

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



**Manipulação de Objetos 3D em Ambientes
Colaborativos com o uso do Dispositivo Kinect**

Wedson Gomes da Silveira Junior

**Dezembro
2012**

Manipulação de Objetos 3D em Ambientes Colaborativos com o uso do Dispositivo Kinect

Wedson Gomes da Silveira Junior

Texto apresentado à Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Mestre em Ciências.

Prof. Edgard A. Lamounier Jr, PhD.
Orientador

Prof. Alexandre Cardoso, Dr.
Co-orientador

Prof. Alexandre Cardoso, Dr.
Coordenador do curso de Pós-Graduação

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

**Manipulação de Objetos 3D em Ambientes Colaborativos com o uso
do Dispositivo Kinect**

Wedson Gomes da Silveira Junior

Texto apresentado à Universidade Federal de Uberlândia, perante a banca de examinadores abaixo, como parte dos requisitos necessários para a obtenção do título de Mestre em Ciências.

Banca Examinadora:

Prof. Edgard A. Lamounier Jr, PhD. – Orientador (UFU)

Prof. Alexandre Cardoso, Dr. – Co-orientador (UFU)

Prof. Mônica Rocha Ferreira de Oliveira, Dr. (ISED)

Prof. Maria Fernanda Soares de Almeida, Dr. (UFU)

Agradecimentos

A Deus, pelas oportunidades colocadas em minha vida.

Aos meus pais, Wedson e Irene, que são meus maiores exemplos de vida e sem sua dedicação e carinho não poderia chegar onde estou.

As minhas irmãs, Cibele e Carolina, pelas risadas e companheirismo.

A minha noiva, Débora, pelo companheirismo e palavras de incentivo nos momentos difíceis.

Aos professores Alexandre Cardoso, Dr. e Edgard Afonso Lamounier Jr., PhD, pelo imenso apoio e conselhos, não somente neste trabalho, mas durante toda a graduação e mestrado. Ambos se tornaram grandes exemplos e referência para minha carreira.

Aos colegas de laboratório que serão para sempre amigos e companheiros de carreira.

Publicações

JUNIOR, W. G. S.; Cardoso, A; Lamounier Jr., E.A.. **Manipulação de Objetos 3D em Ambientes Colaborativos por Meio de Gestos Naturais.** In: IV Jornada Científica e Tecnológica e I Simpósio de Pós-graduação do IFSULDEMINAS 2012, Muzambinho-MG.

Resumo

JUNIOR, Wedson G. S. *Manipulação de Objetos 3D em Ambientes Colaborativos com o uso do Dispositivo Kinect*, Uberlândia, Faculdade de Engenharia Elétrica – UFU, 2012.

Palavras-chave: Interfaces Naturais, Realidade Virtual, Ambientes Virtuais Colaborativos

As pesquisas na área de interação natural vêm crescendo significativamente, pois, com a disseminação de computadores pessoais, existe uma demanda crescente por interfaces que maximizem a produtividade. Dentre essas interfaces podemos destacar a Realidade Virtual e a Realidade Aumentada (KIRNER, TORI e SISCOUTO, 2006), de forma que os usuários possam realizar tarefas simples como: escolha de um objeto 3D, translação, rotação e mudança de escala neste objeto 3D. Essas tarefas normalmente são realizadas através de dispositivos como o teclado e o mouse, podendo, dessa maneira, acontecer uma perda de imersão por parte do usuário no ambiente virtual. Sendo assim, investigar metodologias para interação natural nesses ambientes, pode ajudar a aumentar a imersão do usuário, no ambiente virtual.

Outra questão que vem sendo o foco de muitas pesquisas são os Ambientes Virtuais Colaborativos (KIRNER e TORI, 2004), que permitem aos usuários se comunicarem e compartilhar informações. Esses usuários podem estar próximos fisicamente ou não. O foco principal desse trabalho é justamente a comunicação entre usuários dispersos remotamente.

Dessa maneira, pretende-se, com esse trabalho, propor um sistema que possibilite a manipulação objetos 3D através de gestos naturais e compartilhar dados com usuários dispersos remotamente.

Abstract

JUNIOR, Wedson G. S. *Manipulação de Objetos 3D em Ambientes Colaborativos por Meio de Interfaces Naturais*, Uberlândia, Faculdade de Engenharia Elétrica – UFU, 2012.

Palavras-chave: M-Learning, Virtual Reality, Augmented Reality

The research on natural interaction has been growing significantly since, with the spread of personal computers, there is an increasing demand for interfaces that maximize productivity. Among these we can highlight the interfaces Virtual Reality and Augmented Reality (KIRNER, TORI e SISCOUTO, 2006), where users can perform simple tasks such as choosing a 3D object translation, rotation and scaling in 3D object.

These tasks are usually performed through devices such as the keyboard and mouse, and can, thus, a loss of immersion happen by the user in the virtual environment. Thus, investigating methodologies for natural interaction in these environments, can help increase user immersion in the virtual environment.

Another issue that have been the focus of many researches are Collaborative Virtual Environments (KIRNER e TORI, 2004), where it is possible for users to communicate and share information. These users can be physically close or not. The main focus of this work is precisely the communication among remotely dispersed users.

Thus, it is intended with this work, proposing a system where it is possible to manipulate 3D objects using natural gestures and share data remotely dispersed users.

Sumário

Lista de Figuras	11
Lista de Tabelas e Quadros	13
Lista de Abreviaturas	14
Introdução	1
1.1 Considerações Iniciais.....	1
1.2 Motivação	2
1.3 Objetivos.....	4
Fundamentos.....	6
2.1 Ambientes Virtuais Colaborativos.....	6
2.2.1 Arquitetura Existentes para AVCs	7
✓ Tipo de protocolo de comunicação em AVC	7
✓ Envio de mensagens.....	8
✓ Distribuição dos dados	8
✓ Classificação para o sistema.....	9
2.2.2 Realidade Virtual.....	10
✓ Aplicações em Realidade Virtual.....	10
✓ Medicina.....	10
✓ Engenharia.....	11
✓ Educação	12
✓ Tipos de Sistemas de Realidade Virtual.....	13
✓ Manipulação de Objetos 3D	14
2.2.3 Interfaces	14
Interfaces Gestuais	15
2.2.4 Interfaces Naturais - NUI.....	16
✓ Dispositivos de <i>wearable</i>	17

✓ Técnica <i>Motion Capture</i>	17
✓ Dispositivos de <i>touchless</i>	19
Trabalhos Correlatos	21
3.1 Introdução.....	21
3.2 Flexible Action and Articulated Skeleton Toolkit (FAAST)	22
3.3 Kinoogle	23
3.4 GTAVCS – Arthron	25
3.5 Um Stand Virtual Automobilístico	27
3.6 Fusion 4D	28
3.7 Comparativo entre os trabalhos.....	31
Especificação do Sistema Proposto.....	32
4.1 Introdução.....	32
4.2 Especificação do Sistema.....	33
4.2.1 Requisitos do Sistema.....	33
4.2.2 Diagrama de Caso de Uso do Sistema	34
4.2.3 Diagrama de Atividades do Sistema	36
4.2.4 Arquitetura do Sistema.....	38
4.3 Considerações Finais	40
Detalhes da Implementação	41
5.1 Introdução.....	41
5.2 Tecnológicas de Apoio	41
5.2.1 Dispositivos <i>Touchless</i>	41
Dados Técnicos Kinect.....	42
5.2.2 <i>Softwares</i>	45
• <i>Driver</i> de Comunicação.....	45
• Ambiente de Desenvolvimento	46
5.3 Interface Gráfica.....	48
5.4 Comunicação com o Sensor e Detecção de Gestos	50
5.4.1 Módulo de Captura e Detecção.....	52
5.4.2 Módulo de Interação.....	54
5.5 Considerações Finais	57

Análise dos Resultados	58
6.1 Introdução.....	58
6.2 Manipulação de Objetos 3D.....	58
6.3 Uso de Interfaces Naturais	59
6.4 Ambientes colaborativos com usuários dispersos remotamente	62
6.5 Considerações Finais	63
 Conclusões e Trabalhos Futuros.....	66
7.1 Introdução.....	66
7.2 Conclusões.....	66
7.3 Trabalhos Futuros	67

Lista de Figuras

Figura 1 - Áreas do Conhecimento.....	4
Figura 2 - Unicast, Broadcast e Multicast (CERTIFICATIONS, 2012).....	8
Figura 3 - Ambiente de planejamento, fusão dos dados de ressonância magnética e tomografia do cérebro de um paciente (GUAN, 1998)	11
Figura 4 - Sistema para treinamento de sutura (WEBSTER, 2001).	11
Figura 5 - Software MineInside (COELHO, COELHO e CARDOSO, 2011).	12
Figura 6 - Laboratório Virtual (SILVA, 2008).	13
Figura 7 - Luva de dados (The Free Dictionary, 2012).....	15
Figura 8 - Primeira geração de HMDs da Nasa (KALAWSKY, 1993).....	16
Figura 9 - Evolução da Interface humano-computador (baseado em (REYES, 2009))	17
Figura 10 - Dispositivo GESTUS (COLLET e SILVA, 2011).....	18
Figura 11 - Dispositivo MOVE® (FORUMPC, 2012)	18
Figura 12 - Dispositivo Nintendo Wii® (Nintendo, 2012)	19
Figura 13 - Sensor Kinect® (baseado em MICROSOFT 2011).....	20
Figura 14 - Dispositivo Xtion PRO LIVE® (ASUS, 2011)	20
Figura 15 - Sensor Infra-vermelho.....	21
Figura 16 - Animando um avatar em tempo real com o FFAST (EVAN e BELINDA, 2011).	22
Figura 17 - FFAST + Google Earth (BOULOS, BLANCHARD, et al., 2011)	23
Figura 18 – Kinoogle Google Street View e Earth (BOULOS, BLANCHARD, et al., 2011).	24
Figura 19 – (a) Translação, (b) Escala, (c) e (d) Rotação (BOULOS, BLANCHARD, et al., 2011).	24
Figura 20 - (a) Caminhar em frente (b) Virar para os lados (BOULOS, BLANCHARD, et al., 2011)	25
Figura 21 - Seleção do Modelo 3D (CASTRO, 2012).....	26
Figura 22 - Interface de manipulação (CASTRO, 2012).....	26
Figura 23 - Interface de Utilização do Protótipo Desenvolvido (CARVALHO, SOUSA e OLIVEIRA, 2012).....	27

Figura 24 - Interface do Fusion 4D (MATSUMURA e SONNINO, 2011).....	29
Figura 25 - Vista explodida de um modelo 3D, acionada através do comando de voz "explode (MATSUMURA e SONNINO, 2011).....	30
Figura 26 - Modificação da escala de um objeto 3D, utilizando estereoscopia (MATSUMURA e SONNINO, 2011).	30
Figura 27- Diagrama de caso de uso.	35
Figura 28 - Diagrama de Sequencia (Colaboração entre usuários)	37
Figura 29 - Diagrama de atividade	38
Figura 30 - Arquitetura do sistema	39
Figura 31 - Kinect Desmontado (IFIXIT, 2012).....	43
Figura 32 - Descrição Kinect (baseado em (SDK, 2011)).....	43
Figura 33 - Capacidades físicas do kinect (baseado em KINECT 2012).....	44
Figura 34 - Interação Hardware e Software (CORPORATION, 2011).....	46
Figura 35 - WPF no Framework 3.0 (baseado em (SNEATH, 2008))	47
Figura 36 - Interface do aplicativo desenvolvido	48
Figura 37 - Componentes da Interface Gráfica	49
Código 2 - Modelagem dos objetos 3D	51
Código 3 - Funções para receber as imagens RGB e de profundidade.	52
Figura 38 - Gestos pré-definidos	53
Código 4 – Detecção dos gestos.....	53
Código 5 - Fragmento de Código para o reconhecimento de gestos	54
Código 6 - Fragmento de código para executar o gesto e enviar para o grupo.....	55
Código 7 - Fragmento de código para inserção de novo usuário.....	56
Figura 39 - Detecção com iluminação diferente	59
Figura 40 - Níveis de Reconhecimento dos Gestos.	61
Figura 41 - Teste com pessoa do sexo feminino e 28 anos	61
Figura 42 - Teste realizado com usuário de 53 anos	62
Figura 43 - Comunicação entre usuários.....	63
Figura 44 - Satisfação dos Usuários	64
Tabela 8 - Comparação dos Trabalhos Relacionados	65

Lista de Tabelas e Quadros

Tabela 1 – Quadro comparativo entre os trabalhos correlatos.....	31
Tabela 2 - Descrição do caso de uso "Compartilhar Dados".....	35
Tabela 3 - Descrição do caso de uso "Escolher Objeto 3D".....	36
Tabela 4 - Descrição do caso de uso "Manipular Objeto 3D".....	36
Tabela 5 - Dados Kinect (SDK, 2011)	44
Tabela 6 - Classes do Middleware SDK.....	46
Tabela 7 - Quantidade de Pessoas que participaram dos testes	60
Tabela 8 - Comparação dos Trabalhos Relacionados	65

Lista de Abreviaturas

AVC	Ambientes Virtuais Colaborativos
CSCW	<i>Computer Supported Cooperative Work</i>
CLI	<i>Command User Interface</i>
FPS	<i>Frames per Second</i>
GUI	<i>Graphical User Interface</i>
GTAVCA	Grupo de Trabalho em Vídeo Colaboração em Saúde
HMD	<i>Head-Mounted Display</i>
IDE	<i>Integrated Development Environment</i>
NITE	<i>Natural Interaction Technology for End-user</i>
NUI	<i>Natural User Interfaces</i>
RA	Realidade Aumentada
RGB	<i>Red, Green and Blue</i>
RV	Realidade Virtual
SDK	<i>Software Development Kit</i>
UML	<i>Unified Modeling Language</i>
VRPN	<i>Virtual Reality Peripheral Network</i>
WPF	<i>Windows Presentation Foundation</i>
XAML	<i>Extensible Application Markup Language</i>

Capítulo 1

Introdução

1.1 Considerações Iniciais

Desde que a tecnologia passou a fazer parte da vida humana, existe uma preocupação inerente ao seu uso. O desenvolvimento de novas ferramentas está atrelado, mesmo que inconscientemente, ao conceito de usabilidade (CHARYTONOWICZ, 2000).

Na busca de uma interface de operação ideal, que satisfaça completamente os atuais requisitos de usabilidade e naturalidade de operação, surgiram diversas tecnologias, dentre elas, pode-se destacar a Realidade Virtual (RV) e a Realidade Aumentada (RA) (KIRNER, TORI e SISCOUTO, 2006).

De uma maneira geral, tanto em aplicações de RV quanto de RA, o usuário necessita realizar tarefas simples, tais como: translação, rotação, criação e deleção de objetos virtuais. Estas, muitas vezes, são realizadas de formas não naturais, levando o usuário do ambiente a uma perda na sensação da imersão (BUCHMANN, 2004). Dessa forma, a investigação de metodologias para interação com estes ambientes tem se mostrado como um passo evolutivo na busca por interfaces mais próximas dos usuários.

Recentemente, tem-se observado o crescimento de Ambientes Virtuais Colaborativos (AVCs), devido ao seu elevado potencial de aplicação (KIRNER e TORI, 2004). Eles são caracterizados por cenários tridimensionais com alto índice de interação onde usuários, dispersos geograficamente, podem cooperar e compartilhar dados e informações em tempo real.

1.2 Motivação

A *computação ubíqua*, *ubiquitous computing* ou apenas *ubicomp*, significa a presença dos computadores em todos os lugares, de maneira que, embora muitos destes equipamentos estejam disponíveis no ambiente físico, sejam praticamente invisíveis para seus usuários (ART e SELKER, 1999). Isto é, os computadores e equipamentos não se limitam a desktops e dispositivos portáteis, mas a dispositivos que estejam embutidos em todo ambiente real.

O objetivo maior da computação ubíqua é tornar a interação com os computadores tão simples, comum e corriqueira que as pessoas nem percebam sua presença, mesmo quando os utilizam. Isto aconteceu com outras tecnologias, como a escrita e a eletricidade, absolutamente incorporadas em nossas vidas (JUNIOR, 2011).

Essa integração de computadores ao ambiente irá permitir que as pessoas interajam com tais sistemas de modo mais natural do que hoje, ou seja, de maneira intuitiva. Então, o termo computação ubíqua se refere aos métodos de melhoria do uso do computador, tornando muitos destes disponíveis no ambiente físico, mas de maneira que sejam efetivamente invisíveis para o usuário (WEISER, 1993). Além disso, deve possibilitar que os dispositivos detectem alterações em seus ambientes, adaptando-se automaticamente e atuando conforme tais mudanças, considerando também as preferências ou necessidades dos usuários.

Tradicionalmente, a RV exigia que os seres humanos se adaptassem às suas possibilidades e limitações. Entretanto, as novas tecnologias de geração de imagens 3D no cinema, televisão e, em breve, até em tablets, associados com interfaces extraordinariamente simples, como as presentes nos consoles Nintendo Wii® e Microsoft XBox with Kinect®, mostram que a RV também caminha para integração plena com o mundo real (JUNIOR, 2011).

Assim, a computação ubíqua demanda equipamentos acessíveis por meio de interfaces multimodais, cuja tecnologia deve se adaptar aos requisitos humanos de usabilidade e ergonomia. As interfaces multimodais são aquelas que podem utilizar para entrada de dados, incluindo aqueles multimídia, um conjunto de diferentes modalidades, tais como escrita à mão, gestos, voz ou mesmo *eye-tracking*, além de janelas convencionais, teclado e mouse (OVIATT, 2003).

Dessa forma, ao se combinar os ambientes de RV com interfaces multimodais, obtém-se uma forma de interação natural do usuário com o ambiente virtual. As aplicações de Realidade Virtual e Aumentada podem ser vistas sob um aspecto bastante amplo, variando de uma única pessoa, com um único computador até muitos usuários, usando um sistema distribuído (RINALDI et al., 2006).

As pesquisas sobre sistemas distribuídos ampliaram-se nos últimos anos, com o intuito de promover a aprendizagem cooperativa e colaborativa em ambientes computacionais, bem como o desenvolvimento de competências cognitivas, tais como a resolução de problemas (AAAS, 2001).

O uso das técnicas de sistemas distribuídos em parceria com ambientes de RV possibilitam a criação de Ambientes Virtuais Colaborativos (AVCs), definidos como ambientes virtuais que permitem a participação de diversos usuários ao mesmo tempo (RIBEIRO, 2005), o que contribui para a troca de experiências e aprendizagem de cada usuário.

O uso de AVCs torna a aplicação computacional mais flexível, ou seja, permite a interação entre usuários e com ambientes virtuais diferentes. Os AVCs em conjunto com a interação de gestos naturais ajudam a reforçar ainda mais o conceito de computação ubíqua, de forma que a interação dos dispositivos e sistemas com os usuários acontece de maneira natural.

Assim, uma integração entre as áreas de Interação Natural e de AVCs mostra-se um passo evolutivo a ser considerado a fim de tornar a interação entre usuários, dispersos remotamente, mais natural (SEIFRIED e JETTER, 2011).

Até o momento da escrita deste trabalho, não foram identificados trabalhos que reportem a referida integração.

Portanto, a motivação e contribuição desta dissertação é o desenvolvimento de técnicas para a utilização de gestos naturais em Ambientes Virtuais Colaborativos e distribuídos que permitam a manipulação de objetos virtuais por usuários dispersos remotamente. Ademais, ressalta-se a importância de pesquisas na área de AVCs juntamente com interfaces naturais, pois a troca de experiências entre usuários de maneira natural permite que os usuários utilizem as suas experiências do mundo real como: gestos, expressões faciais, movimentos corporais e da fala para se comunicar no mundo digital (VALLI, 2005).

