

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

RAFAEL HUFF

**Recorte Volumétrico Usando Técnicas de
Interação 2D e 3D**

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

Prof^ª. Dra. Carla Maria Dal Sasso Freitas
Orientadora

Prof^ª. Dra. Luciana Porcher Nedel
Co-orientadora

Porto Alegre, julho de 2006

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Huff, Rafael

Recorte Volumétrico Usando Técnicas de Interação 2D e 3D /
Rafael Huff. – Porto Alegre: PPGC da UFRGS, 2006.

97 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2006. Orientadora: Carla Maria Dal Sasso Freitas; Co-orientadora: Luciana Porcher Nedel.

1. Técnicas de interação. 2. Visualização volumétrica baseada em texturas. 3. Recorte volumétrico interativo. 4. Visualização volumétrica em tempo real. I. Freitas, Carla Maria Dal Sasso. II. Nedel, Luciana Porcher. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Prof^ª. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Flávio Rech Wagner

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Gostaria de agradecer primeiramente à minha orientadora, Prof^a. Carla Maria Dal Sasso Freitas, e ao meu antigo orientador, Prof. Lisandro Zambenedetti Granville, pelo apoio, incentivo e amizade que sempre me deram durante a minha curta vida acadêmica. O meu especial agradecimento também à minha co-orientadora, Prof^a. Luciana Porcher Nedel, pelo auxílio durante a elaboração do presente trabalho, e também aos outros professores do grupo de Computação Gráfica, os professores João Luiz Dihl Comba e Manuel Menezes de Oliveira Neto.

Agradeço a todos os meus amigos e colegas do Instituto de Informática, principalmente aos meus amigos do antigo Labcom, Sala 228 e aos meus colegas do grupo de CG.

Sou grato principalmente à minha família pelo amor, compreensão e estímulo durante esses anos de prazos curtos e noites mal dormidas.

Sou grato também à CAPES pelo suporte financeiro durante o período de 24 meses, possibilitando a realização do presente trabalho.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	9
LISTA DE TABELAS	13
RESUMO	15
ABSTRACT	17
1 INTRODUÇÃO	19
1.1 Contexto e Motivação	21
1.2 Objetivo	22
1.3 Organização do texto	22
2 VISUALIZAÇÃO DIRETA DE VOLUMES	23
2.1 Visualização direta baseada em texturas	25
2.2 Exploração de volumes baseada em recorte volumétrico	27
2.3 Síntese do Capítulo	31
3 INTERAÇÃO	33
3.1 Técnicas de Interação	33
3.1.1 Taxonomia	34
3.1.2 Metáforas de Interação	35
3.2 Dispositivos de Entrada	42
3.2.1 Tipos de Dispositivos de Entrada	43
3.2.2 Modelos teóricos dos Dispositivos de Entrada	45
3.3 Utilização de diferentes dispositivos de entrada em ambientes virtuais 3D	46
3.4 Síntese do Capítulo	48
4 RECORTE VOLUMÉTRICO	49
4.1 Arquitetura do Sistema	49
4.2 Ferramentas de Recorte	52
4.3 Recorte Volumétrico Utilizando Apontadores Virtuais	54
4.3.1 Borracha 2D	54
4.3.2 Escavadeira 2D	56
4.3.3 Guilhotina 2D	57
4.4 Recorte Volumétrico Utilizando Mão Virtual	58
4.4.1 Borracha 3D	59

4.4.2	Escavadeira 3D	60
4.4.3	Guilhotina 3D	60
4.5	Síntese do Capítulo	61
5	AVALIAÇÃO EXPERIMENTAL DAS TÉCNICAS DE INTERAÇÃO . . .	63
5.1	Utilização das Ferramentas	63
5.2	Hipóteses	65
5.3	Cenário	65
5.4	Variáveis Dependentes e Independentes	65
5.5	Sujeitos	67
5.6	Metodologia e Procedimentos	67
5.7	Resultados e Discussão	68
5.7.1	Comparação de Experiência com Desempenho	68
5.7.2	Teste das Hipóteses	68
5.8	Síntese do Capítulo	70
6	AVALIAÇÃO EXPERIMENTAL DOS DISPOSITIVOS DE ENTRADA .	71
6.1	Utilização das Ferramentas	71
6.2	Hipóteses	73
6.3	Cenário	74
6.4	Variáveis Dependentes e Independentes	74
6.5	Sujeitos	75
6.6	Metodologia e Procedimentos	75
6.7	Resultados	76
6.7.1	Comparação de Experiência com Desempenho	76
6.7.2	Teste das Hipóteses	77
6.8	Síntese do Capítulo	78
7	CONCLUSÕES E TRABALHOS FUTUROS	81
	REFERÊNCIAS	83
APÊNDICE A	QUESTIONÁRIO PRÉ-TESTE PARA AVALIAÇÃO DAS TÉCNICAS DE INTERAÇÃO	91
APÊNDICE B	QUESTIONÁRIO PÓS-TESTE PARA AVALIAÇÃO DAS TÉCNICAS DE INTERAÇÃO	93
APÊNDICE C	QUESTIONÁRIO PRÉ-TESTE PARA AVALIAÇÃO DOS DISPOSITIVOS DE ENTRADA	95
APÊNDICE D	QUESTIONÁRIO PÓS-TESTE PARA AVALIAÇÃO DOS DISPOSITIVOS DE ENTRADA	97

LISTA DE ABREVIATURAS E SIGLAS

2D	Bidimensional
3D	Tridimensional
CPU	Unidade de Processamento Central (<i>Central Processing Unit</i>)
DOF	Graus de Liberdade (<i>Degrees of Freedom</i>)
FOB	Detector de Posição (<i>Flock of Birds</i>)
GLUI	<i>GLUT User Interface Library</i>
GLUT	<i>OpenGL Utility Toolkit</i>
GPU	Unidade de Processamento Gráfico (<i>Graphics Processing Unit</i>)
HMD	Capacete de Realidade Virtual (Head Mounted Display)
OpenGL	<i>Open Graphics Library</i>
PC	Computador Pessoal (<i>Personal Computer</i>)
RM	Ressonância Magnética
TC	Tomografia Computadorizada
VOI	Volume de Interesse (<i>Volume of Interest</i>)
WIM	<i>World in Miniature</i>
WSG	<i>Weighted Sweep Graph</i>

LISTA DE FIGURAS

Figura 2.1:	Técnica de <i>ray-casting</i> , onde a projeção é feita disparando-se raios do ponto de visão através de cada pixel da imagem 2D (plano de projeção) e passando pelo volume.	23
Figura 2.2:	Técnica de <i>splatting</i> , onde a projeção é feita “jogando” os voxels do volume no plano de projeção de trás para frente.	24
Figura 2.3:	Técnica de <i>shear-warp</i> : (a) O volume é transformado em um sistema de coordenadas intermediário através da translação de cada fatia para poder realizar a projeção paralela. (b) O volume é transformado em um sistema de coordenadas intermediário através da translação e escala de cada fatia para poder realizar a projeção perspectiva.	24
Figura 2.4:	Visualização direta por mapeamento de texturas; (a) volume de dados; (b) fatias; (c) mapeamento de fatias no espaço de textura; (d) <i>rendering</i> das fatias no frame buffer.	26
Figura 2.5:	Posições das fatias do volume (CULLIP; NEUMANN, 1994): (a) Planos alinhados a um eixo ortogonal do espaço de textura e (b) planos paralelos ao plano de projeção.	26
Figura 2.6:	Fatias da visualização direta por mapeamento de textura perpendiculares à direção de observação.	27
Figura 2.7:	Fatias da visualização direta por mapeamento de textura alinhadas com os planos cartesianos do objeto.	27
Figura 2.8:	Ferramenta de recorte de Van Gelder e Kim (GELDER; KIM, 1996) que utiliza seis planos de corte para eliminar regiões das fatias de amostragem fora do volume de dados.	28
Figura 2.9:	Ferramenta que utiliza o <i>stencil buffer</i> para fazer o recorte de volumes (WESTERMANN; ERTL, 1998).	28
Figura 2.10:	Ferramentas do Volume-Pro (PFISTER et al., 1999): (a) plano de corte com espessura e (b) ferramenta com três planos de corte.	29
Figura 2.11:	Recorte obtido pela especificação de intervalos nos raios da abordagem de visualização de volumes por <i>ray-casting</i> (MANSSOUR et al., 2002)	29
Figura 2.12:	Recorte baseado na profundidade dos voxels (WEISKOPF; ENGEL; ERTL, 2002)	30
Figura 2.13:	Recorte baseado em máscara de visibilidade(adaptada de(WEISKOPF; ENGEL; ERTL, 2002))	30
Figura 2.14:	Ferramenta de recorte utilizada para produzir ilustrações de volumes (adaptada de (BRUCKNER; GRÖLLER, 2005))	30

Figura 3.1:	Taxonomia das técnicas de seleção, manipulação e navegação em ambientes virtuais (BOWMAN; HODGES, 1999).	34
Figura 3.2:	Metáforas e técnicas de manipulação e seleção em ambientes virtuais	35
Figura 3.3:	Metáfora exocêntrica <i>World in Miniature</i> (STOAKLEY; CONWAY; PAUSCH, 1995): (a) modelo em miniatura do ambiente; (b) usuário manipulando o modelo.	36
Figura 3.4:	Metáfora exocêntrica <i>Automatic Scaling</i>	36
Figura 3.5:	Metáforas de interação: (a) mão virtual e (b) apontador virtual	37
Figura 3.6:	Técnica de interação da mão virtual clássica utilizada para mudar de posição as peças de um tabuleiro de xadrez. (NEDEL et al., 2003) . .	37
Figura 3.7:	Técnica de interação <i>Go-go</i> (POUPYREV et al., 1996): (a) a mão do usuário tem alcance mais longo que seu braço; (b) mapeamento não linear do braço real e virtual.	38
Figura 3.8:	Técnica de interação <i>Fast Go-go</i> : sem mapeamento linear próximo ao usuário e com função de crescimento mais rápida que a <i>Go-go</i> (adaptada de (POUPYREV et al., 1996)).	38
Figura 3.9:	Técnicas de interação que utilizam regiões concêntricas ao redor do usuário para aumentar o tamanho do seu braço virtual: (a) <i>Stretch Gogo</i> (BOWMAN; HODGES, 1997): (b) <i>Flying Speed</i> (MINE, 1995)	39
Figura 3.10:	Técnica de interação <i>ray-casting</i> . Um raio que parte da mão do usuário é utilizado como apontador.	39
Figura 3.11:	Técnica de interação de <i>Ray-casting with fishing reel</i> (BOWMAN; HODGES, 1997). Os objetos selecionados por <i>ray-casting</i> são aproximados do usuário para serem manipulados.	40
Figura 3.12:	Técnica de interação <i>Spotlight</i> (MINE; FREDERICK P. BROOKS; SEQUIN, 1997). Um cone é utilizado para fazer a seleção dos objetos.	40
Figura 3.13:	Técnica de interação <i>Aperture</i> (FORSBERG; HERNDON; ZELENIK, 1996). O cone de seleção é alterado pela distância em que se encontra o círculo de abertura em relação ao ponto de visão.	41
Figura 3.14:	Técnica de interação <i>Image Plane</i> (PIERCE et al., 1997) onde o usuário pode selecionar um objeto apontando p/ ele (a), utilizando dois dedos (b), utilizando a palma da mão (c), ou utilizando as duas mãos (d).	41
Figura 3.15:	Técnica de interação <i>Eye Gaze</i> (JACOB, 1990): (a) todo o ambiente; (b) parte dos objetos ainda não selecionados; (c) o objeto magenta é selecionado e seus detalhes internos são revelados.	41
Figura 3.16:	Técnica de interação <i>Eyescop</i> e (UGHINI et al., 2006): (a) mapeamento da distância dos objetos do HMD e do usuário (com zoom); (b) ambiente de avaliação	42
Figura 3.17:	Técnica de interação <i>Homer</i> (BOWMAN; HODGES, 1997). Primeiramente, é utilizado <i>ray-casting</i> para a seleção do objeto e, após, é utilizada a mão virtual para manipulação.	42
Figura 3.18:	Dispositivos 2D mais comuns	43
Figura 3.19:	Dispositivos de entrada 3D: (a) <i>Spaceball</i> e (b) <i>Spacemouse</i>	43
Figura 3.20:	Rastreadores de posição: (a) mecânico, (b) eletromagnético, (c) acústico, (d) óptico, (e) inercial, e (f) rastreador de olhar.	44
Figura 3.21:	Dispositivo táctil CyberGrasp da Immersion	44

Figura 3.22:	Dispositivos para reconhecimento de gestos: (a) CyberGlove da Virtual Technologies e (b) PowerGlove da Mattel	45
Figura 3.23:	Dispositivo para reconhecimento de voz (NASA)	45
Figura 3.24:	Dispositivos avaliados por Hinckley (HINCKLEY et al., 1997) para comparar técnicas de interação 2D e 3D.	47
Figura 4.1:	Arquitetura do sistema. Primeiro, a fase de atualização é utilizada para definir um volume de seleção (<i>SculptMap</i>) baseado na ferramenta atual de recorte. A última fase corresponde ao passo de desenho que gera a visualização final ao mesmo tempo que descarta as regiões definidas na máscara <i>SculptMap</i>	50
Figura 4.2:	Programas de fragmentos <i>FSCCombiner</i> : (a) Abordagem com máscara de visibilidade binária; (b) Abordagem com máscara de visibilidade contendo distâncias euclidianas.	51
Figura 4.3:	Exemplos dos valores das distâncias euclidianas armazenados na máscara de visibilidade.	51
Figura 4.4:	Exemplo prático dos passos executados para recortar a metade de um volume contendo uma esfera vermelha.	52
Figura 4.5:	Elipse formada através de algumas interações com um círculo	53
Figura 4.6:	Ferramentas de recorte 2D: (a) Borracha 2D, (b) Escavadeira 2D e (c) Guilhotina 2D.	53
Figura 4.7:	Ferramentas de Recorte 3D: (a) Borracha 3D, (b) Escavadeira 3D e (c) Guilhotina 3D	53
Figura 4.8:	Técnicas com volumes de seleção cônicos: (a) <i>Aperture Selection</i> ; (b) variação proposta.	54
Figura 4.9:	Borracha 2D baseada na projeção de voxels no plano de visão	55
Figura 4.10:	Programas de fragmentos das ferramentas borrachas: (a) Abordagem com máscara de visibilidade binária; (b) Abordagem com máscara de visibilidade contendo distâncias euclidianas.	55
Figura 4.11:	Organização das fatias da textura 3D na textura 2D intermediária, onde cada “quadrado” das figuras corresponde a um voxel (DIETRICH et al., 2004).	56
Figura 4.12:	Escavadeira 2D baseada na distância euclidiana 2D	57
Figura 4.13:	Programas de fragmentos das ferramentas escavadeiras: (a) Abordagem com máscara de visibilidade binária; (b) Abordagem com máscara de visibilidade contendo distâncias euclidianas.	57
Figura 4.14:	Guilhotina 2D baseado na distância de voxels a um plano de recorte .	58
Figura 4.15:	Programas de fragmentos das ferramentas guilhotinas: (a) Abordagem com máscara de visibilidade binária; (b) Abordagem com máscara de visibilidade contendo distâncias euclidianas.	58
Figura 4.16:	Borracha 3D baseada na projeção de voxels em um plano de projeção	59
Figura 4.17:	Escavadeira 3D baseada na distância euclidiana 3D	60
Figura 4.18:	Guilhotina 3D baseado na distância de voxels a um plano de recorte .	61
Figura 4.19:	Volume cortado três vezes com a ferramenta Guilhotina	61
Figura 5.1:	Interpretação dos movimentos do mouse para transladar as ferramentas. A rotação do volume é feita pressionando-se o botão direito. As ferramentas são aplicadas no volume para fazer o recorte pressionando-se o botão esquerdo.	64

Figura 5.2:	Interface do aplicativo de avaliação das técnicas de interação das ferramentas de recorte volumétrico.	64
Figura 5.3:	Volumes de dados para avaliação das técnicas de interação das ferramentas de recorte: (a) um cubo e um cilindro para a fase de treinamento; (b) um <i>teapot</i> para avaliação das ferramentas <i>Borrachas</i> ; (c) um cogumelo para avaliação das ferramentas <i>Escavadeiras</i> ; (d) um cubo para avaliação das ferramentas <i>Guilhotinas</i> ; e uma maçã mordida para a avaliação das ferramentas simultaneamente.	66
Figura 5.4:	Tempo médio para a execução da tarefa onde o usuário podia usar todas as ferramentas de uma mesma técnica de interação ao mesmo tempo.	69
Figura 5.5:	Taxa de erros nas tarefas onde somente as ferramentas de recorte volumétrico Borracha ou Guilhotina podiam ser utilizadas.	69
Figura 6.1:	Dispositivos utilizados na avaliação experimental: (a) spacemouse e (b) mouse convencional.	72
Figura 6.2:	Interação bimanual com mouse 3D e 2D.	73
Figura 6.3:	Interface do aplicativo de avaliação de dispositivos para o recorte volumétrico.	73
Figura 6.4:	Conjunto de dados do experimento: (a) volume de avaliação da Borracha 3D, (b) volume de avaliação da Escavadeira 3D, (c) volume de avaliação da Guilhotina 3D e (d) volume para treinamento.	74
Figura 6.5:	Tempo médio das ferramentas (em segundos) quando o volume é controlado pelo mouse 2D	77

LISTA DE TABELAS

Tabela 5.1:	Número total de tarefas para avaliar as ferramentas independentemente e juntas utilizando técnicas de interação 2D e 3D.	66
Tabela 5.2:	Experiência anterior dos usuários com aplicativos e dispositivos 2D e 3D	68
Tabela 5.3:	Preferência dos usuários em relação às técnicas de interação empregadas nas ferramentas de recorte volumétrico.	70
Tabela 6.1:	Número total de tarefas decorrentes da combinação de 3 ferramentas de recorte (Borracha 3D, Escavadeira 3D, e Guilhotina 3D) e 2 dispositivos (mouse 2D e mouse 3D) para a manipulação de 2 objetos (volume e ferramentas).	75
Tabela 6.2:	Experiência anterior dos usuários com aplicativos e dispositivos 2D e 3D	77
Tabela 6.3:	Dispositivos preferidos para controlar o volume e as ferramentas . . .	78
Tabela 6.4:	Preferência de combinação de dispositivos	78

RESUMO

A visualização de conjuntos de dados volumétricos é comum em diversas áreas de aplicação e há já alguns anos os diversos aspectos envolvidos nessas técnicas vêm sendo pesquisados. No entanto, apesar dos avanços das técnicas de visualização de volumes, a interação com grandes volumes de dados ainda apresenta desafios devido a questões de percepção (ou isolamento) de estruturas internas e desempenho computacional. O suporte do hardware gráfico para visualização baseada em texturas permite o desenvolvimento de técnicas eficientes de rendering que podem ser combinadas com ferramentas de recorte interativas para possibilitar a inspeção de conjuntos de dados tridimensionais. Muitos estudos abordam a otimização do desempenho de ferramentas de recorte, mas muito poucos tratam das metáforas de interação utilizadas por essas ferramentas.

O objetivo deste trabalho é desenvolver ferramentas interativas, intuitivas e fáceis de usar para o recorte de imagens volumétricas. Inicialmente, é apresentado um estudo sobre as principais técnicas de visualização direta de volumes e como é feita a exploração desses volumes utilizando-se recorte volumétrico. Nesse estudo é identificada a solução que melhor se enquadra no presente trabalho para garantir a interatividade necessária. Após, são apresentadas diversas técnicas de interação existentes, suas metáforas e taxonomias, para determinar as possíveis técnicas de interação mais fáceis de serem utilizadas por ferramentas de recorte.

A partir desse embasamento, este trabalho apresenta o desenvolvimento de três ferramentas de recorte genéricas implementadas usando-se duas metáforas de interação distintas que são frequentemente utilizadas por usuários de aplicativos 3D: apontador virtual e mão virtual. A taxa de interação dessas ferramentas é obtida através de programas de fragmentos especiais executados diretamente no hardware gráfico. Estes programas especificam regiões dentro do volume a serem descartadas durante o rendering, com base em predicados geométricos.

Primeiramente, o desempenho, precisão e preferência (por parte dos usuários) das ferramentas de recorte volumétrico são avaliados para comparar as metáforas de interação empregadas. Após, é avaliada a interação utilizando-se diferentes dispositivos de entrada para a manipulação do volume e ferramentas. A utilização das duas mãos ao mesmo tempo para essa manipulação também é testada. Os resultados destes experimentos de avaliação são apresentados e discutidos.

Palavras-chave: Técnicas de interação, visualização volumétrica baseada em texturas, recorte volumétrico interativo, visualização volumétrica em tempo real.

Volume Sculpting with 2D and 3D Interaction Techniques

ABSTRACT

Visualization of volumetric datasets is common in many fields and has been an active area of research in the past two decades. In spite of developments in volume visualization techniques, interacting with large datasets still demands research efforts due to perceptual and performance issues. The support of graphics hardware for texture-based visualization allows efficient implementation of rendering techniques that can be combined with interactive sculpting tools to enable interactive inspection of 3D datasets. Many studies regarding performance optimization of sculpting tools have been reported, but very few are concerned with the interaction techniques employed.

The purpose of this work is the development of interactive, intuitive, and easy-to-use sculpting tools. Initially, a review of the main techniques for direct volume visualization and sculpting is presented. The best solution that guarantees the required interaction is highlighted. Afterwards, in order to identify the most user-friendly interaction technique for volume sculpting, several interaction techniques, metaphors and taxonomies are presented.

Based on that, this work presents the development of three generic sculpting tools implemented using two different interaction metaphors, which are often used by users of 3D applications: *virtual pointer* and *virtual hand*. Interactive rates for these sculpting tools are obtained by running special fragment programs on the graphics hardware which specify regions within the volume to be discarded from rendering based on geometric predicates.

After development, the performance, precision and user preference of the sculpting tools were evaluated to compare the interaction metaphors. Afterward, the tools were evaluated by comparing the use of a 3D mouse against a conventional wheel mouse for guiding volume and tools manipulation. Two-handed input was also tested with both types of mouse. The results from the evaluation experiments are presented and discussed.

Keywords: Interaction techniques; texture-based volume visualization; interactive volume clipping; real-time volume rendering.

1 INTRODUÇÃO

A visualização de imagens tridimensionais (3D), ou dados volumétricos, é uma tarefa comum em várias áreas, como em imagens médicas, análise meteorológica e exploração de petróleo. Por exemplo, imagens obtidas através de tomografia computadorizada (TC), ressonância magnética (RM) e ultra-som são amplamente utilizadas para ajudar no diagnóstico clínico, no planejamento cirúrgico e no acompanhamento do tratamento de pacientes. Apesar da evolução de técnicas de *rendering* de volumes, a avaliação de cada fatia bidimensional (2D), que corresponde a um corte ou projeção dos dados volumétricos ainda é vastamente utilizada na prática devido a várias razões. Primeiramente, os dados volumétricos podem ser muito grandes e os algoritmos de *rendering* talvez não sejam suficientemente rápidos para suportar sistemas em tempo real. É necessária também a segmentação do volume para inspeção de estruturas internas, o que normalmente não é uma tarefa trivial, pois requer a especificação de funções de transferência ou o uso de ferramentas de recorte, por exemplo. Finalmente, a maioria das pessoas são familiarizadas com interação 2D, o que pode ser facilmente realizado com um mouse convencional ou teclado disponíveis em praticamente qualquer computador. No entanto, a compreensão do volume a partir de imagens bidimensionais não é uma tarefa trivial, e apenas profissionais experientes são capazes de interpretar estas imagens com segurança.

Uma forma mais intuitiva para a visualização de imagens volumétricas é apresentar ao usuário um sistema que, além de ser capaz de reproduzir essas imagens de forma tridimensional, permita a manipulação interativa desse volume. Existem duas abordagens básicas para a solução de problemas de visualização volumétrica (BRODLIE; WOOD, 2001): visualização por superfícies e visualização direta. Na abordagem de visualização por superfícies, os dados volumétricos são convertidos para uma representação geométrica, normalmente malhas de triângulos, a partir da qual é utilizado o *pipeline* tradicional da Computação Gráfica para a geração da visualização. Esta conversão dos dados volumétricos normalmente reduz a quantidade de dados manipulados na formação da imagem, melhorando conseqüentemente o desempenho do sistema. A desvantagem dessa redução, entretanto, é a perda de informações devido à simplificação dos dados. Um exemplo desse tipo de abordagem é a conhecida técnica de *Marching Cubes* (LORENSEN; CLINE, 1987). Já nas técnicas de visualização direta de volumes, a projeção é realizada diretamente a partir dos dados volumétricos, sem passar por nenhuma representação intermediária. Como toda a informação contida no volume é considerada na composição da nova imagem, este método reduz a perda de informação se comparado à visualização de superfícies. Entretanto, esta abordagem geralmente envolve técnicas de que necessitam de um considerável esforço computacional para sua realização em tempo real, como *ray-casting*, por exemplo (LEVOY, 1990; LACROUTE; LEVOY, 1994; WEILER et al., 2003).

As técnicas de texturização tridimensional disponibilizadas pelos avanços atuais no hardware gráfico dos computadores pessoais também podem ser utilizadas na visualização direta de volumes como uma opção ao método de *ray-casting*. Nestas técnicas, o conjunto de imagens adquiridas por tomografia computadorizada ou ressonância magnética pode ser utilizado para a construção de uma textura tridimensional, que é amostrada por uma série de polígonos paralelos. Esta técnica permite a construção de projeções bidimensionais da imagem volumétrica a partir de direções arbitrárias de visualização. Vários pesquisadores (CABRAL; CAM; FORAN, 1994; KAUFMAN et al., 2000; REZK-SALAMA et al., 2000; ENGEL; KRAUS; ERTL, 2001) apresentaram, nos últimos anos, diversas propostas que envolvem a utilização de texturas 3D para a visualização de imagens volumétricas através de visualização direta. As técnicas de Bullit e Aylward (BULLITT; AYLWARD, 2001) e Engel *et al.* (ENGEL; KRAUS; ERTL, 2001), em específico, exploraram a visualização de estruturas anatômicas através do hardware gráfico dos computadores comuns. Com isso, é possível haver visualização direta em tempo real e potencialmente sem perdas. Central a esse tipo de técnica de visualização é a necessidade de ser informado ao algoritmo o mapeamento entre os valores dos elementos individuais no volume (voxels) e os valores de cores e opacidade a serem adotados na visualização. A esse mapeamento dá-se o nome de função de transferência, e as ferramentas existentes para manipulação interativa de imagens visualizadas por visualização direta são geralmente restritas à especificação de parâmetros de classificação dos voxels, na forma dessas funções de transferência. Estas funções fazem somente o realce dos objetos da imagem, através do mapeamento dos valores de intensidade dos voxels em valores de opacidades, e não são capazes de tratar da oclusão dos objetos de interesse causada pelos objetos circundantes, além de serem extremamente prejudicadas pelo ruído na imagem de entrada. Outro aspecto essencial, embora mais comum e inerente às técnicas de visualização, é a determinação dos parâmetros geométricos para a construção da imagem, ou seja, posição e orientação do observador e características de iluminação, por exemplo.

No que se refere à oclusão das regiões de interesse, uma forma genérica de resolver este problema é através do uso de ferramentas de recorte. No presente trabalho, o termo recorte (ou corte) significa a eliminação de voxels que não devem ser visualizados. Manssour (MANSSOUR et al., 2002) apresentou a formalização do comportamento de ferramentas de recorte em imagens 3D multimodais, considerando o recorte baseado tanto em geometria como em conteúdo, ou seja, considerando tanto a eliminação como a manutenção de regiões definidas geometricamente ou através de valores. Já Weiskopf *et al.* (WEISKOPF; ENGEL; ERTL, 2002) propõem novas funções de recorte volumétrico, explorando as características presentes no hardware gráfico atual de computadores pessoais. No trabalho de Weiskopf *et al.*, os métodos propostos envolvem o processamento de toda a imagem 3D na definição do volume de interesse (*volume of interest*, VOI), e portanto necessita uma grande quantidade de memória durante sua execução (principalmente no recorte baseado em texturas 3D). No trabalho de Manssour, ao contrário da proposta de Weiskopf *et al.*, não há grande necessidade de memória para o armazenamento do polígono de recorte, mas sem um hardware gráfico que suporte as técnicas de recorte sugeridas, não é possível alcançar as taxas de *frame-rate* necessárias à manipulação interativa.

A maioria dos sistemas de recorte volumétricos apresentados na literatura, para garantir a interatividade de suas ferramentas, aborda somente o desempenho dessas ferramentas em termos computacionais (*frame-rate*). Normalmente não é feito um estudo mais detalhado sobre as técnicas de interação em si empregadas. Entretanto, estas técnicas de

interação e os dispositivos empregados afetam diretamente o desempenho dos usuários durante a execução de determinadas tarefas (MACKENZIE, 1995). Em ambientes virtuais, as tarefas realizadas podem ser classificadas como um conjunto de ações básicas de interação. Essas ações são: navegação, seleção e manipulação. No presente trabalho, é assumido que o recorte de volumes visualizados com mapeamento de textura é basicamente uma operação de seleção, isto é, as ferramentas de recorte fazem a seleção dos elementos do volume (voxels) a serem eliminados primeiramente, para após removê-los. As inúmeras técnicas de seleção em ambientes virtuais podem ser classificadas por duas metáforas básicas: apontadores virtuais e mão virtual. Em apontadores virtuais, um apontador, ou raio, é utilizado para apontar os objetos a serem selecionados, e na metáfora da mão virtual, essa seleção é feita através de uma representação virtual da mão real do usuário. Existem várias extensões e adaptações dessas metáforas, mas as técnicas baseadas em apontadores virtuais caracterizam-se por serem técnicas de interação 2D, enquanto que as técnicas que utilizam mão virtual são técnicas de interação 3D.

