural human gestures in a better way, in order to promote a more intuitive interface. This work presents a system that allows the user to perform 3D modeling using poses and hand gestures, providing an interface with 3 degrees of freedom. An evaluation was conducted with 10 people, to validate the strategy and application proposed. In this evaluation the participants reported that the system has potential to become an innovative interface, despite its limitations. Overall, the hand tracking approach to 3D modeling seems to be promising and deserves further investigation.

Keywords: Natural User Interface. Ubiquitous Computing. Depth Sensor. 3D Modeling.

Publicações

OLIVEIRA, F. H. M. ; CARDOSO, A. ; LAMOUNIER JÚNIOR, E. A. Proposta de Interação Natural na construção de ambientes de Realidade Virtual e Aumentada. In: VIII Workshop de Realidade Virtual e Aumentada, 2011, Uberaba - MG. Anais do WRVA'2011. Uberaba: IFTM, 2011.

Sumário

Lista de Figuras	xiv
Lista de Tabelas e Quadros	xv
Lista de Abreviaturas e Siglas	xvii
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos e metas	3
1.3 Estrutura da Dissertação	3
2 Fundamentos	5
2.1 Introdução	5
2.2 Interação Humano-Computador	5
2.2.1 Tipos de interação	6
2.2.1.1 Interação de dados	6
2.2.1.2 Interação por imagem	6
2.2.1.3 Interação por voz	7
2.2.1.4 Interação inteligente	7
2.2.2 Evolução das interfaces com o usuário	7
2.2.2.1 Interface por linha de comando	8
2.2.2.2 Interface gráfica	8
2.2.2.3 Interface natural	9
2.3 Dispositivos para suporte de Interfaces Naturais	10
2.3.1 Nintendo® Wii Remote™	10
2.3.2 PlayStation® Move	12

2.3.3	Microsoft® Kinect™	13
2.3.3.1	Sobre o hardware	14
2.3.3.2	Sobre o software	15
2.3.4	Outros dispositivos	16
2.4	Ferramentas para manipulação de sensores de profundidade de baixo custo	17
2.4.1	OpenKinect	17
2.4.2	OpenNI®	17
2.4.2.1	NITE™	18
2.4.3	Microsoft® Kinect™ for Windows® SDK	19
2.4.4	Resumo comparativo	20
2.5	Considerações finais	21
3	Trabalhos Correlatos	22
3.1	Introdução	22
3.2	Manipulação de objetos 3D em aplicações de RA por meio da movimentação da mão	22
3.3	Fusion4D - Interface natural e imersiva para manipulação de objetos 3D	24
3.4	Non-contact multi-hand gestures interaction techniques for architectural design in a Virtual Environment	25
3.5	Bimanual Natural User Interaction for 3D modelling application using stereo computer vision	28
3.6	Resumo comparativo dos trabalhos analisados	30
3.7	Considerações finais	32
4	Especificação do sistema proposto	33
4.1	Introdução	33
4.2	Requisitos do sistema	33
4.3	Diagramas de casos de uso	33
4.4	Diagrama de classes	40
4.5	Arquitetura do sistema	41
4.6	Considerações finais	42

5	Detalhes da implementação	43
5.1	Introdução	43
5.1.1	Tecnologias de apoio	43
5.2	Ambiente de desenvolvimento	44
5.3	Processo de desenvolvimento	44
5.3.1	Captura de imagens	45
5.3.2	Determinação de gestos	45
5.3.3	Comunicação com o Blender	47
5.3.4	Processamento em tempo real	47
5.3.4.1	Receptor	48
5.3.4.2	Processador	48
5.3.4.3	Consumidor	49
5.4	<i>Blender operator</i>	49
5.4.1	Mapeamento do sistema de coordenadas	50
5.4.2	Processo de interação	51
5.4.2.1	Metáfora de interação e seus parâmetros	52
5.4.2.2	Navegação	52
5.4.2.3	Seleção	53
5.4.2.4	Manipulação	54
5.5	Considerações finais	54
6	Discussão dos resultados	55
6.1	Introdução	55
6.2	Protótipo	55
6.3	Avaliação da interface	55
6.3.1	Avaliação heurística	56
6.3.2	Etapas de um teste	56
6.3.3	Medidas de performance	58
6.4	Experimentos	59
6.4.1	Primeira etapa	59
6.4.2	Segunda etapa	60
6.5	Análise dos resultados	62
6.6	Considerações Finais	64

<i>SUMÁRIO</i>	xii
7 Conclusões e trabalhos futuros	66
7.1 Introdução	66
7.2 Considerações finais	66
7.3 Trabalhos futuros	68
Referências Bibliográficas	69
A Manual do Usuário - NI2Blender	76

Lista de Figuras

2.1	Evolução das interfaces humano-computador	8
2.2	Versões do controle para o Nintendo® Wii™	11
2.3	Componentes que integram o PS Move	12
2.4	Diagrama do ambiente cliente/servidor do Move.Me	13
2.5	Sensor Kinect™ e seus componentes	14
2.6	Divisão e reconhecimento das partes do corpo humano	16
2.7	Capacidade e modos de rastreamento	16
2.8	Abstração das camadas do <i>framework</i> OpenNI®	18
2.9	Controle baseado em gestos com as mãos	19
3.1	Sistema desenvolvido por Santos [Santos 2011].	23
3.2	Arquitetura do sistema [Matsumura e Sonnino 2011].	24
3.3	Ampliação de um objeto virtual [Matsumura e Sonnino 2011].	25
3.4	Sistema proposto por [Shiratuddin e Wong 2011].	26
3.5	Alguns dos gestos propostos por [Shiratuddin e Wong 2011].	27
3.6	Seis poses utilizadas para interação [Yang et al. 2012].	28
3.7	Tarefa realizada [Yang et al. 2012].	29
4.1	Diagrama de casos de uso.	35
4.2	Diagrama de classes.	40
4.3	Arquitetura do sistema.	41
5.1	Passos para iniciar uma sessão.	46
5.2	Formato dos dados que trafegam do produtor para o consumidor. . . .	47
5.3	Esquema para suportar a comunicação e processamento em tempo real.	48
5.4	Sistemas de coordenadas.	51

5.5	Projeção em formato de cone para seleção de objeto.	53
6.1	Configuração física do sistema.	59
6.2	Resultado da avaliação preliminar do protótipo.	60
6.3	Exemplo de tarefa usada no segundo experimento.	62
6.4	Resultados acerca das funcionalidades.	62
6.5	Resultados acerca da usabilidade.	63
6.6	Resultados acerca da eficiência.	63
A.1	Passos para iniciar uma sessão.	77
A.2	Sistemas de coordenadas.	77

Lista de Tabelas e Quadros

2.1	Resumo comparativo das ferramentas apresentadas.	20
3.1	Resumo comparativo dos trabalhos correlatos.	31
4.1	Requisitos funcionais do protótipo NI2Blender.	34
4.2	Requisitos não funcionais do protótipo NI2Blender.	34
4.3	Descrição do caso de uso “Calibrar usuário”	36
4.4	Descrição do caso de uso “Manipular objeto 3D”	37
4.5	Descrição do caso de uso “Manipular <i>viewport</i> ”	38
4.6	Descrição do caso de uso “Finalizar sessão”	39
6.1	Heurísticas adotadas (adaptado de [Nielsen 1993]).	57
6.2	Resumo comparativo dos trabalhos correlatos e do presente trabalho. .	64
A.1	<i>Feedback</i> visual do sistema através das mãos.	78
A.2	Instruções para aplicar <i>Zoom/Pan</i> na visão da cena.	79
A.3	Instruções para aplicar rotação na visão da cena.	80
A.4	Instruções para selecionar objeto.	81
A.5	Instruções para transladar objeto.	81
A.6	Instruções para redimensionar objeto.	82
A.7	Instruções para rotacionar objeto.	83

Lista de Abreviaturas e Siglas

3D	Tridimensional
API	<i>Application Programming Interface</i>
AV	Ambiente Virtual
CAD	<i>Computer-Aided Design</i>
EMG	Eletromiográfico
FIFO	<i>First In, First Out</i>
FPS	<i>Frames Per Second</i>
GPL	<i>General Public License</i>
GUI	<i>Graphical User Interface</i>
IHC	Interação Humano Computador
IR	<i>InfraRed</i>
LASER	<i>Light Amplification by Stimulated Emission of Radiation</i>
LED	<i>Light-Emitting Diode</i>
NITE	<i>Natural Interaction Technology for End-user</i>
NUI	<i>Natural User Interface</i>
OGRE	<i>Object-Oriented Graphics Rendering Engine</i>
OpenNI	<i>Open Natural Interaction</i>
PCA	<i>Principle Component Analysis</i>

PS	PlayStation®
RA	Realidade Aumentada
RF	Requisito Funcional
RNF	Requisito Não Funcional
SDK	<i>Software Development Kit</i>
TCP	<i>Transmission Control Protocol</i>
TOF	<i>Time-Of-Flight</i>
UML	<i>Unified Modeling Language</i>
USB	<i>Universal Serial Bus</i>
WIMP	<i>Windows, icons, menus and pointers</i>
WPF	<i>Windows Presentation Foundation</i>
WYSIWYG	<i>What You See Is What You Get</i>

Capítulo 1

Introdução

1.1 Motivação

É natural que o ser humano se comunique através de gestos, expressões e movimentos. O processo de descoberta do mundo se dá por meio da visão e manipulação de objetos físicos, o pressuposto fundamental é que deveria ser permitido interagir com a tecnologia do mesmo modo que o ser humano interage com outro da sua espécie e com o mundo real, como a evolução e a educação os ensinou a fazer [Valli 2008].

Historicamente, a linguagem utilizada entre pessoas e computadores tem sido determinada por restrições tecnológicas, e o homem tem se adaptado a estas linguagens [Chao 2009]. Dix afirma em [Dix et al. 2005] que a interação entre humanos e computadores ainda era uma tarefa difícil, porém hoje nota-se, graças às tecnologias como às empregadas em *smartphones* e *tablets*, que a fronteira entre humanos e computadores se torna cada vez mais sutil.

No sentido de tornar o acesso ou manipulação de informações digitais mais intuitivo, diversos trabalhos têm sido desenvolvidos. Em muitos deles, pode-se encontrar referências a uma modalidade de interação chamada Interação Natural. Valli, por exemplo, define em [Valli 2008] este conceito como sendo uma forma de permitir a manipulação de um ambiente, em que a necessidade de aprendizagem para a interação seja mínima, ou seja, a metáfora utilizada para interação deve ser tão simples, que qualquer pessoa possa utilizá-la com o mínimo de treinamento.

Thalen [Thalen e Voort 2011, Thalen 2011], que fez uma série de experimentos

procurando avaliar a utilização de programas para criação de aplicações interativas 3D, levantou o seguinte questionamento: "Que tipo de ferramenta os *designers* necessitam para criar suas próprias aplicações interativas 3D?". Em seu estudo de caso, pessoas com experiência em *design*, mas que ainda não utilizam, com frequência, ferramentas de modelagem 3D, receberam um treinamento e foram solicitadas a realizar tarefas que envolviam modelagem 3D e outras. O resultado foi: "Eu consigo fazer o trabalho, porém a ferramenta é bastante complexa, principalmente se for usá-la quatro ou cinco vezes no ano.", a grande queixa é que o processo não é intuitivo.

[Penn, Schoen & Berland 2011] realizaram um estudo em que se analisou a conscientização e a percepção das pessoas quanto ao uso e aplicação de tecnologias relacionadas com *Natural User Interface* (NUI) . Dentre os participantes, 85% concordam que as NUI podem colaborar nos vários setores de trabalho, entretenimento, educação, relações sociais, saúde e meio ambiente. Nos países em que a pesquisa foi feita as tecnologias relacionadas com NUI são vistas como o caminho para o futuro e afirmam que as mesmas estão se encaixando facilmente em suas vidas, sociedade e economia. Quanto aos entrevistados, esses declararam estar familiarizados com tecnologia *touchscreen* e simulações em 3D.

A interface é uma camada entre o utilizador e o objeto em uso. Nos últimos anos, diversas áreas que utilizam o computador como componente de trabalho, têm sido beneficiadas pela inserção e avanço das NUI, os trabalhos [Santos 2011, Matsumura e Sonnino 2011, Shiratuddin e Wong 2011] apresentam aplicações que fazem uso das mesmas. Destaca-se, ainda, o uso massivo das NUI nos consoles de videogame, por meio, por exemplo, de controles a distância baseados em gestos corporais.

Dado o atual cenário da Interação Humano Computador e do modo atual de interação vastamente utilizado em programas para modelagem 3D, nota-se a disparidade e a carência de estudo e desenvolvimento para esta área específica, visto que grande parte destes aplicativos emprega técnicas de interação da década de 90.

Sistemas para criação/interação com Ambientes Virtuais (AV) que fazem uso de interfaces não convencionais têm sido investigados por alguns autores nos últimos anos [Thalen e Voort 2011, Shiratuddin e Wong 2011, Dam 2012]. Dentre os desafios apontados por estes autores destacam-se a intuitividade e conforto durante a utilização do sistema, as definições dos gestos para interação e a resposta em tempo real destes

sistemas.

1.2 Objetivos e metas

O objetivo principal deste trabalho é desenvolver uma interface gestual alinhada com NUI, que faça uso de um sensor de profundidade como dispositivo de entrada, a fim de promover interações, baseadas em gestos corporais, com um ambiente para modelagem 3D.

Para tal, foram traçados os seguintes objetivos específicos:

- Desenvolver um sistema para captura e pré-processamento de poses e gestos realizados com as mãos;
- Desenvolver um sistema para processamento das poses e gestos, uma interface que responda aos mesmos e integrá-la com um software que permita conceber modelos virtuais 3D;
- Projetar a comunicação entre o sistema de captura e o de processamento de gestos;
- Habilitar a navegação na cena (através da rotação, translação e *zoom*) e a manipulação de objetos 3D (através da rotação, translação e redimensionamento), ambas por meio de poses e gestos com as mãos.

1.3 Estrutura da Dissertação

A presente dissertação é constituída de 7 capítulos, descritos a seguir.

No Capítulo 1 são apresentados a motivação para o desenvolvimento do trabalho, os objetivos e metas propostos e a estruturação do trabalho.

O Capítulo 2 são apresentados os principais fundamentos de interação humano-computador e tecnológicos relacionados ao trabalho desenvolvido.

O Capítulo 3 é apresentado o estado da arte da linha de pesquisa principal deste trabalho.

Nos Capítulos 4 e 5 são apresentados a especificação do sistema proposto e os principais detalhes de implementação.

O Capítulo 6 apresenta uma breve discussão dos resultados obtidos a partir do desenvolvimento do sistema proposto.

Finalmente, no Capítulo 7 são apresentadas as conclusões e as perspectivas de trabalhos futuros.

Capítulo 2

Fundamentos

2.1 Introdução

Neste capítulo serão abordados alguns dos aspectos relacionados à Interação Humano-Computador (IHC) e tecnologias relacionadas ao presente trabalho.

2.2 Interação Humano-Computador

A interação (com ou sem computador) pode ser considerada como um processo de transferência de informação [Dix et al. 2005], ou ainda um processo de entrada e saída de dados, em que comandos servem de entrada através de uma interface humano-computador e o computador mostra a saída para o usuário, sendo diversas as formas de entrada e saída de dados [Yang e Chen 2009].

A diversidade de dispositivos reflete o fato de que há muitos tipos de dados que precisam servir de entrada/saída em um sistema, assim como há muitos tipos de usuários. No princípio, o processamento em lote era comum até chegar aos dias de hoje, uma grande quantia de dados foi despejada e processada pelo computador. Atualmente, os computadores respondem em milissegundos e os sistemas de computador são integrados com diversos dispositivos [Dix et al. 2005].

2.2.1 Tipos de interação

Os modos de entrada e saída de dados entre humanos e computadores são vários, e os modos de interação também. Sendo divididos basicamente em interação de dados, interação por imagem, interação por voz e interação inteligente [Yang e Chen 2009, Chao 2009]. Nesta seção são apresentadas alguns tipos de interações em IHC, bem como os princípios de funcionamento das mesmas.

2.2.1.1 Interação de dados

A informação neste modo envolve figuras, cores e gráficos. Pode ser obtida por meio de questionamentos, menus de seleção, preenchimento de tabelas, entrada de palavras chave, escaneamento de códigos de barra, reconhecimento óptico de caracteres, gravação de voz e imagens e outras formas. É uma importante forma de interação humano-computador.

O processo geral de interação é:

1. O sistema requisita ao usuário os dados de entrada e por qual meio esses dados serão providos;
2. Então em resposta à entrada do usuário, o sistema fornece um *feedback* e apresenta-o em uma tela ou em outro dispositivo de saída de dados;
3. Ao mesmo tempo, o sistema checka a entrada do usuário e se algum erro for detectado o sistema solicita novamente a entrada de dados.

2.2.1.2 Interação por imagem

Os seres humanos utilizam, principalmente, o sistema visual para obter informações do mundo externo que são processadas e conseqüentemente reconhecidas. Assim, o ponto chave da interação por imagem é como fazer o computador percebê-las e identificá-las. Geralmente, essa habilidade pode ser dividida em três níveis: processamento de imagem, reconhecimento de imagem e percepção da imagem.

2.2.1.3 Interação por voz

A linguagem falada é sempre reconhecida como a forma mais natural, conveniente e eficiente para trocar informações. A interação por voz é um importante canal para a comunicação entre humanos e computadores e é importante meio de estudo sobre como interagir com computadores pela voz natural ou artificial. O estudo é multidisciplinar e envolve áreas como, linguística, psicologia, engenharia e computação.

2.2.1.4 Interação inteligente

Exceto para o uso da voz, as pessoas costumam usar gestos e movimentos do corpo para se expressarem. O comportamento humano é capaz de reforçar as habilidades de comunicação, e por vezes, pode desempenhar papéis importantes em que a voz de forma isolada não seria capaz.

A IHC inteligente é o processo de utilizar o computador para localizar e identificar seres humanos, rastrear seus movimentos corporais a fim de entender as ações e o comportamento humano, para então responder de forma inteligente. Deste tipo de interação nasce uma nova forma de interagir e um novo tipo de interface homem-computador, que preza a "facilidade no aprendizado e no uso" e é nomeada como a próxima geração de interface com o usuário.

2.2.2 Evolução das interfaces com o usuário

Desde sua aparição, as tecnologias relacionadas a interface humano-computador estão em constante evolução. Esta evolução tem como marco a invenção do mouse na década de 60 por Douglas Engelbart e na sequência o desenvolvimento das interfaces gráficas [Dix et al. 2005, Liu 2010]. A Figura 2.1 demonstra, simplificada, a evolução das interfaces humano-computador. Estas interfaces vêm sendo desenvolvidas a algumas décadas com pesquisas na área de interfaces multi-toque, reconhecimento de voz e visão computacional [Buxton 2013].

Ainda na década de 60, Ivan Sutherland apresenta o editor gráfico *Sketchpad* [Sutherland 1964], sendo o primeiro programa de computador a utilizar uma interface gráfica e o primeiro, também, a representar objetos tridimensionais em uma tela bidimensional.



Figura 2.1: Evolução das interfaces humano-computador [Matsumura e Sonnino 2011].

Nesta seção, são apresentadas descrições de alguns tipos de interface homem-computador, que fazem parte da história e contribuíram diretamente para se chegar ao patamar atual, bem como os princípios de funcionamento destas interfaces.

2.2.2.1 Interface por linha de comando

É tida como o ancestral de todas as interfaces com o computador, as interfaces por linha de comando foram o início da IHC, e até hoje são uma das formas mais eficazes para controlar o computador [Liu 2010]. Neste tipo de interface o usuário provê a entrada digitando palavras chave, que são interpretadas como comandos e o sistema responde imprimindo por exemplo, um texto no monitor [Dix et al. 2005].

2.2.2.2 Interface gráfica

As principais características das *Graphical User Interface* (GUI) segundo [Liu 2010, Dix et al. 2005] são:

- Metáfora de área de trabalho, baseada no mundo real e em coisas que já existem. A ideia é aproveitar o conhecimento dos usuários (tal como procedimentos comuns relacionados com um ambiente de trabalho em um escritório) a fim de reduzir o esforço cognitivo requerido;
- Tecnologia baseada em *Windows, icons, menus and pointers* (WIMP) ;
- Manipulação direta, é caracterizada pela visualização de objetos de interesse, ações incrementais e rápido *feedback*, reversibilidade das ações e a substituição

de linguagens de comando complexas por ações para manipular diretamente os objetos visíveis;

- *What You See Is What You Get* (WYSIWYG), é também conhecida como operações visuais, em que os usuários podem, através de uma tela, manipular diretamente o que será o produto final. WYSIWYG garante que o que é visto na tela e o que será impresso não terá diferença.