1.3 Objetivos

Esse trabalho tem como principal objetivo investigar algoritmos e técnicas de interação natural para manipulação de objetos virtuais em ambientes colaborativos, onde os usuários estão dispersos remotamente (Figura 1).

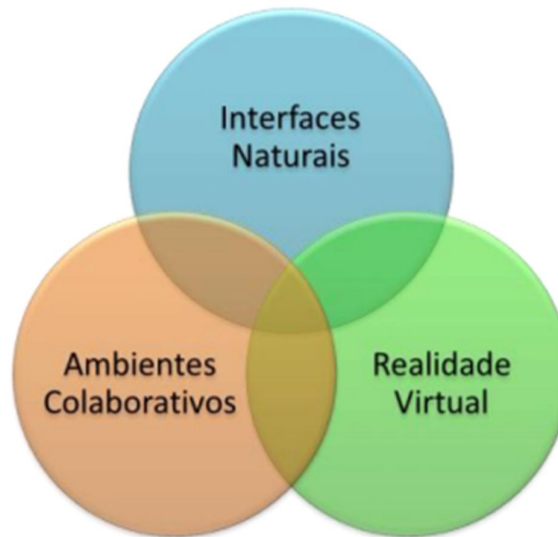


Figura 1 - Áreas do Conhecimento

A interação entre os usuários será realizada de forma síncrona e com o uso de interfaces naturais (com resultados imediatos sobre os objetos 3D que compõem o ambientes virtuais), propiciando a cada usuário a visualização e a verificação das ações próprias e coletivas.

Para a concretização deste trabalho, foram definidos os seguintes objetivos específicos:

- Investigar técnicas de criação de ambientes virtuais, focando principalmente nas formas de interação atuais;
- Estudar técnicas para reconhecimento e interpretação de gestos;
- Escolher uma plataforma que permita a interação entre gestos e objetos virtuais, para desenvolvimento do sistema computacional;
- Estudar arquiteturas de comunicação, em tempo real, para ambientes virtuais;

- Desenvolver um sistema baseado em Ambientes Virtuais Colaborativos com suporte a interação natural, entre os vários usuários dispersos remotamente, para manipulação de objetos 3D;
- Validar o sistema computacional junto a usuários, de forma a obter um *feedback* que possibilite melhorias e um aperfeiçoamento do sistema.

Esta dissertação está estruturada em sete capítulos. No Capítulo 1 foram apresentadas as motivações e organização do trabalho, bem como os objetivos e metas estipuladas. No Capítulo 2 será realizada uma breve fundamentação teórica acerca dos ambientes virtuais colaborativos, interfaces naturais, comunicação e compatibilidade entre dispositivos, além de conceitos sobre Realidades Virtual e Aumentada. Serão expostos, no Capítulo 3, os trabalhos correlatos a este que apresentam relevante contribuição acadêmica. O Capítulo 4 apresenta a arquitetura para o desenvolvimento do sistema computacional proposto.

Os detalhes da implementação do sistema computacional serão tratados no Capítulo 5, que será dividido em três áreas principais, a saber:

- Interfaces Naturais;
- Camada de comunicação de dados;
- Interação entre usuários e objetos 3D.

No Capítulo 6 há uma discussão acerca dos resultados obtidos em cada etapa do desenvolvimento. Por fim, o Capítulo 7 apresenta as conclusões deste trabalho, considerações finais e trabalhos futuros planejados.

Capítulo 2

Fundamentos

2.1 Ambientes Virtuais Colaborativos

Entende-se por trabalho colaborativo aquele em que várias pessoas (local ou remotamente distribuídas) cooperam para a realização de uma mesma tarefa de forma síncrona ou assíncrona (PINHO, 2002). Esse tipo de atividade tem uma grande importância e reflete a evolução das organizações. Diversos trabalhos têm sido realizados na área que recebeu o nome de *Suporte por Computador ao Trabalho Cooperativo* do inglês *Computer Supported Cooperative Work (CSCW)* (POLTROCK e GRUDIN, 1999).

Uma das principais características dos ambientes de CSCW é, dependendo das tarefas que serão realizadas, a maneira como os usuários podem interagir com o ambiente e entre si. Essa característica deve ser comum a todos os usuários.

Os Ambientes Virtuais Colaborativos (AVC) têm uma forte relação com os ambientes tradicionais de CSCW, pois permitem uma troca rápida e suave entre atividades individuais e colaborativas. Isto ocorre porque eles provêm uma grande facilidade de navegação, de comunicação e de mudança na forma de representação dos dados (PINHO, 2002).

Além de possibilitar a troca de atividades e experiências, os ambientes virtuais proporcionam uma noção de espaço tridimensional, que permitem, de maneira mais natural, a manipulação de objetos e as atividades entre os usuários de um AVC.

2.2.1 Arquiteturas Existentes para AVCs

Existem diferentes tipos de AVCs descritos na literatura. Cada um deles aborda as dificuldades e problemas no desenvolvimento do projeto de acordo com uma visão própria, recursos disponíveis, nível de interatividade desejado (dependendo da proposta de desenvolvimento), entre outros.

Segundo (MACEDONIA e ZYDA, 1997) e (CAPIN e THALMANN, 1999) os três principais pilares de um projeto de AVC são:

- O tipo de protocolo de comunicação;
- A política de envio de mensagens entre usuários;
- O modelo de distribuição dos dados

Nas próximas seções, estes aspectos serão discutidos com maiores detalhes.

✓ Tipo de protocolo de comunicação em AVC

A escolha do tipo de protocolo a ser usada em um AVC depende do tipo de mensagem a ser transmitida. Os dois tipos de protocolos mais utilizados são: o TCP/IP¹ e o UDP². As mensagens do tipo “atualização de estado” que informam, por exemplo, a posição atual de um objeto, não precisam ser transmitidas com protocolos confiáveis como o TCP/IP, pois nesse tipo de mensagem cada novo dado irá sobrepor os dados anteriores, uma vez que a única informação relevante é o estado *atual* do objeto (PINHO, 2002).

No caso das mensagens do tipo de “atualização de estado”, é possível a utilização do protocolo UDP, mas a confiabilidade de entrega dos pacotes de dados não é garantida. No entanto, esse é mais rápido que o protocolo TCP/IP. Já nos casos em que há necessidade de um protocolo confiável, como o envio de comandos, (quando se exige uma resposta a cada mensagem enviada ou que os comandos sejam executados na ordem de recebimento), é recomendável o uso do protocolo TCP/IP.

¹ *Transmission Control Protocol* = Protocolo de transmissão de dados confiável.

² *User Datagram Protocol* = Protocolo de transmissão de dados não confiável e rápido.

✓ Envio de mensagens

Para troca de mensagens entre os usuários e consequentemente entre os AVCs, existem três alternativas: *unicast*, *broadcast* e *multicast* (Figura 2). Na comunicação do tipo *unicast* cada cliente envia informação diretamente para outro cliente. Normalmente essa comunicação se torna inviável no caso de muitos usuários, pois a largura de banda utilizada aumenta drasticamente, tornando a transmissão lenta (BENFORD, GREENHALGH, *et al.*, 2001).

Na comunicação do tipo *broadcast* a mensagem sai de um usuário e é replicada a todos os nodos da rede. Por questões de segurança, a maioria dos roteadores da internet bloqueia este tipo de comunicação, limitando seu uso à redes locais (PINHO, 2002).

A comunicação do tipo *multicast* é mais indicada para os AVCs, pois é similar à comunicação *unicast*, com a vantagem que a mensagem é enviada a um grupo específico, utilizando uma largura de banda eficiente (BENFORD, GREENHALGH, *et al.*, 2001). Os usuários cadastrados no grupo passam a fazer parte de um endereço *multicast* quando algum usuário envia uma mensagem para este endereço e todos os usuários do grupo recebem uma cópia desta mensagem.

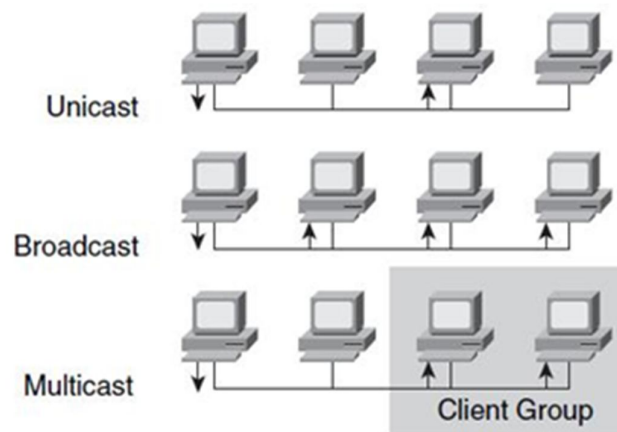


Figura 2 - Unicast, Broadcast e Multicast (CERTIFICATIONS, 2012)

✓ Distribuição dos dados

O modelo de “distribuição dos dados” faz referência à quantidade de informações do AVC que ficam armazenadas em cada nodo do participante da colaboração. Uma maneira de distribuição dos dados é através do modelo de “replicação homogênea” de maneira que cada nodo possui uma cópia de toda a base de dados que descreve o ambiente virtual.

Desta forma, sempre que ocorre alguma alteração por algum participante do AVC, ela deve ser transmitida a todos os demais nodos da aplicação. A replicação homogênea tem a vantagem, de maneira geral, que as mensagens são de tamanho reduzido, pois representam apenas atualizações no estado dos objetos (PINHO, 2002).

Nos modelos baseados em “bancos de dados” existem duas alternativas: os de “bancos de dados centralizados” e os de “bancos de dados distribuídos”. No modelo de “bancos de dados centralizados” existe um nodo central que recebe as requisições de atualização dos usuários e executa, evitando redundâncias nos AVC.

Já no modelo “banco de dados distribuídos”, os objetos são replicados nos nodos de acordo com a necessidade de cada colaborador, ou seja, um objeto é replicado apenas se o usuário requerir. Esse modelo pode ser chamado também de “replicação por demanda”, e seu desempenho é determinado pela necessidade de atualização dos usuários.

✓ Classificação para o sistema

Para o sistema proposto, utiliza-se um protocolo de comunicação do tipo *UDP*, pois este possui a vantagem de ser mais rápido que o protocolo *TCP*. Contudo, apresenta a desvantagem de não ter a garantia de entrega do pacote, como os dados com os valores das novas posições e escala, são atualizadas constantemente a perda de algum pacote não prejudica a posição do objeto 3D, pois logo o sistema receberá um novo pacote com novos dados.

O envio de mensagens através dos usuários será do tipo *multicast*, pois existirá apenas um grupo de usuários para receber as atualizações dos dados. O modelo de distribuição dos dados será do tipo “replicação homogênea”, pois cada usuário possui seus objetos 3D, e as mensagens serão reduzidas contendo apenas os valores de posição e escala.

2.2.2 Realidade Virtual

O termo Realidade Virtual (RV) é creditado a Jaron Lanier que, no início dos anos 80, o criou para diferenciar as simulações tradicionais feitas por computador das simulações envolvendo múltiplos usuários em um ambiente compartilhado (ARAÚJO, 1996).

O termo Realidade Virtual é bastante abrangente. Acadêmicos, desenvolvedores de *software* e pesquisadores tendem a tratá-lo com base em suas próprias experiências, gerando diversas definições na literatura. Segundo (BURDEA e COIFFET, 1994) a RV é uma técnica avançada de interface que permite ao usuário realizar imersão, navegação e interação em um ambiente sintético tridimensional gerado por computador, utilizando canais multi-sensoriais (dispositivos que auxiliam na interação do usuário com o ambiente virtual). Esses canais ajudam a classificar a RV em imersiva e não-imersiva, como será especificado mais adiante neste trabalho.

A interação entre o usuário e o ambiente virtual é um dos aspectos mais importantes da interface e está relacionado com a capacidade que o computador possui de detectar ações do usuário e reagir instantaneamente, modificando os aspectos da aplicação (TORI e KIRNER, 2006).

Essas interações podem ser classificadas como a visualização, exploração, manipulação e acionamento ou alteração dos objetos virtuais, utilizando os movimentos tridimensionais de translação ou rotação.

✓ Aplicações em Realidade Virtual

A todo momento surgem novas aplicações nas mais variadas áreas do conhecimento e de maneira bastante diversificada, em função da demanda e da capacidade criativa das pessoas. Em muitos casos, a RV vem revolucionando a forma de interação das pessoas com sistemas complexos, propiciando melhor desempenho e reduzindo custos (NETTO, MACHADO e OLIVEIRA, 2002).

✓ Medicina

A Realidade Virtual aplicada à cirurgia engloba várias situações, como: planejamento de um procedimento, treinamento de técnicas e a assistência para a sua realização. Um grande número de pessoas podem ser beneficiadas pela RV, uma vez que os profissionais que os atenderão estarão mais bem preparados. (KIRNER, TORI e SISCOUTO, 2006). A Figura 3 apresenta um exemplo disso.

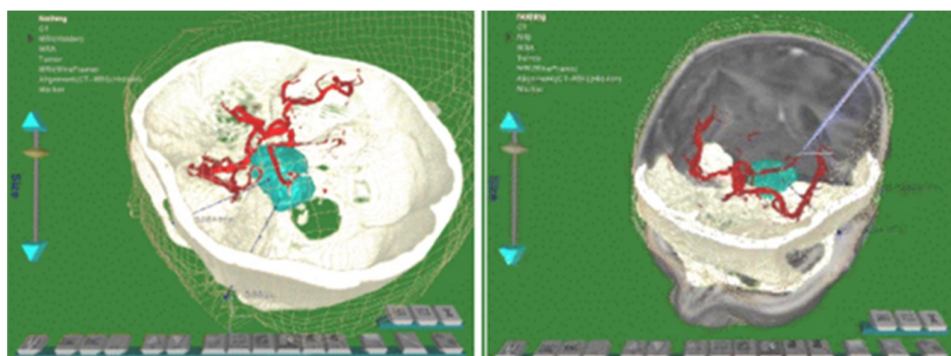


Figura 3 - Ambiente de planejamento, fusão dos dados de ressonância magnética e tomografia do cérebro de um paciente (GUAN, 1998)

Os treinamentos em ambientes virtuais tem como objetivo o desenvolvimento de habilidades práticas. A implementação da RV na prática, é realizada através de simuladores, com a finalidade de preparar os profissionais para situações reais, tal como pode ser visto através da Figura 4 que mostra um sistema de treinamento de sutura.



Figura 4 - Sistema para treinamento de sutura (WEBSTER, 2001).

✓ Engenharia

Na área de engenharia são desenvolvidos vários simuladores e protótipos em Realidade Virtual com a finalidade de realizar treinamentos de funcionários e supervisionar sistemas automáticos.

De acordo com (SILVA, LAMOUNIER e CARDOSO, 2011) um exemplo de treinamento facilitado pelo uso de RV é o de operador de subestação. Sem o uso da RV esse operador tem que se deslocar geograficamente, pois poucos locais oferecem esse treinamento. Com a RV é possível reduzir custos de deslocamentos, simular situações de riscos e permitir o treinamento de várias pessoas simultaneamente.

Um exemplo de utilização da RV em engenharia é através do *MineInside*. Segundo (COELHO, COELHO e CARDOSO, 2011) O *MineInside* consiste em um sistema de gestão de operação em mineração que utiliza RV e dados em tempo real para facilitar o controle e a tomada de decisões. Esse sistema pode ser visto através da Figura 5.

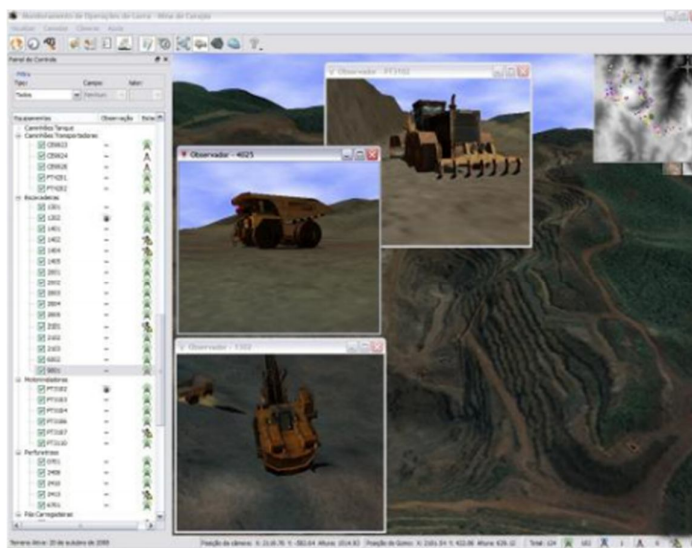


Figura 5 - Software MineInside (COELHO, COELHO e CARDOSO, 2011).

✓ Educação

O uso de RV pode contribuir muito para o aprendizado, pois é possível criar ambientes impossíveis de se visitar e interagir de forma convencional como visitar Marte, viajar dentro do corpo humano ou simular experiências de Física em um laboratório virtual (SILVA, 2008). A Figura 6 apresenta um exemplo do uso da RV na educação: criação de um laboratório virtual.

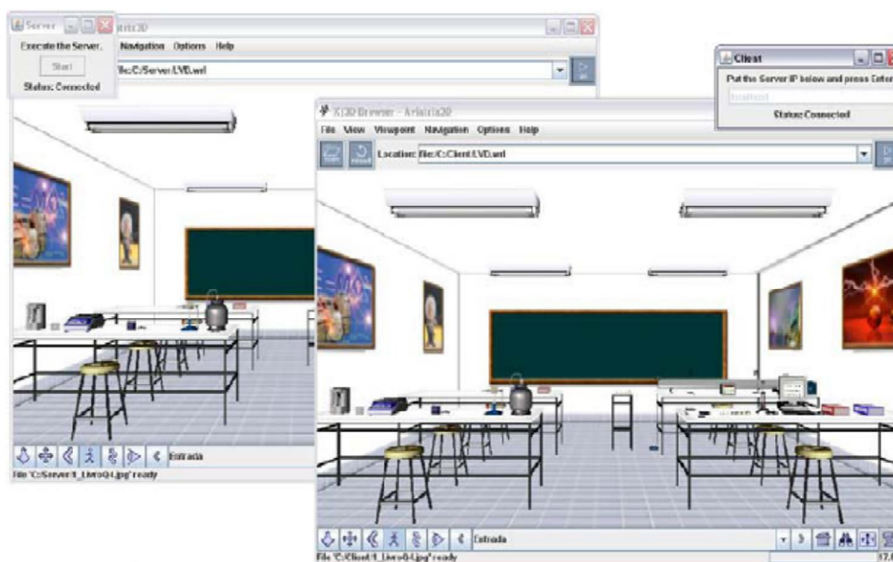


Figura 6 - Laboratório Virtual (SILVA, 2008).

✓ Tipos de Sistemas de Realidade Virtual

Existem dois tipos de sistemas de Realidade Virtual que podem ser classificados, de acordo com o senso de presença do usuário, em imersiva e não-imersiva.

A Realidade Virtual é imersiva quando o usuário tem uma sensação de presença dentro do mundo virtual, e isto é conseguido através de dispositivos multissensoriais como capacete, caverna, entre outros dispositivos (TORI e KIRNER, 2006). Dessa maneira, com o uso desses dispositivos, o usuário se desliga do mundo real e fixa apenas no mundo virtual.