1.1 Contexto e Motivação

Na visualização de dados volumétricos de imagens médicas (ressonância magnética e tomografia computadorizada), freqüentemente é necessária a manipulação do volume de dados para melhorar, ou mesmo permitir, dependendo da situação, a análise das estruturas de interesse. A remoção de estruturas circundantes para solucionar o problema de oclusão dos objetos de interesse é importante para o entendimento de sua anatomia. Neste caso, uma ferramenta de recorte para ser aplicada na segmentação interativa do volume de interesse, como proposto em (DIETRICH et al., 2004), é essencial. Neste trabalho é apresentado um conjunto de ferramentas para a realização da segmentação do VOI, através de sucessivas operações de remoção de partes indesejáveis, esculpindo interativamente um volume de interesse. Neste caso, é necessário o conhecimento da estrutura pelo usuário, uma vez que a segmentação não é automática. As ferramentas apresentadas por Dietrich, baseadas no trabalho de Weiskopf (WEISKOPF; ENGEL; ERTL, 2002), exploram as capacidades recentes das placas gráficas de computadores pessoais para prover a visualização e a interação com volumes de dados. O controle programável dos cálculos realizados pelo hardware gráfico para gerar a aparência de cada pixel na tela é usado para desenvolver ferramentas de manipulação de volumes em tempo real. Apesar deste trabalho fornecer uma avaliação do desempenho das ferramentas desenvolvidas aplicadas ao volume visualizado, ele não apresenta um estudo mais detalhado sobre a usabilidade das técnicas de interação empregadas. Estas ferramentas são baseadas em conhecidas técnicas de interação 2D utilizando um mouse convencional e o teclado para manipulá-las, ao passo que em aplicações médicas, considera-se cada vez mais o uso de técnicas de realidade virtual, devido ao fato da origem do problema ser 3D.

O desempenho de ferramentas de recorte volumétrico vem sendo amplamente estudado para garantir altas taxas de interatividade, mas as técnicas de interação empregadas são pouco pesquisadas. O design das técnicas de interação influencia diretamente a qualidade da interface provida ao usuário (BOWMAN et al., 2001), e esta qualidade pode ser avaliada através de um teste de bancada (*testbed evaluation*) (BOWMAN; JOHNSON; HODGES, 2001).

1.2 Objetivo

O objetivo geral do presente trabalho é a investigação de técnicas interativas 2D e 3D de visualização e recorte de imagens volumétricas em tempo real utilizando computadores pessoais comuns. Estas técnicas devem permitir a interação com estruturas contidas em volumes visualizados por visualização direta e a seleção de VOI pela remoção de estruturas circundantes através de uma interface intuitiva e prática. Embora estas técnicas devam ser genéricas, elas são motivadas por aplicações médicas, onde o recorte de volumes é muito importante para a seleção de um VOI para visualização.

Como objetivos específicos está o desenvolvimento e a avaliação de ferramentas que implementem estas técnicas, utilizando o núcleo de funções de visualização e de recorte proposto por Dietrich (DIETRICH et al., 2004). Estas ferramentas devem conduzir o usuário na definição do recorte, baseado, por exemplo, em características evidenciadas na imagem. Também deve ser avaliado o impacto de diferentes dispositivos de entrada na manipulação das ferramentas utilizadas.

1.3 Organização do texto

Este trabalho apresenta o desenvolvimento e avaliação de técnicas de interação com ferramentas de recorte aplicadas a dados volumétricos. No Capítulo 2 são discutidos os principais conceitos sobre visualização direta de imagens 3D relacionados a este estudo. Também são brevemente revisados os trabalhos relacionados à exploração de volumes baseada em recorte.

O Capítulo 3 apresenta um estudo sobre interação e percepção em ambientes virtuais. São abordadas técnicas de interação e sua classificação em metáforas e taxonomias. Os dispositivos de entrada empregados e a forma como interagem com os sistemas também são apresentados.

No Capítulo 4, baseadas nas técnicas de interação descritas no Capítulo 3, diferentes técnicas de interação são propostas para ferramentas de recorte volumétrico que utilizam a visualização direta de imagens descrita no Capítulo 2.

O Capítulo 5 mostra como foi feita a avaliação de diferentes metáforas de interação em um aplicativo utilizando as ferramentas de recorte apresentadas no Capítulo 4. Após, no Capítulo 6, é feita a avaliação da interação com diferentes dispositivos de entrada utilizando-se as mesmas ferramentas de recorte com uma determinada metáfora de interação.

Finalmente, no Capítulo 7 são feitas as conclusões e considerações finais a respeito do trabalho apresentado. A continuação deste trabalho e outros trabalhos futuros, também são apresentados ao final do texto.

2 VISUALIZAÇÃO DIRETA DE VOLUMES

Visualização direta de volumes envolve o conjunto de técnicas que permite visualizar dados volumétricos sem passar por nenhuma representação intermediária como é o caso de visualização por *iso-superfícies* (LORENSEN; CLINE, 1987). O volume é representado através de voxels que são projetados diretamente em pixels (2D) e armazenados como uma imagem, dispensando o uso de primitivas geométricas. Neste processo, é feito um mapeamento dos valores dos voxels em valores de cor e opacidade, mapeamento este chamado de função de transferência (KINDLMANN; DURKIN, 1998). Desde a sua introdução por Levoy (LEVOY, 1988), a visualização direta de volumes tem sido realizada por diferentes técnicas que podem ser fracamente classificadas como *baseadas em imagem* (ou *backward projection*) e *baseadas em objeto* (ou *forward projection*) (ENGEL; KRAUS; ERTL, 2001). Nas técnicas de visualização volumétrica baseadas em imagem o *rendering* do volume é feito lançando-se raios dos pixels do plano de projeção em direção ao volume, e as contribuições dos voxels ao longo de cada raio são combinadas para formar a cor final do pixel associado ao raio (Figura 2.1). Nas técnicas baseadas em objeto, o *rendering* do volume é feito ao contrário, isto é, a contribuição de cada voxel é computada através da projeção do voxel no plano de projeção (Figura 2.2). Entre as técnicas de visualização de volumes baseadas em imagem, destaca-se a técnica de *ray-casting* (LEVOY, 1988). Nesta técnica, a visualização do volume é feita através da projeção 2D do conjunto de dados volumétricos 3D discretizados. A projeção é feita disparando-se raios do centro de projeção, chamado ponto de visão, através de cada pixel da imagem 2D de saída e passando pelo volume (Figura 2.1). Após, a contribuição dos elementos (voxels) ao longo de cada raio são combinadas de acordo com uma função de transferência para formar a cor final do pixel associado ao raio.

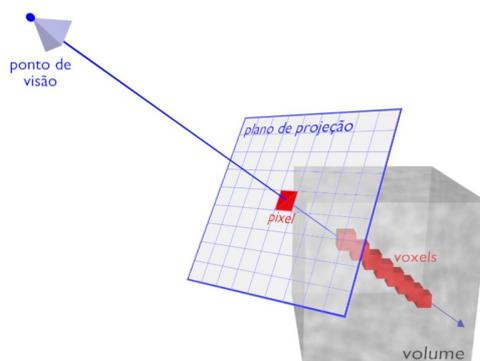


Figura 2.1: Técnica de *ray-casting*, onde a projeção é feita disparando-se raios do ponto de visão através de cada pixel da imagem 2D (plano de projeção) e passando pelo volume.

Entre as técnicas de visualização de volumes baseadas em objeto, destacam-se os algoritmos de *splatting* (WESTOVER, 1990), *shear-warp* (LACROUTE; LEVOY, 1994), e os baseados em textura (CULLIP; NEUMANN, 1994; CABRAL; CAM; FORAN, 1994). Na técnica de *splatting*, cada voxel do volume é “jogado” na imagem de projeção no formato de um disco, por exemplo (Figura 2.2). Esse processo é feito primeiramente partindo dos voxels que se encontram mais atrás no volume até os da frente. A área atingida por cada voxel na imagem de projeção pode ser distribuída de forma gaussiana, mas varia de acordo com a aplicação.

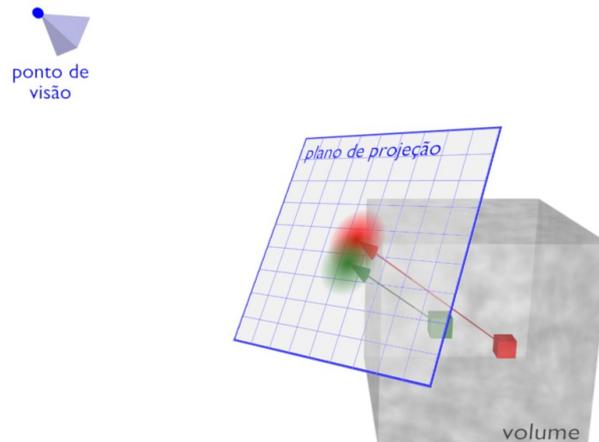


Figura 2.2: Técnica de *splatting*, onde a projeção é feita “jogando” os voxels do volume no plano de projeção de trás para frente.

Quando é feita a manipulação do volume (rotações), normalmente as suas fatias não se encontram paralelas a imagem de projeção e o mapeamento dos voxels do volume em pixels da imagem aumenta o tempo de computação. Para corrigir esse problema, no algoritmo de *shear-warp* (Figura 2.3), primeiramente é feita uma transformação do volume para um sistema de coordenadas intermediário através de uma operação de “*shear*”. Após essa operação as fatias do volume encontram-se paralelas ao plano de imagem e perpendiculares aos raios. A projeção do voxel resulta em uma imagem intermediária distorcida que é corrigida posteriormente através da operação de “*warping*”.

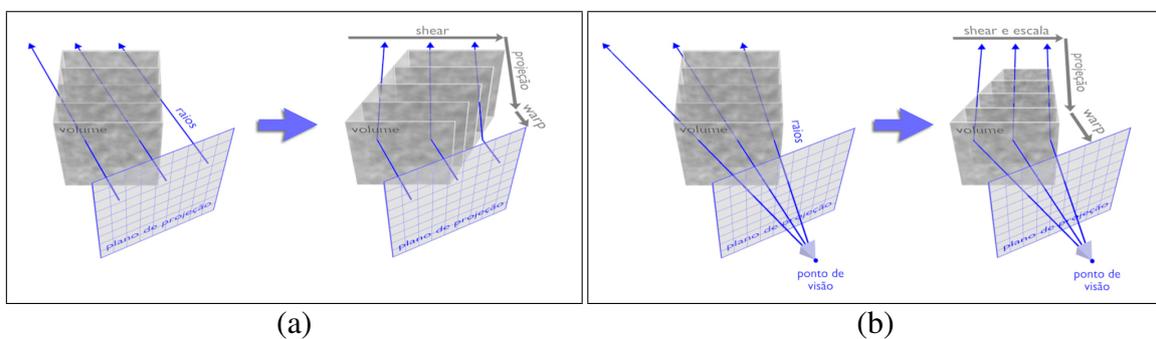


Figura 2.3: Técnica de *shear-warp*: (a) O volume é transformado em um sistema de coordenadas intermediário através da translação de cada fatia para poder realizar a projeção paralela. (b) O volume é transformado em um sistema de coordenadas intermediário através da translação e escala de cada fatia para poder realizar a projeção perspectiva.

A técnica de visualização de volumes baseada no mapeamento de textura é a mais rápida das anteriores apresentadas, mas só é possível através do uso de hardware especial que lê os pixels da memória e os mapeia para a superfície de objetos. Como a partir da terceira geração de GPUs (GeForce3 e GeForce4 da nVIDIA[®], Xbox da Microsoft[®] e Radeon 8500 da ATI[®]), o mapeamento de texturas 3D está incorporado no hardware gráfico, e é uma das principais linhas de pesquisa de visualização volumétrica, esta técnica foi utilizada nas ferramentas de recorte apresentadas nesse trabalho e é descrita detalhadamente na Seção 2.1.

2.1 Visualização direta baseada em texturas

O suporte do hardware gráfico para a aceleração do processo de *rendering* vem motivando a visualização direta de volumes baseada em textura. Akeley (AKELEY, 1993), ao apresentar o Reality Engine da Silicon Graphics, mencionou a possibilidade de utilização do hardware de mapeamento de textura para realizar o *rendering* de volumes. Posteriormente, Cullip e Neumann (CULLIP; NEUMANN, 1994) discutiram a amostragem do volume, representado como uma textura 3D ou uma pilha de texturas 2D, em superfícies alinhadas à direção de visualização constituindo a *proxy-geometry*. Após, o desenvolvimento e a extensão desta técnica, utilizando imagens médicas, foi descrito por Cabral *et al.* (CABRAL; CAM; FORAN, 1994). Esse trabalho demonstrou ser possível a reconstrução e o *rendering* interativo de volumes com hardware contendo aceleração de textura 3D. Essa abordagem foi expandida para combinar aceleração do hardware com iluminação e sombreado de volumes (GELDER; KIM, 1996; WESTERMANN; ERTL, 1998), e iluminação difusa para visualização de volumes semi-transparentes (MEISSNER; HOFFMANN; STRASSER, 1999).

O avanço do hardware gráfico permitiu a visualização direta baseada em textura com imagens de qualidade satisfatória e melhor desempenho que as abordagens baseadas em software (REZK-SALAMA *et al.*, 2000; KNISS; KINDLMANN; HANSEN, 2001). A partir de 2001, as placas gráficas dos computadores pessoais (*Graphics Processing Unit*, GPU) passaram a incorporar o mapeamento de texturas 3D. Alguns autores, motivados pelo ganho de desempenho e cálculo de fragmentos programável, passaram a explorar as capacidades da GPU para a visualização direta (ENGEL; KRAUS; ERTL, 2001; KRÜGER; WESTERMANN, 2003; WEISKOPF; ENGEL; ERTL, 2002). Devido a essas características, a técnica de mapeamento de textura é hoje uma das principais linhas de pesquisa em visualização volumétrica direta.

A idéia básica da visualização direta por mapeamento de textura é amostrar o volume com primitivas geométricas, normalmente planos paralelos à imagem de projeção, e “empilhá-los” ao longo da direção de observação. O *rendering* desses planos é feito como polígonos recortados (fatias) nos limites da textura do volume. Os dados do volume são aplicados como texturas nessas fatias e as imagens resultantes são combinadas de trás para frente em direção à posição de observação (Figura 2.4). Os valores dos pixels de cada fatia são combinados no *frame buffer* durante o *rendering* para obter o efeito de transparência apropriado. Essa combinação dos pixels é feita de acordo com uma função de transferência associada.

Se o hardware suporta texturas 3D, é possível fazer o *rendering* das fatias paralelas à imagem de projeção com relação à direção de observação atual (Figura 2.5), ou seja, se a direção de observação em relação ao volume for alterada, as fatias devem ser recalculadas (REZK-SALAMA *et al.*, 2000) (Figura 2.6). Como a interpolação tri-linear das texturas é

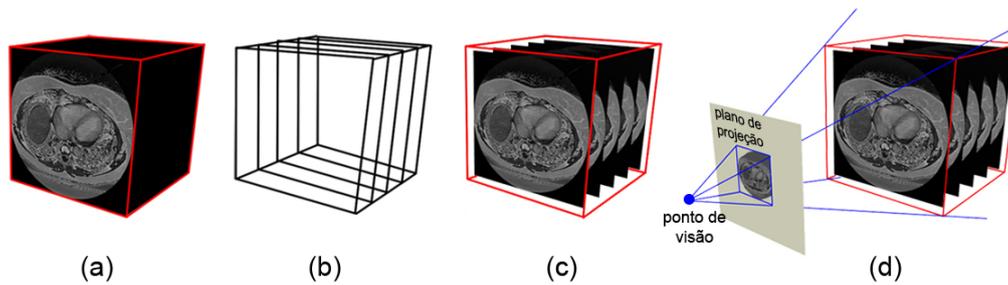


Figura 2.4: Visualização direta por mapeamento de texturas; (a) volume de dados; (b) fatias; (c) mapeamento de fatias no espaço de textura; (d) *rendering* das fatias no frame buffer.

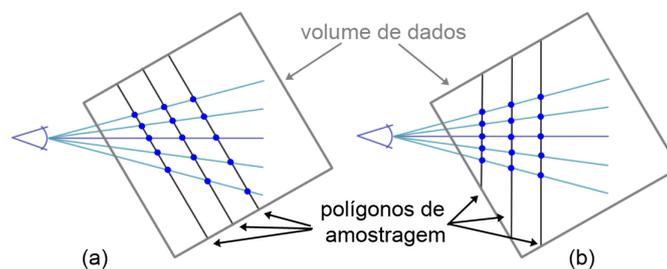


Figura 2.5: Posições das fatias do volume (CULLIP; NEUMANN, 1994): (a) Planos alinhados a um eixo ortogonal do espaço de textura e (b) planos paralelos ao plano de projeção.

suportada pelo hardware, isto pode ser realizado interativamente. Utilizando essa abordagem, a qualidade da imagem pode ser melhorada aumentando-se o número de fatias. Por outro lado, se o hardware suporta somente texturas 2D, as fatias são alinhadas de acordo com a posição do objeto para que a interpolação das texturas seja bi-linear (CABRAL; CAM; FORAN, 1994) (Figura 2.7). Entretanto, se a direção de observação muda mais que 90 graus, a orientação da normal das fatias deve ser alterada. Isso implica na utilização de três cópias do volume de dados na memória principal, ou seja, um conjunto de fatias alinhado a cada eixo do volume. Além da necessidade de maior espaço na memória, esse tipo de implementação gera grandes artefatos indesejáveis na imagem final.

A implementação da visualização direta de volumes baseada no mapeamento de textura pode ser dividida em três estágios (IKITS et al., 2004): *inicialização*, *atualização* e *desenho*. No estágio de inicialização, o conjunto de dados volumétricos é processado e armazenado como uma textura 3D na GPU. Durante esse processo pode ser utilizada uma pré-classificação determinada por uma função de transferência (KINDLMANN; DURKIN, 1998) ou cálculo de valores escalares e gradiente para a iluminação e sombreamento do volume (WEISKOPF; ENGEL; ERTL, 2003). O estágio de atualização é necessário quando o usuário interage com o sistema, modificando parâmetros de visualização, como alterando a posição de observação ou o plano de projeção, e se o hardware suporta texturas 3D, na abordagem com planos paralelos ao plano de projeção, esta modificação necessita que todas as fatias de amostragem sejam recalculadas. A cor de cada fragmento das fatias de amostragem é calculada e combinada no *frame buffer* para gerar a imagem final durante o estágio de desenho.

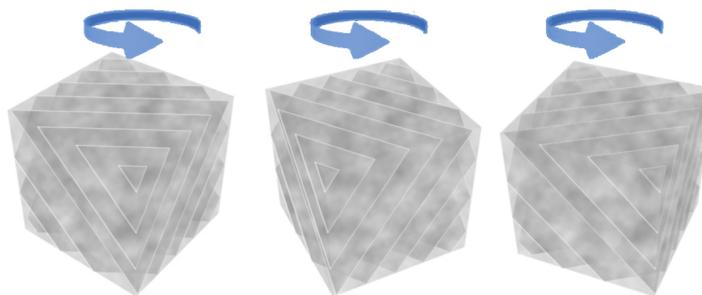


Figura 2.6: Fatias da visualização direta por mapeamento de textura perpendiculares à direção de observação.

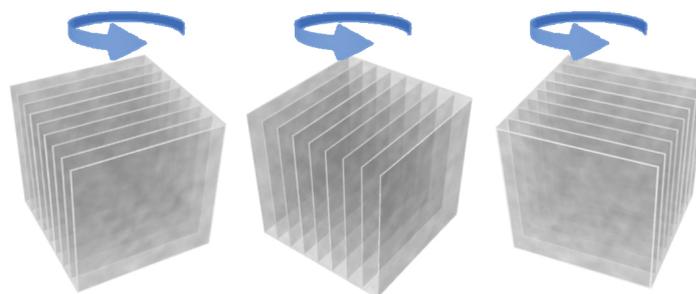


Figura 2.7: Fatias da visualização direta por mapeamento de textura alinhadas com os planos cartesianos do objeto.

2.2 Exploração de volumes baseada em recorte volumétrico

Em visualização direta, freqüentemente são utilizadas funções de transferência para realçar conjuntos de dados volumétricos, mas como essa abordagem é baseada somente nos valores dos dados e/ou valores derivados, às vezes o seu uso é ineficiente. Por exemplo, estruturas internas de tomografias computadorizadas do cérebro são difíceis de serem isoladas devido à semelhança dos seus valores de densidade, e freqüentemente um especialista é necessário para separar manualmente essas estruturas no volume de dados. O recorte volumétrico é então utilizado para auxiliar o especialista a realizar essa tarefa. Assim, o recorte volumétrico desempenha um papel muito importante na compreensão de volumes de dados 3D, pois permite a remoção de partes selecionadas do volume com base na informação de posição dos voxels no conjunto de dados.

A interação com volumes para expor estruturas internas utilizando ferramentas de recorte vem sendo praticamente restrita ao uso de planos de corte. Somente recentes abordagens de recorte têm considerado estruturas geométricas diferentes.

Van Gelder e Kim (GELDER; KIM, 1996) apresentaram um método para visualização direta de volumes chamado *Voltx*. Este método gerava imagens com alto grau de qualidade utilizando mapeamento de texturas 3D suportada por hardware (*Silicon Graphics Workstations*). As texturas eram mapeadas para planos paralelos no formato de quadrados alinhados ao plano de projeção. Devido a esse alinhamento, algumas regiões dessas fatias encontravam-se fora do volume de dados (delimitado por um cubo). Essas regiões eram recortadas utilizando-se um plano de corte para cada face do volume (Figura 2.8). Estes planos também podiam ser utilizados pelo usuário para restringir o *rendering* de partes do volume.

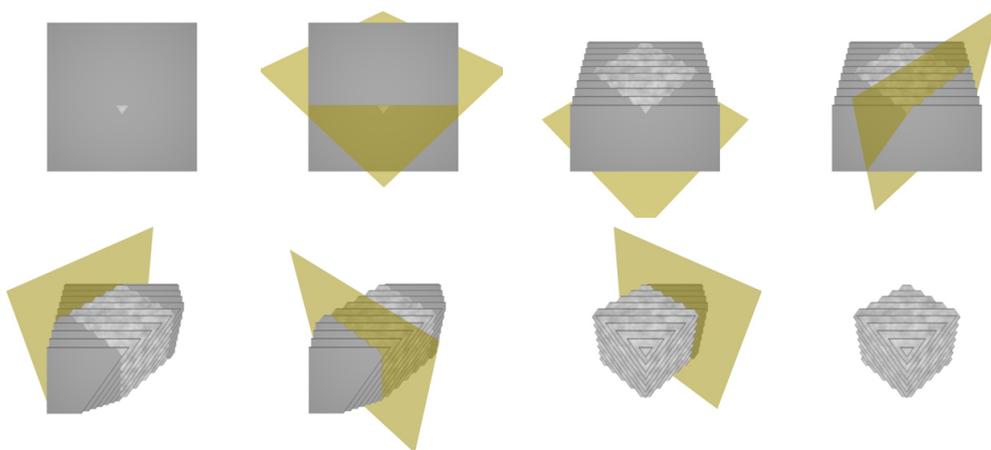


Figura 2.8: Ferramenta de recorte de Van Gelder e Kim (GELDER; KIM, 1996) que utiliza seis planos de corte para eliminar regiões das fatias de amostragem fora do volume de dados.

Westermann e Erl (WESTERMANN; ERTL, 1998) também utilizaram o mapeamento de texturas 3D em fatias para amostrar os dados volumétricos, mas ao invés de utilizar somente o recorte planar, eles realizavam o recorte utilizando estruturas geométricas mais complexas. Essas geometrias podiam ser quaisquer objetos representados por uma lista de triângulos e com superfície fechada. O recorte era feito utilizando-se o *stencil buffer* da OpenGL para determinar os pixels que deveriam ou não ser visualizados. A idéia básica era determinar os pixels presentes na região das fatias interseccionadas pela geometria de recorte. Esses pixels eram marcados no *stencil buffer* para não serem visualizados durante o *rendering* (Figura 2.9).

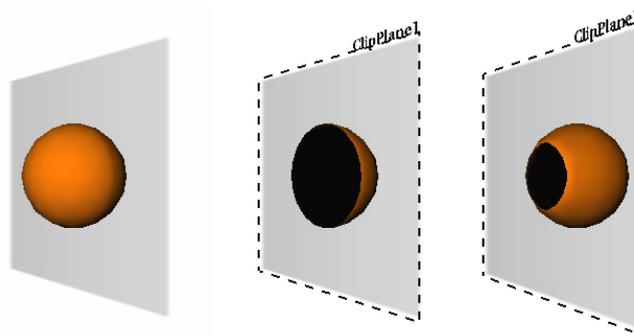


Figura 2.9: Ferramenta que utiliza o *stencil buffer* para fazer o recorte de volumes (WESTERMANN; ERTL, 1998).

Pfister *et al.* (PFISTER *et al.*, 1999) propuseram a arquitetura de hardware *Volume-Pro* com um plano de corte diferente dos anteriores. Nesta implementação, o plano de corte tem uma espessura, e os voxels entre as duas superfícies paralelas do plano de corte são eliminados ou, alternativamente, somente os voxels presentes nessa região são visualizados (Figura 2.10a). Outra ferramenta de recorte presente nessa arquitetura é chamada de *cropping*. Essa ferramenta combina três planos de cortes (também com espessuras), alinhados aos eixos x , y e z , para delimitar a região que será apagada ou visualizada (Figura 2.10b).

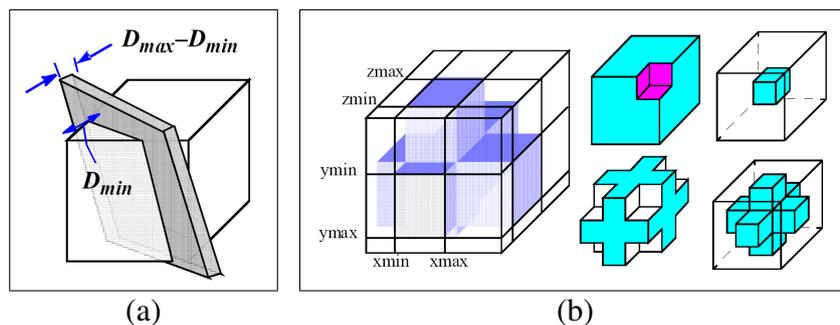


Figura 2.10: Ferramentas do Volume-Pro (PFISTER et al., 1999): (a) plano de corte com espessura e (b) ferramenta com três planos de corte.

Manssour (MANSSOUR et al., 2002) apresentou a formalização do comportamento de ferramentas de recorte em imagens 3D mono ou multimodais. A visualização do volume era realizada através de *ray-casting* e o recorte era feito através da especificação de intervalos de amostragem nos raios que interceptam os voxels do volume para calcular o valor de cada pixel no plano de projeção (Figura 2.11). Esses intervalos eram determinados pela geometria das ferramentas de recorte (recorte baseado em geometria), mas as propriedades dos voxels também poderiam ser utilizadas em conjunto (recorte baseado em conteúdo).

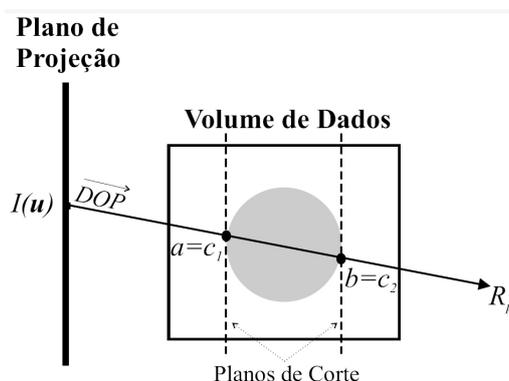


Figura 2.11: Recorte obtido pela especificação de intervalos nos raios da abordagem de visualização de volumes por *ray-casting* (MANSSOUR et al., 2002)

Weiskopf *et al.* (WEISKOPF; ENGEL; ERTL, 2002) exploraram os programas de fragmento presentes no hardware gráfico para realizar os recortes e fazer o *rendering* em alta velocidade. Apesar de eles utilizarem visualização volumétrica baseada em texturas, uma de suas ferramentas de recorte é semelhante à técnica de *ray-casting* utilizada por Manssour *et al.* A intersecção dos raios de amostragem de cada pixel com a geometria de recorte é calculada e armazenada no *depth buffer* (Figura 2.12). O *rendering* das fatias do volume de dados é feito e combinado no *frame buffer*. O teste de profundidade é utilizado para determinar a visibilidade dos fragmentos.

Weiskopf *et al.* propuseram também outras ferramentas de recorte que utilizam objetos voxelizados na CPU e armazenados em texturas 3D para fazer o recorte. Essas texturas são máscaras com informações de visibilidades utilizadas para indicar se um voxel da textura original, que contém os dados volumétricos, contribui na imagem final ou não (Figura 2.13).

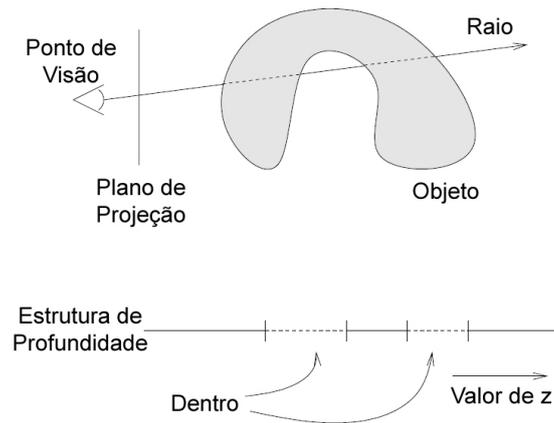


Figura 2.12: Recorte baseado na profundidade dos voxels (WEISKOPF; ENGEL; ERTL, 2002)

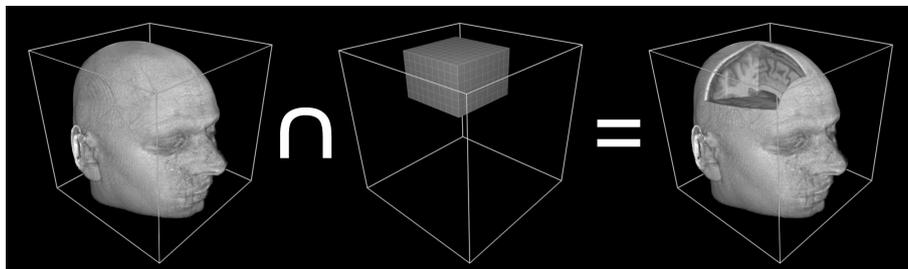


Figura 2.13: Recorte baseado em máscara de visibilidade(adaptada de(WEISKOPF; ENGEL; ERTL, 2002))

Baseado no trabalho de Weiskopf *et al.*, Dietrich *et al.* (DIETRICH *et al.*, 2004) desenvolveram ferramentas de recorte voxelizadas na GPU e especificadas por parâmetros obtidos através da interação 2D com o volume. Semelhante abordagem foi utilizada por Bruckner e Gröller (BRUCKNER; GRÖLLER, 2005), que propuseram um sistema, chamado *VolumeShop*, que utiliza ferramentas de recorte para produzir ilustrações de volumes. Essas ferramentas são utilizadas para eliminar ou alterar os valores de opacidade de certas regiões do volume (Figura 2.14).