2.2.2.3 Interface natural

O conceito de interação natural tem sido definido de várias maneiras. Em geral, refere-se à exploração de habilidades naturais (familiar, inato, universal, transcultural e etc) do ser humano para controlar, de forma implícita ou explícita, um sistema de computador. As NUI visam prover uma experiência com o usuário onde a tecnologia é invisível. Efetivamente, tal definição inclui na prática qualquer atividade ou comportamento humano como por exemplo: gestos, manipulação de objetos físicos, manipulação de objetos virtuais, expressões faciais, movimentos corporais, linguagem natural (falada), língua de sinais e uso de metáforas do mundo real [Iacolina et al. 2010, Jain, Lund e Wixon 2011].

Características das NUI [Liu 2010]:

- Concepção centrada no usuário, visando atender as necessidades de diferentes usuários;
- Faz uso de um ou múltiplos canais sensoriais e motores para capturar as características complementares da intenção do usuário, a fim de aumentar a naturalidade da interação humano-computador;
- Inexata, pois as ações e pensamentos das pessoas muitas vezes não são precisos, portanto o computador deve ser capaz de compreender as solicitações humanas e procurar corrigir o seus enganos;
- Alta largura de banda para entrada de dados e rápida importação de grandes quantidades de informações, como a entrada e compreensão de voz e imagem;
- Interação baseada em voz, imagem e comportamento.

2.3 Dispositivos para suporte de Interfaces Naturais

Nesta seção são apresentados alguns dispositivos que dão suporte ao desenvolvimento de interfaces naturais, assim como as principais características de cada um.

A capacidade de reconhecer gestos é uma área que vem sendo pesquisada, pelo menos, desde a década de 80. Pesquisas pioneiras como *"Put-That-There: Voice and Gesture at the Graphics Interface"* [Bolt 1980] demonstraram já naquela época, a necessidade de desenvolver um interface natural e multimodal.

Nos últimos anos, este tipo de tecnologia recebeu grandes investimentos, graças a uma demanda cada vez maior por interfaces eficazes, intuitivas e baratas. Entre as interfaces naturais, destaca-se o conceito de *touchless* (interação à distância), no qual os comandos são realizados pelo movimento do corpo, gestos e reconhecimento de voz, sem a necessidade de contato físico entre o usuário e dispositivos de entrada [Matsumura e Sonnino 2011].

2.3.1 Nintendo® Wii Remote™

O Wii Remote™ é o controle primário do console Nintendo® Wii™ que foi lançado em 2006. O controle possui as funcionalidades tradicionais e também é um sensor de movimento, que permite ao usuário interagir e manipular itens na tela através de reconhecimento de gestos e apontamentos, a tecnologia empregada se apoia no uso de acelerômetro e sensor óptico [Lee 2008].

O dispositivo foi bem aceito pelos *gamers* e figurou-se como um dos responsáveis pela venda de mais de 20 milhões de unidades do console por todo o mundo, após um ano de seu lançamento. O mesmo relembra um controle de televisão, mas além dos botões, ele possui um acelerômetro com 3 eixos, alta resolução, uma câmera *InfraRed* (IR), um alto-falante, um motor para vibração e conectividade sem fio via Bluetooth® [Lee 2008].

A Figura 2.2 ilustra duas versões do controle. O anexo ao controle (ver Figura 2.2(b)) foi lançado em 2009 para melhorar a precisão e o tempo de resposta do sistema original [Sung 2011].

Embora as especificações do Wii Remote™ não tenham sido publicadas oficialmente, a comunidade hacker fez a engenharia reversa do dispositivo, descobrindo sig-

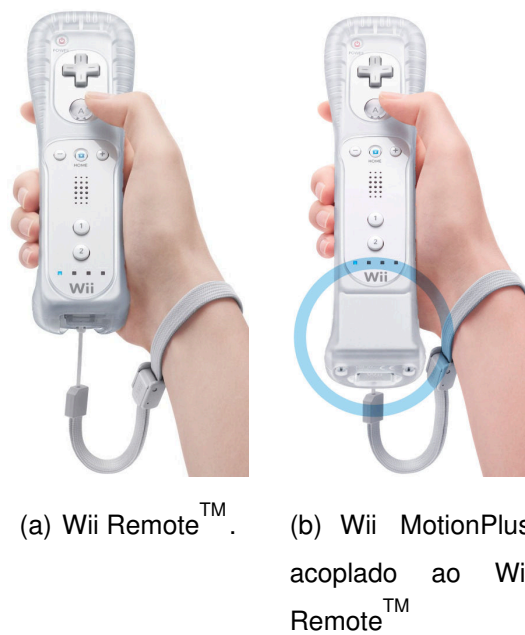


Figura 2.2: Versões do controle para o Nintendo® Wii™ [Nintendo 2013].

nificante porção das informações técnicas sobre o funcionamento interno do mesmo [Lee 2008].

A Nintendo® oferece um kit de desenvolvimento relativamente barato para o console Wii™, este acordo legal limita severamente os tipos de aplicativos autorizados a se desenvolver com o uso desta ferramenta. Alternativamente, pode-se conectar o controle a um computador pessoal via Bluetooth® e iniciar o desenvolvimento de aplicações customizadas, para isso existem as bibliotecas desenvolvidas pela comunidade, que permitem acesso e manipulação dos dados coletados pelo controle, estas bibliotecas estão disponíveis para grande parte das maiores plataformas de desenvolvimento, incluindo o Windows®, Mac OS X e Linux [Lee 2008].

O controle permite várias interações, algumas comumente utilizadas nos jogos, são elas: seleção, navegação, desenho, rotação de objetos e a ação de empurrar e puxar. O mesmo ainda é frequentemente utilizado como ponteiro.

A riqueza em termos de entrada e saída de dados combinados com a fácil conexão a computadores, fez do Wii Remote™ um dispositivo popular para explorar esquemas alternativos de controle para aplicações existentes. Lee [Lee 2008] cita várias áreas de pesquisas em torno do Wii Remote™, como: rastreamento dos dedos e objetos, lousas interativas, rastreamento da visão do usuário para aplicações de Realidade Virtual, rastreamento de movimento 3D, entre outras.

2.3.2 PlayStation® Move

Após o sucesso do Nintendo® Wii™, a Sony® lançou em 2010 o PlayStation® (PS) Move. O controle se assemelha com o controle do concorrente, exceto pela esfera iluminada no topo do dispositivo. O sistema é composto de duas partes: uma câmera denominada PS Eye e um controle sem fio [Bai et al. 2012]. A Figura 2.3 ilustra os dois componentes.



Figura 2.3: Componentes que integram o PS Move [Sony 2013].

O controle (ver Figura 2.3(a)) possui sensores inerciais para detectar movimento. Sendo um acelerômetro, um giroscópio de duplo eixo (x, y), outro de eixo único (z) e um magnetômetro. Ao mesmo tempo, a câmera captura a esfera brilhante (que pode mudar de cor), que se encontra fixa no topo do controle, a fim de calcular a posição exata em tempo real. Esta configuração permite transpor limitações apresentadas quando utiliza-se apenas os sensores inerciais [Williamson et al. 2011, Hirani e Brown 2010].

O controle possui interface de comunicação Bluetooth®, sendo possível a comunicação com um computador pessoal. O sistema (controle e câmera) pode ser acessado através do software Move.Me fornecido pela Sony® [Sony Computer Entertainment Inc 2011]. A Figura 2.4 ilustra o esquema de funcionamento do Move.Me.

Alternativamente, é possível utilizar bibliotecas *open source* para desenvolver aplicações que fazem uso do PS Move. A biblioteca PS Move Lib [Perl 2010] dá suporte

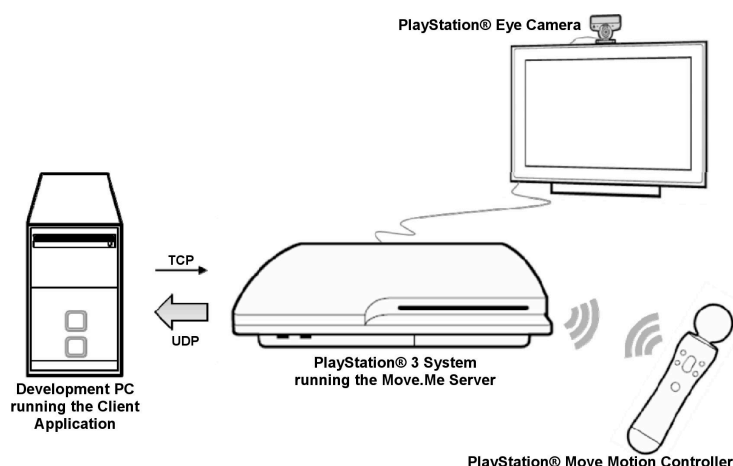


Figura 2.4: Diagrama do ambiente cliente/servidor do Move.Me [Sony Computer Entertainment Inc 2011].

para uso do controle nas plataformas Linux, Mac OS X e Windows[®], permitindo o acesso direto por um computador via Bluetooth[®] ou USB, sem a necessidade do console PS3 como intermediador e ainda habilita o rastreamento 3D usando a PS Eye ou outra câmera apropriada. A PS Move Lib é fruto de pesquisas realizadas na Universidade Tecnológica de Viena, em especial por Perl [Perl 2012].

2.3.3 Microsoft[®] Kinect[™]

Foi lançado em novembro de 2010 como um acessório para o console Xbox[®] 360. É um dispositivo desenvolvido pela empresa PrimeSense[™] em colaboração com a Microsoft[®]. Até janeiro de 2012, mais de 18 milhões de unidades foram vendidas. O anúncio do Kinect[™] em 2009 causou grande expectativa na comunidade acadêmica. O produto promete uma nova forma de interagir em jogos, completamente baseada em gestos e voz [Cruz, Lucio e Velho 2012].

Desde seu lançamento, o Kinect[™] vem sendo largamente utilizado na indústria (jogos, robótica e etc) e no setor de pesquisa. O seu baixo custo, confiabilidade e velocidade na medição contribuem para a sua popularização. Diversos *frameworks* para interpretação dos dados brutos obtidos com o sensor foram lançados antes mesmo do *Software Development Kit* (SDK) oficial da Microsoft[®], fazem parte deste conjunto o "libfreenect" desenvolvido pela comunidade OpenKinect e o "OpenNI[®]" desenvolvido primariamente pela PrimeSense[™] [Webb e Ashley 2012].

O referido dispositivo (Figura 2.5) possui uma câmera RGB, um emissor infravermelho e um sensor infravermelho, que são usados para triangularizar pontos no espaço, além de quatro microfones e uma base motorizada para regulagem do ângulo de visão. Ele é capaz de capturar imagens coloridas e a profundidade de cada pixel da cena. Esses dados contém informações visuais e geométricas da cena, que possibilitam realizar tarefas que antes eram difíceis, se não impossíveis, utilizando apenas imagens coloridas [Smisek, Jancosek e Pajdla 2011, Webb e Ashley 2012, Cruz, Lucio e Velho 2012].



Figura 2.5: Sensor KinectTM e seus componentes (adaptado de [Cruz, Lucio e Velho 2012]).

Vale ressaltar que o sensor possui, até a presente data, duas versões. Sendo a primeira lançada exclusivamente como acessório (controle) para o console Xbox[®] e a segunda voltada para o uso com computadores em geral. A priori, nenhum tipo de calibração é necessária para utilizá-lo, porém alguns autores tratam deste aspecto, como Smisek [Smisek, Jancosek e Pajdla 2011].

2.3.3.1 Sobre o hardware

O campo de visão do dispositivo é de 57.5 graus na horizontal, 43.5 graus na vertical e seu alcance operacional situa-se entre 0.8 e 4 metros no modo padrão, já no "modo próximo" opera entre 0.4 e 3 metros, sendo este suportado apenas pela versão para computadores [Microsoft 2012].

Sensor de profundidade É composto pelo emissor e sensor IR. O emissor projeta um padrão de luz IR no ambiente e o sensor (ou câmera) é usado para observar

e decodificar o padrão projetado, a fim de triangularizar a cena em 3D. A câmera IR opera a 30 Hz e captura imagens com 1200x960 pixels de resolução, estas imagens são redimensionadas para 640x480 pixels. O redimensionamento se faz necessário devido as limitações na vazão de dados suportada pela porta *Universal Serial Bus* (USB) 2.0, tendo em vista que todos os dados coletados pelo KinectTM (mapa de profundidade, RGB, som e acelerômetro) ocupam aproximadamente 70% da capacidade de transmissão de dados [Smisek, Jancosek e Pajdla 2011, Cruz, Lucio e Velho 2012].

Câmera RGB A câmera RGB opera a 30 Hz e pode capturar imagens à 640x480 pixels, sendo a resolução máxima de 1280x1024 pixels a uma taxa de 10 *Frames Per Second* (FPS), produzindo imagens de qualidade média [Smisek, Jancosek e Pajdla 2011, Cruz, Lucio e Velho 2012].

Motor, acelerômetro e microfones O motor tem o papel de inclinar a "cabeça" do dispositivo, em um intervalo de 27 graus (positivo ou negativo). E o acelerômetro é usado pelo próprio KinectTM para determinar em qual posição a "cabeça" está. Quanto aos microfones, os mesmos operam com uma taxa de amostragem de 16 kHz [Cruz, Lucio e Velho 2012].

2.3.3.2 Sobre o software

Os dados de profundidade fornecidos pelo sensor já resolvem a maior parte dos problemas relacionados a rastreamento de movimento, estes dados permitem filtrar qualquer pixel que não seja o usuário. O desafio para Shotton [Shotton et al. 2011] foi transformar esse esboço de uma pessoa em algo que pudesse ser rastreado, a dificuldade era dividir o esboço em partes do corpo reconhecíveis. A partir destas partes do corpo, as articulações podem ser identificadas, e por meio das articulações, o esqueleto pode ser reconstruído. Em um trabalho conjunto, chegou-se a um algoritmo que conseguiria distinguir 31 partes do corpo humano. A Figura 2.6 ilustra parte do resultado alcançado por Shotton e seu grupo.

Para resolver as limitações que impunham poses para reconhecimento e calibração, Shotton decidiu utilizar o poder do aprendizado computacional. Com o uso de muitos dados de poses e pessoas em movimento, um classificador foi desenvolvido,

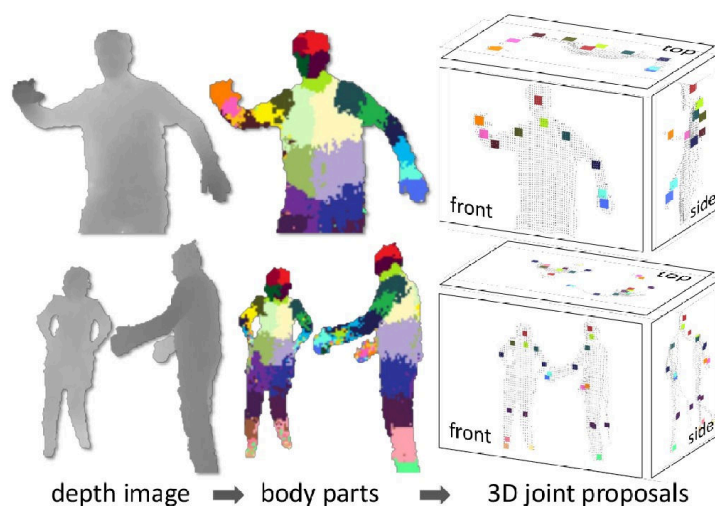


Figura 2.6: Divisão e reconhecimento das partes do corpo humano [Shotton et al. 2011].

dando ao KinectTM a habilidade de identificar uma pessoa sem a mesma determinada pose inicial, sem cortes ou perda da pessoa rastreada e reconhecer diferentes tipos de pessoas.

O sensor é capaz de detectar seis pessoas e rastrear duas (provendo o esqueleto). E ainda rastrear pessoas que estejam sentadas, fornecendo assim os dados das juntas superiores [Microsoft 2012]. A Figura 2.7 exemplifica o processo, sendo a esquerda o rastreamento de duas pessoas e detecção de seis, e a direita os dois modos de rastreamento.

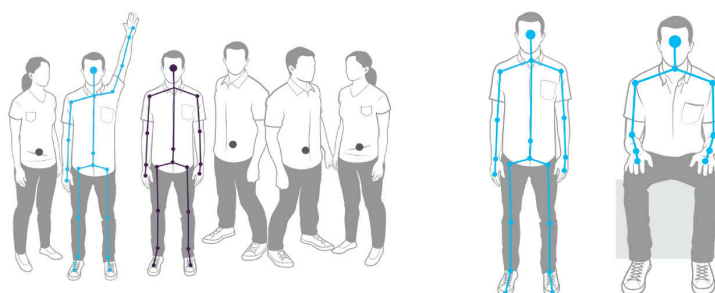


Figura 2.7: Capacidade e modos de rastreamento [Microsoft 2012].

2.3.4 Outros dispositivos

Existem outros dispositivos com o mesmo propósito do Kinect™, tal como o Asus® XTion Pro e o PrimeSense™ Sensor 3D. Contudo, o Kinect™ se destaca pelo preço mais acessível e melhor difusão no mercado (principalmente brasileiro). Antes do desenvolvimento destes sensores, outros dispositivos eram usados para gerar nuvem de pontos 3D. Estes dispositivos podem ser classificados como: câmeras *Time-Of-Flight* (TOF), luz estruturada, *scanner* 2D e 3D a *Light Amplification by Stimulated Emission of Radiation* (LASER) entre outros [Cruz, Lucio e Velho 2012, Duan, Deng e Wang 2012].

2.4 Ferramentas para manipulação de sensores de profundidade de baixo custo

Até o presente momento, há três maiores projetos que disponibilizam bibliotecas que podem ser usadas para coletar e processar os dados provindos de sensores, como o Kinect™, que com seu lançamento deu início a uma corrida para o desenvolvimento de um driver para seu uso, visto que o dispositivo não possuía suporte oficial para computadores pessoais. São eles, OpenNI®, OpenKinect e Microsoft® Kinect™ for Windows®.

As próximas seções apresentam uma breve descrição de cada uma destas bibliotecas, dando maior ênfase ao OpenNI®.

2.4.1 OpenKinect

O OpenKinect é um projeto de código aberto, que utiliza apenas bibliotecas também de código aberto para habilitar o uso do sensor Kinect™ nos sistemas operacionais Linux, Mac OS X e Windows®. O principal foco da comunidade é o desenvolvimento do software "libfreenect", este é o núcleo da biblioteca que permite acessar o Microsoft® Kinect™ USB câmera. A mesma suporta o acesso às imagens RGB, mapa de profundidade, motor, acelerômetro e *Light-Emitting Diode* (LED). A biblioteca é escrita em C, mas possui versões para Python, ActionScript, C++, C#, Java e outras linguagens de programação [Cruz, Lucio e Velho 2012, OpenKinect 2013].

2.4.2 OpenNI[®]

Open Natural Interaction (OpenNI) desenvolvido por um conjunto de empresas sem fins lucrativos que visam estabelecer um *framework* nos padrões industriais para a interoperabilidade de dispositivos para interação natural. Juntamente com o lançamento do *framework* vieram bibliotecas de *middleware*, que podem converter os dados brutos providos do sensor em dados prontos para serem lidos e utilizados em uma aplicação [Villaroman, Rowe e Swan 2011].

Estas bibliotecas estão disponíveis para aplicações escritas em C, C++ e C#. Sendo o Kinect[™] um dispositivo suportado pelo *framework* OpenNI[®], que dá condições para o desenvolvimento de aplicações envolvendo o sensor [Villaroman, Rowe e Swan 2011].

O OpenNI[®] é um *framework* multi-linguagem e multiplataforma que define *Application Programming Interfaces* (API) para a escrita de aplicações que envolvem Interação Natural. O propósito principal é estabelecer uma API padrão para comunicação com sensores de áudio e vídeo, assim como com o *middleware* que fornece a percepção a partir destes dados.

A Figura 2.8 mostra uma visualização em três camadas do conceito utilizado pelo OpenNI, cada camada tem o seguinte significado:

- **Aplicação:** representa o software que implementa aplicações que fazem o uso de gestos e voz para a interação.
- **OpenNI[®]:** fornece interfaces de comunicação com os sensores e os componentes de *middleware*, que analisam os dados dos sensores.
- **Dispositivos:** representam o hardware que captura dados de áudio e vídeo da cena.