A Realidade Virtual não-imersiva pode conter dispositivos multissensoriais como o uso de óculos, estereoscópio e som espacial. Porém, neste caso, o usuário ainda tem a sensação de estar no mundo real, pois ao mudar a direção dos óculos o mundo virtual desaparece.

A Realidade Virtual imersiva apresenta aplicações mais realistas do que a realidade virtual não-imersiva, porém a Realidade Virtual não-imersiva é mais utilizada por ser mais barata e mais simples (KIRNER, TORI e SISCOUTO, 2006).

Apesar de usar dispositivos multissensoriais, as técnicas usadas neste trabalho se enquadram no tipo de Realidade Virtual não-imersiva, pois o usuário não tem a sensação de estar dentro do ambiente virtual, mas apenas a sensação de poder manipular objetos tridimensionais com as mãos.

✓ **Manipulação de Objetos 3D**

A manipulação de objetos 3D pode ser definida como a mudança de algum parâmetro ou estado de um objeto, por exemplo: rotação, mudança de posição, tamanho e outros parâmetros geométricos. Tais operações podem ser realizadas de três maneiras: manipulação direta, apontamento por raios ou interação discreta (BOWMAN, KRUIJFF, *et al.*, 2004).

Na manipulação direta, o usuário opera o objeto virtual com as mãos, da mesma maneira que faria no mundo real. Essa técnica se baseia no rastreamento da posição da mão do usuário, que é refletido no objeto virtual desejado. Outra maneira de manipular objetos é através do apontamento de raios, assim a seleção de um objeto é confirmada caso o raio cruze com ele. Já na seleção discreta, a base para selecionar um objeto é por meio de uma especificação preliminar que possa identificar o mesmo, como por exemplo, o nome. Essa identificação pode ser realizada também por seleção em um *menu*, reconhecimento de voz ou entrada de texto.

Nesse trabalho, a seleção será direta, por meio de gestos realizados com as mãos, pois o foco é a manipulação de objetos 3D em AVCs, através de interfaces naturais.

2.2.3 Interfaces

As tecnologias de interface entre usuários e computadores vêm evoluindo constantemente. Na área de informática, elas passaram dos cartões perfurados e chaves mecânicas às interfaces de textos (controladas por teclado) além de interfaces gráficas que utilizam mouse e outros dispositivos.

Atualmente as interfaces homem máquina desenvolveram de forma a utilizar os sentidos do homem para dar a ele a sensação de realidade no mundo virtual (COLLET e SILVA, 2011). Essas interfaces utilizam os sentidos do usuário, por exemplo: as interfaces hápticas utilizam sensores que detectam e replicam as forças do tato e as interfaces gestuais utilizam dispositivos para detectar e interpretar gestos.

O foco desse trabalho é nas interfaces gestuais, de forma que o usuário irá utilizar as mãos para realizar gestos que serão replicados no ambiente virtual.

Interfaces Gestuais

As interfaces gestuais utilizam dispositivos para interpretar gestos e movimentos do usuário. Como exemplo, o dispositivo *dataglove* (luva de dados) permite ao sistema de RV reconhecer os movimentos da mão do usuário (STURMAN e ZELTZER, 1994). Normalmente, para determinar os movimentos dos dedos são utilizados sensores mecânicos ou de fibra ótica. A Figura 7 ilustra o uso do dispositivo “luva de dados”.



Figura 7 - Luva de dados (The Free Dictionary, 2012).

O *Head-Mounted Display* (HMD) é um dos dispositivos de interface para RV mais populares por se tratar do dispositivo não-convencional de saída de dados que mais isola o usuário do mundo real. Esse dispositivo pode ser usado para exibir imagens estereoscópicas, apresentando os respectivos pontos de vistas de cada olho para cada tela, contribuindo para o efeito imersivo (BOTEGA e CUVINEL, 2009). A Figura 8 mostra um exemplo de utilização do HMD criado pela NASA.



Figura 8 - Primeira geração de HMDs da Nasa (KALAWSKY, 1993).

Atualmente, é grande o interesse por pesquisas na área de interfaces gestuais. A interação do ser humano com esses dispositivos, de maneira mais natural possível, é um dos principais objetivos das pesquisas. Outro importante tema tratado pelas pesquisas se refere à dequação do dispositivo ao usuário e não o contrário, conforme será descrito na próxima seção.

2.2.4 Interfaces Naturais - NUI

Com o crescimento do uso de dispositivos sensíveis ao toque, particularmente em celulares e *tablets*, houve um desejo de interação dos mesmos através de gestos e movimentos naturais do nosso corpo (GNECCO e DIAS, 2012). Essa nova forma de interação ganhou o nome de “interfaces naturais” (do inglês *Natural User Interfaces* ou *NUIs*). As “interfaces naturais” têm a vantagem de utilizar conhecimentos prévios dos usuários, exigindo assim o mínimo de treinamento. A

Figura 9 apresenta uma ilustração da evolução da interfase humano-computador, do uso do teclado até as mãos.

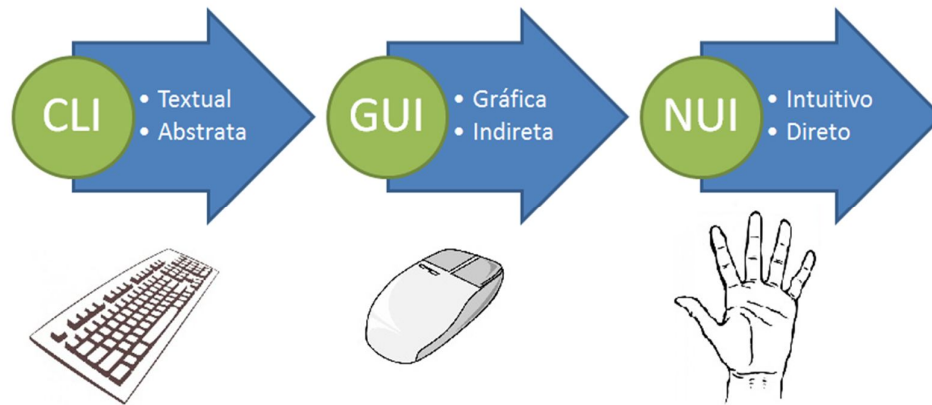


Figura 9 - Evolução da Interface humano-computador (baseado em (REYES, 2009))

As pesquisas nessa área podem ser divididas em duas vertentes:

- ✓ Interfaces vestíveis (*wearable*)
- ✓ Interação à distância (*touchless*)

✓ Dispositivos de *wearable*

As Interfaces vestíveis são as que utilizam dispositivos como luvas, roupas inteiras ou outros dispositivos que contenham sensores ou marcadores para ajudarem na detecção e rastreamento. Apesar de bons resultados, as interfaces vestíveis possuem uma desvantagem: são desconfortáveis.

✓ Técnica *Motion Capture*

A técnica *Motion Capture* consiste em rastrear acessórios detectáveis por câmeras comuns ou infravermelhas. Neste caso, ao invés de sensores presos ao corpo do usuário, sinais visuais (como luzes ou marcadores presos ao corpo), são fornecidos e capturados, facilitando o rastreamento do corpo (FIGUEIREDO e SOUZA, 2012).

O dispositivo GESTUS (Figura 10) ilustra o uso de uma luva com emissores infravermelhos (LEDs) que são detectados pela sua cor. Cada conjunto de *LEDs* que acende forma um gesto pré-definido, a ser detectado pelo computador.



Figura 10 - Dispositivo GESTUS (COLLET e SILVA, 2011)

Outro dispositivo que utiliza a técnica *Motion Capture* é o MOVE® (Figura 11) da empresa Sony. Este dispositivo funciona através de um bastão com um marcador específico em forma de uma esfera, que é monitorado por uma câmera. O MOVE® ainda conta com um controle adicional na mão esquerda para combinar comandos com o controle que contém a esfera.

©2010 Sony Computer Entertainment Inc. All rights reserved.
Design and specifications are subject to change without notice.



Figura 11 - Dispositivo MOVE® (FORUMPC, 2012)

O dispositivo gestual *WiiMote*® (Figura 12) é do tipo *Motion Capture*, realiza a interação (por intermédio de um controle que possui vários acelerômetros) junto a mão do usuário e uma barra de sensores que detecta o movimento da mesma, realizando gestos pré-definidos.



Figura 12 - Dispositivo Nintendo Wii® (Nintendo, 2012)

✓ Dispositivos de *touchless*

A outra vertente é interação à distância, que utiliza detecção por dispositivos externos, em geral câmeras, sem a necessidade de contato físico entre o usuário e nenhum tipo de dispositivo (GNECCO e DIAS, 2012).

A tecnologia *touchless* recebeu, recentemente, grandes investimentos, desenvolvendo câmeras que detectam informações de distância (*Z-buffer*) através de um sensor infra-vermelho e não apenas pela cor (RGB), como fazem as câmeras comuns. Os dispositivos do tipo *touchless* detectam movimentos à distância e não dependem de nenhum tipo de aparato preso ao corpo do usuário.

No ano de 2010, surgiu um dispositivo chamado Kinect® (Figura 13), desenvolvido pela *Microsoft*, de custo acessível ao usuário doméstico. O Kinect

permite a interação sem toque por meio de gestos e comandos de voz, o que ajudou a colaborar com a popularização das interfaces *touchless*. Foram vendidas mais de 8 milhões de unidades desse dispositivo em 60 dias após o seu lançamento por um preço de US\$150.00. Assim, entrou para os recordes Guinness como o “dispositivo eletrônico para consumidores vendido mais rapidamente” (GLOBO, 2012). A Figura 13 mostra uma ilustração do dispositivo Kinect.



Figura 13 - Sensor Kinect® (baseado em MICROSOFT 2011)

Outro dispositivo semelhante ao Kinect é o ASUS Xtion Pro Live® (Figura 14). Lançado em 2011, o Xtion Pro Live® também possui sensor infravermelho e microfones para reconhecer gestos e receber comandos de voz.



Figura 14 - Dispositivo Xtion PRO LIVE® (ASUS, 2011)

A tecnologia do sensor infra-vermelho (Figura 15) tem a vantagem de não depender da intensidade da luz do ambiente e conseguir os valores de profundidade em cada ponto, não havendo necessidade de um ambiente controlado nesse caso.

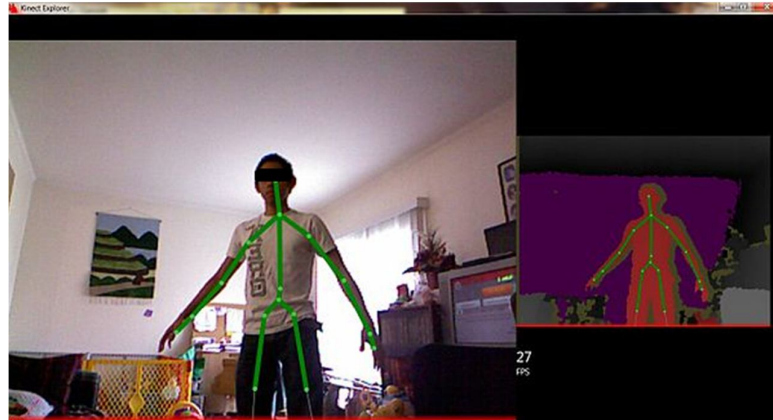


Figura 15 - Sensor Infra-vermelho

Para esse trabalho será utilizado uma interface *touchless* para que o usuário tenha a sensação de liberdade para a realização dos gestos. Dentre os dispositivos citados optou-se por utilizar o Microsoft Kinect, pelo fato de já possuir um *framework* com documentação detalhada e compatibilidade com as linguagens da plataforma Microsoft.

Capítulo 3

Trabalhos Correlatos

3.1 Introdução

Serão apresentados a seguir trabalhos relacionados com interfaces naturais, ambientes colaborativos e Realidade Virtual. Estes trabalhos foram desenvolvidos em áreas e com tecnologias diferentes.

3.2 Flexible Action and Articulated Skeleton Toolkit (FAAST)

O FAAST (EVAN e BELINDA, 2011) foi desenvolvido em 2011 na *University of Southern California* com o objetivo de ser um *framework* para aplicações de Realidade Virtual utilizando interfaces naturais. O FAAST é um mapeador de movimentos do corpo, ou seja, ele captura os movimentos do usuário e transmite para um servidor de dados. Pode ser utilizado para aplicações de computador, ambientes de RV e games.

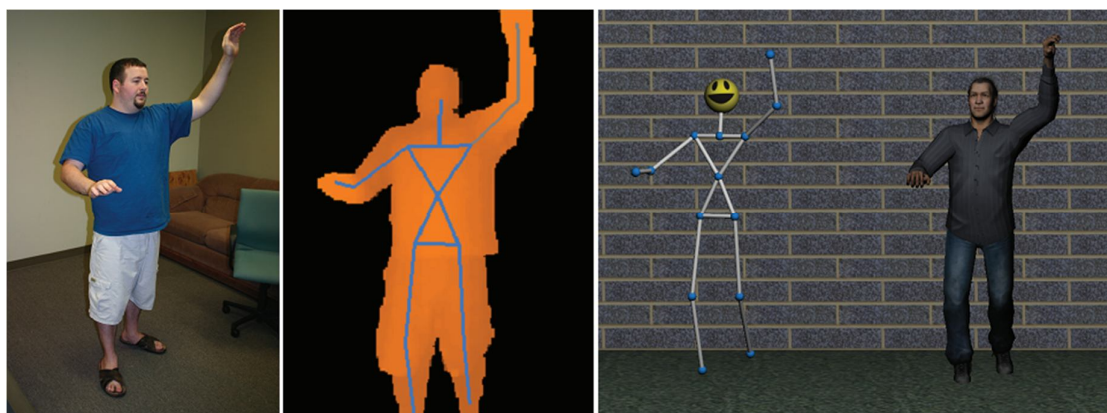


Figura 16 - Animando um avatar em tempo real com o FAAST (EVAN e BELINDA, 2011).

O FAAST utiliza o sensor Microsoft Kinect e o *middleware* OpenNI para permitir o uso de interfaces naturais em jogos que não foram explicitamente desenvolvidos para uso de interfaces naturais (EVAN e BELINDA, 2011).

Após o usuário fazer uma breve calibração do seu corpo, o FAAST possibilita o *streaming* da posição das articulações via rede através de um servidor VRPN (*Virtual Reality Peripheral Network*), é adequada para aplicações mais elaboradas que tenham compatibilidade com esse protocolo. Através dele, é possível ainda configurar comandos de teclado e movimentos do cursor do mouse, funcionando muito bem em jogos com interfaces 2D (PAULA, 2011).

Para a interação e movimentos em objetos 3D, o FAAST apresenta limitações, pois não possui interação natural através de gestos, apenas são repassadas as posições das juntas do usuário. Dessa maneira, o sistema não

permitirá que usuário utilize suas experiências anteriores, exigindo um treinamento prévio para a manipulação de objetos 3D.

Segundo (EVAN e BELINDA, 2011) uma das propostas de trabalho futuro usando o FFAST é a criação de gestos associados a eventos e também a criação de uma arquitetura de comunicação do tipo cliente servidor, para permitir a interação de vários usuários distantes remotamente.

3.3 Kinoogle

O Kinoogle (Software para interação natural do Google *Earth* utilizando o Kinect) foi desenvolvido por um grupo de pesquisadores da Universidade de Plymouth, com a finalidade de criar uma interface natural para o *Google Earth* e para o *Google Street View* (BOULOS, BLANCHARD, *et al.*, 2011) conforme Figura 18.

Inicialmente, o Kinoogle utilizou o FFAST³ para obter a posição das mãos pelo dispositivo Kinect, com a posição das mãos o Kinoogle reconhece alguns gestos dos usuários para replicar no *Google Earth* ou *Google Street View*, como apresentado na Figura 17. O Kinnoogle optou pelo *key bindings* de forma que a posição das mãos substituem o teclado e o mouse. Ou seja, por meio do dispositivo Kinect® são transmitidas as informações sobre a posição das mãos que, por sua vez, substituem algumas teclas e botões dos dispositivos convencionais para a navegação no *Google Earth*.

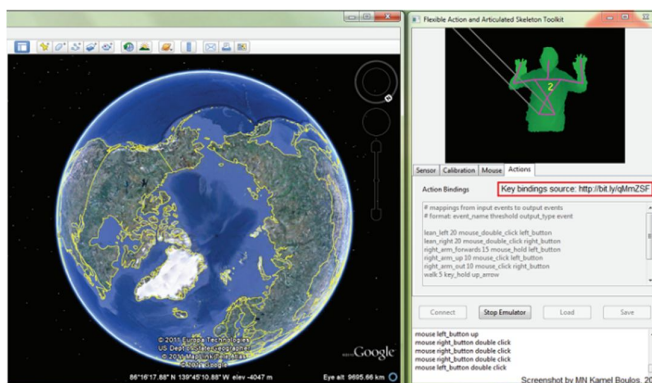


Figura 17 - FFAST + Google Earth (BOULOS, BLANCHARD, *et al.*, 2011)

³ Tratado no item 3.2

Baseado no trabalho de uma empresa húngara que desenvolveu uma solução com interação natural do *Bing Maps* com o Kinect® sem o FFAST (INFOSTRAT, 2012), o Kinoogle desenvolveu sua aplicação para interação com o *Google Earth* e *Google Street View*. A Figura 19 apresenta esses gestos.

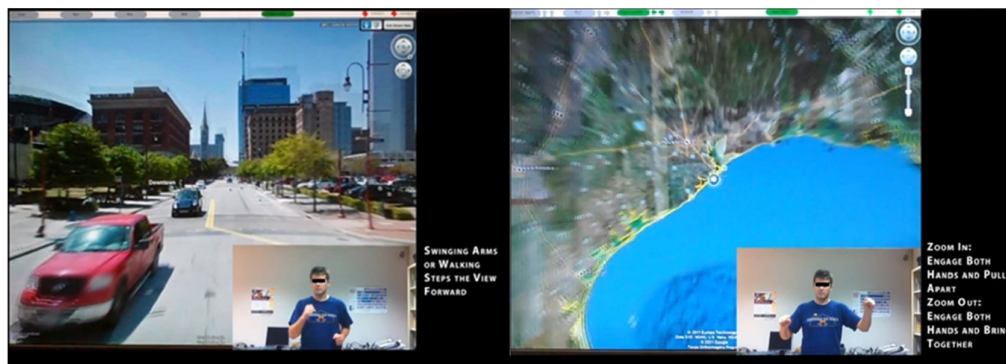


Figura 18 – Kinoogle Google Street View e Earth (BOULOS, BLANCHARD, et al., 2011).

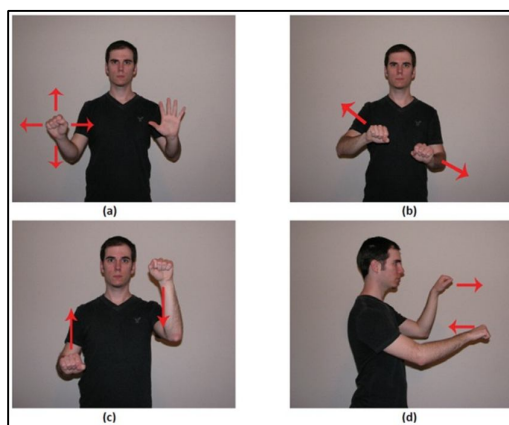


Figura 19 – (a) Translação, (b) Escala, (c) e (d) Rotação (BOULOS, BLANCHARD, et al., 2011).