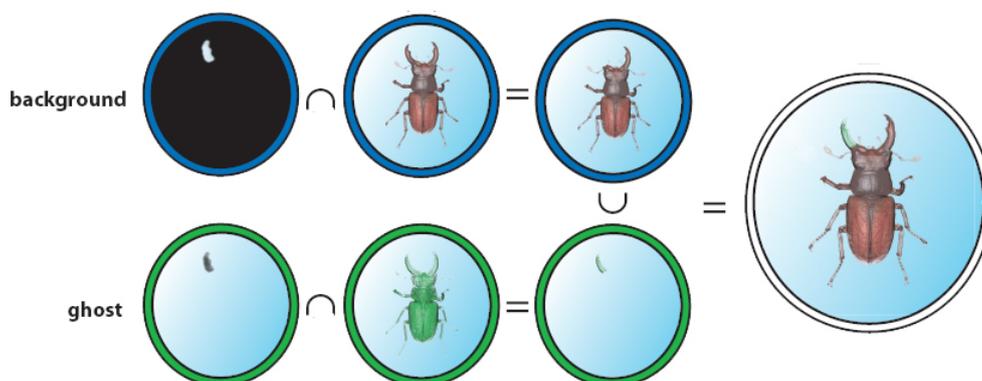


Figura 2.14: Ferramenta de recorte utilizada para produzir ilustrações de volumes (adaptada de (BRUCKNER; GRÖLLER, 2005))

2.3 Síntese do Capítulo

Neste capítulo foram brevemente revistos os conceitos principais relacionados à visualização volumétrica e os trabalhos relacionados à exploração de volumes baseada em recorte.

O presente trabalho utiliza o paradigma proposto por Weiskopf *et al.* (WEISKOPF; ENGEL; ERTL, 2002) para realizar o recorte de volumes visualizados com mapeamento de textura. Para realizar este procedimento, foi utilizado um algoritmo que explora a coerência espacial entre os polígonos de amostragem do volume, chamado WSG (*Weighted Sweep Graph*)(DIETRICH *et al.*, 2004). A implementação das ferramentas propostas nesse trabalho também utiliza o *pipeline* de visualização volumétrica proposto por Ikits (IKITS *et al.*, 2004) e será descrita no Capítulo 4.

3 INTERAÇÃO

O processo de interação em ambientes virtuais consiste na capacidade reativa do sistema em detectar e responder cada ação do usuário através de modificações instantâneas no ambiente virtual (NEDEL et al., 2003). Trata-se de um processo contínuo, onde o usuário efetua ações no ambiente virtual através dos dispositivos de entrada. Essas ações são mapeadas no ambiente virtual de acordo com as técnicas de interação empregadas para executar uma determinada tarefa. A resposta é fornecida ao usuário através de dispositivos de saída que estimulam os sentidos do usuário, que são normalmente a visão e a audição.

Este capítulo apresenta as técnicas de interação mais frequentemente empregadas em ambientes virtuais, mas considerando principalmente àquelas mais relevantes no estudo de um sistema de recorte volumétrico. Também são apresentadas as taxonomias e metáforas dessas técnicas. Por último são mostradas algumas formas de interação com dispositivos de entrada.

3.1 Técnicas de Interação

Em ambientes virtuais, frequentemente é necessário interagir com objetos virtuais ao invés de simplesmente observá-los. A forma como essa interação é desenvolvida determina diretamente se um sistema é intuitivo e fácil de usar. Existem várias técnicas de interação e a especificação da técnica mais apropriada para um determinado sistema depende do tipo da aplicação desenvolvida. A tendência em ambientes virtuais é tornar essas técnicas o mais naturais possível. Uma forma de se fazer isso é mapear as ações realizadas no mundo real para o mundo virtual, ao invés de se utilizar comandos abstratos ou especificações de coordenadas e ângulos de rotação. Essas ações podem ser movimentos como segurar, girar e mudar um objeto de posição. Esse tipo de técnica de interação é chamada de *metáfora do mundo-real* (BOWMAN; HODGES, 1997).

Entretanto a metáfora do mundo-real tem várias limitações. Frequentemente, um problema encontrado é com relação ao mapeamento do braço real no mundo virtual, pois este fica confinado a um espaço limitado ao redor do usuário, não sendo possível alcançar objetos mais distantes. Conseqüentemente é necessária uma técnica adicional de navegação para movimentar o usuário até uma nova posição suficientemente próxima ao objeto para manipulá-lo. Outro problema encontrado é com relação à manipulação de objetos muito grandes, pois esses podem obscurecer a visão do usuário.

Devido aos problemas existentes, a metáfora do mundo-real é estendida em diversas outras técnicas de interação. Essas técnicas, por sua vez, podem ser agrupadas em outras metáforas de interação e tratam principalmente de três ações básicas de interação com objetos: *navegação*, *seleção*, e *manipulação*. A taxonomia das tarefas realizadas por cada

ação é descrita a seguir na Seção 3.1.1, e as metáforas mais comumente empregadas são discutidas após, na Seção 3.1.2.

3.1.1 Taxonomia

Apesar do grande número de possíveis tarefas realizadas em um ambiente virtual serem específicas para a sua aplicação ou domínio, as ações que os usuários executam podem ser classificadas em poucas interações básicas. Através da identificação e estudo dessas ações básicas, é possível desenvolver e avaliar técnicas de interação para essas tarefas. As técnicas de interação resultantes podem então ser úteis para uma grande variedade de outros ambientes. Portanto, a compreensão e identificação de tais tarefas básicas de um ambiente virtual é um importante passo inicial para o tratamento de dificuldades de usabilidade de complexos ambientes virtuais interativos (BOWMAN; HODGES, 1999). Muitas dessas tarefas básicas podem ser classificadas como: *navegação*, *seleção* e *manipulação*.

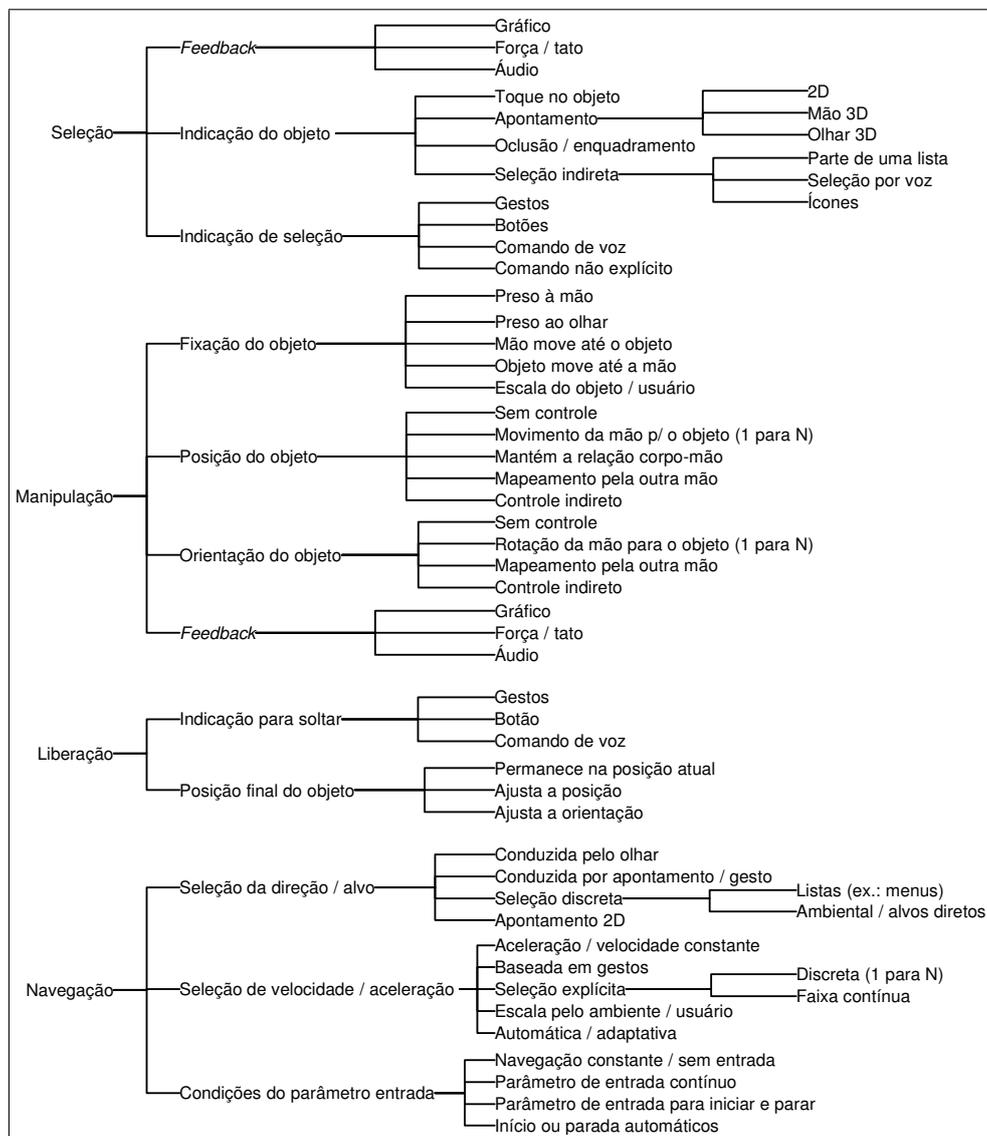


Figura 3.1: Taxonomia das técnicas de seleção, manipulação e navegação em ambientes virtuais (BOWMAN; HODGES, 1999).

As tarefas de navegação são as ações que o usuário executa para interativamente posicionar e orientar o seu ponto de visão no ambiente, ou seja, para se movimentar de um lugar a outro no mundo virtual. Seleção é a tarefa que envolve escolher um ou mais objetos para um determinado propósito. Por último, as tarefas de manipulação se referem a posicionar, orientar, escalar, ou até mesmo modificar o formato de objetos virtuais. A Figura 3.1 ilustra a taxonomia das técnicas de navegação, seleção e manipulação. Ainda é acrescentada a sub-tarefa de *liberação*, que corresponde à ação de “soltar” um objeto após este ser manipulado.

No presente trabalho, as técnicas de interação são estudadas no contexto de recorte volumétrico. Como no sistema proposto o volume que se deseja esculpir encontra-se próximo ao usuário, às técnicas de navegação fogem do escopo desse trabalho. Também é assumido que a tarefa de recortar um volume corresponde às ações de selecionar primeiramente os voxels que se deseja eliminar e removê-los posteriormente. A ação de remover pode ser considerada como uma forma de liberação desses voxels. Apesar de não ser realizada nenhuma operação de manipulação nos voxels selecionados, as técnicas de manipulação também são discutidas nesse trabalho, pois para se esculpir um volume, normalmente também é necessário rotacioná-lo e, além disso, algumas metáforas utilizadas para a manipulação de objetos, podem ser utilizadas para a seleção e vice-versa.

3.1.2 Metáforas de Interação

As técnicas de interação podem ser divididas em duas metáforas básicas: *exocêntricas* e *egocêntricas* (POUPYREV et al., 1998). Na metáfora exocêntrica, também conhecida como “*God’s eye viewpoint*”, os usuários interagem com os objetos de fora do ambiente. Na metáfora egocêntrica, por outro lado, o usuário interage de dentro do próprio ambiente. A Figura 3.2 apresenta as diferentes classificações das técnicas mais empregadas de interação.

Alguns exemplos de metáforas exocêntricas são a *World in Miniature* (STOAKLEY; CONWAY; PAUSCH, 1995) e *Automatic Scaling* (MINE; FREDERICK P. BROOKS; SEQUIN, 1997). A metáfora do *World in Miniature* (WIM) procura solucionar problemas

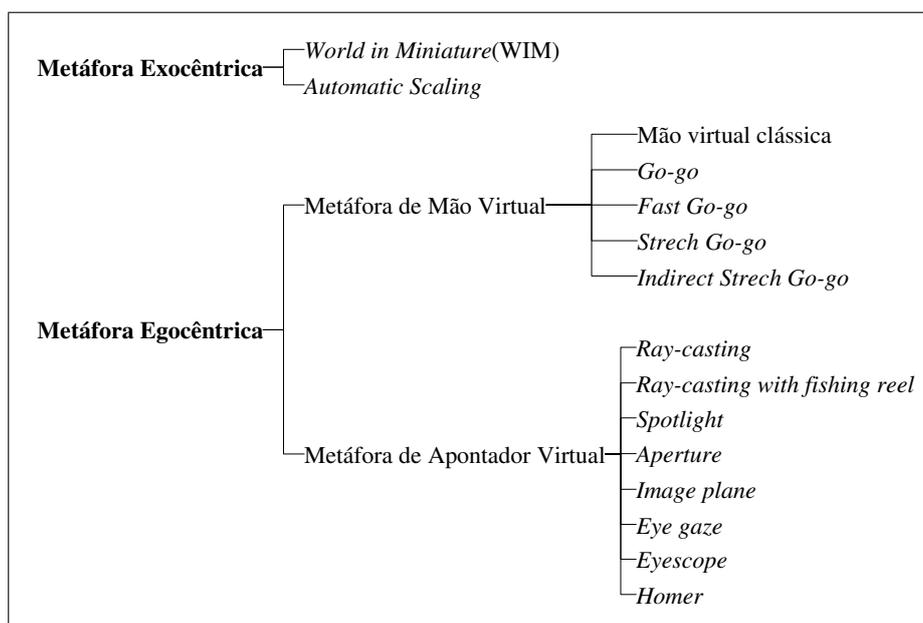


Figura 3.2: Metáforas e técnicas de manipulação e seleção em ambientes virtuais

de limitações e ambigüidades no mundo virtual que normalmente também estão presentes no mundo real, como objetos que não são alcançáveis por estarem muito longes, objetos oclusos ou fora de vista, e objetos que parecem estar próximos uns dos outros por se encontrarem alinhados ao longo da linha de visão. Para tentar corrigir esses problemas, a WIN possibilita a manipulação de objetos através da interação com suas representações num modelo em miniatura do ambiente (Figura 3.3a) que é segurado pelo usuário (Figura 3.3b). Outro exemplo de metáfora exocêntrica é a *Automatic Scaling* (Figura 3.4), que redimensiona automaticamente o ambiente virtual quando um objeto é selecionado para que este objeto se encontre ao alcance do usuário para ser manipulado. Após a manipulação, quando o objeto é liberado, o ambiente volta ao tamanho original, mas mantendo as transformações aplicadas ao objeto durante o processo de manipulação.

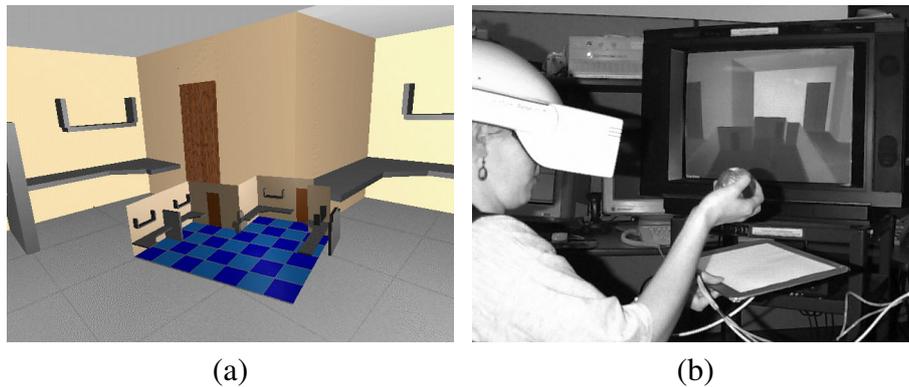


Figura 3.3: Metáfora exocêntrica *World in Miniature* (STOAKLEY; CONWAY; PAUSCH, 1995): (a) modelo em miniatura do ambiente; (b) usuário manipulando o modelo.

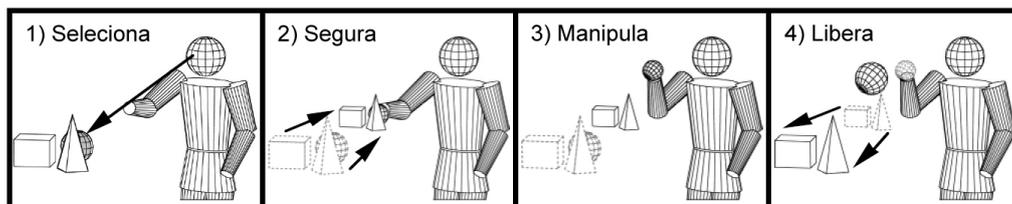


Figura 3.4: Metáfora exocêntrica *Automatic Scaling*

As técnicas exocêntricas encontram-se fora do escopo do presente trabalho pelo mesmo motivo que não são necessárias técnicas de navegação, pois o volume de dados para se realizar o recorte encontra-se próximo do usuário, e em geral é totalmente visualizado. As técnicas egocêntricas, por outro lado, são analisadas para possibilitar uma implementação eficiente do sistema de recorte volumétrico. Essas técnicas são descritas a seguir.

3.1.2.1 Metáforas Egocêntricas para Seleção de Objetos em Ambientes Virtuais

A seleção é normalmente utilizada para especificar um objeto ao qual se deseja aplicar algum comando, um grupo de objetos relacionados, ou um objeto que se deseja manipular. A técnica mais óbvia e trivial para seleção de objetos é simplesmente selecioná-los quando a representação virtual da mão do usuário entra em contato com eles. Este método é chamado de *mão virtual* (Figura 3.5a) e funciona bem em situações onde o usuário

pode interagir com os objetos no mundo virtual semelhantemente como faria no mundo real. Entretanto, esse método não seria apropriado para ambientes mais complexos, onde a representação virtual da mão do usuário poderia não ser suficientemente precisa para selecionar objetos muito pequenos e aglomerados, e também não seria adequado para selecionar objetos que se encontram fora do alcance do usuário. Levando isso em consideração, várias aplicações utilizam a técnica de *apontadores virtuais* (Figura 3.5b) para seleção de objetos. Nesta técnica, um apontador, que pode ser um raio, é utilizado para apontar (selecionar) o objeto desejado. Isto elimina o problema de precisão, pois o apontador seleciona um único ponto, e elimina também o problema de distância, pois qualquer objeto pode ser selecionado de qualquer distância, desde que o usuário possa visualizá-lo, mas gera outros problemas como a dificuldade de apontar objetos quando a resolução não é adequada.

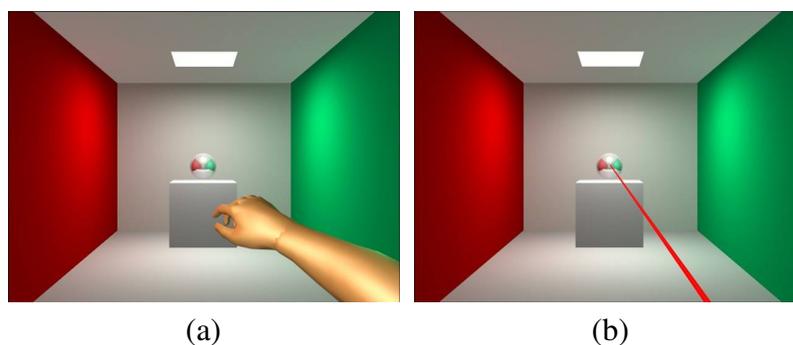


Figura 3.5: Metáforas de interação: (a) mão virtual e (b) apontador virtual

Considerando as vantagens e restrições das metáforas da mão virtual e de apontadores virtuais, novas extensões dessas técnicas foram implementadas. As variações dessas técnicas são resumidas a seguir.

- **Mão virtual clássica:** Com a mão virtual, os usuários podem segurar e posicionar objetos “tocando” neles com uma representação virtual de sua mão real, permitindo uma interação natural e intuitiva (Figura 3.6). Os fatores mais importantes nesse tipo de técnica são a escolha dos dispositivos de entrada para a manipulação da mão virtual e o mapeamento entre a posição e orientação da mão real para a virtual (POUPYREV et al., 1998). Entretanto, a maior desvantagem dessa técnica é justamente esse mapeamento 1:1 entre a mão real e virtual, pois os usuários não conseguem interagir diretamente com objetos fora de seu alcance (SONG; NORMAN, 1993).

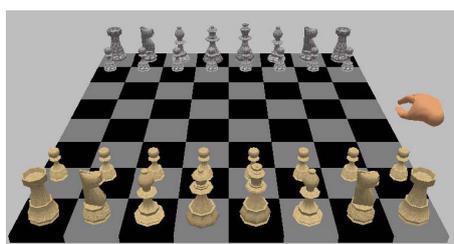


Figura 3.6: Técnica de interação da mão virtual clássica utilizada para mudar de posição as peças de um tabuleiro de xadrez. (NEDEL et al., 2003)

- **Go-go:** Essa técnica é uma extensão da mão virtual clássica para permitir tanto a seleção e manipulação de objetos próximos quanto de objetos distantes (POUPYREV et al., 1996). Isto é realizado através do mapeamento linear entre a mão real e virtual quando essa se encontra próxima do usuário. Quando o usuário estende o braço mais longe, o mapeamento passa a ser não-linear aumentando o braço virtual do usuário (Figura 3.7). A desvantagem dessa técnica é relacionada à falta de precisão e facilidade na seleção e manipulação de objetos muito distantes, pois nesses casos o braço virtual aumenta e diminui com muita rapidez.

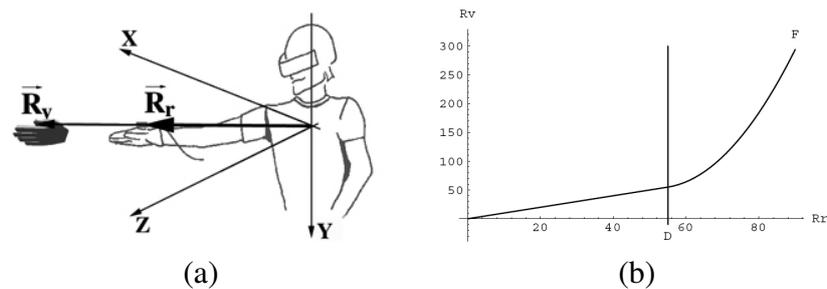


Figura 3.7: Técnica de interação *Go-go* (POUPYREV et al., 1996): (a) a mão do usuário tem alcance mais longo que seu braço; (b) mapeamento não linear do braço real e virtual.

- **Fast Go-go:** Esta é uma técnica de extensão da *Go-go* que não utiliza o mapeamento linear na região próxima ao usuário (BOWMAN; HODGES, 1997), e possui uma função com crescimento mais rápido (Figura 3.8). Isto é feito para tornar a seleção de objetos mais rápida, mas continua apresentando as mesmas desvantagens da *Go-go*.

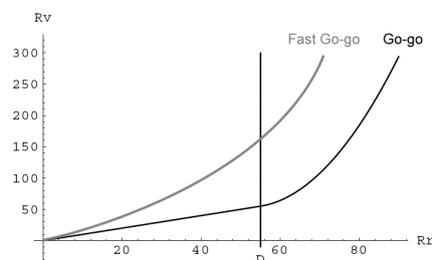


Figura 3.8: Técnica de interação *Fast Go-go*: sem mapeamento linear próximo ao usuário e com função de crescimento mais rápida que a *Go-go* (adaptada de (POUPYREV et al., 1996)).

- **Stretch Go-go:** A *Stretch Go-go* é uma variante da *go-go* para selecionar qualquer objeto em qualquer ambiente, pois possibilita a extensão do braço virtual infinitamente (BOWMAN; HODGES, 1997). Isto é feito através da divisão do espaço ao redor do usuário em três regiões concêntricas, semelhantemente a técnica *Flying Speed* descrita em (MINE, 1995). Quando a mão do usuário está na região mais afastada, o seu braço virtual cresce numa velocidade constante (Figura 3.9). Entretanto, quando a mão do usuário está na região mais próxima dele, o braço retrai na mesma velocidade. A região do meio é utilizada para fazer o mapeamento natural, e portanto o braço não muda de tamanho. Apesar de qualquer tamanho de braço ser possível com essa técnica, ela é mais difícil de ser utilizada do ponto de vista cognitivo que a *Go-go*.

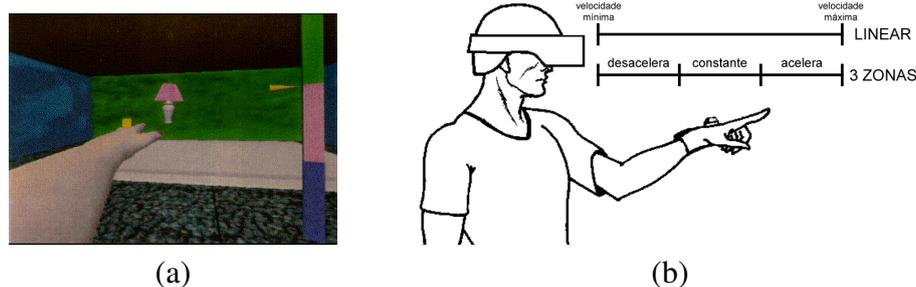


Figura 3.9: Técnicas de interação que utilizam regiões concêntricas ao redor do usuário para aumentar o tamanho do seu braço virtual: (a) *Stretch Gogo* (BOWMAN; HODGES, 1997): (b) *Flying Speed* (MINE, 1995)

- **Indirect Stretch Go-go:** Esta técnica é semelhante à *Stretch Go-go*, mas ao invés de se utilizar movimentos do braço real para estender e retrair o braço virtual, são utilizados botões de um mouse 3D (BOWMAN; HODGES, 1997). Um botão estende o braço numa velocidade constante, enquanto que o outro o retrai na mesma velocidade. Apesar dessa técnica não ser tão natural, ela é mais precisa e fisicamente mais fácil de usar que a *Stretch Go-go*.
- **Ray-casting:** Esta é uma das técnicas mais utilizadas de apontadores virtuais, cujo apontador é um raio que parte da mão do usuário (MINE, 1995). Para selecionar um objeto (Figura 3.10), o usuário aponta o raio para ele, e após interceptá-lo, a seleção é feita pressionando-se um botão do dispositivo de entrada. Qualquer objeto visível pode ser selecionado nesta técnica, mas apresenta vários problemas de manipulação. Um problema quanto à rotação, é que o objeto só pode ser rotacionado ao redor do eixo do raio que o intercepta, e um problema quanto a translação é a impossibilidade de transladar um objeto sem fazer ao mesmo tempo uma rotação ao redor do usuário.

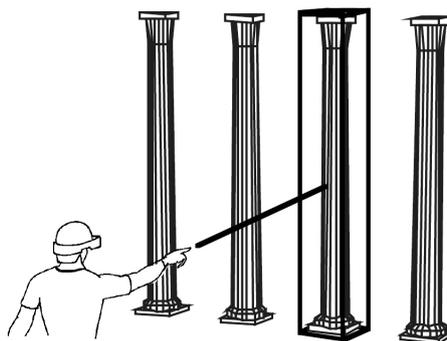


Figura 3.10: Técnica de interação *ray-casting*. Um raio que parte da mão do usuário é utilizado como apontador.

- **Ray-casting with fishing reel:** Esta modificação do *ray-casting* foi desenvolvida para possibilitar a translação de objetos para mais próximo ou distante do usuário (BOWMAN; HODGES, 1997). Isto é feito de forma semelhante à *Indirect Stretch Go-go*, onde dois botões de um dispositivo de entrada são utilizados para aproximar ou distanciar objetos selecionados (Figura 3.11).

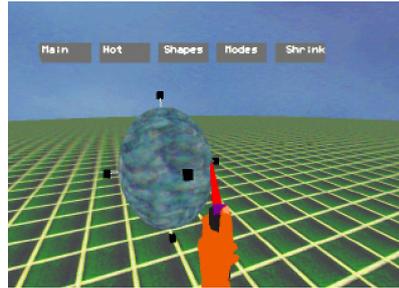


Figura 3.11: Técnica de interação de *Ray-casting with fishing reel* (BOWMAN; HODGES, 1997). Os objetos selecionados por *ray-casting* são aproximados do usuário para serem manipulados.

- **Spotlight:** Esta técnica é uma variação da técnica de *ray-casting*. Com *ray-casting* é difícil a seleção de objetos muito pequenos devido a seleção por um raio. A *Spotlight* corrige esse problema com a utilização de um cone de seleção, como se fosse uma lanterna (Figura 3.12). O ápice do cone encontra-se na mão do usuário e a base é direcionada aos objetos que se deseja selecionar (MINE; FREDERICK P. BROOKS; SEQUIN, 1997). O problema dessa técnica é em relação a objetos aglomerados, pois a seleção de um único objeto pode ser complicada.

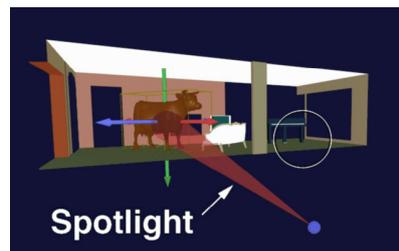


Figura 3.12: Técnica de interação *Spotlight* (MINE; FREDERICK P. BROOKS; SEQUIN, 1997). Um cone é utilizado para fazer a seleção dos objetos.

- **Aperture:** Esta é uma modificação da técnica de *Spotlight*, que utiliza um cursor para fazer a orientação do volume de seleção (FORSBERG; HERNDON; ZELENIK, 1996). O ápice do cone de seleção é posicionado no ponto de visão do usuário e o vetor de direção do cone é o vetor que parte do ponto de visão através do centro do cursor (Figura 3.13). O raio desse cursor é fixo e o tamanho do volume de seleção é determinado pela distância entre o ponto de visão e o cursor, ou seja, o tamanho do volume pode ser modificado aproximando ou afastando o cursor do ponto de visão.
- **Image plane:** Este é um conjunto de técnicas de seleção baseadas na projeção do ambiente em um plano de visão (PIERCE et al., 1997). Ao invés dos objetos serem selecionados no ambiente 3D, eles são selecionados em 2D na projeção do ambiente no plano de visão. Essa seleção pode ser feita de diversas formas: o usuário pode apontar com o dedo indicador para ele (Figura 3.14a), o usuário pode selecioná-lo entre o polegar e o indicador (Figura 3.14b), o usuário pode selecioná-lo colocando-o em sua palma da mão (Figura 3.14c), ou o usuário pode usar as duas mãos para selecioná-lo.