2.4.2.1 NITE[™]

Natural Interaction Technology for End-user (NITE) é um *middleware* desenvolvido pela empresa PrimeSense Ltd e tem como função prover uma interface de alto nível para comunicação com dispositivos, como o Kinect, tendo a vantagem de poder ser utilizado com outros dispositivos que possuam recursos similares [PrimeSense 2011].

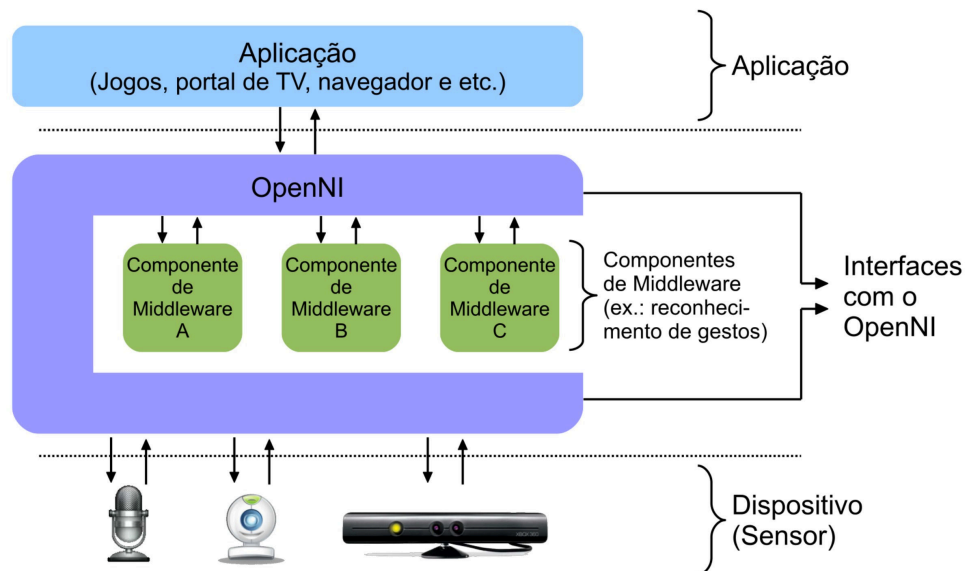


Figura 2.8: Abstração das camadas do *framework* OpenNI[®] (adaptado de [OpenNI 2011]).

O *middleware* inclui métodos para processamento de dados de vídeo e recurso de mapeamento da profundidade da cena, contendo uma infraestrutura de algoritmos capaz de realizar identificação de pessoas, características de objetos e reconhecimento de gestos, permitindo ao desenvolvedor criar aplicações controladas por gestos, encapsulando diversos algoritmos de processamento de imagens que são utilizados largamente em aplicações de visão computacional.

A Figura 2.9 ilustra uma das funcionalidades do NITE[™], a identificação de um espectador específico e rastreamento da palma da mão do mesmo, permitindo o controle da aplicação de navegação.

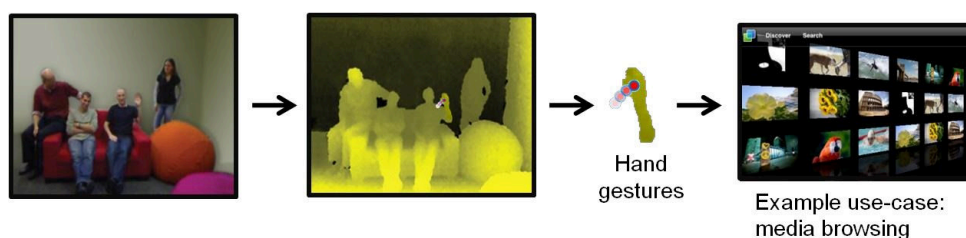


Figura 2.9: Controle baseado em gestos com as mãos [PrimeSense 2011].

2.4.3 Microsoft[®] Kinect[™] for Windows[®] SDK

É um conjunto de bibliotecas que permitem a criação de aplicações para plataformas Microsoft[®] utilizando o sensor Kinect[™] como dispositivo de entrada. Com elas pode-se programar aplicações baseadas em *Windows Presentation Foundation* (WPF), WinForms e XNA. O SDK suporta o uso do sensor Xbox[®] Kinect[™], todavia para ter acesso às funcionalidades como o "modo próximo" é necessário o hardware do Kinect[™] para Windows[®] [Webb e Ashley 2012].

As aplicações podem ser construídas em C++, C# e Visual Basic. O SDK oferece uma série de funcionalidades, como o reconhecimento do esqueleto no modo sentado, rastreamento do esqueleto, rastreamento facial e reconhecimento de voz [Cruz, Lucio e Velho 2012].

2.4.4 Resumo comparativo

No Quadro 2.1¹ é apresentada uma comparação entre as ferramentas descritas anteriormente, que podem ser utilizadas para acessar e processar os dados providos do Kinect[™].

Quadro 2.1: Resumo comparativo das ferramentas apresentadas.

Ferramentas	Rastreamento do corpo	Rastreamento das mãos	Suporte embutido para gestos	Gravação de áudio	Reconhecimento de voz	Rastreamento facial	Multiplataforma	Suporte a outros sensores	Controle do motor de inclinação	Acesso aos dados do acelerômetro
OpenNI [®]	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗
libfreenect	✗	✗	✗	✓	✗	✗	✓	✗	✓	✓
MS Kinect [™] SDK	✓	✗	✗	✓	✓	✓	✗	✗	✓	✓

¹ Levantamento realizado em Janeiro de 2013

Após a análise, expressa no Quadro 2.1, algumas considerações são relevantes para este trabalho:

- A biblioteca *libfreenect* é a que fornece menos recursos, em destaque o rastreamento das mãos, indispensável no desenvolvimento deste trabalho.
- O SDK da Microsoft[®] habilita o acesso a diversas funcionalidades e recursos, entretanto não possui suporte a múltiplas plataformas e não viabiliza a captura dos movimentos das mãos de forma independente do corpo e ainda carece de gestos pré-definidos.
- O *framework* OpenNI[®] suporta uma gama de funcionalidades similar ao SDK da Microsoft[®], todavia se destaca quando alinhado aos objetivos e especificações delineados para este trabalho, são as funcionalidades: rastreamento das mãos de forma independente e o suporte a múltiplas plataformas. E a desvantagem é o não suporte ao controle do motor de inclinação, situação que pode ser contornada pelo uso de códigos de terceiros para habilitar este controle.

Para este trabalho foram realizados testes preliminares com as ferramentas MS Kinect[™] SDK, *libfreenect* e por fim optou-se pelo uso do OpenNI[®] em conjunto com o NITE[™], pelo fato desse possuir suporte a outros sensores de profundidade, ser multiplataforma e o reconhecimento de alguns gestos já virem embutidos no *middleware*.

2.5 Considerações finais

Neste capítulo foram apresentados alguns dos fundamentos de Interação Humano-Computador associados ao desenvolvimento e construção de Interfaces Naturais com o Usuário. Dentre os tópicos abordados, estão técnicas e tipos de interação, estudo sobre a evolução das interfaces com o usuário, estudo dos dispositivos para suporte de Interfaces Naturais e ferramentas que viabilizam o uso dos mesmos.

Logo, o objetivo deste capítulo é prover ao leitor os conceitos necessário para o entendimento do protótipo apresentado nesta dissertação. No próximo capítulo será apresentado um estudo de alguns trabalhos correlatos ao tema deste trabalho.

Capítulo 3

Trabalhos Correlatos

3.1 Introdução

Visando analisar o estado da arte dos sistemas que combinam interfaces naturais e manipulação de objetos 3D realizou-se um levantamento de alguns trabalhos correlatos, procurando avaliar sua aplicabilidade, suas limitações e perspectivas de trabalhos futuros. Neste capítulo serão discutidos os aspectos de maior relevância abordados nestes estudos.

3.2 Manipulação de objetos 3D em aplicações de RA por meio da movimentação da mão

Santos [Santos 2011] apresenta um ambiente de Realidade Aumentada (RA)¹ no qual o usuário pode interagir através de movimentos com as mãos, sem nenhum aparato preso ao seu corpo.

O sistema permite efetuar as operações de rotação, translação e escala em objetos 3D pré-cadastrados, para tal, é utilizado o sensor KinectTM para capturar os dados de profundidade da cena. Como trata-se de um ambiente de RA as imagens capturadas pela câmera RGB são misturadas com o ambiente virtual. A Figura 3.1(b) exibe a interface desenvolvida por Santos.

¹Pode ser definida de acordo com [Kirner e Siscoutto 2007], como sendo o enriquecimento do ambiente real com objetos virtuais, a partir da utilização de ferramentas computacionais que permitam essa execução em tempo real.

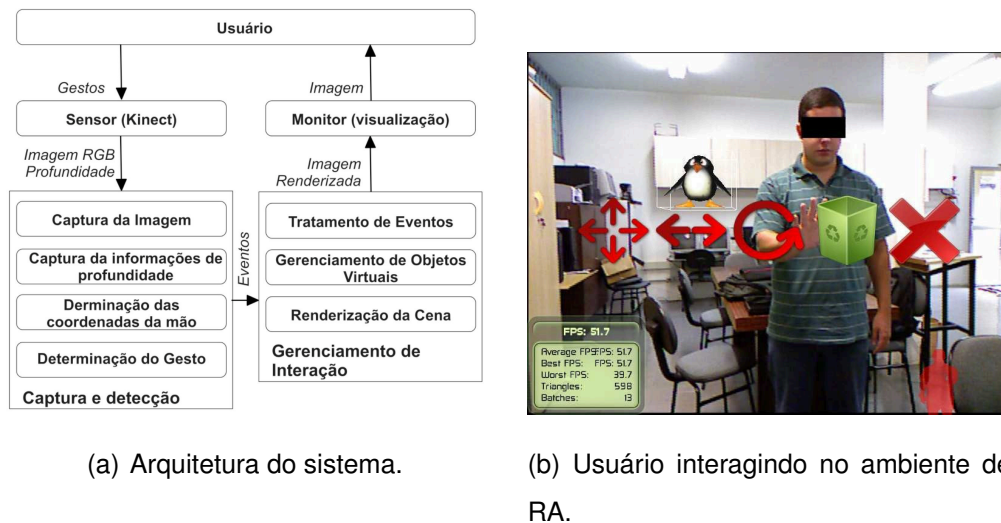


Figura 3.1: Sistema desenvolvido por Santos [Santos 2011].

Observa-se na Figura 3.1(b) um usuário interagindo no sistema por meio de movimentos com as mãos, a interface mistura elementos de uma GUI e algumas interações inerentes de NUI. A arquitetura (ver Figura 3.1(a)) desenvolvida contempla a captura de dados utilizando o sensor já discriminado e o *framework* OpenNI, constituindo um módulo da aplicação seguido pelo segundo que é responsável pelo gerenciamento do ambiente virtual e real servindo-se do motor de renderização *Object-Oriented Graphics Rendering Engine* (OGRE)².

O trabalho desenvolvido por Santos é um autêntico passo na área de RA, permitindo a interação com ambientes de RA sem a utilização dos tradicionais marcadores, como prova de conceito algumas interações foram desenhadas, habilitando o usuário a efetuar manipulações básicas em um ou mais objetos virtuais. O sistema foi desenvolvido com ferramentas multiplataforma e possui base para outros sensores suportados pelo OpenNI. Entretanto, a arquitetura não prevê o desenvolvimento independente entre os dois módulos (captura e renderização), restringindo o uso de outras formas de captura de dados. Ainda, interações mais complexas (como edição de malha) também não foram contempladas no escopo deste estudo.

²É um motor de renderização orientado a cenas e escrito em C++ [OGRE 2013].

3.3 Fusion4D - Interface natural e imersiva para manipulação de objetos 3D

Matsumura e Sonnino [Matsumura e Sonnino 2011] apresentam a implementação de uma interface gestual para manipulação direta de objetos virtuais 3D, de maneira a criar um ambiente para estudo das partes do corpo humano, como parte do projeto VIDA³.

Para atingir tais objetivos, o sistema utiliza o dispositivo KinectTM como método de entrada e uma série de algoritmos de reconhecimento que permitem definir comandos e manipulações *touchless* (interação a distância). Adicionalmente, a interface do mesmo é baseada em estereoscopia. A Figura 3.2 apresenta a arquitetura deste sistema.

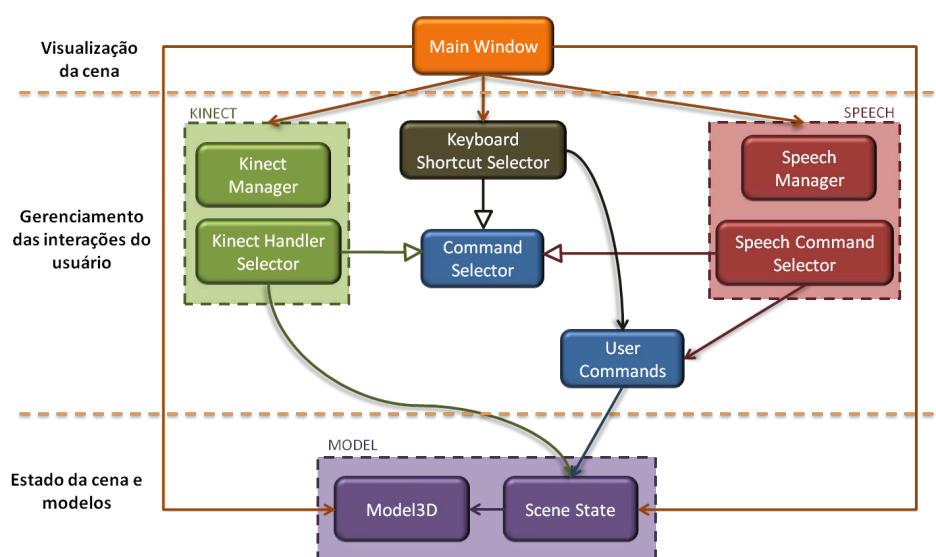


Figura 3.2: Arquitetura do sistema [Matsumura e Sonnino 2011].

As principais funcionalidades do sistema são (aplicadas a um objeto 3D):

- Translação nos eixos paralelos e perpendiculares à tela;
- Rotação em torno dos 3 eixos;
- Escala;

³ *Virtual and Interactive Distance-learning on Anatomy*, projeto desenvolvido pelo Interlab-USP que propõe o desenvolvimento de um atlas anatômico virtual 3D

- Seleção.

Para interagir utiliza-se manipulações diretas a distância, através de movimentos com as mãos para manipular os objetos e comandos de voz (em inglês) para alternar entre os modos de interação. A Figura 3.3 ilustra a ampliação de um objeto.

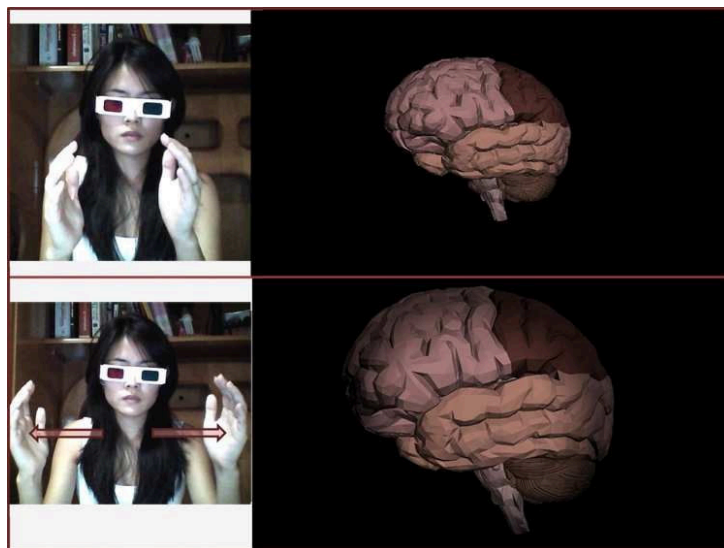


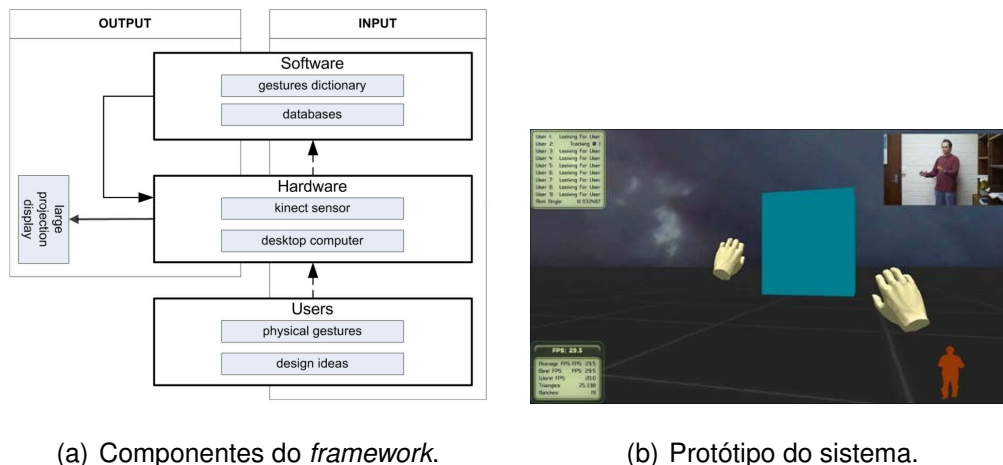
Figura 3.3: Ampliação de um objeto virtual [Matsumura e Sonnino 2011].

O sistema abordado neste trabalho apresenta diversas interações com objetos 3D pré-cadastrados e imersão do usuário. No entanto, não se inclui o suporte a outros sensores de movimento, similares ao KinectTM, nem a adição e edição (tratando-se de malha, visando a modelagem) de objetos 3D em tempo real. A utilização de uma arquitetura independente que contemple o suporte a qualquer dispositivo de entrada também não foi abordada nesta pesquisa.

3.4 Non-contact multi-hand gestures interaction techniques for architectural design in a Virtual Environment

Shiratuddin e Wong [Shiratuddin e Wong 2011] propõem um *framework* que possibilite o uso de técnicas de interação *touchless*, através de gestos com as mãos. A proposta tem como foco os sistemas que são utilizados para projetos arquitetônicos.

A fim de validar a proposta um protótipo foi desenvolvido, o mesmo consiste em um software que reconhece e interpreta gestos capturados pelo sensor KinectTM. A Figura 3.4(a) apresenta os principais componentes do *framework* proposto.

(a) Componentes do *framework*.

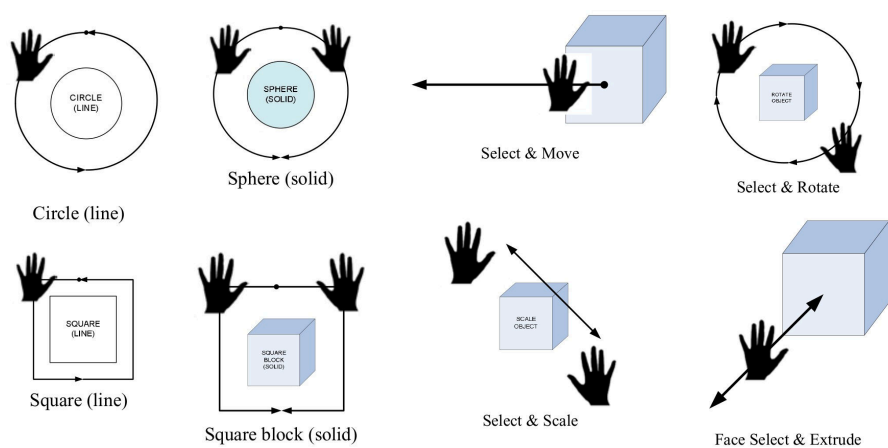
(b) Protótipo do sistema.

Figura 3.4: Sistema proposto por [Shiratuddin e Wong 2011].

Os gestos propostos foram categorizados em quatro tipos: navegação, criação de objetos, seleção de objetos e transformação de objetos. Um objeto pode ser selecionado pela sua face, seu vértice ou sua aresta, e as transformações possíveis são: translação, rotação e escala. Alguns dos gestos (ver Figura 3.5) propostos foram implementados no protótipo apresentado na Figura 3.4(b).

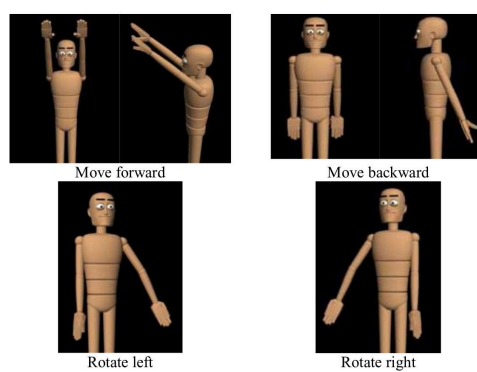
Os autores ressaltam que embora alguns gestos tenham sido implementados, mais testes devem ser feitos para confirmar se tais gestos são realmente adequados para a tarefa de modelagem arquitetural 3D. Ainda é apresentada uma comparação entre uma interface baseada em WIMP e uma *touchless*, com foco na utilização de programas do tipo *Computer-Aided Design* (CAD) 3D.

Este trabalho propõe um *framework* para desenvolvimento de técnicas para interação, baseadas em gestos com as mãos, com aplicações do tipo CAD. O mesmo apresenta um protótipo, no qual alguns gestos foram implementados. Alguns aspectos sobre os gestos não foram esclarecidos, por exemplo como eles são ativados e quais algoritmos foram utilizados. Quanto ao protótipo nenhum teste de usabilidade foi realizado, sendo uma das sugestões para trabalhos futuros por parte dos autores. Desta maneira, não foi realizada a implementação de todos os gestos propostos e seu projeto não aborda o suporte de outras formas de captura de dados.



(a) Criação de objetos.

(b) Transformação de objetos.



(c) Navegação.

Figura 3.5: Alguns dos gestos propostos por [Shiratuddin e Wong 2011].

3.5 Bimanual Natural User Interaction for 3D modelling application using stereo computer vision

Yang *et al.* [Yang et al. 2012] apresentam um sistema que permite a manipulação de objetos virtuais 3D através do uso de poses e movimentos com as mãos, o sistema se apoia nos conceitos de NUI utilizando técnicas de visão computacional. O controle bi-manual habilita o uso de poses com a mão esquerda para seleccionar os modos de controle e a mão direita para controlar os movimentos no AV 3D, utilizando as coordenadas 3D reais da mão.

Um protótipo foi desenvolvido e usa o rastreamento 3D das mãos e o reconhecimento de poses para controlar parte do software destinado a modelagem 3D. A posição da mão esquerda do usuário é extraída dos mapas de profundidade provindos do sensor KinectTM e as poses da mão direita são reconhecidas utilizando a técnica de *Principle Component Analysis* (PCA).

A Figura 3.6 apresenta as poses utilizadas para interação no sistema. As poses e as coordenadas 3D das mãos são enviadas para o software de modelagem 3D chamado Blender, via conexão *Transmission Control Protocol* (TCP). E por meio de *scripts* internos, o usuário é habilitado a acessar as funções de modelagem dentro do Blender.

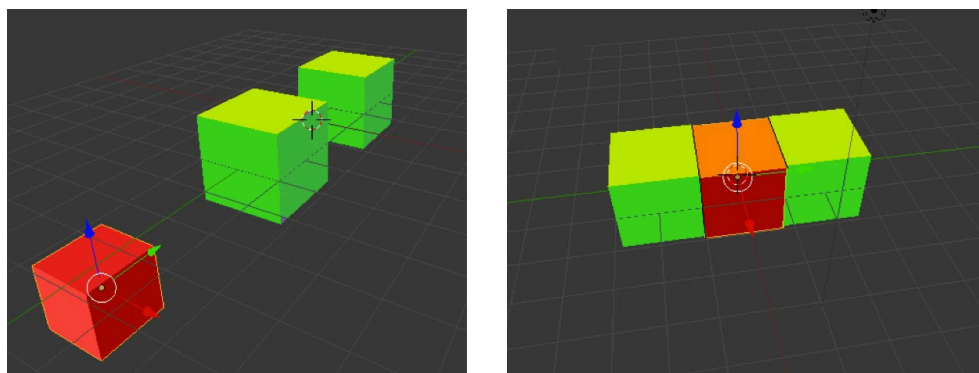


Figura 3.6: Seis poses utilizadas para interação [Yang et al. 2012].

A fim de avaliar a performance da concepção, do controle baseado em poses com a mão na tarefa de modelagem 3D, em um cenário real, um teste de usabilidade foi realizado e aplicado a dez pessoas. Os participantes foram orientados a realizar tarefas que envolviam a movimentação de objetos em um espaço virtual 3D. Estes realizaram as tarefas por um período de tempo, usando o mouse e o sistema de rastreamento 3D das mãos.

A Figura 3.7 exemplifica uma das tarefas executadas na avaliação, sendo o objetivo

posicionar o cubo vermelho (Figura 3.7(a) - Posição inicial) entre o par de cubos verdes (Figura 3.7(b) - Posição final).



(a) Cubo vermelho na posição inicial

(b) Cubo vermelho posicionado

Figura 3.7: Tarefa realizada [Yang et al. 2012].

Os resultados indicaram que os participantes que utilizaram o sistema, que rastreia a mão, completaram a tarefa mais rápido que os utilizadores do mouse. Entretanto, ressaltam os autores que aproximadamente metade dos participantes relataram que foi mais fácil realizar a tarefa por intermédio do mouse do que através do sistema de rastreamento das mãos. E ainda, todos reportaram que o referido sistema pode ser cansativo se utilizado por longos períodos.

Yang *et al.* aponta que a abordagem desenvolvida no trabalho parece promissora e merece futuras investigações, pois há vários desafios a serem pesquisados, como por exemplo:

- Melhorar o suporte para os braços, visto que os gestos na atual configuração são realizados no ar, o que tem gerado fadiga e cansaço. Uma possível abordagem para a redução das mesmas é diminuir a magnitude dos movimentos de forma a possibilitar o descanso dos braços.
- Quanto ao reconhecimento das poses, as mesmas obtiveram uma baixa taxa de reconhecimento. Portanto, foi sugerido como um dos tópicos para investigações futuras.

O sistema apresentado neste trabalho habilita o usuário a utilizar alguns recursos de um software para modelagem 3D por meio de movimentos e poses com as mãos. A comunicação entre o subsistema de rastreamento/reconhecimento das poses da

mão e a ferramenta de modelagem ocorre via conexão TCP, o que proporciona uma certa independência entre a captura de dados e a ferramenta manipulada. Contudo, a ausência de uma arquitetura bem definida impacta de forma negativa, dificultando a portabilidade e a reutilização do sistema, até mesmo o envolvimento de outras tecnologias que possam desempenhar papel similar as atualmente utilizadas. A avaliação conduzida visa apurar o desempenho da interface proposta/desenvolvida, as tarefas conduzidas no teste podem ser consideradas simples quando comparadas a realização de tarefas complexas, como a edição de uma malha 3D através de extrusões.

3.6 Resumo comparativo dos trabalhos analisados

Apesar de apresentarem escopos diferentes, os trabalhos relacionados apresentados neste capítulo contribuem para a elaboração desta dissertação de diferentes formas. O Quadro 3.1 sintetiza uma comparação destes trabalhos, destacando os elementos que motivaram o presente estudo. Observando o escopo e as técnicas abordadas em cada um destes trabalhos, foram escolhidos como parâmetros de comparação os elementos de maior impacto na portabilidade, flexibilidade e usabilidade dos sistemas, assim como a complexidade das interações e o conjunto de funcionalidades abrangidas. São eles:

- Método de captura dos movimentos corporais;
- Processamento dos movimentos;
- Flexibilidade da arquitetura;
- Portabilidade do sistema;
- Gestos/poses/movimentos utilizados para interação;
- Funcionalidades contempladas.

Quadro 3.1: Resumo comparativo dos trabalhos correlatos.

Trabalhos relacionados	Captura de dados independente	Ambiente para modelagem 3D	Gestos e poses para navegação na cena e manipulação de objetos	Edição de malha	Teste de usabilidade
Santos, 2011	✗	✗	P	✗	✓
Matsumura e Sonnino, 2011	✗	✗	P	✗	✓
Shiratudin, 2011	✗	✗	P	P	✗
Yang, 2012	✓	✓	P	✗	✓

Com o propósito de esclarecer os tópicos sintetizados no Quadro 3.1 segue uma breve explanação de cada um:

- **Captura de dados independente:** Neste tópico avaliou-se se os sistemas correlatos suportam vários tipos de sensores para a captura dos gestos e poses.
- **Ambiente para modelagem 3D:** Neste outro, se o dado sistema prove interações em um ambiente para modelagem 3D.
- **Gesto e poses para navegação na cena e manipulação de objetos:** Neste terceiro tópico é condensado se o sistema permite a completa interação por meio de gestos e poses com as mãos.
- **Edição de malha:** Se o referido trabalho suporta e/ou possibilita a edição de uma malha 3D, também por meio de gestos com as mãos.
- **Teste de usabilidade:** Se o sistema desenvolvido foi submetido a testes de usabilidade envolvendo potenciais usuários.

⁴ P = Em partes.

3.7 Considerações finais

Neste capítulo foram apresentados os principais elementos de alguns trabalhos correlatos ao tema desta dissertação. As principais limitações dos sistemas abordados estão relacionadas a portabilidade e flexibilidade da arquitetura, de forma a possibilitar o uso de outras formas de captura de dados e a utilização destes sistemas nas principais plataformas. E ainda, a investigação e implementação de interações mais complexas (por exemplo, edição de malha) que podem cooperar para o desenvolvimento das tarefas relacionadas a modelagem 3D.

A especificação e a implementação do sistema proposto neste trabalho serão apresentadas nos próximos capítulos.

Capítulo 4

Especificação do sistema proposto

4.1 Introdução

Neste capítulo serão discutidos os principais requisitos do protótipo¹ desenvolvido neste trabalho. Primeiramente serão apresentados os requisitos funcionais (funcionalidades) e não funcionais (requisitos de desempenho e resposta) do sistema proposto. Em seguida são apresentados os diagramas de casos de uso, o diagrama de classes, e por fim a arquitetura do sistema.

4.2 Requisitos do sistema

A partir dos objetivos deste trabalho, que estão definidos na seção 1.2, foram distinguidos os requisitos do sistema. Os Quadros 4.1 e 4.2 distinguem, respectivamente, os requisitos funcionais (RF) e não funcionais (RNF).

4.3 Diagramas de casos de uso

Nesta seção serão apresentados os principais casos de uso do protótipo desenvolvido. Os diagramas (em linguagem *Unified Modeling Language* (UML)) foram concebidos de acordo com as recomendações apresentadas pela norma ISO/IEC 19505 [OMG 2013].

¹Apelidado como NI2Blender, que significa *Natural Interface to Blender*.

Quadro 4.1: Requisitos funcionais do protótipo NI2Blender.

RF01	Reconhecimento de diversos tipos de potenciais usuários da espécie humana.
RF02	Operar em ambientes não controlados (ex. ambiente com variação de luz; várias pessoas).
RF03	Capturar de forma contínua o movimento das mãos do usuário e mapear no sistema.
RF04	Reconhecer gestos sem o uso de dispositivos e/ou roupas especiais acopladas ao corpo do usuário.
RF05	Prover interação por gestos/poses.
RF06	Manipular o ambiente virtual para modelagem 3D.
RF07	Não necessitar de exaustivas etapas de calibração e/ou treinamento.

Quadro 4.2: Requisitos não funcionais do protótipo NI2Blender.

RNF01	Ser compatível com os SO Linux, Mac OS X e Windows [®] .
RNF02	Tempo de resposta rígido (executar a tarefa no tempo correto).
RNF03	Operar com taxa mínima de 15 FPS.
RNF04	Ser considerado como “confortável” (comprovar por meio de pesquisa com usuários).
RNF05	Possuir “boa” usabilidade (comprovar por meio de pesquisa com usuários).

A Figura 4.1 apresenta os casos de uso do sistema. Visando um melhor entendimento dos principais casos de uso ilustrados na Figura 4.1 os Quadros 4.3, 4.4, 4.5 e 4.6 demonstram a descrição dos mesmos.

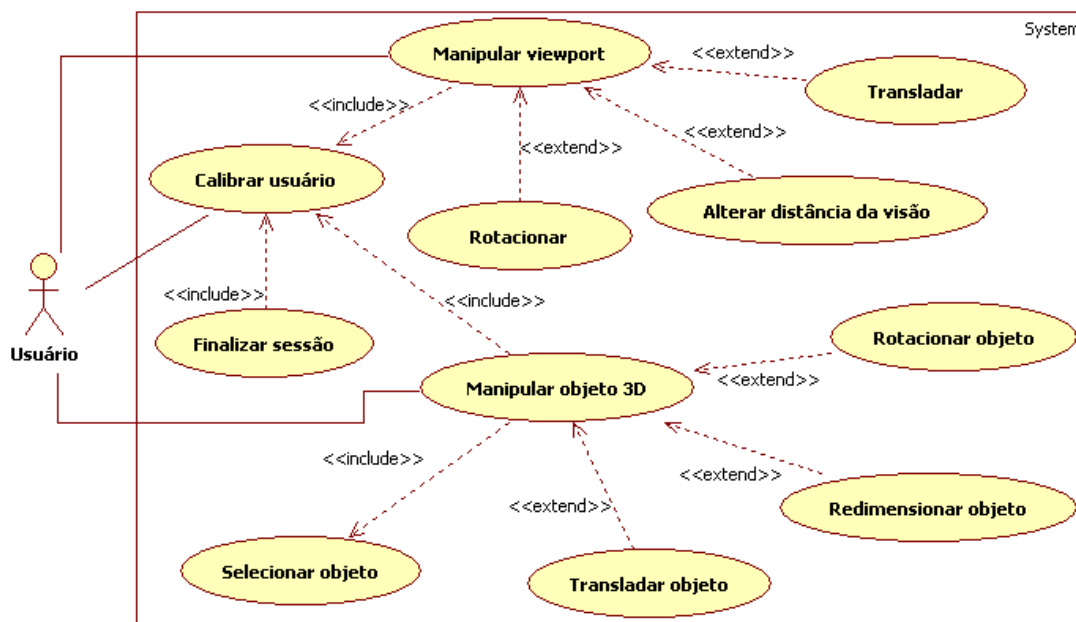


Figura 4.1: Diagrama de casos de uso.

Quadro 4.3: Descrição do caso de uso “Calibrar usuário”

Caso de Uso: Calibrar usuário
Resumo: Este caso de uso descreve o processo para inicializar uma sessão de interação.
Ator principal: Usuário
Precondição: O sistema estar com nenhum usuário ativo.
Fluxo básico: <ol style="list-style-type: none">1. O usuário faz a pose pré-determinada;2. O sistema reconhece a pose;3. A calibração do usuário é efetuada;4. O usuário acena com o gesto pré-determinado;5. O sistema reconhece o gesto;6. A sessão é inicializada.
Fluxo alternativo: Pose não reconhecida <ul style="list-style-type: none">• Refazer os passos 1, 2 e 3 do fluxo básico. Gesto não reconhecido <ul style="list-style-type: none">• Refazer os passos 4, 5 e 6 do fluxo básico.
Pós-condição: Sessão iniciada.
Objetivo: Iniciar uma sessão de interação com o sistema.

Quadro 4.4: Descrição do caso de uso “Manipular objeto 3D”

Caso de Uso: Manipular objeto 3D
Resumo: Este caso de uso descreve o processo para manipular objetos 3D.
Ator principal: Usuário
Precondição: O sistema estar inicializado e calibrado com usuário ativo.
Fluxo básico: <ol style="list-style-type: none">1. O usuário realiza o gesto pré-determinado para seleção de objeto;2. O sistema reconhece o gesto;3. O objeto é selecionado;4. O usuário efetua o gesto referente a manipulação desejada;5. O sistema reconhece o gesto;6. Através do movimento da(s) mão(s), o usuário realiza a operação escolhida;7. Parar as mãos por x segundos para liberar o objeto e aplicar as modificações realizadas.
Fluxo alternativo: Posição da mão perdida <ul style="list-style-type: none">• Realizar os passos 4, 5 e 6 do caso de uso descrito no Quadro 4.3.
Pós-condição: Modificação realizada no objeto 3D.
Objetivo: Modificar a posição, ângulos de rotação ou escala do objeto virtual selecionado.

Quadro 4.5: Descrição do caso de uso “Manipular *viewport*”

Caso de Uso: Manipular <i>viewport</i>
Resumo: Este caso de uso descreve o processo para manipular a <i>viewport</i> (ponto de vista da cena).
Ator principal: Usuário
Precondição: O sistema estar inicializado e calibrado com usuário ativo.
Fluxo básico: <ol style="list-style-type: none">1. O usuário realiza o gesto/pose referente a manipulação desejada;2. O sistema reconhece o gesto/pose;3. Através do movimento da(s) mão(s), o usuário realiza a operação escolhida;4. Parar as mãos por x segundos para liberar a <i>viewport</i> e aplicar as modificações realizadas.
Fluxo alternativo: Posição da mão perdida <ul style="list-style-type: none">• Realizar os passos 4, 5 e 6 do caso de uso descrito no Quadro 4.3.
Pós-condição: Modificação realizada na <i>viewport</i> .
Objetivo: Modificar a posição, ângulos de rotação ou distância da visão (<i>viewport</i>).

Quadro 4.6: Descrição do caso de uso “Finalizar sessão”

Caso de Uso: Finalizar sessão
Resumo: Este caso de uso descreve o processo para finalizar um sessão de interação.
Ator principal: Usuário
Precondição: O sistema estar inicializado e calibrado com usuário ativo.
Fluxo básico: <ol style="list-style-type: none">1. O usuário posiciona as mãos e as mantém por x segundos fora do campo de visão do sensor;2. O sistema perde o rastreamento das mãos e finaliza parte da sessão;3. O usuário sai do campo de visão do sensor;4. O sistema reconhece a ausência do usuário e finaliza a sessão.
Fluxo alternativo: -
Pós-condição: Sessão finalizada e sistema aguardando novo usuário.
Objetivo: Finalizar sessão de interação com o sistema.

4.4 Diagrama de classes

Com base nos requisitos apresentados anteriormente, foi concebida a estrutura para o protótipo deste estudo. O diagrama de classes dessa estrutura é apresentado na Figura 4.2.

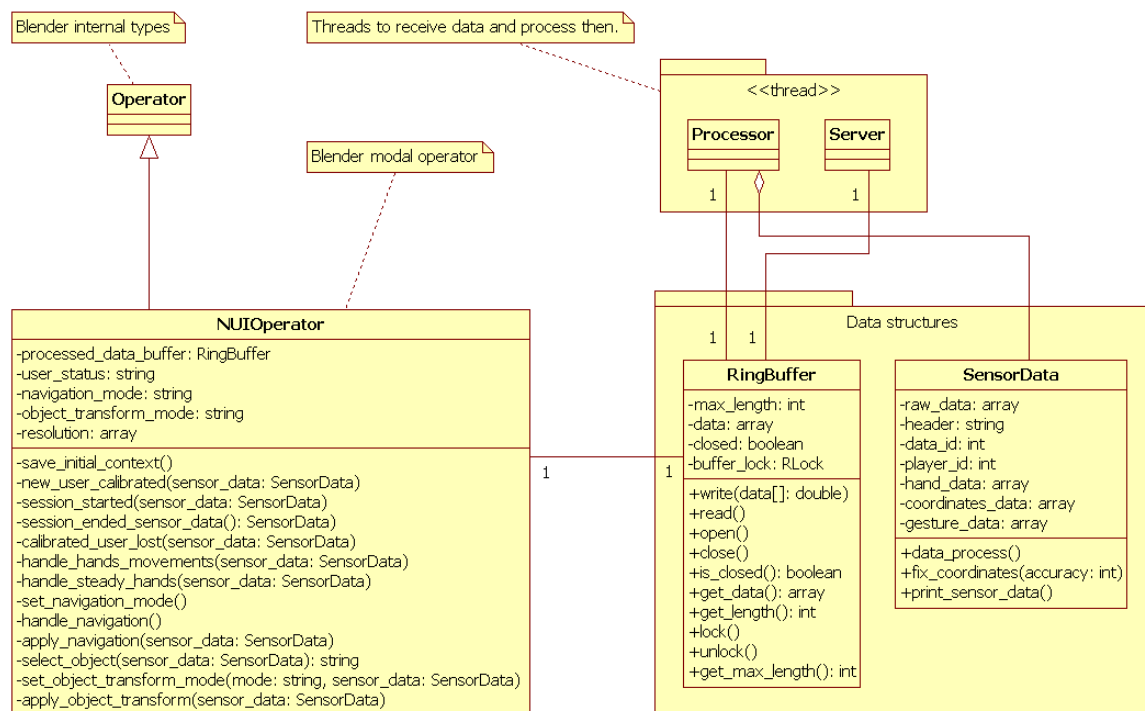


Figura 4.2: Diagrama de classes.

No diagrama apresentado na Figura 4.2, os dados alimentam o sistema através da *thread Server*, esta logística será melhor apresentada no Capítulo 5. A *thread Processor* tem a função de processar os pacotes recebidos, criando para cada pacote um objeto do tipo *SensorData*. Vale ressaltar que a classe *RingBuffer* tem por função prover uma zona para a troca de informações e por fim o diagrama apresenta o operador *NUI* que consome essas informações e atualiza o Ambiente Virtual.

4.5 Arquitetura do sistema

A arquitetura do NI2Blender (mostrada na Figura 4.3) é dividida em dois módulos e figura uma arquitetura baseada no modelo *pipeline*².

Observa-se que o fluxo de execução segue a captura da imagem e do mapa de posicionamento 3D do usuário e também de determinação do gesto pelo módulo produtor. Já no módulo consumidor, os dados são recebidos do produtor por meio de uma conexão *socket* e são armazenados temporariamente em *buffers* de dados que os recebem e processam, disponibilizando-os posteriormente para interpretação e reconhecimento. Por fim, a cena é atualizada de acordo com os dados capturados e processados.

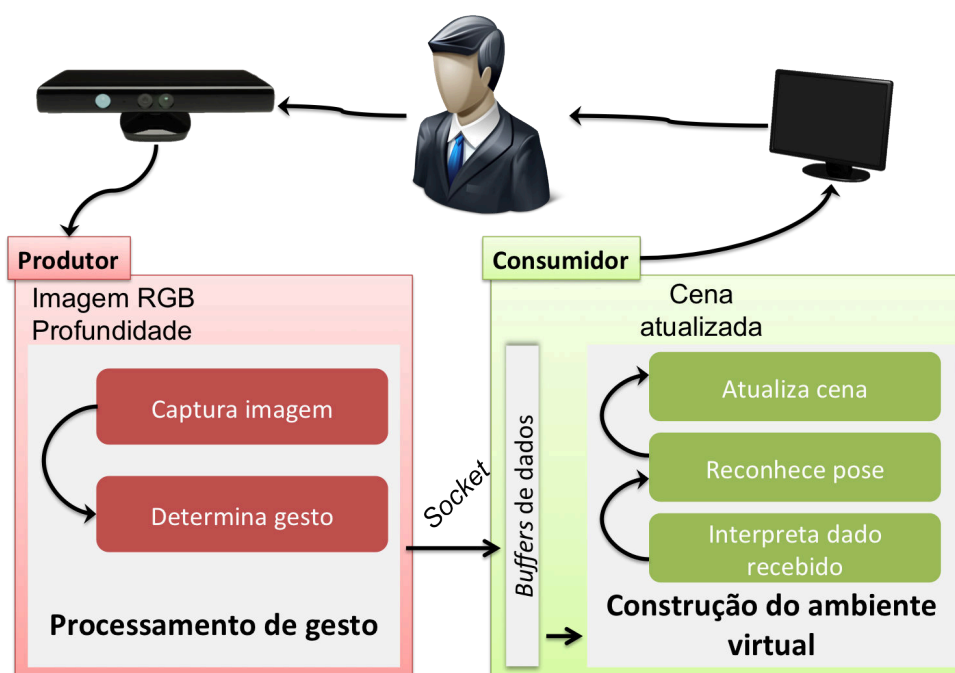


Figura 4.3: Arquitetura do sistema.

O detalhamento dos módulos é explanado a seguir:

Produtor

²No modelo *pipeline*, uma tarefa é dividida em sub-tarefas. Estas sub-tarefas são executadas seguindo uma ordem que produza o resultado desejado. A tarefa é concluída gradualmente e cada passo seguinte depende do passo anterior. Entretanto, o programa é desenhado para produzir múltiplos resultados desejados, e cada passo é projetado para operar em paralelo, mantendo todos os componentes sempre ocupados [IBM 2001].

- **Captura imagem:** recebe os dados (imagem RGB e mapa de profundidade) de um sensor. É responsável por exibir as imagens RGB e disponibilizar o mapa de profundidade para processamento e análise;
- **Determina gesto:** processa, analisa e determina qual gesto foi realizado de acordo com os dados capturados pelo bloco anterior.

Entre os dois módulos uma conexão *socket* provê a transmissão dos dados entre o produtor e o consumidor. Para que os dados possam trafegar é previsto a codificação destes em um formato intercambiável e a decodificação dos mesmos.

Consumidor

- **Interpreta dado recebido:** faz a leitura do *buffer* de dados, que simula uma estrutura de dados do tipo *First In, First Out* (FIFO), e interpreta o próximo dado da fila distinguindo-o entre as categorias apresentadas no Capítulo 5 e direciona o fluxo de execução do sistema;
- **Reconhece pose:** mediante o dado interpretado pelo bloco anterior este bloco entra em ação, seu papel é reconhecer as poses que o usuário faz caso sejam relevantes para a operação do sistema;
- **Atualiza cena:** bloco responsável por atualizar a cena que será exibida ao usuário com as modificações efetuadas.

4.6 Considerações finais

Neste capítulo foi apresentada uma breve especificação do sistema proposto, contendo seus principais requisitos e uma primeira estruturação do mesmo. No próximo capítulo serão explanados alguns detalhes da implementação, bem como um refinamento da especificação inicial.

Capítulo 5

Detalhes da implementação

5.1 Introdução

Neste capítulo serão apresentados alguns dos detalhes da implementação do protótipo discutido neste trabalho, o qual desenvolvido de acordo com os requisitos discutidos no Capítulo 4.

5.1.1 Tecnologias de apoio

O desenvolvimento do protótipo apresentando por este trabalho demandou a utilização de um sensor de movimento e alguns módulos de *software*. As informações detalhadas sobre o *hardware* e os programas citados nesta seção se encontram no Capítulo 2.

Levando em consideração o explanado na seção 2.3.4 do Capítulo 2 o sensor adotado para desenvolvimento deste trabalho foi o KinectTM, devido à sua versatilidade, fácil acesso, bom suporte e facilidade de obtenção. Quanto aos programas, o *framework* OpenNI[®] juntamente com o *middleware* NITETM foram adotados pelo fato de serem multiplataforma e disporem de um maior conjunto de funcionalidades de interesse do projeto, de acordo com os requisitos apresentados no Capítulo 4.

Para o papel de modelador 3D o *software* escolhido foi o Blender¹, por ser uma fer-

¹Blender é um programa de computador de código aberto, desenvolvido pela *Blender Foundation*, para modelagem, animação, texturização, composição, renderização, edição de vídeo e criação de aplicações interativas em 3D. Está disponível sob uma licença dupla: *Blender License/GNU General Public License* (GPL). O programa é multiplataforma e inclui suporte ao Python como linguagem de

ramenta multiplataforma, de código aberto e permitir o acesso às suas funcionalidades via programação, utilizando a linguagem Python [Brito 2010].

5.2 Ambiente de desenvolvimento

Conforme o RNF01 (ver seção 4.2), foi determinado que a aplicação desenvolvida neste trabalho deve ser compatível com os SO Linux, Mac OS X e Windows®. Dessa forma, todos os módulos de *software* utilizados devem possuir versões compatíveis com esses SO.

Apesar da compatibilidade das ferramentas utilizadas com os SO em questão, a preparação de cada ambiente exige a instalação específica para o SO e para a arquitetura do computador, estes pré-requisitos são:

- NITE™ - versão 1.5.2.21;
- OpenNI® - versão 1.5.4;
- Blender - versão 2.60 ou superior.

Quanto ao *hardware*, os equipamentos utilizados para desenvolvimento e teste foram:

- MacBookPro: Processador Intel® Core™ i7 2.9 GHz, Memória RAM 8 GB, Placa de vídeo Intel® HD 4000 512 MB;
- Desktop: Processador Intel® Core™ 2 Quad Q8400 @ 2.66 GHz, Memória RAM 6 GB, Placa de vídeo GeForce® GTS 250.

5.3 Processo de desenvolvimento

Durante o desenvolvimento do protótipo apresentado, o Scrum² foi utilizado na modalidade “solo” de forma a auxiliar no processo de planejamento e execução do desenvolvimento do projeto.

script [Brito 2010, Blender Foundation 2013].

²É um processo de desenvolvimento iterativo e incremental para gerenciamento de projetos e desenvolvimento ágil de *software*. O Scrum solo é uma adaptação para o uso de programadores solo.

5.3.1 Captura de imagens

Para desenvolver o presente protótipo, foi necessário o uso do Kinect para capturar os movimentos do usuário, a fim de mapeá-los posteriormente. Tal etapa é de suma importância, visto que é a base para todos os outros processos.

O *framework* OpenNI dispõe de uma série de métodos para a inicialização do sensor em questão. Inicialmente, é necessário definir quais recursos serão utilizados pela aplicação, tais recursos podem ser observados logo a baixo:

- Gerador do mapa de profundidade e imagem RGB, configurado para gerar imagens com resolução de 640x480 a 30 FPS;
- Rastreador de usuário, mãos e gestos, com o parâmetro de suavização configurado empiricamente.

Enquanto as imagens são capturadas, elas são exibidas ao usuário. Para exibir o fluxo de imagens utilizou-se a biblioteca FreeGLUT³.

5.3.2 Determinação de gestos

Após a inicialização dos módulos do OpenNI, cada um dos recursos (ver subseção 5.3.1) fica disponível através de um objeto (instância de uma classe), tornado-se um canal de comunicação que possibilita o acesso e manipulação dos dados do sensor. O passo seguinte é configurar e inicializar o *middleware* NITE. Nesse deve-se trabalhar com a configuração e criação dos objetos responsáveis pela detecção de gestos. Os detectores empregados neste trabalho são:

- *XnVWaveDetector* - verifica se a mão efetuou quatro trocas de direção dentro de um espaço de tempo pré-determinado;
- *XnVPushDetector* - verifica se ocorreu um movimento de aproximação e afastamento da mão do sensor;

³É um conjunto de ferramentas escritas em cima da biblioteca OpenGL. GLUT torna consideravelmente mais fácil utilizar os recursos do OpenGL, permitindo ao usuário criar e gerenciar janelas. A biblioteca FreeGLUT é multiplataforma, de código aberto e possui implementação em diversas linguagens, incluindo C e C++ [Olszta, Umbach e Baker 2013].

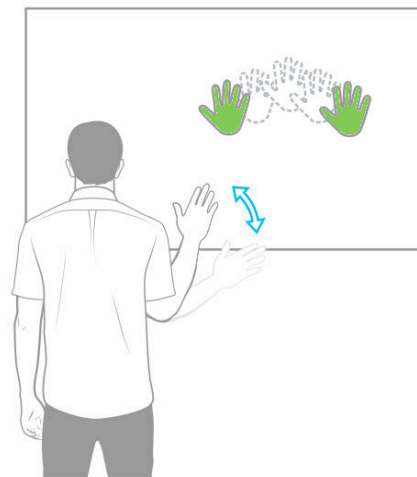
- *XnVSteadyDetector* - é ativado se a mão ficar parada;
- *XnVCircleDetector* - detecta se o usuário efetuou um movimento circular completo com a mão;

O reconhecimento dos gestos e a atualização da posição da mão são efetuados por meio do mecanismo de eventos da classe *XnVSessionManager*. Com o início da sessão é possível monitorar a ocorrência de ações do usuário através das classes que são os detectores (*XnVWaveDetector*, *XnVPushDetector*, *XnVSteadyDetector* e *XnVCircleDetector*), e para cada ação implementar *callbacks* para tratar estes eventos.

Para iniciar uma sessão são requeridos dois passos, o primeiro é uma pose a ser feita pelo usuário, passo que pode ser dispensável, porém optou-se por requerê-lo, a fim de suportar de maneira eficiente a operação do sistema em ambientes onde haja muitos usuários no campo de visão do sensor. E o segundo passo, que de fato inicializa a sessão, é o movimento *wave*. A Figura 5.1(a) ilustra a pose “psi”, que deve ser feita para calibração do usuário e a Figura 5.1(b) exemplifica o movimento *wave*, usado para iniciar a sessão.



(a) Pose “psi”.



(b) Movimento *wave*.

Figura 5.1: Passos para iniciar uma sessão.

Ambos os passos fornecem um *feedback* visual, que sinaliza se o passo foi efetuado corretamente ou não. O Apêndice A apresenta detalhes quanto aos *feedbacks* e a utilização geral do sistema.

5.3.3 Comunicação com o Blender

Visando implementar a comunicação entre o sistema de captura/determinação dos gestos e o Blender, adotou-se o modelo *pipeline*, conforme explanado na seção 4.5. Para a aplicação apresentada, um canal de comunicação é estabelecido de forma unidirecional, ou seja, do sistema de captura e determinação para o modelador 3D.

A comunicação se dá via *socket* através do protocolo TCP, os dados trafegam no formato *string* sendo formatados no módulo produtor e interpretados no consumidor (mais especificamente na *thread* processador - ver Figura 5.3). A Figura 5.2 demonstra o padrão dos dados que transitam de um módulo para outro.

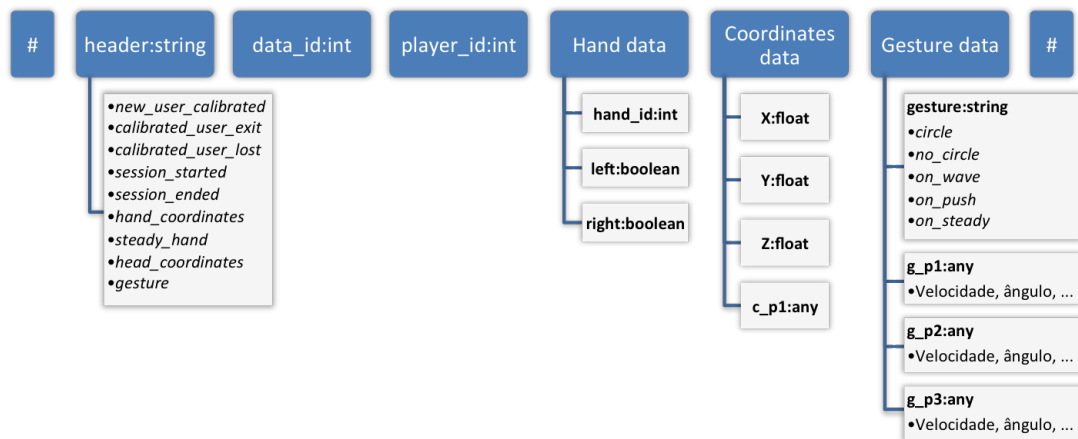


Figura 5.2: Formato dos dados que trafegam do produtor para o consumidor.

5.3.4 Processamento em tempo real

A aplicação apresentada demanda funcionamento em tempo real, conforme especificado na seção 4.2. Para tanto, utilizou-se estratégia similar a apresentada por Mattioli [Mattioli 2012]. Mattioli apresenta um simulador de aquisição/classificação em tempo real de sinais eletromiográficos (EMG). Para simular um sistema de aquisição real, os dados são enviados a uma frequência fixa, que pode ser configurada para corresponder à frequência de amostragem do sistema de aquisição. Esta estratégia se mostrou eficiente em seu estudo de caso e tendo em vista que a logística da aplicação desenvolvida por Mattioli e a apresentada neste trabalho são similares, vislumbrou-se o reuso deste esquema adequando-o a aplicação desenvolvida.

A Figura 5.3 ilustra o esquema desenvolvido para suportar a comunicação entre as duas aplicações.

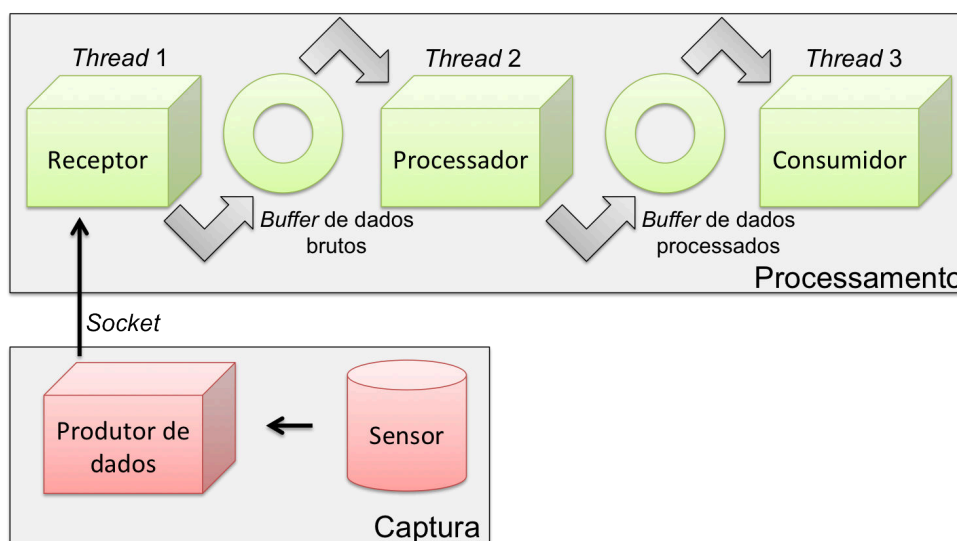


Figura 5.3: Esquema para suportar a comunicação e processamento em tempo real.

No módulo produtor (bloco captura), os dados são capturados a partir de um sensor de profundidade, formatados de acordo com o formato apresentado na Figura 5.2 e enviados diretamente ao consumidor (bloco processamento) via *socket*. A frequência de dados enviados segue a taxa de FPS setado na aplicação que captura os dados do sensor, neste caso 30 FPS.

No módulo consumidor (bloco processamento), 3 *threads* operam em paralelo fazendo a recepção dos dados, processamento e formatação e por fim consumo dos mesmos. O funcionamento de cada uma destas *threads* será detalhado a seguir.

5.3.4.1 Receptor

Esta *thread* serve como portão de entrada, ela é responsável por gerenciar a conexão *socket*, receber os dados e alimentar o *buffer* circular, denominado *buffer* de dados brutos que serão mais tarde processados.

5.3.4.2 Processador

A *thread* de processamento tem por função capturar os dados armazenados no *buffer* de dados brutos (pela *thread* de recepção) e fazer a interpretação destes da-

dos. Para tal, os dados são contados, pois cada pacote recebido via conexão TCP pode conter mais de um dado, e checados no quesito integridade (conforme o formato apresentado na Figura 5.2), na sequência esses são interpretados de forma a serem encapsulados em um objeto do tipo *SensorData*, conforme o diagrama de classes apresentado na seção 4.4, sendo um objeto para cada dado recebido. Por fim, cada objeto é armazenado no *buffer* de dados processados.

5.3.4.3 Consumidor

A *thread* que, de fato, irá consumir os dados e produzir *feedbacks* visuais, espera que informações sobre posicionamento de mão, gestos realizados e etc estejam disponíveis no *buffer* de dados processados. Quando uma informação está disponível, ela é consumida por esta *thread* que possui características diferentes e também comportamento diferenciado das demais explanadas anteriormente. Tais peculiaridades são detalhadas na seção 5.4.

5.4 Blender operator

O Blender é um *software* que possui diversas funcionalidades (apresentadas na seção 5.1.1), e todas essas podem ser acessadas através da linguagem de programação Phyton que é embutida no Blender. O *software* possui uma API vasta e para desenvolver a *thread* consumidor (que pode ser entendida como um *plugin* ou *add-on*), brevemente descrita na seção 5.3.4.3, foi necessário um estudo sobre a arquitetura e funcionamento do Blender, parte deste estudo pode ser encontrado em [Blender Developers 2013, Blender Developers 2013, Blender Foundation 2013].

Para ter acesso as funcionalidades do Blender é imprescindível seguir o mínimo de normas para desenvolvimento de *addons*. A classe *Operator* (do Blender) fornece uma interface avançada para interação com o *software* em questão, ela deve ser herdada por outra classe e então as personalizações podem ser implementadas na classe filha. Esse procedimento pode ser visualizado na Figura 4.2 que representa o diagrama de classes do protótipo apresentado. Esses operadores são interfaces que permitem o desenvolvimento de *addons* para o Blender e conseqüentemente o acesso as funções que o mesmo possui.

O operador desenvolvido (*NUIOperator*) é do tipo *modal timer*, isso significa que ele se executa de tempo em tempo e trabalha em segundo plano. Nele estão definidos os métodos para manipular o ponto de vista da cena, selecionar os objetos virtuais 3D e manipulá-los.

Muitas das operações no Blender (rotação, translação e escala), tanto no que concerne ao ponto de vista da cena e a manipulação de objetos 3D, possuem um operador, cuja função é abstrair demasiados detalhes matemáticos acerca da operação em questão.

Antes de adentrar nos detalhes de cada método do *NUIOperator* é fundamental explanar sobre o mapeamento do sistema de coordenadas Kinect-Blender.

5.4.1 Mapeamento do sistema de coordenadas

Conforme explanado na seção 5.3.1, as imagens do Kinect foram configuradas para serem capturadas na resolução de 640x480, para mapear essa resolução dentro do Blender aplicou-se a Equação 5.1 a cada ponto x e y provindo do Kinect.

$$bp = kp - (kr/2)/(kr/br) \quad (5.1)$$

sendo bp o ponto no Blender, kp o ponto capturado pelo Kinect, kr a resolução do Kinect e br a “resolução” adotada no Blender.

Já para os pontos no eixo z usou-se a Equação 5.2.

$$bp = kp/(kr/br) \quad (5.2)$$

sendo bp o ponto no Blender, kp o ponto capturado pelo Kinect, kr a resolução do Kinect e br a “resolução” adotada no Blender.

Nota-se que as Equações 5.1 e 5.2 se diferem, o motivo é que para os eixos x e y do Kinect usou-se metade da resolução para mapear a parte positiva desses mesmos eixos no Blender e a outra metade a parte negativa, já no eixo z utilizou-se apenas a parte positiva. A Figura 5.4 ilustra o sistema de coordenadas vigente em cada módulo.

Conforme pode-se observar na Figura 5.4(b) o Blender possui um plano (ou palco) que serve para orientação e organização durante a utilização do *software*, este plano, por padrão, é formado por uma matriz de 16x16 quadrados (propriedade configurá-

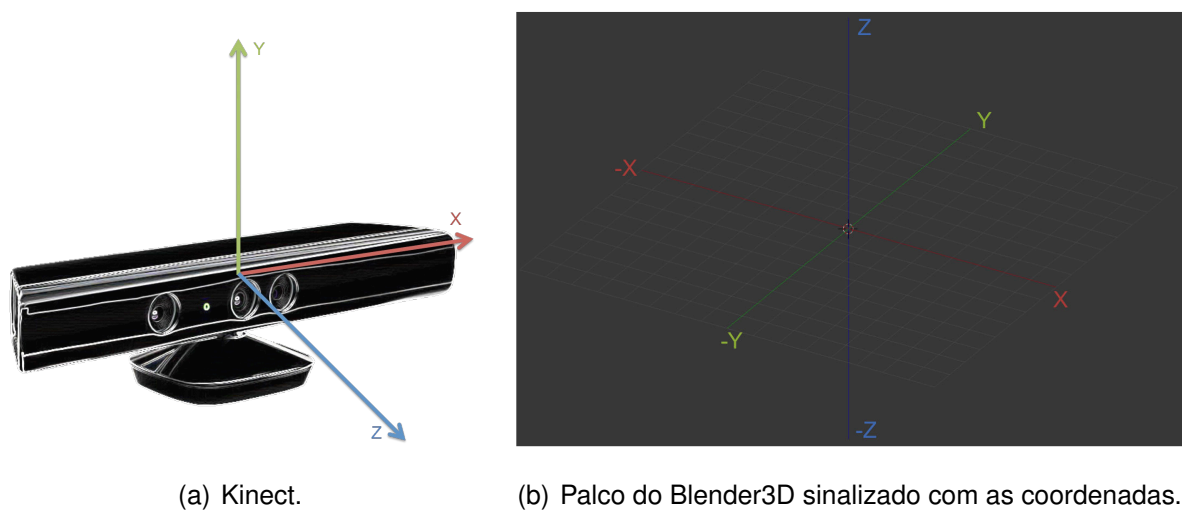


Figura 5.4: Sistemas de coordenadas.

vel). Portanto, para habilitar a movimentação das mãos virtuais (que são as mãos do usuário mapeadas no sistema) atribuiu-se ao elemento br da Equação 5.1 o valor 16 (dezesseis), permitindo assim a completa movimentação das mãos virtuais nos eixos x e y no plano do Blender. E o valor 8 (oito) para br no caso da Equação 5.2, possibilitando a movimentação na parte positiva do eixo z .

Resumindo, a resolução de 640x480 pixels x 4 metros do Kinect corresponde a 16x16x8 unidades no Blender.

5.4.2 Processo de interação

De acordo com Pinho [Pinho 2000], nos ambiente tridimensionais não-imersivos as imagens são exibidas na tela do computador e o processo de interação em um ambiente virtual, seja ele imersivo ou não, é considerado como sendo um sistema contínuo em que cada ação do usuário deve se respondida com uma ação do controlador do sistema de ambiente virtual.

Pinho [Pinho 2000] ainda define que o processo interativo pode-se dividir em dois: interação não-imersiva e interação imersiva. Para este trabalho adotou-se a interação não-imersiva, na qual o usuário observa um cenário tridimensional através da tela do computador.

Nas seções seguintes são apresentados os aspectos da metáfora de interação os parâmetros que a controlam e os detalhes de cada operação.

5.4.2.1 Metáfora de interação e seus parâmetros

A metáfora adotada opera de forma relativa, ou seja, varia de acordo com o estado atual do cenário.

Para as operações de translação e *zoom* do ponto de vista da cena, translação, rotação e redimensionamento de objetos trabalha-se com modos de operação, seguindo o fluxo abaixo:

1. Fazer pose ou gesto com as mãos para ativar o modo;
2. Movimentar a mão (x , y ou z) para visualizar as mudanças específicas de cada modo;
3. Parar as mãos por ~ 1.5 segundos para aplicar as alterações.

As operações de translação e *zoom* do ponto de vista da cena, translação, rotação e redimensionamento de objetos não podem acontecer em paralelo. Quanto as operações de rotação do ponto de vista da cena e seleção de objeto, para estas não é necessário entrar em um modo de operação, elas podem ser realizadas de forma direta.

O Quadro A.1 do Apêndice A apresenta os *feedbacks* visuais gerados pelo sistema e seus significados.

Para cada situação e operação adotou-se uma resolução, baseando essas em testes e avaliações.

5.4.2.2 Navegação

Entende-se por navegação o processo de mudar a posição do observador do ambiente tridimensional. O escopo de navegação na cena (para este trabalho) pode ser dividido em: rotação, translação e *zoom*.

A rotação ocorre mediante determinada posição da mão direita ou esquerda no mundo real (ver Quadro A.3 do Apêndice A) e se dá em torno do eixo x da tela e do eixo z do Blender. Para esta operação optou-se efetuá-la usando a matriz de rotação da visão da cena, pelo fato de permitir maior flexibilidade no cálculo e precisão.

Já a translação e o *zoom* podem ocorrer de forma “simultânea”, ou seja, para ativar o modo de translação/*zoom* a pose a ser efetuada pelo usuário é a mesma (ver

Quadro A.2 do Apêndice A), porém o movimento que realiza cada transformação é diferente, sendo para transladar o gesto de deslocar a mão nos eixos x e y , e para afastar ou aproximar o ponto de vista da cena o deslocar da mão no eixo z . Para ambas transformações utilizou-se os operadores nativos do Blender.

5.4.2.3 Seleção

Por seleção entende-se o processo de definir, dentre os objetos do ambiente tridimensional, sobre qual deles se deseja operar. Esta operação demandou o desenvolvimento de um algoritmo específico para seleção de objetos.

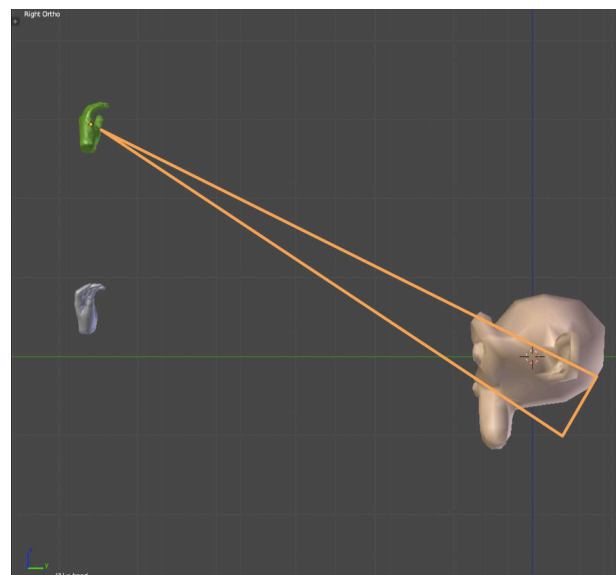
O algoritmo é dividido em 3 (três) partes:

1. Projeção em formato de cone a partir do ponto de vista do usuário;
2. Detecção de quais objetos essa projeção intercepta;
3. Determinação de qual objeto deve ser selecionado.

A Figura 5.5 ilustra o esquema de projeção adotado, caso haja vários objetos que são interceptados pelo cone o mais próximo da mão é tomado como o objeto alvo da seleção.



(a) Ponto de vista do usuário.



(b) Visão lateral direita com a projeção em formato de cone.

Figura 5.5: Projeção em formato de cone para seleção de objeto.

O Quadro A.4 que se encontra no Apêndice A descreve como selecionar um objeto.

5.4.2.4 Manipulação

A manipulação consiste na tarefa de alterar algum parâmetro de um objeto tridimensional que acarrete mudança em suas propriedades geométricas (tamanho, posição ou orientação) ou não. Neste tópico desenvolveu-se as 3 (três) operações de manipulação básicas, sendo: translação, rotação e redimensionamento. E ainda o suporte a extrusão de faces.

Para as manipulações básicas utilizou-se majoritariamente os operadores nativos do Blender. Já para a seleção de face o algoritmo é similar ao apresentado na seção 5.4.2.3 e a extrusão pode ser decomposta em:

1. Duplicar a face selecionada;
2. Calcular a posição final da face, baseado no movimento da mão do usuário;
3. Transladar a face para a posição final.

Os Quadros A.5, A.6 e A.7 que se encontram no Apêndice A descrevem as manipulações (implementadas) de um objeto.

5.5 Considerações finais

Neste capítulo foram discutidos os principais detalhes da implementação do protótipo apresentado neste trabalho. No próximo capítulo será apresentada uma breve discussão dos resultados obtidos, bem como alguns experimentos realizados para se fundamentar as conclusões deste trabalho.

Capítulo 6

Discussão dos resultados

6.1 Introdução

Neste capítulo serão apresentados os resultados obtidos neste trabalho de pesquisa, assim como uma discussão sobre os mesmos.

6.2 Protótipo

O código fonte do protótipo, resultado do desenvolvimento deste trabalho, será disponibilizado no fórum *blenderartists.org* na seção de trabalhos em andamento¹. O vídeo² em nota demonstra uma versão desse sendo operado por um usuário.

6.3 Avaliação da interface

Para avaliar o produto deste trabalho é imprescindível o julgamento do ponto de vista do usuário. Resumidamente, pode-se distinguir que uma avaliação desta natureza tem três grandes objetivos: avaliar a funcionalidade do sistema, avaliar o efeito da interface junto ao usuário e identificar problemas específicos do sistema [Rocha e Baranauskas 2003].

Conforme Rocha e Baranauskas [Rocha e Baranauskas 2003], para atender a esses objetivos pode-se classificar os métodos de avaliação em duas dimensões: se

¹ *Released Scripts and Themes*: <http://goo.gl/fKZVF>.

² <http://goo.gl/a10TD>.

usuários reais estão ou não envolvidos e se a interface está ou não implementada. Dentre os métodos pesquisados adotou-se o “testes de usabilidade”, no qual se traduz em métodos de avaliação centrados no usuário que incluem métodos experimentais ou empíricos, métodos observacionais e técnicas de questionamento. Rocha e Baranauskas apontam que métodos empíricos ou testes de usabilidade são o principal modo de avaliar interfaces e certamente o mais tradicional.

Em meio aos diversos métodos de inspeção existentes escolheu-se a Avaliação Heurística - é feita a inspeção da interface tendo como base uma pequena lista de heurísticas de usabilidade.

6.3.1 Avaliação heurística

A avaliação heurística é o principal método que compõem a denominada engenharia econômica de usabilidade, proposta por Curtis e Nielsen [Curtis e Nielsen 1995].

Esta avaliação é feita em um primeiro momento individualmente. Durante a sessão de avaliação cada avaliador percorre a interface diversas vezes (pelo menos duas) inspecionando os diferentes componentes do diálogo e ao detectar problemas os relata associando-os com as heurísticas de usabilidade que foram violadas. As heurísticas (definidas em [Nielsen 1993]), são regras gerais que objetivam descrever propriedades comuns de interfaces usáveis.

As heurísticas adotadas na avaliação do protótipo em questão se limitou à Tabela 6.1, pelo fato de o sistema ainda estar em desenvolvimento, por se tratar de um tipo de interface relativamente recente e por vezes carente de padrões para desenvolvimento e avaliação.

6.3.2 Etapas de um teste

Basicamente um teste é composto de quatro etapas [Rocha e Baranauskas 2003]:

- **Preparação** - Nessa etapa se garante que tudo estará pronto antes do usuário chegar.
- **Introdução** - Nessa os usuários são apresentados à situação de teste e colocados a vontade.

Tabela 6.1: Heurísticas adotadas (adaptado de [Nielsen 1993]).

Visibilidade do status do sistema

O sistema precisa manter o usuário informado sobre o que está acontecendo, fornecendo um *feedback* adequado dentro de um tempo razoável.

Compatibilidade do sistema com o mundo real

Seguir convenções do mundo real, fazendo com que a informação apareça em uma ordem natural e lógica.

Controle do usuário e liberdade

Prover funções *undo*, *redo* e claras saídas de emergência para estados indesejados.

Consistência e padrões

Seguir convenções de plataforma computacional.

Prevenção de erros

Design cuidadoso o qual previne o erro antes dele acontecer.

Estética e design minimalista

Diálogos não devem conter informações irrelevante ou raramente necessária.

- **Teste.**
- **Sessão final** - Após completar as tarefas os participantes são convidados a fazerem comentários ou sugestões gerais, ou a responderem um questionário específico.

6.3.3 Medidas de performance

Medidas típicas de usabilidade, que são quantificáveis, incluem [Rocha e Bara-nauskas 2003]:

- O tempo que o usuário gasta para fazer uma determinada tarefa;
- O número de erros do usuário;
- O número de comandos ou outras características que foram utilizados pelo usuário;
- O número de comandos ou outras características que nunca utilizados pelo usuário;
- O número de características do sistema que o usuário consegue se lembrar na sessão subsequente ao teste;
- A frequência de uso de manuais ou do sistema de *help* e o tempo gasto usando esses elementos;
- Quão frequentemente o manual/sistema de *help* ajudou a resolver o problema do usuário;
- A proporção entre comentários do usuário, favoráveis e críticos com relação ao sistema;
- O número de vezes que o usuário expressou frustração ou alegria;
- A proporção de usuários que disse preferir o sistema a outro sistema competidor;
- A quantidade de “tempo morto” - quando o usuário não está interagindo com o sistema (ou esperando resposta ou pensando);
- O número de vezes que o usuário desviou do objetivo da tarefa.

6.4 Experimentos

Umas das partes que compõem um experimento é a configuração física desse (quando aplicável), a Figura 6.1 demonstra esta configuração.

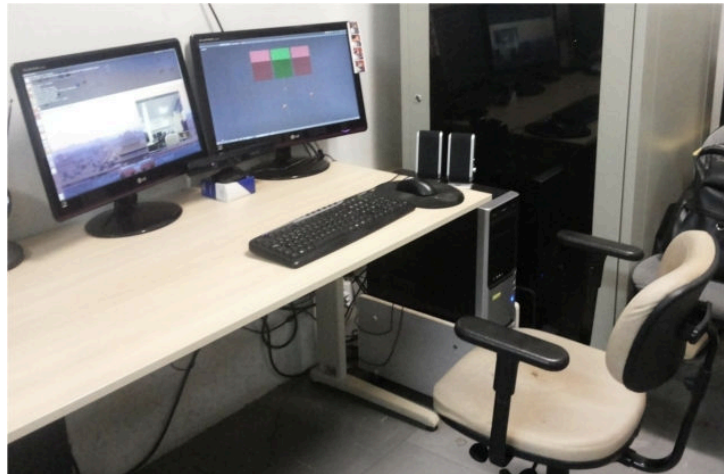


Figura 6.1: Configuração física do sistema.

Nesta configuração o usuário se senta na cadeira, com a possibilidade de apoiar os braços, em frente ao(s) monitor(es). O sensor Kinect é posicionado logo a frete do monitor e na altura do abdômen do usuário.

Os experimentos foram divididos em duas etapas. A primeira tem por objetivo avaliar, de forma preliminar, as estratégias para navegação na cena, e a segunda avaliar, mais uma vez, a navegação na cena, agora em conjunto com a estratégia para manipulação de objetos, além de julgar o protótipo como um todo.

6.4.1 Primeira etapa

Durante o desenvolvimento do protótipo conduziu-se uma avaliação preliminar³, em que se julgou o primeiro estágio de desenvolvimento da navegação na cena, assim como a comunicação entre os módulos do sistema. O vídeo⁴ em nota demonstra o protótipo no estágio mencionado.

Participaram desta avaliação 12 (doze) voluntários (as), sendo que 6 (seis) não possuíam experiência com ferramentas para modelagem geométrica 3D. Para estes apresentou-se os métodos tradicionais para navegação na cena do Blender (através de uma vídeo⁵ aula com aproximadamente 1 (um) minuto) e posteriormente o sistema

proposto neste trabalho (por meio, novamente, de uma vídeo aula⁴ com duração similar a anterior e um breve manual⁶ de inicialização do sistema), aos demais somente o último foi apresentado.

As tarefas a serem realizadas pelos participantes são: navegação na cena utilizando o mouse/teclado e o sistema NI2Blender sobre condições experimentais. Entende-se por navegação na cena a rotação, translação e *zoom* do ponto de visão da mesma.

A Figura 6.2 apresenta os gráficos que representam o resultado da avaliação preliminar realizada. Vale ressaltar que todas as seções de teste foram filmadas para posterior análise.

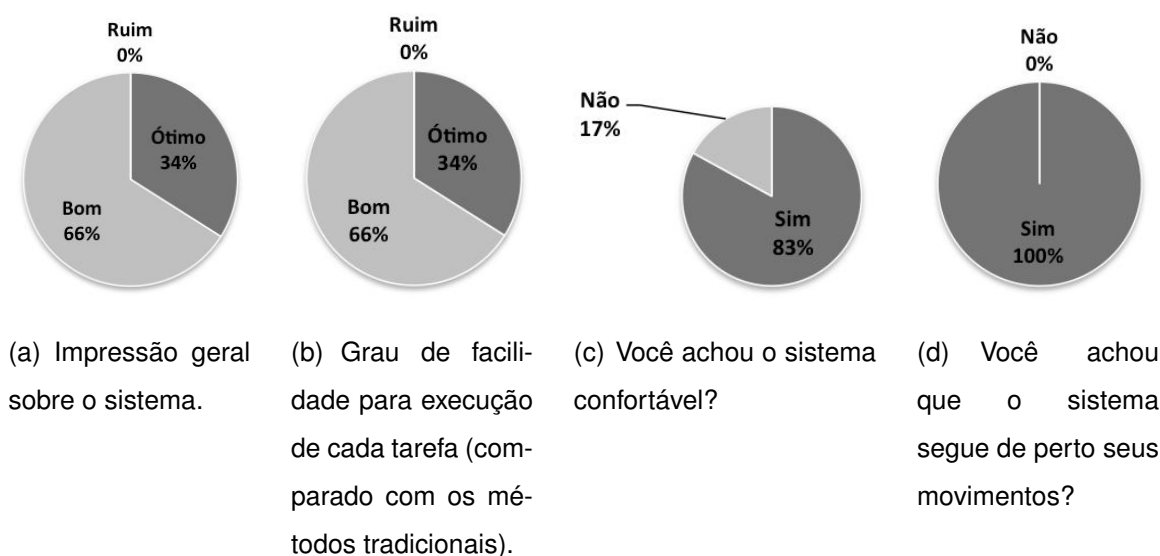


Figura 6.2: Resultado da avaliação preliminar do protótipo.

6.4.2 Segunda etapa

Nesta segunda avaliação, agora mais elaborada, objetivou-se julgar as funcionalidades e estratégias adotadas na versão mais recente do protótipo, que visa cumprir os requisitos especificados na seção 4.2 do Capítulo 4. Podendo-se destacar os seguintes tópicos:

- A navegação no ambiente tridimensional;

³Questionário correspondente ao primeiro experimento: <http://goo.gl/292jl>.

⁴<http://goo.gl/Gpo7j>

⁵<http://goo.gl/zdWIh>

⁶<http://goo.gl/HGKI2>

- A seleção e manipulação de objetos tridimensionais inseridos na cena;
- A estratégia de *feedback* usada;
- Os gestos/poses requeridas;
- O tempo de resposta.

A avaliação em questão segue o seguinte protocolo:

1. Apresentar ao usuário os dois sistemas (*mouse*/teclado e NI2Blender), ambos possuem um vídeo demonstrativo (já citado) e o segundo um manual (ver Apêndice A);
2. Deixar o usuário se familiarizar com ambos sistemas;
3. Explicar a tarefa ao usuário (mostrar uma *print* do objetivo);
4. Solicitar a execução da tarefa e contabilizar o tempo gasto usando-se ambos os sistemas;
5. Solicitar que o usuário responda o questionário⁷.

Demais rotinas:

- Filmar os testes;
- Gravar os resultados finais obtidos pelos avaliadores.

As tarefas solicitadas envolvem mover um objeto em um ambiente 3D para uma posição específica e alterá-lo de forma a ficar com as dimensões e orientação similares aos já pré posicionados (ver Figura 6.3). A posição inicial do cubo verde (a ser movido) é aleatória, porém a posição final deverá ser sempre a mesma.

Este experimento está em fase de execução, até o momento, conduziu-se-o com 3 (três) participantes. Na próxima seção serão apresentados e discutidos os apontamentos destas avaliações.

⁷Questionário correspondente ao segundo experimento: <http://goo.gl/HCc13>, para esse baseou-se na norma 9126-1 da ISO [ABNT 2003] e nas medidas apresentadas na seção 6.3.3.

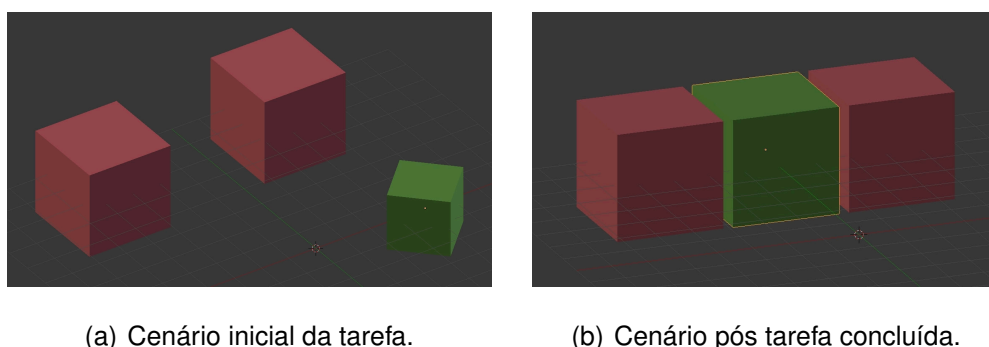


Figura 6.3: Exemplo de tarefa usada no segundo experimento.

6.5 Análise dos resultados

Conduziu-se um teste de usabilidade o qual objetiva avaliar o desempenho global do sistema. Neste estudo destacam-se dois objetivos principais, o primeiro é avaliar o desempenho global do sistema e o segundo é testar a usabilidade da interface desenvolvida. Devido aos testes não terem sido concluídos a análise desses tem caráter preliminar.

As análises podem ser divididas em 3 (três) vertentes: Funcionalidade, Usabilidade e Eficiência. As Figuras 6.4, 6.5 e 6.6 apresentam os gráficos que expressam parte dos resultados obtidos a partir da avaliação do sistema.

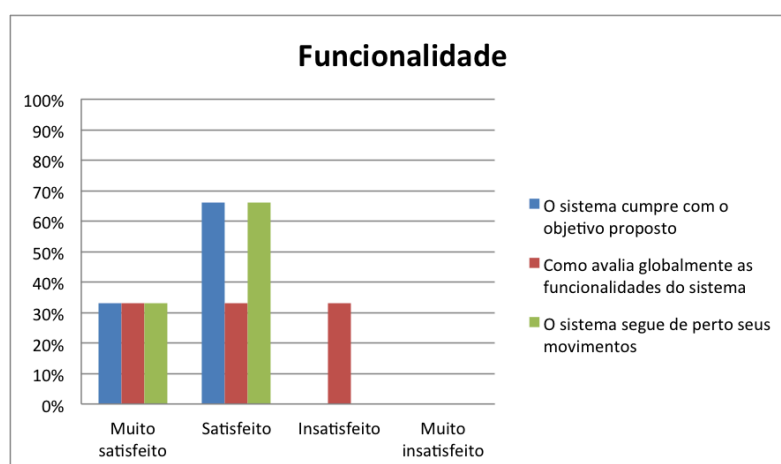


Figura 6.4: Resultados acerca das funcionalidades.

A Figura 6.4 expressa a capacidade do software em prover funcionalidades que satisfaçam o usuário em suas necessidades declaradas e implícitas, dentro de um determinado contexto de uso (controlar um modelador 3D de forma básica - translação,

rotação e escala).

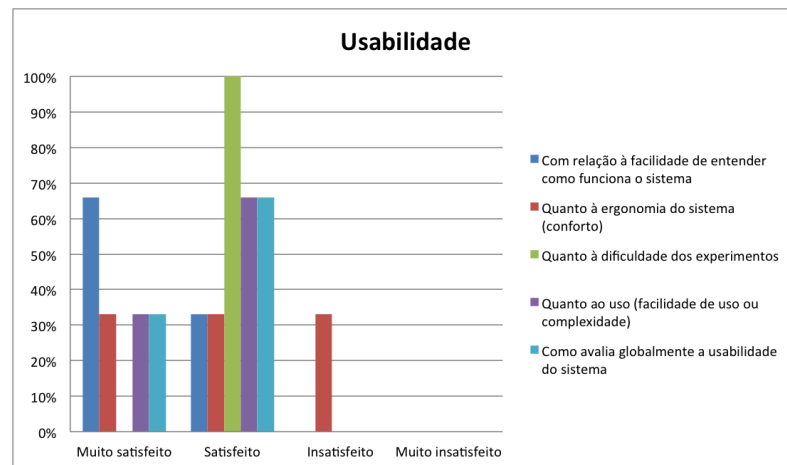


Figura 6.5: Resultados acerca da usabilidade.

A Figura 6.5 apresenta a capacidade do software ser compreendido, seu funcionamento aprendido, ser operado e ser atraente ao usuário.

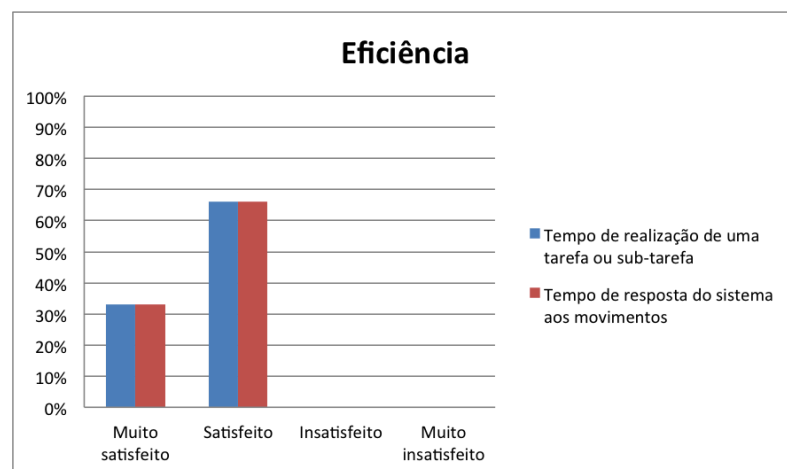


Figura 6.6: Resultados acerca da eficiência.

A Figura 6.6 exibi os resultados acerca do tempo de execução e se os recursos envolvidos são compatíveis com o nível de desempenho do software (se o sistema responde em tempo real).

O questionário (referente ao segundo experimento) também contém questões dissertativas e o resultado dessas podem ser condensados conforme a seguir:

- Funcionalidades que mais se destacaram de forma positiva: rotação, translação e zoom da visão da cena e o redimensionamento de objetos.

- Funcionalidades que mais se destacaram de forma negativa: o sistema para seleção de objetos e a inicialização do sistema (pose *psi*).
- Dificuldades encontradas: o entendimento do plano 3D e a execução do movimento *click*.

6.6 Considerações Finais

Neste Capítulo apresentou-se os métodos empregados para avaliação do sistema desenvolvido, os experimentos conduzidos a fim de validar o protótipo e por fim os resultados.

O Quadro 6.2 apresenta a contribuição deste trabalho em comparação aos trabalhos correlatos analisados no Capítulo 3. No próximo capítulo serão apresentadas as conclusões deste trabalho e as sugestões para trabalhos futuros.

Quadro 6.2: Resumo comparativo dos trabalhos correlatos e do presente trabalho.

Trabalhos relacionados	Captura de dados independente	Ambiente para modelagem 3D	Gestos e poses para navegação na cena e manipulação de objetos	Edição de malha	Teste de usabilidade
Santos, 2011	✗	✗	P	✗	✓
Matsumura e Sonnino, 2011	✗	✗	P	✗	✓
Shiratuddin, 2011	✗	✗	P	P	✗
Yang, 2012	✓	✓	P	✗	✓
Oliveira, 2013	✓	✓	✓	P	✓

⁸ P = Em partes.

A explanação acerca de cada elemento analisado e condensado no Quadro 6.2 se encontra na seção 3.6 do capítulo 3.

Capítulo 7

Conclusões e trabalhos futuros

7.1 Introdução

Neste capítulo destacam-se os aspectos da pesquisa e considerações finais acerca do trabalho apresentado nesta dissertação. Além disso, serão apresentadas algumas sugestões para trabalhos futuros e a contribuição científica do presente trabalho.

7.2 Considerações finais

O desconforto dos usuário (e potenciais) perante a usual e complexa interface dos programas para modelagem 3D é um dos fatores que culminam na desistência de seu uso por parte de muitos usuários.

Neste sentido, o objetivo deste trabalho consistiu em investigar formas de interação que se figurem mais naturais e utilizá-las em uma aplicação para modelagem 3D. A partir do estudo do estado da arte percebeu-se que a utilização das mãos, através de poses e gestos, consiste em uma interface que pode ser mais intuitiva e cognitivamente vantajosa para utilização.

Desta forma, este trabalho apresenta um sistema protótipo que usa o rastreamento 3D das mãos e reconhecimento de poses para controlar o Blender 3D. Neste sistema usa-se o sensor Kinect para obter o rastreamento 3D em tempo real. A posição das mãos do usuário são extraídas por meio do mapa de profundidade provido pelo Kinect e algoritmos desempenham o reconhecimento de poses e gestos. No sistema utiliza-se a mão esquerda e direita para interagir, sendo a mão esquerda utilizada de forma

secundária (auxiliando nas poses e movimentos que envolvem as duas mãos) e a direita em primeiro plano (utilizada para os gestos e controles de movimento envolvendo o espaço virtual 3D). As coordenadas 3D das mãos e eventuais outras partes do corpo são enviadas ao *software* Blender via conexão TCP. E por meio de *scripts* internos, o usuário é habilitado a acessar as funções de modelagem dentro do Blender.

A aplicação foi desenvolvida com as linguagens de programação C++ e Python, usando bibliotecas compatíveis com os sistemas operacionais Windows, Linux e Mac OS X, visando a possibilidade de se utilizar a aplicação nestes SOs, sendo necessário para isso a compilação da aplicação para o SO em questão.

Quanto ao desempenho, o conjunto apresentou taxas de quadros por segundo superiores a 25 FPS. Isso se deve a um conjunto de fatores, que vão desde a linguagem de programação escolhida, até a eficiência e capacidade dos módulos de *software/hardware* implementados e utilizados.

Foram conduzidos experimentos a fim de comparar o sistema de rastreamento 3D da mão com o uso do mouse e teclado nas tarefas de navegação na cena e manipulação de objetos, a avaliação teve natureza qualitativa (em primeira instância). Uma avaliação quantitativa da eficiência do mouse/teclado versus interface gestual baseada no rastreamento das mãos sugere uma interessante proposta para trabalhos futuros.

Portanto, qual é a vantagem/utilidade de atuar em modelagem a distância? Trabalhar utilizando três graus de liberdade somado a gestos com as mãos para interagir em um ambiente 3D é mais lógico, porém ainda não tão desenvolvido quanto o mouse. Entretanto, pelo fato de ser mais lógico e intuitivo deduz-se, e a experiência comprova, que este modo de interação pode chegar perto de ter um desempenho similar ao mouse.

Por meio de uma interface gestual pode-se fazer, por exemplo, o trabalho de pré modelagem e o protótipo apresentado neste trabalho pode ser aplicado como um jogo ou ainda como uma ferramenta para reabilitação física.

Com este trabalho foi possível avaliar alguns dos principais componentes de um sistema voltado para interação em ambientes tridimensionais (com foco em prover uma interface gestual para modelagem 3D) a partir de gestos com as mãos. As principais contribuições deste trabalho são:

- Proposta da arquitetura de comunicação produtor/consumidor entre o sistema

de captura de dados e o ambiente para modelagem 3D, em uma aplicação em tempo real;

- Proposta e implementação dos algoritmos para mapeamento das interações em conjunto com a metáfora de interação utilizada;
- Análise da aplicação do Kinect na tarefa de capturar movimentos para manipulação de um modelador 3D;
- A comprovação de que é possível adequar os programas, sobretudo os para modelagem 3D, já existentes a outras modalidades de interação;
- Proposta de avaliação para sistemas de natureza e objetivos similares.

7.3 Trabalhos futuros

Os desdobramentos deste trabalho podem ser divididos em dois conjuntos: o que compreende nas melhorias e incrementos a "curto prazo" e o que engloba a investigação e desenvolvimento a "médio e longo prazo".

Curto prazo:

- Avaliação e proposição de outras formas para aplicar/finalizar uma operação;
- Suporte a grades 3D para apoiar movimentos com mais precisão;
- Inclusão da possibilidade de mudança da resolução durante a interação;
- Quantificação da eficiência do uso de *buffer* circular como estratégia para intercâmbio e processamento de dados;
- Criação de objetos por revolução/extrusão e operações booleanas;
- Redução da fadiga induzida a partir da utilização de sistemas desta natureza, cuja os movimentos não podem ser executados sempre com os braços apoiados. Para amenizar este problema uma possível abordagem é ajustar a sensibilidade no controle, então a magnitude dos movimentos requeridos será reduzida;
- Quantificação detalhada do quesito eficiência, analisado no capítulo 6.

Médio e longo prazo:

- Investigação de novas *widgets* para interfaces *touchless*;
- Customização da captura dos gestos;
- Desenvolvimento de uma meta interface, com a possibilidade de ser customizável;
- Suporte a gestos em ambas as mãos, similar ao encontrado na tecnologia *multi-touch*;
- Extensão da arquitetura de forma a suportar sensores mais precisos;
- Suporte a gestos personalizados para a interação no processo de modelagem 3D;
- Condução de avaliações usando outras escalas, como a de Likert por exemplo.

Referências Bibliográficas

[ABNT 2003]ABNT. *NBR ISO/IEC 9126-1: Engenharia de software - Qualidade de produto-Parte 1: Modelo de qualidade*. [S.l.]: Rio de Janeiro: Associação Brasileira de Normas Técnicas, 2003.

[Bai et al. 2012]BAI, L. et al. Application of low cost inertial sensors to human motion analysis. In: *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*. [S.l.: s.n.], 2012. p. 1280 –1285. ISSN 1091-5281.

[Blender Developers 2013]Blender Developers. *Operator design notes*. 2013. Disponível em: <http://wiki.blender.org/index.php/Dev:2.5/Source/Architecture/Window_Manager>. Acesso em: 5 jul 2013.

[Blender Developers 2013]Blender Developers. *Scripts Development*. 2013. Disponível em: <<http://wiki.blender.org/index.php/Dev:2.5/Py/Scripts>>. Acesso em: 5 jul 2013.

[Blender Foundation 2013]Blender Foundation. *Blender home page*. 2013. Disponível em: <<http://www.blender.org/>>. Acesso em: 5 jul 2013.

[Blender Foundation 2013]Blender Foundation. *Operator*. 2013. Disponível em: <http://www.blender.org/documentation/blender_python_api_2_67_1/bpy.ops.html>. Acesso em: 5 jul 2013.

[Bolt 1980]BOLT, R. A. Put-that-there: Voice and gesture at the graphics interface. In: *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1980. (SIGGRAPH '80), p. 262–270. ISBN 0-89791-021-4. Disponível em: <<http://doi.acm.org/10.1145/800250.807503>>.

[Brito 2010]BRITO, A. *Blender 3D-Guia do Usuário*. [S.l.]: Novatec Editora, 2010.

- [Buxton 2013]BUXTON, B. *Multi-Touch Systems that I Have Known and Loved*. 2013. Disponível em: <<http://www.billbuxton.com/multitouchOverview.html>>. Acesso em: 5 jan 2013.
- [Chao 2009]CHAO, G. Human-computer interaction: Process and principles of human-computer interface design. In: *Computer and Automation Engineering, 2009. ICCAE '09. International Conference on*. [S.l.: s.n.], 2009.
- [Cruz, Lucio e Velho 2012]CRUZ, L.; LUCIO, D.; VELHO, L. Kinect and rgbd images: Challenges and applications. In: *Graphics, Patterns and Images Tutoriais (SIBGRAPI-T), 2012 25th SIBGRAPI Conference on*. [S.l.: s.n.], 2012. p. 36 –49.
- [Curtis e Nielsen 1995]CURTIS, B.; NIELSEN, J. Applying discount usability engineering. *IEEE Softw.*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 12, n. 1, p. 98–100, jan. 1995. ISSN 0740-7459. Disponível em: <<http://dx.doi.org/10.1109/52.363161>>.
- [Dam 2012]DAM, P. F. *Um estudo sobre técnicas de navegação e seleção em Ambientes Virtuais usando o Microsoft Kinect*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro, Novembro 2012.
- [Dix et al. 2005]DIX, A. et al. *Human-computer interaction*. [S.l.]: Prentice hall, 2005.
- [Duan, Deng e Wang 2012]DUAN, Y.; DENG, H.; WANG, F. Depth camera in human-computer interaction: An overview. In: *Intelligent Networks and Intelligent Systems (ICINIS), 2012 Fifth International Conference on*. [S.l.: s.n.], 2012. p. 25 –28.
- [Hirani e Brown 2010]HIRANI, K.; BROWN, N. *Sony Computer Entertainment Europe Research & Development Division*. Abril 2010. Nordic Game Conference. Disponível em: <<http://develop.scee.net/files/presentations/nordic/NordicGameDevServices-Moveand3D.pdf>>. Acesso em: 5 jan 2013.
- [Iacolina et al. 2010]IACOLINA, S. et al. Natural interaction and computer graphics applications. In: THE EUROGRAPHICS ASSOCIATION. *Eurographics Italian Chapter Conference 2010*. [S.l.], 2010. p. 141–146.
- [IBM 2001]IBM. *Application Development Guide: Core Components*. [S.l.]: IBM Corporation, 2001.

- [Jain, Lund e Wixon 2011]JAIN, J.; LUND, A.; WIXON, D. The future of natural user interfaces. In: *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2011. (CHI EA '11), p. 211–214. Disponível em: <<http://doi.acm.org/10.1145/1979482.1979527>>.
- [Kirner e Siscoutto 2007]KIRNER, C.; SISCOOTTO, R. *Realidade virtual e aumentada: conceitos, projeto e aplicações*. [S.l.]: Porto Alegre: Sociedade Brasileira de Computação, 2007.
- [Lee 2008]LEE, J. Hacking the nintendo wii remote. *Pervasive Computing, IEEE*, v. 7, n. 3, p. 39 –45, july-sept. 2008. ISSN 1536-1268.
- [Liu 2010]LIU, W. Natural user interface- next mainstream product user interface. In: *Computer-Aided Industrial Design Conceptual Design (CAIDCD), 2010 IEEE 11th International Conference on*. [S.l.: s.n.], 2010. v. 1, p. 203 –205.
- [Matsumura e Sonnino 2011]MATSUMURA, K. K.; SONNINO, R. *Fusion4D - Interface natural e imersiva para manipulação de objetos 3D*. 2011. Monografia apresentada à Escola Politécnica da Universidade de São Paulo.
- [Mattioli 2012]MATTIOLI, F. E. R. *Utilização de Técnicas de Realidade Virtual na Simulação de Próteses de Mão*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Fevereiro 2012.
- [Microsoft 2012]MICROSOFT. *Human Interface Guidelines: Kinect for Windows v1.5.0*. [S.l.], 2012.
- [Nielsen 1993]NIELSEN, J. *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 0125184050.
- [Nintendo 2013]NINTENDO. *Nintendo Wii Homepage*. 2013. Disponível em: <<http://www.nintendo.com/wii>>. Acesso em: 5 jan 2013.
- [OGRE 2013]OGRE. *About*. 2013. Disponível em: <<http://www.ogre3d.org/about>>. Acesso em: 10 jul 2013.
- [Olszta, Umbach e Baker 2013]OLSZTA, P. W.; UMBACH, A.; BAKER, S. *FreeGLUT home page*. 2013. Disponível em: <<http://freeglut.sourceforge.net/>>. Acesso em: 5 jul 2013.