O Kinoogle permite também a navegação no *Google Street View* através de gestos naturais pré-estabelecido tais como os apresentados pela Figura 20.



Figura 20 - (a) Caminhar em frente (b) Virar para os lados (BOULOS, BLANCHARD, et al., 2011)

O Kinoogle apresenta algumas restrições quanto à execução dos gestos, por exemplo, existe um *menu* que apresenta o gesto a ser realizado em uma barra de progresso e demora algum tempo para ser identificado e realizado. Esse *menu*, apesar de auxiliar o usuário quanto aos gestos, acaba por diminuir a imersão do usuário.

Como trabalho futuro os autores (BOULOS, BLANCHARD, *et al.*, 2011), sugerem o uso do Kinoogle em ambientes de colaboração como, por exemplo, em teleconferência com vários usuários, os mesmos poderiam se encontrar virtualmente em um local estipulado do mapa e obter ou criar objetos 3D para esse local através do dispositivo Kinect®.

3.4 GTAVCS – Arthron

O “Arthron” (LAVID, 2012) é uma ferramenta de gerenciamento remoto para a captura e distribuição de múltiplos fluxos simultâneos de mídia com a finalidade de prover suporte a cenários de videocolaboração. Essa ferramenta foi desenvolvida pelo GTAVCA (Grupo de Trabalho em Vídeo Colaboração em Saúde) da Universidade Federal da Paraíba em João Pessoa (CASTRO, 2012).

Com o objetivo de enriquecer ainda mais a experiência do usuário, foi desenvolvido um recurso adicional por meio do qual ele poderá navegar e manipular modelos 3D de partes do corpo humano através de interação natural. Por uma questão de relevância, a ênfase será no recurso adicional de manipulação e de modelos 3D através de Interfaces Naturais.

O desenvolvimento do *software* utilizou o *framework* OpenNI e o dispositivo Kinect para rastrear as mãos do usuário. O usuário pode escolher vários modelos 3D para a visualização através de um *menu*, como mostrado na Figura 21.



Figura 21 - Seleção do Modelo 3D (CASTRO, 2012)

Após a escolha do modelo 3D, a interface permite que o usuário faça movimentos de rotação e escala. Existe um *menu*, composto por 4 círculos, sendo que em cada círculo o usuário pode escolher o tipo de movimento, mantendo o cursor por um tempo determinado, conforme Figura 22.

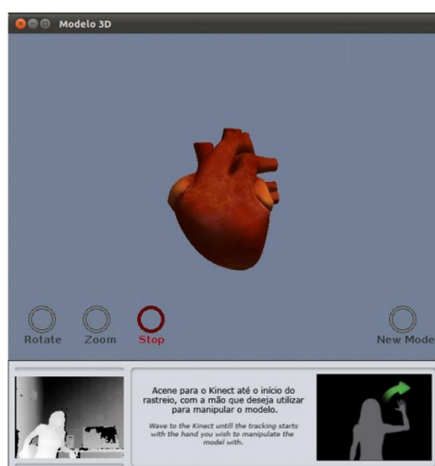


Figura 22 - Interface de manipulação (CASTRO, 2012).

O sistema apresentado necessita de um gesto inicial para que inicialize. Esse gesto é devido ao *framework* OpenNI, que exige uma calibração inicial para iniciar.

As limitações apresentadas pelo sistema é o uso de *menus* para a realização da manipulação dos objetos 3D, o que diminui a sensação de imersão, e a falta de um ambiente colaborativo, de forma a contribuir para o ensino de anatomia, um vez que todos os objetos 3D são partes do corpo humano.

Segundo (CASTRO, 2012) uma sugestão de aprimoramento do *software* seria a eliminação do *menu* com botões circulares, de maneira que o usuário pudesse realizar a rotação e a escala sem a necessidade de escolha em um *menu* para os gestos, aumentando o nível de imersão do usuário.

3.5 Um Stand Virtual Automobilístico

Os autores (CARVALHO, SOUSA e OLIVEIRA, 2012) desenvolveram um trabalho que contempla a visualização de objetos 3D e a manipulação de forma natural, o objeto 3D em questão é um carro modelado em três dimensões. O estudo de caso referente à análise de um veículo pelo usuário (comprador), sendo possível a ele manipular e personalizar o produto.

O trabalho permite a manipulação com as mãos de forma natural, através de *menus* como pode ser visto através da Figura 23.



Figura 23 - Interface de Utilização do Protótipo Desenvolvido (CARVALHO, SOUSA e OLIVEIRA, 2012)

Para acessar o *menu*, o usuário deve manter a mão posicionada por um tempo em uma das opções disponíveis para que o sistema permita a rotação, alteração da escala ou mudança de cor do objeto.

Outra forma de ter acesso ao *menu*, para escolha do tipo de manipulação, é através do uso de comandos de voz. Basta o usuário pronunciar, por exemplo: “Rotacionar”, “zoom” ou “alteração de cor”.

Para alterar a cor do objeto, o usuário (após permanecer com a mão no *menu* “alterar cor” ou usar o comando de voz) deve realizar um movimento de 20 centímetros com a mão na posição horizontal em um tempo de 2 segundos.

Para a rotação do objeto 3D, o usuário deve utilizar as duas mãos, distanciando-as na posição vertical. É possível apenas rotacionar em torno do eixo Z.

Para a escala é necessário também à utilização das duas mãos, e deve-se distanciar as duas na posição horizontal. Quanto mais distantes, maior será o aumento da escala e quanto mais próximas menor será a escala.

Os autores (CARVALHO, SOUSA e OLIVEIRA, 2012) ainda analisaram dados como eficiência, visualização dos modelos, usabilidade, nível de interatividade, controle sobre a atividade e modelo de interface.

De acordo com o grau de conhecimento, os usuários foram separados pelos avaliadores em dois grupos (de acordo com o nível de conhecimento de cada um) para aplicação de um questionário. O questionário aplicado era composto por doze questões sobre o nível de satisfação com o uso do sistema. O resultado mostrou alto grau de satisfação pelos usuários na utilização do sistema descrito.

Como trabalhos futuros, os autores pretendem desenvolver novos estudos de casos em outras áreas e estudar novas técnicas de interação.

Apesar de o trabalho não mencionar nenhuma perspectiva de interação entre usuários dispersos remotamente, esta seria uma boa opção para ajudar a venda de produtos, pois permitiria a comunicação e interação entre um vendedor e o comprador.

3.6 Fusion 4D

O *Fusion 4D* (Interface Natural e Imersiva para Manipulação de Objetos 3D) (MATSUMURA e SONNINO, 2011) tem por objetivo a implementação de uma interface gestual baseada em interação com objetos 3D a partir de interfaces naturais, de maneira a criar um ambiente imersivo e natural, para o estudo de partes do corpo humano modelados em 3D do projeto VIDA (TORI, NUNES, *et al.*, 2009).

Para a realização do objetivo proposto, o *Fusion 4D* utiliza o dispositivo Kinect® como sensor de movimentos e adicionalmente uma interface com óculos

estereoscópicos anaglifos. A Figura 24 mostra uma imagem criada através desse dispositivo.

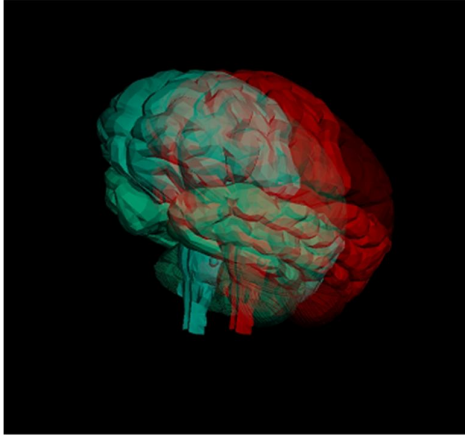


Figura 24 - Interface do Fusion 4D (MATSUMURA e SONNINO, 2011).

O Fusion 4D não possui nenhum tipo de *menu* ou botões, pois um de seus objetivos é ser o mais imersivo possível. Dessa maneira, não há necessidade de utilização de mouse ou teclado para a interação com o sistema. A escolha do modelo 3D ou os tipos de interação são realizados através de comandos de voz (Figura 25) e gestos pré-definidos.

Utilizando o efeito de estereoscopia e óculos estereoscópicos, é possível ao usuário realizar manipulações em um objeto utilizando Realidade Aumentada (RA), pois, o mesmo irá visualizar o objeto 3D em um cenário real, e poderá manipular esse objeto através de gestos, sem nenhum tipo de marcador (Figura 26).

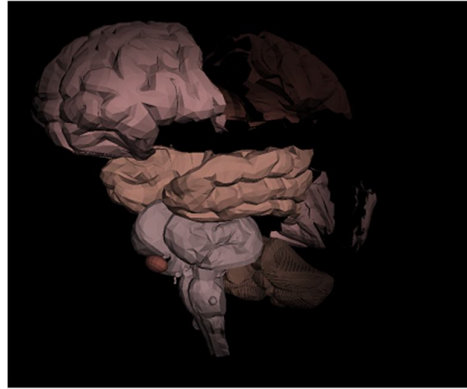


Figura 25 - Vista explodida de um modelo 3D, acionada através do comando de voz "explode (MATSUMURA e SONNINO, 2011).

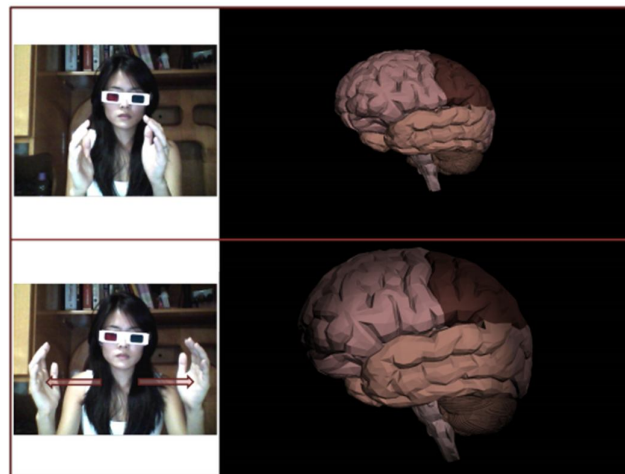


Figura 26 - Modificação da escala de um objeto 3D, utilizando estereoscopia (MATSUMURA e SONNINO, 2011).

O Fusion 4D permite a visualização de objetos 3D tanto em Realidade Virtual quanto em Realidade Aumentada, utilizando óculos estereoscópicos. Além disso, realiza as manipulações de translação, rotação e escala através de gestos e comandos de voz. Segundo os autores, o Fusion 4D atingiu com sucesso os resultados propostos, apresentando uma boa alternativa para manipulações de objetos 3D.

Como limitações e proposta de trabalhos futuros, os autores sugerem, a integração do óculos 3D para que o sistema detecte automaticamente o modo anaglifo, sem utilização de mouse, teclado ou comando de voz.

Uma contribuição para o trabalho é a colaboração com mais usuários, já que o *software* é uma ferramenta de estudo das partes do corpo humano e essa interação poderá, desta forma, enriquecer ainda mais o aprendizado.

3.7 Comparativo entre os trabalhos

O FFAST é uma alternativa para incorporar interfaces gestuais em sistemas que não as possui. Porém, existe a necessidade de um servidor VRPN compatível com a aplicação e a interpretação dos gestos também depende da aplicação. Já o Kinoogle possui uma série de gestos pré-definidos para a integração com as aplicações da Google, mas sem suporte e colaboração de outros usuários. Segundo os autores a colaboração é importante para que se possa compartilhar posições e realizar vídeo conferências entre usuários, sendo uma das propostas de trabalho futuro citada pelos autores.

O Arthron é um sistema que possui a manipulação de objetos 3D, que é uma funcionalidade importante em sistemas de Realidade Virtual. Apesar do FFAST não possuir essa funcionalidade de forma explícita, é possível realizar esse movimento em conjunto com outra aplicação. O Kinoogle não tem como foco a manipulação de objetos 3D, mas, segundo o autor (EVAN e BELINDA, 2011), uma das propostas é a implementação de um módulo para a realização da manipulação 3D.

O Fusion 4D é um trabalho semelhante ao Arthron, com o foco na manipulação de objetos 3D. No entanto, o Fusion 4D apresenta um nível maior de imersão por não utilizar *menus* para a escolha dos objetos 3D e nem para a escolha do tipo de movimento a ser realizado. Além disso, o Fusion 4D suporta comandos de voz, o que ajuda ainda mais na imersão do usuário.

A tabela 1 apresenta uma comparação entre os trabalhos correlatos apresentados

Tabela 1 – Quadro comparativo entre os trabalhos correlatos

Tópico Abordado	FFAST	Kinoogle	Arthron	Fusion 4D	Manipulação Automobilístico
Interação <i>touchless</i>	✓	✓	✓	✓	✓

RV	✗	✗	✓	✓	✓
Movimentação de objetos 3D	✗	✗	✓	✓	✓
Gestos Naturais	✗	✓	✓	✓	✓
Detecção dos gestos sem menu	✗	✗	✗	✓	✗
Colaboração entre usuários	✗	✗	✗	✗	✗

Para realizar a comparação entre os trabalhos correlatos levou-se em consideração características em comum entre a maioria deles e propostas de trabalhos futuros.

O trabalho que mais se assemelha com este é o Fusion 4D, que foca na interação e imersão do usuário para realização da manipulação de objetos 3D. Porém, o Fusion 4D e os demais trabalhos não fornecem suporte para a colaboração entre usuários, ou seja, não permitem que dois ou mais usuários distantes remotamente realizem trabalhos em colaboração.

A proposta para essa dissertação é o estudo de técnicas de interação natural em ambientes colaborativos, tendo como produto um sistema computacional para manipulação de objetos 3D por meio de gestos naturais. A contribuição dessa dissertação diante os demais trabalhos é o desenvolvimento de um ambiente de colaboração que suporte gestos naturais, com a vantagem de ser um sistema mais imersivo e que permite a troca de experiência entre os usuários.

Capítulo 4

Especificação do Sistema Proposto

4.1 Introdução

Tomando como base os estudos relacionados nos capítulos anteriores e nos fundamentos de Engenharia de *Software*, elaborou-se o projeto do sistema. Neste

capítulo são abordados os principais pontos do projeto proposto através da análise de requisitos, da apresentação de alguns diagramas comportamentais da UML (*Unified Modeling Language*) e da discussão da arquitetura implementada. No Capítulo 5 o projeto será mais detalhado, com apresentação das ferramentas escolhidas e códigos-fontes.

4.2 Especificação do Sistema

Com a finalidade de ajudar na compreensão das funcionalidades e relações entre os módulos do sistema desenvolvido, esta seção irá apresentar alguns diagramas UML, considerados importantes para o entendimento do funcionamento global do sistema desenvolvido.

4.2.1 Requisitos do Sistema

Segundo (SUMMERVILLE, 2007) os requisitos de um sistema consistem na descrição dos serviços a serem disponibilizados por este, assim como suas restrições operacionais. Dessa maneira, os requisitos podem ser resumidos como sendo as necessidades do público alvo do sistema.

Os requisitos podem ser divididos em duas categorias: funcionais e não funcionais. Essa definição é apresentada por (SUMMERVILLE, 2007) da seguinte maneira:

- **Requisitos funcionais:** Consiste na declaração dos serviços que o sistema deve fornecer ao usuário, como será seu comportamento em relação entradas de dados específicas e o comportamento em diferentes situações.
- **Requisitos não funcionais:** Contempla as restrições sobre os serviços ou funções oferecidas pelo sistema.

Tendo como base as definições descritas acima, foram classificados os requisitos funcionais e não funcionais do sistema proposto.

- Requisitos funcionais:
 - o Escolher o objeto 3D a ser manipulado;

- o Compartilhamento de movimentos e posições do objeto 3D com outros usuários dispersos remotamente em um ambiente colaborativo;
 - o Permitir o uso de gestos pré-definidos;
- Requisitos não funcionais:
 - o Inicializar automaticamente o sistema;
 - o Operar com a taxa mínima de 30 FPS (*Frames per Second*);
 - o Independência da luminosidade do ambiente de trabalho;
 - o Não necessitar de etapas de calibração e/ou treinamento.

4.2.2 Diagrama de Caso de Uso do Sistema

Para facilitar o entendimento dos requisitos do sistema foi desenvolvido o diagrama de Caso de Uso apresentado na Figura 27. Observa-se que há apenas um papel: o do usuário que age com o objeto 3D. Ele atua na escolha do objeto, manipulação do mesmo e compartilhamento da posição com outros usuários.

Para facilitar ainda mais o entendimento do Caso de Uso apresentado na Figura 27, as Tabelas 2, 3 e 4 mostram a descrição dos casos principais.

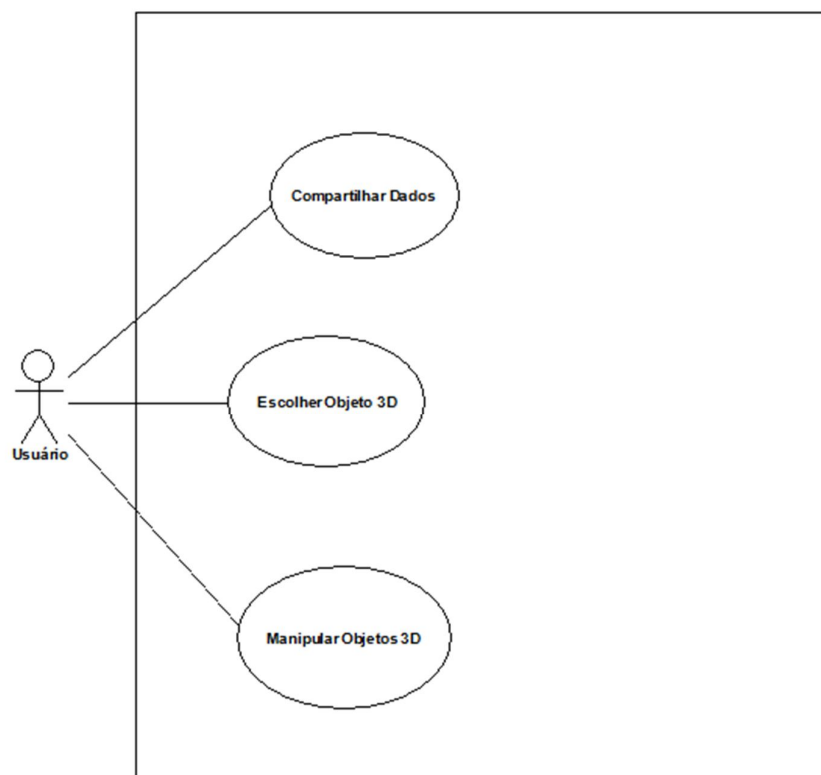


Figura 27- Diagrama de caso de uso.

Tabela 2 - Descrição do caso de uso "Compartilhar Dados".

Caso de Uso: Compartilhar Dados
Resumo: Compartilhar posições e movimentos dos objetos 3D entre usuários. Atores envolvidos: Usuário Descrição: O usuário, após selecionar um objeto 3D efetua através das mãos, gestos para manipular o objeto escolhido, as posições desse objeto são enviadas para todos os usuários conectados ao sistema, possibilitando esses usuários a compartilharem manipulações no objeto 3D. Exceções: Caso o usuário não efetue nenhum movimento, o sistema fica apto para receber dados de outros usuários. Objetivo: Criar um ambiente de colaboração entre usuários distantes remotamente.