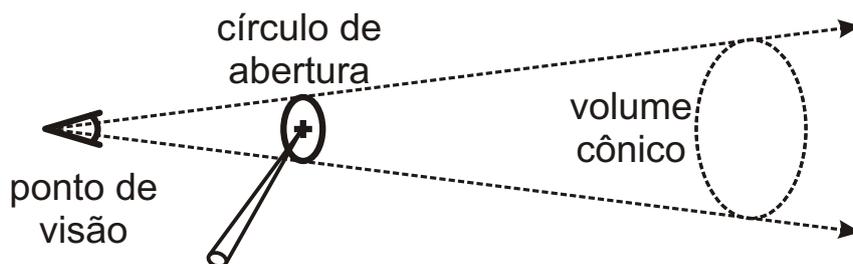


Figura 3.13: Técnica de interação *Aperture* (FORSBERG; HERNDON; ZELEZNIK, 1996). O cone de seleção é alterado pela distância em que se encontra o círculo de abertura em relação ao ponto de visão.

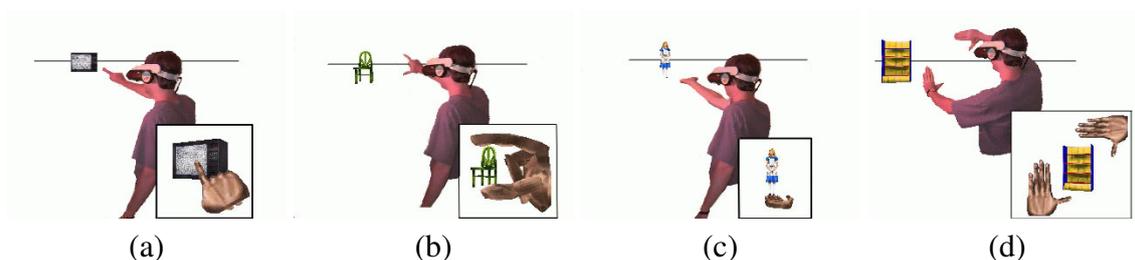


Figura 3.14: Técnica de interação *Image Plane* (PIERCE et al., 1997) onde o usuário pode selecionar um objeto apontando p/ ele (a), utilizando dois dedos (b), utilizando a palma da mão (c), ou utilizando as duas mãos (d).

- **Eye Gaze:** Esta é outra técnica que pode ser classificada com um tipo de apontador virtual (JACOB, 1990). Através da detecção da direção do olhar do usuário é feito o apontamento do objeto (Figura 3.15). A seleção é realizada pressionando-se um botão, ou mantendo o olhar fixo no objeto por alguns segundos.

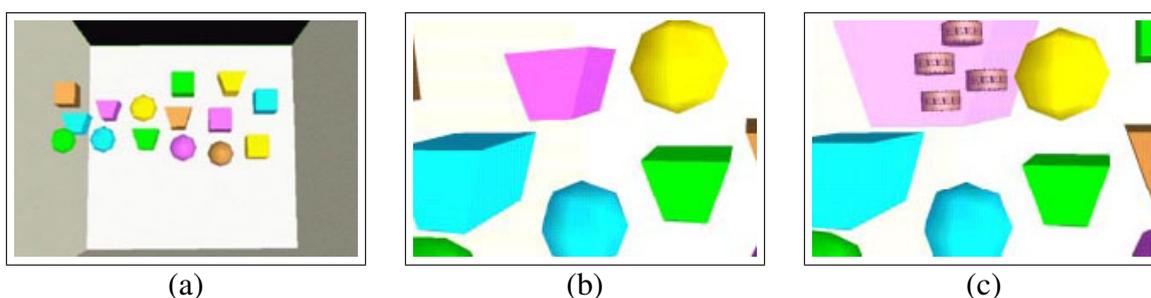


Figura 3.15: Técnica de interação *Eye Gaze* (JACOB, 1990): (a) todo o ambiente; (b) parte dos objetos ainda não selecionados; (c) o objeto magenta é selecionado e seus detalhes internos são revelados.

- **Eyesclope:** Esta é uma extensão da Eye Gaze desenvolvida para evitar os problemas gerados pela baixa resolução dos capacetes de realidade virtual, por ruídos, ou por objetos muito distantes, pequenos ou semi-occlusos (UGHINI et al., 2006). Nesta técnica, é utilizado um controle de zoom para focalizar uma região de interesse específica e aumentá-la, fornecendo um controle maior ao usuário para selecionar objetos (Figura 3.16).

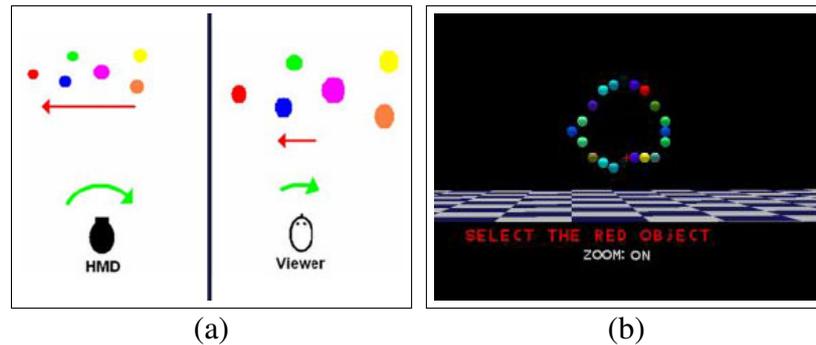


Figura 3.16: Técnica de interação *Eyescope* (UGHINI et al., 2006): (a) mapeamento da distância dos objetos do HMD e do usuário (com zoom); (b) ambiente de avaliação

- **Homer (*Hand-centered Object Manipulation Extending Ray-casting*):** Nesta técnica os objetos são selecionados através da técnica de *ray-casting*, mas ao invés dos objetos ficarem presos ao raio, a mão virtual do usuário é deslocada até a posição do objeto e o objeto é preso à ela para a manipulação (BOWMAN; HODGES, 1997) (Figura 3.17). Quando o objeto é solto, a mão retorna a posição natural. Apesar dessa técnica ser originalmente classificada como uma técnica híbrida (mão virtual e *ray-casting*), em relação somente a seleção, ela é uma técnica que utiliza a metáfora de apontador virtual.

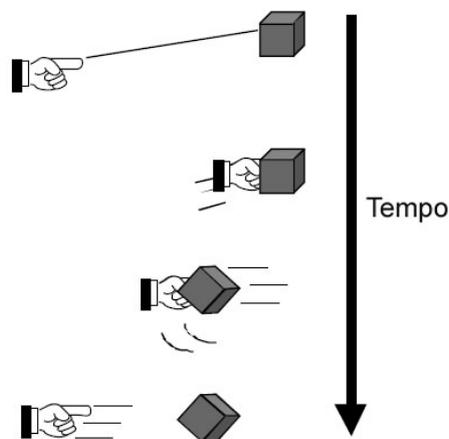


Figura 3.17: Técnica de interação *Homer* (BOWMAN; HODGES, 1997). Primeiramente, é utilizado *ray-casting* para a seleção do objeto e, após, é utilizada a mão virtual para manipulação.

3.2 Dispositivos de Entrada

Inúmeros dispositivos de entrada foram projetados para interação com ambientes virtuais. A escolha do dispositivo de entrada mais apropriado para uma determinada técnica de interação é crucial para o desenvolvimento de um sistema. Esta seção lista alguns dos dispositivos mais comuns atualmente e apresenta uma visão geral sobre suas diversas formas de classificação. Após também é discutido como pode ser realizada a escolha dos dispositivos para uma determinada aplicação.

3.2.1 Tipos de Dispositivos de Entrada

Dispositivos de Entrada 1D e 2D: Atualmente, apesar dos ambientes virtuais serem caracteristicamente 3D, os dispositivos de entrada mais utilizados para a interação com esses ambientes são 1D e 2D. Isto é, provavelmente, devido ao fato dos usuários estarem mais acostumados a interagir com sistemas tipicamente 2D, como editores de texto e imagens. O dispositivo 1D mais comum é o teclado, pois utiliza uma seqüência de teclas pressionadas para navegar e interagir com ambientes. Já o dispositivo 2D normalmente movimenta um cursor em duas dimensões na tela. O dispositivo 2D mais comum é o mouse, mas existem vários outros amplamente utilizados também como *joysticks*, mesas digitalizadoras, *lightpens*, *trackballs*, *touchpads*, *pointing sticks*, *head pointers*, *light guns*, *touch screens*, *gamepads*, *power pads* entre outros. Estes dispositivos são ilustrados na Figura 3.18.



Figura 3.18: Dispositivos 2D mais comuns

Spaceballs e Mouse 3D (Spacemouse): São dispositivos fixos à uma base com seis graus de liberdade. Com eles é normalmente possível fazer a translação e rotação de objetos no ambiente virtual simultaneamente. Esta classe de dispositivos é ilustrado na Figura 3.19.



Figura 3.19: Dispositivos de entrada 3D: (a) *Spaceball* e (b) *Spacemouse*

Rastreadores de posição: Tipicamente, os rastreadores de posição utilizam seis graus de liberdade para descrever a posição e orientação do usuário. Três graus de liberdade são utilizados para determinar a posição do usuário (x , y , z) e os outros três são utilizados para determinar a orientação (*pitch*, *yaw*, *roll*), descritas como ângulos de rotação ao redor dos eixos x , y e z . Diversas formas de tecnologia são utilizadas para detectar esses valores de posição e orientação e podem ser classificadas como mecânicas, eletromagnéticas, acústicas, ópticas, inerciais e de olhar (Figura 3.20).

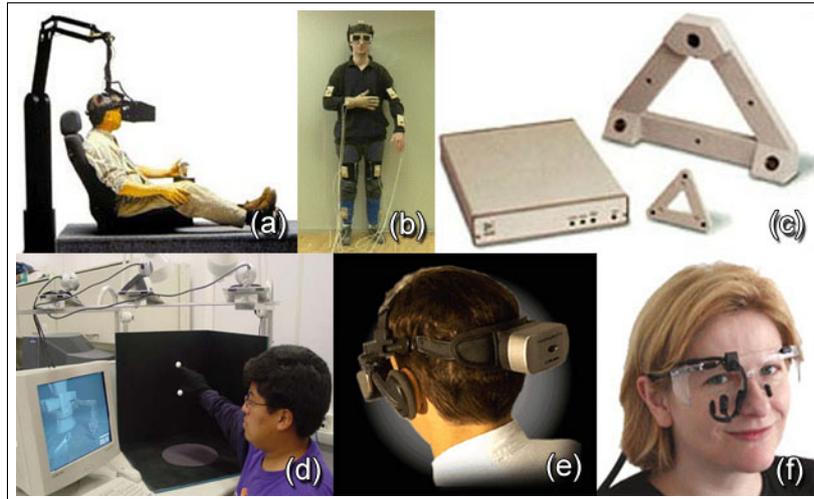


Figura 3.20: Rastreadores de posição: (a) mecânico, (b) eletromagnético, (c) acústico, (d) óptico, (e) inercial, e (f) rastreador de olhar.

Dispositivos Hápticos: Os dispositivos hápticos podem ser considerados tanto dispositivos de entrada quanto de saída. Estes dispositivos utilizam atuadores para fornecer um retorno de força ao usuário (Figure 3.21). O retorno de força pode ajudar o usuário a realizar certas tarefas que necessitam grande precisão. Estes dispositivos foram utilizados inicialmente para simulação médica e para operação de robôs remotamente.

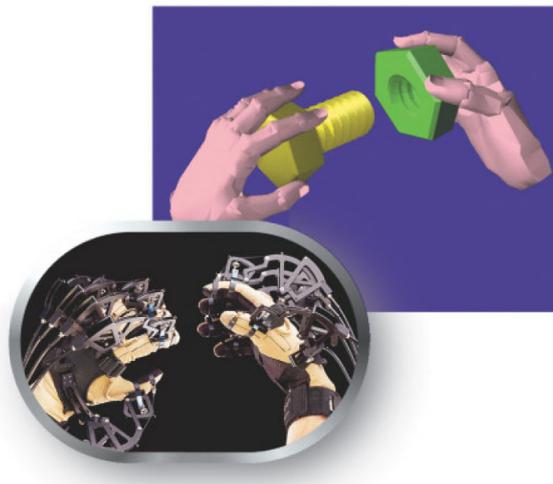


Figura 3.21: Dispositivo tátil CyberGrasp da Immersion

Reconhecimento de Gestos: O reconhecimento de gestos oferece uma maneira natural e intuitiva dos usuários interagirem com o sistema. Tipicamente, os usuários vestem uma luva ou roupa que detectam a posição ou flexão de determinadas juntas. As tecnologias utilizadas para determinar essas informações incluem fibra óptica (DataGlove da VPL), sistemas de câmeras baseados em diodos (LED Gloves do MIT), sensores de resistência (CyberGlove da Virtual Technologies, (Figure 3.22)a), e células de carga (*strain gages*, Power Glove da Mattel, (Figure 3.22)b). O processamento de imagens, que não necessita nenhuma vestimenta, também pode ser utilizado para o reconhecimento de gestos.



Figura 3.22: Dispositivos para reconhecimento de gestos: (a) CyberGlove da Virtual Technologies e (b) PowerGlove da Mattel

Reconhecimento de Voz: A tecnologia de reconhecimento de voz utiliza a voz humana para interagir com o sistema. Nestes sistemas, a fala dos usuários é comparada com um conjunto de palavras e frases já conhecidas. Alguns sistemas necessitam de treinamento para reconhecer as informações ditas por um novo usuário, enquanto outros são genéricos. Os dispositivos de entrada para esse tipo de tecnologia são normalmente microfones, mas há outros mais complexos como sensores conectados a nervos da garganta (Figura 3.23).



Figura 3.23: Dispositivo para reconhecimento de voz (NASA)

3.2.2 Modelos teóricos dos Dispositivos de Entrada

Desde o início dos anos 70, cientistas tentam estabelecer uma classificação dos dispositivos de entrada e suas técnicas de interação para poder determinar quando um particular dispositivo deve ser utilizado. Em 1974, Foley e Wallace (FOLEY; WALLACE, 1974), fizeram uma separação entre tarefas de entrada e dispositivos de entrada. Em teoria, a tarefa de entrada era independente do dispositivo de entrada, e Foley e Wallace desenvolveram um modelo de entrada que era independente dos dispositivos. Eles apresentaram quatro dispositivos virtuais, o *pick*, o *button*, o *locator* e o *valuator*, que existiam independentes do tipo de dispositivo de entrada utilizado. O primeiro padrão independente de dispositivos para sistemas gráficos, o *Core Graphics System* (DUNN; HERZOG, 1977), separava os dispositivos de entrada do código do computador, possibilitando a implementação rápida e flexível de protótipos. Posteriormente no padrão GKS, o modelo original de Foley e Wallace foi refinado por Enderle, Kansy e Pfaff (ENDERLE; KANSY; PFAFF, 1986). Eles descreveram dispositivos virtuais - *pick*, *choice*, *locator*, *valuator*, *stroke* e *string*, como parte do padrão.

No início dos anos 80, os dispositivos de entrada tiveram maior atenção e diversas taxonomias foram criadas. Buxton (BUXTON, 1983) desenvolveu uma taxonomia de dispositivos de entrada baseada em seus graus de liberdade e algumas propriedades por eles geradas (posição, movimentação e pressão). Dessa forma, foram identificados os “atributos pragmáticos” de cada dispositivo. Após, Mackinlay, Card e Robertson (MACKINLAY; CARD; ROBERTSON, 1990) expandiram essa taxonomia para incluir propriedades contínuas e discretas, e constataram que o desempenho humano é importante para entender como um dispositivo realmente funciona em uma determinada situação. Além disso, desenvolveram técnicas de avaliação para os dispositivos.

Foley, Wallace e Chan (FOLEY; WALLACE; CHAN, 1984) também estabeleceram uma taxonomia semelhante à taxonomia de Buxton. Essa taxonomia considerava os atributos pragmáticos de diferentes dispositivos de entrada. Eles mapearam tarefas básicas de interação em dispositivos capazes de realizar essas tarefas, e propuseram que, baseando-se nos requisitos das tarefas, somente um conjunto limitado de dispositivos de entrada poderia ser utilizado. Bleser e Sibert (BLESER; SIBERT, 1990) foram mais além e desenvolveram uma ferramenta interativa de design para orientar a escolha de dispositivos de entrada baseando-se em regras heurísticas e reconhecimento de padrões.

Apesar dos modelos teóricos e das taxonomias terem ajudado a estabelecer um quadro para compreender e escolher dispositivos de entrada, eles eram incompletos. Enquanto que a teoria de modelos independentes de dispositivos fornecia o ponto inicial para modelos de dispositivos de entrada, estudos e experiências futuras mostraram que sistemas considerados equivalentes por essa teoria, na prática eram completamente diferentes. As características cognitivas e perceptuais foram ignoradas até Jacob e Sibert (JACOB; SIBERT, 1992) desenvolverem um modelo baseado na teoria de Garner (GARNER; FELEOLDY, 1970) de processamento de “estruturas perceptuais” em espaços multidimensionais. A teoria de Garner descreve como objetos em espaços multidimensionais têm diferentes estruturas perceptuais, e como os observadores percebem os objetos diferentemente baseados nessas estruturas. O modelo de Jacob e Sibert mostrou que dispositivos de entrada e técnicas de entrada que possuem as mesmas estruturas perceptuais são mapeados com sucesso.

3.3 Utilização de diferentes dispositivos de entrada em ambientes virtuais 3D

A interação com dispositivos de entrada e saída está cada vez mais se aproximando da forma como se interage com o mundo real de forma natural e intuitiva. Apesar disso, existem vários sistemas 3D que mapearam tradicionais dispositivos 2D para ambientes 3D com sucesso (CHEN; MOUNTFORD; SELLEN, 1988; BIER, 1990; MACKINLAY; CARD; ROBERTSON, 1990; HOUDE, 1992). Portanto, é importante se realizar um estudo comparativo sobre a necessidade de se empregar dispositivos 3D ou 2D em ambientes virtuais.

Hinckley *et al.* (HINCKLEY *et al.*, 1997) compararam 3 dispositivos de entrada em tarefas de rotação 3D. Duas técnicas de interação para rotação utilizando o mouse 2D (*Virtual Sphere* e *Arball*), e dois dispositivos de entrada 3D (*3D Ball* e um rastreador de posição, Figura 3.24) foram utilizados para rotacionar um objeto até que sua orientação coincidissem com a de um objeto de controle. Eles observaram que, utilizando os dispositivos 3D, os sujeitos eram capazes de completar a rotação até 36% mais rápido. Os sujeitos também preferiram utilizar a *3D Ball* aos dispositivos 2D. Além disso, Hinckley *et al.*

também observaram que não há diferença de precisão entre os dispositivos.

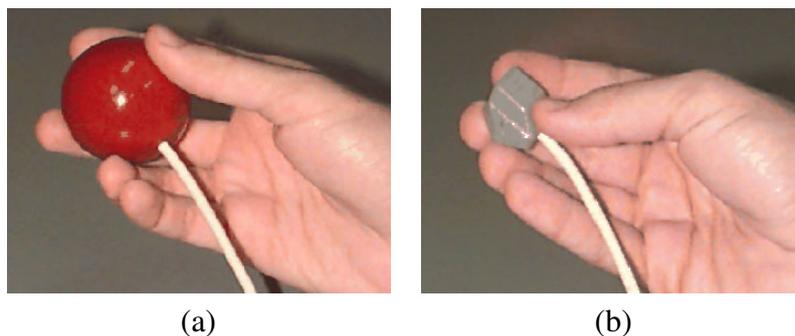


Figura 3.24: Dispositivos avaliados por Hinckley (HINCKLEY et al., 1997) para comparar técnicas de interação 2D e 3D.

Ware e Jessome (WARE; JESSOME, 1988) compararam um *3D Bat* com um mouse convencional para a manipulação de objetos 3D. Os sujeitos acharam a manipulação dos objetos 3D mais fácil com o *3D Bat* do que com o mouse convencional. Ware e Jessome sugeriram que essa preferência ocorre devido a necessidade dos sujeitos terem de quebrar mentalmente as tarefas 3D em tarefas 2D com a utilização do mouse.

Hinckley *et al.* e Ware e Jessome ilustraram a necessidade de se utilizar dispositivos de entrada 3D em ambientes virtuais, e mostraram que dispositivos 3D podem obter maior sucesso que dispositivos tradicionais de interação em tarefas 3D. Apesar disso, Jacob e Sibert (JACOB; SIBERT, 1992) mostraram que dispositivos de entrada 3D não são sempre os melhores dispositivos para realizar tarefas 3D, e o melhor dispositivo pode depender das “estruturas perceptuais” de cada tarefa.

Jacob e Sibert compararam o desempenho de duas tarefas nas quais os sujeitos deveriam manipular objetos utilizando um rastreador de posição 3D e um mouse 2D convencional. Na primeira tarefa, os sujeitos deveriam igualar a posição (em coordenadas x e y) e o tamanho de dois quadrados. Na segunda tarefa, os sujeitos deveriam igualar a posição (também em coordenadas x e y) e o tom de cinza (cor) dos quadrados. Os sujeitos realizaram a primeira tarefa mais rápido utilizando o rastreador 3D, e a segunda tarefa foi realizada mais rápida com a utilização do mouse convencional. A hipótese desse trabalho é que a escolha do melhor dispositivo de entrada está relacionada com a diferenciação que os sujeitos fazem das estruturas perceptuais das tarefas. Na primeira tarefa, os parâmetros (x , y , tamanho) estão relacionados (*integral attributes*, ou seja, são parâmetros integrados, espaciais), e na segunda, os parâmetros (x , y , e cor) são independentes (*separable attributes*, atributos distintos perceptualmente, espacial e de característica). Portanto, na primeira tarefa, a utilização do dispositivo de entrada 3D é mais intuitiva, enquanto que na segunda tarefa é o 2D.

Shaw (SHAW, 1998) comparou duas técnicas de interação 3D, utilizando uma ou duas mãos (THRED, *Two-Handed Refining Editor*), e um mouse 2D. Ambas as técnicas de interação 3D utilizavam rastreadores de posição como dispositivos de entrada. Shaw comparou várias características dos três dispositivos, incluindo performance, preferência dos usuários e fadiga, para realizar duas tarefas distintas. As tarefas eram a modelagem de um retângulo 2D e de uma caixa 3D, respectivamente. A preferência geral foi pela utilização da técnica de interação 3D com as duas mãos (THRED). Entretanto, quando as tarefas foram quebradas em componentes individuais, foi possível observar que os usuários preferiram o mouse para a seleção de vértices e menus. Shaw sugeriu que isto

pode ser devido ao alto grau de precisão do mouse convencional. Não houve diferença significativa em relação à fadiga dos sujeitos.

A utilização de ambas as mãos para controlar dois dispositivos de entrada ao mesmo tempo vem sendo freqüentemente empregada como uma técnica de melhorar a eficiência da interação humano-computador. Isto fornece ao usuário a possibilidade de realizar duas sub-tarefas em paralelo (BUXTON; MEYERS, 1986), e também transfere a habilidade de manipulação de ferramentas com duas mãos da vida real para o computador com pouco ou nenhum treino (KABBASH; BUXTON; SELLEN, 1994). Interação bi-manual também pode ajudar usuários a melhor perceber o ambiente 3D, pois aproveita a habilidade inata das pessoas saberem a posição de sua mão em relação à outra (HINCKLEY et al., 1994; SACHS; ROBERTS; STOOPS, 1991). Leganchuck *et al.* (LEGANCHUK; ZHAI; BUXTON, 1998) também percebeu que a utilização de ambas as mãos em uma aplicação melhora a capacidade manual e cognitiva e fornece uma perspectiva complementar, explorando os benefícios potenciais de aplicações rotineiras.

3.4 Síntese do Capítulo

Este capítulo apresentou as técnicas de interação freqüentemente empregadas em ambientes virtuais e seu enquadramento como diferentes metáforas e na taxonomia de Bowman (Figura 3.1). Os dispositivos de entrada existentes foram brevemente revisados, assim como os diferentes modelos teóricos de dispositivos virtuais. Finalmente, introduziu-se a necessidade de investigar as vantagens e desvantagens dos dispositivos frente a tarefas a serem realizadas no espaço virtual. No presente trabalho, esta questão é tratada no contexto de recorte volumétrico.

Como o objetivo do presente trabalho é propor ferramentas simples em hardware mais comum, os dispositivos adotados nesta avaliação foram o Magellan Spacemouse (Figura 3.19b), porque ele tem seis graus de liberdade (degrees of freedom, DOFs) e é relativamente simples e de fácil aprendizagem, e um mouse convencional, pois é o dispositivo de entrada mais amplamente utilizado. Sua usabilidade em ferramentas de recorte volumétrico é avaliada no Capítulo 6.

4 RECORTE VOLUMÉTRICO

O desenvolvimento das ferramentas de recorte volumétrico implementadas nesse trabalho foi feito utilizando-se as técnicas de mapeamento de textura para visualização direta de volumes de dados discutidas no Capítulo 2. As técnicas de interação aplicadas a essas ferramentas foram estudadas seguindo as metáforas de interação apresentadas no Capítulo 3. Neste capítulo é mostrada a visão geral da arquitetura do sistema desenvolvido e as diferentes implementações das ferramentas propostas.

4.1 Arquitetura do Sistema

As ferramentas de recorte volumétrico desenvolvidas foram integradas em um sistema de visualização volumétrica direta utilizando mapeamento de texturas. Seguindo a notação descrita por Ikits (IKITS et al., 2004), o *pipeline* de visualização volumétrica pode ser dividido em três estágios: *inicialização*, *atualização* e *desenho*. Os estágios de atualização e desenho são ilustrados na Figura 4.1.

No estágio de inicialização, o conjunto de dados volumétricos é processado e armazenado como uma textura 3D (chamada *VolumeTex*) na GPU. Durante esse processo, se necessário, uma pré-classificação dos valores dos voxels de acordo com uma função de transferência (KINDLMANN; DURKIN, 1998) ou cálculos de valores escalares e gradientes para iluminação e sombreado (WEISKOPF; ENGEL; ERTL, 2003) podem ser usados.

Atualizações são necessárias quando o usuário interage com o sistema, tanto alterando parâmetros de visualização quanto recortando o volume. As mudanças nos parâmetros de visualização são feitas através da manipulação da câmera, e o recorte é produzido por qualquer ferramenta. As ferramentas são implementadas por algoritmos de processamento de fragmentos (*shaders*) que executam na GPU e são utilizadas para descartar fragmentos da imagem final.

As ferramentas desenvolvidas no presente trabalho seguem o paradigma proposto por Weiskopf *et al.* (WEISKOPF; ENGEL; ERTL, 2002). É utilizada uma máscara de visibilidade armazenada em uma textura 3D (*SculptMap*, na Figura 4.1) para indicar quais voxels da textura de dados original serão visualizados. As ferramentas de recorte são aplicadas à essa máscara de visibilidade. Para realizar isso, as ferramentas são associadas a certas primitivas geométricas que são geradas por programas de fragmento na GPU (a geometria e interação dessas ferramentas são discutidas na Seção 4.2). Através da voxelização das ferramentas na GPU (DIETRICH et al., 2004), é possível modificar as suas geometrias de recorte interativamente. Quando isto é feito na CPU (WEISKOPF; ENGEL; ERTL, 2002), as ferramentas dificilmente podem ser voxelizadas em tempo real. A voxelização na CPU deve ser feita durante o estágio de inicialização e, após, as ferramen-

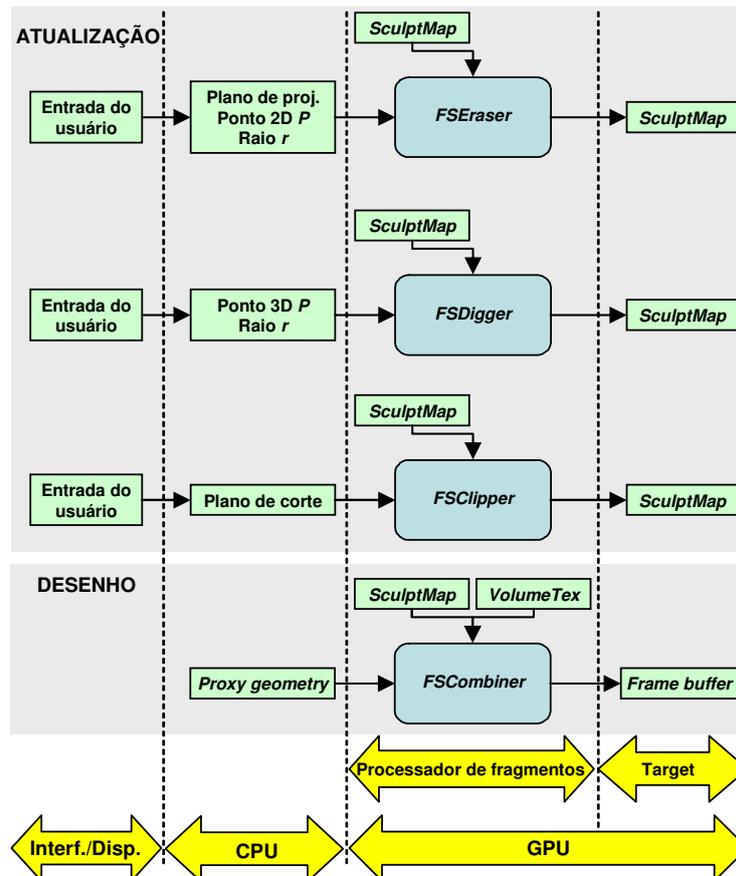


Figura 4.1: Arquitetura do sistema. Primeiro, a fase de atualização é utilizada para definir um volume de seleção (*SculptMap*) baseado na ferramenta atual de recorte. A última fase corresponde ao passo de desenho que gera a visualização final ao mesmo tempo que descarta as regiões definidas na máscara *SculptMap*.

tas devem ser armazenadas em diferentes *buffers* de textura. Isto se torna inviável quando o aplicativo utiliza grandes volumes de dados, como em aplicativos médicos. Além disso, a modificação das geometrias de recorte dessas ferramentas é limitada a rotações, translações e escalas nas suas coordenadas de textura, diminuindo provavelmente a precisão do recorte.