- [OMG 2013]OMG. *Documents associated with Unified Modeling Language (UML)*. 2013. Disponível em: <<http://www.omg.org/spec/UML/2.4.1/>>. Acesso em: 5 jun 2013.
- [OpenKinect 2013]OpenKinect. *libfreenect Readme*. 2013. Disponível em: <<https://github.com/OpenKinect/libfreenect/>>. Acesso em: 5 jan 2013.
- [OpenNI 2011]OPENNI. *OpenNI User Guide*. [S.l.], 2011.
- [Penn, Schoen & Berland 2011]Penn, Schoen & Berland. *Global opportunities for Natural User Interface technology*. Janeiro 2011.
- [Perl 2010]PERL, T. *PS Move API*. 2010. Disponível em: <<http://thp.io/2010/psmove/>>. Acesso em: 5 jan 2013.
- [Perl 2012]PERL, T. *Cross-Platform tracking of a 6DoF motion controller*. Dissertação (Mestrado) — Vienna University of Technology, Dezembro 2012.
- [Pinho 2000]PINHO, M. S. *Interação em Ambiente Tridimensionais*. 2000. Disponível em: <<http://www.inf.pucrs.br/~pinho/3DInteraction/>>. Acesso em: 11 jul 2013.
- [PrimeSense 2011]PRIMESENSE. *NITE Controls User Guide*. [S.l.], 2011.
- [Rocha e Baranauskas 2003]ROCHA, H. V. da; BARANAUSKAS, M. C. C. *Design e avaliação de interfaces humano-computador*. [S.l.]: Universidade Estadual de Campinas, 2003.
- [Santos 2011]SANTOS, E. de S. *Manipulação de objetos 3D em aplicações de Realidade Aumentada por meio da movimentação da mão*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Julho 2011.
- [Shiratuddin e Wong 2011]SHIRATUDDIN, M.; WONG, K. W. Non-contact multi-hand gestures interaction techniques for architectural design in a virtual environment. In: *Information Technology and Multimedia (ICIM), 2011 International Conference on*. [S.l.: s.n.], 2011. p. 1 –6.

- [Shotton et al. 2011]SHOTTON, J. et al. Real-time human pose recognition in parts from single depth images. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. [S.l.: s.n.], 2011. p. 1297–1304. ISSN 1063-6919.
- [Smisek, Jancosek e Pajdla 2011]SMISEK, J.; JANCOSSEK, M.; PAJDLA, T. 3D with Kinect. In: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. [S.l.: s.n.], 2011. p. 1154–1160.
- [Sony 2013]SONY. *Playstation Move Homepage*. 2013. Disponível em: <<http://br.playstation.com/ps3/playstation-move/>>. Acesso em: 5 jan 2013.
- [Sony Computer Entertainment Inc 2011]Sony Computer Entertainment Inc. *Move.Me User's Guide*. [S.l.], 2011.
- [Sung 2011]SUNG, K. Recent videogame console technologies. *Computer*, v. 44, n. 2, p. 91–93, feb. 2011. ISSN 0018-9162.
- [Sutherland 1964]SUTHERLAND, I. E. Sketch pad a man-machine graphical communication system. In: *Proceedings of the SHARE design automation workshop*. New York, NY, USA: ACM, 1964. (DAC '64), p. 6.329–6.346. Disponível em: <<http://doi.acm.org/10.1145/800265.810742>>.
- [Thalen 2011]THALEN, J. *Using Blender for Rapid Virtual Reality Prototyping*. Outubro 2011. Blender Conference. Disponível em: <<http://download.blender.org/documentation/bc2011/JosThalen.pdf>>. Acesso em: 5 fev 2013.
- [Thalen e Voort 2011]THALEN, J.; VOORT, M. van der. User centred methods for gathering vr design tool requirements. In: THE EUROGRAPHICS ASSOCIATION. *Joint Virtual Reality Conference of EGVE-EuroVR*. [S.l.], 2011. p. 75–81.
- [Valli 2008]VALLI, A. The design of natural interaction. *Multimedia Tools and Applications*, p. 295–305, 2008.
- [Villaroman, Rowe e Swan 2011]VILLAROMAN, N.; ROWE, D.; SWAN, B. Teaching natural user interaction using openni and the microsoft kinect sensor. In: *Proceedings of the 2011 conference on Information technology education*. [S.l.: s.n.], 2011. p. 227–232.

- [Webb e Ashley 2012]WEBB, J.; ASHLEY, J. *Beginning Kinect Programming with the Microsoft Kinect SDK*. [S.l.]: Apress, 2012.
- [Williamson et al. 2011]WILLIAMSON, B. et al. Natural full body interaction for navigation in dismounted soldier training. In: NTSA. *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*. [S.l.], 2011. v. 2011.
- [Yang et al. 2012]YANG, R. S. et al. Bimanual natural user interaction for 3d modelling application using stereo computer vision. In: *Proceedings of the 13th International Conference of the NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction*. New York, NY, USA: ACM, 2012. (CHINZ '12), p. 95–95. ISBN 978-1-4503-1474-9. Disponível em: <<http://doi.acm.org/10.1145/2379256.2379279>>.
- [Yang e Chen 2009]YANG, X.; CHEN, G. Human-computer interaction design in product design. In: *Education Technology and Computer Science, 2009. ETCS '09. First International Workshop on*. [S.l.: s.n.], 2009. v. 2, p. 437 –439.

Apêndice A

Manual do Usuário - NI2Blender

Autores:

Fábio Henrique M. Oliveira e

Grupo GRVA UFU

Dicas:

- **Mantenha as mãos no campo de visão do sensor;**
- **Não se aproxime demais do sensor, pois isto prejudica o rastreamento das mãos;**
- **Procure não fazer movimentos bruscos.**

1 - Realizar a pose ilustrada na Figura A.1(a)¹ para que o sistema o reconheça como usuário ativo (aguardar *feedback* no ambiente virtual - As mãos virtuais mudarão de cor, para amarelo).

2 - Realizar gesto *wave*² ilustrado na Figura A.1(b) para inicializar uma sessão de interação com o sistema (aguardar o *feedback* no ambiente virtual - As mãos virtuais mudarão de cor, para verde).

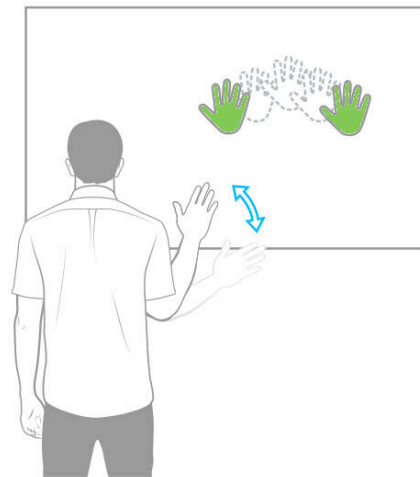
¹Todas as imagens do sistema NI2Blender estão espelhadas.

²Gesto utilizado para tomar o controle da aplicação para a mão desejada (pode ser usado após o usuário ser reconhecido e também a qualquer momento após este reconhecimento, a fim de passar o controle dos gestos para a mão que o faz).

³Uma versão rascunho deste manual está disponível em: <http://goo.gl/ii5dw>

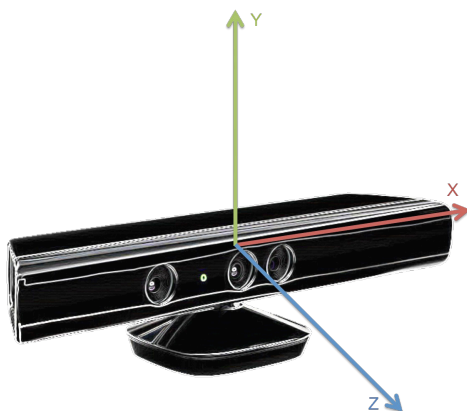


(a) Pose “psi”.

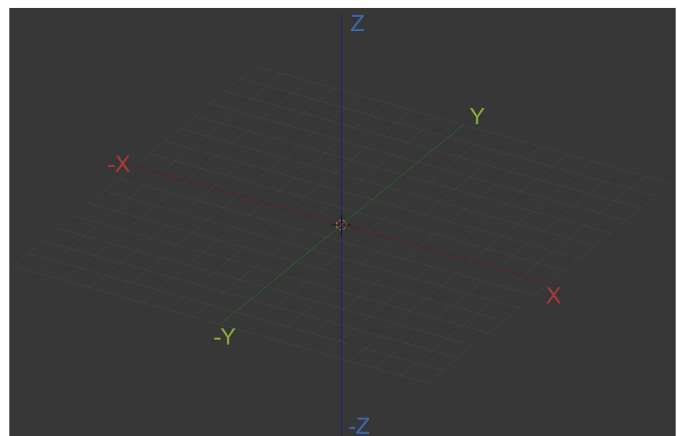


(b) Movimento wave.

Figura A.1: Passos para iniciar uma sessão.







(a) Kinect.

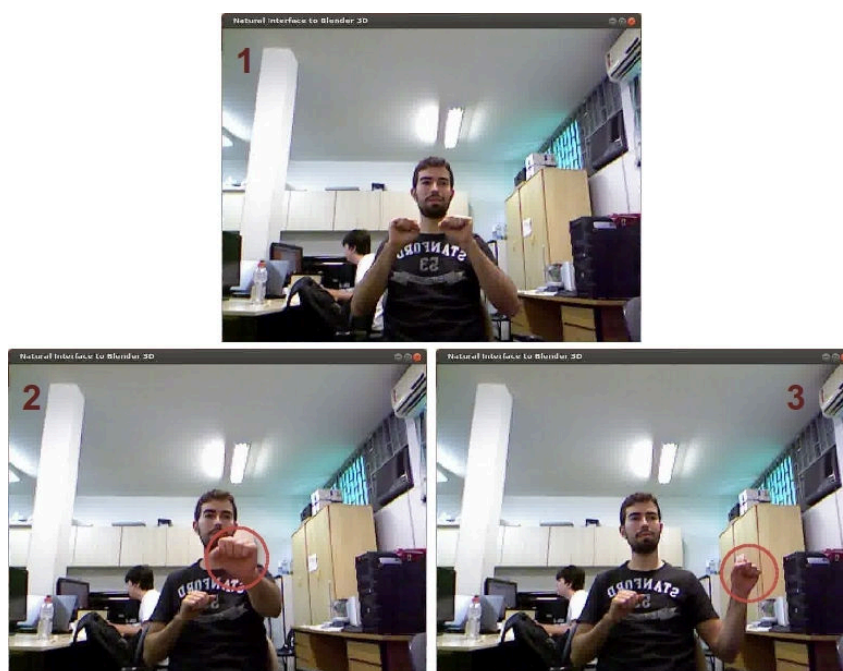


(b) Palco do Blender3D sinalizado com as coordenadas.

Figura A.2: Sistemas de coordenadas.

Quadro A.1: *Feedback* visual do sistema através das mãos.

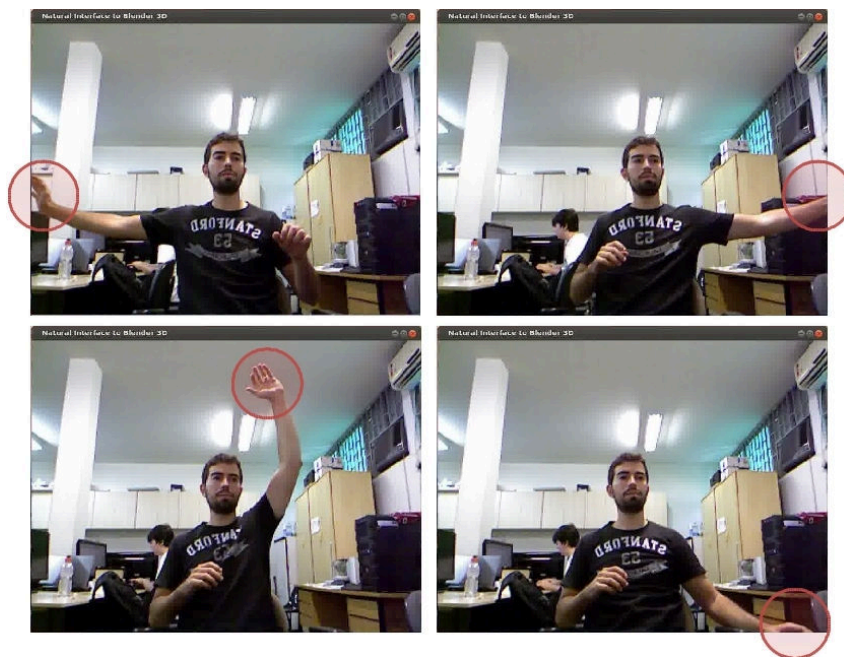
Cor das mãos	Significado
 Vermelha	<ul style="list-style-type: none">- Sistema inicializado, porém nenhum usuário foi detectado.- Para calibrar, o usuário precisa manifestar interesse. Neste caso, através da posição corporal “psi” (ver Figura A.1(a)).
 Amarela	<ul style="list-style-type: none">- Usuário reconhecido e calibrado, porém ainda não operando o sistema.- Para operar, o usuário precisa “chamar a atenção do sistema”, com o gesto <i>wave</i> (ver Figura A.1(b)).
 Verde	<ul style="list-style-type: none">- Sistema operando e capturando os gestos e posições do usuário.
 Azul	<ul style="list-style-type: none">- Sinaliza que uma operação está em andamento e qual mão está responsável/envolvida na mesma.

Quadro A.2: Instruções para aplicar *Zoom/Pan* na visão da cena.

- Posicionar as mãos conforme a Figura 1 e manter por ~1.5 segundos (a mão direita ficará azul, indicando que a pose foi reconhecida).
- Movimentar a mão direita conforme o movimento que se deseja executar no ambiente virtual. A Figura 2 e 3 exemplificam, respectivamente, os movimentos para *zoom* e *pan* na visão da cena.
- Para aplicar as alterações pare as mãos por ~1.5 segundos.

Observação: Para efetuar *zoom/pan*, nenhum objeto pode estar selecionado.

Quadro A.3: Instruções para aplicar rotação na visão da cena.



- Para rotacionar a visão da cena o usuário pode, a qualquer momento, “tocar” a extremidade do campo de visão do sensor (conforme as Figuras acima).

Observação: Apenas no caso em que a mão está acima da cabeça do usuário, não é requerido que o mesmo “toque” a extremidade superior do campo de visão do sensor.

Quadro A.4: Instruções para seleccionar objeto.



- Para seleccionar um objeto posicione a mão virtual sobre o mesmo e efetuar o gesto "click" (a borda do objeto mudará de cor, caso o gesto seja reconhecido).

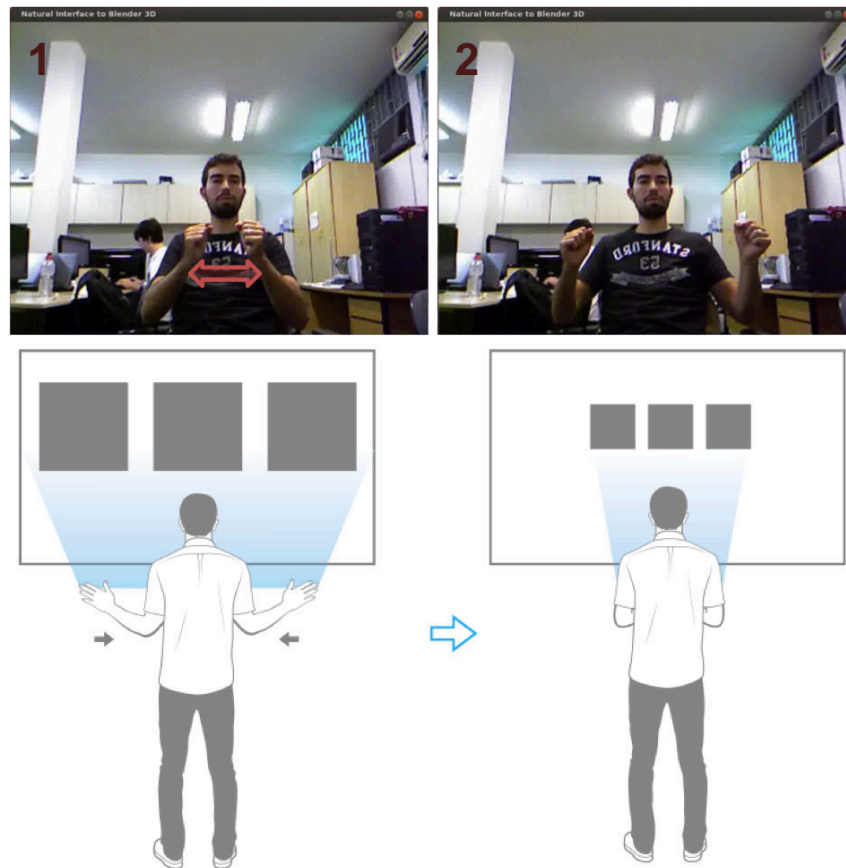
Observações:

- Para desseleccionar o objeto clique em qualquer parte vazia da cena.
- Ocorre, as vezes, do sistema de detecção de gestos passar o controle para a mão esquerda (o NI2Blender usa por padrão a mão direita), para voltar o controle para a mão direita execute o movimento wave ilustrado na Figura 5.1(b) a partir da mesma.

Quadro A.5: Instruções para transladar objeto.

- Efetuar novamente o gesto "click" sobre o objeto, a mão virtual direita ficará azul, a partir deste *feedback* o usuário pode movimentar a mão direita que o objeto seguirá tais movimentos.
- Para aplicar a translação pare as mãos por ~1.5 segundos.

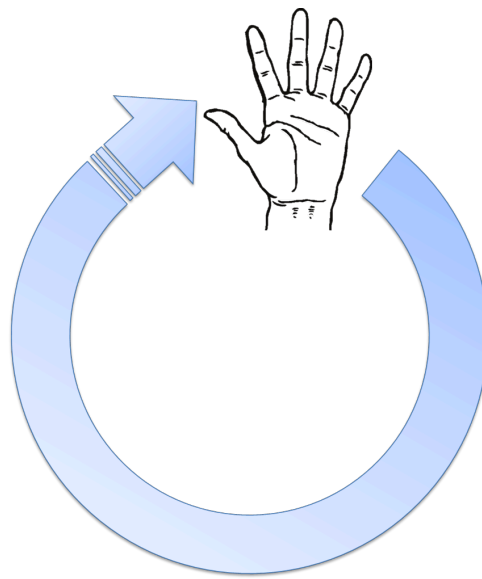
Quadro A.6: Instruções para redimensionar objeto.



- Para redimensionar o objeto o usuário deve tê-lo selecionado previamente.
- O usuário deve posicionar as mãos de acordo com a Figura 1 ou 2 e manter as mãos paradas por ~1.5 segundos.
- Na sequência, realizar o movimento demonstrado na Figura acima.
- Para aplicar as modificações pare as mãos por ~1.5 segundos.

Observações: Caso nenhum objeto esteja selecionado, o modo de navegação *zoom/pan* será ativado (pois trata-se da mesma posição de ativação).

Quadro A.7: Instruções para rotacionar objeto.



- Para rotacionar o objeto o usuário precisa tê-lo selecionado previamente.
- Então com o gesto de mover a mão em forma de círculo (Figura acima) o modo de rotação de objeto é ativado (a mão virtual direita ficará azul, caso o gesto seja reconhecido).
- Após ativado o modo, basta movimentar a mão direita para cima/baixo ou para a esquerda/direita, que o objeto será rotacionado seguindo os movimentos da mão.