Tabela 3 - Descrição do caso de uso "Escolher Objeto 3D".

Caso de Uso: Escolher Objeto 3D
Resumo: Escolher qual objeto 3D será manipulado pelos usuários. Atores envolvidos: Usuário Descrição: O usuário, através de um menu, tem a opção de escolher entre três diferentes tipos de objeto 3D. Exceções: Caso o usuário não escolha nenhum objeto, o sistema fica aguardando para receber dados de outros usuários. Objetivo: Criar um ambiente onde irá existir um objeto 3D para ser manipulado.

Tabela 4 - Descrição do caso de uso "Manipular Objeto 3D".

Caso de Uso: Manipular Objeto 3D
Resumo: Altera a posições dos objetos 3D através de gestos. Atores envolvidos: Usuário Descrição: O usuário, após selecionar um objeto 3D, pode efetuar a manipulação do mesmo através de gestos pré definidos, como rotação e alteração da escala. Exceções: Caso o usuário não efetue nenhum movimento, o sistema pode receber e alterar a posição do objeto 3D. Objetivo: Permitir que o usuário faça a manipulação de um objeto 3D através de gestos naturais.

A partir da definição do diagrama de caso de uso, foi possível desenvolver uma visão geral do sistema e estabelecer uma ordem de funcionamento, através dos diagramas de atividades, apresentados na próxima seção.

4.2.3 Diagrama de Atividades do Sistema

O diagrama de atividades faz parte dos que compõem a UML. O principal objetivo do diagrama de atividades é modelar o comportamento do sistema (através

da definição de caminhos lógicos que um processo pode seguir) a partir de representações gráficas.

Será apresentado o diagrama de atividades de maneira simples para que o usuário entenda o funcionamento de maneira geral. O diagrama está apresentado na Figura 29.

Primeiramente o sistema deve localizar o usuário, ou seja, deve rastrear a posição das suas mãos antes de realizar qualquer gesto.

Após a identificação do usuário e cadastro no grupo *multicast* o sistema aguarda a seleção de algum objeto 3D através do *menu*. O objeto, depois de selecionado, deve aparecer na cena com a posição inicial, caso tenha outro usuário enviando dados. Uma *thread* (Figura 28 [alt Dados]) fica recebendo os dados e atualizando a posição do objeto 3D, conforme apresentado no diagrama de sequência da Figura 28. Quando o sistema para de receber dados, o usuário tem a oportunidade de realizar gestos e o sistema tenta identifica-lo, caso seja um gesto definido, as alterações necessárias ocorrem no objeto 3D e são enviada as novas posições aos demais usuários cadastrados.

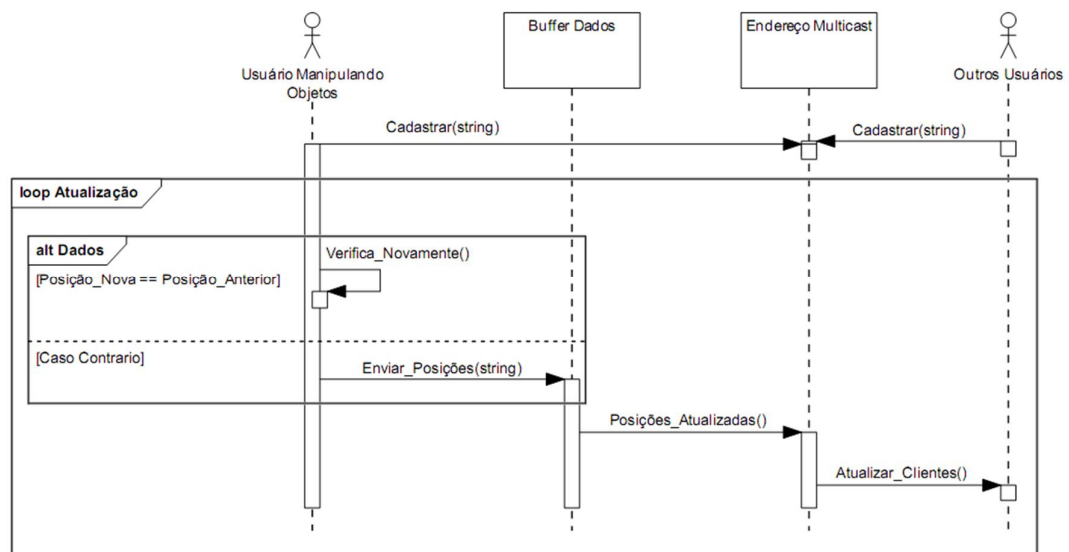


Figura 28 - Diagrama de Sequencia (Colaboração entre usuários)

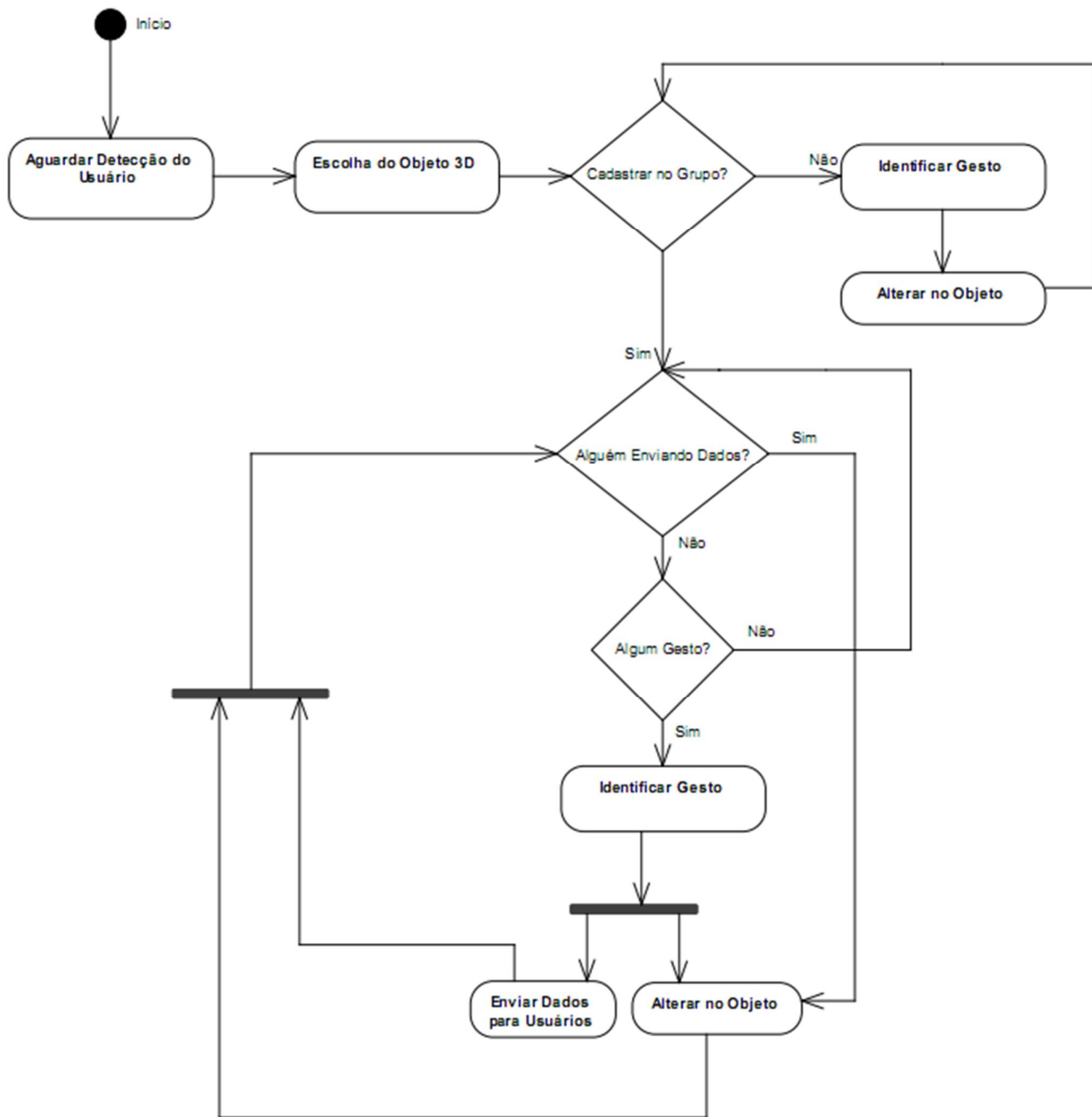


Figura 29 - Diagrama de atividade

4.2.4 Arquitetura do Sistema

Com o objetivo de apresentar o funcionamento global do sistema, a Figura 30 mostra um fluxograma da arquitetura do sistema desenvolvido. Observa-se que o fluxo de execução inicia-se com a captura da imagem do usuário, passa pelo

processamento e atuação de acordo com o conteúdo recebido e, por fim, é renderizada a imagem correspondente à nova cena.

Podemos separar o processo descrito acima em dois blocos: **Captura e detecção** e **Interação**, sendo que cada bloco possui as atividades com mais afinidade.

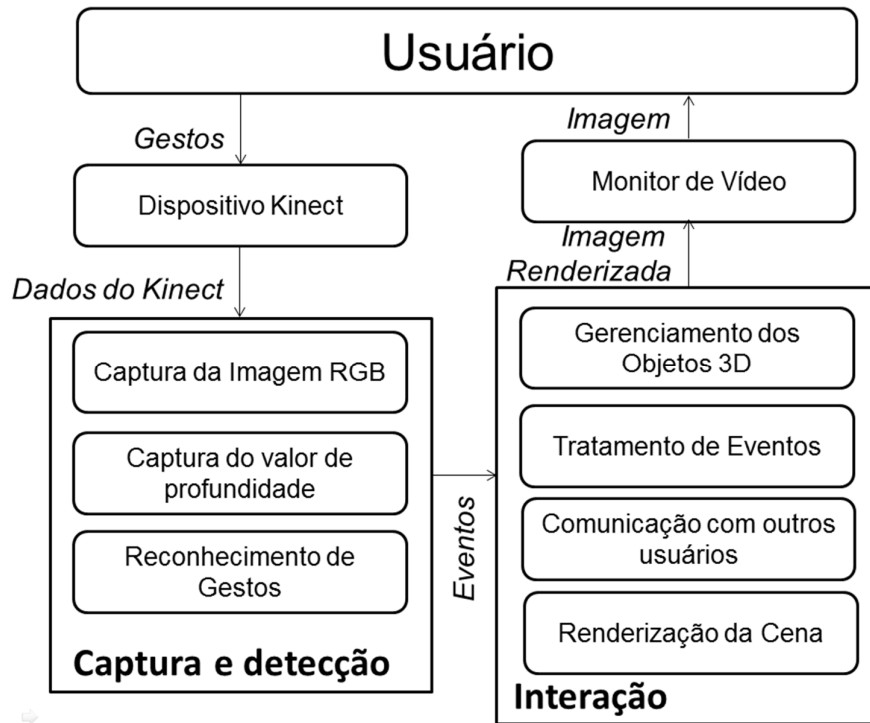


Figura 30 - Arquitetura do sistema

A comunicação entre os dois blocos é feita através de eventos, esses eventos podem ser dos tipos:

- **Captura da Imagem RGB/Profundidade:** Sempre que o usuário é detectado pelo dispositivo Kinect, as atividades de *captura da imagem RGB* e a “captura do valor de profundidade” são enviadas para o *software*, com a finalidade do reconhecimento da posição das mãos, sendo que a captura da imagem informa a posição X e Y da mão, baseado na profundidade (Z);
- **Reconhecimento de Gesto:** Sabendo a posição exata das mãos do usuário, o sistema realiza a atividade de *reconhecimento de gestos* e

tenta detectar algum gesto pré-definido, que será enviado por “evento” ao bloco de Interação. Os gestos são baseados na posição e distâncias entre as mãos e o tronco do usuário;

- **Tratamento de Eventos:** No bloco de interação a primeira atividade a ser executada é a de “tratamento de eventos”, que tenta detectar se um gesto pré-definido foi executado para repassar a manipulação ao *gerenciamento de objetos 3D*.
- **Gerenciamento de objetos 3D:** o bloco de gerenciamento de objetos 3D aplica as novas posições ao objeto selecionado e comunica com a atividade de “**Comunicação com outros usuários**”, enviando para todo o grupo de usuários na rede a nova posição (Figura 28 e Figura 29), após rotacionar ou modificar a escala de seu objeto 3D.
- **Renderização de Cena:** Após a atualização da posição e escala dos objetos, a cena é renderizada pela atividade *renderização da cena*, gerando vários quadros e sendo apresentada a uma taxa de 30 quadros por segundo, no monitor de vídeo, para os usuários.

4.3 Considerações Finais

A criação dos diagramas de atividade e sequência auxiliará no desenvolvimento das técnicas de comunicação entre os usuários e a interação natural com os objetos 3D, e possibilitam ter uma visão geral do funcionamento do *software*, possibilitando um desenvolvimento mais preciso em relação ao atendimento dos requisitos propostos.

O próximo capítulo aborda o desenvolvimento com mais detalhes de implementação.

Capítulo 5

Detalhes da Implementação

5.1 Introdução

Este capítulo tem o objetivo de explicar e detalhar as tecnologias e equipamentos utilizados para o desenvolvimento deste trabalho.

5.2 Tecnologias de Apoio

O desenvolvimento do sistema descrito neste trabalho necessitou da utilização de um dispositivo *touchless* e alguns módulos de *software*. Esta seção tem a finalidade de apresentar e descrever as características destas ferramentas, de modo a auxiliar no entendimento global do sistema.

5.2.1 Dispositivos *Touchless*

Dentre os dispositivos atuais de *touchless* destaca-se o dispositivo Microsoft Kinect® pela sua popularidade. Ademais, o baixo custo em relação aos concorrentes além da sua documentação, foram fatores determinantes para a escolha da utilização do Kinect® como dispositivo *touchless*.

Outro quesito que influenciou a escolha do dispositivo foi a afinidade do autor com os produtos da empresa Microsoft, o que agilizou a aprendizagem para o desenvolvimento deste trabalho.

O Kinect é um sensor de movimento que foi desenvolvido e vendido pela empresa Microsoft com a finalidade de funcionar como um controle do vídeo-game Xbox 360®. Seu nome original era “Projeto Natal” e foi desenvolvido para competir com o Wii® Remote vendido pela Nintendo e com o PlayStation Move® produzido

pela Sony, para interagir com os consoles *Wii®* e *PlayStation 3*, respectivamente (AVANCINI, 2011).

Como a finalidade do Kinect não se restringiu a um controle de game inovador, rapidamente, muitos pesquisadores de diversas universidades ao redor do mundo começaram a desenvolver projetos para essa tecnologia. Para realizar a comunicação entre o Kinect e o computador, os desenvolvedores utilizavam um *middleware open source* chamado “*Natural Interaction Technology for End-user*” (NITE) desenvolvido pela PrimeSense, uma empresa israelense que contribuiu com o “Projeto Natal” (REDMOND e TEL AVIV, 2010).

Atualmente, a empresa PrimeSense trabalha em conjunto com a ASUS, e desenvolveu um dispositivo similar ao Kinect chamado *WAVI Xtion®*, com a compatibilidade direta com o computador, para os sistemas operacionais Windows e Linux (ASUS, 2013).

Observando a crescente possibilidade de aplicações utilizando o Kinect em várias áreas, juntamente com o desenvolvimento do *WAVI Xtion®*, a Microsoft decidiu liberar um *driver* gratuito para os desenvolvedores chamado: *Kinect Software Development Kit (SDK) for Windows* em dezesseis de Junho de 2011 (CENTER, 2011).

Dados Técnicos Kinect

Pode-se notar na Figura 31, o Kinect® não é uma peça única e sim um produto composto por vários componentes e tecnologias diferentes, com a finalidade de gerar uma nova forma de interação.



Figura 31 - Kinect Desmontado (IFIXIT, 2012)

O Kinect foi projetado com duas câmeras do tipo infra-vermelho para detectar a distância de obstáculos. Uma câmera faz a projeção dos pontos infra-vermelhos e a outra recebe os dados. A câmera que recebe os dados tem uma resolução de 320x240 e possui valores de 16 bits para calcular a profundidade do obstáculo. O Kinect ainda possui uma câmera RGB de resolução 640x480 com 32 bits de cores e 30 frames por segundo (Figura 32) (KERKHOVE, 2012).

Além das câmeras que realizam a detecção de objetos ou pessoas, o Kinect ainda possui um motor que ajusta o ângulo de visão. Esse ângulo varia em 57° na horizontal e 43° na vertical e o mecanismo que fica na base consegue mover 27° para subir ou descer na vertical, modificando a visão das câmeras (KERKHOVE, 2012).



Figura 32 - Descrição Kinect (baseado em (SDK, 2011))

Internamente o Kinect ainda possui quatro microfones distribuídos na parte de baixo que servem para detectar e reconhecer comandos de voz dos usuários. Os microfones conseguem diferenciar comandos de voz de barulhos do ambiente (KERKHOVE, 2012).

Tabela 5 - Dados Kinect (SDK, 2011)

Dados do Sensor	Especificação
Ângulo de Visão	43° na Vertical e 57° na Horizontal
Motor na base (tilt)	±28° na vertical
Taxa de Quadros por segundo	30 FPS
Resolução (profundidade)	QVGA (320x240)
Resolução (cor)	VGA (640x480)
Audio	16-KHz, 16 bits (PCM)

O Kinect tem a capacidade de detectar e rastrear até seis pessoas ao mesmo tempo, assim como vinte partes específicas (Figura 33) do corpo de cada uma dessas pessoas. A distância que o Kinect consegue rastrear objetos e pessoas é de 0.8 metros até 4.0 metros, porém em distâncias de 1.2 metros até 3.5 metros é possível obter melhores resultados (KINECT, 2012).

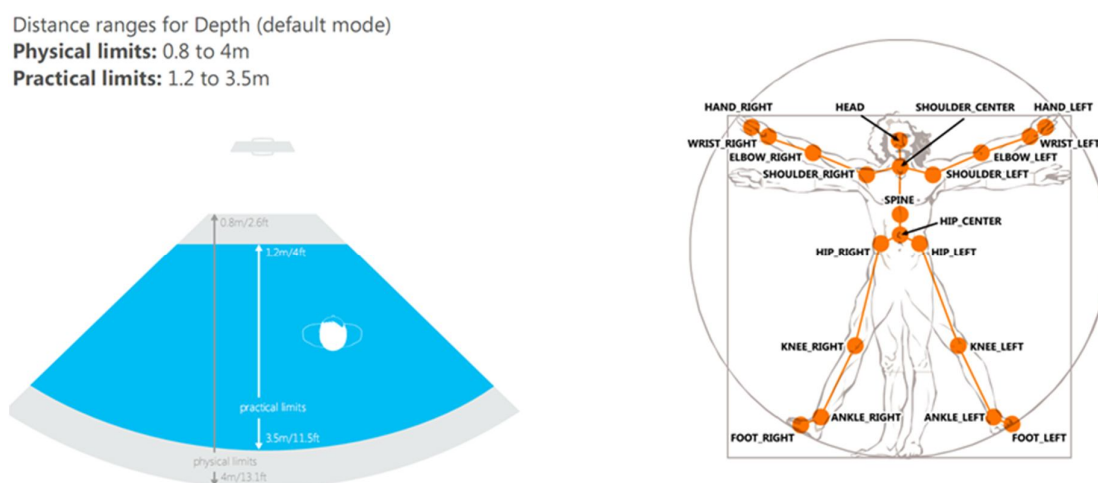


Figura 33 - Capacidades físicas do kinect (baseado em KINECT 2012)

Quanto ao *hardware*, pode-se considerar que a configuração utilizada se aproxima dos equipamentos utilizados no Brasil, ou seja, não é necessária nenhuma

placa de vídeo especial. Os equipamentos utilizados para o desenvolvimento e teste foram:

- Notebook: Processador Intel®Core™2 Duo T6670 @ 2.20GHz, memória RAM: 4GB, placa de vídeo integrada.
- PC: Placa mãe Asus P5K Premium™, processador Intel Intel®Core™2 Duo E4700 @ 2.60GHz, memória RAM: 4GB, placa de vídeo: ATI™Radeon HD 3450.