Quando uma ferramenta de recorte é aplicada a um volume, os voxels que se encontram dentro da geometria associada a cada ferramenta são marcados para serem removidos durante o estágio de desenho (veja o estágio de desenho na Figura 4.1). Isto é feito utilizando-se os valores de opacidade dos voxels equivalentes na *SculptMap* como flags de visibilidades. Os voxels a serem eliminados têm os seus valores de opacidade alterados para 0, enquanto que os voxels que devem ser visualizados tem seus valores fixados em 1 quando uma abordagem binária é utilizada. Essas operações são feitas pelos programas de fragmento *FSDigger*, *FSEraser* e *FSClipper*. Baseado no trabalho de Weiskopf *et al.* (WEISKOPF; ENGEL; ERTL, 2002), essa abordagem também foi modificada para evitar o serrilhamento presente em alguns volumes. Isto pode ser feito através do armazenamento de distâncias euclidianas ao ponto mais próximo da ferramenta de corte e da interpolação tri-linear durante o *rendering*.

A visualização do volume é feita através da amostragem da textura 3D por um conjunto de fatias (*proxy-geometry*) alinhadas à direção de observação (veja o estágio de desenho

na Figura 4.1). Durante o *rendering*, esses planos são desenhados ordenadamente. Entretanto, antes de ser feito o *rendering* desses polígonos, os parâmetros de *rendering* são configurados. Neste sistema, a combinação é realizada no programa de fragmento chamado *FSCCombiner*, ilustrado na Figura 4.1 e descrito na Figura 4.2. Os parâmetros de entrada são duas texturas 3D – o volume original representado no diagrama da Figura 4.1 como *VolumeTex*, e a informação de visibilidades armazenada em *SculptMap*. A imagem final no *framebuffer* é produzida combinando o *VolumeTex* com o *SculptMap*. Como frequentemente, por questões de desempenho, a dimensão da máscara de recorte é menor que o volume original, a interpolação tri-linear de *SculptMap* é normalmente feita durante o processo de combinação.

<pre> struct vertex2fragment { float2 Tex0: TEXCOORD0; }; void main(vertex2fragment v2f, uniform sampler3D SculptMap: TEXUNIT0, uniform sampler3D VolumeTex: TEXUNIT1, out float4 Col : COLOR0) { // Texture mapping float4 mask_voxel = tex3D(SculptMap, v2f.Tex0); float4 volume_voxel = tex3D(VolumeTex, v2f.Tex0); // Textures combination Col.rgb = volume_voxel.rgb; Col.a = volume_voxel.a * mask_voxel.a; } </pre>	<pre> struct vertex2fragment { float2 Tex0: TEXCOORD0; }; void main(vertex2fragment v2f, uniform sampler3D SculptMap: TEXUNIT0, uniform sampler3D VolumeTex: TEXUNIT1, out float4 Col : COLOR0) { // Textures mapping float4 mask_voxel = tex3D(SculptMap, v2f.Tex0); float4 volume_voxel = tex3D(VolumeTex, v2f.Tex0); // Textures combination Col.rgb = volume_voxel.rgb; Col.a = volume_voxel.a * floor(mask_voxel.a); } </pre>
(a)	(b)

Figura 4.2: Programas de fragmentos *FSCCombiner*: (a) Abordagem com máscara de visibilidade binária; (b) Abordagem com máscara de visibilidade contendo distâncias euclidianas.

A Figura 4.2 mostra os passos executados pelo *FSCCombiner* em duas abordagens diferentes: utilizando máscara de visibilidade binária (Figura 4.2a) e utilizando máscara com distâncias euclidianas (Figura 4.2b). Na primeira, como os valores de alfa dos voxels da máscara são “0”s e “1”s, só é necessário multiplicá-los pelos valores de alfa da textura (lembrando da possibilidade de nem todos os voxels do volume original serem visíveis), ou seja, quando o valor da máscara for “0”, o voxel da textura original é removido, e quando for “1”, ele é mantido. Já na segunda abordagem, os valores presentes na máscara de visibilidade encontram-se no intervalo fechado de “0” a “1”, conforme os exemplos ilustrados na Figura 4.3. Portanto, todos os valores menores que “1” são arredondados para “0”.

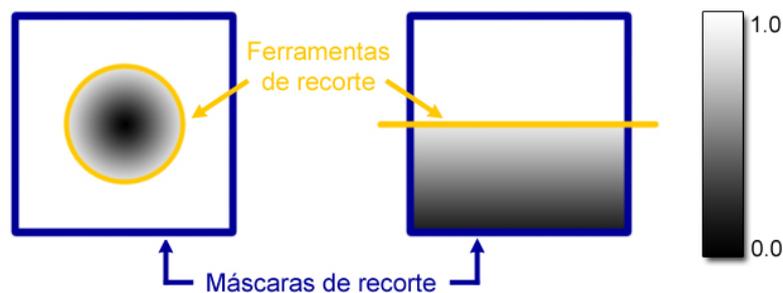


Figura 4.3: Exemplos dos valores das distâncias euclidianas armazenados na máscara de visibilidade.

A Figura 4.4 ilustra um exemplo prático do recorte de um volume contendo uma esfera vermelha (*VolumeTex*). Neste exemplo, é mostrado como cada passo do *pipeline* da Figura 4.1 afeta o volume e a máscara de visibilidade (*SculptMap*) até o resultado ser visualizado no *frame buffer*. A ferramenta utilizada é manipulada por um mouse 3D e tem o formato de um plano que corta o volume pela metade eliminando uma das partes. Essa ferramenta, chamada Guilhotina 3D, é explicada posteriormente na Seção 4.3.3.

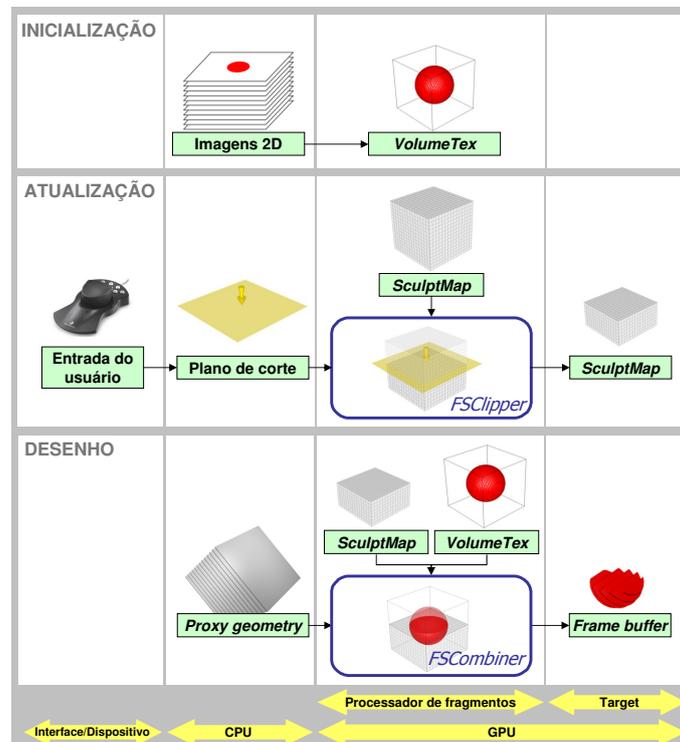


Figura 4.4: Exemplo prático dos passos executados para recortar a metade de um volume contendo uma esfera vermelha.

4.2 Ferramentas de Recorte

O recorte volumétrico pode ser essencialmente tratado como uma tarefa de seleção. O recorte de voxels é realizado pela seleção e, na seqüência, remoção desses voxels. A representação das técnicas de seleção é dividida em duas categorias básicas: *Apontador Virtual* e *Mão Virtual*. Com a metáfora de apontadores virtuais, o usuário seleciona objetos apontando para eles. As técnicas baseadas nesta metáfora podem se diferenciar no formato dos apontadores, na definição da direção do apontador, e nos métodos de eliminação de ambigüidade dos objetos que o usuário deseja selecionar. Com a mão virtual, o usuário “toca” os objetos com uma representação virtual de sua mão real. A escolha dos dispositivos e o mapeamento entre a posição e orientação da mão real e virtual são os diferenciais entre essas técnicas.

Com o objetivo de comparar ambas as metáforas de interação para o recorte volumétrico, foi desenvolvido um conjunto de ferramentas utilizando diferentes técnicas de interação. As geometrias dessas ferramentas de recorte são especificadas através do envio de poucos parâmetros para o programa de fragmentos da GPU. Apesar de terem sido escolhidas primeiramente para melhor se adequar ao recorte de volumes médicos, elas são suficientemente genéricas para serem utilizadas em outras aplicações, pois implementam

funções básicas de recorte que podem simular outras geometrias mais complexas através de algumas interações. Por exemplo, uma elipse pode ser simulada através de algumas interações com um círculo (Figura 4.5).

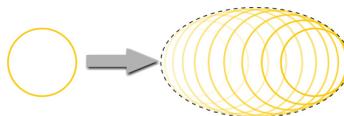


Figura 4.5: Elipse formada através de algumas interações com um círculo

Primeiramente, devido ao fato da utilização de ponteiros virtuais ser essencialmente uma operação 2D (BOWMAN; JOHNSON; HODGES, 2001), foi implementado um conjunto de ferramentas de recorte 2D utilizando o paradigma de “pintura”. Este paradigma é utilizado na maioria dos softwares de edição de imagens, onde o usuário controla canetas, pincéis, borrachas e outros instrumentos artísticos para pintar ou apagar em uma tela virtual. As ferramentas desenvolvidas que utilizam esse paradigma são chamadas *Borracha 2D*, *Escavadeira 2D*, e *Guilhotina 2D* (Figura 4.6). O desenvolvimento da Borracha 2D e da Escavadeira 2D foram baseados na implementação de Dietrich *et al.* (DIETRICH *et al.*, 2004). Embora o usuário utilize essas ferramentas para recortar um volume tridimensional, usar essas ferramentas 2D é como editar uma imagem 2D normal.

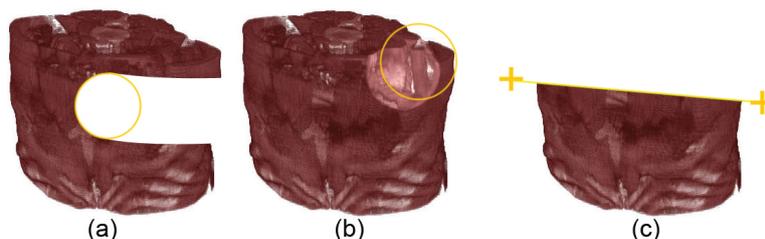


Figura 4.6: Ferramentas de recorte 2D: (a) Borracha 2D, (b) Escavadeira 2D e (c) Guilhotina 2D.

Após as ferramentas 2D, foram desenvolvidas versões 3D das mesmas ferramentas. Isso foi feito utilizando a metáfora da mão virtual. Nesta metáfora, a mão virtual é representada como diferentes cursores e seus formatos reproduzem em três dimensões as áreas de atuação das ferramentas 2D no volume 3D. Através da utilização dessa metáfora, a continuidade entre o mundo real e o virtual é aumentada. As ferramentas de recorte 3D desenvolvidas são chamadas de *Borracha 3D*, *Escavadeira 3D* e *Guilhotina 3D* (Figura 4.7).

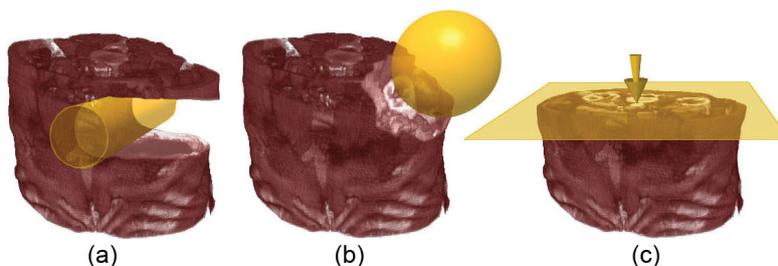


Figura 4.7: Ferramentas de Recorte 3D: (a) Borracha 3D, (b) Escavadeira 3D e (c) Guilhotina 3D

4.3 Recorte Volumétrico Utilizando Apontadores Virtuais

Conforme discutido no Capítulo 2, existem várias maneiras de se implementar seleção de objetos utilizando apontadores virtuais. Todas elas são técnicas de interação 2D onde o conceito de apontar objetos em imagens bi-dimensionais é estendido para ambientes tri-dimensionais. No sistema de recorte volumétrico, como os objetos que se deseja selecionar (os voxels do volume a serem eliminados), apesar de estarem próximos ao observador, são muito pequenos, uma variante da técnica de *ray-casting* foi utilizada (DIETRICH et al., 2004). Para a implementação das ferramentas Borracha 2D e Escavadeira 2D foi utilizada uma variação da técnica de seleção chamada *Aperture Selection* (FORSBERG; HERNDON; ZELEZNIK, 1996). Esta técnica, inspirada na técnica *spotlight* (LIANG; GREEN, 1994), utiliza um cone de apontamento, ao invés de um raio, para facilitar a seleção de objetos pequenos. Na versão original dessa técnica, o volume do cone é controlado pela aproximação ou afastamento de um círculo de abertura em relação ao ponto de visão (Figura 4.8a). Na versão utilizada no presente trabalho, a distância do círculo é fixa, e o volume do cone é modificado através da diminuição ou aumento de seu raio (Figura 4.8b). Esse círculo de abertura é utilizado como cursor nas ferramentas de Borracha 2D e Escavadeira 2D, e sua manipulação é feita como em editores de imagens 2D convencionais. O usuário recorta o volume 3D como se fosse uma imagem 2D. A ferramenta de Guilhotina 2D foi desenvolvida utilizando-se a técnica de *ray-casting* tradicional para determinar a região de recorte. O funcionamento de cada uma das ferramentas que utilizam técnicas de interação 2D para realizar o recorte volumétrico é discutido a seguir.

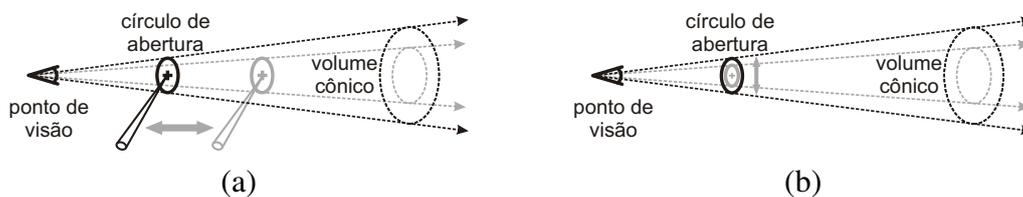


Figura 4.8: Técnicas com volumes de seleção cônicos: (a) *Aperture Selection*; (b) variação proposta.

4.3.1 Borracha 2D

Como mencionado anteriormente, a Borracha 2D utiliza uma variação da técnica de *Aperture Selection* (FORSBERG; HERNDON; ZELEZNIK, 1996). Na *aperture selection* o cursor é um círculo alinhado com o plano de visão e a origem do apontador virtual é posicionada no mesmo local do ponto de visão do usuário (Figura 4.9). A direção do vetor deste apontador é o vetor que possui a origem no ponto de visão do usuário e passa através do centro do cursor. Esse vetor é chamado de “linha de visão”, ou “direção de observação”. O tamanho do raio do cursor r e a sua posição no plano de visão P são estabelecidos pelo usuário. A posição é especificada através da seleção de um pixel na tela. O tamanho e a posição 2D do cursor no plano de visão, juntos com a sua distância do ponto de visão, determinam o alcance e o tamanho do volume de seleção, que possui o formato de um cone. Quando a borracha 2D é aplicada no volume, a posição de cada voxel é testada contra o volume de seleção. Se estiver dentro, o voxel é removido. O teste é realizado através da projeção perspectiva do centro de cada voxel no plano de visão. Se a distância euclidiana d entre o voxel projetado v e o centro do cursor P for menor que o raio r do cursor, o voxel está dentro do volume.

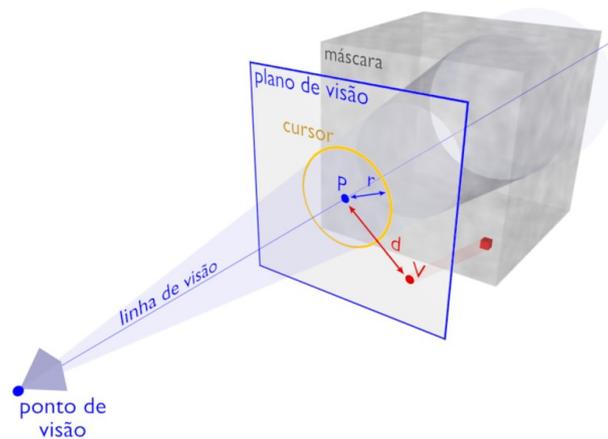


Figura 4.9: Borracha 2D baseada na projeção de voxels no plano de visão

O programa de fragmentos que realiza as operações da Borracha 2D é chamado *FSE-raser* (Figura 4.10). Este programa recebe como parâmetros de entrada o plano de visão, a posição 2D P do cursor no plano de visão, o raio r do cursor, e a máscara de visualização *SculptMap*. Para cada fragmento das fatias do volume (*proxy-geometry*), os passos mostrados nos programas de fragmentos da Figura 4.10 são executados.

<pre> struct vertex2fragment { float2 Tex0: TEXCOORD0; }; void main(vertex2fragment v2f, uniform float4x4 ProjPlane : C0, //P.xy = 2D Point 'P' P.z = tool radius 'r' uniform float3 P : C4, uniform sampler3D SculptMap : TEXUNIT0, uniform sampler2D PolygonCoord: TEXUNIT1, out float4 Col : COLOR0) { // Texture mapping float4 texCoord = tex2D(PolygonCoord, v2f.Tex0); Col = tex3D(SculptMap, texCoord.xyz); // Projection float4 PlaneCoordinates = mul(float4(texCoord.xyz - 0.5f, 1.0f), ProjPlane); PlaneCoordinates.xyz /= PlaneCoordinates.w; // 2D Euclidean distance float D = length(PlaneCoordinates.xy - P.xy); if (D < P.z) { Col.a = 0.0f; } } </pre>	<pre> struct vertex2fragment { float2 Tex0: TEXCOORD0; }; void main(vertex2fragment v2f, uniform float4x4 ProjPlane : C0, //P.xy = 2D Point 'P' P.z = tool radius 'r' uniform float3 P : C4, uniform sampler3D SculptMap : TEXUNIT0, uniform sampler2D PolygonCoord: TEXUNIT1, out float4 Col : COLOR0) { // Texture mapping float4 texCoord = tex2D(PolygonCoord, v2f.Tex0); Col = tex3D(SculptMap, texCoord.xyz); // Projection float4 PlaneCoordinates = mul(float4(texCoord.xyz - 0.5f, 1.0f), ProjPlane); PlaneCoordinates.xyz /= PlaneCoordinates.w; // 2D Euclidean distance float D = length(PlaneCoordinates.xy - P.xy) / P.z; if (D < Col.a) { Col.a = D; } } </pre>
(a)	(b)

Figura 4.10: Programas de fragmentos das ferramentas borrachas: (a) Abordagem com máscara de visibilidade binária; (b) Abordagem com máscara de visibilidade contendo distâncias euclidianas.

O hardware gráfico existente até pouco tempo atrás não permitia que programas de fragmentos retornassem texturas 3D. Por isso, primeiramente, é feito um mapeamento de texturas para gerar uma versão “plana” da textura 3D que possa ser gravada no *frame buffer*. Nesta abordagem, proposta em (DIETRICH et al., 2004), os voxels são ordenados na textura 2D para poderem ser copiados diretamente do *frame buffer* para a memória 3D. A Figura 4.11 ilustra um exemplo da disposição dos voxels de uma máscara de recorte de $4 \times 4 \times 4$ em uma textura 2D de 8×8 .

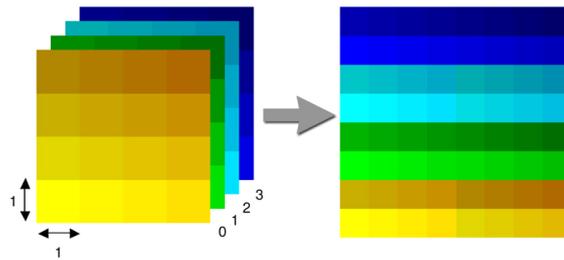


Figura 4.11: Organização das fatias da textura 3D na textura 2D intermediária, onde cada “quadrado” das figuras corresponde a um voxel (DIETRICH et al., 2004).

Posteriormente, no programa de fragmentos, é feita a projeção do voxel atual no plano de visão, a distância euclidiana desse voxel projetado até o centro do cursor é calculada e por último, é retornado o valor de visibilidade desse voxel. Na abordagem com máscara de visibilidade binária (Figura 4.10a), se a distância do voxel for menor que o raio do cursor, o seu valor de alfa é atualizado para zero, ou seja, o voxel é totalmente removido. Já na abordagem utilizando distâncias euclidianas para evitar serrilhamento (Figura 4.10b), primeiramente a distância do voxel atual até o cursor é normalizada para valores ente “0” e “1”, conforme ilustrado na Figura 4.3, e após, essa distância é armazenada no valor alfa do voxel caso o seu valor seja maior.

4.3.2 Escavadeira 2D

Em recorte de volumes, muitas vezes é necessário “cavar um buraco” no volume ao invés de remover todos os voxels em profundidade como é feito pela ferramenta Borracha 2D. Isto pode se realizado através da ferramenta Escavadeira 2D. A Escavadeira 2D considera os mesmos voxels da borracha 2D para serem eliminados (os voxels dentro de um círculo), mas somente aqueles dentro de um raio de ação que parte do primeiro ponto P_i da superfície do volume interseccionado pela linha de visão (Figura 4.12). Esse raio de ação é chamado de *profundidade* da escavadeira 2D. O cursor dessa ferramenta, assim como na borracha virtual, é um círculo que pode ser translacionado sobre o plano de visão e a sua profundidade é determinada em relação ao raio do cursor r , a distância do primeiro ponto da superfície interseccionado pela linha de visão ds e a distância do plano de visão dv até o ponto de visão, como mostrado na Equação 4.1:

$$profundidade = \frac{ds}{dv} \times r \quad (4.1)$$

Quando a escavadeira 2D é aplicada, ela funciona como uma “bomba que explode” na superfície do volume e elimina todos os voxels dentro da região delimitada pela profundidade. Isto é feito selecionando-se o primeiro voxel atravessado pela linha de visão na superfície do volume e calculando a distância euclidiana 3D d até todos os outros voxels do volume. Se essa distância for menor que a profundidade da ferramenta, os voxels são apagados.

O programa de fragmentos que implementa a Escavadeira 2D é chamado *FSDigger* (Figura 4.13), e recebe como parâmetros de entrada a posição 3D P do cursor, o raio r do cursor, e a máscara de visibilidade *SculptMap*. O programa *FSDigger*, que é executado para cada fragmento da *proxy-geometry* é descrito nas duas abordagens (binária e com distâncias euclidianas) na Figura 4.13.

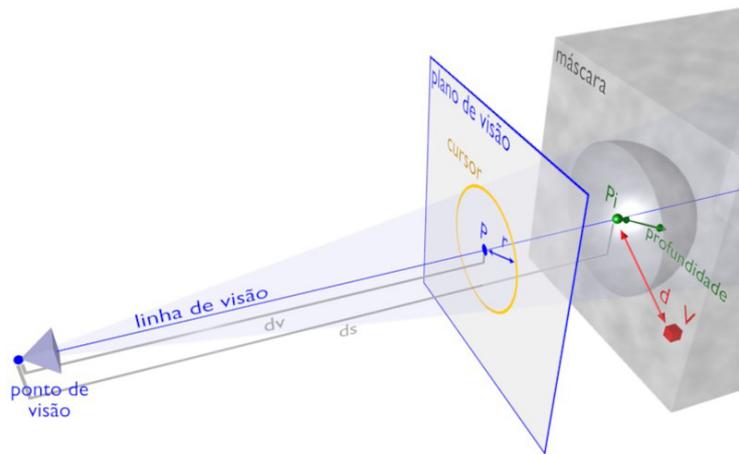


Figura 4.12: Escavadeira 2D baseada na distância euclidiana 2D

<pre> struct vertex2fragment { float2 Tex0: TEXCOORD0; }; void main(vertex2fragment v2f, //P.xyz = 3D Point 'P' P.a = tool radius 'r' uniform float4 P : C0, uniform sampler3D SculptMap : TEXUNIT0, uniform sampler2D PolygonCoord: TEXUNIT1, out float4 Col : COLOR0) { // Texture mapping float4 texCoord = tex2D(PolygonCoord, v2f.Tex0); Col = tex3D(SculptMap, texCoord.xyz); // 3D Euclidean distance float D = distance(float4(P.xyz,1), texCoord); if (D < P.a) { Col.a = 0.0f; } } </pre>	<pre> struct vertex2fragment { float2 Tex0: TEXCOORD0; }; void main(vertex2fragment v2f, //P.xyz = 3D Point 'P' P.a = tool radius 'r' uniform float4 P : C0, uniform sampler3D SculptMap : TEXUNIT0, uniform sampler2D PolygonCoord: TEXUNIT1, out float4 Col : COLOR0) { // Texture mapping float4 texCoord = tex2D(PolygonCoord, v2f.Tex0); Col = tex3D(SculptMap, texCoord.xyz); // 3D Euclidean distance float D = distance(float4(P.xyz,1), texCoord) / P.a; if (D < Col.a) { Col.a = D; } } </pre>
(a)	(b)

Figura 4.13: Programas de fragmentos das ferramentas escavadeiras: (a) Abordagem com máscara de visibilidade binária; (b) Abordagem com máscara de visibilidade contendo distâncias euclidianas.

4.3.3 Guilhotina 2D

Diferentemente das outras ferramentas de recorte 2D, o cursor da Guilhotina 2D não é definido por um círculo, mas por um segmento de linha sobre o plano de visão (Figura 4.14). A Guilhotina 2D utiliza a técnica tradicional de *ray-casting* para projetar dois pontos que o usuário seleciona na tela no plano de visão. Estes pontos são utilizados para desenhar um segmento de linha que representa o cursor da ferramenta de recorte. Esse cursor corta a imagem na posição definida pelo usuário eliminando um dos lados. A Guilhotina 2D define uma região convexa dentro do volume original 3D. O recorte dessa ferramenta é realizado através do cálculo de um plano de corte utilizando três pontos: o ponto de visão e os dois pontos projetados no plano de visão. A seleção de um voxel para eliminação é feita através do cálculo da menor distância do centro do voxel até o plano. Essa distância é calculada por um produto escalar entre as coordenadas do voxel e o vetor normal ao plano. O voxel é removido se a distância for negativa. Embora seja utilizado um plano de corte e distâncias tri-dimensionais, devido a projeção perspectiva, o usuário visualiza isso como uma operação 2D.

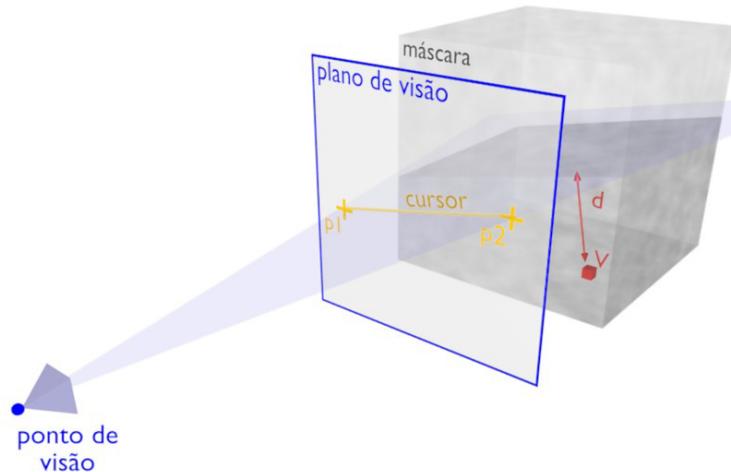


Figura 4.14: Guilhotina 2D baseado na distância de voxels a um plano de recorte

O programa de fragmentos que implementa a Guilhotina 2D é o *FSClipper*, ilustrado na Figura 4.1 e descrito na Figura 4.15. Este programa recebe como parâmetros de entrada os coeficientes do plano de corte e a máscara de visibilidade *SculptMap*.

<pre> struct vertex2fragment { float2 Tex0: TEXCOORD0; }; void main(vertex2fragment v2f, uniform float4 clipPlane : C0, uniform sampler3D SculptMap : TEXUNIT0, uniform sampler2D PolygonCoord: TEXUNIT1, out float4 Col : COLOR0) { // Texture mapping float4 texCoord = tex2D(PolygonCoord, v2f.Tex0); Col = tex3D(SculptMap, texCoord.xyz); // Distance D = clamp(dot(clipPlane, float4(texCoord.xyz-0.5f, 1.0f))+1.0f, 0.0f, 1.0f) if (D < 0) { Col.a = 0; } } </pre>	<pre> struct vertex2fragment { float2 Tex0: TEXCOORD0; }; void main(vertex2fragment v2f, uniform float4 clipPlane : C0, uniform sampler3D SculptMap : TEXUNIT0, uniform sampler2D PolygonCoord: TEXUNIT1, out float4 Col : COLOR0) { // Texture mapping float4 texCoord = tex2D(PolygonCoord, v2f.Tex0); Col = tex3D(SculptMap, texCoord.xyz); // Distance D = clamp(dot(clipPlane, float4(texCoord.xyz-0.5f, 1.0f))+1.0f, 0.0f, 1.0f); if (D < Col.a) { Col.a = D; } } </pre>
(a)	(b)

Figura 4.15: Programas de fragmentos das ferramentas guilhotinas: (a) Abordagem com máscara de visibilidade binária; (b) Abordagem com máscara de visibilidade contendo distâncias euclidianas.

4.4 Recorte Volumétrico Utilizando Mão Virtual

A forma mais simples de interação direta é o mapeamento direto entre os movimentos da mão e os movimentos do objeto no ambiente virtual (MINE, 1995). Com base nisso, foram criadas as técnicas de interação para o recorte volumétrico utilizando a metáfora da mão virtual. Nessas técnicas, uma representação virtual da mão real é utilizada para fazer a seleção ou manipulação de objetos no mundo virtual. A mão virtual nesses sistemas, não necessita ter o formato de uma mão propriamente. Outros cursores de diferentes formatos podem ser empregados. Baseado nisso e nas implementações 2D das ferramentas de recorte descritas na Seção 4.3, as ferramentas de recorte 3D utilizando a metáfora da mão

virtual foram desenvolvidas. Essas ferramentas são a *Borracha 3D*, a *Escavadeira 3D* e a *Guilhotina 3D* que são descritas a seguir nessa seção.

4.4.1 Borracha 3D

Observando a Figura 4.9, é possível perceber que a Borracha Virtual 2D é interpretada tri-dimensionalmente como um cone perpendicular ao plano de visão com ápice no ponto de visão do usuário e base no infinito. O formato de um cone é devido à projeção perspectiva aplicada na visualização do volume. O ângulo de visão é o ângulo de abertura do cone. Utilizando-se uma projeção ortogonal o formato resultante seria um cilindro. O uso de um cilindro faz mais sentido quando a metáfora de interação da mão virtual é utilizada, pois o cursor deve ser independente dos parâmetros de visão. Quando o volume de seleção não é restringido por parâmetros de visão, o usuário pode fazer rotações e translações desse volume livremente. Considerando isso, a ferramenta da Borracha 3D foi implementada com a utilização de um cilindro como cursor (Figura 4.16).

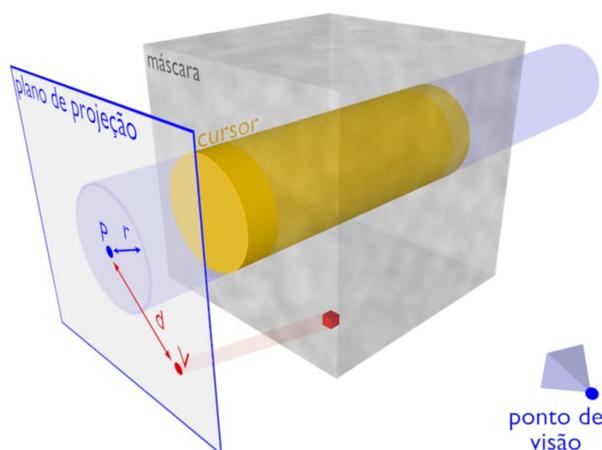


Figura 4.16: Borracha 3D baseada na projeção de voxels em um plano de projeção

No cálculo para verificar se um voxel está dentro da região volumétrica definida pelo cilindro da Borracha 3D, os mesmos parâmetros usados pela Borracha 2D são utilizados. Esses parâmetros são um plano de projeção, um ponto P neste plano e o raio r do cilindro. Ao contrário da Borracha 2D, o plano de projeção da Borracha 3D não é o plano de visão. Quando o cursor é rotacionado e transladado, estas transformações também são aplicadas ao plano de projeção. O cálculo para a eliminação dos voxels é feito primeiramente projetando-se ortogonalmente o centro de cada voxel no plano de projeção. Para cada centro projetado, a distância até P é calculada (chamada de distância projetada d do voxel). Esta distância é a distância euclidiana 2D, pois ela é calculada no plano de projeção. Após, todos os voxels com $d < r$ são removidos.

O programa de fragmentos que implementa a Borracha 3D é chamado *FSEraser* (o mesmo da Borracha 2D), o qual recebe como parâmetros de entrada o plano de projeção (e não o plano de visão como na Borracha 2D), o ponto 2D P , o raio r da ferramenta, e a máscara *SculptMap* (Figura 4.1). Para cada fragmento da *proxy-geometry* os passos presentes no programa de fragmentos da Figura 4.10 são realizados.

4.4.2 Escavadeira 3D

O volume de seleção da Escavadeira 2D pode ser representado tri-dimensionalmente como uma esfera com o seu centro posicionado na superfície do volume. O tamanho e posição dessa esfera são restringidos pelos parâmetros de visualização. O seu tamanho varia de acordo com a Equação 4.1 e a posição é determinada pela linha de visão. Eliminando-se a associação aos parâmetros de visão da Escavadeira 2D é possível modificar o tamanho e posição da esfera livremente. A Escavadeira virtual 3D foi desenvolvida utilizando-se esta última abordagem (Figura 4.17). O cursor utilizando mão virtual da escavadeira 3D é uma esfera com raio r e posição central P especificados pelo usuário. A seleção de um voxel para a eliminação é feita primeiramente através do cálculo da distância euclidiana 3D d do seu centro até P . Se essa distância é menor que r , este voxel é removido.

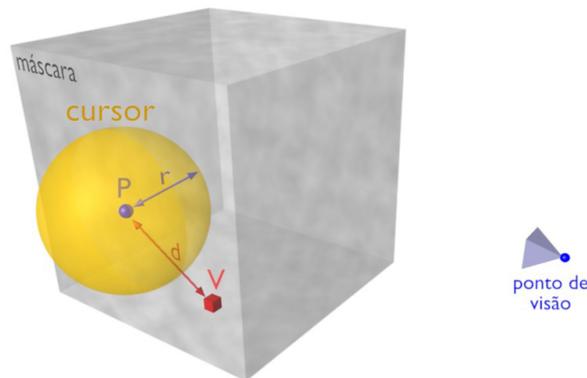


Figura 4.17: Escavadeira 3D baseada na distância euclidiana 3D

O programa de fragmentos que implementa a Escavadeira 3D é o *FSDigger* (o mesmo da Escavadeira 2D), e recebe como parâmetros de entrada o ponto P , o raio r e a máscara de textura *SculptMap* (Figura 4.1) especificados pelo usuário. O programa *FSDigger* executado para cada fragmento da *proxy-geometry* é mostrado na Figura 4.13).

4.4.3 Guilhotina 3D

A extensão da ferramenta do Plano de Corte 2D para utilizar a metáfora da mão virtual é simples. Ao invés de uma linha, o usuário pode manipular um quadrado como cursor para a ferramenta da Guilhotina 3D (Figura 4.18). Este quadrado representa um plano que o usuário pode translacionar e rotacionar livremente e, diferentemente da Guilhotina 2D, este plano não é preso ao ponto de visão do usuário. Quando o plano de corte definido pelo usuário é aplicado para a remoção de voxels, ele fatia o volume atual e cria um novo formato com uma face adicional. Este processo pode ser repetido quantas vezes forem necessárias (Figura 4.19) e o seu cálculo é feito utilizando o produto escalar entre as coordenadas do voxel e o vetor normal ao plano, como na ferramenta da Guilhotina 2D.

FSClipper é o programa de fragmento que implementa a ferramenta da Guilhotina 3D (o mesmo programa da Guilhotina 2D). Ele recebe como parâmetros de entrada os coeficientes do plano de recorte e a máscara de recorte *SculptMap* (Figura 4.1). O código em Cg para o *FSClipper* é mostrado na Figura 4.15.

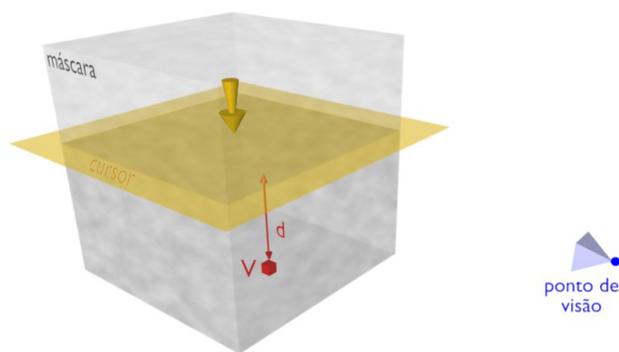


Figura 4.18: Guilhotina 3D baseado na distância de voxels a um plano de recorte

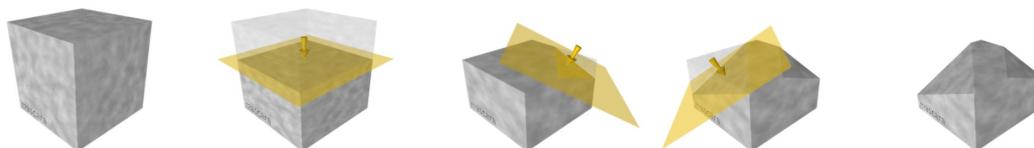


Figura 4.19: Volume cortado três vezes com a ferramenta Guilhotina

4.5 Síntese do Capítulo

Neste capítulo foram descritas as ferramentas de recorte implementadas para exploração de volumes. Enquanto as ferramentas 2D são baseadas na metáfora de apontador virtual, as 3D são baseadas na mão virtual. Os cursores para as ferramentas 2D são círculos (Borracha 2D e Escavadeira 2D) e uma linha (Guilhotina 2D), e para as ferramentas 3D são um cilindro (Borracha 3D), uma esfera (Escavadeira 3D) e um quadrado (Guilhotina 3D).

Assim como no modelo teórico de dispositivos de entrada proposto inicialmente por Foley e Wallace (FOLEY; WALLACE, 1974), as técnicas de interação dessas ferramentas foram desenvolvidas independentes dos dispositivos de entrada utilizados. Essa abordagem possibilita que as ferramentas sejam mapeadas até mesmo para dispositivos com diferentes graus de liberdade.

As técnicas de interação que utilizam metáforas 2D são mais comuns nos sistemas utilizados pela maioria dos usuários, como editores de texto e imagens, mas teoricamente as técnicas que utilizam metáforas 3D são mais naturais. Logo, para verificar quais das ferramentas descritas neste capítulo possuem o melhor desempenho, ambas as metáforas foram avaliadas (Capítulo 5).

Segundo trabalhos anteriores, descritos no Capítulo 3, a eficiência, intuitividade e facilidade de uso de um determinado sistema não são verificadas somente avaliando-se as técnicas de interação empregadas, mas também os dispositivos nelas mapeados. Portanto, para a avaliação das ferramentas apresentadas, também foram posteriormente testados diferentes dispositivos de entrada nelas mapeados (Capítulo 6).

5 AVALIAÇÃO EXPERIMENTAL DAS TÉCNICAS DE INTERAÇÃO

O primeiro experimento de avaliação realizado teve como objetivo medir o desempenho e preferência dos usuários com relação às duas classes de técnicas de interação descritas no Capítulo 4: 2D (apontadores virtuais) e 3D (mão virtual). A idéia motivadora da avaliação foi investigar o desempenho comparando metáforas 2D comuns com metáforas 3D que, teoricamente, seriam mais naturais.

5.1 Utilização das Ferramentas

O principal objetivo no desenvolvimento do aplicativo de avaliação das técnicas de interação para o recorte volumétrico foi torná-las intuitivas e fáceis de manipular utilizando somente um mouse convencional, pois a avaliação dos dispositivos de entrada para manipular as ferramentas de recorte foi feita separadamente (Capítulo 6). Nesta outra avaliação, como será visto, foi possível verificar que apesar da tendência dos usuários em preferir o mouse 3D para manipular ferramentas de recorte volumétrico e a performance desse dispositivo aumentar depois de um certo tempo de uso, o mouse 2D convencional gera melhores resultados nas primeiras vezes que as ferramentas são manipuladas. Isso ocorre mesmo quando a metáfora 3D da mão virtual é utilizada. A justificativa provável para esse resultado é o fato do mouse 2D ser o dispositivo de entrada mais comumente utilizado. Além disso, diferentes técnicas de interação permitem aos usuários realizar tarefas de formas diferentes mesmo utilizando-se do mesmo dispositivo de entrada (FOLEY; WALLACE; CHAN, 1984; HERNDON; DAM; GLEICHER, 1994).

Neste aplicativo, portanto, o volume e as ferramentas são controlados usando um mouse convencional. Para solucionar o problema de mapeamento de controle de dois objetos para um mesmo dispositivo, algumas alterações foram necessárias. A rotação do volume foi implementada através da operação “arrastar” normalmente utilizada nos softwares atuais. Essa operação é feita pressionando o botão direito do mouse e o movimentando. O botão esquerdo do mouse, por outro lado, foi utilizado para aplicar as ferramentas no volume (Figura 5.1).

A manipulação das ferramentas 2D pelo mouse convencional é trivial, pois ambos possuem os mesmos graus de liberdade (*Degrees of freedom*, DOFs) e funcionam como em aplicativos de edição de imagem convencionais. O maior problema foi com relação ao controle das ferramentas 3D utilizando somente o mouse 2D. Para resolver esse problema, a falta de DOFs do mouse 2D teve de ser compensada. Translações ao longo dos eixos x e y foram normalmente implementadas utilizando a movimentação do mouse 2D, mas a translação na dimensão z foi desenvolvida utilizando o botão de rolagem do mouse.

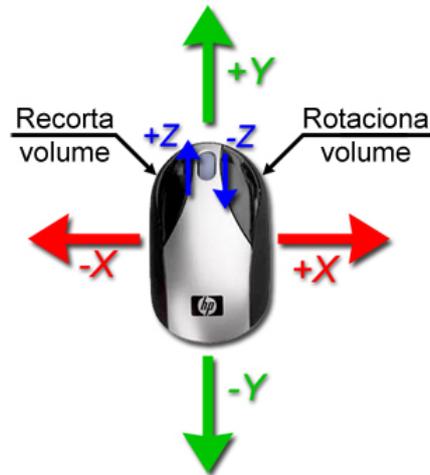


Figura 5.1: Interpretação dos movimentos do mouse para transladar as ferramentas. A rotação do volume é feita pressionando-se o botão direito. As ferramentas são aplicadas no volume para fazer o recorte pressionando-se o botão esquerdo.

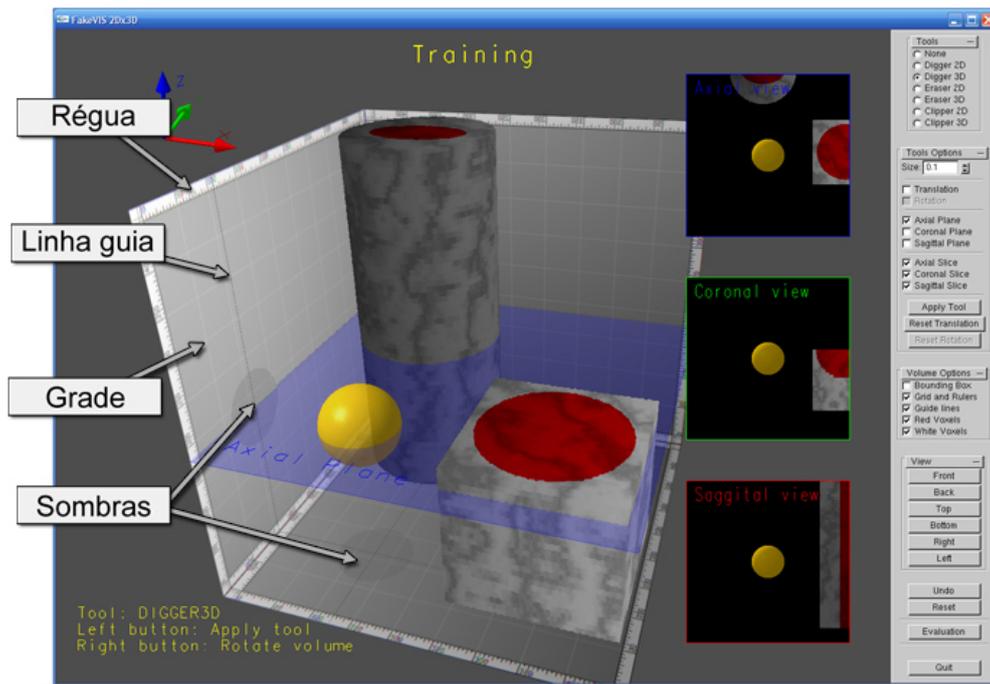


Figura 5.2: Interface do aplicativo de avaliação das técnicas de interação das ferramentas de recorte volumétrico.

Outro problema foi com relação ao retorno visual de profundidade ao usuário. Isto foi solucionado utilizando-se iluminação, sombras, linhas guias, grade e régua (Figura 5.2).

A interface do aplicativo fornecia alguns outros itens para ajudar o usuário. Fatias Axiais, Coronais e Sagitais do volume podiam ser habilitadas para mostrar estruturas internas do volume durante a utilização das ferramentas Escavadeiras. A Borracha 3D podia ser rotacionada e opções adicionais de translação também eram fornecidas.

5.2 Hipóteses

A avaliação experimental das técnicas de interação de ferramentas de recorte volumétrico foi baseada em três hipóteses, levando em consideração a comparação de técnicas de interação 2D e 3D, baseadas nas metáforas de apontamento virtual e mão virtual respectivamente:

- **H1.** Os usuários são mais rápidos usando as ferramentas 2D.
- **H2.** A taxa de erros diminui quando o usuário usa as ferramentas 3D.
- **H3.** Os usuários preferirão utilizar as ferramentas 3D para o recorte de volumes.

A primeira hipótese é baseada no fato que usuários comuns estão acostumados a utilizar o mouse 2D para executar tarefas tipicamente 2D em seu dia-a-dia e o período de treinamento fornecido ante da execução das tarefas não é suficiente para que eles adquiram a destreza necessária para a utilização das ferramentas 3D. A segunda hipótese, por outro lado, é baseada no fato que tarefas 3D possivelmente são melhores executadas utilizando uma metáfora de interação 3D, pois alguns fatores presentes no mapeamento entre a dimensão 2D para 3D são eliminados, como a projeção. Finalmente, a última hipótese considera vantajoso o fato da metáfora da mão virtual não necessitar nenhum tipo de mapeamento entre diferentes dimensões, diminuindo o esforço cognitivo por parte dos usuários.

5.3 Cenário

O experimento consistiu na eliminação de voxels vermelhos e cinzas de um conjunto de dados volumétricos. A aplicação de avaliação tinha como atributo de entrada diferentes conjuntos de dados volumétricos, cada um sendo um conjunto de imagens com pixels vermelhos e cinzas que descreviam o volume. Foram desenvolvidos cinco conjuntos de dados diferentes: um para treinar o usuário antes de começar o processo de avaliação (Figura 5.3a); três volumes para a análise separada das ferramentas (*Escavadeiras*, *Borrachas* e *Guilhotinas*), independentes das técnicas de interação aplicadas (Figura 5.3d); e um quinto volume para avaliar a utilização das três ferramentas concorrentemente, mas utilizando uma mesma técnica de interação (Figura 5.3e). Esses conjuntos de dados volumétricos eram opacos e, para facilitar a visualização de cortes e buracos no volume, uma textura procedural de mármore e iluminação foram utilizadas. A transparência dos volumes na Figura 5.3 foi utilizada para melhor ilustrar as estruturas internas dos volumes no presente trabalho.

5.4 Variáveis Dependentes e Independentes

Variáveis independentes são as variáveis do experimento que geram diferentes condições a serem comparadas. Foram utilizadas: idade; gênero; ocupação; experiência com dispositivos 2D e 3D; experiência com aplicações 2D e 3D (como editores de imagens, CAD, 3D Studio, Blender, etc.), e finalmente as oito tarefas (Tabela 5.1) que os usuários deveriam realizar. Essas variáveis, excluindo as oito tarefas, foram coletadas através de um questionário pré-teste, disponível no Apêndice A.

Variáveis dependentes são medidas durante a execução de tarefas. Elas podem ser objetivas, como tempo gasto na realização de uma determinada tarefa, ou subjetivas, como o

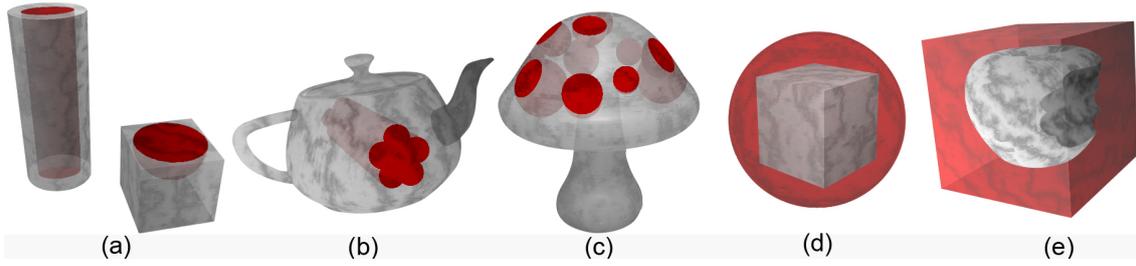


Figura 5.3: Volumes de dados para avaliação das técnicas de interação das ferramentas de recorte: (a) um cubo e um cilindro para a fase de treinamento; (b) um *teapot* para avaliação das ferramentas *Borrachas*; (c) um cogumelo para avaliação das ferramentas *Escavadeiras*; (d) um cubo para avaliação das ferramentas *Guilhotinas*; e uma maçã mordida para a avaliação das ferramentas simultaneamente.

Tabela 5.1: Número total de tarefas para avaliar as ferramentas independentemente e juntas utilizando técnicas de interação 2D e 3D.

Tarefas	Ferramentas	Volume
T1	Borracha 2D	<i>Teapot</i>
T2	Escavadeira 2D	Cogumelo
T3	Guilhotina 2D	Cubo
T4	Borracha 2D Escavadeira 2D Guilhotina 2D	Maçã
T5	Borracha 3D	<i>Teapot</i>
T6	Escavadeira 3D	Cogumelo
T7	Guilhotina 3D	Cubo
T8	Borracha 3D Escavadeira 3D Guilhotina 3D	Maçã

nível de satisfação. As variáveis dependentes coletadas neste experimento foram: tempo gasto para completar uma tarefa; taxa de erros; e técnica de interação preferida, esta última coletada após os testes através de questionários respondidos pelos usuários avaliados (Apêndice B).

O tempo gasto para a execução de uma tarefa foi medido a partir do início de uma tarefa, até o usuário terminá-la pressionando um botão de encerramento. A taxa de erros (Equação 5.1) levou em consideração a quantidade de erros depois de uma tarefa de recorte ser completada e o tempo gasto para realizar essa tarefa.

$$taxa_de_erro = \epsilon \times T \quad (5.1)$$

Nesta equação, T é o tempo gasto e ϵ é calculado pela Equação 5.2, onde V_ϵ é o número restante de voxels vermelhos e C_ϵ o número de voxels cinza removidos.

$$\epsilon = \frac{V_\epsilon + C_\epsilon}{\#voxels} \quad (5.2)$$

5.5 Sujeitos

Os testes foram realizados com dezesseis sujeitos, todos estudantes de Ciência da Computação e a maioria desenvolvendo algum trabalho na área de computação gráfica. Apesar disso, analisando os questionários pré-teste (Apêndice A), poucos possuíam conhecimento de aplicativos 3D. A maioria dos sujeitos eram homens, havendo somente três mulheres, e tinham entre 20 e 29 anos, sendo a média 23,5. Cada um deles testou as seis ferramentas (descritas no Capítulo 4) em 8 tarefas do experimento.

5.6 Metodologia e Procedimentos

Os testes foram realizados em um notebook HP Pavilion zd8000 com processador Intel Pentium 4 de 3.20GHz, 512MB de RAM, e placa gráfica ATI Mobility Radeon X600 PCI Express x16 com 256 MB.

Os seguintes passos foram aplicados na mesma ordem para todos os usuários:

- Instruções sobre o aplicativo e os dispositivos foram fornecidas;
- Os usuários preencheram o questionário de pré-teste presente no Apêndice A;
- Os usuários podiam treinar antes de iniciar a avaliação sem limite de tempo;
- As oito tarefas eram realizadas em uma ordem pré-definida;
- O arquivo de *log* era atualizado;
- O questionário pós-teste, mostrado no Apêndice B, era preenchido pelos usuários.

Antes de realizar as tarefas de avaliação, os usuários possuíam tempo ilimitado para aprender e tornarem-se familiarizados com o aplicativo e os dispositivos. Para isso, os usuários realizavam uma fase de treinamento (Figura 5.3a) para praticar e somente durante essa fase ele podiam escolher livremente qual dispositivo utilizar. Assim que os usuários se sentiam confortáveis, as oito tarefas eram realizadas.

As tarefas eram selecionadas em uma ordem pré-definida. A descrição da tarefa era exibida na tela e, para iniciá-la, o usuário deveria pressionar o botão *Start*. As tarefas não eram finalizadas automaticamente, e sim quando o usuário sentisse que havia atingido o seu objetivo e pressionasse o botão *Finish*. Após, uma nova tarefa era selecionada pelo aplicativo e apresentada na tela, e assim sucessivamente até que as oito tarefas fossem concluídas.

Os usuários foram divididos em dois grupos com a mesma quantidade de pessoas. A metade deles testou primeiro as ferramentas 2D e depois as ferramentas 3D; a outra metade fez o inverso. Isso foi realizado para evitar o fator ordem nos resultados dos experimentos. Na utilização de cada conjunto de dados com a mesma técnica de interação, 2D ou 3D, as tarefas eram executadas na mesma ordem. Primeiramente a ferramenta Escavadeira era avaliada, sendo seguida pela Borracha e pela Guilhotina. A última tarefa a ser executada era a tarefa que avaliava o uso das três ferramentas ao mesmo tempo.

O arquivo de *log* gerado para cada usuário continha os seguintes dados associados com cada tarefa: descrição da tarefa, número total de voxels cinzas e vermelhos no volume avaliado, voxels brancos e vermelhos restantes, tempo gasto na execução da tarefa e histórico das ações realizadas. Essas ações correspondiam ao número de vezes que o usuário havia utilizado cada ferramenta, e as operações de *undo* e *reset*, assim como a ordem que elas foram executadas.

5.7 Resultados e Discussão

Para a análise estatística dos dados coletados, um grande número de variáveis foi considerado. Estas variáveis foram obtidas através de questionário pré e pós-testes e arquivos de *log*. Os resultados são discutidos nas próximas seções.

5.7.1 Comparação de Experiência com Desempenho

O questionário pré-teste respondido pelos usuários forneceu as experiências anteriores que eles possuíam com aplicativos e dispositivos 2D e 3D (Tabela 5.2). Esta tabela mostra que os usuários possuíam excelente experiência com mouse convencional 2D e apesar de muitos possuírem conhecimento em jogos 3D, poucos possuíam conhecimento em outros aplicativos 3D.

Tabela 5.2: Experiência anterior dos usuários com aplicativos e dispositivos 2D e 3D

	Excelente	Boa	Pouca	Nenhuma
Aplicativos 2D	25%	31.25%	37.5%	6.25%
Aplicativos 3D	12.5%	31.25%	50%	6.25%
Jogos 2D	68.75%	31.25%	0%	0%
Jogos 3D	68.75%	31.25%	0%	0%
Mouse 2D	100%	0%	0%	0%
Mouse 3D	0%	43.75%	25%	31.25%
Teclado	100%	0%	0%	0%
HMD	6.25%	43.75%	25%	31.25%
<i>Data Glove</i>	6.25%	31.25%	25%	37.5%
FOB	6.25%	18.75%	37.5%	37.5%

A correlação entre os dados do questionário pré-teste e os arquivos de *log* (variáveis independentes e dependentes) foi calculada. Não foi possível verificar nenhuma correlação entre a experiência que os usuários possuíam com os desempenhos atingidos. Outras variáveis, como idade e gênero, também não apresentaram nenhuma correlação.

5.7.2 Teste das Hipóteses

O teste ANOVA (Análise de Variância) foi utilizado para verificar se as diferenças obtidas em tempo e erros na realização das tarefas eram significativas. Para avaliar a primeira hipótese (*Os usuários são mais rápidos usando as ferramentas 2D*) o tempo gasto pelos usuários para completar cada tarefa foi analisado. O teste ANOVA mostrou que a diferença do tempo gasto em cada tarefa foi significativa ($F = 9,4330, p < 0,0049$) somente para as tarefas que utilizavam todas as ferramentas ao mesmo tempo para fazer o recorte do volume (T4 e T8). Neste caso a primeira hipótese foi confirmada, pois o tempo médio para realizar o recorte do volume com as ferramentas que utilizavam a técnica de interação 3D foi muito maior que o tempo gasto com as ferramentas 2D (Figura 5.4).

Para verificar a segunda hipótese (*A taxa de erros diminui quando o usuário usa as ferramentas 3D*), a taxa de erros foi calculada através da multiplicação dos erros dos usuários (normalizados) pelo tempo gasto em cada tarefa. O erro era calculado somando-se o número restante de voxels vermelhos e o número de voxels cinzas eliminados (lembrando que o objetivo da tarefa era eliminar somente os voxels vermelhos). A diferença de desempenho foi significativa de acordo com os resultados da ANOVA nas tarefas que os usuários avaliavam as ferramentas de recorte das Borrachas ($F = 42,2222, p < 6.90207E - 07$) e

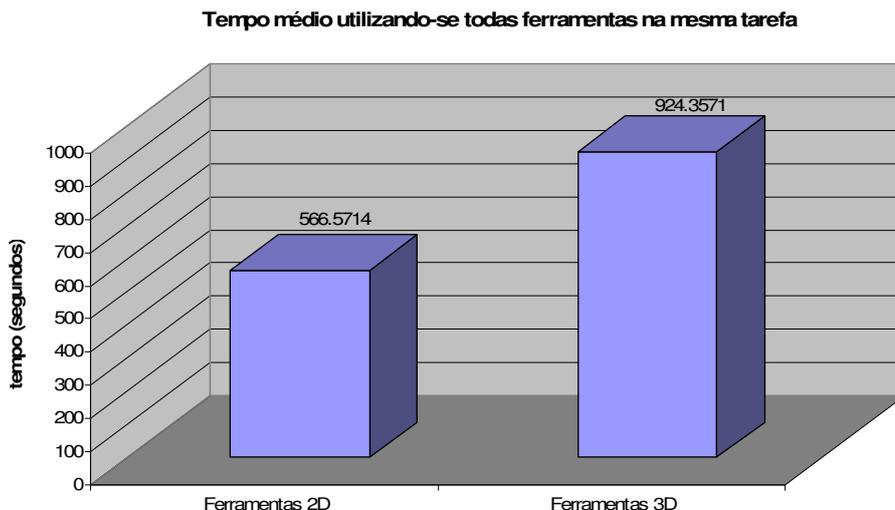


Figura 5.4: Tempo médio para a execução da tarefa onde o usuário podia usar todas as ferramentas de uma mesma técnica de interação ao mesmo tempo.