5.2.2 Softwares

As ferramentas de *software* para o desenvolvimento do aplicativo, descritas a seguir, foram escolhidas levando em conta alguns fatores como: preço, eficiência, facilidade e afinidade do desenvolvedor. O objetivo principal dessa seção não é avaliar as ferramentas e sim realizar uma breve descrição.

• **Driver de Comunicação**

Para realizar a comunicação entre o dispositivo *touchless* (Kinect®) e a aplicação computacional é necessário um *framework*. Entre as opções mais conhecidas se destacam os *frameworks* *OpenNI* e o Microsoft *Software Development Kit (SDK)*.

O Microsoft *SDK* é um conjunto de ferramentas que inclui *drivers* para instalação do Kinect e componentes de *middleware*⁴ para desenvolvedores. O *middleware* inclui a biblioteca **NUI Library**, com classes, estruturas e tipos de dados, para a comunicação com a aplicação (Figura 34) (CORPORATION, 2011).

⁴ *Middleware*: É geralmente constituído por módulos dotados com APIs de alto nível que proporcionam a sua integração com aplicações desenvolvidas em diversas linguagens de programação. Seu objetivo é mascarar a heterogeneidade e fornecer um modelo de programação mais produtivo para os programadores de aplicativos.

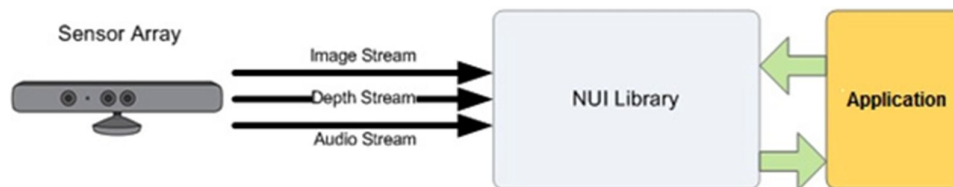


Figura 34 - Interação Hardware e Software (CORPORATION, 2011).

A Tabela 6 apresenta algumas das principais classes do *middleware* Microsoft SDK, cada classe facilita o desenvolvimento de aplicativos com o Kinect.

Tabela 6 - Classes do Middleware SDK (CORPORATION, 2011).

Classes	
Nome	Descrição
Camera	Representa o dispositivo Kinect instalado.
ImageFrame	Representa a imagem gerada com os dados do kinect.
Runtime	Representa o “pipeline” de execução do Kinect.
SkeletonData	Armazena os dados do esqueleto encontrado.
JoinsCollection	Apresenta a coleções de partes do corpo rastreado, e suas posições

O Microsoft SDK pode ser incluído na IDE (*Integrated Development Environment*) do Microsoft Visual Studio™, podendo incluir a biblioteca do Kinect e acessar as classes do *middleware* Microsoft SDK. As classes podem ser utilizadas nas linguagens da plataforma .NET, especialmente as linguagens C++ e C#.

• Ambiente de Desenvolvimento

Visual Studio™ é um conjunto completo de ferramentas de desenvolvimento para construção de aplicações Web ASP.NET, serviços Web XML, aplicações *desktop* e aplicativos móveis. Visual Basic, Visual C# e Visual C++ usam todos o mesmo ambiente de desenvolvimento integrado (IDE), que permite o compartilhamento de ferramentas e facilita a criação de soluções com mistura de linguagens (MICROSOFT, 2012).

O Microsoft Visual Studio™ possui diversas versões: *Ultimate*, *Premium*, *Professional* e *Express*; com exceção da versão *Express*, as demais são pagas, por isto, optou-se por se utilizar a versão *Express*.

O DirectX® é uma biblioteca gráfica, desenvolvida pela Microsoft, que foi apresentada em 1995, oferecendo uma nova alternativa ao OpenGL, com alto desempenho para gráficos bidimensionais, entrada, comunicação, som e por fim, gráficos tridimensionais (NATHAN, 2007). Atualmente o DirectX está na sua versão 9, com um ganho considerável de recurso para gráficos 3D, porém, a biblioteca DirectX9 fica limitada, quando se trabalha com interfaces.

A Microsoft reconheceu a necessidade de desenvolver um novo tipo de interface, diferente do Windows Form, para que fosse possível utilizar todos os recursos do DirectX9®, sem esquecer da performance do Windows Form (NATHAN, 2007).

Desta maneira, surgiu o *Windows Presentation Foundation* (WPF), sendo uma solução para os desenvolvedores de software e para os *designers* gráficos. O WPF foi incluído no .NET Framework 3.0 SP1 (Figura 35), o Microsoft Visual Studio 2010 já vem com o .NET Framework 3.5, não sendo necessária a atualização do .NET Framework.

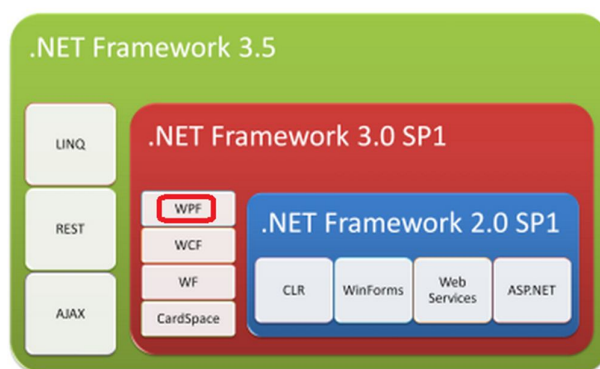


Figura 35 - WPF no Framework 3.0 (baseado em (SNEATH, 2008))

Utilizando os recursos gráficos do WPF juntamente com o *middleware* Microsoft SDK no Microsoft Visual Studio 2010, deve-se escolher entre as linguagens de programação C++ ou C#. Para este trabalho optou-se pela linguagem de programação C# por afinidade do desenvolvedor já que as duas linguagens possuem os mesmos recursos do *middleware* Microsoft SDK.

Outro fator que contribuiu pela escolha da IDE Microsoft Visual Studio 2010 foram as bibliotecas que permitem a troca de dados entre usuários. O Microsoft Visual Studio possui algumas bibliotecas para o uso de *Threads* e *Sockets* que auxiliaram o desenvolvimento do aplicativo.

5.3 Interface Gráfica

A interface gráfica consiste em uma maneira de visualização do ambiente de RV pelo usuário. Por isso, elaborou-se uma interface com foco principal nos objetos 3D do ambiente (Figura 36), ficando em segundo plano detalhes como *menu* e usuários conectados.

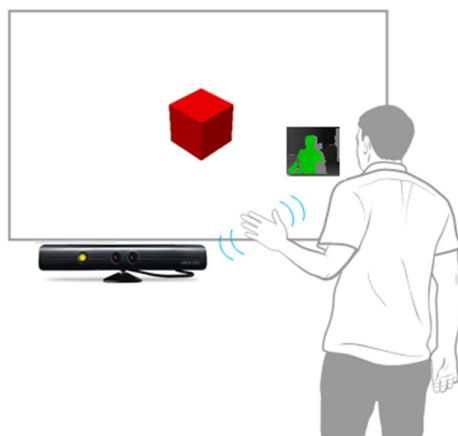


Figura 36 - Interface do aplicativo desenvolvido

O usuário deverá se orientar observando a sua imagem no canto direito inferior da tela e, dessa maneira, identificar a posição de suas mãos para a execução dos gestos que irão modificar o objeto selecionado.

A cena que compõe o ambiente de RV foi construída com os seguintes componentes:

- **Câmera:** é o objeto responsável por captar os gestos. Ela fica fixa em uma posição do espaço;
- **Menu:** Lista os usuários que estão conectados ao sistema e que podem receber e enviar dados aos demais usuários;

- Label: Informa se o usuário está recebendo os dados ou se o sistema está liberado para que os mesmos possam ser enviados;
- Painéis: são componentes que informam a posição e a quantidade de usuários detectados;
- Objetos 3D: apresenta o objeto selecionado, o qual irá sofrer as modificações de rotação e escala.

A Figura 37 apresenta a constituição da interface gráfica.

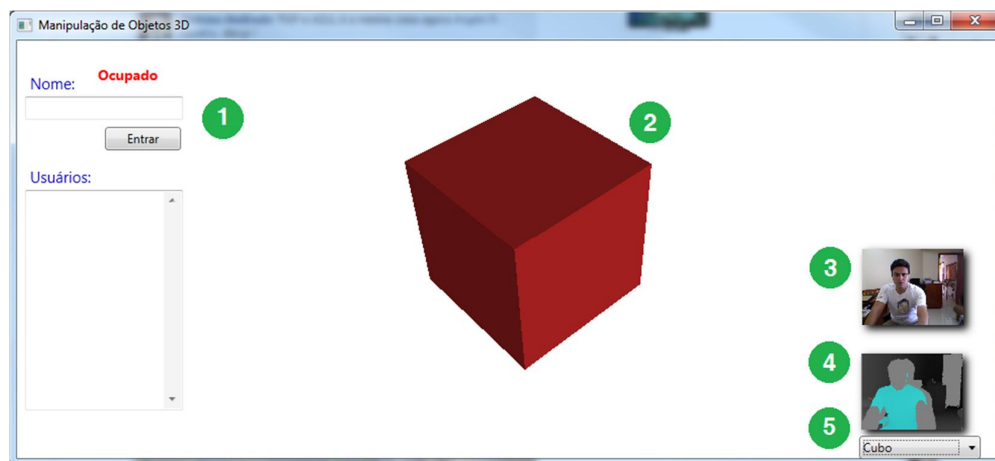


Figura 37 - Componentes da Interface Gráfica

Na Figura 36, pode-se observar os principais componentes da interface gráfica, sendo que:

1. Consiste em um painel para que o usuário possa se conectar a um grupo de usuários, através da internet. Ao se conectar a esse grupo, o usuário poderá enviar e receber atualizações da posição e escala do objeto 3D. Na próxima seção será mais detalhado o desenvolvimento do módulo de comunicação.
2. Apresenta o objeto 3D que será manipulado. Esse objeto 3D pode ser escolhido através do *submenu* do item 5 ou através da rede por outros usuários.
3. Apresenta a imagem RGB do usuário. A partir desta, o usuário pode observar sua imagem, como em um espelho, e determinar seus gestos.

4. O item 4 apresenta a imagem obtida pela nuvem de pontos, com o corpo do usuário em uma cor destacada. Sendo assim, é possível observar se o usuário foi detectado pelo dispositivo *touchless* e se está apto a realizar os gestos pré-definidos.

Nas próximas sessões será detalhado o desenvolvimento de cada componente e a como eles atuam no sistema. Os principais módulos são:

- Comunicação com o Sensor e Detecção de Gestos;
- Detecção de Gestos e renderização da cena;
- Comunicação entre os Usuários.

5.4 Comunicação com o Sensor e Detecção de Gestos

A obtenção dos dados do sensor, assim como o reconhecimento do usuário e dos gestos, podem ser consideradas as partes principais do sistema, uma vez que a maioria dos outros módulos dependem dessas etapas.

A interface do sistema foi construída utilizando o *Windows Presentation Foundation* (WPF) do Visual Studio 2010TM. O WPF é constituído por uma linguagem de marcação chamada de XAML (*Extensible Application Markup Language*), que é um tipo de arquivo XML (*Extensible Markup Language*), onde são determinados os componentes utilizados, conforme observado no fragmento de Código 1, esse código é responsável pela criação de um painel onde os objetos 3D serão apresentados (linha 9) e também pelo posicionamento da iluminação da cena (linha 14 à linha 20).

Código 1 - XAML para criação da Interface

```

1 <Window
2   x:Class="Kinect3dManipulation.MainWindow"
3   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
4   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5   xmlns:cp="http://schemas.charlespetzold.com/2007/xaml"
6   Title="Manipulação de Objetos 3D"
7   Height="450"
8   Width="985" Loaded="Window_Loaded">
9   <Grid Height="338" Width="575">
10    <Viewport3D x:Name="viewport" Height="440" VerticalAlignment="Top" Margin="0,-73,-45,0">
11
12      <!-- Lighting -->
13      <ModelVisual3D x:Name="iluminacao">
14        <ModelVisual3D.Content>
15          <Model3DGroup>
16            <DirectionalLight Color="#808080" Direction="2 -3 -1" />
17            <AmbientLight Color="#808080" />
18          </Model3DGroup>
19        </ModelVisual3D.Content>
20      </ModelVisual3D>
21    </Viewport3D>
22  </Grid>
23 </Window>

```

A linguagem XAML foi utilizada para modelar os objetos 3D utilizados neste trabalho, os objetos foram modelados diretamente no Visual Studio 2010TM, como apresentado no Código 2, sendo os tipos primitivos do WPF.

Código 2 - Modelagem dos objetos 3D

```

24 <!-- Cubo -->
25 <cp:Cuboid Height="1" Depth="1" Width="1" x:Name="cubo">
26   <cp:Cuboid.Material>
27     <DiffuseMaterial>
28       <DiffuseMaterial.Brush>
29         <SolidColorBrush Color="Firebrick"/>
30       </DiffuseMaterial.Brush>
31     </DiffuseMaterial>
32   </cp:Cuboid.Material>
33 </cp:Cuboid>
34
35 <!-- Cilindro -->
36 <cp:Cylinder Radius1=".5" Radius2="1.0" x:Name="cilindro">
37   <cp:Cylinder.Material>
38     <DiffuseMaterial>
39       <DiffuseMaterial.Brush>
40         <SolidColorBrush Color="DarkOrange"/>
41       </DiffuseMaterial.Brush>
42     </DiffuseMaterial>
43   </cp:Cylinder.Material>
44   <!--<cp:Cylinder.Transform>
45     <TranslateTransform3D OffsetX="2.1" OffsetZ="-1.0"/>
46   </cp:Cylinder.Transform> -->
47 </cp:Cylinder>

```

Para a modelagem dos tipos primitivos de objetos no XAML, são fornecidos os valores de: raio ou tamanho, tipo de material e cores.

Para a modelagem de objetos 3D complexos podemos utilizar diretamente a linguagem XAML ou importar modelos criados em outras ferramentas, como Blender® ou 3D Studio Max®, para o WPF e adicionar à interface.

Segundo (BJORKE, 2012) o Helix 3D Toolkit® promete exportar arquivos nos formatos: Kerkythea render engine, Wavefront files, X3D, Collada, VRML97 e POV-Ray para o WPF. De acordo com (BJORKE, 2012) o Helix 3D Toolkit® não está totalmente finalizado e ainda possui erros como modificação da escala, alteração da cor e do material.

5.4.1 Módulo de Captura e Detecção

Sempre que o aplicativo se inicia é realizada a comunicação entre o dispositivo Kinect e a aplicação. Isso é necessário para que sejam recebidos os dados da imagem RGB e de profundidade (Código 3). Esse processo faz parte do módulo de captura e detecção (Figura 30).

Código 3 - Funções para receber as imagens RGB e de profundidade.

```

53 //Quando o programa inicia, as imagens da interface são constantemente atualizadas.
54 void MainWindow_Loaded(object sender, RoutedEventArgs e)
55 {
56     this.viewport.Camera.Transform = this.trackball.Transform;
57
58     this.imageWorker = new BackgroundWorker();
59     this.imageWorker.DoWork += new DoWorkEventHandler(imageWorker_DoWork);
60 }
61 //Função que recebe os dados do dispositivo Kinect.
62 void imageWorker_DoWork(object sender, DoWorkEventArgs e)
63 {
64     Thread.Sleep(20);
65     Dispatcher.BeginInvoke((Action)delegate
66     {
67         //Comando para receber os dados do Kinect.
68         this.depthImage.Source = Kinect.GetDepthFrame();
69         this.rawImage.Source = Kinect.GetColorVideoFrame();
70     });
71 }

```

No módulo de captura e detecção ocorre também o processo para o reconhecimento de gestos. É possível ao usuário realizar gestos de rotação nos três eixos (X, Y e Z) e modificar a escala (aumentar ou diminuir) do objeto 3D selecionado.

Para o reconhecimento destes gestos é necessária a identificação da posição das mãos do usuário. Com esta finalidade identifica-se algumas posições pre-definidas para rotação (Figura 38a) e para escala (Figura 38b).

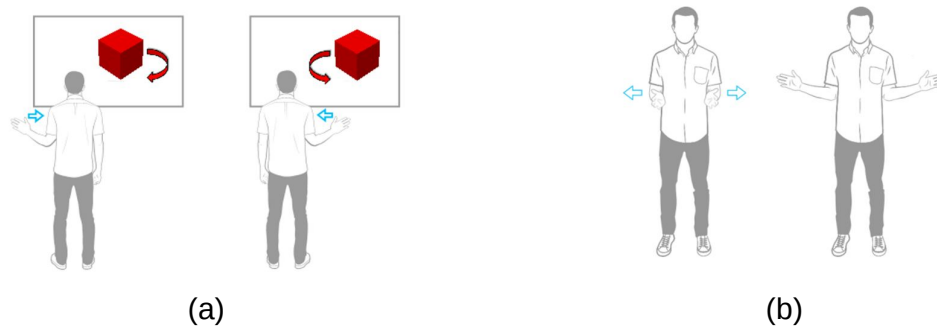


Figura 38 - Gestos pré-definidos

Para determinar os gestos, além da identificação da posição das mãos, é necessário que seja definida a distância que existe entre as mãos e o tronco do usuário. Neste trabalho foram utilizados quatro centímetros é a distância mínima para considerar que a mão não está próxima ao tronco (Código 4).

A partir desses dados, é possível definir através de qual eixo o objeto irá rotacionar e qual o sentido de rotação. Sendo assim, uma mão deve ficar junta ao tronco e a outra mão irá realizar o movimento para rotacionar o objeto através do eixo (Figura 38a).

Código 4 – Detecção dos gestos.

```

const double MinTorsoDistance = 0.4d;
63 void RefreshSensorPositions()
64 {
65     //Comandos para detectar as mãos do usuário.
66     this.leftHand = Kinect.LeftHand;
67     this.rightHand = Kinect.RightHand;
68     //Comandos para obter a posição das mãos do usuário.
69     this.rightHandPoint = Kinect.GetRightHandDisplayPosition(Width, Height);
70     this.leftHandPoint = Kinect.GetLeftHandDisplayPosition(Width, Height);
71     //Comando pra obter a posição do tronco do usuário.
72     this.torso = Kinect.Torso;
73     //Atualização de valores, para detecção dos gestos.
74     this.rightHandCanMove =
75         Math.Abs(this.torso.Z - this.rightHand.Z) >= MinTorsoDistance &
76         Math.Abs(this.torso.Z - this.leftHand.Z) < MinTorsoDistance;
77
78     this.leftHandCanMove =
79         Math.Abs(this.torso.Z - this.leftHand.Z) >= MinTorsoDistance &
80         Math.Abs(this.torso.Z - this.rightHand.Z) < MinTorsoDistance;
81
82     this.canZoom =
83         Math.Abs(this.torso.Z - this.rightHand.Z) >= MinTorsoDistance &
84         Math.Abs(this.torso.Z - this.leftHand.Z) >= MinTorsoDistance;
85 }

```

Os dados são passados para o módulo de Interação através de eventos para o processo de tratamento de eventos. Será mais detalhado o funcionamento na próxima seção.

5.4.2 Módulo de Interação

Após receber os dados (através de eventos do módulo de Captura e Detecção) o processo de tratamento de eventos irá verificar o tipo de movimento a ser realizado (Código 5).

Esse processo é verificado de dez em dez milissegundos com a finalidade de sempre manter atualizados os gestos e a nova posição do objeto, assim como enviar a nova posição para os outros usuários cadastrados ao grupo *multicast*.

Código 5 - Fragmento de Código para o reconhecimento de gestos

```

121     if (this.movingWithLeft & this.leftHandCanMove)
122     {
123         this.Move(this.leftHandPoint);
124         return;
125     }
126
127     if (this.movingWithRight & this.rightHandCanMove)
128     {
129         this.Move(this.rightHandPoint);
130         return;
131     }
132
133     if (!this.zooming & this.canZoom)
134     {
135         this.zooming = true;
136         this.StartZoom(this.rightHandPoint);
137         return;
138     }
139
140     if (this.zooming & this.canZoom)
141     {
142         this.Zoom(this.rightHandPoint);
143         return;
144     }

```

Após a identificação do gesto realizado, a função *TimerTick* (que faz parte do processo de Gerenciamento de objetos 3D e é utilizada para a execução do gesto) transmite a posição da mão para o método que irá executar a manipulação do objeto

3D. Os métodos *Move* e *Zoom* são responsáveis, respectivamente, pela rotação e escala dos objetos e por enviar a nova posição (Código 6) para os usuários da rede, a fim de que todos vejam a modificação no objeto.

As linhas 211, 212 e 213 do Código 6 são responsáveis pelo envio de dados da nova posição após a rotação para o grupo de usuários. E as linhas 229, 230 e 231 do Código 6 envia os dados da escala.

Código 6 - Fragmento de código para executar o gesto e enviar para o grupo.

```

206 void StartZoom(Point point)
207 {
208     if (previousPosition2D != point)
209     {
210         //Envia os dados de posição para o grupo Multicast
211         buffer = Encoding.Unicode.GetBytes(point.ToString());
212         IPEndPoint remoteep = new IPEndPoint(multicastaddress, 2222);
213         udpclient.Send(buffer, buffer.Length, remoteep);
214         this.previousZoomPosition2D = point;
215         this.previousPosition3D = ProjectToTrackball(
216             EventSource.ActualWidth, EventSource.ActualHeight, point);
217     }
218 }
219
220 void Move(Point currentPosition)
221 {
222     Dispatcher.BeginInvoke(new Action(() =>
223     {
224         if (previousPosition2D != currentPosition)
225         {
226             Track(currentPosition);
227             previousPosition2D = currentPosition;
228             //Envia os dados de posição para o grupo Multicast
229             buffer = Encoding.Unicode.GetBytes(currentPosition.ToString());
230             IPEndPoint remoteep = new IPEndPoint(multicastaddress, 2222);
231             udpclient.Send(buffer, buffer.Length, remoteep);
232         }
233     }));
234 }

```

O processo de comunicação com outros usuários é realizado através do envio de mensagens de atualização para os objetos 3D. Essas mensagens são enviadas utilizando um grupo de usuários *multicast*.

Sempre que o aplicativo desenvolvido se inicia, o usuário tem a opção de se conectar ao grupo (Código 7). Ao se conectar é criado um cliente do tipo *UDP*, que será utilizado para enviar e receber mensagem. Quando um usuário está realizando algum gesto, todos os outros usuários recebem os valores atualizados de posição e de escala. Quando este usuário para de realizar gestos, o sistema fica disponível para que outro usuário realize algum novo gesto e compartilhe.

Código 7 - Fragmento de código para inserção de novo usuário

```

107 private void nomeButton_Click(object sender, RoutedEventArgs e)
108 {
109     //adicionando um novo usuário ao grupo.
110     UdpClient udpclient = new UdpClient();
111     IPAddress multicastaddress = IPAddress.Parse("239.0.0.220");
112     IPEndPoint localEp = new IPEndPoint(IPAddress.Any, 2222);
113     byte[] buffer;
114     try
115     {
116         Byte[] data = udpclient.Receive(ref localEp);
117         usersTextBox.Text = data.ToString();
118     }
119     catch
120     {
121         MessageBox.Show("Error");
122     }
123     usersTextBox.Text += nomeTextBox.Text + "\n";
124     nomeTextBox.Clear();
125
126     buffer = Encoding.Unicode.GetBytes(usersTextBox.Text);
127     IPEndPoint remoteep = new IPEndPoint(multicastaddress, 2222);
128     udpclient.Send(buffer, buffer.Length, remoteep);

```

Dessa maneira, cada usuário se conecta ao grupo através de seu endereço IP, que fica referenciado ao *multicastaddress* “239.0.0.220” (Linha 111 do Código 5). Assim, sempre que algum usuário envia algum dado para o IP “239.0.0.220”, todos os usuários conectados ao grupo recebem (Linha 128 do Código 7).

Após o usuário ter se conectado ao grupo, estará apto a enviar e receber as posições atualizadas, assim como o valor de escala dos objetos 3D de outros usuários. Para receber os dados, existe um Thread que fica monitorando o módulo de captura e detecção e atualiza os valores de posição para os objetos 3D.

Como a posição do objeto 3D é constantemente atualizada, pois as mensagens tem um tamanho reduzido, é dessa forma que o processo de renderização da cena não é prejudicado em termos de processamento, conseguindo gerar a imagem a uma taxa de 30 quadros por segundo e envia-la ao monitor para ser apresentado ao usuário.

Dessa maneira, todos os processos interagem entre si em uma sequência determinada, conforme (Figura 27) apresentada no Capítulo 4. No Capítulo 6 serão apresentados e discutidos os resultados obtidos com o aplicativo desenvolvido.

5.5 Considerações Finais

Após a implementação dos módulos de “captura e detecção” e de “interação”, foi possível se obter uma aplicação computacional que comporta gestos de interação natural em um ambiente colaborativo, com usuários dispersos remotamente.

A aplicação computacional utilizou técnicas de endereços *multicast* e *sockets* para a troca de informações entre os usuários, para o ambiente colaborativo, e a partir da posição das mãos foram implementados gestos para rotação e escala em torno de todos os eixos.

Capítulo 6

Análise dos Resultados

6.1 Introdução

Este capítulo tem o objetivo de apresentar os resultados obtidos com o desenvolvimento do aplicativo proposto neste trabalho.

Para a discussão serão levantados os pontos propostos no Capítulo 3, que são:

- Manipulação de objetos 3D;
- Uso de Interfaces Naturais;
- Ambientes Colaborativos com usuários dispersos remotamente;

Para facilitar o entendimento e a visualização de um modo geral, o tratamento dado a esses problemas será apresentado e discutido com o auxílio de imagens e resultados dos testes realizados.

6.2 Manipulação de Objetos 3D

Uma das dificuldades da aplicação computacional, apresentada neste trabalho, consiste na detecção da posição das mãos e no reconhecimento dos gestos de uma maneira rápida, eficiente e com uma taxa aceitável de acertos (independente do ambiente e da luz em que o usuário se encontra). Além disso, deve-se atentar para o requisito de execução em tempo real, necessário para a imersão do usuário no ambiente de RV.

O uso do dispositivo Kinect e do *driver* SDK da Microsoft permite que o usuário, assim como a posição das suas mãos, sejam detectados a todo instante independente de obstáculos ou a iluminação do ambiente conforme Figura 39.

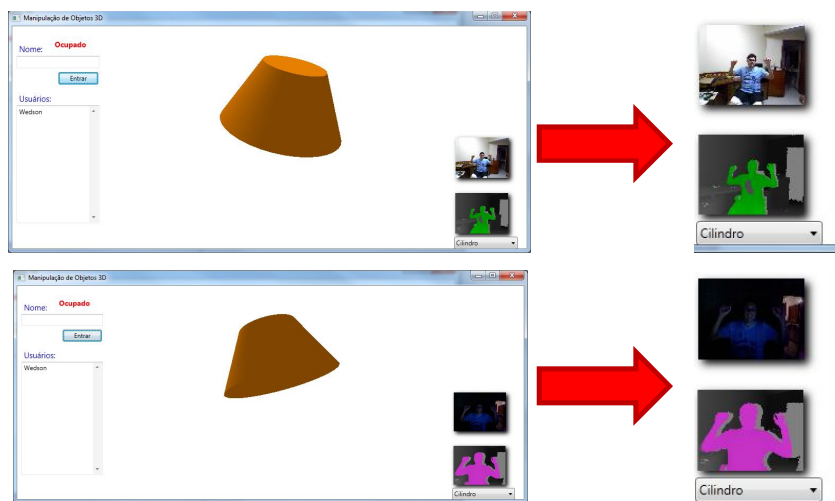


Figura 39 - Detecção com iluminação diferente

As posições das mãos são atualizadas constantemente, de forma que a qualquer alteração na posição ou um novo gesto ou movimento criados pelo usuário sejam detectados.

Os gestos, conforme apresentado no Capítulo 5, dependem das posições das mãos e da distância delas até o tronco do usuário. Dessa forma, é possível determinar o tipo de gesto desejado e aplicar ao objeto um valor para rotacionar ou alterar a escala do mesmo, aumentando ou diminuindo.

6.3 Uso de Interfaces Naturais

A utilização de interfaces naturais, neste trabalho, foi importante para usabilidade, já que o objetivo é evitar que usuário necessite de algum tipo de treinamento prévio antes da utilização do sistema.

Além da vantagem de não necessitar de um treinamento prévio, a utilização de interfaces naturais auxiliam na imersão do usuário no sistema de RV, já que são utilizados apenas gestos com as mãos.

Foi realizada uma avaliação junto aos usuários, com o objetivo de quantificar o nível de reconhecimento dos gestos e o nível de facilidade ao utilizar o sistema sem um treinamento prévio.

- Reconhecimento do gesto:
 - o Níveis de reconhecimento:
 - Não reconheceu o gesto;
 - Reconheceu o gesto errado;
 - Reconheceu o gesto correto, porém com atraso;
 - Reconheceu o gesto correto em tempo real.

- Satisfação do usuário:
 - o Níveis de satisfação:
 - Insatisfeito;
 - Pouco satisfeito;
 - Parcialmente satisfeito;
 - Satisfeito;

Os testes foram realizados em um ambiente com outros objetos e iluminação recomendada para uma sala comercial, ou seja de 20 à 50 lux segundo (NBR 5413, 1992). A quantidade de usuários e suas características estão apresentadas na Tabela 7.

Tabela 7 - Quantidade de Pessoas que participaram dos testes

Idade	Quantidade de Pessoas
18 – 25 anos	3
26 – 35 anos	3
36 – 45 anos	2
46 anos ou mais	2

O teste foi realizado da seguinte maneira: cada usuário deveria realizar cinco gestos de rotação e cinco gestos de escala, totalizando dez gestos.

Antes de o usuário realizar os dez gestos, o mesmo testou o sistema quanto à distância e a sensibilidade do Kinect, e após esse tempo, foram realizados os dez gestos. Através de um questionário (Apêndice A) foram ponderados e quantificados

os acertos e os níveis de reconhecimento de gestos. A Figura 40 apresenta os resultados.

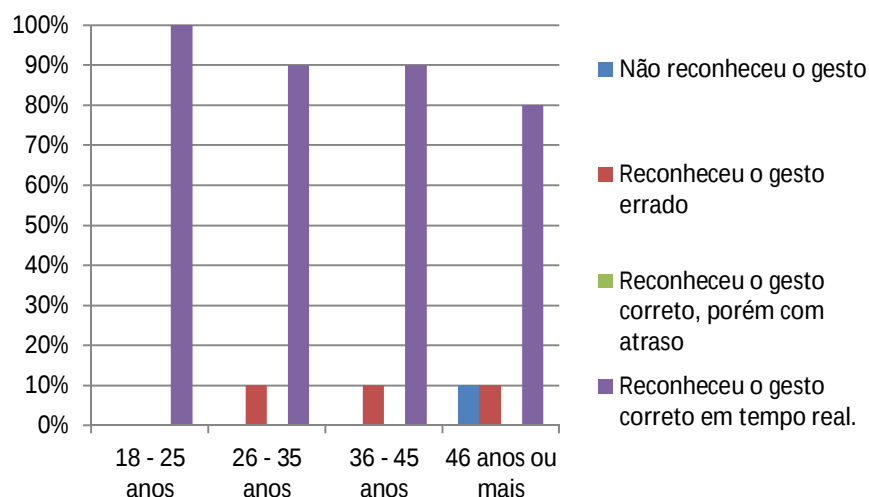


Figura 40 - Níveis de Reconhecimento dos Gestos.

As Figuras 41 e 42 mostram dois usuários realizando os gestos de rotação e escala. Eles foram informados sobre a finalidade do teste e que era possível manipular o objeto 3D através de gestos com as mãos. Cada um deles experimentou algum tipo de movimento até conseguir realizar o movimento desejado. Dessa forma, utilizaram sua experiência prévia sobre manipulação dos objetos 3D, apenas com o conhecimento de que o dispositivo Kinect® consegue captar as posições das mãos.

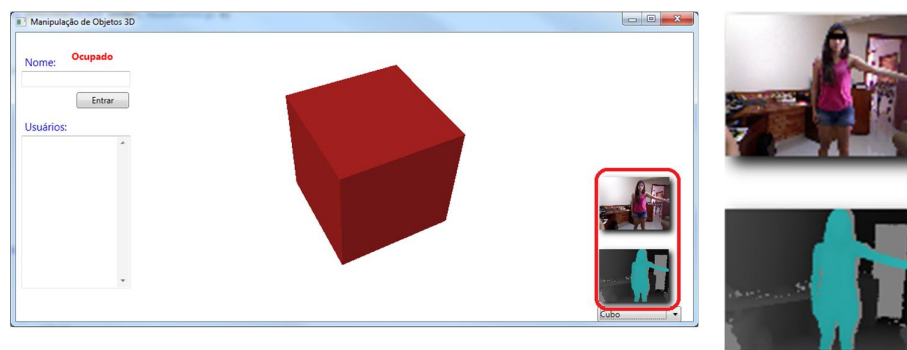


Figura 41 - Teste com pessoa do sexo feminino e 28 anos

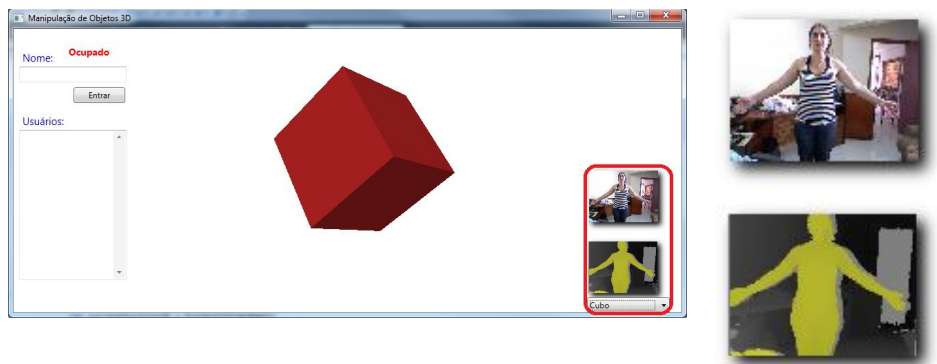


Figura 42 - Teste realizado com usuário de 53 anos

O questionário (Apêndice A), sobre o nível de satisfação dos usuários, ponderou à satisfação dos usuários de um modo geral, tendo ênfase na questão do ambiente colaborativo.

Os usuários foram separados em pares, sendo que cada um ficou em frente à um computador com dispositivo Kinect® diferente, enquanto um usuário realiza os 10 gestos propostos o outro usuário apenas observava, e após o primeiro usuário finalizar os 10 gestos era feito o contrário, o usuário que observou primeiramente passou a manipular os objetos e o outro usuário passou a observar.

6.4 Ambientes colaborativos com usuários dispersos remotamente

O dispositivo Kinect permite que dois ou mais usuários sejam reconhecidos. Ademais, as posições das suas juntas também podem ser detectadas, quando, os dois usuários estão no mesmo local e em posição de captura de um mesmo dispositivo Kinect. Porém não existe nenhum módulo ou classe pronta desenvolvida para o dispositivo Kinect®, que permita aos usuários dispersos remotamente a comunicação e a troca de experiências.

Como etapa deste trabalho é desenvolver uma arquitetura que permita a comunicação entre usuários dispersos remotamente. Para isso, foi criada uma camada de comunicação para troca de dados, com as posições e escala dos objetos.

Quando o aplicativo permite que o usuário envie os dados, um texto escrito “Liberado” na cor verde fica visível, e para os demais usuários conectados na rede, aparecerá um texto escrito “Ocupado” em vermelho, conforme Figura 43.

Como a troca de dados entre os usuários é apenas para a atualização das posições e da escala, caso um pacote de dados seja perdido, outro irá chegar com a nova atualização da posição, sendo dessa maneira, imperceptível para o usuário o atraso ou perda dos dados.

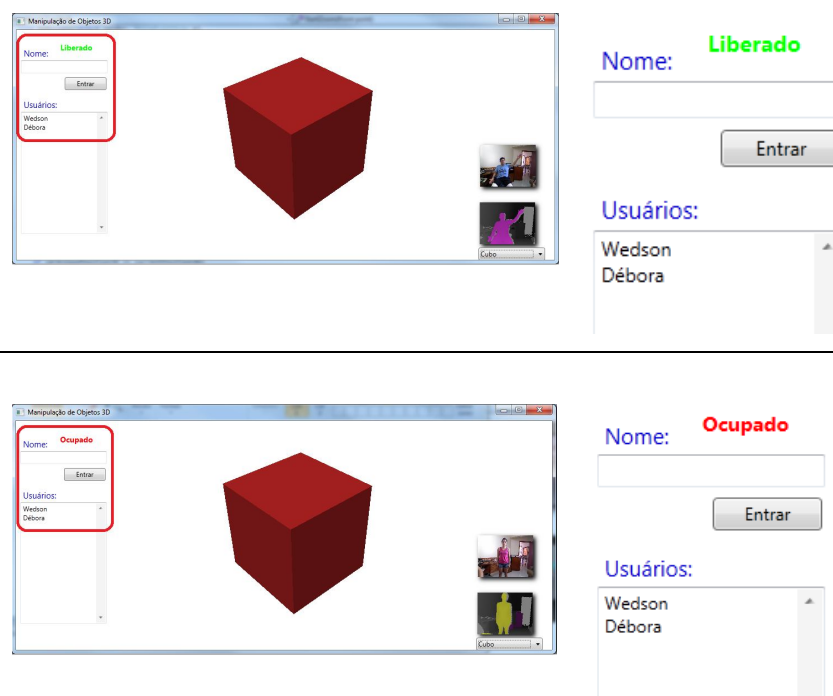


Figura 43 - Comunicação entre usuários.

6.5 Considerações Finais

Levando em consideração os testes de reconhecimento de gestos naturais para manipulação de objetos 3D e a colaboração com usuários dispersos remotamente, o sistema conseguiu cumprir a proposta descrita no Capítulo 3. Além do questionário sobre o nível de reconhecimento dos gestos, foi realizado também um questionário sobre o nível de satisfação do usuário, de acordo com a parte de colaboração do sistema, conforme apresentado na sessão 6.3 deste Capítulo.

A Figura 44 apresenta a satisfação dos usuários em relação ao módulo de colaboração entre usuários dispersos remotamente. Vale ressaltar que, seria

importante realizar testes de usabilidade mais completos, no entanto, estes fogem do escopo deste trabalho.

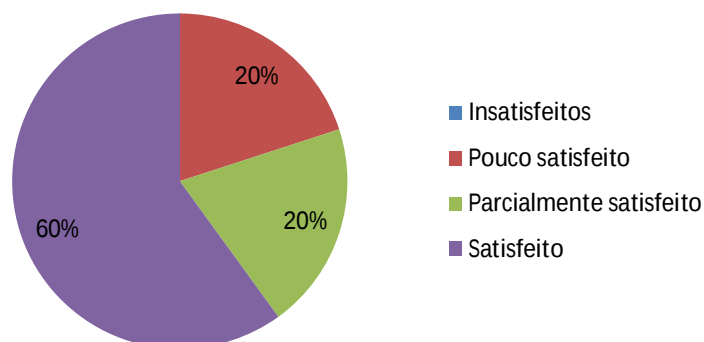


Figura 44 - Satisfação dos Usuários

Alguns usuários relataram críticas e sugestões, em relação ao sistema, essas críticas foram analisadas e algumas incluídas como trabalhos futuros:

- Utilizar objetos 3D (cubo) com faces de cores diferentes para facilitar a visualização quando for realizada uma rotação;
- Permitir a importação de outros objetos 3D e incluir movimentos a esses objetos através dos gestos;
- Desenvolver uma ferramenta para a comunicação entre os usuários do grupo, como: chat ou vídeo conferência.

Por fim, é possível observar a contribuição (Tabela 8) deste trabalho, através da comparação com outras pesquisas relacionadas, como foi apresentado anteriormente na Tabela 1 do Capítulo 3.

Tabela 8 - Comparação dos Trabalhos Relacionados

Tópico Abordado	FAAST	Kinoogle	Arthron	Fusion 4D	Manipulação Automobilístico	Manipulação de Objetos 3D em Ambientes Colaborativos por NUI
Interação <i>touchless</i>	✓	✓	✓	✓	✓	✓
RV	✗	✗	✓	✓	✓	✓
Movimentação de objetos 3D	✗	✗	✓	✓	✓	✓
Gestos Naturais	✗	✓	✓	✓	✓	✓
Deteção dos gestos sem menu	✗	✗	✗	✓	✗	✓
Colaboração entre usuários	✗	✗	✗	✗	✗	✓

Capítulo 7

Conclusões e Trabalhos Futuros

7.1 Introdução

Este capítulo tem como foco destacar os principais pontos estudados nessa dissertação, avaliar a contribuição para a comunidade científica e apresentar possíveis melhorias e trabalhos futuros.

7.2 Conclusões

O objetivo deste trabalho consistiu em pesquisar formas de interação natural para manipulação de objetos 3D e aplicá-las em Ambientes Virtuais Colaborativos, permitindo que usuários dispersos remotamente, possam manipular objetos 3D por meio de gestos naturais e compartilhar suas experiências com outros usuários.

Com base no estudo dos fundamentos teóricos, percebeu-se que o uso de interfaces naturais com as mãos, por meio de gestos, consiste em uma interface intuitiva e aumenta a imersão do usuários para a manipulação 3D. Outra questão relevante foi a utilização de gestos naturais em ambientes colaborativos, permitindo os usuários trocarem experiências e informações em tempo real.

Dessa forma, este trabalho apresentou o desenvolvimento de uma aplicação computacional que provê ao usuário a possibilidade de manipular objetos 3D, por meio de gestos naturais, e compartilhar as novas posições dos objetos 3D com outros usuários de forma síncrona, sendo um sistema que contempla interação natural em conjunto com Ambientes Virtuais Colaborativos.

Para tal, foi utilizado o dispositivo Kinect®, juntamente com o *framework* Microsoft SDK, que são módulos de *software* para uso específico em aplicações com interação, via gestos. Já a troca de dados entre os usuários foi desenvolvida

utilizando bibliotecas de comunicação do Visual Studio 2010TM, a renderização da cena utilizou a interface WPF também parte do Visual Studio 2010TM.

Com relação à troca de dados entre usuários, o sistema não apresentou nenhuma lentidão no envio dos dados e na renderização da cena, mantendo a taxa de renderização sempre acima de 25 quadros por segundo, já que são enviados pacotes apenas com as posições de atualizações para os usuários, sendo assim, a perda de um pacote não afeta o desempenho do sistema.

Dessa maneira, considerou-se que os objetivos traçados foram cumpridos de forma satisfatória.

7.3 Trabalhos Futuros

A aplicação desenvolvida neste trabalho apresenta uma arquitetura para várias outras possibilidades de aplicações específicas, no que tange as áreas de interação natural e ambientes virtuais colaborativos. Como trabalhos futuros, pretende-se aplicar os conhecimentos e recursos desenvolvidos em aplicações específicas.

Sendo assim, pode-se trabalhar no sentido de mapear gestos e envia-los a um avatar gerado em um ambiente virtual ou equipamento no mundo real, fazendo com que usuários dispersos remotamente possam se comunicar e realizar tarefas em conjunto, para isso deve-se desenvolver técnicas de seleção e manipulação para que dois ou mais usuários possam realizar manipulação em um objeto ao mesmo tempo.

Outro ponto importante seria estudar e desenvolver gestos mais naturais para a manipulação de objetos 3D, como a utilização dos dedos para seleção e manipulação dos objetos 3D.

Esta proposta pode ser aplicada para treinamento em áreas como medicina, construção civil, ensino à distância, etc.

Por fim, outra proposta para trabalhos futuros é incluir comandos de voz para realização de determinadas tarefas, tornando a interação e a imersão dos usuários ainda mais natural.

Referências Bibliográficas

globo.com. ISSN Kinect entra para o Guinness como o aparelho eletrônico de comercialização mais rápida da história. Disponível em: <<http://oglobo.globo.com/tecnologia/kinect-entra-para-guinness-como-aparelho-eletronico-de-comercializacao-mais-rapida-da-historia-2902111>>. Acesso em: 17 set. 2012.

AAAS. Atlas of Science Literacy. **American Association for the Advancement of Science**, Washington, v. project 2061, 2001.

ARAÚJO, R. B. Especificação e análise de um sistema distribuído de realidade virtual. **Tese de Doutorado**, São Paulo, 1996.

ART, W. S.; SELKER, T. A look at human interaction with pervasive computers. **IBM Systems Journal**, v. 38, n. 4, p. 504 - 507, 1999.

ASUS. www.asus.no. **asus**, 2011. Disponível em: <<http://www.asus.no/News/0q7YpBa7YsXGwMYb/>>. Acesso em: 01 out. 2012.

ASUS. ASUS CHANGES THE LIVING ROOM EXPERIENCE WITH A LAPTOP. **primesense**, 2013. Disponível em: <<http://www.primesense.com/casestudies/xtion/>>. Acesso em: 20 Dezembro 2012.

AVANCINI, M. Using Kinect to emulate an Interactive Whiteboard. **UNIVERSITÀ DEGLISTUDI DI TRENTO - Dissertação de Mestrado**, 2011. 21 - 25.

BENFORD, et al. Collaborative Virtual Environments. **COMMUNICATIONS OF THE ACM**, v. 44, n. 7, p. 79 - 85, July 2001.

BJORKE, O. CodePlex. **Helix 3D Toolkit**, 2012. Disponível em: <<http://helixtoolkit.codeplex.com/>>. Acesso em: 27 Fevereiro 2013.

BOTEGA, L.; CUVINEL, P. Realidade Virtual: Histórico, Conceitos e Dispositivos. In: _____ **Aplicações de Realidade Virtual e Aumentada**. Porto Alegre: SVR , 2009. Cap. 1, p. 14 - 16.

BOULOS, M. N. et al. Web GIS in practice X: a Microsoft Kinect natural user interface for Google Earth navigation. **INTERNATIONAL JOURNAL OF HEALTH GEOGRAPHICS**, 2011.

BOWMAN, D. et al. 3D user interfaces: theory and practice, Boston, 2004.

BUCHMANN, E. Fing AR tips: gesture based direct manipulation in Augmented Reality. In: _____ **GRAPHITE' 04: Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Austral Asia and South East Asia**. New York: [s.n.], 2004. p. 212 - 221.

BURDEA, G.; COIFFET, P. **Virtual reality technology**. New York: John Wiley & Sons, 1994.

CAPIN, T. K.; THALMANN, D. A. A Taxonomy of Networked Virtual Environments. **IAA Humans in Space Symposium**, Santorini, Grécia, 1999.

CARDOSO, A.; LAMOUNIER, E. A Realidade Virtual na Educação e Treinamento. In: TORI, R.; KIRNER, C. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. Belém: SBC, 2006. Cap. 19, p. 304 - 312.

CARVALHO, A.; SOUSA, O.; OLIVEIRA,. Uso da Realidade Virtual e Interação Natural para Manipulação de Objetos. **Workshop de Realidade Virtual e Aumentada**, Paranaíba, 2012.

CASTRO, R. H. Desenvolvimento de Aplicações com uso de Interação Natural: Um Estudo de Caso voltado para Vídeo Colaboração em Saúde. **UFPB**, João Pessoa , Agosto 2012.

CENTER, M. N. Microsoft Releases Kinect for Windows SDK Beta for Academics and Enthusiasts. **Microsoft**, 2011. Disponível em: <<http://www.microsoft.com/en-us/news/press/2011/jun11/06-16MSKinectSDKPR.aspx>>. Acesso em: 20 Dezembro 2012.

CERTIFICATIONS, I. ICND1 Break Down — Broadcasts, Unicasts, And Multicasts. **IT Certifications**, 2012. Disponível em: <<http://kl2217.wordpress.com/2009/07/15/broadcasts-unicasts-and-multicasts/>>. Acesso em: 10 Dezembro 2012.

CHARYTONOWICZ, J. Tomorrow's ergonomics. **Proceedings of the Human Factors and Ergonomics Meeting**, v. 44, n. 33, p. 194 - 197, 2000.

COELHO, R.; COELHO, D.; CARDOSO, A. Use of Virtual Reality in process control in a mine in Brazil - a case study. **SVR**, Uberlândia, 2011.

COLLET, L.; SILVA, L. GESTUS: Hardware e Framework de Baixo Custo. **Workshop de realidade virtual e aumentada.**, Uberaba, 2011.

CORPORATION, M. **Kinect for Windows Software Development Kit**. Microsoft Corporation. [S.l.]. 2011.

EVAN, A. S.; BELINDA, L. FFAST: The Flexible Action and Articulated Skeleton Toolkit. **IEEE Virtual Reality**, Singapore, 19-23 March 2011. 245 - 246.

FIGUEIREDO, L.; SOUZA, P. Interação Natural a partir de Rastreamento de Mãos. **Tendências e Técnicas em Realidade Virtual e Aumentada**, Rio de Janeiro, 2012. 125 - 158.

FORUMPC. <http://www.forumpcs.com.br>. **Forum PC**, 2012. Disponível em: <<http://www.forumpcs.com.br/comunidade/viewtopic.php?t=265052>>. Acesso em: 01 out. 2012.

GLOBO. Globo. **Globo Economia Digital e mídia**, 2012. Disponível em: <<http://oglobo.globo.com/tecnologia/kinect-entra-para-guiness-como-aparelho-eletronico-de-comercializacao-mais-rapida-da-historia-2902111>>. Acesso em: 17 set. 2012.

GNECCO, B. B.; DIAS, R. C. Desenvolvimento de Interfaces Naturais de Interação usando o Hardware Kinect. **Tendências e Técnicas em Realidade Virtual e Aumentada**, Rio de Janeiro, 2012. 37 - 45.

GUAN, C. G. Volume based tumor neurosurgery planning in virtual workbench. **Proc. VRAIS IEEE**, 1998. 167 - 173.

IFIXIT. Ifix it. **http: //www.ifixit.com**, 2012. Disponível em: <<http://www.ifixit.com/Teardown/Microsoft-Kinect-Teardown/4066/3>>. Acesso em: 29 Outubro 2012.

INFOSTRAT. www.InfoStrat.com. **InfoStrat**, 2012. Disponível em: <<http://www.infostrat.com/solutions/kinectgis>>. Acesso em: 15 Outubro 2012.

JUNIOR, P. J. **Computação, Ubiquidade e Transparência**. UniAnchieta. Jundiaí, p. 16. 2011.

KERKHOVE, T. **Human Interface Guidelines**. [S.l.]: Books, General, 2012.

KIRNER, C.; TORI, R. Introdução à Realidade Virtual, Realidade Misturada e Hiper-realidade. In: KIRNER, C.; TORI, R. **Realidade Virtual: Conceitos, Tecnologia e Tendências**. 1ª. ed. São Paulo: SENAC, 2004. p. 3 - 20.

KIRNER, C.; TORI, R.; SISCOUTO, R. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. Porto Alegre: SBC, 2006.

LAVID. LAVID. **Artrhon**, 2012. Disponível em: <<http://www.lavid.ufpb.br/gtmda/artrhon.html>>. Acesso em: 15 Outubro 2012.

MACEDONIA, M.; ZYDA, M. A Taxonomy for Networked Virtual Environments. **IEEE Multimedia**, Los Alamitos, v. 4, n. 1, p. 48 - 56, Janeiro 1997.

MATSUMURA, K. K.; SONNINO, R. Trabalho de Conclusão de Curso - Escola Politécnica da Universidade de São Paulo. **Fusion 4D: interface natural e imersiva para manipulação de objetos 3D**, São Paulo, 2011. 109.

MICROSOFT. Microsoft MSDN. **Microsoft**, 2012. Disponível em: <[http://msdn.microsoft.com/pt-br/library/vstudio/6x6bk1f4\(v=vs.100\).aspx](http://msdn.microsoft.com/pt-br/library/vstudio/6x6bk1f4(v=vs.100).aspx)>. Acesso em: 20 Outubro 2012.

NATHAN, A. **Windows Presentation Foundation Sem Limites**. Rio de Janeiro: Alta Books, 2007.

NBR 5413, A. **Normas Brasileiras - Iluminância de interiores**. Rio de Janeiro: [s.n.], 1992.

NETTO, A. V.; MACHADO, L. S.; OLIVEIRA, M. C. **Realidade Virtual - Definições, Dispositivos e Aplicações**. São Paulo: USP, 2002.

NINTENDO , 2012. Disponível em: <<http://www.nintendo.com/wii>>. Acesso em: 01 out. 2012.

OVIATT, S. Multimodal interfaces. In: JACKO, J.; SEARS, A. **A Handbook of Human-Computer Interaction**. New Jersey: Lawrence Erlbaum and Associates, 2003. p. 286 - 304.

PAULA, B. C. D. SBC - Proceedings of SBGames 2011. **Adaptando e desenvolvendo jogos para uso com o Microsoft Kinect**, Salvador, 9 nov. 2011.

PINHO, M. S. Manipulação Simultânea de Objetos em Ambientes Virtuais Imersivos. **Tese de Doutorado em Ciências da Computação**, Porto Alegre, 2002. 23.

POLTROCK, S.; GRUDIN, J. CSCW, Groupware and Workflow: experiences, state of art, and future trends. **Conference on Human Factors in Computing Systems**, Pittsburgh: University of Pensilvânia, 1999.

PREECE, J. E. **Human-Computer Interaction**. England: Addison-Wesley, 1994.

REDMOND, W.; TEL AVIV, I. Kinect Sales Surpass Ten Million. **Xbox**, 2010. Disponível em: <<http://www.microsoft.com/en-us/news/press/2010/mar10/03-31PrimeSensePR.aspx>>. Acesso em: 29 Outubro 2012.

REYES, A. Evolution of the user interface. **Wikipedia**, 2009. Disponível em: <<http://en.wikipedia.org/wiki/File:CLI-GUI-NUI.png>>. Acesso em: 20 set. 2012.

RIBEIRO, M. W. **Arquitetura para distribuição de Ambientes Virtuais Multidisciplinares**. Uberlândia: UFU, 2005.

ROSS, J. **REDES DE COMPUTADORES**. 1. ed. [S.l.]: Antenna Edições Técnicas, 2008. 95 p.

SANTOS, E. S.; SILVA, M. Proposta de uma Interface não Convencional para Reabilitação Motora Suportada por Ambientes de Realidade Virtual. **6º Workshop de Realidade Virtual e Aumentada**, 2009.

SDK. **Kinect for Windows Programming Guide**. Microsoft. [S.l.]. 2011.

SEIFRIED, T.; JETTER, H. Lessons Learned from the Design and Implementation of Distributed Post-WIMP User Interfaces. **Distributed User Interfaces 2011**, 2011. 95-102.

SILVA, L. Ambientes distribuídos em Realidade Virtual como suporte a Aprendizagem Cooperativa para a Resolução de Problemas. **Tese de Doutorado**, Uberlândia, 2008.

SILVA, R. C.; LAMOUNIER, E.; CARDOSO, A. Virtual Substation. **WRVA**, Uberaba, 2011.

SNEATH, T. What does Windows Vista SP1 Mean for Developers? **MSDN**, 2008. Disponível em: <<http://blogs.msdn.com/b/tims/archive/2008/03/19/what-does-windows-vista-sp1-mean-for-developers.aspx>>. Acesso em: 20 Dezembro 2012.

STURMAN, D. J.; ZELTZER, D. A survey of glove-based input. **IEEE Computer Graphics**, Janeiro 1994. 30 - 39.

SUMMERVILLE, I. **Engenharia de Software**. 8^a. ed. São Paulo: Pearson Addison-Wesley, 2007.

TORI, R. et al. **Vida: atlas anatômico 3d interativo para treinamento a distância**., 2009. 1801 - 1810.

TORI, R.; KIRNER, C. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. Belém: SBC - Sociedade Brasileira de Computação, 2006.

VALLI, A. Notes on natural interaction, 2005.

WEBSTER, R. A prototype haptic suturing simulator. **Studies in Health Technology and Informatics**, v. 81, p. 567 - 569, 2001.

WEISER, M. Ubiquitous Computing. **IEEE Computer**, v. 4, n. 10, p. 71 - 72, October 1993.

Apêndice A

Avaliação do Sistema

Avaliador: _____

Idade: () 18 – 25 anos () 26 – 35 anos
 () 36 – 45 anos () 46 anos ou mais

Rotação de objetos 3D	Gesto 1	Gesto 2	Gesto 3	Gesto 4	Gesto 5
Reconheceu o gesto correto, em tempo real					
Reconheceu o gesto correto, porém com atraso					
Reconheceu o gesto errado					
Não reconheceu o gesto					

Escala de objetos 3D	Gesto 1	Gesto 2	Gesto 3	Gesto 4	Gesto 5
Reconheceu o gesto correto, em tempo real					
Reconheceu o gesto correto, porém com atraso					
Reconheceu o gesto errado					
Não reconheceu o gesto					

No que diz respeito a manipulação e visualização dos objetos 3D, entre os usuários (troca de dados), você se considera como em relação ao sistema:

() Satisfeito () Pouco Satisfeito
 () Parcialmente Satisfeito () Insatisfeito

Observações (opcional): _____