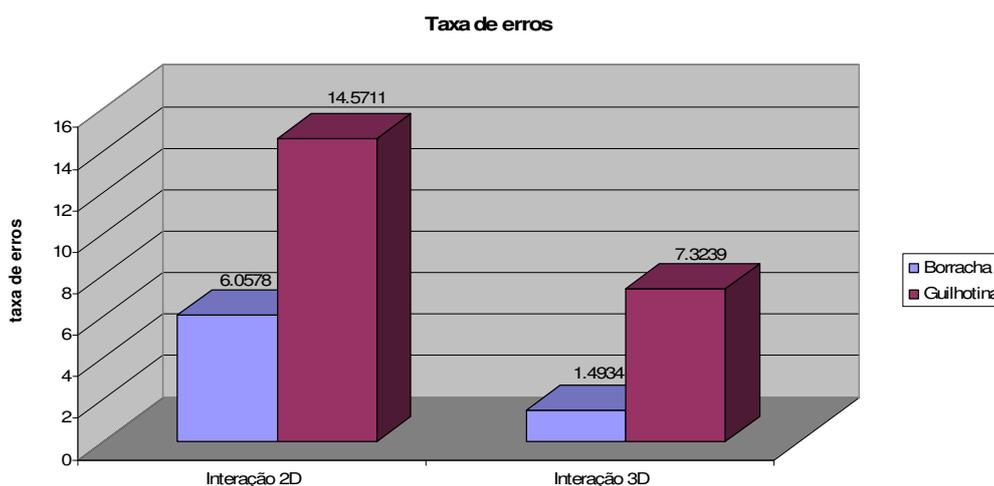


Figura 5.5: Taxa de erros nas tarefas onde somente as ferramentas de recorte volumétrico Borracha ou Guilhotina podiam ser utilizadas.

das Guilhotinas ($F = 4,2646$, $p < 0,04901$). Nestes casos a segunda hipótese foi confirmada, pois a taxa de erro utilizando as ferramentas 2D eram maiores que a taxa de erro das ferramentas 3D (Figura 5.5).

A análise dos questionários pós-teste confirmou a terceira hipótese (*Os usuários preferirão utilizar as ferramentas 3D para o recorte de volumes*). A Tabela 5.3 mostra que a grande maioria dos usuários preferiram a técnica de interação 3D para a realização do recorte volumétrico. O único caso em que essa hipótese não foi confirmada foi na avaliação das borrachas. Apesar dos usuários acharem mais fácil utilizar a Borracha 2D, a análise do desempenho (Figura 5.5) mostra que a performance dessa ferramenta foi muito pior. Isso se deve provavelmente ao fato da ilusão da Borracha 2D apagar o volume como se fosse um cilindro perpendicular ao plano de visão, enquanto que na metáfora 2D é um cone.

Tabela 5.3: Preferência dos usuários em relação às técnicas de interação empregadas nas ferramentas de recorte volumétrico.

	Interação 2D	Interação 3D
Em geral	37.5%	62.5%
Borracha	56.25%	43.75%
Escavadeira	18.75%	81.25%
Guilhotina	31.25%	68.75%

5.8 Síntese do Capítulo

Neste capítulo foi feita a avaliação experimental de técnicas de interação 2D e 3D utilizadas para implementar as ferramentas de recorte volumétrico descritas no Capítulo 4. Através de um experimento com usuários, foi possível observar o desempenho e as preferências dessas técnicas de interação.

Apesar da diferença do tempo gasto para a utilização das ferramentas separadamente não ter sido significativa, no caso mais próximo da utilização real do sistema, onde os usuários podem escolher e trocar as ferramentas para recortar um volume, a análise dos resultados foi expressiva. O tempo médio de utilização das ferramentas com técnicas de interação 3D foi muito maior que o tempo médio com as técnicas 2D. Isto provavelmente se deve ao fato dos usuários estarem acostumados com esse tipo de interação em aplicativos do seu dia-a-dia.

Embora o tempo médio de utilização das ferramentas 3D ter sido maior que o das ferramentas 2D, a performance das ferramentas 3D foi melhor que a das ferramentas 2D nos resultados significantes. A performance leva em consideração, além do tempo gasto na utilização das ferramentas, o erro gerado por elas. Isso significa que, apesar dos sujeitos terem utilizado as ferramentas 2D mais rápido, eles também erraram muito mais. Normalmente essa métrica é mais importante para a avaliação que somente a análise do tempo separadamente. Assim como nesse trabalho, outros autores também obtiveram melhores desempenhos com técnicas de interação 3D (HINCKLEY et al., 1997).

A grande maioria dos usuários também preferiu as técnicas de interação 3D para o recorte de volumes. Ware e Jessome (WARE; JESSOME, 1988) obtiveram o mesmo resultado, e eles apontam a necessidade dos usuários ter de quebrar mentalmente as tarefas 3D em sub-tarefas 2D como principal causa disso.

6 AVALIAÇÃO EXPERIMENTAL DOS DISPOSITIVOS DE ENTRADA

Os dispositivos de entrada e as técnicas de interação utilizadas afetam diretamente o desempenho dos usuários durante a execução de uma determinada tarefa (MACKENZIE, 1995). Para avaliar o desempenho e as preferências dos usuários durante a interação com conjuntos de dados volumétricos utilizando as ferramentas de recorte desenvolvidas, um experimento de manipulação e recorte de volumes foi projetado utilizando-se a combinação de diferentes dispositivos de entrada. Esta seção descreve a avaliação experimental dos diferentes dispositivos de entrada para realizar o recorte volumétrico e apresenta os resultados obtidos.

6.1 Utilização das Ferramentas

Diferentes dispositivos de entrada têm sido utilizados em aplicativos de recorte volumétrico, tais como mouse convencional 2D (WANG; KAUFMAN, 1995), mouse 3D (GALYEAN; HUGHES, 1991), e até mesmo ambos os mouses 2D e 3D ao mesmo tempo (PARVIAINEN; SAINIO; RAISAMO, 2004). A manipulação bimanual de dispositivos de entrada é freqüentemente utilizada para melhorar a eficiência da interação entre o usuário e o computador através da possibilidade do usuário realizar duas sub-tarefas em paralelo (BUXTON; MEYERS, 1986). Esse tipo de manipulação também transfere para o mundo virtual a habilidade bimanual natural dos usuários em utilizar ferramentas no seu dia-a-dia com pouco ou nenhum treino (KABBASH; BUXTON; SELLEN, 1994; LEGANCHUK; ZHAI; BUXTON, 1998). A interação com duas mãos pode também ajudar o usuário a perceber melhor um ambiente virtual, como em relação à profundidade e distâncias, pois as pessoas também possuem habilidade natural de saber onde uma mão está em relação à outra (SACHS; ROBERTS; STOOPS, 1991; HINCKLEY et al., 1994).

O principal objetivo das técnicas de interação adotadas no aplicativo para a avaliação experimental dos dispositivos de entrada foi aumentar a continuidade entre a visualização e a interação. Para isso, foram utilizadas as ferramentas descritas na Seção 4.4, ou seja, técnicas de interação 3D utilizando a metáfora da mão-virtual. Este tipo de interação não força o usuário a decompor tarefas 3D em uma série de tarefas 2D ou 1D (LIANG; GREEN, 1993).

Os dispositivos adotados nesta avaliação, conforme mencionado no Capítulo 3, foram o Magellan Spacemouse (Figura 6.1a), porque ele tem seis graus de liberdade (*degrees of freedom*, DOFs) e é relativamente simples e de fácil aprendizagem, e um mouse convencional (Figura 6.1b), pois é o dispositivo de entrada mais amplamente utilizado.

No aplicativo desenvolvido, dois tipos de objetos podiam ser manipulados pelos dis-

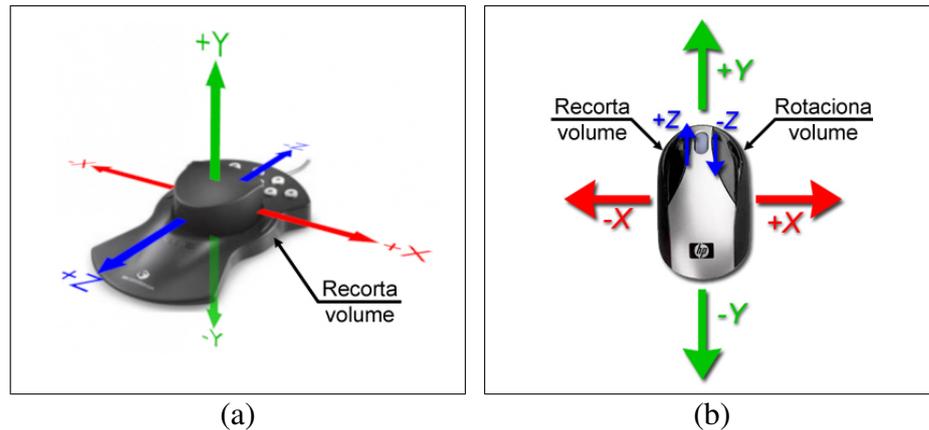


Figura 6.1: Dispositivos utilizados na avaliação experimental: (a) spacemouse e (b) mouse convencional.

positivos de entrada: as ferramentas e o volume de interesse (*Volume of Interest, VOI*). As ferramentas podem ser rotacionadas ao redor de si próprias e transladadas nos eixos x , y e z . O VOI também pode ser rotacionado ao redor dele mesmo, mas não pode ser transladado.

O mouse 3D pode facilmente controlar rotações e translações 3D na técnica da mão virtual, pois ele tem 6 DOFs, mas essas transformações não são trivialmente executadas pelo mouse convencional 2D. Com o objetivo de comparar ambos os dispositivos, foi necessário compensar a falta de DOFs do mouse 2D. Translações ao longo do eixo x e y foram normalmente realizadas através do movimento do mouse 2D, mas translações na dimensão z foram realizadas através do botão de rolagem do mouse (*wheel button*), como ilustrado na Figura 6.1b. As rotações foram compensadas através do movimento do mouse com um botão pressionado (movimento de “arrastar”).

O usuário pode selecionar livremente na interface do aplicativo qual objeto (ferramentas ou VOI) ele quer manipular com cada mouse (2D ou 3D). A manipulação de ambos os objetos com o mesmo dispositivo ao mesmo tempo não é permitida, como, por exemplo, usar o mouse 2D rotacionar o volume e apagá-lo com a borracha ao mesmo tempo. Neste caso, o usuário poderia realizar primeiro a rotação e em seguida apagar o volume, ou vice-versa, alterando os objetos controlados pelos dispositivos através da interface. Para manipular dois objetos ao mesmo tempo, a interação bimanual foi habilitada através da utilização do mouse 2D e do mouse 3D ao mesmo tempo (Figura 6.2). Neste caso, o usuário pode escolher o dispositivo a ser associado a cada objeto. Devido a dificuldades de implementação com as APIs escolhidas, dois mouses 3D ou dois mouses 2D não podem ser utilizados ao mesmo tempo.

O aplicativo utilizado para fazer a avaliação experimental dos dispositivos de entrada para o recorte de dados volumétricos foi desenvolvido em OpenGL com a biblioteca GLUT para a criação da interface gráfica. Neste aplicativo, foram oferecidos controles como botões, *radio buttons* e um *spinner* para auxiliar o usuário na realização das tarefas e escolha de dispositivos e ferramentas, mas as ações relacionadas diretamente à manipulação do volume e aplicação das ferramentas para o recorte volumétrico foram implementadas utilizando-se apenas os dispositivos de entrada. Outras estruturas como informações sobre as ferramentas e volumes, *bounding box* e janelas para a visualização de fatias do volume também foram disponibilizadas. A Figura 6.3 ilustra a interface da aplicação de avaliação.

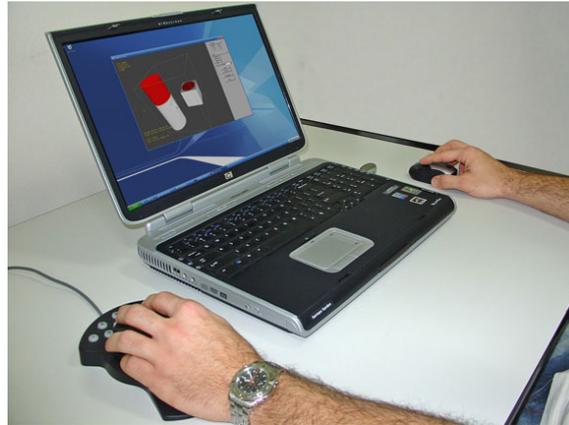


Figura 6.2: Interação bimanual com mouse 3D e 2D.

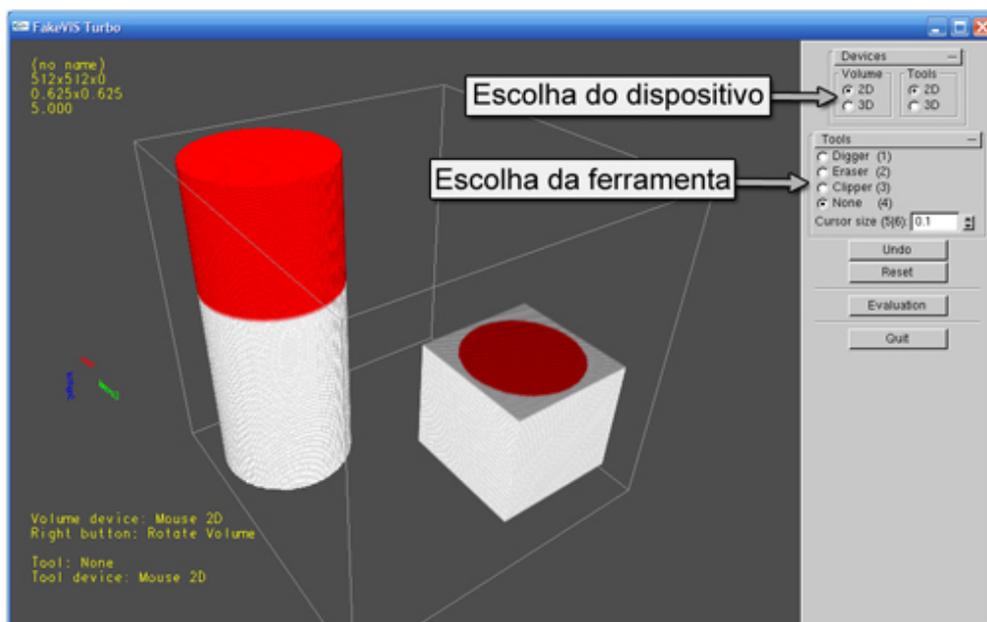


Figura 6.3: Interface do aplicativo de avaliação de dispositivos para o recorte volumétrico.

6.2 Hipóteses

A avaliação experimental dos dispositivos de entrada foi baseada em três hipóteses, levando em consideração as diferentes combinações dos dispositivos utilizados:

- **H1.** A utilização das ferramentas é mais rápida usando o mouse convencional.
- **H2.** A taxa de erros diminui quando o usuário usa o dispositivo 3D associado às ferramentas de recorte.
- **H3.** Os sujeitos preferirão utilizar ambos os dispositivos simultaneamente para manipular o volume e as ferramentas.

A primeira hipótese é baseada na idéia de que usuários comuns estão acostumados ao mouse convencional e durante o período de treinamento da aplicação de avaliação com o mouse 3D os usuários não adquirem as mesmas habilidades que já possuem com

o mouse convencional. A segunda hipótese, por outro lado, é baseada na afirmação que dispositivos 3D são mais indicados para técnicas de interação 3D, pois os movimentos são diretamente mapeados para os objetos 3D (LIANG; GREEN, 1993). Finalmente, a última hipótese conta com a vantagem cognitiva de se utilizar ambas as mãos para manipular o volume e as ferramentas ao mesmo tempo, reduzindo o esforço mental de visualizar e realizar as tarefas de forma pouco natural como normalmente as técnicas unimanuais tradicionais impõem (BUXTON; MEYERS, 1986).

6.3 Cenário

O experimento consistia na eliminação de voxels vermelhos em um volume vermelho e branco. O usuário deveria remover o número máximo de voxels vermelhos do volume e evitar a eliminação de voxels brancos ao mesmo tempo.

A aplicação de avaliação tinha como dados de entrada diferentes conjuntos de imagens com pixels vermelhos e brancos que descreviam o volume. Foram criados três conjuntos de dados diferentes, um para cada ferramenta, devido à geometria associada a cada uma. Um quarto volume foi utilizado para treinar os usuários antes de começarem o processo de avaliação (Figura 6.4).

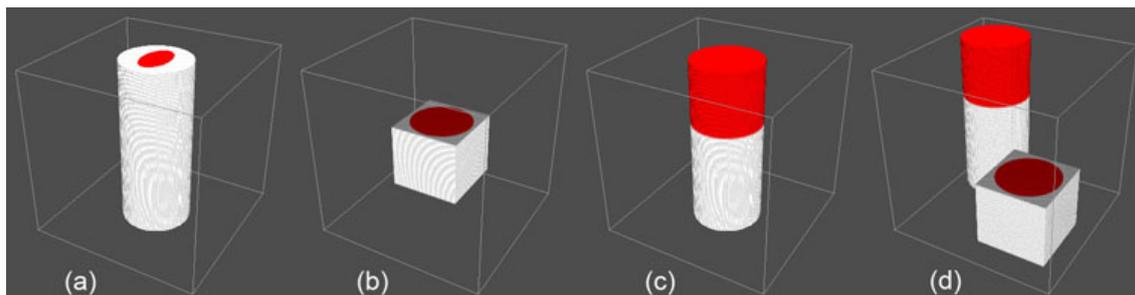


Figura 6.4: Conjunto de dados do experimento: (a) volume de avaliação da Borracha 3D, (b) volume de avaliação da Escavadeira 3D, (c) volume de avaliação da Guilhotina 3D e (d) volume para treinamento.

Para testar as três hipóteses, foram definidas doze tarefas básicas utilizando diferentes combinações de dispositivos de entrada. A combinação de três ferramentas (Borracha 3D, Escavadeira 3D e Guilhotina 3D), dois objetos a serem manipulados (ferramentas e volume) e dois dispositivos de entrada (mouse convencional e mouse 3D) resulta no conjunto de doze tarefas distintas (Tabela 6.1).

6.4 Variáveis Dependentes e Independentes

As variáveis independentes utilizadas foram: idade; gênero; atividade profissional; experiência com dispositivos 2D e 3D; experiência em aplicações 2D e 3D (como jogos, CAD, 3D Studio, Blender, etc.), e finalmente as doze tarefas. Essas variáveis, excluindo as doze tarefas, foram coletadas através de um questionário pré-teste, disponível no Apêndice C. Já as variáveis dependentes coletadas durante a avaliação experimental dos dispositivos de entrada foram: tempo de realização das tarefas; taxa de erro, dispositivo preferido e facilidade de uso dos dispositivos, estas duas últimas coletadas após os testes através de questionários respondidos pelos usuários avaliados (Apêndice D).

Tabela 6.1: Número total de tarefas decorrentes da combinação de 3 ferramentas de recorte (Borracha 3D, Escavadeira 3D, e Guilhotina 3D) e 2 dispositivos (mouse 2D e mouse 3D) para a manipulação de 2 objetos (volume e ferramentas).

Tarefas	Ferramentas	Controle do volume	Controle das ferramentas
T1	Borracha 3D	Mouse 2D	Mouse 2D
T2	Borracha 3D	Mouse 2D	Mouse 3D
T3	Borracha 3D	Mouse 3D	Mouse 2D
T4	Borracha 3D	Mouse 3D	Mouse 3D
T5	Escavadeira 3D	Mouse 2D	Mouse 2D
T6	Escavadeira 3D	Mouse 2D	Mouse 3D
T7	Escavadeira 3D	Mouse 3D	Mouse 2D
T8	Escavadeira 3D	Mouse 3D	Mouse 3D
T9	Guilhotina 3D	Mouse 2D	Mouse 2D
T10	Guilhotina 3D	Mouse 2D	Mouse 3D
T11	Guilhotina 3D	Mouse 3D	Mouse 2D
T12	Guilhotina 3D	Mouse 3D	Mouse 3D

O tempo de realização foi medido desde o começo de cada tarefa até a sua finalização pelo usuário. A taxa de erro levou em consideração o erro após a tarefa de recorte ser finalizada e o tempo gasto para realizar a tarefa, como demonstrado pela Equação 6.1:

$$taxa_de_erro = \epsilon \times T \quad (6.1)$$

onde ϵ é o número de erros após a execução da tarefa e T é o tempo gasto durante a tarefa. ϵ é calculado conforme a Equação 6.2:

$$\epsilon = \frac{V_{\epsilon} + B_{\epsilon}}{\#voxels} \quad (6.2)$$

com V_{ϵ} sendo o número restante de voxels vermelhos e B_{ϵ} o número restante de voxels brancos. Finalmente, a facilidade de uso e o dispositivo preferido foram verificados através da análise de questionários (Apêndice D).

6.5 Sujeitos

Os testes foram realizados com quinze sujeitos (um professor e quatorze estudantes), a maioria da área de Ciência da Computação, sendo três mulheres e doze homens com idades entre 20 e 40 anos, com média de 25 anos de idade. Embora somente dois estudantes não trabalhassem com computação gráfica, a maioria deles tinha pouca ou nenhuma experiência com mouse 3D. Cada um deles testou as três ferramentas (Seção 4.4) e os dois dispositivos (Figura 6.1), através da realização das doze tarefas.

6.6 Metodologia e Procedimentos

Os testes foram realizados em um notebook HP Pavilion zd8000 com processador Intel Pentium 4 de 3.20GHz, 512MB de RAM, e placa gráfica ATI Mobility Radeon X600 PCI Express x16 com 256 MB.

Os seguintes passos foram aplicados na mesma ordem para todos os usuários:

- Instruções sobre o aplicativo e os dispositivos foram fornecidas;
- Os usuários preencheram o questionário de pré-teste presente no Apêndice D;
- Os usuários podiam treinar antes de iniciar a avaliação sem limite de tempo;
- As doze tarefas eram realizadas randomicamente;
- O arquivo de *log* era salvo;
- O questionário pós-teste, presente no Apêndice E, era preenchido pelos usuários.

Antes de realizar as tarefas de avaliação, os usuários tiveram tempo ilimitado para aprender e tornarem-se familiarizados com o aplicativo e os dispositivos. Para isso, os usuários realizavam uma fase de treinamento (Figura 6.4) para praticar e somente durante essa fase ele podiam escolher livremente qual dispositivo utilizar. Assim que os usuários se sentiam confortáveis, as doze tarefas eram realizadas.

As tarefas eram selecionadas em ordem randômica. A descrição da tarefa era exibida na tela, e para iniciá-la o usuário deveria pressionar o botão *Start*. As tarefas não eram finalizadas automaticamente, e sim quando o usuário sentisse que havia atingido o seu objetivo e pressionasse o botão *Finish*. Após, uma nova tarefa era sorteada pela aplicação e apresentada na tela, e assim sucessivamente até que as doze tarefas fossem concluídas.

O arquivo de *log* gerado para cada usuário contém os seguintes dados associados com cada tarefa: descrição da tarefa, número total de voxels brancos e vermelhos, número de voxels brancos e vermelhos restantes, e tempo gasto durante a execução da tarefa.

Os usuários demoraram em média 17 minutos e 55 segundos para realizar todas as tarefas (sem contar o tempo gasto durante a fase de treinamento). O usuário mais rápido levou 9 minutos e 2 segundos, e o mais demorado levou 32 minutos e 20 segundos.

6.7 Resultados

Para a análise estatística dos dados coletados, várias variáveis foram consideradas. Estas variáveis foram obtidas através dos questionários pré e pós-teste, e através dos arquivos de *log*. Os resultados são discutidos a seguir.

6.7.1 Comparação de Experiência com Desempenho

O questionário pré-teste respondido pelos usuários forneceu as experiências anteriores que eles possuíam com aplicações e dispositivos 2D e 3D (Tabela 6.2). Esta tabela mostra que os usuários possuíam excelente experiência com mouse convencional 2D e teclado, mas somente poucos tinham experiência com mouse 3D.

A correlação entre os dados do questionário pré-teste e os arquivos de *log* (variáveis independentes e dependentes) foi calculada. Foi possível observar que o desempenho nas tarefas com a Borracha 3D e a Escavadeira 3D correlacionava-se com a experiência que os usuários possuíam com dispositivos 3D, especialmente quando os usuários tinham que manipular as ferramentas com o mouse 2D e o volume com o mouse 3D ($r > 0.514$, $p < 0.05$).

Não foi constatada correlação entre desempenho e outras variáveis, como idade e gênero.

Tabela 6.2: Experiência anterior dos usuários com aplicativos e dispositivos 2D e 3D

	Excelente	Boa	Pouca	Nenhuma
Aplicativos 3D	40 %	20%	26.66%	13.33%
Jogos 2D	73.33%	20%	6.66%	0%
Jogos 3D	60%	20%	20%	0%
Mouse 2D	100%	0%	0%	0%
Mouse 3D	0%	6.66%	66.66%	26.66%
Teclado	100%	0%	0%	0%
HMD	6.66%	53.33%	20%	20%
<i>Data Glove</i>	13.33%	40%	26.66%	20%
FOB	13.33%	26.66%	20%	40%

6.7.2 Teste das Hipóteses

O teste ANOVA (Análise de Variância) foi utilizado para verificar se as informações de tempo e erro obtidas nos arquivos de *log* eram significantes. Para avaliar a primeira hipótese (*A utilização das ferramentas é mais rápida usando-se o mouse convencional*) o tempo gasto pelos usuários para completar cada tarefa foi analisado. Os resultados do ANOVA mostraram que a diferença do tempo gasto em cada tarefa foi significativa somente para a Borracha 3D ($F = 4.663$, $p < 0.04$) e para a Escavadeira 3D ($F = 5.308$, $p < 0.02$). Quando o mouse 2D controlava a manipulação do volume, o tempo médio utilizando-se o mouse 3D para manipular as ferramentas era maior do que o tempo gasto utilizando-se o mouse 2D, tanto para a Borracha 3D quanto para a Escavadeira 3D (Figura 6.5).

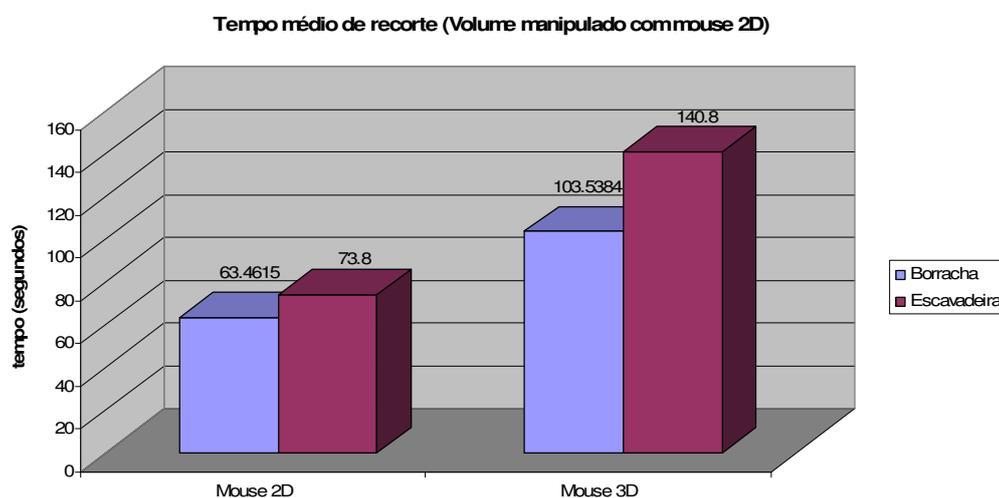


Figura 6.5: Tempo médio das ferramentas (em segundos) quando o volume é controlado pelo mouse 2D

Para verificar a segunda hipótese (*A taxa de erros diminui quando o usuário usa o dispositivo 3D associado às ferramentas*), a taxa de erros foi calculada através da multiplicação dos erros dos usuários pelo tempo gasto em cada tarefa. O total de erros era calculado somando-se o número restante de voxels vermelhos e o número de voxels brancos eliminados (lembrando que o objetivo da tarefa era eliminar somente os voxels vermelhos). A diferença de desempenho não foi significativa em nenhum caso de acordo com os

resultados da ANOVA, ou seja, a segunda hipótese não pôde ser confirmada.

A análise dos questionários pós-teste confirmou a terceira hipótese (*Os sujeitos preferirão utilizar ambos dispositivos simultaneamente para manipular o volume e as ferramentas*). A Tabela 6.3 mostra que a grande maioria dos usuários preferiram o mouse 2D para manipular as ferramentas e o mouse 3D para controlar o volume separadamente. A Tabela 6.4 mostra que os usuários também preferiram esta combinação de dispositivos para a manipulação com as duas mãos.

Tabela 6.3: Dispositivos preferidos para controlar o volume e as ferramentas

	Mouse 2D	Mouse 3D
Volume	26.67%	73.33%
Borracha 3D	73.33%	26.67%
Escavadeira 3D	66.67%	33.33%
Guilhotina 3D	53.33%	46.67%

Tabela 6.4: Preferência de combinação de dispositivos

	Preferência
Volume (Mouse 2D) e Ferramentas (Mouse 2D)	0%
Volume (Mouse 2D) e Ferramentas (Mouse 3D)	26.66%
Volume (Mouse 3D) e Ferramentas (Mouse 2D)	66.66%
Volume (Mouse 3D) e Ferramentas (Mouse 3D)	6.66%

6.8 Síntese do Capítulo

Neste capítulo foi descrita a avaliação de dois dispositivos de entrada, mouse 2D e 3D, para a manipulação das ferramentas de recorte volumétrico e para o controle da posição e orientação do volume de dados. Para atingir esse objetivo, um experimento com usuários foi desenvolvido, e dados sobre os seus desempenhos e preferências foram coletados.

A análise dos resultados mostrou que somente em dois casos, quando o volume era manipulado pelo mouse 2D, os resultados do tempo médio das ferramentas Borracha 3D e Escavadeira 3D foram significativos. Nesses casos, o tempo médio gasto utilizando-se o mouse 2D para controlar as ferramentas foi menor que utilizando-se o mouse 3D. Esses resultados confirmam outros apresentados na literatura, comparando o mouse convencional com dispositivos de interação 3D. Nedel *et al.* (NEDEL et al., 2003) compararam o uso de mouse convencional com uma *data glove* em tarefas de seleção e manipulação de objetos num ambiente virtual, encontrando diferença significativa favorecendo o mouse convencional. Entretanto, conforme mencionada em trabalho anterior (HUFF et al., 2006), o sucesso obtido pelo uso do mouse 2D deve-se provavelmente a familiaridade dos usuários com este dispositivo, em detrimento de outros menos convencionais. A realização de experimentos com usuários que nunca tenham utilizado mouse 2D, ou ainda, que possuam uma grande experiência com o mouse 3D pode reverter os resultados obtidos neste trabalho. Jacob e Silbert (JACOB; SIBERT, 1992) também constataram que a utilização de dispositivos 3D não é sempre melhor para realizar tarefas 3D.

A análise dos resultados também mostrou que, apesar dos poucos casos significativos onde o tempo médio utilizando-se dois dispositivos simultaneamente para realizar as tarefas foi maior, a grande maioria dos usuários preferiu a utilização de dois dispositivos

ao mesmo tempo. Isso pode ser devido a varias razões já apresentadas em trabalhos anteriores, como, por exemplo, a possibilidade de realizar tarefas em paralelo (BUXTON; MEYERS, 1986), a transferência das formas de manipulação da vida real para o computador (KABBASH; BUXTON; SELLEN, 1994), e a melhoria na percepção do ambiente com a utilização de duas mãos (HINCKLEY et al., 1994).

Foi possível observar que em, geral, os usuários preferem manipular o volume de dados com o mouse 3D e as ferramentas com o mouse 2D. Outro fato interessante, é que nenhum usuário preferiu usar apenas o mouse 2D para realizar o recorte volumétrico. Estes resultados foram publicados recentemente em ((HUFF et al., 2006)).

7 CONCLUSÕES E TRABALHOS FUTUROS

Três ferramentas interativas de recorte volumétrico foram apresentadas. Essas ferramentas são baseadas em facilidades fornecidas pelas GPUs na utilização de texturas tridimensionais. As ferramentas Borracha, Escavadeira e Guilhotina são implementadas em programas especiais de fragmentos que descartam fragmentos baseados em informações de visibilidade armazenadas numa máscara.

Cada uma das três ferramentas foi desenvolvida em duas técnicas de interação distintas: apontador virtual e mão virtual. Isso possibilitou a avaliação de técnicas de interação 2D e 3D para o recorte de conjuntos de dados volumétricos.

Um experimento inicial com usuários foi realizado para avaliar as técnicas de interação utilizadas nas ferramentas de recorte volumétrico. Técnicas de interação 2D foram testadas contra técnicas possivelmente mais intuitivas e fáceis de serem usadas utilizando metáfora 3D. A análise dos resultados comprovou a preferência dos usuários pelas técnicas de interação 3D. Em poucos casos, onde a diferença de performance dos usuários durante a avaliação dessas técnicas de interação foi significativa, comprovou-se que o desempenho com interação 3D é superior ao desempenho com interação 2D.

Após, um segundo experimento com usuários possibilitou a utilização de diferentes dispositivos para manipular as ferramentas de recorte e controlar a orientação do volume de dados. A análise dos resultados mostrou que na grande maioria dos casos a diferença de tempo e desempenho dos dois dispositivos avaliados (mouse 2D e mouse 3D) não é significativa. Apesar disso, foi possível observar que a maioria dos usuários preferem utilizar o mouse 3D para controlar o volume e o mouse 2D para manipular as ferramentas. Também foi possível verificar que a preferência da maioria dos usuários, obtida através de questionários, é utilizar as duas mãos ao mesmo tempo para controlar os dois dispositivos.

Analisando os resultados dos dois experimentos realizados, é possível afirmar que apesar dos usuários estarem acostumados a dispositivos e técnicas de interação 2D, em ambientes para o recorte de conjunto de dados volumétricos, as abordagens 3D geram melhores resultados. Nessas primeiras avaliações, com um número relativamente pequeno de usuários, os quais possuíam pouca experiência em técnicas e dispositivos de interação 3D, alguns dos resultados foram significantes. Acredita-se, portanto, que com um número maior de usuários, ou à medida que os usuários adquiram maior experiência com técnicas e dispositivos 3D, a preferência e a melhora de desempenho com interação 3D será mais evidente.

O formato das ferramentas com técnicas de interação 2D são círculos (Borracha 2D e Escavadeira 2D) e uma linha (Guilhotina 2D), e o formato das ferramentas com técnicas de interação 3D são um cilindro (Borracha 3D), uma esfera (Escavadeira 3D) e um quadrado (Guilhotina 3D). Esses formatos foram escolhidos por serem geometrias básicas que podem simular outras geometrias mais complexas através de algumas interações. Os

formatos das ferramentas também foram inspirados na necessidade existente em aplicativos de recorte de volumes médicos. A versão das ferramentas baseadas em metáforas de interação 2D são semelhantes a algumas ferramentas utilizadas em editores de imagens convencionais. Portanto, seria também interessante investigar a aplicabilidade de outras ferramentas presentes nesses editores para a realização de recorte volumétrico, como, por exemplo, cursores quadrados, cursores com bordas suavizadas, e seleção de uma região com valores semelhantes. Após, seguindo a mesma abordagem desse trabalho, essas ferramentas poderiam ser estendidas com técnicas de interação 3D.

Com relação à avaliação dos dispositivos de entrada para o recorte volumétrico, devido a problemas e dificuldades de implementação encontrados, não foi possível comparar a utilização de dois mouses 3D ao mesmo tempo e, conseqüentemente, também não foi comparado o uso de dois mouses 2D. Mesmo com a adição desses dois novos casos, no processo de avaliação dos dispositivos, os testes com os usuários não seriam mais longos, pois o número total de tarefas que eles executariam continuaria sendo 12 (resultado da combinação de dois dispositivos, duas mãos e três ferramentas de recorte). A futura implementação desses novos casos pode gerar resultados interessantes, pois, apesar do presente trabalho não ter encontrado nenhuma correlação do fato dos usuários serem destros ou canhotos com os resultados de tempo, desempenho e preferência obtidos, a comparação desses valores com relação a cada mão poderá ser realizada e talvez apresentar alguma informação relevante.

Atualmente está sendo desenvolvido um novo aplicativo que utilizará as ferramentas e técnicas de interação presentes nesse trabalho para analisar um conjunto de dados obtidos através de imagens médicas. Esses dados são utilizados no planejamento de cirurgia hepática, o que tem sido a motivação do desenvolvimento de vários trabalhos no grupo de pesquisa. O objetivo principal desse novo aplicativo será separar o fígado dos dados volumétricos de um tórax para analisar a sua estrutura e calcular o seu volume, com vistas a planejamento de cirurgias de transplante de fígado. Acredita-se, que numa futura avaliação desse aplicativo médico com relação às técnicas de interação e dispositivos de entrada utilizando médicos como usuários, a preferência por técnicas e dispositivos 3D será ainda mais acentuada, pois os sujeitos possuem maior experiência com esse tipo de interação, como, por exemplo, através da utilização de instrumentos de laparoscopia. Pelo mesmo motivo, provavelmente os sujeitos preferirão o uso das duas mãos ao mesmo tempo para interagir com o sistema.

Outra possibilidade de trabalho futuro no que se refere à edição de volumes de dados é a modificação dos valores originais dos voxels do volume, para acrescentar material ou modificar a natureza de regiões do volume. Esta facilidade tem aplicação direta em reconstrução óssea para a construção de modelos médicos que podem ser utilizados na fabricação de próteses a serem implantadas em cirurgias de reconstrução pós-trauma (SOUZA; CENTENO; PEDRINI, 2003).

REFERÊNCIAS

AKELEY, K. Reality Engine Graphics. In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, SIGGRAPH, 20., 1993, Anaheim, CA, USA. **Proceedings...** New York: ACM Press, 1993. p.109–116.

BIER, E. A. Snap-dragging in three dimensions. In: SYMPOSIUM ON INTERACTIVE 3D GRAPHICS, SI3D, 1990, Snowbird, UT, USA. **Proceedings...** New York: ACM Press, 1990. p.193–204.

BLESER, T. W.; SIBERT, J. Toto: a tool for selecting interaction techniques. In: ANNUAL ACM SIGGRAPH SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, UIST, 3., 1990, Snowbird, UT, USA. **Proceedings...** New York: ACM Press, 1990. p.135–142.

BOWMAN, D. A.; HODGES, L. F. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. In: SYMPOSIUM ON INTERACTIVE 3D GRAPHICS, SI3D, 1997, Providence, RI, USA. **Proceedings...** New York: ACM Press, 1997. p.35–38.

BOWMAN, D. A.; HODGES, L. F. Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. **Journal of Visual Languages and Computing**, [S.l.], v.10, n.1, p.37–53, 1999.

BOWMAN, D. A.; JOHNSON, D.; HODGES, L. F. Testbed Evaluation of Virtual Environment Interaction Techniques. **Presence: Teleoperators and Virtual Environments**, [S.l.], v.10, n.1, p.75–95, 2001.

BOWMAN, D. A.; KRUIJFF, E.; JR., J. J. L.; POUPYREV, I. An Introduction to 3D User Interface Design. **Presence: Teleoperators and Virtual Environments**, [S.l.], v.10, n.1, p.96–108, 2001.

BRODLIE, K.; WOOD, J. Recent Advances in Volume Visualization. **Computer Graphics Forum**, Malden, MA, USA, v.20, n.2, p.125–148, 2001.

BRUCKNER, S.; GRÖLLER, M. E. VolumeShop: an interactive system for direct volume illustration. In: IEEE CONFERENCE ON VISUALIZATION, VIS, 2005. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2005. p.671–678.

BULLITT, E.; AYLWARD, S. Volume Rendering of Segmented Tubular Objects. In: INTERNATIONAL CONFERENCE ON MEDICAL IMAGE COMPUTING AND COMPUTER-ASSISTED INTERVENTION, MICCAI, 4., 2001, Utrecht, Netherlands. **Proceedings...** London: Springer-Verlag, 2001. p.161–168.

BUXTON, W. Lexical and Pragmatic Considerations of Input Structures. **ACM SIGGRAPH Computer Graphics**, New York, NY, USA, v.17, n.1, p.31–37, 1983.

BUXTON, W.; MEYERS, B. A. A Study in Two-Handed Input. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1986, Boston, MA, USA. **Proceedings...** New York: ACM Press, 1986. p.321 – 326.

CABRAL, B.; CAM, N.; FORAN, J. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In: SYMPOSIUM ON VOLUME VISUALIZATION, VVS, 1994, Tysons Corner, VA, USA. **Proceedings...** New York: ACM Press, 1994. p.91–98.

CHEN, M.; MOUNTFORD, S. J.; SELLEN, A. A study in interactive 3-D rotation using 2-D control devices. In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, SIGGRAPH, 15., 1988, Atlanta, GA, USA. **Proceedings...** New York: ACM Press, 1988. p.121 – 129.

CULLIP, T. J.; NEUMANN, U. **Accelerating Volume Reconstruction With 3D Texture Hardware**. Chapel Hill, NC, USA: [s.n.], 1994.

DIETRICH, C. A.; NEDEL, L. P.; OLABARRIAGA, S. D.; COMBA, J. L. D.; ZANCHET, D. J.; SILVA, A. M. M. da; SOUZA MONTERO, E. F. de. Real-time interactive visualization and manipulation of the volumetric data using GPU-based methods. In: SPIE MEDICAL IMAGING: VISUALIZATION, IMAGE-GUIDED PROCEDURES, AND DISPLAY, 2004, San Diego, CA, USA. **Proceedings...** [S.l.: s.n.], 2004. p.181–192.

DUNN, R. M.; HERZOG, B. Status report of the graphic standards planning committee. **ACM SIGGRAPH Computer Graphics**, New York, NY, USA, v.11, n.3, p.31–47, 1977.

ENDERLE, G.; KANSY, K.; PFAFF, G. **Computer Graphics Programming: GKS – the graphics standard**. New York: Springer-Verlag, 1986. xvi + 542p.

ENGEL, K.; KRAUS, M.; ERTL, T. High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading. In: ACM SIGGRAPH/EUROGRAPHICS WORKSHOP ON GRAPHICS HARDWARE, HWWS, 2001, Los Angeles, CA, USA. **Proceedings...** New York: ACM Press, 2001. p.9–16.

FOLEY, J. D.; WALLACE, V. L. The Art of Natural Graphic Man-Machine Conversation. **Proceedings of the IEEE**, Piscataway, NJ, USA, v.62, n.4, p.462–471, 1974.

FOLEY, J. D.; WALLACE, V. L.; CHAN, P. The Human Factors of Computer Graphics Interaction Techniques. **IEEE Computer Graphics and Applications**, Los Alamitos, CA, USA, v.4, n.11, p.13–48, 1984.

FORSBERG, A.; HERNDON, K.; ZELEZNIK, R. Aperture Based Selection for Immersive Virtual Environments. In: ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, UIST, 9., 1996, Seattle, WA, USA. **Proceedings...** New York: ACM Press, 1996. p.95–96.

GALYEAN, T. A.; HUGHES, J. F. Sculpting: an interactive volumetric modeling technique. In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, SIGGRAPH, 18., 1991, Las Vegas, NV, USA. **Proceedings...** New York: ACM Press, 1991. p.267–274.

GARNER, W. R.; FELEOLDY, G. L. Integrality of Stimulus Dimensions in Various Types of Information Processing. **Cognitive Psychology**, [S.l.], v.1, n.3, p.225–241, 1970.

GELDER, A. V.; KIM, K. Direct Volume Rendering with Shading via Three-Dimensional Textures. In: SYMPOSIUM ON VOLUME VISUALIZATION, VVS, 1996, San Francisco, CA, USA. **Proceedings...** New York: ACM Press, 1996. p.23–30.

HERNDON, K.; DAM, A. van; GLEICHER, M. The Challenges of 3D Interaction: A CHI'94 workshop. **SIGCHI Bulletin**, New York, NY, USA, v.26, n.4, p.36–43, 1994.

HINCKLEY, K.; PAUSCH, R.; GOBLE, J. C.; KASSELL, N. F. A Survey of Design Issues in Spatial Input. In: ANNUAL ACM SIGGRAPH SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, UIST, 7., 1994, Marina del Rey, CA, USA. **Proceedings...** New York: ACM Press, 1994. p.213–222.

HINCKLEY, K.; TULLIO, J.; PAUSCH, R.; PROFFITT, D.; KASSELL, N. Usability Analysis of 3D Rotation Techniques. In: ANNUAL ACM SIGGRAPH SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, UIST, 10., 1997, Banff, AB, Canada. **Proceedings...** New York: ACM Press, 1997. p.14–17.

HOUDE, S. Iterative Design of an Interface for Easy 3-D Direct Manipulation. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1992, Monterey, CA, USA. **Proceedings...** New York: ACM Press, 1992. p.135–142.

HUFF, R.; DIETRICH, C. A.; NEDEL, L. P.; FREITAS, C. M. D. S.; COMBA, J. L. D.; OLABARRIAGA, S. D. Erasing, Digging and Clipping in Volumetric Datasets with One or Two Hands. In: ACM INTERNATIONAL CONFERENCE ON VIRTUAL REALITY CONTINUUM AND ITS APPLICATIONS, VRCIA, 2006, Hong Kong. **Proceedings...** New York: ACM Press, 2006. p.271–278.

HUFF, R.; SILVA, I. C. S. da; FREITAS, C. M. D. S.; NEDEL, L. P. Seleção de Objetos em Ambientes Virtuais com Mouse 3D. In: SBC SYMPOSIUM ON VIRTUAL REALITY, SVR, 8., 2006, Belém, PA, Brazil. **Proceedings...** [S.l.]: Brazilian Computer Society, 2006. p.349–360.

IKITS, M.; KNISS, J. M.; LEFOHN, A.; HANSEN, C. D. **Volume Rendering Techniques**. Boston, MA, USA: Addison Wesley, 2004. p.667–6692.

JACOB, R. J. K. What You Look At Is What You Get: eye movement-based interaction techniques. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1990, Seattle, WA, USA. **Proceedings...** New York: ACM Press, 1990. p.11–18.

JACOB, R. J. K.; SIBERT, L. E. The Perceptual Structure of Multidimensional Input Device Selection. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1992, Monterey, CA, USA. **Proceedings...** New York: ACM Press, 1992. p.211–218.

KABBASH, P.; BUXTON, W.; SELLEN, A. Two-Handed Input in a Compound Task. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1994, Boston, MA, USA. **Proceedings...** New York: ACM Press, 1994. p.417–423.

KAUFMAN, A.; DACHILLE, F.; CHEN, B.; BITTER, I.; KREEGER, K.; ZHANG, N.; TANG, Q. Real-time volume rendering. **International Journal of Imaging Systems and Technology**, [S.l.], v.11, n.1, p.44–52, 2000.

KINDLMANN, G. L.; DURKIN, J. W. Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering. In: SYMPOSIUM ON VOLUME VISUALIZATION, VVS, 1998, Research Triangle Park, NC, USA. **Proceedings...** New York: ACM Press, 1998. p.79–86.

KNISS, J.; KINDLMANN, G.; HANSEN, C. Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets. In: IEEE CONFERENCE ON VISUALIZATION, VIS, 2001, San Diego, CA, USA. **Proceedings...** Washington: IEEE Computer Society, 2001. p.255–262.

KRÜGER, J.; WESTERMANN, R. Linear algebra operators for GPU implementation of numerical algorithms. **ACM Transactions on Graphics, TOG**, New York, NY, USA, v.22, n.3, p.908–916, 2003.

LACROUTE, P.; LEVOY, M. Fast volume rendering using a shear-warp factorization of the viewing transformation. In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, SIGGRAPH, 21., 1994, Orlando, FL, USA. **Proceedings...** New York: ACM Press, 1994. p.451–458.

LEGANCHUK, A.; ZHAI, S.; BUXTON, W. Manual and Cognitive Benefits of Two-Handed Input: an experimental study. **ACM Transactions on Computer-Human Interaction**, New York, NY, USA, v.5, n.4, p.326–359, 1998.

LEVOY, M. Display of Surfaces From Volume Data. **IEEE Computer Graphics and Applications**, Los Alamitos, CA, USA, v.8, n.3, p.29–37, 1988.

LEVOY, M. Efficient Ray Tracing of Volume Data. **ACM Transactions on Graphics, TOG**, New York, NY, USA, v.9, n.3, p.245–61, 1990.

LIANG, J.; GREEN, M. Geometric modeling using six degrees of freedom input devices. In: INTERNATIONAL CONFERENCE ON COMPUTER AIDED DESIGN AND COMPUTER GRAPHICS, CAD/CG, 3., 1993, Beijing, China. **Proceedings...** [S.l.: s.n.], 1993. p.217–222.

LIANG, J.; GREEN, M. JDCAD: a highly interactive 3d modeling system. **Computers and Graphics**, [S.l.], v.18, n.4, p.499–506, 1994.

LORENSEN, W. E.; CLINE, H. E. Marching cubes: a high resolution 3d surface construction algorithm. In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, SIGGRAPH, 14., 1987, Anaheim, CA, USA. **Proceedings...** New York: ACM Press, 1987. p.163–169.

MACKENZIE, I. S. Input devices and interaction techniques for advanced computing. **Virtual environments and advanced interface design**, New York, NY, USA, p.437–470, 1995.

MACKINLAY, J.; CARD, S. K.; ROBERTSON, G. G. A Semantic Analysis of the Design Space of Input Devices. **Human-Computer Interaction**, Mahwah, NJ, USA, v.5, n.2, p.145–190, 1990.

MANSSOUR, I. H.; FURUIE, S. S.; OLABARRIAGA, S. D.; FREITAS, C. M. D. S. Visualizing Inner Structures in Multimodal Volume Data. In: BRAZILIAN SYMPOSIUM ON COMPUTER GRAPHICS AND IMAGE PROCESSING, SIBGRAPI, 15., 2002, Fortaleza, CE, Brazil. **Proceedings...** Washington: IEEE Computer Society, 2002. p.51–58.

MEISSNER, M.; HOFFMANN, U.; STRASSER, W. Enabling classification and shading for 3D texture mapping based volume rendering using OpenGL and extensions. In: IEEE CONFERENCE ON VISUALIZATION, VIS, 10., 1999, San Francisco, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1999. p.207–214.

MINE, M. R. **Virtual Environment Interaction Techniques**. Chapel Hill, NC, USA: [s.n.], 1995.

MINE, M. R.; FREDERICK P. BROOKS, J.; SEQUIN, C. H. Moving objects in space: exploiting proprioception in virtual-environment interaction. In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, SIGGRAPH, 24., 1997, Los Angeles, CA, USA. **Proceedings...** New York: ACM Press, 1997. p.19–26.

NEDEL, L. P.; FREITAS, C. M. D. S.; JACOB, L. J.; PIMENTA, M. S. Testing the Use of Egocentric Interactive Techniques in Immersive Virtual Environments. In: IFIP TC13 INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION, INTERACT, 9., 2003, Zürich, Switzerland. **Proceedings...** Amsterdam: IOS Press, 2003. p.471 – 478.

PARVIAINEN, J.; SAINIO, N.; RAISAMO, R. Perceiving Tools in 3D Sculpting. In: ASIA-PACIFIC COMPUTER AND HUMAN INTERACTION, APCHI, 6., 2004, Rotorua, New Zealand. **Proceedings...** Berlin: Springer-Verlag, 2004. p.328–337.

PFISTER, H.; HARDENBERGH, J.; KNITTEL, J.; LAUER, H.; SEILER, L. The VolumePro Real-Time Ray-Casting System. In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, SIGGRAPH, 26., 1999, Los Angeles, CA, USA. **Proceedings...** New York: ACM Press, 1999. p.251–260.

PIERCE, J. S.; FORSBERG, A. S.; CONWAY, M. J.; HONG, S.; ZELEZNIK, R. C.; MINE, M. R. Image Plane Interaction Techniques in 3D Immersive Environments. In: SYMPOSIUM ON INTERACTIVE 3D GRAPHICS, SI3D, 1997, Providence, RI, USA. **Proceedings...** New York: ACM Press, 1997. p.39–43.

POUPYREV, I.; BILLINGHURST, M.; WEGHORST, S.; ICHIKAWA, T. The Go-Go Interaction Technique: non-linear mapping for direct manipulation in VR. In: ANNUAL ACM SIGGRAPH SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, UIST, 9., 1996, Seattle, WA, United States. **Proceedings...** New York: ACM Press, 1996. p.79–80.

POUPYREV, I.; WEGHORST, S.; BILLINGHURST, M.; ICHIKAWA, T. Egocentric Object Manipulation in Virtual Environments: evaluation of interaction techniques. **Computer Graphics Forum**, Malden, MA, USA, v.17, n.3, p.41–52, 1998.

REZK-SALAMA, C.; ENGEL, K.; BAUER, M.; GREINER, G.; ERTL, T. Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-Textures and Multi-Stage Rasterization. In: ACM SIGGRAPH/EUROGRAPHICS WORKSHOP ON GRAPHICS HARDWARE, HWWS, 2000, Interlaken, Switzerland. **Proceedings...** New York: ACM Press, 2000. p.109–118.

SACHS, E.; ROBERTS, A.; STOOPS, D. 3-Draw: a tool for designing 3d shapes. **IEEE Computer Graphics and Applications**, Los Alamitos, CA, USA, v.11, n.6, p.18–26, 1991.

SHAW, C. D. Pain and Fatigue in Desktop VR: initial results. In: GRAPHICS INTERFACE, GI, 1998, Vancouver, BC, Canada. **Proceedings...** [S.l.]: Canadian Human-Computer Communications Society, 1998. p.185–192.

SONG, D.; NORMAN, M. Nonlinear Interactive Motion Control Techniques for Virtual Space Navigation. In: IEEE VIRTUAL REALITY ANNUAL INTERNATIONAL SYMPOSIUM, VR, 1993, Seattle, WA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1993. p.111–117.

SOUZA, M.; CENTENO, T.; PEDRINI, H. Integrando reconstrução 3D de imagens tomográficas e prototipagem rápida para a fabricação de modelos médicos. **Revista Brasileira de Engenharia Biomédica**, [S.l.], v.19, n.2, p.103–115, 2003.

STOAKLEY, R.; CONWAY, M.; PAUSCH, R. F. Virtual Reality on a WIM: interactive worlds in miniature. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1995, Vancouver, BC, Canada. **Proceedings...** New York: ACM Press, 1995. p.265–272.

UGHINI, C. S.; BLANCO, F. R.; PINTO, F. M.; FREITAS, C. M. D. S.; NEDEL, L. P. EyeScope: a 3D interaction technique for accurate object selection in immersive environments. In: SBC SYMPOSIUM ON VIRTUAL REALITY, SVR, 8., 2006, Belém, PA, Brazil. **Proceedings...** [S.l.]: Brazilian Computer Society, 2006. p.77–88.

WANG, S. W.; KAUFMAN, A. E. Volume Sculpting. In: SYMPOSIUM ON INTERACTIVE 3D GRAPHICS, SI3D, 1995, Monterey, CA, USA. **Proceedings...** New York: ACM Press, 1995. p.151–156.

WARE, C.; JESSOME, D. R. Using the Bat: a six-dimensional mouse for object placement. In: GRAPHICS INTERFACE, 1988, Edmonton, AB, Canada. **Proceedings...** Toronto: Canadian Information Processing Society, 1988. p.119–124.

WEILER, M.; KRAUS, M.; MERZ, M.; ERTL, T. Hardware-Based Ray Casting for Tetrahedral Meshes. In: IEEE CONFERENCE ON VISUALIZATION, VIS, 2003, Seattle, WA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2003. p.333–340.

WEISKOPF, D.; ENGEL, K.; ERTL, T. Volume Clipping via Per-Fragment Operations in Texture-Based Volume Visualization. In: IEEE CONFERENCE ON VISUALIZATION, VIS, 2002, Boston, MA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2002. p.93–100.

WEISKOPF, D.; ENGEL, K.; ERTL, T. Interactive Clipping Techniques for Texture-Based Volume Visualization and Volume Shading. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, CA, USA, v.9, n.3, p.298–312, 2003.

WESTERMANN, R.; ERTL, T. Efficiently Using Graphics Hardware in Volume Rendering Applications. In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, SIGGRAPH, 25., 1998, Orlando, FL, USA. **Proceedings...** New York: ACM Press, 1998. p.169–177.

WESTOVER, L. Footprint Evaluation for Volume Rendering. In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, SIGGRAPH, 17., 1990, Dallas, TX, USA. **Proceedings...** New York: ACM Press, 1990. p.367–376.

APÊNDICE A QUESTIONÁRIO PRÉ-TESTE PARA AVALIAÇÃO DAS TÉCNICAS DE INTERAÇÃO

Nome:

Idade:

Ocupação/Área de atuação:

1) Assinale com um X o seu grau de familiaridade com relação aos equipamentos e programas abaixo relacionados:

	Ótimo	Bom	Fraco	Nenhum
Editores de imagens (Ex.: Photoshop, Gimp, Corel Photo-Paint, etc.)				
Aplicativos para manipulação de objetos 3D (Ex. CAD, 3D Studio, Blender, etc)				
Jogos 2D				
Jogos 3D				
Mouse convencional 2D				
Mouse 3D				
Teclado				
Capacete de Realidade Virtual (<i>HMD</i>)				
Óculos de Realidade Virtual (<i>Shutter Glasses</i>)				
Luva de Realidade Virtual (<i>Data Glove</i>)				
Rastreadores de posição (<i>FOB</i>)				
Outros:				

2) Você é destro?

Sim

Não

3) Você é daltônico?

Sim

Não

APÊNDICE B QUESTIONÁRIO PÓS-TESTE PARA AVALIAÇÃO DAS TÉCNICAS DE INTERAÇÃO

1. Em geral, qual o conjunto de ferramentas mais fácil de usar?
 Ferramentas 2D Ferramentas 3D

2. Sobre as ferramentas *Borracha*, qual foi a mais fácil de usar?
 Borracha 2D Borracha 3D

3. Sobre as ferramentas *Escavadeira*, qual foi a mais fácil de usar?
 Escavadeira 2D Escavadeira 2D

4. Sobre as ferramentas *Guilhotina*, qual foi a mais fácil de usar?
 Guilhotina 2D Guilhotina 3D

5. A utilização das seguintes opções de ferramentas foi útil na execução das tarefas?

	2 D		3 D	
	Sim	Não	Sim	Não
Planos Axial, Coronal e Sagital				
Fatias Axial, Coronal e Sagital				
Menu de Translação				
Menu de Rotação				

6. A utilização das seguintes opções do volume foi útil na execução das tarefas?

	Sim	Não
Bounding Box		
<i>Grid and Rulers</i> (Paredes)		
<i>Guide Lines</i> (Linhas e sombras da <i>Escavadeira</i>)		
Visualização somente da parte vermelha ou cinza do volume		

7. A visualização da fatia de corte da ferramenta Guilhotina 2D foi útil?
 Sim Não

8. A interface foi fácil de usar?
 Sim Não

9. Comentários:

APÊNDICE C QUESTIONÁRIO PRÉ-TESTE PARA AVALIAÇÃO DOS DISPOSITIVOS DE ENTRADA

Nome:

Idade:

Ocupação / Área de atuação:

1) Assinale com um X o seu grau de familiaridade com relação aos equipamentos e programas abaixo relacionados:

	Ótimo	Bom	Fraco	Nenhum
Aplicativos para visualização de objetos 3D (Ex. CAD, 3D Studio, Blender, etc)				
Jogos 2D				
Jogos 3D				
Mouse convencional 2D				
Mouse 3D				
Teclado				
Capacete (<i>HMD</i>)				
Óculos (<i>Shutter Glasses</i>)				
Luva (<i>Data Glove</i>)				
Rastreadores de posição (<i>FOB</i>)				
Guidão virtual				
Outros:				

2) Você é: () Destro () Canhoto

APÊNDICE D QUESTIONÁRIO PÓS-TESTE PARA AVALIAÇÃO DOS DISPOSITIVOS DE ENTRADA

1. Em geral, qual foi o dispositivo mais fácil de usar?

Mouse 2D

Mouse 3D

2. Para a manipulação do volume, qual o dispositivo mais fácil de usar?

Mouse 2D

Mouse 3D

3. Para a manipulação da ferramenta *Borracha*, qual o dispositivo mais fácil de usar?

Mouse 2D

Mouse 3D

4. Para a manipulação da ferramenta *Escavadeira*, qual o dispositivo mais fácil de usar?

Mouse 2D

Mouse 3D

5. Para a manipulação da ferramenta *Guilhotina*, qual o dispositivo mais fácil de usar?

Mouse 2D

Mouse 3D

6. Em geral, qual a melhor combinação para a execução das tarefas?

Volume com mouse 2D e ferramenta com mouse 2D

Volume com mouse 2D e ferramenta com mouse 3D

Volume com mouse 3D e ferramenta com mouse 2D

Volume com mouse 3D e ferramenta com mouse 3D

7. Observações: