

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

**Visualização de Estruturas Internas
em Volumes de Dados Multimodais**

por

ISABEL HARB MANSSOUR

Tese submetida à avaliação,
como requisito parcial para a obtenção do grau de
Doutor em Ciência da Computação

Porto Alegre, setembro de 2002

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Manssour, Isabel Harb

Visualização de Estruturas Internas em Volumes de Dados Multimodais / por Isabel Harb Manssour. — Porto Alegre: PPGC da UFRGS, 2002.

135 f.: il.

Tese (doutorado) — Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2002. Orientador: Freitas, Carla Maria Dal Sasso; Co-orientador: Furuie, Sérgio Shiguemi.

1. Visualização volumétrica. 2. Estruturas internas. 3. Ferramentas de corte. 4. Funções de transferência. 5. Volumes de dados multimodais. 6. Imagens médicas. I. Freitas, Carla Maria Dal Sasso. II. Furuie, Sérgio Shiguemi. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof^a. Wrana Maria Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fensterseifer

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

Coordenador do PPGC: Prof. Carlos Alberto Heuser

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

*“Se as coisas são inatingíveis...ora!
Não é motivo para não querê-las...
Que tristes os caminhos, se não fora a presença distante das estrelas!”
(Mário Quintana)*

Agradecimentos

Um agradecimento especial à minha orientadora Carla M. D. S. Freitas, com quem muito aprendi nos mais de dez anos em que recebi sua orientação, inicialmente como bolsista de iniciação científica e depois como aluna de mestrado e doutorado. Seu apoio foi fundamental nas horas em que pensei que jamais concluiria o doutorado. Obrigada também por todo incentivo e atenção que recebi, e pelas várias tardes de feriados e finais de semana em que nos reuníamos para escrever *papers*, tutoriais e para conversar sobre a tese.

Um agradecimento especial, também, à professora Silvia D. Olabarriaga pelas várias reuniões, pela ajuda na elaboração de *papers*, pelas sugestões e ensinamentos e, principalmente, por ter auxiliado a encontrar a famosa “pergunta” da minha tese.

Ao meu co-orientador, professor Sérgio S. Furuie, que teve uma forte influência na escolha do tema da tese, pela atenção que recebi todas as vezes que estive em São Paulo.

À Luciana P. Nedel, pela grande ajuda na definição do *framework*, pelas várias conversas, troca de idéias e auxílio na elaboração de *papers*.

Ao Marcelo Yamaguti, que muito me ajudou na modelagem UML, e, principalmente, sugeriu a utilização do padrão MVC.

Ao Marcelo R. M. Silva, pelos vários e vários *e-mails* que trocamos. Pela paciência e pela ajuda na implementação do protótipo.

Ao João Batista Oliveira, por suas idéias e sugestões, e por solucionar muitas das minhas dúvidas, relacionadas à tese ou não.

Ao Márcio S. Pinho, pela ajuda na implementação do algoritmo de *ray casting*.

Ao Anderson Maciel, por ter ajudado a compilar o código no ambiente Linux, pelas várias conversas e sugestões na etapa de elaboração do *framework* e pela imagem cedida para utilização no *paper* do VG’01.

À Isabel Cristina S. da Silva, pela implementação do plano de corte.

Ao João Comba, pelo auxílio na revisão de *papers* e pela troca de idéias sobre a visualização através do mapeamento de textura 3D.

Aos meus Bolsistas de Iniciação Científica, Fernando S. Pereira, Tito L. Lapis e Leandro Iglezias, pelos ícones elaborados para a interface do protótipo, pelo desenvolvimento de alguns dos seus módulos e pela modelagem no *Rational*.

Ao Pedro Paulo Magalhães, que teve a paciência de olhar o meu código para tentar encontrar o problema da iluminação.

Ao Sérgio Cechim e Rafael Avila pela ajuda com o Latex.

Ao Maurício Anes, do Centro de Diagnóstico por Imagens do Hospital São Lucas da PUCRS, pela atenção que teve comigo e por conseguir algumas imagens para serem utilizadas no desenvolvimento do meu trabalho.

Ao Instituto de Informática da UFRGS, pelos auxílios recebidos que permitiram que eu participasse de congressos que foram fundamentais para a minha formação. Também aos seus funcionários, especialmente o Luis Otávio, a Silvania, a Ida Rossi e a Beatriz Haro, pela atenção que recebi durante todos estes anos que trabalhei nas dependências do Instituto.

À Faculdade de Informática da PUCRS, pelo auxílio que recebi para participar do VG’01.

Às agências de fomento, CAPES e FAPERGS, pelo apoio financeiro.

Aos meus pais, Ivone e Elias (*in memoriam*) pela educação, dedicação, incentivo, amor e carinho que sempre recebi. E principalmente à minha mãe, por toda paciência, apoio, compreensão e ajuda.

À tia Gina e ao tio Cheim, por todo carinho, apoio, compreensão e ajuda, principalmente da tia Gina, que junto com a mãe, muito me ajudou com as provas...

Aos meus familiares, Sérgio, Ana, Flávia, Kika, Lari, Elô, Linda e Gus, por compreenderem minha ausência, e impaciência, e por me ajudarem muitas vezes nas minhas atividades de professora e, ao mesmo, de aluna. Um agradecimento especial ao Sérgio e à Ana pela paciência em digitar as referências no BibTex e revisar o texto.

Às minhas amigas Ale, Lu, Dê, Roberta e Lis, pelo apoio, pelas várias conversas e troca de idéias, e pela amizade. À Ale também pela revisão de parte do texto da tese.

E a todos que me ajudaram e, diretamente ou indiretamente, contribuíram para a realização deste trabalho.

Muito Obrigada!

Sumário

Lista de Abreviaturas	8
Lista de Figuras	9
Lista de Tabelas	11
Resumo	12
Abstract	13
1 Introdução	14
1.1 Contexto do Trabalho	14
1.2 Objetivos e Metodologia	16
1.3 Organização do Texto	18
2 Visualização Volumétrica	19
2.1 Introdução	19
2.2 Algoritmo de <i>Ray Casting</i>	22
2.3 Visualização de Estruturas Internas	26
2.4 Visualização de Múltiplos Volumes de Dados	27
2.5 Comentários Finais	29
3 Ferramentas de Corte	30
3.1 Abordagem Unificada para Ferramentas de Corte	30
3.2 Corte por Geometria	33
3.3 Corte por Geometria e Conteúdo	38
3.4 Discussão	42
3.5 Comentários Finais	45
4 Funções de Transferência	46
4.1 Introdução	46
4.2 Função de Transferência de Opacidade	47
4.3 Função de Transferência de Cor	53
4.4 Especificação de Funções de Transferência	55
4.5 Comentários Finais	62
5 Visualização de Volumes Multimodais	63
5.1 Requisitos	63
5.2 Sistemas de Visualização de Volumes Multimodais	64
5.3 Técnicas de Visualização de Volumes Multimodais	66
5.3.1 Normal Projection Technique	68
5.3.2 Normal Fusion Technique	69
5.3.3 Spectral Volume Rendering	70
5.3.4 Integração Baseada no Algoritmo de <i>Ray Casting</i>	72
5.4 Comentários Finais	75

6	Visualização de Estruturas Internas em Volumes de Dados Multimodais	76
6.1	Alternativas para a Visualização de Estruturas Internas	76
6.2	Abordagem Adotada para Visualização de Dados Multimodais	79
6.3	Abordagem para a Visualização de Estruturas Internas em Volumes de Dados Multimodais	81
6.4	Discussão	88
7	<i>Framework</i> para Visualização de Imagens Médicas	91
7.1	Arquitetura do <i>Framework</i>	91
7.2	Ambiente de Implementação e Interface do Protótipo	95
7.3	Comentários Finais	98
8	Estudo de Caso	100
8.1	Introdução	100
8.2	Imagens Médicas	101
8.3	Visualização Monomodal	103
8.4	Visualização de Volumes de Dados Multimodais	105
8.5	Comentários Finais	109
9	Conclusões e Trabalhos Futuros	114
9.1	Introdução	114
9.2	Avaliação Geral e Contribuições	114
9.3	Trabalhos Futuros	115
9.3.1	Curto Prazo	116
9.3.2	Médio Prazo	116
	Anexo Volumes de Dados Sintetizados	118
	Bibliografia	121

Lista de Abreviaturas

CT	Tomografia Computadorizada (ou Computed Tomography)
CTA	Angiografia por Tomografia Computadorizada (ou Computed Tomography Angiography)
MRI	Ressonância Magnética (ou Magnetic Resonance Imaging)
MRA	Angiografia por Ressonância Magnética (ou Magnetic Resonance Angiography)
fMRI	Ressonância Magnética Funcional (ou Functional Magnetic Resonance Imaging)
PET	Positron-Emission Tomography
SPECT	Single-Photon Emission Tomography
2D	Bidimensional(is)
3D	Tridimensional(is)
MIP	Maximum Intensity Projection
CVR	Confocal Volume Rendering
CVP	Closest Vessel Projection
RADC	Ray Acceleration by Distance Coding
MI	Mutual Information
ROI	Região de Interesse (ou Region of Interest)
VOI	Volume de Interesse (ou Volume of Interest)
RGB	Red-Green-Blue
HLS	Hue-Lightness-Saturation
HSV	Hue-Saturation-Value
API	Application Programming Interface
MVC	Model-View-Controller
UML	Unified Modeling Language
FLTK	Fast Light Tool Kit
VPat	Visualization and Interaction with Virtual Patients

Lista de Figuras

FIGURA 2.1 – Etapas da visualização através de superfícies	20
FIGURA 2.2 – Etapas da visualização direta de volume	21
FIGURA 2.3 – Esquema genérico do algoritmo de <i>ray casting</i> (<i>front-to-back</i>)	23
FIGURA 2.4 – <i>Pipeline</i> de <i>volume rendering</i>	25
FIGURA 2.5 – Imagens 2D usadas para gerar a estrutura 3D	28
FIGURA 3.1 – Pontos de intersecção (r_1, r_2) do raio com o volume de dados	31
FIGURA 3.2 – Planos de corte (c_1, c_2) que delimitam o intervalo de amostragem	31
FIGURA 3.3 – Volumes sintetizados	32
FIGURA 3.4 – Exemplo de (a) um corte axial	34
FIGURA 3.5 – Casos especiais na utilização de planos de corte	35
FIGURA 3.6 – Exemplo da utilização de (a) um ou (b,c) dois planos de corte	35
FIGURA 3.7 – Exemplo de um cubo usado para realizar o corte por exclusão	36
FIGURA 3.8 – Dois ângulos de visão do corte com um cubo	36
FIGURA 3.9 – Exemplo de um cubo usado para especificar um VOI	37
FIGURA 3.10 – Seleção de um subvolume com um cubo	37
FIGURA 3.11 – Controle dos parâmetro no CVR [MUL 2000]	39
FIGURA 3.12 – Técnica de <i>Confocal Volume Rendering</i>	40
FIGURA 3.13 – Princípio do CVP	41
FIGURA 3.14 – Comparação entre a (a) técnica de <i>Closest Vessel Projection</i>	41
FIGURA 3.15 – Corte por inclusão (a=conteúdo, b=geometria)	44
FIGURA 3.16 – Visualização usando intervalo de valores	45
FIGURA 4.1 – Função de transferência linear	48
FIGURA 4.2 – Exemplos da aplicação de funções de transferência lineares	48
FIGURA 4.3 – Função tipo “rampa”	48
FIGURA 4.4 – Função de transferência no formato de um trapézio	49
FIGURA 4.5 – Função de transferência exponencial	49
FIGURA 4.6 – Função de transferência do tipo bloco	50
FIGURA 4.7 – Imagens resultantes da utilização da função de transferência	50
FIGURA 4.8 – Exemplos da utilização da função de transferência de opacidade	52
FIGURA 4.9 – Uso (a) da magnitude do gradiente	52
FIGURA 4.10 – Imagem gerada com a utilização de uma tabela de cor	54
FIGURA 4.11 – Exemplo do uso de funções de transferência 2D (a) e 3D (b)	55
FIGURA 4.12 – Interface do sistema que apresenta um conjunto de imagens	58
FIGURA 4.13 – Interface baseada em grafo	59
FIGURA 4.14 – Interface baseada em planilha	59
FIGURA 4.15 – Especificação de uma função de transferência	60
FIGURA 5.1 – Exemplo da técnica de <i>linked feature display</i>	67
FIGURA 5.2 – Janela Multimodal	68
FIGURA 5.3 – Princípio da técnica de <i>Normal Projection</i> [ZUI 95]	69
FIGURA 5.4 – Visualização obtida com a técnica de <i>Normal Fusion</i>	71
FIGURA 5.5 – Visualização esquemática do <i>spectral volume rendering</i>	71
FIGURA 5.6 – <i>Pipeline</i> de visualização de dados multimodais	73
FIGURA 5.7 – Integração na etapa de intensidade da imagem	74

FIGURA 5.8 – Integração na etapa de acumulação	74
FIGURA 5.9 – Integração na etapa de iluminação	75
FIGURA 6.1 – Combinação de técnicas de corte e funções de transferência	77
FIGURA 6.2 – Combinação de técnicas de <i>rendering</i>	78
FIGURA 6.3 – Volumes sintetizados usados para exemplificar	80
FIGURA 6.4 – Visualização integrada dos volumes sintetizados	81
FIGURA 6.5 – Visualização integrada considerando que os volumes sintetizados	81
FIGURA 6.6 – Tabela de cor usada na geração das imagens	83
FIGURA 6.7 – Exemplos de corte por geometria	84
FIGURA 6.8 – Exemplos de corte (a,b,c) por conteúdo	85
FIGURA 6.9 – Visualização dos volumes multimodais	86
FIGURA 6.10 – Visualização dos volumes multimodais através da técnica de CVR	86
FIGURA 6.11 – Corte por inclusão (a=conteúdo, b=geometria)	87
FIGURA 6.12 – (a) Um volume da esfera usado para definir a cor	88
FIGURA 6.13 – Visualização de três volumes de dados multimodais	90
FIGURA 7.1 – Diagrama de caso de uso	92
FIGURA 7.2 – Modelo conceitual simplificado	93
FIGURA 7.3 – Diagrama de colaboração	94
FIGURA 7.4 – Interface do protótipo	96
FIGURA 7.5 – Extração de um plano de corte oblíquo	97
FIGURA 7.6 – Protótipo em execução no Linux	98
FIGURA 8.1 – Visualização do volume de dados de um dente	103
FIGURA 8.2 – Visualização com plano de corte	104
FIGURA 8.3 – Exemplo de corte por conteúdo	104
FIGURA 8.4 – Utilização da técnica de CVR com o volume de dados do dente	105
FIGURA 8.5 – Exemplo da visualização integrada	106
FIGURA 8.6 – Imagens geradas por um exame de fMRI	106
FIGURA 8.7 – Imagens dos dados de fMRI que foram sintetizados	107
FIGURA 8.8 – Visualização dos dados multimodais (MRI e fMRI sintetizado)	108
FIGURA 8.9 – Utilização das ferramentas de (a) volume de corte	109
FIGURA 8.10 – Imagens geradas por um exame de medicina nuclear	109
FIGURA 8.11 – Imagem SPECT	110
FIGURA 8.12 – Tabela de cor utilizada na geração das imagens	110
FIGURA 8.13 – Visualização dos volumes de dados de MRI e medicina nuclear	111
FIGURA 8.14 – Utilização da técnica de CVR	112
FIGURA 8.15 – Exemplos da utilização de técnicas de corte	113
FIGURA A.1 – Cortes axial, coronal e sagital (de cima para baixo)	118
FIGURA A.2 – Tabela de cor usada para gerar as imagens	119
FIGURA A.3 – Visualização do volume de dados que contém uma esfera	119
FIGURA A.4 – Visualização do volume de dados que contém vários parale- lepípedos	120
FIGURA A.5 – Visualização do volume de dados que contém um “cubo”	120

Lista de Tabelas

TABELA 3.1 – Determinação de a e b para cada técnica de corte	42
TABELA 3.2 – Diferentes combinações de acordo com a e b	43
TABELA 4.1 – Abordagens para a especificação das funções	62
TABELA 6.1 – Diferentes combinações em volumes de dados multimodais	82

Resumo

Com o aperfeiçoamento de técnicas de aquisição de imagens médicas, como, por exemplo, a tomografia computadorizada e a ressonância magnética, a capacidade e a fidelidade do diagnóstico por imagens foram ampliadas. Atualmente, existe a tendência de utilizarem-se imagens obtidas através de diversas modalidades para um único diagnóstico, principalmente no caso de doenças graves. Entretanto, o registro e a fusão dessas imagens, chamadas multimodais, em uma única representação 3D do paciente é uma tarefa extremamente difícil, que consome tempo e que está sujeita a erros. Sendo assim, a integração de imagens de diferentes modalidades tem sido objeto de pesquisa sob a denominação de Visualização de Volumes de Dados Multimodais. Sistemas desenvolvidos com este objetivo são usados, principalmente, para combinar informações metabólicas e funcionais com dados de anatomia, aumentando a precisão do diagnóstico, uma vez que possibilitam extrair uma superfície ou região da imagem que apresenta a anatomia e, então, observar a atividade funcional na outra modalidade.

Durante a análise de tais imagens, os médicos estão interessados em visualizar e quantificar diferentes estruturas. Seus objetivos envolvem, por exemplo, a visualização de artérias e órgãos do corpo humano para análise de patologias, tais como tumores, má-formações artério-venosas, ou lesões em relação às estruturas que as circundam. Assim, um dos principais objetivos de um algoritmo de visualização volumétrica é permitir a identificação e exploração de estruturas internas no volume. Como o volume é normalmente um “bloco de dados”, não se pode visualizar o seu interior, a menos que se assuma que é possível ver através de voxels transparentes, ou que é possível remover voxels que estão na frente da região na qual o usuário está interessado, o que é feito através de técnicas de segmentação ou de corte.

Este trabalho apresenta uma abordagem para a visualização de estruturas internas em volumes de dados multimodais. A abordagem está fundamentada na utilização de ferramentas de corte, tanto geométricas quanto baseadas em conteúdo, evitando, assim, o uso de técnicas de segmentação; e na integração dos dados multimodais na etapa de acumulação do *pipeline* de visualização volumétrica. Considerando que as aplicações que suportam este tipo de visualização envolvem a integração de várias ferramentas, tais como registro, corte e visualização, também é apresentado o projeto de um *framework* que permite esta integração e um alto grau de interação com o usuário. Para teste e validação das técnicas de visualização de estruturas internas propostas e do algoritmo desenvolvido, que consiste numa extensão do algoritmo de *ray casting* tradicional, foram implementadas algumas classes desse *framework*. Uma revisão baseada na análise e na classificação das ferramentas de corte e funções de transferência, que correspondem a técnicas que permitem visualizar estruturas internas, também é apresentada.

Palavras-chave: Visualização volumétrica, estruturas internas, ferramentas de corte, funções de transferência, volumes de dados multimodais, imagens médicas.

TITLE: “INNER STRUCTURES VISUALIZATION IN MULTIMODAL VOLUME DATA”

Abstract

With the evolution of image acquisition techniques, e.g. computed tomography and magnetic resonance imaging, the capacity and fidelity of image diagnosis were extended. Nowadays, the simultaneous use of images obtained from different sources is common in medical diagnosis, especially in serious diseases. However, mental integration of these multivariate data into a single 3D representation is non-trivial, time-consuming and error-prone. To overcome this problem, many researchers have been developing suitable strategies to integrate important characteristics of multiple data sets into a single visual representation, with the “label” of Multimodal Volume Data Visualization. Systems developed with this goal are used to combine metabolic and functional information with anatomical data, improving the diagnosis precision, because it allows the extraction of a surface or region that represents the anatomy and the further processing of functional activity in the other modality.

During the analysis of such images, physicians are interested in visualizing and quantifying different structures. Their goals involve, for example, the visualization of vessels and human organs for the analysis of pathologies, like tumor or arterio-venous malformations, or lesions related with the surrounding structures. One of the main goals of volume visualization algorithm is to allow the identification and exploration of the volume inner structures. Since the volume is usually a “block of data”, we cannot see inside it unless we assume either that we can see through transparent voxels, or we remove voxels that are in front of the region where the user is interested, with image segmentation or volume cutting techniques.

This work presents an approach for the inner structures visualization in multimodal volume data. This approach is based on the utilization of cutting tools, which use geometrical objects as well as voxel information, avoiding the utilization of segmentation techniques, and on the integration of multimodal data at the accumulation level of the volume rendering pipeline. Considering that the applications which support this type of visualization involve the integration of several tools, such as registration, cutting and visualization, the design of a framework to allow this integration with a high degree of user interaction is also presented. The main classes of this framework were implemented to validate the algorithm and the proposed inner structures visualization techniques. A review based on analysis and classification of cutting tools and transfer functions, which are the techniques that allow inner structures visualization, is also presented.

Keywords: volume visualization, inner structures, cutting tools, transfer function, multimodal volume data, medical imaging.

1 Introdução

1.1 Contexto do Trabalho

A Visualização Científica congrega técnicas oriundas principalmente da Computação Gráfica e do Processamento de Imagens. Suas aplicações visam, sobretudo, possibilitar a análise visual e a exploração de dados coletados da natureza ou obtidos a partir do processamento científico. A Visualização Volumétrica, dentro do contexto da Visualização Científica, denota o conjunto de técnicas utilizadas na apresentação de dados científicos de volume, que consistem em dados associados a regiões ou pontos regular ou irregularmente distribuídos no espaço. Estes dados são geralmente produzidos por fontes como supercomputadores e sensores diversos como, por exemplo, instrumentos de aquisição de imagens médicas. O principal objetivo da Visualização Volumétrica é mostrar o interior desses volumes a fim de explorar de diversas maneiras sua estrutura complexa e facilitar sua compreensão. Muitas técnicas descritas na literatura dedicam-se a investigar diferentes maneiras de visualizar estruturas internas em volumes de dados.

Uma das grandes áreas de aplicação da Visualização Científica é a Medicina, pois com o desenvolvimento de técnicas de aquisição de imagens médicas, que fornecem informações mais detalhadas do ponto de vista de resolução e de distinção de tecidos, a capacidade e a fidelidade do diagnóstico por imagens foram ampliadas. Exemplos de exames utilizados para fazer o diagnóstico por imagens incluem a Tomografia Computadorizada (CT), a Ressonância Magnética (MRI) e as técnicas de CT por emissão, como a *Positron-Emission Tomography* (PET). O aumento da utilização destas imagens fez com que pesquisadores da área de Computação Gráfica e Processamento de Imagens começassem a desenvolver novas técnicas de visualização e interação com as mesmas.

Considerando as modalidades de aquisição de dados médicos existentes, é possível identificar duas grandes categorias: anatômicas e funcionais. As modalidades anatômicas, que retratam principalmente a morfologia, incluem exames de raio X, CT, MRI e ultra-som. A segunda categoria, que possibilita a representação de informações funcionais e metabólicas, engloba imagens de cintilografia, *Single Photon Emission Computed Tomography* (SPECT) e PET, que correspondem às imagens de medicina nuclear, e MRI funcional (fMRI). Estas modalidades mostram aspectos diferentes, complementares e/ou parcialmente sobrepostos da anatomia examinada [MAI 98, STO 98, PLU 2002].

A partir da década de 80 foram desenvolvidos métodos para a visualização tridimensional de volumes formados por várias imagens obtidas a partir das modalidades mencionadas. A análise e a visualização destas imagens em estações de trabalho passaram a ser utilizadas como alternativa (embora ainda restrita a grandes centros) à interpretação tradicional feita através de negatoscópio.

Durante a análise de tais imagens, os médicos estão interessados em visualizar e quantificar diferentes estruturas. Seus objetivos envolvem, por exemplo, a visualização de artérias e órgãos do corpo humano, para análise de patologias, tais como tumores, padrões anormais de crescimento ou malformações artério-venosas. Também é muito importante mostrar uma lesão em relação à anatomia normal ou em relação às estruturas que a circundam. Algumas técnicas específicas para a vi-

sualização de estruturas internas foram desenvolvidas. Tais técnicas são centradas em medidas determinadas com base na anatomia [MUL 2000], ou voltadas para a visualização de estruturas específicas, como vasos sanguíneos [ZUI 95].

Algumas vezes os médicos precisam analisar imagens de uma mesma modalidade obtidas em tempos diferentes, por exemplo, para monitorar o crescimento de um tumor, com o objetivo de verificar a resposta ao tratamento realizado [ROB 94, MAE 98]. Recentemente, vem crescendo a tendência de se utilizar imagens obtidas através de diversas modalidades para um único diagnóstico, principalmente no caso de doenças graves, como o câncer. Entretanto, o registro e a fusão destas imagens denominadas multimodais em uma única representação 3D do paciente é uma tarefa extremamente difícil, que consome tempo e que é sujeita a erros. Sendo assim, a integração de imagens de diferentes modalidades tem sido objeto de pesquisa sob a denominação de Visualização de Volumes de Dados Multimodais.

As técnicas de visualização de dados multimodais preocupam-se com a integração adequada de dados (bi ou tridimensionais) obtidos de aparelhos diferentes. O seu principal objetivo é combinar de maneira adequada os dados complementares das diversas modalidades em uma única imagem. Basicamente, dois tipos de integração multimodal podem ser distinguidos: a combinação de dados anatômicos de modalidades diferentes, tais como CT e MRI; e a combinação de dados funcionais e anatômicos [STO 98].

O primeiro passo para a visualização integrada de volumes multimodais é o alinhamento das imagens provenientes de diferentes modalidades e/ou instantes de tempo. De forma simplificada, este alinhamento, chamado de **registro**, consiste em um processo de mapeamento de pontos de uma imagem para outra para trazê-las para o mesmo sistema de referência. Depois do registro, é necessário combinar as imagens de maneira adequada para exibir as informações dos dois volumes de dados simultaneamente. Este processo é chamado de **fusão**, apesar de muitas vezes os termos registro e fusão aparecerem com significados diferentes na literatura, que podem identificar um único passo ou todo o processo [MAI 98].

Um algoritmo de visualização volumétrica pode ser usado para a exibição simultânea dos múltiplos volumes integrados ou registrados. Além da visualização é importante fornecer **ferramentas de interação**. Tais ferramentas devem permitir extrair medidas e informações funcionais a partir da imagem final (um processo chamado de **quantificação**), e diferentes maneiras de visualizar, explorar e navegar no interior dos volumes.

Freqüentemente, dependendo da aplicação, a implementação de uma técnica de **segmentação** torna-se necessária. O seu principal objetivo é subdividir uma imagem nas partes ou objetos que a constituem, até que todos os objetos de interesse para uma aplicação específica sejam isolados [GON 93]. Em sistemas que suportam a visualização de dados multimodais, um passo de segmentação pode ser necessário para a etapa de registro, visualização ou extração de medidas. Sendo assim, uma vez que pode afetar outras etapas, o desenvolvimento de um método de segmentação pode ser um passo decisivo durante o processo de visualização de dados multimodais. É possível concluir, então, que a construção de um sistema para a visualização de dados multimodais envolve o desenvolvimento e a integração de várias ferramentas, tais como registro, segmentação e visualização interativa.

Sistemas de visualização de dados multimodais são usados, principalmente, para combinar informações metabólicas e funcionais com dados de anatomia, aumen-

tando a precisão do diagnóstico, uma vez que possibilitam extrair uma superfície ou região da imagem que apresenta a anatomia e, então, processar a atividade funcional na outra modalidade. Além disso, também aumentam a confiança na localização de anormalidades funcionais de acordo com a anatomia [STO 98]. Vários autores apresentam um grande número de aplicações que se beneficiam da visualização de dados multimodais. Uma aplicação, considerando apenas a combinação de dados anatômicos, consiste na simulação interativa de neurocirurgia [ROB 94, PLU 2002]. Neste caso, por exemplo, imagens de MRI podem ser usadas para determinar a localização precisa de uma lesão em relação ao osso (imagens de CT), com o objetivo de obter diagnóstico e tratamento mais precisos [STO 98].

A visualização integrada de imagens funcionais e anatômicas é útil para possibilitar a análise do fluxo sanguíneo cerebral e a realização do mapeamento anatômico de funções cerebrais [ROB 94]. Para o planejamento cirúrgico em casos de epilepsia são usadas imagens PET e MRI [STO 98]. Imagens PET e CT são utilizadas para análise da atividade cerebral em pacientes que sofrem de esquizofrenia e depressão [MAG 91]. Quatro modalidades de imagens, PET, SPECT, CT e MRI, são analisadas para avaliação do mal de Alzheimer em pessoas idosas e de sintomas de demência em pacientes com HIV positivo [MAG 91].

Diferentes volumes de dados anatômicos são usados para o planejamento de radioterapia, cuja finalidade é identificar a dose exata que deve ser aplicada num tumor para minimizar os danos causados nos tecidos que o circundam. Assim, imagens de MRI podem ser usadas para delinear a estrutura do tumor e imagens de CT podem ser usadas na execução dos cálculos para preparar o tratamento [ROB 94, MAE 98, COL 98, CAI 99, PLU 2002].

O crescimento da pesquisa em realidade aumentada também provocou a demanda por sistemas de visualização de dados multimodais que permitam a realização de uma cirurgia guiada por imagem. Neste caso, imagens pré-operatórias são utilizadas durante uma cirurgia com o objetivo de navegar numa determinada estrutura [LEV 98, GER 2001].

Da observação destas aplicações é possível constatar que um dos principais objetivos (e usos) de um algoritmo de visualização volumétrica é permitir a identificação de estruturas internas ao(s) volume(s). Como um volume é normalmente um “bloco de dados”, não se pode visualizar o seu interior, a menos que se assuma que é possível ver através de voxels transparentes, ou que é possível remover voxels que estão na frente da região na qual o usuário está interessado, o que é feito através de técnicas de segmentação ou de corte. A primeira possibilidade pode ser alcançada com o uso de tabelas de classificação que atribuem cores e opacidades a diferentes intervalos de valores de voxels, onde opacidade é a medida de transparência de cada voxel.

1.2 Objetivos e Metodologia

Segundo Kaufman et al. [KAU 93], o principal objetivo das técnicas de visualização volumétrica é possibilitar a representação de estruturas existentes no interior do volume de dados, que, assim, podem ser reveladas e exploradas. Além disso, a visualização integrada, principalmente de imagens funcionais e anatômicas, é extremamente útil em várias aplicações médicas. Este tipo de visualização permite uma

localização precisa das anormalidades funcionais em relação à anatomia e aumenta a confiança do observador na sua localização [STO 98]. A partir da identificação destas necessidades, surgiu a motivação para o desenvolvimento de novos algoritmos para visualização de estruturas internas e de arquiteturas para sistemas interativos de visualização de dados mono ou multimodais.

Esta tese se propõe a prover uma abordagem para a visualização de estruturas internas em volumes de dados multimodais, baseada na utilização de ferramentas de corte, tanto associadas a elementos geométricos como a propriedades dos voxels, e evitando a utilização de técnicas de segmentação. Além disso, considerando a integração das várias ferramentas envolvidas, tais como segmentação, registro e visualização, surgiu a necessidade do projeto de um *framework* para integração destas ferramentas, permitindo um alto grau de interação com o usuário. A idéia, neste caso, é possibilitar a utilização na arquitetura proposta de alguns algoritmos otimizados já descritos na literatura.

Para alcançar estes objetivos, o desenvolvimento das seguintes tarefas foram fundamentais: revisão, análise e classificação das técnicas que permitem “ver através” do volume, ou, em outras palavras, que permitem visualizar estruturas internas; e estudo para identificar os requisitos e as questões em aberto no que tange ao desenvolvimento de sistemas de visualização de dados multimodais. As técnicas para a visualização do interior dos volumes incluem as ferramentas de corte e as funções de transferência utilizadas no algoritmo de *ray casting* tradicional, que foi o escolhido para investigação e implementação do protótipo desenvolvido. O principal objetivo desta etapa da pesquisa foi facilitar o entendimento dos pontos positivos e negativos de cada método e a sua aplicabilidade em diferentes tarefas de inspeção interativas. Além disso, a proposta de uma classificação para as ferramentas de corte introduz uma nova abordagem para o seu desenvolvimento, facilitando a implementação e o entendimento do seu funcionamento. A formalização para a descrição das funções de transferência também facilita a sua compreensão e a sua implementação em sistemas que venham a ser elaborados. O protótipo desenvolvido permitiu testar as técnicas de visualização de estruturas internas abordadas e a interação entre as ferramentas, tais como corte e visualização.

O algoritmo de *ray casting* foi implementado neste trabalho para possibilitar o desenvolvimento de novas técnicas de visualização integrada e de estruturas internas e para a realização de um estudo de caso. Outros algoritmos de visualização volumétrica, como por exemplo *splatting* [WES 89, WES 90] ou *shear-warp* [LAC 94], poderiam ter sido utilizados. Entre as diversas razões que levaram à escolha do algoritmo de *ray casting* estão: a sua grande utilização para visualização de dados médicos, uma vez que possibilita a exploração do volume de dados fornecendo imagens de alta qualidade a partir de uma implementação simples; a sua utilização em algumas técnicas de visualização de estruturas internas que foram classificadas e estendidas para dados multimodais (por exemplo, *Confocal Volume Rendering* [MUL 2000]); experiência do grupo de pesquisa na implementação e utilização deste algoritmo [MAN 92, MAN 94, MAN 94a, SIL 2000] e de ferramentas interativas [GRA 96, SIL 2000], o que facilitou a investigação das técnicas para visualizar estruturas internas.

1.3 Organização do Texto

O restante do texto está organizado da seguinte maneira: o capítulo 2 apresenta uma introdução à visualização volumétrica, com ênfase no algoritmo de *ray casting*, escolhido como base para o desenvolvimento do protótipo implementado. Neste capítulo é estabelecida a notação utilizada para a descrição das técnicas de visualização. Também são apresentadas as diferentes abordagens para visualização de estruturas internas e de múltiplos volumes de dados.

No capítulo 3, as ferramentas de corte são descritas, analisadas e apresentadas num esquema genérico. As funções de transferência utilizadas para a geração de diferentes tipos de imagens, incluindo as abordagens existentes para a sua especificação, estão descritas no capítulo 4.

O capítulo 5 aborda os requisitos para a visualização de dados multimodais e descreve algumas técnicas e sistemas relatados na literatura. O algoritmo desenvolvido e as técnicas implementadas para a visualização de estruturas internas em volumes de dados multimodais são apresentados no capítulo 6.

A arquitetura proposta para sistemas de visualização de dados multimodais e o protótipo desenvolvido estão detalhados no capítulo 7. O capítulo 8 apresenta um breve estudo de caso com imagens obtidas por tomografia computadorizada e ressonância magnética no qual são utilizadas as ferramentas implementadas. Uma avaliação geral do trabalho desenvolvido, bem como suas contribuições e possíveis extensões são discutidas no capítulo 9, como conclusão.

2 Visualização Volumétrica

Este capítulo contém uma breve introdução à área de visualização volumétrica, com o objetivo de estabelecer a terminologia utilizada ao longo do trabalho (seção 2.1) e apresentar um panorama geral das pesquisas descritas na literatura. O algoritmo de *ray casting*, que foi o escolhido para investigação, é apresentado na seção 2.2. Uma revisão das técnicas existentes para visualização de estruturas internas é feita na seção 2.3, enquanto a seção 2.4 aborda a visualização de múltiplos volumes de dados. Parte do conteúdo deste capítulo estará sendo publicado brevemente [MAN 2002a].

2.1 Introdução

A visualização volumétrica, que emergiu na década de 1990, denota o conjunto de técnicas de Computação Gráfica interativa relacionadas com a extração de informações de conjuntos de dados 3D complexos [KAU 93]. Estas técnicas fornecem mecanismos para exibir e explorar o interior dos dados volumétricos, possibilitando a identificação de regiões e estruturas internas. As informações normalmente são armazenadas em uma grade regular, mas grades retilíneas, curvilíneas, irregulares [ELV 92] e malhas triangulares [SCH 2001] e tetraédricas [KIN 2001a] também são utilizadas.

Um grande número de trabalhos sobre visualização volumétrica é encontrado na literatura [KAU 91, LIC 98, PRO 98, MÜE 2001]. Aos algoritmos clássicos propostos no início da década de 90 (*ray casting* [LEV 88], *splatting* [WES 89, WES 90] e *shear-warp* [LAC 94]), somaram-se outros, tais como para visualização de dados multimodais [ZUI 96, STO 98] e o *confocal volume rendering* [MUL 2000], comprovando a importância desta área de pesquisa.

Também é objeto de estudo o desenvolvimento de novos algoritmos que possibilitem a interação em tempo real, incluindo a utilização de placas aceleradoras gráficas [HAS 98, REZ 2000, ENG 2001], arquiteturas paralelas ou *hardware* especializado [KAU 90, RAY 99], e a definição de critérios para avaliar e comparar a qualidade das imagens geradas por diferentes técnicas [WIL 96, SAN 96, KIM 2001, MEI 2000].

Na literatura vários termos e expressões são utilizados para caracterizar as diferentes classes de técnicas de visualização de volumes. Por exemplo, Elvins [ELV 92] considera duas categorias de algoritmos, *direct volume rendering* e *surface-fitting*, enquanto Müller et al. [MÜE 99] reconhece os mesmos dois grupos como *direct volume rendering* e *indirect volume rendering*. Entretanto, apesar das diferentes nomenclaturas, as duas categorias de técnicas de visualização de volumes se traduzem nas que trabalham com a extração de uma isosuperfície representada através de primitivas gráficas, e nas que trabalham gerando a imagem diretamente a partir do volume.

Neste trabalho serão utilizados os termos definidos por Kaufman, que classifica as técnicas de visualização de volumes em dois grandes grupos: ***surface rendering*** (ou visualização através de superfícies) e ***volume rendering*** (traduzido aqui para visualização direta de volume) [KAU 91]. Este autor ainda coloca que a visualização direta de volumes pode ser realizada de três maneiras diferentes [KAU 91, KAU 2000]: *object-order* (ou *forward-projection*), que envolve o

mapeamento de amostras de dados no plano da imagem; *image-order* (ou *backward-projection*), que determina para cada pixel do plano da imagem quais são as amostras que contribuem no cálculo da sua intensidade; e *domain-based*, quando os dados 3D são transformados para outro domínio, como frequência ou *wavelet*.

Técnicas de visualização através de superfícies envolvem a extração e a representação de uma isosuperfície que é posteriormente visualizada através da utilização de técnicas convencionais da Computação Gráfica, como ilustra a figura 2.1. A reconstrução da isosuperfície pode ser feita a partir de contornos planares, como mostra o trabalho de Weinstein [WEI 2000]. Neste caso, considerando um volume de dados de CT ou MRI, a estrutura de interesse deve ser identificada (ou segmentada) em cada uma das fatias para posterior composição da malha de polígonos. A isosuperfície também pode ser definida através de um limiar (*threshold*) [ELV 92] ou através de uma malha de polígonos extraída diretamente de um volume 3D previamente segmentado. Durante a etapa de visualização o usuário fornece parâmetros que estabelecem o tipo e a direção de projeção e os parâmetros de iluminação (figura 2.1).

Entre os algoritmos de visualização através de superfícies destacam-se: conexão de contornos [FUC 77, KEP 75] e cubos marchantes [LOR 87]. Alguns dos problemas que devem ser tratados nestes algoritmos são: a geração ocasional de pedaços de superfícies falsos; a manipulação incorreta de pequenas características dos dados; e a dificuldade de representação da superfície de algumas estruturas do corpo humano, no caso da visualização de imagens médicas. As grandes vantagens desta técnica, entretanto, são a velocidade para a geração e exibição da imagem final e o pouco espaço de armazenamento requerido. Representações deste tipo são apropriadas quando existem isosuperfícies bem definidas nos dados, mas não são eficientes quando o volume é composto por muitas micro-estruturas, como ocorre em várias imagens médicas [MÜE 99].

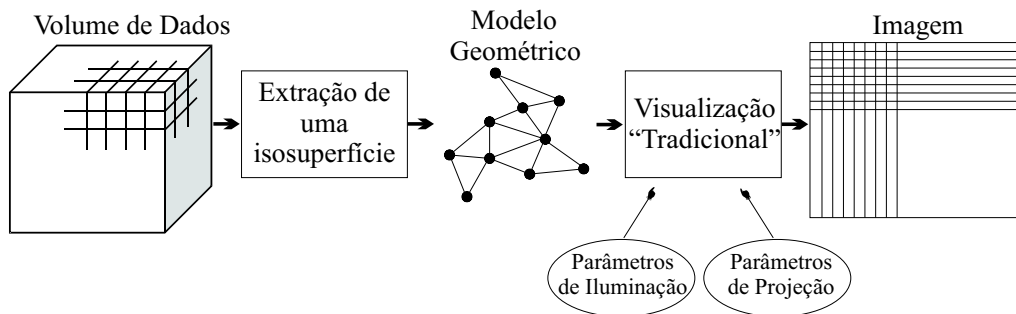


FIGURA 2.1 – Etapas da visualização através de superfícies

A segunda classe, visualização direta de volume, consiste em representar o volume através de voxels 3D que são projetados diretamente em pixels 2D e armazenados como uma imagem, dispensando o uso de primitivas geométricas (figura 2.2). Neste caso, numa etapa de classificação, é usada uma função de transferência, que corresponde ao mapeamento dos valores dos voxels (densidade do tecido, por exemplo) para propriedades visuais, tais como cor e opacidade. A visualização das estruturas de interesse dentro do volume é realizada a partir da “visita” a todos os voxels (ou quase todos, dependendo do algoritmo) e da aplicação da função de transferência para a construção da imagem.

Estas técnicas possuem um alto custo computacional, pois normalmente envolvem um tipo de interpolação, implícita ou explícita, nos pontos ao longo da direção de visualização. Por outro lado, produzem imagens de excelente qualidade, uma vez que todos os voxels podem ser usados na síntese das imagens, possibilitando a visualização do interior dos objetos. O algoritmo mais popular deste grupo é o *ray casting* [LEV 88, LEV 90], que como foi utilizado no desenvolvimento deste trabalho, está descrito em detalhe na seção 2.2. Outros algoritmos dessa categoria são *splatting* [WES 89, WES 90], *shear-warp* [LAC 94], *shell rendering* [UDU 93], *cell-projection* [WIL 91] e *V-Buffer* [UPS 88].

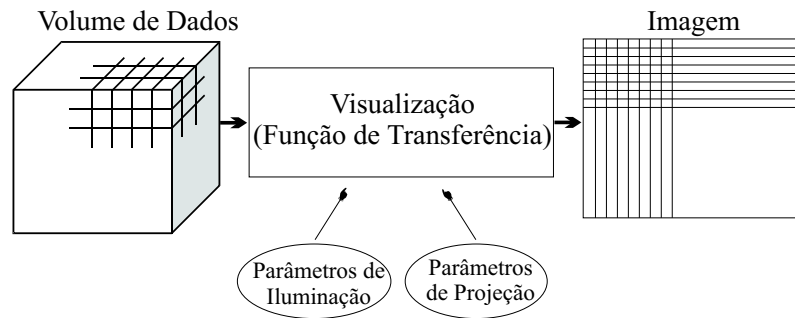


FIGURA 2.2 – Etapas da visualização direta de volume

Alguns autores [MÜE 99, MEI 2000a] fazem uma distinção entre os algoritmos de visualização direta de volume de acordo com a natureza do valor interpolado, que é dependente da ordem das etapas de visualização. Neste caso, quando os voxels do volume de dados são classificados e iluminados como um passo de pré-processamento e depois é realizada a interpolação nos pontos ao longo da direção de visualização, o modo de visualização é conhecido por *pre-shaded volume rendering*. O processo no qual os valores originais dos voxels são interpolados para depois ser feita a classificação e iluminação é denominado *post-shaded volume rendering*.

Desde a proposição dos primeiros algoritmos, uma das grandes dificuldades no desenvolvimento e utilização de sistemas de visualização interativa de dados volumétricos, principalmente quando se trabalha com técnicas de visualização direta de volume, é a quantidade de dados envolvida em relação às limitações na capacidade de armazenamento e processamento dos computadores. Apesar do desenvolvimento de algoritmos otimizados, apenas o avanço das tecnologias de *hardware* nos últimos anos possibilitou uma grande expansão na utilização destas aplicações. Neste caso, três estratégias diferentes podem ser adotadas:

1. Utilização de computadores com arquitetura paralela, ou execução dos sistemas em vários computadores ligados em rede, forçando a implementação do programa de forma distribuída [ZUI 96];
2. Utilização de placas aceleradoras gráficas e, para o seu melhor aproveitamento, desenvolvimento de novas técnicas de *rendering* para tirar vantagem do mapeamento de textura por *hardware*, incluindo a utilização de multi-texturas [HAS 98, WES 98, MEI 99, REZ 2000, ENG 2001];
3. Utilização de *hardware* dedicado para a visualização de volumes, como por exemplo o *EM-Cube* e o *VolumePro*; o primeiro é um exemplo de arquitetura

VLSI para visualização de volumes com baixo custo e alta qualidade [OSB 97]; a *VolumePro* consiste no primeiro chip para computadores pessoais que permite fazer o *rendering* de um volume em tempo real [PFI 99].

Kaufman et al. [KAU 90] e Ray et al. [RAY 99] abordaram estes tópicos em detalhe.

2.2 Algoritmo de *Ray Casting*

A idéia central do algoritmo de *ray casting* foi inicialmente proposta por Appel [APP 68], com o objetivo de detectar superfícies visíveis através do lançamento de raios de luz imaginários a partir da posição do observador em direção à cena. *Ray casting* é freqüentemente usado como sinônimo de *ray tracing*, mas também é usado como referência somente para o algoritmo de remoção de elementos ocultos [FOL 90].

O algoritmo de *ray casting* é um dos mais utilizados para visualização direta de volume, principalmente para a apresentação de dados médicos, pois gera imagens com grande qualidade que podem mostrar o interior do volume de dados. Apesar de ter um alto custo computacional, a geração de imagens em tempo real tornou-se viável com a evolução do *hardware* e por ser fácil de paralelizar. Este algoritmo foi inicialmente apresentado para a visualização de volumes por Levoy [LEV 88, LEV 90] e, posteriormente, vários autores desenvolveram extensões para melhorar o seu desempenho e incluir técnicas de interação [ZUI 95, NOO 97, KRE 98].

Resumidamente, o funcionamento do *ray casting* consiste na varredura dos pixels da janela de visualização e no lançamento de raios, a partir desses pixels, na direção do volume de dados. O plano de projeção, que é equivalente à janela de visualização, é determinado pela posição e orientação do observador. A idéia é simular a trajetória inversa dos raios de luz, partindo do observador em direção à cena. Sendo assim, os principais elementos são o volume de dados e suas propriedades, os parâmetros de visualização, a janela de visualização e os raios ao longo dos quais são processados os pontos de amostragem.

O volume de dados $V(\mathbf{x})$, que corresponde ao objeto da cena, refere-se a valores \mathbf{v} associados a posições de amostragem discretas $\mathbf{x} = [x, y, z]$, em geral regularmente espaçadas no domínio 3D, sendo que $x \in [1, N_x]$, $y \in [1, N_y]$ e $z \in [1, N_z]$. A existência de apenas um ou vários valores \mathbf{v} associados a uma posição de amostragem depende das propriedades dos dados que o volume (ou volumes) $V(\mathbf{x})$ representa. Por exemplo, no caso de um volume de dados de CT, há apenas um valor de densidade \mathbf{v} associado a uma posição \mathbf{x} . Quando se está trabalhando com visualização de dados multimodais, pode haver dois ou três valores representados por \mathbf{v} , relativos a cada volume de dados, tais como CT, MRI e PET. No caso de dados meteorológicos, é comum haver mais de um valor escalar associado à mesma posição \mathbf{x} , tais como temperatura, pressão e umidade relativa do ar.

Os parâmetros de visualização tradicionalmente especificam a posição do observador, o tipo de projeção (paralela ou perspectiva), a direção da projeção, o plano de projeção e os planos de corte 3D [FOL 90]. No contexto da visualização direta de volume, a direção de projeção \overrightarrow{DOP} , dada por um vetor posicionado no espaço do volume de dados, determina a orientação dos raios lançados no volume. O plano de projeção é mapeado para uma janela de visualização I no plano da imagem, a qual corresponde aos valores de intensidade associados às posições dos pixels $\mathbf{u} = [u, v]$, com $u \in [1, N_u]$ e $v \in [1, N_v]$ (figura 2.3).

A intensidade para cada pixel da imagem $I(\mathbf{u})$ é determinada pelo lançamento de um raio \mathcal{R}_p deste pixel ao longo da direção de visualização \overrightarrow{DOP} , conforme mostra a figura 2.3. Ao longo de cada raio são processadas amostras s , em geral com um espaçamento constante. O processamento de uma amostra corresponde ao cálculo de um valor de cor e um valor de opacidade dependentes do(s) valor(es) \mathbf{v} da amostra. Caso a posição da amostra não coincida com as posições discretas \mathbf{x} do volume, o cálculo é feito a partir da cor e opacidade dos seus voxels vizinhos. A cor e opacidade das amostras ao longo do raio são acumuladas para obtenção da cor do pixel. Os parâmetros de iluminação, tais como direção e tipo da fonte de luz, também influenciam o processamento das amostras.

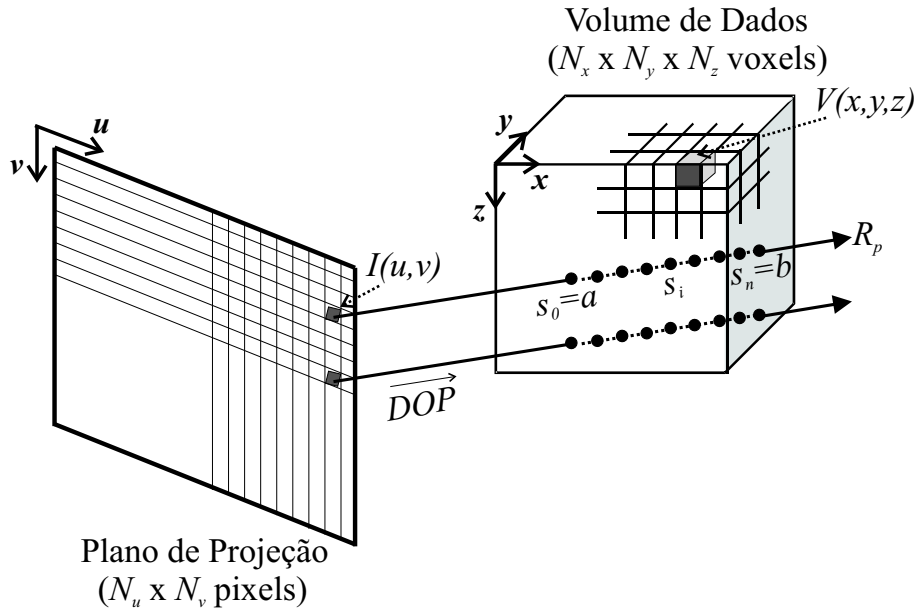


FIGURA 2.3 – Esquema genérico do algoritmo de *ray casting* (*front-to-back*)

Para facilitar, neste trabalho foi utilizada uma notação paramétrica para representar o raio. Sendo assim, s_i denota o i -ésimo ponto de amostragem usado para navegação ao longo de \mathcal{R}_p , que vai do pixel $I(\mathbf{u})$ em direção ao volume de dados, com $i \in [0, n]$, e sendo $n + 1$ o número total de amostras. Portanto, como $\mathcal{R}_p(s_i) = [x_{s_i}, y_{s_i}, z_{s_i}]$, o ponto de amostragem corresponde à distância do observador ao longo do raio (profundidade).

A obtenção do valor $V(\mathbf{x}_{s_i})$ pode requerer uma interpolação do volume de dados discreto. Por exemplo, caso a amostra a ser processada esteja entre duas posições discretas \mathbf{x}_1 e \mathbf{x}_2 , $V(\mathbf{x}_{s_i})$ é calculado considerando os valores dos voxels vizinhos $V(\mathbf{x}_1)$ e $V(\mathbf{x}_2)$. Interpolação do vizinho mais próximo e trilinear são exemplos de métodos de interpolação utilizados no cálculo das amostras [LIC 98].

As propriedades físicas de cada ponto amostrado s_i ao longo do raio \mathcal{R}_p são consideradas para calcular a intensidade do pixel $I(\mathbf{u})$, de tal maneira que:

$$I(\mathbf{u}) = g(\mathcal{R}_p) \quad (2.1)$$

onde g é uma função que determina a cor de um pixel em função das amostras processadas ao longo do raio \mathcal{R}_p que intercepta o volume V . Usualmente, a função g é implementada da seguinte maneira:

$$g(\mathcal{R}_p) = \sum_{s_i=a}^b f(\mathbf{x}_{s_i}), \quad (2.2)$$

onde $s_0 = a$ é a primeira amostra (frontal) que contribui para $I(\mathbf{u})$, $s_n = b$ é a última amostra (traseira), e $f(\mathbf{x}_{s_i})$ é uma função que determina a contribuição de cada ponto amostrado na intensidade final do pixel, sendo que cada um é processado de acordo com as propriedades visuais do voxel $V(\mathbf{x}_{s_i})$.

Na figura 2.3 e na equação 2.2 considerou-se que o processamento do algoritmo de *ray casting* é realizado na ordem *front-to-back*, isto é, as amostras são processadas a partir do primeiro ponto de intersecção a do raio com o volume até o segundo ponto de intersecção b . Entretanto, também é possível realizar o processamento na ordem *back-to-front*, caso em que as amostras são processadas na ordem inversa, de b para a . Os resultados gerados são iguais, a diferença está apenas na implementação da função g .

Tanto a função g como a função f podem ser implementadas de várias maneiras e com diferentes argumentos. Um exemplo de uma função f é apresentado no algoritmo *spectral volume rendering* desenvolvido por Noordmans et al. [NOO 2000] (seção 5.3.3). Nesta abordagem, além das bandas $RGB\alpha$, outras bandas espectrais e propriedades de iluminação são consideradas nas funções de transferência.

Na abordagem $RGB\alpha$ introduzida por Levoy [LEV 88] e adotada por outros autores [DRE 88, LIC 98, PFI 99], f é uma função que faz o mapeamento dos valores dos voxels em cor e opacidade, que são, então, combinados com um modelo de iluminação simples. A formulação apresentada por Levoy, com processamento *back-to-front*, é rescrita a seguir de acordo com a notação adotada neste trabalho:

$$g(\mathcal{R}_p) = \sum_{i=0}^K [T_{C_\lambda}(\mathbf{x}_{s_i}) T_\alpha(\mathbf{x}_{s_i}) \prod_{j=i+1}^K (1 - T_\alpha(\mathbf{x}_{s_j}))] \quad (2.3)$$

onde K é o número total de amostras processadas, sendo $i = 0$ a amostra mais distante do observador (*back-to-front*); $T_{C_\lambda}(\mathbf{x}_{s_i})$ representa a função de transferência que determina a cor da amostra s_i de acordo com $V(\mathbf{x}_{s_i})$ para uma determinada banda espectral, no caso RGB; $T_\alpha(\mathbf{x}_{s_i})$ representa a função de transparência que determina o valor de opacidade associado à amostra s_i . T_{C_λ} é a cor de fundo e T_α é 1 quando $i = 0$. Na prática, este processamento é feito para R, G e B separadamente.

Posteriormente, Levoy também apresentou o cálculo dos valores de cor e opacidade de forma recursiva usando a composição *front-to-back* [LEV 90]. Neste caso, a função g seria implementada como:

$$g(\mathcal{R}_p) = C(\mathbf{x}_{s_n}) / \alpha(\mathbf{x}_{s_n}) \quad (2.4)$$

com

$$C(\mathbf{x}_{s_i}) = \begin{cases} T_{C_\lambda}(\mathbf{x}_{s_i}) T_\alpha(\mathbf{x}_{s_i}), & i = 0 \\ C(\mathbf{x}_{s_{i-1}}) + T_{C_\lambda}(\mathbf{x}_{s_i}) T_\alpha(\mathbf{x}_{s_i}) (1 - \alpha(\mathbf{x}_{s_{i-1}})), & i > 0 \end{cases} \quad (2.5)$$

$$\alpha(\mathbf{x}_{s_i}) = \begin{cases} T_\alpha(\mathbf{x}_{s_i}), & i = 0 \\ \alpha(\mathbf{x}_{s_{i-1}}) + T_\alpha(\mathbf{x}_{s_i}) (1 - \alpha(\mathbf{x}_{s_{i-1}})), & i > 0 \end{cases} \quad (2.6)$$

onde $C(\mathbf{x}_{s_i})$ e $\alpha(\mathbf{x}_{s_i})$ são, respectivamente, os valores de cor e opacidade do pixel acumulados até s_i , $T_{C_\lambda}(\mathbf{x}_{s_i})$ e $T_\alpha(\mathbf{x}_{s_i})$ são as funções de transferência que determinam cor e opacidade, e $C(\mathbf{x}_{s_n})$ e $\alpha(\mathbf{x}_{s_n})$ são a cor e a opacidade final do pixel.

Considerando que a equação 2.3, bem como as equações 2.5 e 2.6, pode ser vista como módulos do *pipeline* do algoritmo implementado por Levoy, os passos para o cálculo da cor de um pixel são apresentados na figura 2.4. Este *pipeline* começa com o volume de dados, que pode ter passado anteriormente por uma etapa de pré-processamento, por exemplo, para a aplicação de filtros que eliminem ruído. Estes dados são usados como entrada para o modelo de colorização e para a etapa de classificação [LEV 88]. A classificação permite a visualização das estruturas do interior do volume através da atribuição de uma nova propriedade visual para cada voxel, que, no caso, consiste na opacidade. A opacidade é a medida de transparência de cada voxel (um número entre 0 e 1 descreve o quanto de luz é absorvido pelo voxel) e seu valor é resultado de um mapeamento das propriedades do voxel, tais como sua intensidade e a magnitude do gradiente¹. Este mapeamento é feito através de uma função de transferência, que também pode ser usada para estabelecer a relação entre um valor do voxel, usualmente densidade, e uma cor [LIC 98]. A utilização de cores permite discriminar as várias estruturas presentes em um volume de dados. A linha tracejada que aparece na figura 2.4 representa a multiplicação da cor de amostra pela sua opacidade (equação 2.3) [WIT 98]. Ao longo de cada raio lançado, valores de cor e opacidade para cada amostra são processados e combinados, na ordem *front-to-back* ou *back-to-front*, para especificar a cor final do pixel [LEV 88].

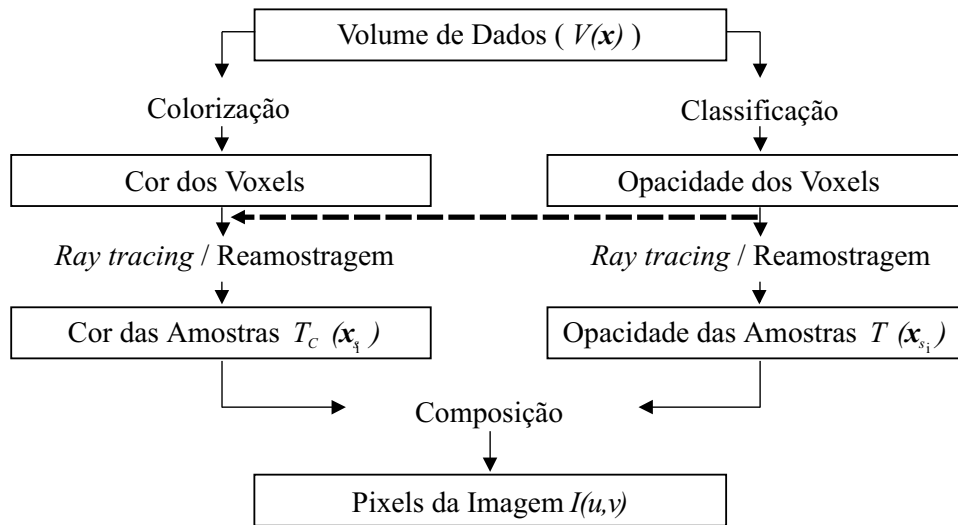


FIGURA 2.4 – *Pipeline* de *volume rendering* adaptado de Levoy [LEV 88, LEV 2000]

Apesar de muito utilizado, o algoritmo de *ray casting* possui um alto custo computacional devido ao grande número de cálculos necessários para determinar a cor do pixel. Por isso, uma série de otimizações foram propostas e são apresentadas na literatura, conforme descrito em uma publicação anterior [NED 99]. Algumas das mais populares, propostas pelo próprio Levoy, são:

- Pré-cálculo de dados intermediários [LEV 88], como por exemplo, a geração de um volume de cor e de um volume de opacidade a partir do volume de inten-

¹Aqui, o gradiente é um vetor tridimensional, equivalente ao vetor normal de uma superfície, que indica a direção e o sentido da variação das intensidades dos voxels [LIC 98]. A magnitude do gradiente indica a quantidade desta variação.

cidade original; desta maneira é possível diminuir o processamento realizado durante a execução do algoritmo;

- Utilização de *octrees* (enumeração espacial hierárquica) [LEV 90], para possibilitar um rápido avanço no cálculo de novas amostras, uma vez que permite “pular” regiões do volume onde não existem voxels de interesse;
- Terminação adaptativa do raio [LEV 90], que faz com que as amostras não sejam mais processadas quando a opacidade chega ao seu valor máximo (1);
- Refinamento progressivo (baixa qualidade da imagem, devido ao lançamento de um número menor de raios, durante o ajuste de parâmetros de rotação e translação) [LEV 90a], para possibilitar a interação com os dados;

2.3 Visualização de Estruturas Internas

Muitos trabalhos descritos na literatura apresentam maneiras de incorporar aos algoritmos de visualização direta de volume capacidade de visualizar estruturas internas, ou em outras palavras, “ver através” do volume. Dois tipos principais de abordagens são identificados nas técnicas existentes:

- Remoção dos voxels que estão fora da região de interesse;
- Controle da transparência do voxel durante a etapa de classificação.

A primeira abordagem, **remoção dos voxels**, pode ser tratada de maneira complementar a uma tarefa de seleção, na qual são selecionados apenas os voxels que estão dentro da região de interesse. A tarefa de remover (ou selecionar) voxels pode ser feita, basicamente, através de duas técnicas diferentes: segmentação e corte [YEN 96].

As técnicas de segmentação, que não serão aqui descritas por estarem fora do escopo deste trabalho, possuem como objetivo isolar um ou vários objetos de interesse, de maneira que eles possam ser exibidos como modelos geométricos. Por outro lado, a segmentação também pode ser usada para isolar e remover uma determinada estrutura do volume. Técnicas de segmentação manual podem ser usadas para extração de contornos em cada fatia de um volume de dados, com o objetivo de identificar os voxels que correspondem à ROI. A partir destes contornos, usando uma malha de triângulos, é gerada a superfície que deve ser visualizada [WEI 2000]. No método descrito por Bullitt e Aylward [BUL 2001] é feita uma combinação do algoritmo de *ray casting* com objetos tubulares segmentados, tais como vasos sanguíneos, que constituem as estruturas de interesse. Neste caso, após a etapa de segmentação, são processados apenas os raios que interseccionam o objeto. Já no trabalho apresentado por Kanitsar et al. [KAN 2001], uma ferramenta de segmentação interativa é usada para remover ossos de imagens de angiografia por tomografia computadorizada². Neste caso, usando uma composição do tipo MIP³, toda a estrutura dos vasos sanguíneos pode ser visualizada.

²Angiografia consiste na imagem dos vasos sanguíneos obtida pela injeção de material de contraste na corrente sanguínea [GRE 90].

³MIP (*Maximum Intensity Projection*) é uma operação de composição muito simples que consiste em escolher o maior valor escalar encontrado ao longo de um raio [LIC 98].

Técnicas de corte, que estão detalhadamente descritas no capítulo 3, preocupam-se com a remoção de partes do volume que o usuário **não** deseja ver, revelando a ROI e fornecendo uma melhor compreensão do conjunto de dados [LUC 92]. Por exemplo, um plano de corte pode ser usado para expor uma nova superfície e permitir a visualização dos objetos do seu interior. Inicialmente, este conceito foi introduzido como *volume slicing*, que consiste em visualizar o volume de dados como imagens 2D ortogonais aos três eixos principais ou paralelas ao plano de visualização [NIE 90]. Posteriormente houve uma evolução para ferramentas de corte mais flexíveis, que podem ser usadas para remover, ou até mesmo selecionar, blocos de voxels de várias maneiras, conforme será apresentado no capítulo 3.

Na segunda abordagem, **controle da transparência do voxel**, a idéia geral é manipular a visibilidade das estruturas através do controle das amostras que serão consideradas ao longo do raio, de forma que as estruturas relevantes para um determinado estudo sejam destacadas (ver capítulo 4). No trabalho pioneiro de Levoy [LEV 88], e em vários outros algoritmos de visualização direta de volume, tabelas de classificação são usadas para atribuir cor e opacidade a diferentes intervalos de valores de voxels. Tais tabelas representam as funções de transferência que fazem o mapeamento dos valores dos voxels para propriedades visuais. Assim, um valor de opacidade é especificado para cada estrutura identificada. Dependendo da função de transferência utilizada, se a transparência também estiver associada à magnitude do vetor gradiente [LIC 98], as superfícies existentes no volume são realçadas de tal maneira que as estruturas são facilmente distinguidas. Quando o volume de dados é previamente segmentado, uma máscara binária fica disponível para cada objeto, e, por exemplo, cor e opacidade podem ser atribuídas a cada um, assim como é feito por Noordmans et al. [NOO 2000]. A opacidade também pode estar associada à posição do voxel no volume, conforme exemplificado na seção 4.5, ou à profundidade da amostra ao longo do raio, como descrevem Mullick et al. [MUL 2000], no algoritmo denominado *Confocal Volume Rendering*.

2.4 Visualização de Múltiplos Volumes de Dados

O volume de dados normalmente corresponde a informações, tal como densidade, associadas a posições (frequentemente regulares) de um domínio de espaço 3D. No caso da visualização de imagens médicas, o volume é composto de seqüências de imagens 2D adquiridas através de uma ou mais técnicas de aquisição de imagens, como tomografia computadorizada e ressonância magnética. Estas imagens são então “empilhadas” para recriar a estrutura 3D, como exemplifica a figura 2.5.

É possível visualizar múltiplos volumes de dados simultaneamente, combinando de maneira adequada em uma única imagem os aspectos diferentes, complementares e/ou parcialmente sobrepostos existentes em cada volume. Uma área que se beneficia deste tipo de visualização é a medicina. Jean e Hodges [JEA 92] consideram que existem três categorias de atributos para se representar na visualização de imagens médicas: anatômicas, fisiológicas e temporais. A primeira refere-se às principais estruturas do corpo humano, que podem ser identificadas em imagens de CT ou MRI. Atributos fisiológicos são representados pelos dados de textura, e podem ser obtidos a partir da modalidade de aquisição de imagens médicas funcional. Atributos temporais registram as características dos dados que variam com o tempo,

tal como num exame de MRI cardíaco. Sendo assim, a integração de mais de um volume de dados (médicos) envolve:

- Imagens de uma mesma parte do corpo do paciente obtidas pelo mesmo aparelho em momentos diferentes, com o objetivo de mostrar a evolução ou não de uma lesão;
- Imagens de uma mesma parte do corpo do paciente obtidas através de diversas modalidades, tais como CT, MRI e PET, com o objetivo de obter informações complementares;
- Imagens de pacientes diferentes da mesma modalidade, com o objetivo de realizar estudos comparativos.

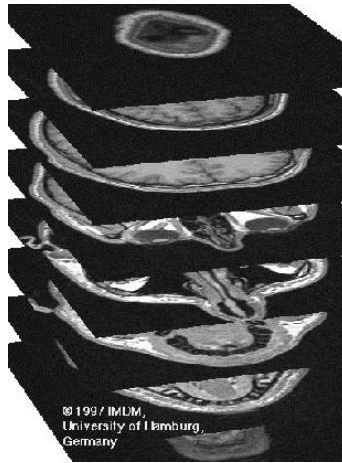


FIGURA 2.5 – Imagens 2D usadas para gerar a estrutura 3D (extraída de Schiemann e Höhne 2002 [SCH 2002])

Desta forma, é possível revelar informações adicionais que facilitem o diagnóstico, o que é uma vantagem em relação à interpretação realizada diretamente sobre um único volume de dados. No capítulo 5, são descritas algumas técnicas para a visualização de volumes de dados multimodais, que é o tipo de integração abordada neste trabalho. A ideia é combinar dados funcionais e anatómicos, que tem se mostrado como uma das maiores aplicações deste tipo de visualização. A visualização de estruturas internas em volumes de dados multimodais é apresentada no capítulo 6.

A principal diferença entre a visualização de um único ou de múltiplos volumes é a integração necessária no segundo caso, também chamada de “*data intermixing*” [CAI 99]. Conforme descrito na seção 2.2, quando se trabalha com visualização de múltiplos volumes, mono ou multimodais, pelo menos dois valores \mathbf{v}_1 e \mathbf{v}_2 são usados. Estes valores representam as propriedades de cada um dos volumes de dados associadas a uma posição \mathbf{x}_{s_i} . A função f usada no cálculo da intensidade do pixel $I(\mathbf{u})$ é implementada de forma diferente, uma vez que os cálculos de cor e opacidade são feitos de acordo com dois ou mais valores de intensidades, por exemplo, associados a $V_1(\mathbf{x}_{s_i})$ e $V_2(\mathbf{x}_{s_i})$.

Um exemplo de visualização integrada de volumes multimodais consiste em usar um volume como entrada para a função de transferência de opacidade (define

a anatomia), e o outro para a função de transferência de cor. Neste caso, a fórmula apresentada por Levoy (equação 2.3) poderia ser alterada da seguinte maneira:

$$g(\mathcal{R}_p) = \sum_{i=0}^K [T_{C_\lambda}(\mathbf{x}1_{s_i}) T_\alpha(\mathbf{x}2_{s_i}) \prod_{j=i+1}^K (1 - T_\alpha(\mathbf{x}2_{s_j}))] \quad (2.7)$$

onde, $\mathbf{x}1$ e $\mathbf{x}2$ correspondem aos dois volumes de dados multimodais.

2.5 Comentários Finais

Sistemas de visualização volumétrica são usados para criar imagens com grande qualidade visual e que facilitem o entendimento e a análise dos volumes de dados. A importância desta área está relacionada, principalmente, com o grande número de aplicações que se beneficiam dos seus resultados. Isto explica a extensa literatura encontrada e o seu constante crescimento. Além da existência de questões em aberto, principalmente no tocante a desempenho e fidelidade, o desenvolvimento de novas tecnologias de *hardware* cria outros desafios e aumenta o leque de possibilidades de pesquisa. Brodli e Wood [BRO 2001] recentemente fizeram uma revisão dos avanços na área de visualização volumétrica nos últimos anos.

Entre os algoritmos de visualização direta de volume, o algoritmo de *ray casting* ainda é um dos mais utilizados. Além de possibilitar uma exploração do volume de dados fornecendo imagens de alta qualidade a partir de uma implementação simples, um grande grau de paralelismo pode ser explorado neste algoritmo. Técnicas para sua otimização, bem como diferentes formas de interação e visualização do interior do volume de dados, estruturados ou não, ainda são objetos de pesquisa. Um exemplo é a grande preocupação com o projeto e utilização das funções de transferência [PFI 2001], fundamentais para a geração de imagens com qualidade e que mostrem o interior do volume.

Além das funções de transferência, técnicas de corte também podem ser usadas em conjunto ou como alternativa para as funções de transferência no processo de visualizar o interior do volume de dados. Apesar de ainda não terem sido completamente exploradas com esta abordagem, consistem em poderosas ferramentas que fornecem diferentes maneiras de investigar visualmente o volume de dados. A união de técnicas de visualização de estruturas internas com algoritmos de visualização de volumes multimodais resultaria em poderosas ferramentas que trariam inúmeros benefícios, principalmente para a área médica.

3 Ferramentas de Corte

Recortar (ou cortar) significa eliminar do processamento pixels ou primitivas gráficas que não estejam na janela de seleção ou região de interesse [FOL 90]. A seleção de uma seção oblíqua e a utilização de um plano ou volume de corte para eliminar uma parte dos voxels do processamento são alguns exemplos de ferramentas de corte utilizadas na área de visualização volumétrica. Na literatura, estas ferramentas tipicamente correspondem a elementos geométricos (como, por exemplo, um plano ou um cubo) usados para definir o subconjunto de voxels que será removido. De maneira complementar, usando estas mesmas ferramentas, é possível selecionar apenas o subconjunto de voxels de interesse para o processamento. Além disso, propriedades dos voxels também podem ser usadas para definir qual é a parte do volume de dados que será recortada ou selecionada.

Assim, do ponto de vista do usuário, tais ferramentas podem ser usadas para controlar a região de interesse de duas maneiras diferentes:

- Por exclusão, quando as ferramentas são usadas para especificar toda a região do volume que será retirada do processamento;
- Por inclusão, quando as ferramentas são usadas para especificar toda a região do volume na qual será feito o processamento.

Por exemplo, no caso de um volume de corte, ou as amostras são processadas nas posições do raio que estão fora do volume, ou são processadas somente nas posições do raio que estão no seu interior.

Quando técnicas de exploração de dados volumétricos são pesquisadas, dois termos são constantemente utilizados: corte e seleção. Entretanto, a diferença entre eles nem sempre é bem definida. Neste trabalho, assume-se que a operação de corte é equivalente ao processamento por exclusão, e a operação de seleção, ao contrário, é equivalente ao processamento por inclusão, pois somente as amostras da ROI são processadas (de acordo com o exemplo anterior, o interior do volume de corte).

Neste capítulo, é introduzida uma abordagem unificada para as técnicas de corte, por exclusão e por inclusão, que permitem visualizar o interior dos volumes de dados. As seções 3.2 e 3.3 apresentam diversas técnicas de corte, tanto aquelas onde o corte é feito explicitamente como implicitamente. Na seção 3.4 é feita uma análise sobre a utilização de técnicas de corte padrão como ferramentas de interação. Um resumo deste capítulo está incluído em outra publicação [MAN 2002a].

3.1 Abordagem Unificada para Ferramentas de Corte

Conforme o esquema geral apresentado na seção 2.2, um raio \mathcal{R}_p é lançado para cada pixel $I(\mathbf{u})$, e sua intensidade é determinada pelo processamento das amostras s_i ao longo do raio. De acordo com a equação 2.2,

$$g(\mathcal{R}_p) = \sum_{s_i=a}^b f(\mathbf{x}_{s_i}),$$

e considerando a ordem de processamento *front-to-back*, o intervalo de amostragem ao longo de \mathcal{R}_p é dado por $[a, b]$. Na ausência de ferramentas de corte, a e b

correspondem, respectivamente, aos dois pontos de intersecção de cada raio com o volume de dados (figura 3.1): r_1 , que indica a posição na qual o raio entra no volume de dados, e r_2 , que corresponde à posição na qual o raio sai do volume (deste modo, $a = r_1$ e $b = r_2$).

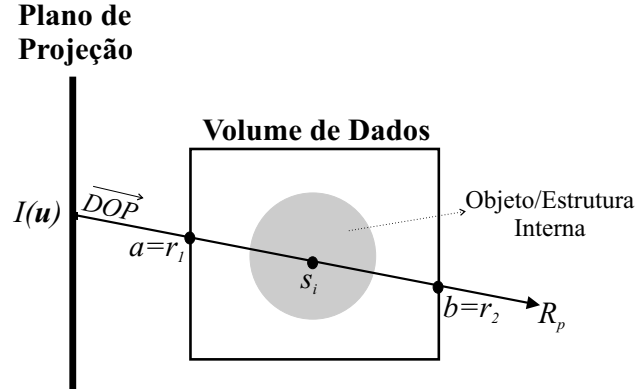


FIGURA 3.1 – Pontos de intersecção (r_1, r_2) do raio com o volume de dados (ver figura 2.3)

Quando diferentes ferramentas de corte estão disponíveis, o intervalo de amostragem pode ser determinado usando outras estratégias. Basicamente, a e b podem ser especificados de duas maneiras: a partir de **elementos geométricos**, como por exemplo dois planos de corte (figura 3.2); ou a partir de **propriedades dos voxels**, tais como valores de densidade. A figura 3.2 mostra a utilização de um plano de corte frontal e um traseiro, cuja intersecção com o raio é dada por c_1 e c_2 . Assim, o intervalo de amostragem é determinado por $a = c_1$ e $b = c_2$.

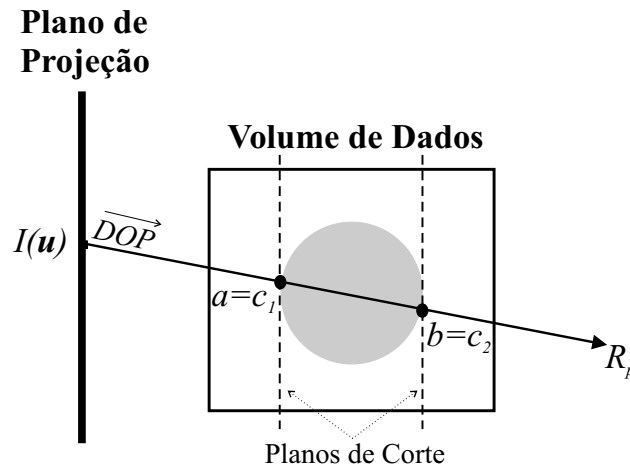


FIGURA 3.2 – Planos de corte (c_1, c_2) que delimitam o intervalo de amostragem em um raio (ver figura 2.3)

Seguindo esta linha de raciocínio, as ferramentas de corte podem ser generalizadas como duas funções, f_a e f_b , que são usadas para determinar os valores de a e b :

$$a = f_a(U_a, \mathcal{R}_p, V) \quad (3.1)$$

$$b = f_b(U_b, \mathcal{R}_p, V) \quad (3.2)$$

onde U_a e U_b correspondem a parâmetros fornecidos interativamente pelo usuário para configurar a ferramenta de corte, \mathcal{R}_p é o raio e V é o volume de dados.

Diferentes ferramentas de corte podem ser implementadas dependendo da escolha dos parâmetros U e das funções f_a e f_b . Os parâmetros podem ser um plano, superfície ou volume de corte; uma distância ou um valor de intensidade do voxel.

Partindo deste ponto de vista, conclui-se que as ferramentas de corte variam de acordo com:

- Tipo de operação (corte por exclusão ou por inclusão);
- Parâmetros de entrada U_a e U_b fornecidos pelo usuário e usados para especificar a ROI;
- Funções f_a e f_b usadas para calcular a e b , que podem receber informações de geometria ou de propriedades do voxel (conteúdo).

Considerando este esquema genérico, é feita uma discussão de como as ferramentas de corte mais utilizadas descritas na literatura implementam U_a , U_b , f_a e f_b : se consideram apenas os elementos geométricos (seção 3.2), se consideram o conteúdo do volume de dados (seção 3.3), ou se consideram tanto os elementos geométricos, como o conteúdo.

Para exemplificar as ferramentas de corte descritas neste capítulo foram utilizados dois volumes de dados sintetizados. Estes volumes, cujo conteúdo está detalhadamente descrito no anexo, consistem em esferas que estão ilustradas na figura 3.3. A esfera da figura 3.3a possui três camadas com densidades diferentes (80, 0 e 255) e um paralelepípedo de três voxels de espessura no seu interior (densidade=255). A esfera da figura 3.3b possui apenas uma camada externa e uma série de paralelepípedos no seu interior (densidades de 150 a 255). Para analisar as ferramentas de corte, com exceção da posição do observador, foram sempre utilizados os mesmos parâmetros de visualização. A função de transferência de opacidade utilizada é a linear (ver seção 4.2) a partir do valor de intensidade 40. Apenas os valores de a e b foram variados a cada execução para demonstrar as diferentes técnicas.

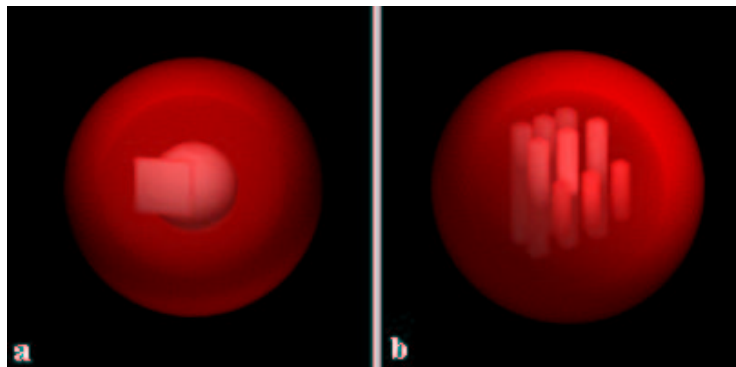


FIGURA 3.3 – Volumes sintetizados usados para ilustrar a descrição das técnicas de corte

3.2 Corte por Geometria

A exibição de volumes de dados como objetos 3D através de algoritmos de visualização volumétrica possui algumas limitações quanto à exploração do seu interior. Isto faz com que as ferramentas para extração de seções sejam muito utilizadas, principalmente no diagnóstico médico [MAC 95]. Estas ferramentas, também conhecidas como *slicing tools*, permitem a visualização de um corte 2D do volume (espessura = 1 voxel). Neste caso, como $a = b$, somente uma amostra s_i é levada em consideração no processamento de cada pixel $I(\mathbf{u})$. Assim, $f = f_a = f_b$ determina a localização da amostra através do cálculo de intersecção entre \mathcal{R}_p e um plano. Este plano corresponde à informação de geometria que é fornecida pelo usuário através da configuração de $U = U_a = U_b$.

Para métodos de corte com **seções planares**, ortogonais ou oblíquas, U consiste na definição da posição e orientação de um plano de corte. Em uma abordagem genérica, o usuário especifica a orientação do plano [ROB 98, UDU 2000] e f_a é responsável pelo cálculo do ponto de intersecção c_1 entre este plano e \mathcal{R}_p (figura 3.2). Por outro lado, se pode considerar que o plano de corte é sempre paralelo ao plano de projeção, de tal maneira que U é especificado como uma distância constante do plano de projeção. Neste caso, seções ortogonais e oblíquas diferem apenas na direção \overrightarrow{DOP} .

Quando se está trabalhando com um plano de corte, a diferença está na orientação do plano: para uma seção ortogonal, o plano é sempre ortogonal a um dos três eixos principais do sistema de referência do objeto, no caso, o volume de dados; para uma seção oblíqua, o plano é orientado arbitrariamente. Nielson e Hamann [NIE 90], Fonseca et al. [FON 97] e Robb e Barillot [ROB 89] exemplificam a utilização de seções ortogonais, inclusive para imagens médicas [ROB 89], e Rhodes et al. [RHO 80] trabalharam com seções oblíquas.

Para o corte com **seções não-planares**, como por exemplo uma seção curvilínea [ROB 89, ROB 98], também é necessário processar o ponto de intersecção entre \mathcal{R}_p e, neste caso, uma superfície. Esta superfície, especificada através de U , pode ser uma malha ou uma superfície paramétrica.

Sendo assim, conclui-se que a geração de seções ortogonais, oblíquas e não-planares tem um processamento simples, e consiste em um corte por inclusão, uma vez que o usuário sempre especifica o plano ou superfície onde será feita a amostragem. Os cortes axial e sagital ilustrados nas figuras 3.4a e 3.4b são exemplos de seções (ou planos) ortogonais. A figura 3.4c mostra um exemplo de uma seção oblíqua onde o observador está posicionado a frente e acima do volume, e a opacidade foi calculada através de uma função de transferência linear a partir do valor de intensidade 40 (equação 4.4). Uma aproximação da posição do observador está ilustrada na figura 3.4. A cor em tons de cinza foi obtida a partir dos valores de densidade armazenados no volume.

Apesar das ferramentas de corte para a extração de seções ortogonais e oblíquas serem as mais simples, são também as mais importantes, o que pode ser comprovado pela sua presença em praticamente todos os sistemas de visualização volumétrica. Para ilustrar a sua importância, Machiraju e Yagel [MAC 95] dedicaram-se ao desenvolvimento de um método apenas para caracterizar e medir o erro introduzido na reconstrução destas seções.

Um conjunto de ferramentas mais flexível, onde $a \neq b$, envolve o processamento

de várias amostras para cada \mathcal{R}_p , resultando em um subvolume. Neste caso, planos de corte adicionais são introduzidos no espaço do objeto (volume de dados) [TIE 96, LIC 98]. Uma primeira possibilidade consiste em especificar um **plano de corte frontal** para remover todos os voxels que estão entre o plano de projeção e este plano [WES 98]. Assim, U_a consiste na definição dos parâmetros do plano, da mesma maneira que descrito anteriormente, de tal forma que $f_a = c_1$ e $f_b = r_2$. De maneira análoga, o usuário também pode especificar um **plano de corte traseiro** através da configuração dos parâmetros do plano dada por U_b . Desta maneira $f_b = c_2$, isto é, todos os voxels localizados após este plano são também removidos.



FIGURA 3.4 – Exemplo de (a) um corte axial, (b) um corte sagital e (c) uma seção oblíqua determinada arbitrariamente, onde as linhas identificam os limites da seção

Esta classe de ferramenta de corte pode ser exemplificada pelo trabalho de Yen et al. [YEN 96], que apresentaram um método no qual o *rendering* é feito entre um conjunto de planos de corte paralelos, perpendiculares à direção de visualização. Neste caso, a porção do volume entre estes dois planos de corte, que pode ter uma espessura arbitrária, é chamada de *thin slab*, que é o nome atribuído ao método. Desta forma, é possível que pequenas estruturas sejam melhor representadas do que no caso de uma única fatia 2D. Similarmente, uma seqüência de seções paralelas adjacentes a uma seção oblíqua são usadas para gerar uma única imagem no trabalho apresentado por Robb et al. [ROB 89].

As ferramentas de corte descritas até aqui normalmente são exibidas junto com o volume de dados e são manipuladas interativamente pelo usuário [ROB 98]. Por isso, muitas vezes elas necessitam de um tratamento especial em algumas situações, como mostra a figura 3.5. Um primeiro problema ocorre quando o plano de corte é posicionado e regiões que não estão dentro do volume de dados são consideradas no processamento, uma vez que $f_a = c_1$ e $f_b = c_2$ (figura 3.5a). O segundo problema é quando os planos de corte frontal e traseiro são posicionados de tal maneira que ocorre uma intersecção entre eles (figura 3.5b). No primeiro caso, o cálculo das amostras deve ser feito apenas dentro do volume de dados (figura 3.5a). Sendo assim, $f_a = far(c_1, r_1)$ e $f_b = near(c_2, r_2)$, onde *far* é uma função que retorna o parâmetro que está mais afastado do plano de projeção, e *near* retorna o parâmetro que está mais próximo. No segundo caso, f_a e f_b também são iguais a $far(c_1, r_1)$ e $near(c_2, r_2)$, respectivamente, porém se $near(f_a, f_b) = f_b$, então não deve ser processada nenhuma amostra.

A utilização de planos de corte para a especificação de regiões onde o processamento será realizado (corte por inclusão) ou não (corte por exclusão) é exemplificada na figura 3.6. A figura 3.6a mostra a utilização de um plano de corte frontal ($f_a = c_1$

e $f_b = r_2$). Na figura 3.6b foram utilizados dois planos de corte para definir a ROI, de tal maneira que $f_a = c_1$ e $f_b = c_2$. A figura 3.6c é um exemplo da utilização de planos de corte como ferramentas de corte por inclusão, uma vez que são processadas apenas as amostras que estão exatamente sobre os planos. Desta maneira, diferentes planos que pertencem ao volume de dados podem ser exibidos, permitindo a visualização do interior do objeto.

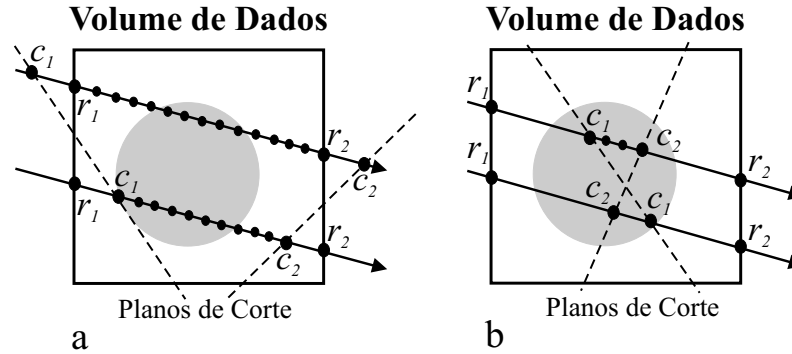


FIGURA 3.5 – Casos especiais na utilização de planos de corte

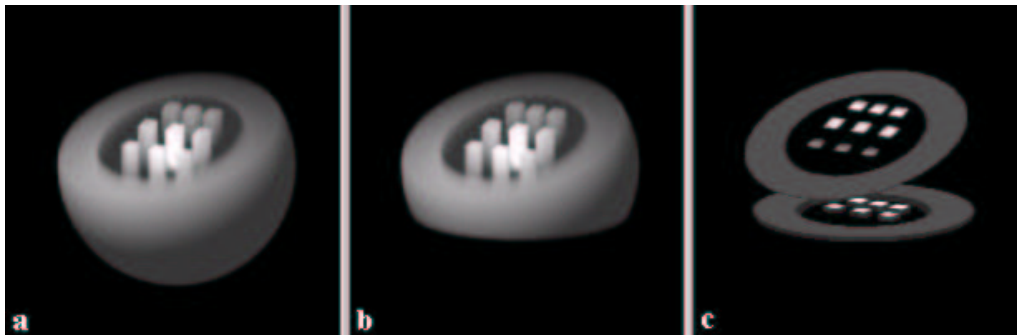


FIGURA 3.6 – Exemplo da utilização de (a) um ou (b,c) dois planos de corte

Uma alternativa para a utilização de planos de corte é o uso de **superfícies de corte não-planares**, exemplificado por Silva [SIL 2000]. A diferença em relação ao plano de corte está nos parâmetros U_a e U_b , que, neste caso, especificam objetos 3D mais complexos, e nas funções que calculam os pontos de intersecção c_1 e c_2 , que também são menos triviais do que no caso do cálculo de intersecção de um plano com uma reta (raio). A essência, entretanto, ainda está na idéia de que um subvolume é selecionado através do cálculo da intersecção entre ferramentas de corte frontal/traseira.

Ehricke et al. [EHR 93] apresentaram uma forma diferente de realizar um corte no volume através da *VisiCam*, uma ferramenta para exploração de dados que possui técnicas de visão computacional integradas. Neste caso, foi usada uma câmera com um plano frontal flexível com capacidade de se adaptar a uma superfície que esteja dentro do volume de dados. O *ray casting* é executado usando o plano frontal da câmera como um plano de corte frontal.

Um **objeto volumétrico convexo de corte** também pode ser usado para eliminar todos os voxels localizados na intersecção do mesmo com o volume, isto é,

que estão **dentro** do seu interior (corte por exclusão), como ilustra a figura 3.7. Este objeto pode ser um cubo, uma esfera ou pode ter uma geometria qualquer. Neste exemplo, U_a determina um paralelepípedo de corte. A função f_a , então, processa a intersecção entre este cubo e \mathcal{R}_p . Se não houver intersecção, $f_a = r_1$; se houver intersecção, $f_a = c_2$, que corresponde à posição na qual o raio deixa o cubo de corte. Nos dois casos $f_b = r_2$. No caso de um volume de dados estar posicionado no meio do volume de dados, um processamento especial é necessário para gerenciar dois intervalos de amostragem: $f_a = r_1$ e $f_b = c_1$, que corresponde às amostras que estão mais próximas do plano de projeção; $f_a = c_2$ e $f_b = r_2$, que correspondem às amostras que estão posicionadas depois do objeto de corte ao longo do raio, e conseqüentemente mais afastadas do plano de projeção.

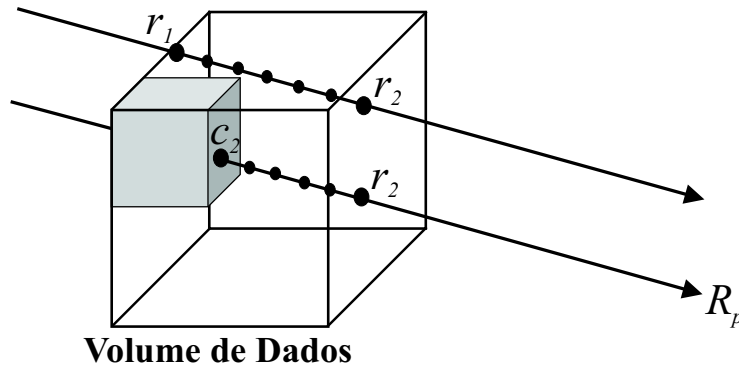


FIGURA 3.7 – Exemplo de um cubo usado para realizar o corte por exclusão

Um experimento usando um cubo como volume de corte é exibido na figura 3.8. Para gerar esta imagem, primeiro foram definidas as dimensões e a localização deste volume de corte. Depois, para cada raio, os pontos de intersecção são processados para permitir a especificação do parâmetro a .

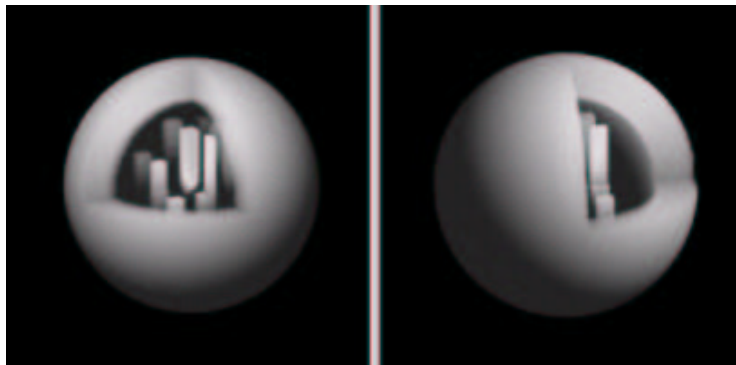


FIGURA 3.8 – Dois ângulos de visão do corte com um cubo

De maneira complementar, um objeto volumétrico pode ser usado para especificar o volume de interesse (ou VOI) através do corte de todos os voxels que estão **fora** do seu interior. Neste caso, quando se tem o corte por inclusão, somente as amostras dentro do objeto geométrico são consideradas no cálculo da cor final do pixel, conforme descrito por Fonseca et al. [FON 97]. As funções f_a e f_b equivalem, respectivamente, ao primeiro e segundo ponto de intersecção entre o volume de corte dado por $U_a = U_b$ e \mathcal{R}_p , de tal maneira que $f_a = c_1$ e $f_b = c_2$ (figura 3.9).

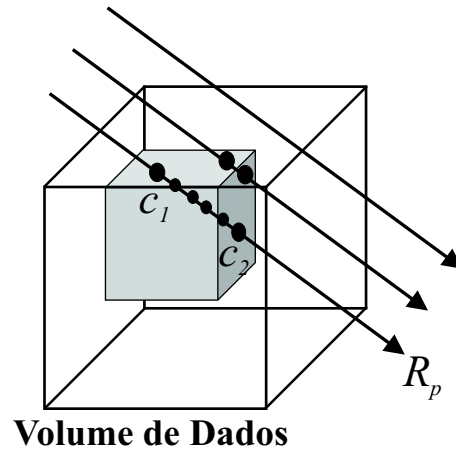


FIGURA 3.9 – Exemplo de um cubo usado para especificar um VOI em uma operação de corte por inclusão

A seleção de um subvolume é exemplificada na figura 3.10. Neste caso, um cubo é usado para selecionar a ROI, o que faz com que todos os voxels que estão fora do seu interior sejam eliminados.

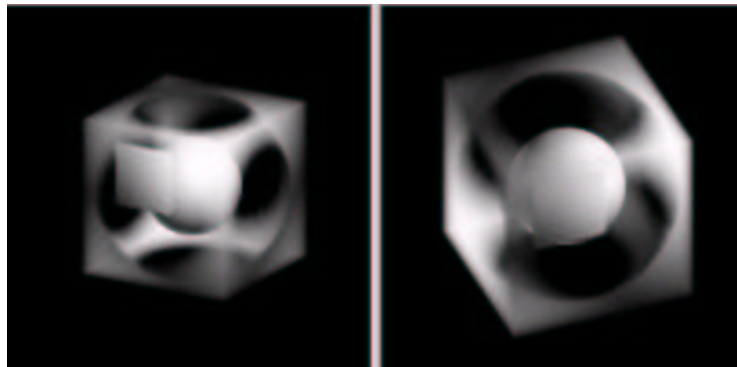


FIGURA 3.10 – Seleção de um subvolume com um cubo

Além de primitivas gráficas, tais como cubo e esfera, outros objetos volumétricos também podem ser usados para realizar o corte, sendo necessário apenas alterar o parâmetro U e o processamento dos pontos de intersecção. Para exemplificar, os trabalhos de Westermann e Ertl [WES 98], e de Hastreiter et al. [HAS 98a], que trabalham com a visualização direta de volume baseada em textura 3D, permitem a extração de um subvolume ou VOI através do uso de uma geometria arbitrária para o corte interativo.

É interessante comentar que a utilização de um volume de corte por exclusão é equivalente a uma operação de diferença realizada na técnica CSG (*Constructive Solid Geometry* [FOL 90]). Já a seleção de um VOI é equivalente a uma operação de intersecção, onde apenas a região em comum não é removida.

A idéia de usar planos de corte também pode ser empregada em abordagens diferentes, como por exemplo no algoritmo de *z-buffer* baseado em *rendering* por CSG, apresentado por Stewart et al. [STE 98]. Neste caso, o corte utilizando o *z-buffer* corresponde ao processo de armazenar uma superfície no *z-buffer* e detectar quais são os pixels que satisfazem as restrições das operações CSG de intersecção e

diferença. Lucas [LUC 92] também utiliza o *z-buffer* para fazer o corte durante o *rendering*. De forma semelhante à abordagem descrita [STE 98], o *z-buffer* pode ser usado para remover a parte posterior de uma cena, apenas com a sua inicialização com valores z que representam a superfície a partir da qual os elementos devem ser removidos. Porém, quando é feito o corte com um objeto arbitrário, é utilizado um *z-buffer* adicional para armazenar a superfície frontal, contra a qual também será feito o corte. Assim, os *buffers* frontal e traseiro armazenam os extremos do volume de corte em cada pixel [LUC 92]. Apesar da diferença entre realizar o *rendering* usando estas técnicas e a abordagem unificada aqui apresentada, resultados similares podem ser obtidos.

Simplificadamente, a técnica de *volume sculpting* consiste numa forma interativa de aplicar ferramentas de corte flexíveis. Sendo assim, entre as ferramentas apresentadas os volumes de corte aparecem como as mais adequadas. Por exemplo, no trabalho apresentado por Pflessner et al. [PFL 2000] estas ferramentas são representadas geometricamente ou através de polígonos, sendo fácil identificar quando as amostras estão ou não dentro de tais ferramentas. Já o trabalho desenvolvido por Baerentzen [BÆR 98] possuía duas maneiras de fazer a modelagem (ou *sculpting*): através de ferramentas CSG ou de *spray*.

Outra alternativa para o corte por geometria foi apresentada por Lorensen [LOR 93]: texturas booleanas é uma técnica de visualização que usa mapeamento de textura para recortar primitivas gráficas de acordo com a sua distância a superfícies implícitas. Uma textura booleana possui cor, intensidade e opacidade, que variam de acordo com a distância da superfície de corte, de tal maneira que cada ponto do modelo possui uma coordenada de textura correspondente. Por exemplo, no caso de um plano de corte, para cada ponto de um modelo geométrico sua distância ao plano é calculada e transformada para um valor de coordenada de textura válido, que é atribuído ao ponto. Esta coordenada indica se o ponto está antes, sobre ou depois do plano. O processamento consiste, então, em interpolar as coordenadas de textura, usá-las como uma tabela de cores e atribuir o resultado ao respectivo pixel. Esta abordagem é semelhante à utilização de planos ou volumes de corte, porém possui uma maior complexidade, uma vez que é necessário gerar e armazenar a textura booleana.

3.3 Corte por Geometria e Conteúdo

Em todas as técnicas de corte mencionadas na seção 3.2, U_a e U_b consistem em informações geométricas, tais como um plano ou um volume de corte. A técnica de *Confocal Volume Rendering* (CVR) apresentada por Mullick et al. [MUL 2000, SUM 2001], entretanto, introduz uma noção diferente, onde as propriedades do voxel são usadas em conjunto com a informação geométrica para especificar a e b . CVR é usado para controlar o “foco” numa região através de três parâmetros:

1. S-Band (*See-Through Band*): usado para indicar a região onde não haverá processamento, ou seja, para especificar as estruturas que serão removidas; consiste na definição de uma propriedade do voxel (tal como um intervalo de densidade) que caracteriza a superfície externa de uma estrutura, e de uma distância (por exemplo em mm) a partir desta superfície; a propriedade do voxel pode ser especificada pelo usuário ou automaticamente, e a distância,

que garante que o processamento irá começar apenas na ROI, é um parâmetro definido pelo usuário;

2. T-Band (*Transition Band*): indica exatamente o “quanto” da estrutura em termos de profundidade (por exemplo, em mm) o usuário deseja visualizar a partir do final da S-Band;
3. Função de Escala (*Depth Enhance*): configura a função de escala de opacidade dentro da porção visível do volume, permitindo que o usuário realce ou identifique “como” uma estrutura específica dentro da ROI será exibida; esta função é definida de acordo com o tipo de realce desejado.

A figura 3.11 ilustra como os parâmetros do CVR são usados para controlar o “foco” numa região. Neste caso, como mostra a linha pontilhada na figura 3.11, é feita uma combinação de aumento e declínio linear da função de escala.

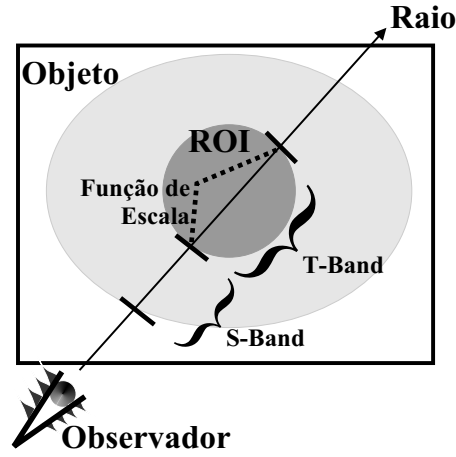


FIGURA 3.11 – Controle dos parâmetro no CVR [MUL 2000]

O primeiro parâmetro (S-Band) corresponde a U_a , que é usado para definir a de acordo com a propriedade do voxel $P(V(\mathbf{x}_{s_i}))$ e uma distância constante d_1 . Considerando que v é o valor procurado que representa a estrutura de interesse, no caso a superfície mais externa de um objeto, conclui-se que:

$$P(V(\mathbf{x}_{s_i})) = \begin{cases} Verdadeiro, & V(\mathbf{x}_{s_i}) = v \\ Falso, & V(\mathbf{x}_{s_i}) \neq v \end{cases} \quad (3.3)$$

Assim:

$$f_a = first(\mathcal{R}_p, v) + d_1 \quad (3.4)$$

onde $first$ é a função que denota a posição s_i procurada, ou seja, a primeira amostra s_i no raio \mathcal{R}_p na qual $P(V(\mathbf{x}_{s_i})) = Verdadeiro$.

O segundo parâmetro, T-Band, corresponde a U_b , que é usado para especificar b de acordo com a e uma distância constante d_2 da seguinte maneira:

$$f_b = a + d_2 \quad (3.5)$$

Observa-se, então, que CVR corresponde a uma técnica de corte por inclusão, sendo o intervalo de amostragem $[a, b]$ determinado por uma combinação das propriedades do voxel (em f_a) e da geometria (distâncias em f_a e f_b).

A figura 3.12 mostra exemplos da utilização desta técnica. As linhas I e II que aparecem nas vistas axial e sagital correspondem aos parâmetros S-Band e T-Band, respectivamente. Neste exemplo, estes parâmetros foram especificados de tal maneira que o “foco” muda em cada situação, isto é, na figura 3.12a a estrutura interna visualizada é diferente daquela mostrada na 3.12b. É possível observar que o tipo de imagem gerada pelo CVR é muito diferente das imagens geradas pelas técnicas de corte exemplificadas anteriormente, tais como seção oblíqua (figura 3.4c) e plano de corte (figura 3.6), pois a seleção do subconjunto de voxels que deve ser cortado é realizada de outra maneira.

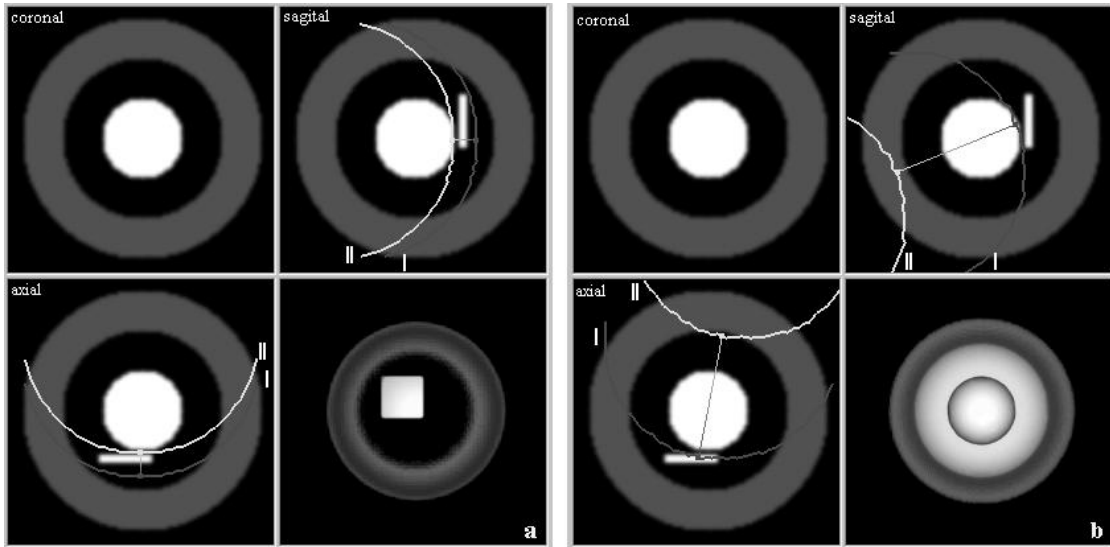


FIGURA 3.12 – Técnica de *Confocal Volume Rendering*. As imagens no canto inferior direito de (a) e (b) correspondem ao resultado da aplicação do CVR.

Uma técnica usada para exibir informações vasculares, apresentada por Zuiderveld [ZUI 95] e descrita na seção 6.1, também pode ser considerada uma técnica de corte. *Closest Vessel Projection* (CVP) consiste em terminar de processar as amostras quando o raio deixa o primeiro vaso sanguíneo encontrado. Esta técnica, que é usada como uma alternativa para a composição do tipo MIP, tem como objetivo aumentar a percepção de profundidade das imagens geradas. Por exemplo, as áreas escuras na figura 3.13, que ilustra o princípio desta técnica, representam altas intensidades:

- a) com MIP padrão, o menor vaso sanguíneo do lado esquerdo não é visível na imagem gerada, uma vez que o vaso maior do lado direito possui uma intensidade maior (figura 3.13a);
- b) com CVP, o *ray casting* termina quando o raio deixa o primeiro vaso sanguíneo que ele encontra (figura 3.13b); assim, o vaso menor fica visível na imagem resultante, pois, uma vez que a borda do vaso é mais escura, este é claramente delineado.

Desta maneira, na imagem final, alguns vasos parecem estar na frente de outros. Além de uma melhoria na qualidade da imagem, CVP também oferece um

melhor desempenho, uma vez que o processamento das amostras ao longo do raio pode ser terminado assim que o raio deixa o primeiro vaso encontrado.

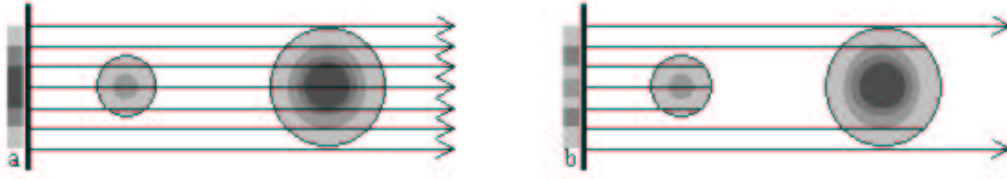


FIGURA 3.13 – Princípio do CVP (imagem adaptada de Zuiderveld [ZUI 95])

No CVP, f_a corresponde ao primeiro ponto de intersecção com o volume (r_1), e f_b é especificado de acordo com a propriedade do voxel $P(V(\mathbf{x}_{s_i}))$ para $s_i > a$, que, neste caso, é a densidade do limiar do vaso sanguíneo. Assim, b pode ser processado da seguinte maneira:

$$f_b = \text{second}(\mathcal{R}_p, v) \quad (3.6)$$

onde second é a função que denota a posição s_i procurada, ou seja, a segunda amostra s_i ao longo do raio \mathcal{R}_p na qual $P(V(\mathbf{x}_{s_i})) = \text{Verdadeiro}$, considerando que v representa o valor da superfície externa do vaso sanguíneo. Se o valor de $P(V(\mathbf{x}_{s_i}))$ não for encontrado, $f_b = r_2$. Esta técnica também pode ser considerada como um corte por inclusão.

A figura 3.14 mostra as imagens geradas com (3.14a) e sem (3.14b) a utilização da técnica de CVP. Os parâmetros de visualização utilizados para gerar as imagens foram os mesmos, com uma opacidade linear a partir do valor de intensidade 80, só que na figura 3.14a o processamento do raio termina quando o primeiro paralelepípedo é encontrado. No detalhe ampliado é possível verificar que as “bordas” dos objetos ficam melhor delineadas quando a técnica de CVP é utilizada, facilitando a percepção de quais objetos estão mais a frente.

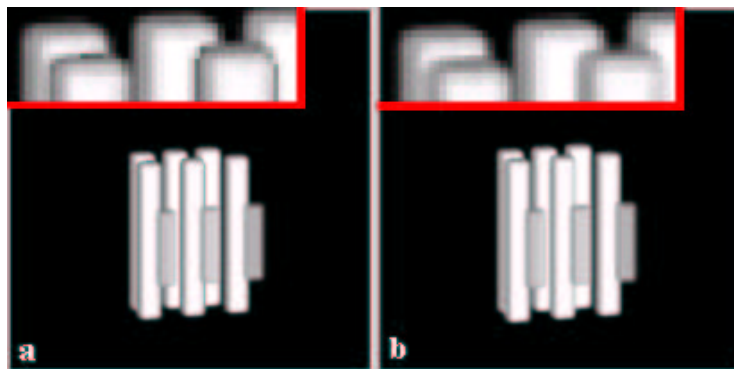


FIGURA 3.14 – Comparação entre a (a) técnica de *Closest Vessel Projection* e (b) o algoritmo de *ray casting* tradicional

Yoo et al. [YOO 91] também desenvolveram um trabalho voltado para a seleção de uma região a partir de critérios semânticos e não geométricos. Porém, neste caso, ao invés do resultado ser efetivamente uma ferramenta de corte, voxels da mesma natureza são agrupados apenas para definir uma ROI, para uma posterior etapa de segmentação.

3.4 Discussão

Considerando o que foi apresentado nas seções anteriores, é possível afirmar que as ferramentas de corte tradicionais podem ser expressas como diferentes implementações de f_a e f_b , sendo que os valores e a forma como são fornecidos os parâmetros U_a e U_b também variam. Estes parâmetros consistem em uma informação de geometria, tal como um volume de corte, ou em uma propriedade do voxel, tal como um valor de densidade. A tabela 3.1 mostra como diversas técnicas de corte se enquadram no esquema genérico apresentado (f_a e f_b), referenciando alguns trabalhos já desenvolvidos.

TABELA 3.1 – Determinação de a e b para cada técnica de corte (Equação 2.2)

Ferramenta de Corte	f_a	f_b	Exemplos
1. Seção planar (ortogonal ou oblíqua)	c_1	c_1	[RHO 80, GRA 96, ROB 89, NIE 90, MAC 95, FON 97, ROB 98, UDU 2000]
2. Seção não planar	c_1	c_1	[ROB 89, ROB 98]
3. Plano de corte frontal	c_1	r_2	[TIE 96, LIC 98, WES 98]
4. Planos de corte frontal e traseiro	c_1	c_2	[ROB 89, YEN 96, SIL 2000]
5. Superfície de corte não-planar frontal	c_1	r_2	[SIL 2000]
6. Superfícies de corte não-planares, frontal e traseira	c_1	c_2	[SIL 2000]
7. Objeto volumétrico convexo de corte	r_1	r_2	[SIL 2000]
	c_2	r_2	
	r_1	c_1	
8. Volume de interesse	c_1	c_2	[FON 97, WES 98, HAS 98a]
9. CVR	$first(\mathcal{R}_p, v) + d_1$	$a + d_2$	[MUL 2000]
10. CVP	r_1	r_2	[ZUI 95]
	r_1	$second(\mathcal{R}_p, v)$	

De acordo com os parâmetros de entrada U_a e U_b , as diferentes implementações das funções f_a e f_b , os resultados esperados, e os objetivos do usuário, é possível escolher a ferramenta de corte que melhor se adapta às necessidades. A tabela 3.2 mostra as possíveis combinações que podem ser feitas do ponto de vista do usuário, do tipo de operação e da especificação de a e b .

TABELA 3.2 – Diferentes combinações de acordo com a especificação de a e b . Os números 1 a 4 representam alternativas não exploradas para ferramentas de corte

a e b	Corte por Exclusão	Corte por Inclusão
$a \neq b$: geometria	Um ou dois planos de corte, superfícies de corte não-planares, objeto volumétrico convexo de corte	Dois planos de corte, volume de interesse, superfícies de corte não-planares
$a = b$: geometria	1	Seções ortogonais, oblíquas e não-planares
$a = b$: conteúdo	2	MIP
a : geometria b : conteúdo	3	CVP
a : geometria e conteúdo b : geometria	4	CVR

Diferentes imagens podem ser geradas usando as mesmas funções f_a e f_b e alterando o tipo de operação, corte por exclusão ou por inclusão. Por exemplo, as ferramentas de volume de corte podem ser usadas para **eliminar** tudo que estiver no seu interior (figura 3.8), ou para **selecionar** o que estiver no seu interior (VOI), como ilustra a figura 3.10. Da mesma maneira, quando se tem dois planos de corte (figura 3.6): as amostras que estão entre o primeiro plano e o observador e após o segundo plano são eliminadas (corte por exclusão), ou serão processadas apenas as amostras que estiverem sobre os planos, como se fossem duas seções oblíquas (corte por inclusão). As seções ortogonais, oblíquas e não-planares correspondem a corte por inclusão, uma vez que é necessário indicar somente a seção, ou fatia, que se deseja visualizar.

Na técnica CVP o usuário apenas especifica f_b (uma propriedade do voxel), uma vez que $f_a = r_1$. Desta maneira, se indica “o quanto” ou “até onde” se deseja visualizar. Já no algoritmo de CVR, a sempre depende da propriedade do voxel e de uma distância d_1 , e b é uma distância d_2 em relação ao parâmetro a . Todos os parâmetros fornecidos nestes casos são usados para especificar um VOI (corte por inclusão). CVP e CVR são exemplos de ferramentas de corte baseadas em conteúdo. Entretanto, outras combinações de parâmetros podem ser realizadas. Para ilustrar, CVR poderia ser alterado de tal maneira que as amostras fossem processadas entre $f_a = first(\mathcal{R}_p, v)$ e $f_b = r_2$, dado um valor v para o qual $P(V(\mathbf{x}_{s_i})) = Verdadeiro$. Para ilustrar a diferença neste caso, a figura 3.15 apresenta como seria a visualização dos volumes de teste considerando que o limiar de densidade v é 230. Assim, seriam consideradas apenas as amostras processadas a partir do primeiro valor de voxel igual a 230 encontrado ao longo do raio.

É possível afirmar que a técnica de MIP diferencia-se da visualização tradicional somente na etapa de composição do algoritmo de *ray casting* [LIC 98], apesar da etapa de colorização geralmente não ser considerada na geração deste tipo de imagem. Entretanto, alguns autores tratam MIP como uma técnica de *rendering* diferente [UDU 2000, HAU 2001]. Neste trabalho, MIP está sendo abordada de outra forma: uma vez que é considerada apenas a amostra de maior intensidade e as outras são eliminadas, MIP pode ser tratada como uma ferramenta de corte por

inclusão. Neste caso, chega-se à seguinte equação para MIP:

$$f_a = f_b = \max(\mathcal{R}_p) \quad (3.7)$$

onde \max é uma função que denota a posição s_i na qual a amostra em \mathcal{R}_p tem o maior valor $V(\mathbf{x}_{s_i})$.

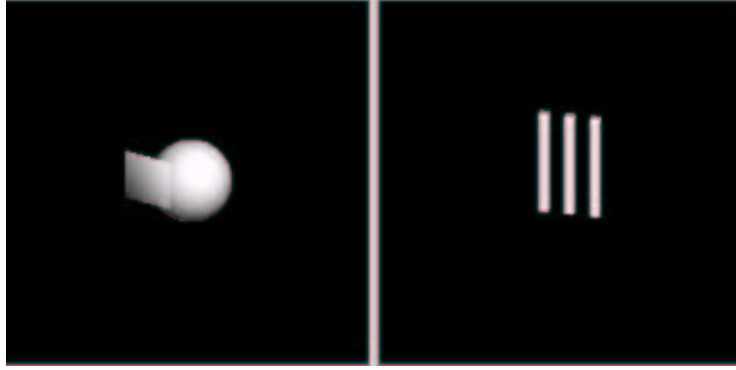


FIGURA 3.15 – Corte por inclusão (a=conteúdo, b=geometria) nos dois volumes de dados sintetizados

Analisando a tabela 3.2 também é possível observar que, teoricamente, existem situações não encontradas na literatura (1.4), as quais consistem em ferramentas ainda não exploradas, mas que podem ser facilmente identificadas dentro da abordagem apresentada. Por exemplo, nas situações 1 e 2, como $a = b$, a informação de geometria ou de propriedade do voxel poderia ser usada para eliminar uma determinada seção ou estrutura da visualização do volume de dados. Provavelmente, a imagem gerada na situação 1 não seria útil, uma vez que, por exemplo, apenas um plano seria eliminado do processamento. De maneira similar, apenas uma estrutura determinada por um *threshold* não é considerada no processamento na situação 2. A situação 3 corresponde ao complemento da CVP, onde, por exemplo, todas as amostras que estiverem entre $f_a = r_1$ e $f_b = \text{second}(\mathcal{R}_p, v)$ não são processadas. Da mesma forma, na situação 4 são eliminadas todas as amostras que estiverem entre $f_a = \text{first}(\mathcal{R}_p, v) + d_1$ e $f_b = a + d_2$.

Alternativas diferentes não incluídas na tabela 3.2 podem ser facilmente identificadas e implementadas. Por exemplo, $a \neq b$, ambos definidos de acordo com propriedades do voxel, corresponde a técnicas onde intervalos de valores (ou *thresholds*) são usados para especificar as amostras que devem ser cortadas ou selecionadas, dependendo da operação de corte. Assim, o usuário poderia determinar que só as amostras ao longo do raio que estivessem entre dois valores de intensidade seriam processadas. De maneira análoga, seria possível eliminar do processamento as amostras que estivessem entre estes dois valores. Por exemplo, no caso de imagens médicas, o valor de intensidade de cada voxel corresponde à densidade do tecido. Então, para cada raio $f_a = \text{first}(\mathcal{R}_p, v_a)$, com $v_a = 60$, e $f_b = a + \text{first}(\mathcal{R}_p, v_b)$, com $v_b = 200$, seriam processadas apenas as amostras que estivessem entre a pele e o osso, que possuem densidades próximas de 60 e 200, respectivamente. A figura 3.16 ilustra, no caso de um dos volumes de teste, como seria a visualização se o usuário especificasse que, em cada raio, são visualizadas somente as amostras cujos valores de intensidade estão entre 100 e 255. Outras combinações do processamento de f_a e f_b que se baseiam nas propriedades do voxel ainda podem ser consideradas. Uma

alternativa, por exemplo, é especificar a e b a partir de uma geometria e de uma propriedade do voxel.



FIGURA 3.16 – Visualização usando intervalo de valores para o corte por inclusão

3.5 Comentários Finais

A abordagem unificada introduzida neste capítulo possibilita tratar da mesma maneira as várias ferramentas de corte apresentadas isoladamente na literatura. O principal objetivo dessa abordagem e da análise aqui realizada foi facilitar o entendimento do funcionamento destas ferramentas e mostrar a possibilidade de desenvolvimento de novas ferramentas. Ferramentas de corte são muito utilizadas e na maioria das vezes são tratadas de forma simplificada, sem que suas potencialidades sejam devidamente exploradas. Por exemplo, na revisão do estado da arte em visualização volumétrica feita por Brodlie e Wood [BRO 2001], são descritas apenas as *slicing tools*, que resultam numa imagem 2D. Além disso, apesar das ferramentas de corte geralmente estarem associadas a elementos geométricos, a abordagem introduzida neste capítulo, onde propriedades dos voxels são usadas para definir a parte do volume de dados que será recortada, sem segmentação explícita, pode ser uma forma mais natural e intuitiva de interagir com os dados.

A partir desta abordagem, que une simplicidade e completude, técnicas de corte podem ser facilmente estendidas, facilitando a manipulação de parâmetros e a implementação dos sistemas, e abrindo um leque de alternativas para o desenvolvimento de novas ferramentas de exploração de volume de dados.

4 Funções de Transferência

As funções de transferência são responsáveis pela associação de propriedades visuais, tais como cor e opacidade, aos valores originais do volume de dados que está sendo visualizado [PFI 2001]. Uma descrição sobre o processo de especificação destas funções é apresentada na seção 4.1. As seções 4.2 e 4.3 descrevem, respectivamente, as funções de transferência de opacidade e de cor mais utilizadas. As diferentes abordagens para sua especificação são apresentadas na seção 4.4. Um resumo deste capítulo também está incluído em outra publicação [MAN 2002a].

4.1 Introdução

As funções de transferência permitem explorar as estruturas existentes nos volumes de dados sem a necessidade de definir explicitamente a sua forma ou extensão. Para alcançar este objetivo, geralmente, um valor de opacidade é associado a cada voxel para descrever quanta energia luminosa é absorvida, indicando se é possível ou não enxergar através desse voxel, e permitindo a definição de estruturas transparentes, semitransparentes e opacas [LIC 98].

A tarefa de especificar funções de transferência que gerem imagens de qualidade e que transmitam as informações requeridas não é trivial e tem sido amplamente discutida [PFI 2001]. Um problema, por exemplo, é que normalmente os valores de opacidade são definidos através da edição de gráficos que representam mapeamentos e que possuem pontos de controle alteráveis. O processo torna-se mais complexo ainda quando há interação com outros parâmetros, tais como fonte de luz e modelo de tonalização [KIN 98]. Desta maneira, encontrar uma função de transferência de cor ou opacidade tende a ser uma tarefa demorada e realizada através de “tentativa e erro”, pois pequenas alterações no gráfico podem causar drásticas mudanças nas imagens geradas. Por isso, devido à sua complexidade, recentemente houve um aumento na pesquisa de técnicas automáticas e semi-automáticas para a criação destas funções, e no desenvolvimento de interfaces interativas mais amigáveis para auxiliar nesta tarefa. Isto pode ser facilmente observado pelos vários trabalhos recentes encontrados na literatura [MAR 97, KIN 98, LIC 98, LIU 99, KNI 2001, KÖN 2001, PFI 2001, KIN 2001, KNI 2002, BOT 2002].

Segundo Lorensen [PFI 2001], técnicas automáticas ou que requerem muita interação com o usuário não são as mais adequadas. Quando há necessidade de uma grande interação, como por exemplo nos gráficos definidos através de “tentativa e erro”, a tarefa de encontrar uma função de transferência adequada pode levar muito tempo. Por outro lado, o usuário não deve ser afastado do processo de especificação destas funções, uma vez que o objetivo das técnicas de visualização é permitir explorar e entender o volume de dados, e não apenas gerar imagens bonitas.

Inicialmente, assume-se a existência de uma função de transferência genérica T , usada para atribuir propriedades visuais a um voxel. Esta função pode considerar ou ter como argumento uma ou mais propriedades do voxel (equação 4.1):

$$v_o = T(V(\mathbf{x}_{s_i}), \|\nabla V(\mathbf{x}_{s_i})\|, \nabla^2 V(\mathbf{x}_{s_i}), \dots) \quad (4.1)$$

onde v_o é a propriedade visual do voxel (cor ou opacidade), T é a função de transferência, $\|\nabla V(\mathbf{x}_{s_i})\|$ é a magnitude do gradiente no ponto de amostragem e $\nabla^2 V(\mathbf{x}_{s_i})$

é o Laplaciano, um operador da derivada de segunda ordem. Existem diversas implementações de T , e cada uma pode utilizar diferentes propriedades do voxel. Nas seções 4.2 e 4.3 são apresentadas algumas implementações de T já descritas na literatura, incluindo as técnicas e ferramentas utilizadas para especificação das funções. Para gerar as imagens que exemplificam as diferenças entre as diferentes funções também são usados os volumes de dados sintetizados descritos no anexo e ilustrados na figura 3.3. Nenhuma ferramenta de corte é, porém, utilizada.

Do ponto de vista de visualização de estruturas internas, a função T que mais influencia o resultado final é a função de transferência de opacidade. A função para atribuir cor é necessária porque a maioria dos volumes de dados não possuem valores de cor associados a cada voxel [LIC 98], e a utilização de cores pode ser muito útil, principalmente para distinguir e caracterizar estruturas. Emissão e índice de refração também são exemplos de propriedades visuais que podem ser processadas, mas não são tão utilizadas quanto cor e opacidade [KIN 2001].

4.2 Função de Transferência de Opacidade

As funções de transferência de opacidade (T_α) mais simples especificam a opacidade apenas de acordo com a intensidade do voxel, ou de acordo com a intensidade do voxel e a magnitude do gradiente local. Sendo assim, neste trabalho é usada a seguinte expressão genérica para representar uma função de transferência de opacidade:

$$\alpha(\mathbf{x}_{s_i}) = T_\alpha(\mathbf{x}_{s_i}) \quad (4.2)$$

onde o valor de opacidade da amostra $\mathbf{x}_{s_i} \in [0, 1]$ e T_α é a função de transferência que determina um valor de opacidade.

Atualmente existe um conjunto padrão de funções de transferência de opacidade que normalmente é fornecido e/ou utilizado nos sistemas. Entre estas funções destacam-se: rampa, trapézio, bloco, exponencial e linear, esta, a mais simples de todas. O mapeamento destas funções, que estão descritas a seguir, normalmente é realizado considerando apenas a intensidade do voxel.

Função linear. Quanto maior for a intensidade do voxel, maior é o valor de opacidade que ele recebe. T_α neste caso é calculada como (equação 4.3):

$$T_\alpha(\mathbf{x}_{s_i}) = V(\mathbf{x}_{s_i})/I_m \quad (4.3)$$

onde I_m é o maior valor de intensidade que um voxel do volume de dados pode ter. No caso de dados médicos, por exemplo, também é possível desprezar valores de intensidade muito pequenos, que geralmente correspondem a ruídos da imagem, atribuindo zero para a opacidade. Assim, a equação 4.3 seria alterada para:

$$T_\alpha(\mathbf{x}_{s_i}) = \begin{cases} 0, & V(\mathbf{x}_{s_i}) < I_l \\ (V(\mathbf{x}_{s_i}) - I_l)/(I_m - I_l), & V(\mathbf{x}_{s_i}) \geq I_l \end{cases} \quad (4.4)$$

onde, I_l é o valor de intensidade que corresponde ao limiar (ou *threshold*), responsável pela identificação dos voxels que devem ser considerados no processamento (figura 4.1). A figura 4.2 ilustra a diferença da aplicação de uma função de transferência linear onde são usadas as equações 4.3 (figura 4.2a) e 4.4 com a atribuição do valor 75 para I_l (figura 4.2b).

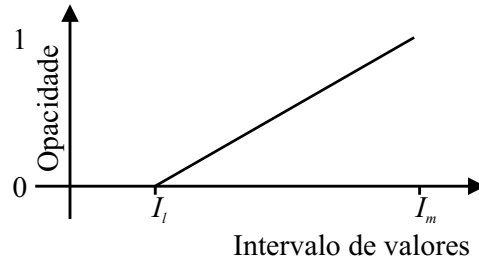


FIGURA 4.1 – Função de transferência linear

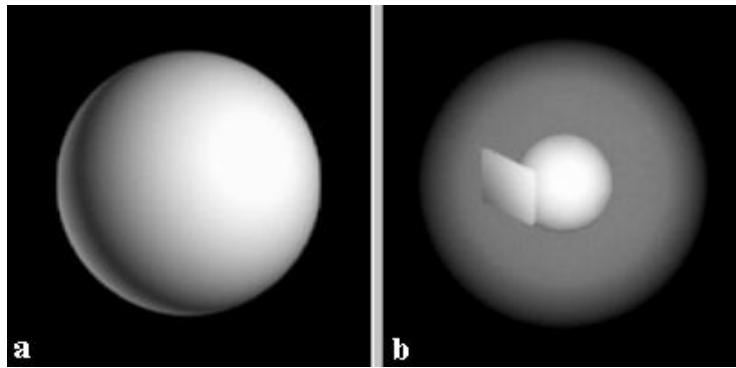


FIGURA 4.2 – Exemplos da aplicação de funções de transferência lineares

Função rampa. Esta variação da função linear, chamada de “rampa” por König e Gröller [KÖN 2001], ocorre quando o valor de intensidade do voxel que possui opacidade 1 não é o maior, mas sim o central, como mostra a figura 4.3. Assim, alterando a equação 4.3 obtém-se a equação 4.5 para a função rampa:

$$T_{\alpha}(\mathbf{x}_{s_i}) = \begin{cases} \frac{V(\mathbf{x}_{s_i})}{I_c}, & V(\mathbf{x}_{s_i}) \leq I_c \\ \frac{(I_m - V(\mathbf{x}_{s_i}))}{I_c}, & V(\mathbf{x}_{s_i}) > I_c \end{cases} \quad (4.5)$$

onde I_c é o valor de intensidade médio (entre 0 e I_m) que um voxel pode ter.

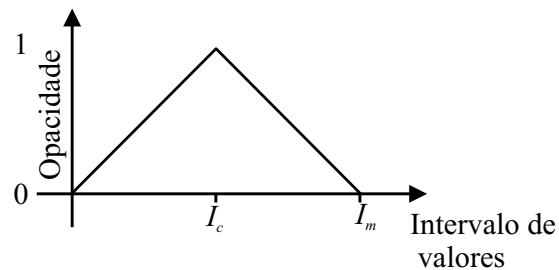


FIGURA 4.3 – Função tipo “rampa” corresponde a uma variação da função de transferência linear

Função trapézio. König e Gröller [KÖN 2001] mostram uma forma muito utilizada para T_{α} , na qual os valores são definidos de tal maneira que o formato do gráfico da função é um trapézio (figura 4.4). Neste caso, os valores dos voxels centrais são mapeados para opacidade máxima, enquanto os voxels dos extremos são mapeados

para valores crescentes ou decrescentes de opacidade, da seguinte maneira:

$$T_\alpha(\mathbf{x}_{s_i}) = \begin{cases} 0, & 0 \leq V(\mathbf{x}_{s_i}) \leq c - \frac{w}{2} \\ \frac{V(\mathbf{x}_{s_i}) - c - \frac{w}{2}}{d}, & c - \frac{w}{2} < V(\mathbf{x}_{s_i}) \leq c - \frac{w}{2} + d \\ 1, & c - \frac{w}{2} + d < V(\mathbf{x}_{s_i}) < c + \frac{w}{2} - d \\ 1 - \frac{V(\mathbf{x}_{s_i}) - c - \frac{w}{2} + d}{d}, & c + \frac{w}{2} - d < V(\mathbf{x}_{s_i}) \leq c + \frac{w}{2} \\ 0, & c + \frac{w}{2} < V(\mathbf{x}_{s_i}) \leq 1 \end{cases} \quad (4.6)$$

onde w corresponde ao intervalo de valores de voxels, c é o ponto central deste intervalo e d é a distância (ou intervalo de valores) na qual o valor de opacidade passa de 0 a 1, como mostra a figura 4.4.

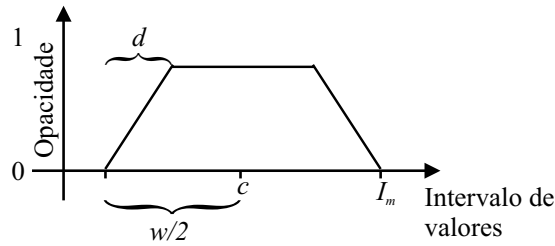


FIGURA 4.4 – Função de transferência no formato de um trapézio

Função exponencial. Outro exemplo de função de transferência simples que também é muito utilizada. A equação 4.7 apresenta um exemplo desta técnica, considerando que os valores de intensidade do voxel variam de 0 a 255. A figura 4.5 ilustra o formato do gráfico.

$$T_\alpha(\mathbf{x}_{s_i}) = \log_{10} \left(\frac{9 V(\mathbf{x}_{s_i})}{255} + 1 \right) \quad (4.7)$$

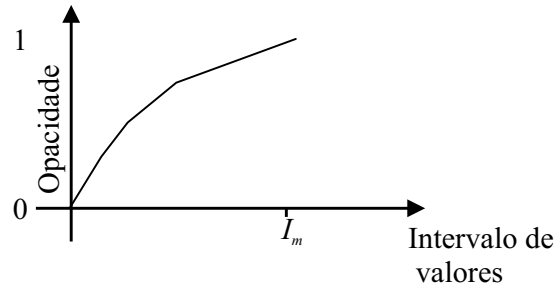


FIGURA 4.5 – Função de transferência exponencial

Função bloco. Quando há necessidade de atribuir um mesmo valor de opacidade a um conjunto de voxels cujas intensidades estão num determinado intervalo, se utiliza uma função de transferência do tipo bloco (figura 4.6). Um exemplo de equação para esta função é apresentado na equação 4.8:

$$T_\alpha(\mathbf{x}_{s_i}) = \begin{cases} 1, & I_c \leq V(\mathbf{x}_{s_i}) \leq I_m \\ 0, & \text{Caso contrário} \end{cases} \quad (4.8)$$

Cabe ressaltar que a equação 4.8 é apenas um exemplo, pois normalmente o mapeamento é realizado através da definição de mais do que um bloco, ou intervalo de

intensidades. Neste caso, basta aumentar o número de alternativas da equação 4.8, acrescentando uma para cada bloco. Para ilustrar, a figura 4.7 exemplifica o uso desta função na qual a opacidade dos voxels de densidade entre 79 e 81 foi mapeada para o valor 0.05, entre 240 e 255 para o valor 1, e das demais densidades para o valor 0.

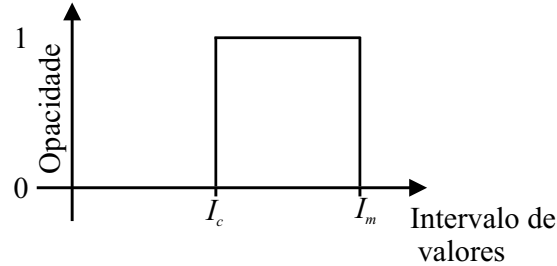


FIGURA 4.6 – Função de transferência do tipo bloco

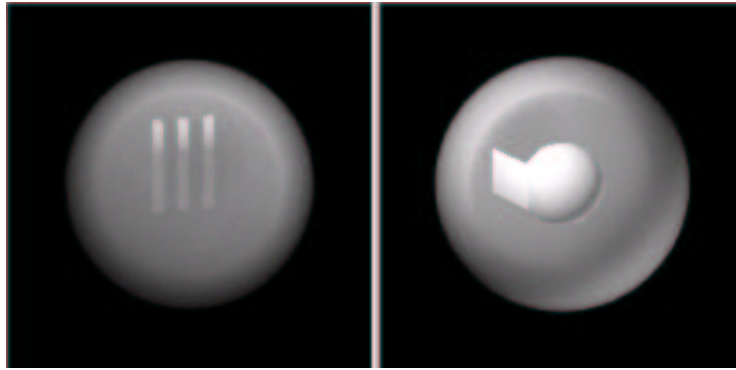


FIGURA 4.7 – Imagens resultantes da utilização da função de transferência de opacidade do tipo bloco

Função definida através de uma tabela. Considerando que a equação 4.8 pode ter um grande número de alternativas, uma forma mais simples de atribuir diferentes valores de opacidade para vários intervalos de intensidade dos voxels é através de uma tabela. Neste caso, por exemplo, um alto valor de opacidade pode ser mapeado para todos os voxels cujas intensidades estejam em um ou vários intervalos que identificam a(s) estrutura(s) de interesse, e um valor de opacidade baixo pode ser atribuído para outras estruturas que não são de interesse, de tal forma que fiquem transparentes. Com o uso de tabelas, geralmente criadas interativamente pelo usuário, cada valor de intensidade pode ser mapeado para um valor de opacidade diferente, sem utilizar uma função. Esta tabela pode ser interpretada como um vetor cujo índice indica o valor de intensidade e o conteúdo o valor de opacidade [SIL 2000]:

$$T_{\alpha}(\mathbf{x}_{s_i}) = \text{vet}[\text{round}(V(\mathbf{x}_{s_i}))] \quad (4.9)$$

onde, *vet* é o vetor representando a tabela e *round* é uma função de arredondamento.

Os valores de opacidade podem ser atribuídos de diversas maneiras nesta tabela. Uma possibilidade consiste em inserir um ou mais pontos de controle e especificar para cada um deles o valor de opacidade. As demais posições da tabela, entre os pontos, são preenchidas através da utilização de uma função de interpolação,

tal como a função linear. Outra alternativa é apresentar uma tabela na qual os pontos de controle estão inicialmente posicionados de maneira a caracterizar uma função mais simples, como a rampa, que pode ser alterada pelo usuário no sentido de acrescentar e/ou remover pontos de controle. Outra possibilidade, ainda, consiste no usuário selecionar diferentes intervalos de intensidades que terão valores de opacidade maiores do que zero, e, para cada intervalo, selecionar também a função que será utilizada, tal como linear e trapézio [KÖN 2001].

Funções baseadas na intensidade do voxel e na magnitude do gradiente.

Até aqui foram apresentadas as funções de opacidade mais simples, utilizadas quando a atribuição de um valor de opacidade para um voxel apenas de acordo com a sua intensidade é suficiente. Entretanto, para outros volumes, como por exemplo MRI, voxels que fazem parte do tecido cerebral e da pele podem ter a mesma intensidade. Assim, torna-se inviável fazer com que apenas uma destas estruturas, tal como a pele, seja transparente ou semitransparente para mostrar o interior do volume. Neste caso, é necessário usar outras informações para definir a função de transferência. O mais usual é usar a magnitude do gradiente local. Quanto maior for este valor, significa que há uma rápida mudança nos dados, indicando que existe uma superfície ou uma borda entre dois materiais [LIC 98]. Assim, simplifica-se a tarefa de definir uma superfície de um objeto dentro do volume de dados como opaca.

Levoy [LEV 88] usou a intensidade do voxel e a magnitude do gradiente na etapa de classificação para a extração de uma isosuperfície. Naquele trabalho era atribuída uma opacidade α_v para voxels que possuíam um valor selecionado I_v , e uma opacidade 0 para todos os outros voxels. Entretanto, para evitar o *aliasing*, voxels com valores próximos de I_v deveriam ter uma opacidade próxima de α_v . A aproximação adotada foi tal que a opacidade diminuía conforme o valor de intensidade se afastava de I_v , numa razão inversamente proporcional à magnitude do gradiente. Este mapeamento é aqui representado na equação 4.10 (também rescrita por Lichtenbelt et al. [LIC 98]):

$$T_\alpha(\mathbf{x}_{s_i}) = \begin{cases} 1, & \text{Se } \|\nabla V(\mathbf{x}_{s_i})\| = 0 \text{ e } V(\mathbf{x}_{s_i}) = I_v \\ 1 - \frac{1}{r} \left| \frac{I_v - V(\mathbf{x}_{s_i})}{\|\nabla V(\mathbf{x}_{s_i})\|} \right|, & \text{Se } \|\nabla V(\mathbf{x}_{s_i})\| > 0 \text{ e} \\ & V(\mathbf{x}_{s_i}) - r\|\nabla V(\mathbf{x}_{s_i})\| \leq I_v \leq V(\mathbf{x}_{s_i}) + r\|\nabla V(\mathbf{x}_{s_i})\| \\ 0, & \text{Caso contrário} \end{cases} \quad (4.10)$$

onde, r é a espessura desejada para a região de transição em voxels. A figura 4.8 ilustra a utilização desta função de transferência com a atribuição de diferentes valores para as variáveis I_v e r : 225 e 2.0, respectivamente, para geração da figura 4.8a; e 255 e 0.6 para a figura 4.8b.

Silva [SIL 2000] apresentou uma forma alternativa para a determinação do valor de opacidade baseado na magnitude do gradiente (equação 4.11):

$$T_\alpha(\mathbf{x}_{s_i}) = \frac{V(\mathbf{x}_{s_i})}{I_m} + P_{\|\nabla V\|} \left(\frac{\|\nabla V(\mathbf{x}_{s_i})\|}{\|\nabla V\|_m} - \frac{V(\mathbf{x}_{s_i})}{I_m} \right) \quad (4.11)$$

onde $\|\nabla V\|_m$ é o valor máximo de magnitude, e $0 \leq P_{\|\nabla V\|} \leq 1 \in \mathfrak{R}$ é um “peso” que resulta nas seguintes características: se $P_{\|\nabla V\|} = 0$, somente a intensidade influenciará a opacidade; se $P_{\|\nabla V\|} = 1$, somente a magnitude do gradiente influenciará o

valor obtido; caso contrário, o resultado será obtido pela soma ponderada dos dois valores. O valor máximo de magnitude pode ser obtido através do cálculo do gradiente de todos os voxels do volume, ou aproximado através do cálculo da diferença entre os valores mínimo e máximo dos voxels do volume [SIL 2000].

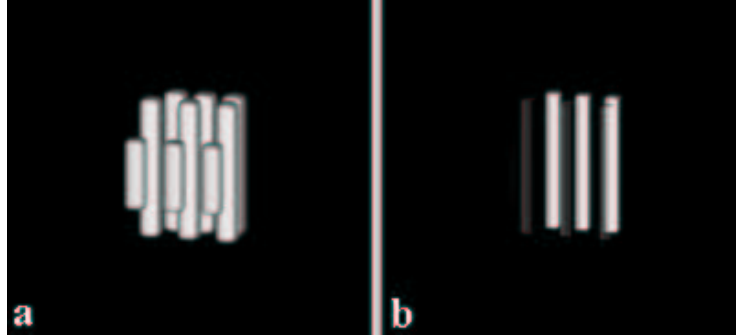


FIGURA 4.8 – Exemplos da utilização da função de transferência de opacidade apresentada por Levoy [LEV 88]

Para ilustrar a diferença do uso da magnitude do gradiente como argumento na função de transferência de opacidade, a figura 4.9 apresenta duas imagens geradas da seguinte forma. Na figura 4.9b, foi utilizada uma função linear onde os valores de intensidade menores do que 70 são desprezados (equação 4.4). A figura 4.9a é resultado da utilização de uma função de transferência baseada na magnitude do gradiente, calculada de acordo com a equação 4.12, com $I = 40$.

$$T_{\alpha}(\mathbf{x}_{s_i}) = \begin{cases} 0, & V(\mathbf{x}_{s_i}) < I \\ \frac{\|\nabla V(\mathbf{x}_{s_i})\|}{\|\nabla V\|_m}, & V(\mathbf{x}_{s_i}) \geq I \end{cases} \quad (4.12)$$

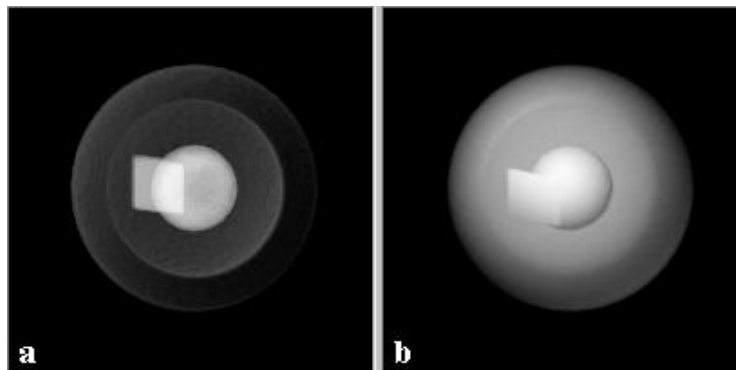


FIGURA 4.9 – Uso (a) da magnitude do gradiente considerando que $I = 40$ e (b) de uma função linear com $I_l = 40$

Funções baseadas em outros parâmetros. Outros parâmetros foram recentemente introduzidos para definir a função de transferência: a segunda derivada [KIN 98] e o histograma da imagem [LIU 99]. Enquanto a magnitude do gradiente é uma aproximação da derivada de primeira ordem, um operador utilizado para aproximar a derivada de segunda ordem é o Laplaciano ($\nabla^2 V$) [GON 93]. Ambos os operadores são utilizados para detecção de arestas. O máximo local da magnitude do gradiente identifica uma aresta (ou superfície), da mesma maneira que o valor

mínimo da segunda derivada. Portanto, uma alternativa para detecção de arestas ou isosuperfícies é localizar o zero da segunda derivada ou usar estas duas informações em conjunto.

O histograma de uma imagem digital mostra uma estimativa da probabilidade de ocorrência de cada tonalidade [GON 93]. No caso de um volume de dados, o histograma ($h(V)$) mostra quantos voxels existem para cada valor de intensidade. Portanto, através de um histograma se obtém informações que podem ajudar na escolha da função de transferência. Um exemplo é a facilidade para a identificação e posterior filtragem do ruído de uma imagem, bastando, neste caso, atribuir opacidade zero para todos os voxels com um determinado valor [LIC 98]. A habilidade de especificar uma “forma” particular para o histograma, capaz de salientar certos intervalos de valores, também pode ser interessante [GON 93].

Liu et al. [LIU 99] trabalharam com a geração das funções de transferências a partir da análise do histograma de intensidades, que permite distinguir os diferentes materiais. Já Kindlmann e Durkin [KIN 98] desenvolveram um método onde um “histograma de volume” é criado através da utilização das seguintes propriedades: valor escalar (ou intensidade do voxel); primeira derivada aproximada pela magnitude do gradiente ($\|\nabla V\|$); e segunda derivada aproximada pelo Laplaciano ($\nabla^2 V$). Estas propriedades são na verdade os três parâmetros utilizados para a geração semi-automática de funções de transferência usadas para visualização de isosuperfícies. A partir do estudo do histograma, é possível extrair características das superfícies que representam a transição entre materiais de valores relativamente constantes, que sempre aparecem como uma curva de uma forma específica no histograma. O usuário controla que porções das superfícies devem ser opacas sem a necessidade de saber os valores dos voxels.

4.3 Função de Transferência de Cor

Além das funções de transferência de opacidade, o desenvolvimento de funções de transferência de cor também é muito importante, pois, em geral, não há uma cor associada a cada voxel dos volumes de dados, há apenas um valor de intensidade, que no caso de dados médicos representa a densidade de um tecido. Porém, muitas vezes o uso de cores na visualização do volume de dados é útil. A atribuição de cores é muito semelhante à etapa de classificação e necessita de funções de transferência para transformar o valor de intensidade do voxel em uma cor RGB. Apesar do modelo RGB ser um dos mais utilizados nos algoritmos de visualização direta de volume [LEV 88, DRE 88, HE 96, LIC 98, PFI 99], outros modelos de cor, tais como HLS (*Hue-Lightness-Saturation*) e HSV (*Hue-Saturation-Value*), também são utilizados. König e Gröller [KÖN 2001] utilizaram o modelo de cor HLS, e Stokking [STO 98] avaliou o uso do modelo HSV na visualização integrada de dados funcionais e anatômicos, com as tabelas de cor manipuladas interativamente.

Uma formulação genérica para função de transferência de cor, considerando o modelo RGB, seria:

$$R = T_{c_r}(\mathbf{x}_{s_i}) \quad (4.13)$$

$$G = T_{c_g}(\mathbf{x}_{s_i}) \quad (4.14)$$

$$B = T_{c_b}(\mathbf{x}_{s_i}) \quad (4.15)$$

onde T_{cr} , T_{cg} e T_{cb} são as funções de transferência usadas para especificar, respectivamente, os valores RGB associados a parâmetros dos voxels de dados. Estas funções podem ser implementadas da mesma maneira que algumas funções de opacidade, tais como linear, rampa e por tabela, sendo que neste caso, um valor RGB deve ser especificado para cada ponto de controle. Apesar da grande utilidade, é necessário muito cuidado na definição das tabelas de cores, pois da mesma maneira que o uso de cores pode melhorar a visualização, também pode prejudicar. A sua má utilização pode ocultar características importantes do volume de dados, ou até mesmo “criar” características inexistentes [LEV 91].

O uso de uma tabela de cor e opacidade está ilustrado na figura 4.10. A tabela foi especificada através de uma função linear: até o valor de intensidade 70 a opacidade é zero, de 70 à 255 a opacidade aumenta linearmente. Para voxels com valor de densidade 0, ou entre 230 e 255, foi associada a cor branca. A cor vermelha foi utilizada para o valor de densidade 80, e a cor verde para 150. Nos demais valores de densidade a cor é interpolada.

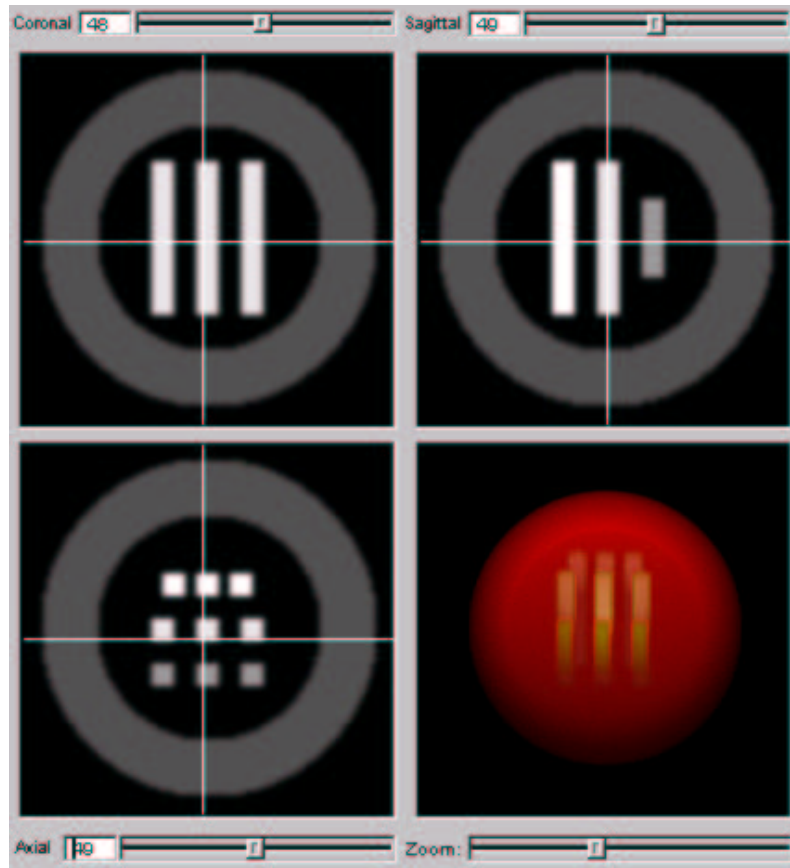


FIGURA 4.10 – Imagem gerada com a utilização de uma tabela de cor, que representa uma função de transferência linear

Geralmente apenas o valor de intensidade do voxel é usado como parâmetro para funções de transferência de cor, mas nada impede que outros parâmetros também sejam fornecidos [HE 96, LIC 98]. Kniss et al. [KNI 2001, KNI 2002] mostraram uma classe de funções de transferência baseadas em três parâmetros: valor escalar (intensidade), magnitude do gradiente e segunda derivada direcional. Tais funções, chamadas de multidimensionais, são especificadas através da manipulação

de *widgets*, ou seja, objetos geométricos projetados para prover o usuário com uma interface 3D. A escolha pelo uso do terceiro parâmetro está relacionada com a detecção de arestas, uma vez que permite uma definição mais precisa do limiar das estruturas em um volume. A figura 4.11 mostra um exemplo desta interface através do uso de funções de transferência 2D e 3D. Na imagem gerada com a função 2D (4.11a), os limiares dos diferentes materiais se sobrepõem. Porém, através do uso da função 3D (4.11b) foi possível selecionar com maior precisão as superfícies que deveriam ser exibidas na imagem final, revelando, neste caso, a borda do esmalte do dente.

Na figura 4.11 os *widgets* de classificação consistem em um triângulo invertido com o ponto da base associado a um eixo de valores de dados. Este ponto, a altura e a largura do triângulo, podem ser alterados. Também é possível evitar a associação de um valor de opacidade quando a magnitude do gradiente for baixa, colocando uma barra que é controlada por uma bola na aresta do lado direito do triângulo. A região do trapézio define os valores de dados e magnitudes do gradiente que recebem cor e opacidade. A cor é constante, mas a opacidade varia, tendo como valor máximo o centro do triângulo e diminuindo linearmente até chegar a zero nas arestas da esquerda e da direita. [KNI 2001, KNI 2002]

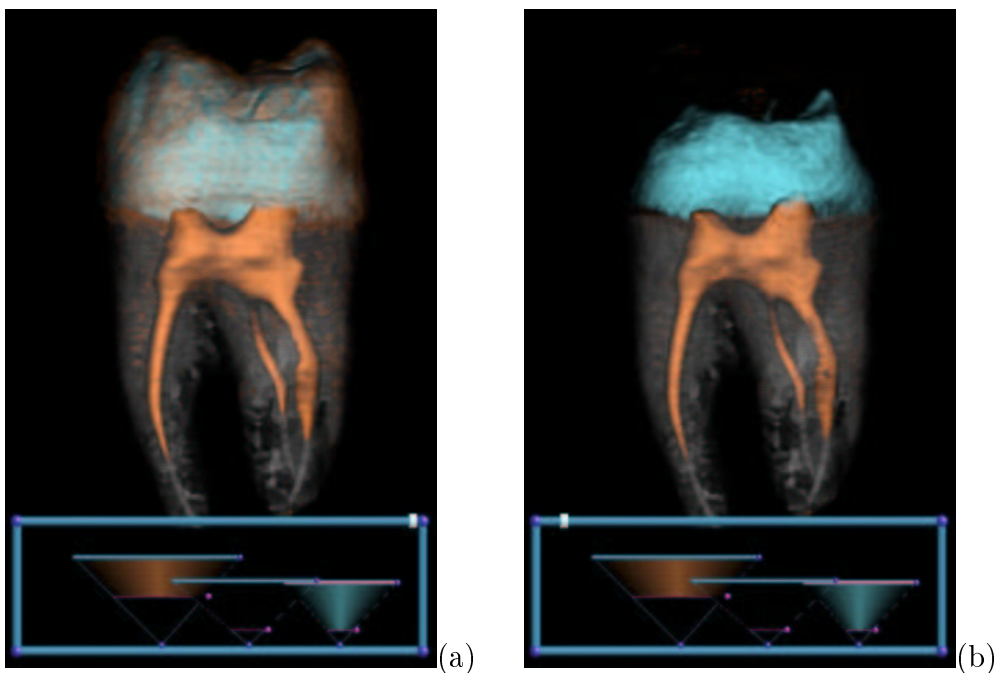


FIGURA 4.11 – Exemplo do uso de funções de transferência 2D (a) e 3D (b) no qual os *widgets* de classificação triangulares baseiam-se na função de transferência de opacidade de Levoy (equação 4.10) para extração de uma isosuperfície - imagens extraídas de Kniss et al. [KNI 2001]

4.4 Especificação de Funções de Transferência

Através da análise dos vários trabalhos descritos na literatura, foi possível observar que as diferentes abordagens para especificação das funções de transferência variam, basicamente, de acordo com:

- Parâmetros utilizados (intensidade do voxel, magnitude do gradiente, segunda derivada, histograma);
- Grau de interação do usuário na especificação das funções (automática, semi-automática ou manual);
- Abordagem adotada (tentativa e erro, centralizada na imagem, centralizada no dado, entre outras).

Pfister et al. [PFI 2001] foram pioneiros na difícil tarefa de classificar as funções de transferência. Eles identificaram quatro abordagens diferentes para a especificação de funções de transferência:

- “Tentativa e erro”;
- Centralizada no dado, sem usar um modelo de dados [BAJ 97];
- Centralizada no dado, usando um modelo de dados [KIN 98];
- Centralizada na imagem [MAR 97].

Entre estas abordagens, as centralizadas no dado foram consideradas as mais promissoras.

Kindlmann [KIN 2001], em um trabalho recente, organizou a apresentação dos métodos de especificação de funções de transferência nas seguintes classes:

- “Tentativa e erro”;
- Baseadas na detecção de características espaciais [FAN 98, RHE 2001, HLA 2001];
- Centralizada na imagem [HE 96, MAR 97, PAT 98, MA 99, JAN 2001, KÖN 2001];
- Centralizada no dado [BAJ 97, TEN 2001, PEK 2001, KIN 98];
- Outros [HLA 2000, KNI 2001].

Nesta seção, foi seguida a organização proposta por Kindlmann para descrever brevemente as diferentes abordagens para especificação das funções de transferência, já que esta engloba todos os trabalhos analisados por Pfister et al. [PFI 2001], além de outros recentemente apresentados na literatura. O principal objetivo aqui é ilustrar as diferenças e semelhanças entre as abordagens que fazem parte de uma mesma classe.

Tentativa e erro. A especificação de funções de transferência através de tentativa e erro consiste em um método manual onde o usuário especifica as tabelas de cor e opacidade. Isto pode ser feito de diversas maneiras, mas o mais usual é permitir que pontos de controle, para os quais o usuário especifica a cor e opacidade, sejam inseridos e removidos das tabelas. Depois, uma função de interpolação é usada para preencher todas as posições intermediárias da tabela.

Baseada na detecção de características espaciais. Fang et al. [FAN 98] apresentaram um modelo de função de transferência baseado em imagem que integra técnicas de processamento de imagem 3D, tais como realce e detecção de arestas, no

pipeline de visualização. Nesta abordagem, as funções de transferência são representadas como uma seqüência de técnicas de processamento de imagens 3D, e permitem que os usuários ajustem um conjunto de parâmetros para gerar as imagens de acordo com o tipo de visualização desejada.

A técnica de *Volume Illustration* [RHE 2001] combina o modelo de iluminação baseado em física com a habilidade de realçar características importantes usando técnicas de *rendering* não-fotorealístico. Assim, é fornecido um *framework* para realçar a percepção da estrutura dos modelos através do aumento de certas características e adição de efeitos de iluminação. Esta técnica, complexa e flexível, pode ser considerada como um mecanismo de representação de funções de transferência multivariadas. Com esta abstração, por exemplo, os parâmetros que controlam o realce de arestas, bem como a transparência de acordo com a distância, especificam funções de transferência multivariadas a partir de um conjunto de valores que incluem valor e localização do voxel, gradiente, entre outros.

Hladùvka et al. [HLA 2001] trabalharam no desenvolvimento de uma técnica de filtragem para a identificação de superfícies e pequenas estruturas em volumes de dados. O algoritmo se baseia no reconhecimento de áreas onde o valor da segunda derivada é negativo, que são usadas para definir uma função que mostra os voxels que possuem um conteúdo significativo para a visualização. Segundo Kindlmann [KIN 2001], estes três trabalhos baseados na detecção de características espaciais, são extremamente flexíveis e não correspondem exatamente ao desenvolvimento de funções de transferências.

Centralizada na imagem. Com o objetivo de auxiliar o usuário na escolha da função de transferência adequada, He et al. [HE 96] trataram o processo de encontrar esta função como um problema de otimização de parâmetros que é resolvido com técnicas de busca estocásticas. Após a geração de um conjunto inicial de funções de transferência randômicas ou pré-definidas, o usuário avalia as imagens e o processo se repete até que resultados satisfatórios sejam encontrados. Certas funções de transferência, tal como a função linear, são utilizadas inicialmente para gerar, de forma automática, um primeiro grupo de funções com as suas respectivas imagens (figura 4.12). Para isto, o sistema usa uma biblioteca que possui algumas das funções de transferências mais difundidas (linear, rampa, entre outras).

De maneira similar, uma vez que também permite que o usuário analise as funções de transferência através de imagens, Marks et al. [MAR 97] desenvolveram uma abordagem genérica para a especificação de diversas funções usadas para a geração de imagens ou animações. Um conjunto de imagens ou animações, gerado através da variação de parâmetros de entrada, é apresentado de forma organizada para o usuário na chamada *Design GalleryTM* (DG). A fase de construção da DG demanda muito processamento e pode ser feita como uma etapa de pré-processamento, quando o equipamento não está sendo utilizado. Após a sua construção, é possível explorar rapidamente as imagens geradas. No caso de técnicas de visualização direta de volume, quando uma das imagens apresentadas na DG é selecionada, tem-se acesso às funções de transferência, que são parametrizadas por uma polilinha com pontos de controle.

Apesar de não ter sido abordado por Kindlmann [KIN 2001], o trabalho apresentado por Liu et al. [LIU 99] também pode ser classificado como centralizado na imagem. Estes autores desenvolveram um método para a geração de funções de

transferência adequadas às necessidades dos usuários. Inicialmente, considerando o histograma do volume, são definidas várias funções de transferência com o objetivo de distinguir os diferentes materiais. Depois, para cada função, é definido um conjunto de parâmetros (por exemplo, *threshold* de opacidade e *threshold* de densidade). As imagens geradas são então submetidas à avaliação do usuário e de acordo com esta avaliação novos parâmetros podem ser especificados até que seja alcançado o resultado esperado.



FIGURA 4.12 – Interface do sistema que apresenta um conjunto de imagens geradas com diferentes funções de transferência (extraída de He et al. [HE 96])

Uma interface com o usuário baseada em grafo foi apresentada por Pattern e Ma [PAT 98, MA 99]. Nos trabalhos desenvolvidos, as imagens resultantes do processo de visualização são conectadas a outras imagens de acordo com os parâmetros passados para o sistema, facilitando, assim, o entendimento de como as mudanças nos parâmetros afetam a visualização. A figura 4.13 mostra um exemplo da interface no caso da visualização de um volume de dados MRI. Nesta figura, a linha pontilhada corresponde à ação de trocar a função de transferência do nó 3 pela função de transferência do nó 7, o que ocasiona um novo processamento para gerar a(s) imagem(ns) final(is) naquela ramificação.

Outra interface desenvolvida por Ma e Jankun-Kelly [JAN 2001] é baseada em uma planilha. Conforme mostra a figura 4.14, a interface representa uma janela 2D num espaço de parâmetros de visualização multidimensional e a exploração de dados é feita através da navegação neste espaço. A visualização é definida por uma combinação de parâmetros facilmente identificada.

König e Gröller [KÖN 2001] introduziram um novo paradigma de interface para especificação de funções de transferência em sistemas de visualização de dados médicos. A idéia principal é que para cada intervalo de intensidades que contribuem para a imagem final, sua cor e sua opacidade são determinadas de forma

independente, uma de cada vez, em interfaces separadas. O usuário seleciona o(s) intervalo(s) de interesse, seleciona uma função para fazer o mapeamento para valores de opacidade em cada intervalo (linear, rampa, trapézio, etc.), e seleciona uma função para especificar a cor também em cada intervalo. A figura 4.15 mostra toda a interface do sistema, na qual foram definidas três estruturas de interesse diferentes: pele, vasos sanguíneos e ossos. Para cada estrutura foram utilizadas diferentes funções de transferência.

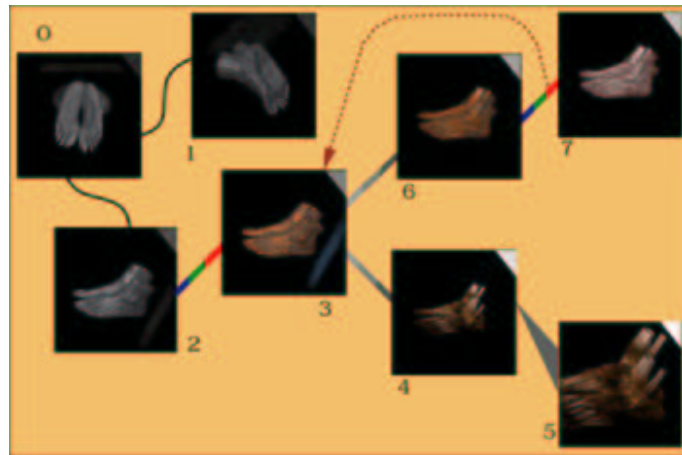


FIGURA 4.13 – Interface baseada em grafo (extraída de [MA 99a])

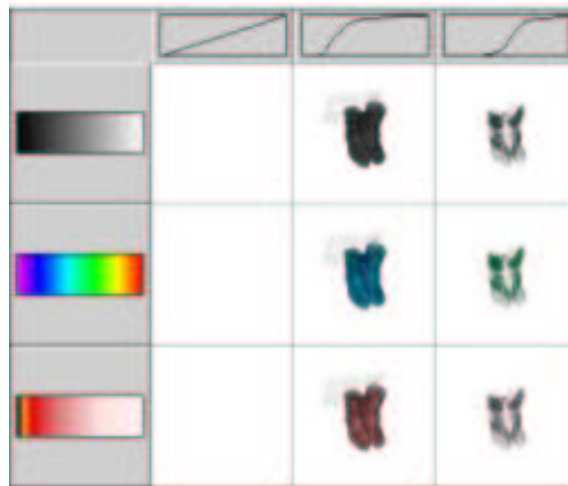


FIGURA 4.14 – Interface baseada em planilha (extraída de [JAN 2001])

Conforme Kindlman afirma [KIN 2001], é possível observar que estas técnicas são centralizadas na imagem. Em outras palavras, o foco está mais nas imagens geradas e na interface do sistema do que na especificação da função de transferência. O usuário analisa as imagens e os parâmetros podem ser alterados até que seja gerado o resultado esperado. Em praticamente todos estes sistemas são utilizadas funções simples, tais como linear e rampa, que se baseiam na intensidade do voxel. A geração das imagens depende da interação com o usuário, que seleciona imagens e/ou manipula parâmetros para a especificação da função de transferência para alcançar os resultados esperados.

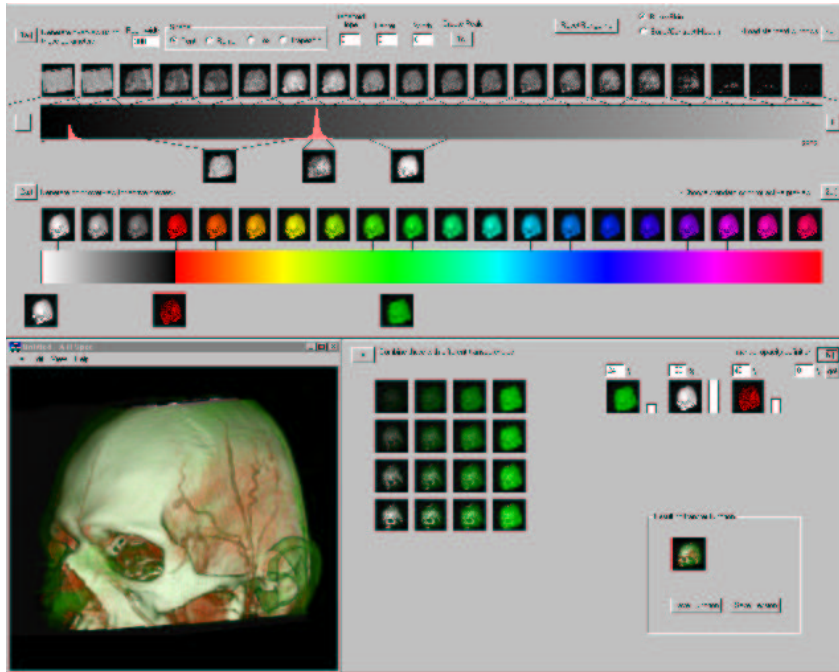


FIGURA 4.15 – Especificação de uma função de transferência com três contribuições (extraída de König e Gröller 2001 [KÖN 2001])

Centralizada no dado. Esta abordagem, ao contrário das anteriores, é centralizada na análise dos dados. Isto significa que, ao invés do usuário analisar a imagem final para então realizar uma alteração de parâmetro ou solicitar ao sistema que um novo conjunto de imagens seja gerado, ele analisa as características do volume de dados para encontrar o isovalor que irá permitir a visualização da(s) superfície(s) de interesse.

Um componente de interface com o usuário que melhora a qualidade da interação foi desenvolvido por Bajaj et al. [BAJ 97]. *Contour Spectrum* tem como objetivo auxiliar o usuário a encontrar uma função de transferência adequada, identificando o melhor “isovalor” para extração das principais estruturas de um volume de dados. Esta técnica corresponde a uma assinatura que consiste no processamento em tempo real de propriedades de uma isosuperfície, tais como área da superfície de contorno e integral do gradiente. Estas propriedades são apresentadas para o usuário como uma coleção de gráficos que auxiliam na seleção de isovalores.

Tenginakai et al. [TEN 2001] também utilizam uma assinatura para explorar os dados e identificar isosuperfícies. Porém, neste caso, as assinaturas são obtidas a partir de momentos centrais de k -ésima ordem, que simplificada consistem em formas de avaliar a tendência da distribuição dos dados (um histograma é um exemplo de um momento central de primeira ordem). O forte relacionamento existente entre os vários momentos centrais são influenciados pela presença de materiais de interface. Assim, as assinaturas podem ser usadas para localizar os isovalores que estão presentes nestes materiais.

Um método computacionalmente eficiente para a detecção automática de transições de intensidade em volume de dados foi apresentado por Pekar et al. [PEK 2001]. Neste método, as transições dominantes correspondem ao valor máximo nos histogramas de valores de cinza processados de acordo com o valor do voxel

e o Laplaciano, e acumulados de acordo com um limiar (*threshold*), que basicamente corresponde à uma superfície. As transições de intensidade detectadas podem ser usadas para a especificação das funções de transferência. Já Kindlman e Durkin [KIN 98] desenvolveram uma técnica para geração semi-automática de funções de transferência de opacidade. Tais funções são geradas a partir do relacionamento entre os valores de intensidade, a primeira e a segunda derivada ao longo da direção do gradiente. Estas informações são usadas para a construção de um histograma que possibilita a identificação das transições de materiais.

A proposta de Fujishiro et al. [FUJ 99], que baseia-se na análise da topologia de campos 3D, também é aqui classificada como centralizada no dado. Neste caso, a abordagem de *Reeb graph* usada para a modelagem topológica de superfícies 3D, foi estendida para capturar o esqueleto topológico de um campo volumétrico. Tomando como base o resultado desta análise, que é representado na forma de um *hyper Reeb graph*, são determinados os candidatos apropriados para as funções de transferência.

Novos domínios e tipos de interação (correspondente à classe “Outros” de Kindlman [KIN 2001]). Hladùvka et al. [HLA 2000] preocuparam-se com a definição semi-automática das funções de transferência no domínio das magnitudes da curvatura principal. As curvaturas principais consistem em dois números reais que indicam o quanto a normal varia nas direções dos vetores tangentes em cada ponto da superfície. Desta forma, as propriedades visuais são determinadas de acordo com a forma do objeto.

Kniss et al. [KNI 2001] desenvolveram uma forma diferente de interação para definir funções de transferência multi-dimensionais, as quais se baseiam no valor de intensidade, na magnitude do gradiente e na segunda derivada [KIN 98]. Neste caso, um conjunto de *widgets* é manipulado para especificar as funções de transferência de forma intuitiva.

As funções de transferência geralmente estão associadas ao conteúdo do volume de dados e não à geometria, ou seja, são definidas em função do valor de densidade e/ou da magnitude do gradiente de cada voxel. Entretanto, em uma abordagem diferente, Mullick et al. [MUL 2000] utilizam a opacidade linear ou rampa baseada na distância do observador. No CVR, na região onde as amostras são processadas, a opacidade aumenta conforme o processamento das amostras avança ao longo do raio, independente dos valores de intensidade. Neste caso, se chega a uma equação análoga à equação 4.3:

$$T_{\alpha}(\mathbf{x}_{s_i}) = d_i/d_m \quad (4.16)$$

onde d_i é a distância da amostra s_i em relação a s_0 e d_m é a distância máxima. As outras funções descritas anteriormente, como por exemplo a rampa, também podem ser utilizadas da mesma maneira. A única diferença está em considerar a distância ao invés dos valores dos voxels no momento de definir as funções e/ou tabelas de opacidade.

Existem ainda outros trabalhos descritos na literatura que estão relacionados com a definição de funções de transferência, apesar de não tratarem diretamente a sua especificação. Por exemplo, Bergman et al. [BER 95] desenvolveram uma abordagem interativa para guiar o usuário na seleção de mapa de cores, o que é extremamente útil quando as funções são definidas através de tabelas. Botha e Post [BOT 2002] apresentaram uma técnica para fornecer respostas visuais em tempo real durante a especificação da função de transferência. Esta técnica não

substituí, por exemplo, as abordagens apresentadas por Bajaj et al. [BAJ 97] e Kindlman e Durkin [KIN 98], mas poderia ser usada para complementá-las.

4.5 Comentários Finais

A observação das várias técnicas permite identificar que o que varia em cada uma são os parâmetros de entrada, o tipo de interação permitida e o método básico adotado para a especificação da função de transferência. Além disso, apesar das funções de transferência terem como um dos objetivos mostrar isosuperfícies, que correspondem à(s) estrutura(s) de interesse na visualização, apenas algumas técnicas se preocupam especificamente com a extração de isosuperfícies [HLA 2001, BAJ 97, TEN 2001]. A tabela 4.1 apresenta o enquadramento dos principais trabalhos encontrados na literatura e descritos nesta seção, nas classes identificadas.

TABELA 4.1 – Abordagens para a especificação das funções de transferência (classificação baseada em Kindlmann [KIN 2001])

Classificação	Característica	Abordagens descritas na literatura
Tentativa e erro	O usuário é responsável pela edição manual das funções de transferência ou tabelas de cor e opacidade	[PFI 2001]
Baseada na detecção de características espaciais	Operam em um espaço de parâmetro diferente (não são exatamente funções de transferência)	[FAN 98, RHE 2001, HLA 2001]
Centralizada na imagem	Especificação das funções de transferência através dos resultados da visualização	[HE 96, MAR 97, PAT 98, MA 99, LIU 99, JAN 2001, KÖN 2001]
Centralizada no dado	Especificação das funções de transferência através da análise do volume de dados	[BAJ 97, KIN 98, FUJ 99, TEN 2001, PEK 2001]
Novos domínios e tipos de interação	Desenvolvimento de novas formas de interação ou uso de novos domínios	[MUL 2000, HLA 2000, KNI 2001]

Em 1992, Elvins [ELV 92] afirmou que existia a necessidade de se desenvolver interfaces com o usuário que fossem intuitivas e ferramentas de classificação semi-automáticas para os sistemas de visualização volumétrica. Em 2001, a tarefa de definir uma função de transferência foi considerada como um dos dez principais problemas de visualização de volumes [PFI 2001]. Portanto, hoje, dez anos depois, observa-se que ainda há espaço para investigar o projeto de funções de transferência, principalmente quando se trabalha com volumes de dados complexos, difíceis de visualizar e cujo conteúdo não é conhecido [PFI 2001].

5 Visualização de Volumes Multimodais

Neste capítulo, inicialmente, são descritos os requisitos necessários (seção 5.1) para visualização integrada de dados multimodais. As principais características de alguns sistemas de visualização de dados multimodais são apresentadas (seção 5.2), e são analisadas as técnicas de visualização integrada já desenvolvidas (seção 5.3).

5.1 Requisitos

Para a visualização de imagens geradas a partir de volumes de dados multimodais, o primeiro passo fundamental consiste em **registrar** os volumes ou imagens. Simplificadamente, o registro corresponde ao processo de mapear pixels de uma imagem (2D ou 3D), chamada de referência, para outra imagem, chamada de teste. Tais imagens podem ser obtidas do mesmo objeto em momentos diferentes, por mais de um aparelho ou a partir de ângulos de visualização distintos [MAI 98, UDU 2000]. O objetivo desta operação é representar as informações que pertencem ao mesmo objeto de interesse em um sistema de coordenadas comum. Conforme mencionado por diversos autores [BRO 92, BAN 98, UDU 2000], o registro é necessário para:

- Integrar informações obtidas de sensores diferentes;
- Encontrar alterações nas imagens obtidas em momentos diferentes ou em condições diferentes;
- Obter informações 3D a partir de imagens onde a câmera ou os objetos da cena foram movidos.

Muitas técnicas de registro têm sido desenvolvidas para vários tipos de dados. No caso de imagens médicas, a precisão das técnicas descritas na literatura está parcialmente relacionada com a precisão do equipamento e do processo de aquisição [VEM 97, MAI 98]. Já no caso específico da visualização de dados multimodais, o alinhamento de imagens obtidas através de diferentes equipamentos é feito para permitir uma análise correta das suas características. Isto é muito útil para diagnóstico clínico e planejamento radioterápico ou pré-cirúrgico, quando imagens de diferentes modalidades são freqüentemente adquiridas para fornecer informações complementares.

A análise dos algoritmos de registro já desenvolvidos mostra que as técnicas de registro podem ser diferenciadas de acordo com vários critérios: dimensionalidade, 2D/2D, 2D/3D ou 3D/3D; maneira como o processamento é realizado, por exemplo, a partir de objetos artificiais introduzidos na imagem, ou baseado na propriedade do voxel; variedade de modalidades envolvidas, monomodal ou multimodal; entre outras. Assim, considerando as características em comum que podem ser distinguidas e enumeradas, Maintz e Viergever [MAI 98] classificaram os métodos de registro de imagens médicas com base em nove critérios: dimensionalidade, base do registro, natureza e domínio da transformação, grau de interação com o usuário, procedimento de otimização, modalidades envolvidas, especificação do tópico de estudo (por exemplo, se são imagens de um mesmo paciente ou não), e parte do corpo onde foi feita a aquisição das imagens.

A maioria dos algoritmos tenta melhorar os resultados finais em termos de performance, precisão e confiabilidade usando diferentes abordagens [MAI 98, BRO 92, ZUI 96]. No caso específico de registro de imagens multimodais, o algoritmo mais difundido é o de *Mutual Information* (MI) [STO 98, COL 98], que é um método de registro automático baseado na propriedade do voxel e na teoria da informação [COL 95, VIO 95, MAI 96, MAE 98].

Depois do registro, um passo de **fusão**, que envolve um algoritmo de visualização volumétrica, é requerido para a visualização simultânea de dois volumes de dados. Esta etapa é muito importante, uma vez que imagens multimodais são formadas através da combinação de características complementares de volumes diferentes, cujas modalidades também são diferentes. Várias abordagens para este tipo de visualização foram propostas e algumas delas estão descritas na seção 5.3.

Após a exibição da imagem final, **ferramentas de interação** são desenvolvidas para possibilitar a exploração dos dados. Tais ferramentas incluem: técnicas de corte (capítulo 3) e de especificação de funções de transferência (capítulo 4); navegação no volume de dados; manipulação da câmera e dos dados; ajuste de parâmetros para visualização de estruturas internas; e extração de medidas e informações funcionais. A extração de medidas é útil no desenvolvimento de técnicas de quantificação e exploração volumétrica. Por exemplo, quando o usuário seleciona uma estrutura na imagem e deseja saber o seu volume, a sua dimensão ou a informação funcional de um determinado ponto da imagem [UDU 2000].

Algumas ferramentas de interação podem requerer um passo de pré-processamento, tal como **segmentação**. Além da sua utilidade para isolar uma determinada estrutura, algumas técnicas de extração de medidas, registro e visualização de volumes multimodais também dependem da segmentação dos dados. Assim, conclui-se que os requisitos para o desenvolvimento de um sistema de visualização de dados multimodais incluem técnicas de segmentação, registro, visualização e interação.

5.2 Sistemas de Visualização de Volumes Multimodais

Em geral, os sistemas de visualização são desenvolvidos com o objetivo específico de visualizar volumes multimodais, ou são criados como módulos de sistemas de visualização científica mais genéricos. O AVS (*Application Visualization System*) [UPS 89] é um exemplo do segundo caso, pois é uma ferramenta genérica de visualização, com uma linguagem visual, orientada a objetos e multi-plataforma que converte, por exemplo, imagens médicas em representações visuais interativas. Recentemente o AVS incorporou um módulo que permite a visualização de volumes de dados multimodais [INC 2002].

O ANALYZETM [ROB 94, ROB 98], que foi desenvolvido pelo *Biomedical Imaging Resource* na clínica Mayo, é um sistema que fornece um conjunto integrado de ferramentas de exibição, manipulação e extração de medidas, permitindo uma investigação e avaliação detalhada de imagens biomédicas tridimensionais. Este sistema foi projetado e programado para ser altamente eficiente, interativo e genérico. Os algoritmos implementados incluem: visualização interativa de volumes, segmentação automática através da utilização de morfologia matemática, registro

de imagens usando similaridade de superfícies e classificação de tecidos através de análise multi-espectral. O sistema foi implementado na linguagem de programação C e pode ser executado em diversas plataformas. A sua arquitetura permite uma fácil inclusão de novos módulos [ROB 94, MAY 2002].

Algumas características do ANALYZETM, que pode ser usado com várias modalidades de aquisição (CT, MRI, PET, SPECT, Ultra-som e imagens obtidas com microscópios), são:

- Gerenciamento e recuperação de imagens;
- Geração e exibição de imagens 2D e 3D (visualização direta de volume incluindo técnicas de interação e diversas ferramentas relacionadas);
- Possibilidade de composição de objetos, que correspondem a estruturas segmentadas que podem ser exibidas com diferentes níveis de transparência, com o objetivo de visualizar estruturas internas;
- Extração de medidas e análise quantitativa;
- Análise multi-espectral para facilitar o processo de segmentação;
- Técnicas de classificação;
- Registro de imagens multimodais por correlação usando alinhamento paramétrico Chamfer [BRO 88].

O VROOM, um sistema orientado a objetos que permite fazer a visualização integrada de volumes de dados multimodais, foi desenvolvido por Zuiderveld na sua tese de doutorado [ZUI 95]. Entre as funcionalidades fornecidas, destacam-se: técnica de otimização do algoritmo de *ray casting*; otimização de operações de manipulação dos volumes; suporte para utilização de arquiteturas paralelas; e utilização de cores para a visualização integrada de informação funcional e superfícies anatômicas. Técnicas de segmentação e registro não foram abordadas, tornando-se necessário assumir a realização de um pré-processamento adequado dos dados. Além da implementação de diversos algoritmos de visualização, foi desenvolvida uma técnica chamada *Normal Projection* (seção 5.3.1) para a visualização integrada de informação quantitativa e de superfícies anatômicas. VROOM foi implementado utilizando a linguagem de programação C++ para execução em plataformas Unix. A interface com o usuário foi implementada em Tcl/Tk [TCL 2002].

Stokking [STO 98] também desenvolveu uma ferramenta para a visualização integrada de imagens funcionais e anatômicas, sendo seu trabalho de doutorado voltado para as estruturas cerebrais. O registro foi realizado tanto pelo uso de marcadores externos em conjunto com um *software* desenvolvido localmente, como pela técnica de MI [MAE 97], ou ainda pelo uso das facilidades de alinhamento de superfícies do ANALYZETM. CACTUS, um método de segmentação automática desenvolvido pelo autor, que se baseia em operações simples como *thresholding*, crescimento de região e erosão, também foi utilizado em alguns estudos de caso para a segmentação de conjuntos de dados. Para a visualização 3D integrada foi utilizado o VROOM com a inclusão de outras classes desenvolvidas para a implementação da técnica de *Normal Fusion* (seção 5.3.2). Esta técnica faz um mapeamento de valores quantitativos, por exemplo SPECT, sobre uma superfície extraída de dados

MRI, independente da posição de observação e com precisão. Também foi apresentada uma avaliação do uso do modelo de cores HSV com a técnica de *Normal Fusion* [STO 98, ZUI 95, ZUI 96]. Neste trabalho foram utilizadas algumas ferramentas já apresentadas na literatura, não sendo, então, desenvolvido um único sistema ou *framework*.

Já o foco da tese de doutorado de Noordmans [NOO 97], resumidamente, foi a análise interativa de imagens volumétricas microscópicas. Foi projetado um sistema, chamado 3DBench, para resolver o problema das diferentes interfaces e metodologias de extração de medidas usadas na maioria dos sistemas existentes. Também foram desenvolvidas técnicas de otimização para visualização direta de volume baseadas em *ray casting* com o objetivo de permitir a interação com o volume de dados. Além disso, foi especificada uma abordagem baseada em física para atribuição de cor e foi desenvolvida uma simulação da interação da luz quando são utilizados múltiplos volumes. Entre as funcionalidades fornecidas estão: garantia da ausência de artefatos e identificação dos diferentes materiais pela cor; visualização de objetos em conjunto com a percepção da estrutura da superfície de objetos circundantes; e possibilidade de ver simultaneamente a superfície e a estrutura volumétrica de um objeto. 3DBench possui rotinas genéricas para manipular e visualizar dados volumétricos. Também possui um *slicer* interativo e um visualizador direto de volume para visualização 2D e 3D de imagens volumétricas não processadas, segmentadas, multimodais, multi-canais ou qualquer combinação delas.

Um *framework* baseado em Open Inventor [WER 94], que possibilita operações de fusão, foi desenvolvido por Hastreiter e Ertl [HAS 98] para a visualização simultânea e interativa de volumes de dados registrados. O algoritmo de registro implementado baseia-se em *Mutual Information*. Para integrar as técnicas de registro e visualização, o método desenvolvido permite visualizar dois volumes de dados registrados usando mapeamento de textura 3D acelerado por *hardware* disponível nas estações de trabalho gráficas. Além disso, devido à plena integração na hierarquia de classe da API Open Inventor e ao uso de OpenGL, uma variedade de funções está disponível, como por exemplo, a utilização de planos de corte separados e tabelas de cor individuais para cada volume de dados.

Além dos trabalhos acima mencionados, outros sistemas de visualização de volumes de dados multimodais são utilizados para auxiliar o procedimento cirúrgico. O principal objetivo destes sistemas é fazer o alinhamento entre imagens médicas obtidas antes da cirurgia (por exemplo, através de exames de CT e MRI) e o ambiente real do paciente durante uma operação. Isto possibilita ao cirurgião ver o paciente e, ao mesmo tempo, exibir as estruturas internas alinhadas com esta posição de observação. Neste caso, a maior preocupação está em fazer o registro entre as imagens obtidas antes da cirurgia e as imagens reais. Os trabalhos desenvolvidos por Peters et al. [PET 96], Leventon [LEV 97] e Gering [GER 2001] são exemplos de aplicações deste tipo que, como fogem do escopo deste trabalho, não serão abordadas.

5.3 Técnicas de Visualização de Volumes Multimodais

Como as imagens multimodais são formadas pela combinação de características complementares de volumes de dados diferentes, a etapa de fusão é essencial. Algumas abordagens para este tipo de visualização integrada têm sido propostas e

podem ser encontradas na literatura. Stokking [STO 98] apresenta uma revisão de técnicas para visualização integrada de volumes de dados multimodais de imagens do cérebro.

As técnicas de visualização de volumes multimodais descritas na literatura baseiam-se tanto em visualização através de superfícies como em visualização direta de volume. Diferentes formas de interação com as imagens resultantes também são apresentadas, como por exemplo, a técnica chamada *linked feature display*, que é uma extensão de uma técnica 2D que exhibe as fatias correspondentes de duas (ou mais) modalidades em várias janelas separadas, cada uma com seu próprio controle [STO 98]. Apesar de simples, esta técnica pode ser muito útil quando é estendida para incluir “cursos acoplados” que mostram as localizações correspondentes nas fatias de diferentes modalidades. Para trabalhar com volumes, uma localização na imagem 3D é relacionada com a posição equivalente em cada imagem 2D de cada modalidade. Neste caso, recursos como pontos, linhas ou planos podem ser usados.

Um exemplo da utilização desta técnica foi desenvolvido por Levi et al. [LEV 89], que implementaram um sistema interativo de seleção de planos para a visualização simultânea de seções ortogonais e imagens 3D. Nesta aplicação, um cursor no formato de uma linha pode ser movido com o mouse através do modelo 3D do cérebro, e as imagens de uma ou mais seções correspondentes, tais como MRI e PET, são atualizadas no lado direito da tela, conforme exemplificado na figura 5.1.

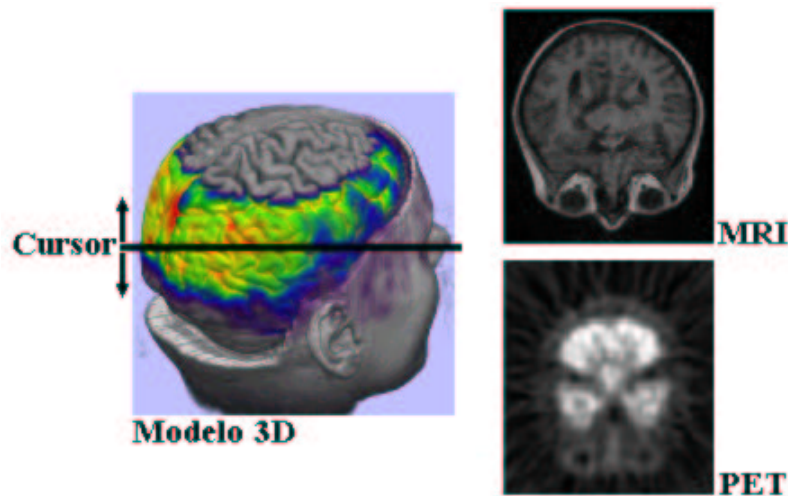


FIGURA 5.1 – Exemplo da técnica de *linked feature display*

A técnica chamada de **janela multimodal**, que foi revisada por Stokking [STO 98], assume que a visualização 3D dos volumes de dados é basicamente uma representação 2D dos dados e todas as técnicas para visualização integrada 2D podem ser aplicadas. Nesta técnica, uma “janela” na visualização de volumes de dados anatômicos é usada para revelar a parte equivalente dos dados funcionais. Uma vez que o *rendering* 3D dos volumes funcionais e anatômicos está disponível, a integração é rápida. Por exemplo, é possível trocar uma parte do *rendering* volumétrico baseado em dados MRI (figura 5.2a), pela imagem MIP dos dados SPECT equivalentes (figura 5.2b).

Stokking [STO 98] também destaca a grande utilização de ferramentas para **extração de planos de corte** em volumes de dados multimodais. Estas ferramen-

tas são descritas e discutidas no capítulo 6.

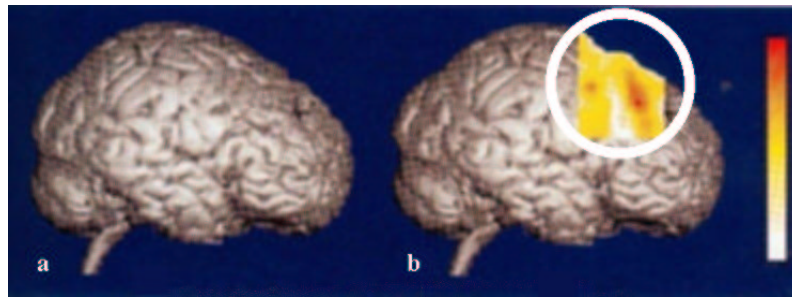


FIGURA 5.2 – Janela Multimodal (extraída de [STO 98])

Técnicas de **mapeamento de textura em uma superfície** integram as informações fazendo o mapeamento das informações funcionais como uma textura em uma superfície ou isosuperfície. Porém, desta maneira, somente uma pequena parte da informação funcional de interesse é apresentada em uma imagem. Payne e Toga [PAY 90] apresentaram um sistema no qual dados funcionais eram mapeados para superfícies arbitrárias extraídas de volumes de dados anatômicos de um encéfalo. Neste caso, é possível pensar em um volume de dados funcionais do encéfalo como uma textura sólida onde variações de cores correspondem a graus de atividades funcionais.

5.3.1 Normal Projection Technique

Para Zuiderveld [ZUI 95] a visualização integrada exige a solução de dois problemas: como atribuir valores quantitativos a pontos relevantes de uma superfície anatômica, e como mapear o valor obtido sobre a superfície, de maneira que a imagem resultante possa ser usada para fins de diagnóstico. Ele tratou estes problemas através do desenvolvimento da técnica *Normal Projection*, que é usada para obter medidas quantitativas de pontos localizados em superfícies anatômicas. Neste método, vetores normais à superfície são obtidos através da aplicação de operadores gradiente a cada ponto da superfície que deve ser visualizada. Esses vetores normais são usados para obter informação quantitativa na superfície anatômica pré-definida ao longo da direção definida pelo vetor inverso ao vetor normal (na outra modalidade). Por exemplo, a figura 5.3 mostra uma situação onde a atividade SPECT pode ser avaliada na trajetória ao longo do vetor com direção oposta ao vetor normal, com a possibilidade de ajustar a profundidade e o passo de amostragem. A técnica de *Normal Projection* processa o caminho ao longo do qual os valores quantitativos serão obtidos, e a visualização integrada da informação quantitativa e da superfície anatômica é realizada através do mapeamento dos valores calculados para cores na superfície.

A atribuição de cor aos dados quantitativos numa superfície é direta se for usado o modelo de cor HSV, o qual permite um melhor mapeamento das sensações visuais causadas pela luz colorida. O componente de tonalidade da cor é atribuído de acordo com o valor quantitativo obtido pela técnica de *Normal Projection*. O componente de saturação pode ser utilizado para indicar a importância do dado. Quando o dado é irrelevante (por exemplo, em regiões anatômicas que não são de interesse), se utiliza baixa saturação; alta saturação é utilizada apenas em áreas que

são relevantes do ponto de vista do diagnóstico.

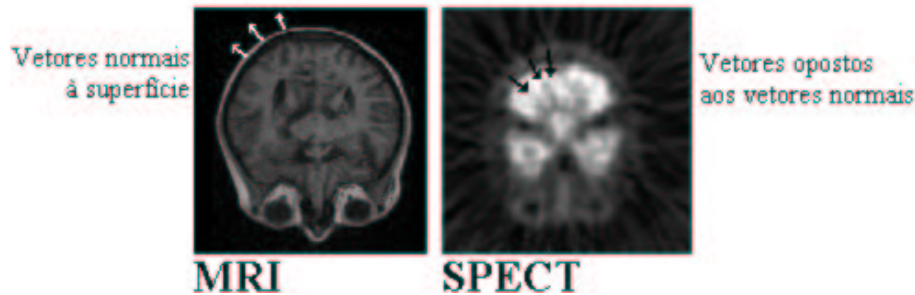


FIGURA 5.3 – Princípio da técnica de *Normal Projection* [ZUI 95]

5.3.2 Normal Fusion Technique

Stokking desenvolveu a técnica de *Normal Fusion* [STO 98] para a visualização integrada de dados funcionais e anatômicos. Este nome foi utilizado porque as informações amostradas ao longo da trajetória inversa do vetor normal à superfície são projetadas em uma superfície. O seu princípio, o mesmo da técnica de *Normal Projection* (seção 5.3.1), é que o gradiente calculado num procedimento padrão de visualização de imagens MRI é usado para avaliar os dados SPECT correspondentes, que são mapeados para cores sobre a superfície extraída.

Qualquer técnica que realize a amostragem da atividade funcional abaixo da superfície e ao longo da direção de visualização introduz artefatos de localização. Neste caso, a informação funcional obtida é uma representação da informação integrada ao longo da direção de visualização, ao invés da atividade local sob a superfície cerebral. A técnica de *Normal Fusion* solucionou esses problemas através do lançamento de um raio secundário ao longo da trajetória inversa do gradiente ou vetor normal (figura 5.3) para amostrar a atividade funcional local e representar com cores a informação amostrada sobre a superfície obtida a partir dos dados anatômicos. Essa técnica foi usada para a visualização simultânea de SPECT, PET ou fMRI com imagens de MRI [STO 98].

Portanto, levando em conta a utilização de dados SPECT e MRI, o valor em um voxel de superfície é calculado a partir de amostras de SPECT obtidas na direção inversa do gradiente calculado na modalidade MRI; as faixas de profundidade e de amostragem são definidas pelo usuário. Apesar de outros métodos terem sido implementados, o valor do voxel da superfície é processado de duas maneiras: através de uma média dos valores amostrados, e através de MIP. A atividade processada é então codificada para uma cor para a visualização da estrutura anatômica obtida a partir de dados MRI. As principais vantagens deste método são a independência da direção de visualização e a precisão na localização anatômica, pois segue a curvatura da superfície para calcular a informação quantitativa local.

Também foram estudados os efeitos causados pelo uso de diferentes profundidades para o processamento das amostras, passos de amostragem e estratégias para extrair novos valores para a superfície na visualização final. A alteração do intervalo em profundidade no qual os dados SPECT são amostrados oferece a possibilidade de revelar regiões de atividade superficiais ou mais internas. Mudanças no passo de amostragem envolvem um balanço entre precisão e velocidade. A estratégia de

avaliação depende da informação na qual o usuário está interessado. Por exemplo, considerando os dados de SPECT e MRI, MIP parece a melhor alternativa para revelar regiões de interesse, mas o método da média dos valores amostrados pode ser mais adequado para exibir o intervalo de intensidades SPECT amostradas.

O esquema de codificação de cor usado na técnica de *Normal Fusion* para combinar informação funcional e anatômica baseia-se no modelo de cor HSV; os componentes de matiz e saturação são usados para a informação funcional obtida pela técnica, e o componente de valor para a informação de *rendering*, como por exemplo, a intensidade da luz refletida da superfície anatômica. A preferência pelo modelo HSV sobre o RGB deve-se ao fato de que o HSV mostrou-se mais apropriado para codificar para cores os dados quantitativos sobre as superfícies e manter separadas as informações funcional e de tonalização, facilitando a interpretação. Além disso, a conversão entre HSV e RGB, que possui um mapeamento conveniente para *hardware*, é simples.

Em sua tese, Stokking [STO 98] menciona que diversos autores mostram a utilidade da codificação de cores na interpretação de imagens funcionais, porém, pouco trabalho tem sido feito para padronizar ou validar o uso de diferentes escalas de cores. Além disso, a maioria dos usuários possuem as suas próprias preferências. Então, para avaliar o modelo HSV para a técnica de *Normal Fusion*, e para testar diferentes estratégias para a sua manipulação, ele adotou uma abordagem que permite a manipulação da tabela de cores. Essa tabela caracteriza-se por possuir poucos pontos de controle com suas respectivas entradas de matiz e de saturação, tornando a manipulação bastante direta. A localização desses pontos de controle, que muda a codificação de cores da informação quantitativa na visualização integrada, também pode ser alterada.

Um exemplo de visualização de dados multimodais usando essa técnica é apresentado na figura 5.4, que mostra os hemisférios direito e esquerdo de um cérebro segmentado de uma imagem MRI. A informação obtida a partir dos dados SPECT é exibida sobre o córtex. Foi utilizada uma codificação de cores para indicar a quantidade de perfusão de sangue no cérebro: de cinza claro para baixa perfusão, passando por amarelo, até chegar em vermelho para alta perfusão. A visualização mostra claramente uma área de perfusão acentuada no hemisfério direito (a área circunda um tumor) [PLU 2002]. Embora a técnica de *Normal Fusion* forneça uma apresentação da perfusão do sangue em relação à anatomia do cérebro, compreensível e valiosa do ponto de vista de diagnóstico, outras opções, tal como *linked cursor*, podem ser desenvolvidas para melhorar o diagnóstico [STO 98].

5.3.3 Spectral Volume Rendering

Noordmans [NOO 97] projetou um sistema que possibilita a análise interativa de imagens 3D de microscópio e inclui a extração de medidas. Foram implementados algoritmos de visualização direta de volume em conjunto com uma série de técnicas de otimização para permitir a interação em tempo real [NOO 97, NOO 2000, NOO 2001]. A técnica de *Spectral Volume Rendering* consiste em modelar as mudanças espectrais que ocorrem quando a luz interage com os materiais em um voxel, que absorve e reflete parte do espectro de iluminação. A luz refletida (ou dispersada) é recebida por um detector de luz, o qual converte o espectro em um sinal RGB, da mesma maneira que o olho humano. Assim, ao invés de simplesmente associar a cada voxel um valor de opacidade e de cor RGB, assume-se que o valor do voxel

representa a densidade do material. No caso da combinação de diversos volumes, por exemplo, multimodais, um voxel conterá diversos materiais.

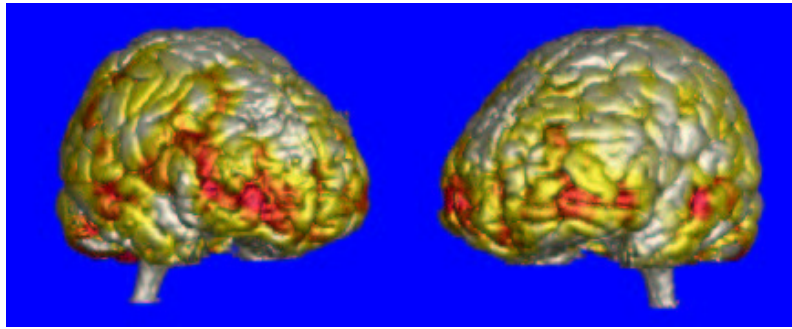


FIGURA 5.4 – Visualização obtida com a técnica de *Normal Fusion* (extraída de [STO 98])

Esta abordagem fisicamente realística tem a vantagem implícita de garantir a ausência de artefatos. Por exemplo, para visualizar um objeto B que está no interior de um objeto A (figura 5.5), se assume que o espectro de iluminação consiste em duas bandas de comprimentos de onda principais, caracterizados por λ_1 e λ_2 . O material do objeto A absorve grande parte da banda espectral ao redor de λ_1 mas não absorve quase nada da banda espectral ao redor de λ_2 . O objeto B absorve bandas espectrais tanto em torno de λ_1 como de λ_2 . O objeto A apenas reflete a banda espectral em torno de λ_1 , enquanto o objeto B reflete somente a banda espectral em torno de λ_2 . O detector recebe ambas as bandas e as converte para um valor RGB. O resultado é a visualização de detalhes da superfície do objeto A ao mesmo tempo em que se visualiza o objeto B através dele [NOO 2001].

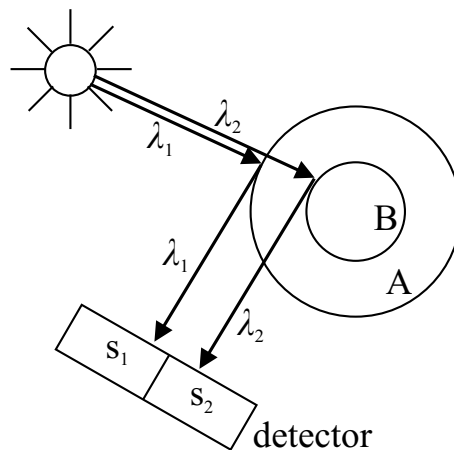


FIGURA 5.5 – Visualização esquemática do *spectral volume rendering* (adaptada de Noordmans et al. [NOO 2001, NOO 2002])

Em um sistema de visualização direta de volume baseado no espectro luminoso existem três momentos nos quais os materiais podem influenciar o espectro de luz: o material pode absorver alguns dos componentes espectrais durante a iluminação; os materiais podem mudar o espectro no próprio processo de reflexão; e os materiais podem absorver alguns dos componentes espectrais refletido por outros materiais.

Assim, três tipos de materiais de reflexão e dois tipos de coloração, enumerados abaixo, foram considerados na investigação a fim de obter a melhor base física para uma visualização direta de volume.

- partícula de reflexão acromática (por exemplo, uma partícula perfeitamente refletora);
- partícula de reflexão elástica (contrastando com o primeiro tipo);
- partícula de reflexão inelástica (por exemplo, fluorescência);
- meio acromático (todos os componentes espectrais são absorvidos na mesma intensidade);
- meio cromático (um componente espectral pode ser mais absorvido do que o outro).

Noordmans [NOO 97] concluiu que reflexão inelástica combinada com absorção acromática fornece potencial para discriminação de material através de cor, melhor visibilidade alcançada através de ferramentas interativas, habilidade de visualizar imagens não segmentadas e transmitância neutra (objetos que não influenciam na cor de outro objeto visto ou iluminado através deles). Portanto, este método oferece características que permitem a visualização de objetos inclusos em conjunto com a superfície do objeto volumétrico que o circunda.

5.3.4 Integração Baseada no Algoritmo de *Ray Casting*

Na técnica de **integração dos dados** para visualização direta de volume [CAI 99], estruturas volumétricas de várias modalidades são integradas em um único conjunto de dados e exibidas posteriormente através de técnicas de *rendering* padrão. Foi considerada a técnica de visualização direta de volume, onde o algoritmo desenvolvido é baseado no *ray casting* clássico [LEV 88] e os dados passam pelo *pipeline* através de três etapas diferentes (figura 5.6):

1. Transformações geométricas;
2. Integração, que inclui percorrer o volume, realizar a amostragem, interpolação, iluminação e acumulação;
3. Mapeamento.

A composição dos dados pode ocorrer em diferentes passos da etapa de integração [CAI 99], resultando, assim, em diferentes *pipelines*, conforme descrito a seguir.

Quando a integração ocorre na etapa da intensidade da imagem (ou pixel) (figura 5.7), uma imagem (intensidade e opacidade) é gerada para cada volume e a imagem final é processada integrando as opacidades e intensidades dos pixels das imagens oriundas de cada volume. Entretanto, existem diferentes opções para integração nesta etapa:

- Somente intensidade (intensidade da imagem resultante é uma composição linear das duas imagens);

- Intensidade com opacidade (utilização de opacidade para melhorar a noção de profundidade);
- Intensidade com opacidade e *Z-Buffer* (utilização do valor do *Z-Buffer* de cada pixel).

A integração nesta etapa é bastante simples e não há necessidade de alterar o algoritmo de visualização, mas os resultados não fornecem uma noção exata de profundidade entre os dois volumes.

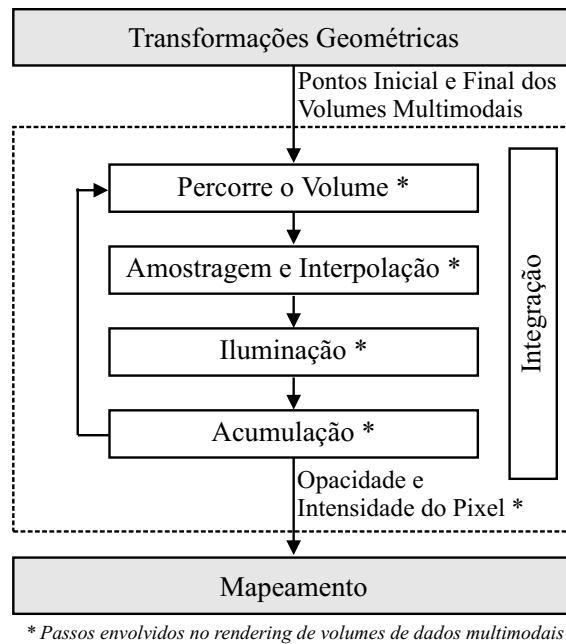


FIGURA 5.6 – *Pipeline* de visualização de dados multimodais (de acordo com Cai e Sakas [CAI 99])

Na integração na etapa de acumulação, um raio é lançado em cada volume para processar as amostras que são, então, combinadas. Em outras palavras, há uma mistura das suas contribuições visuais, passo a passo, ao longo do raio, na etapa de acumulação (figura 5.8). A acumulação envolve opacidade e intensidade, que são obtidas a partir da densidade do voxel por uma função de transferência no modelo de iluminação. Como diferentes volumes possuem intervalo de valores e propriedades físicas específicas, suas funções de transferência também são diferentes. Por outro lado, suas opacidades e intensidades possuem o mesmo intervalo e a mesma propriedade visual depois do mapeamento da função de transferência. Três maneiras de integrar opacidade foram projetadas:

1. Opacidade em um ponto de amostragem é uma das opacidades calculadas a partir dos dois volumes; neste caso, para cada amostra é usada a opacidade processada a partir de um volume A, se esta for diferente de zero, caso contrário, é usada a opacidade processada a partir de um volume B;
2. Opacidade em um ponto de amostragem é a soma do efeito cumulativo causado pelas opacidades dos dois volumes - pode ser aplicada na maioria dos casos de integração se for possível mapear o valor do voxel em opacidade e intensidade;

3. Tanto a abordagem 1 quanto a abordagem 2 podem ser consideradas como operandos ou operadores, que podem ser usados recursivamente entre mais volumes. Por exemplo, um volume segmentado e um volume de CT integrados de acordo com a abordagem 1, podem ser integrados com um terceiro volume (*iso-dose*, codificado para tons de cinza) pela abordagem 2, e assim por diante [CAI 99].

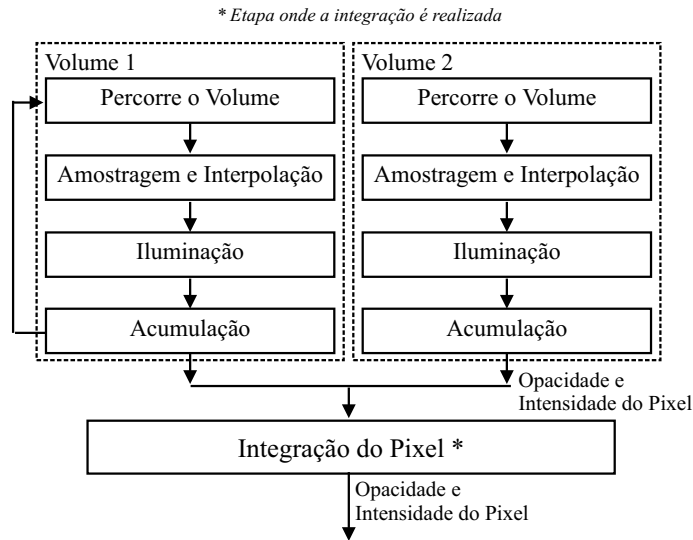


FIGURA 5.7 – Integração na etapa de intensidade da imagem (de acordo com Cai e Sakas [CAI 99])

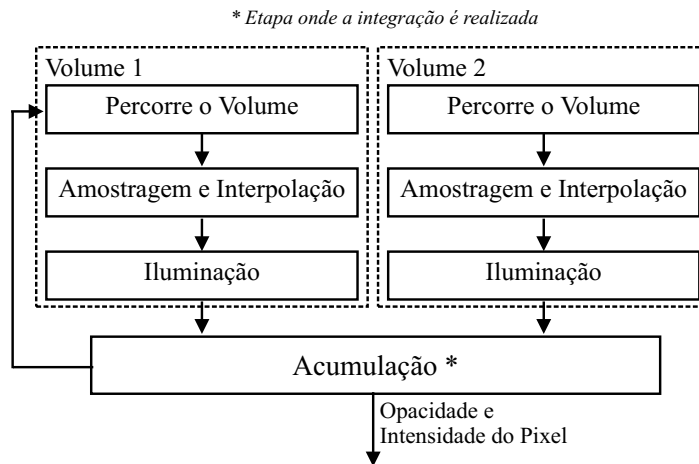


FIGURA 5.8 – Integração na etapa de acumulação (de acordo com Cai e Sakas [CAI 99])

Para integração na etapa do modelo de iluminação (figura 5.9), a opacidade e a intensidade em cada ponto de amostragem são calculadas diretamente a partir do modelo de iluminação, ao invés de serem integradas a partir de diversos valores de opacidade e intensidade. O *pipeline* incluindo acumulação é igual à visualização de uma modalidade. Neste caso, existem duas maneiras diferentes de fazer a integração:

- Coeficiente de atenuação: o coeficiente de atenuação final é considerado como uma combinação de todos os coeficientes individuais dos diversos volumes;
- Propriedade de iluminação: absorção, emissão e sombreamento são usadas, cada uma de um volume diferente [CAI 99].

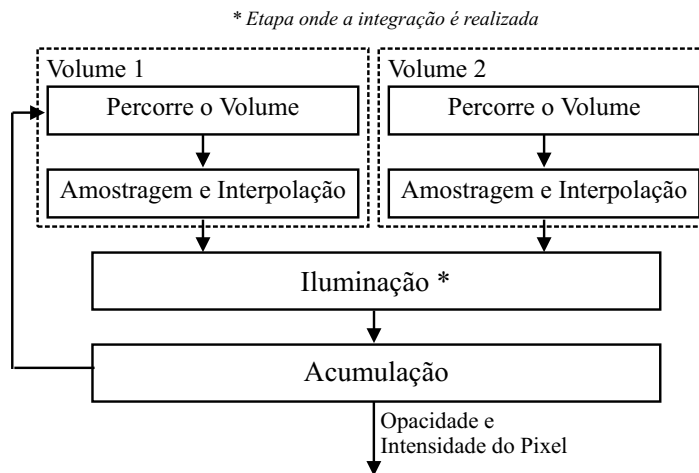


FIGURA 5.9 – Integração na etapa de iluminação (de acordo com Cai e Sakas [CAI 99])

5.4 Comentários Finais

Sistemas que suportam a visualização de volumes de dados multimodais envolvem a integração de várias ferramentas, necessárias para preencher o conjunto de requisitos descritos na seção 5.1. A integração de dados multimodais em uma única imagem exige o desenvolvimento de técnicas de visualização que façam a combinação dos diferentes dados, fornecendo uma representação significativa dos mesmos.

A seção 5.2 apresentou as principais características de alguns sistemas de visualização de imagens multimodais, e as diferentes técnicas e algoritmos para a visualização de dados multimodais foram brevemente descritos na seção 5.3. Entre os sistemas abordados, o ANALYZETM destaca-se como o mais completo, fornecendo várias ferramentas de corte por geometria que permitem explorar o volume de dados. Entretanto, por ser um *software* comercial, detalhes sobre a sua arquitetura não foram encontrados na literatura.

A técnica de *Spectral Volume Rendering* (seção 5.3.3) apresentada por Noordmans [NOO 97] permite visualizar os volumes de dados multimodais e inclui um *slicer* que possibilita a extração de cortes ortogonais e oblíquos, porém carece de recursos para visualização de estruturas internas. Como trabalham com o mapeamento de valores quantitativos para uma superfície anatômica, as técnicas de *Normal Projection* e *Normal Fusion* apresentadas por Zuiderveld [ZUI 95] e Stokking [STO 98], respectivamente, não possuem mecanismos para explorar o interior do volume de dados anatômico. A técnica de *Data Intermixing*, apresentada por Cai e Sakas [CAI 99], aborda apenas a questão de como compor as informações, sem considerar alternativas para visualização de estruturas internas.

6 Visualização de Estruturas Internas em Volumes de Dados Multimodais

Inicialmente, na seção 6.1, é feita uma análise e comparação das técnicas de visualização de estruturas internas, que consistem na utilização e/ou integração de ferramentas de corte e funções de transferência. O algoritmo implementado para a integração das imagens multimodais é descrito na seção 6.2, e as técnicas desenvolvidas para a visualização de estruturas internas são apresentadas na seção 6.3. Na seção 6.4 são discutidos os resultados obtidos, comparando-os com as técnicas encontradas na literatura. O conteúdo das seções 6.2 e 6.3, e de parte da 6.4 está sendo publicado em [MAN 2002].

6.1 Alternativas para a Visualização de Estruturas Internas

Conforme descrito na seção 2.3, para visualizar estruturas internas são identificados dois tipos principais de abordagens:

- controle da transparência do voxel, e
- remoção dos voxels que estão fora da região de interesse, que pode ser realizada através de ferramentas de corte ou segmentação.

Sendo assim, tanto as ferramentas de corte como as funções de transferência são usadas, separadas ou em conjunto, para “ver através” de um volume de dados. A grande diferença entre estas técnicas está no fato de que o corte é usado para remoção, ou seleção, de uma região do volume, e as funções de transferência são usadas para atribuição de cores e diferentes níveis de transparência, característica esta relacionada com a percepção humana.

Neste trabalho, o estudo das técnicas de corte foi realizado considerando a expressão genérica do algoritmo de *ray casting* (equação 2.2), apresentada na seção 2.2. Assim, no caso da remoção dos voxels, ferramentas de corte foram descritas a partir de um esquema no qual cada uma pode ser representada como um conjunto diferente de parâmetros que determinam o intervalo de amostragem (capítulo 3). Também foi demonstrado como algumas técnicas de visualização direta de volume com mecanismos sofisticados podem se enquadrar no esquema apresentado, mesmo quando, por exemplo, o conceito de “corte” não está explícito no método, como no caso do CVR [MUL 2000].

Considerando o controle da transparência do voxel (capítulo 4), as diversas abordagens para a especificação das funções de transferência foram representadas como um conjunto de parâmetros usados para determinar como as amostras são processadas, com o intuito de evidenciar a importância e a diversidade de funções. As abordagens existentes podem ser enquadradas nas seguintes categorias: tentativa e erro; baseada na detecção de características espaciais; centralizada na imagem; centralizada no dado; e novos domínios e tipos de interação. Segundo Lorenzen [PFI 2001] e Tenginakai et al. [TEN 2001], as mais promissoras são as que pertencem à categoria centralizada no dado.

Uma alternativa para visualizar estruturas internas é combinar a remoção dos voxels com o controle da sua transparência. Em outras palavras, podem ser aplicadas diferentes funções de transferência no processamento das amostras realizado entre a e b . Além disso, imagens similares são obtidas através da utilização de um plano de corte ou quando se atribui opacidade zero a todos os voxels que estiverem na região do volume que está sendo recortada. A diferença é que, na segunda situação, o valor de transparência é determinado pela posição do voxel e não pelo seu conteúdo. Além disso, o tempo de geração da imagem pode ser maior, uma vez que há um processamento inicial das amostras, mesmo que a opacidade seja zero.

Outro tipo de combinação pode ser feita quando existe a possibilidade de se utilizar várias tabelas de classificação em uma mesma visualização. Por exemplo, quando se utiliza um plano de corte, uma tabela de cor e opacidade pode ser usada para os voxels ao longo do raio que se encontram no subespaço anterior ao plano; outra tabela pode ser utilizada para os voxels que estão exatamente sobre o plano; e outra tabela, ainda, pode ser usada para os voxels que estão na região posterior ao plano [SIL 2000]. A figura 6.1 ilustra a visualização do volume de teste com a utilização de um paralelepípedo de corte (figura 6.1b) e com a utilização de um valor de opacidade menor, mas diferente de zero, para os voxels que estão na região do paralelepípedo de corte (figura 6.1a). Assim, o valor de opacidade não é atribuído de acordo com o valor de intensidade do voxel, mas sim, de acordo com a posição do voxel no volume de dados, fazendo uma associação da função de transferência à geometria, e não ao conteúdo, de maneira semelhante à abordagem apresentada por Mullick et al. [MUL 2000].

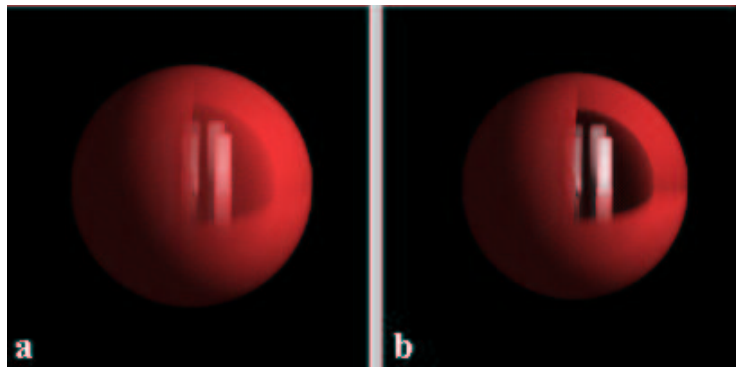


FIGURA 6.1 – Combinação de técnicas de corte e funções de transferência

Além da possibilidade de combinar a utilização de ferramentas de corte com diferentes funções de transferência, estruturas internas de um volume podem ser visualizadas usando cada uma das ferramentas apresentadas no capítulo 3 e nas seções 4.2, 4.3 e 4.4. Na figura 6.1b, ao invés da aplicação de um paralelepípedo de corte, poderia ter sido usada uma função de transferência que resultasse num valor de opacidade zero para os voxels que estivessem na região englobada pelo paralelepípedo de corte, de maneira análoga ao procedimento realizado para gerar a figura 6.1a.

Embora não seja utilizado para dados multimodais, o trabalho desenvolvido por Hauser et al. [HAU 2000, HAU 2001] ilustra um esquema diferente para a visualização de estruturas internas, chamado *two-level volume rendering* (2IVR). Neste caso, é possível usar seletivamente diversas técnicas de *rendering*, tais como visua-

lização direta de volume, visualização através de superfície ou MIP para diferentes objetos contidos em um mesmo volume de dados que está sendo visualizado. Estes objetos são identificados usando técnicas de visualização diferentes e os resultados são combinados em uma etapa de fusão que ocorre, geralmente, na composição. Assim, por exemplo, MIP (também possível de ser enquadrada como uma técnica de corte, seção 3.4) pode ser combinada com visualização através de superfície e visualização direta de volume, que são adequadas, respectivamente, para a visualização da pele, dos vasos sanguíneos e dos ossos, como ilustra a figura 6.2.

É fácil observar que a combinação de técnicas de *rendering* é uma alternativa viável para a exibição de volumes de dados multimodais.



FIGURA 6.2 – Combinação de técnicas de *rendering* usando a abordagem 2IVR (extraída de [HAU 2002])

Um experimento para a visualização de estruturas internas em dados monomodais realizado por Zuiderveld [ZUI 95] (seção 5.3.1) foi a combinação da técnica de aceleração do algoritmo de *ray casting*, *Ray Acceleration by Distance Coding* (RADC), com a técnica de *Closest Vessel Projection* (CVP) descrita na seção 3.3. O seu principal objetivo foi permitir a visualização de informações vasculares ao mesmo tempo que diminuía o tempo de geração das imagens. O método RADC assume que os dados volumétricos foram corretamente segmentados e classificados durante uma etapa de pré-processamento e cria um volume binário onde é atribuído o valor 1 (um) para todos os voxels de interesse, e 0 (zero) caso contrário. Em seguida, é processado um volume que contém, para cada voxel vazio, a distância para o voxel de interesse mais próximo. Distâncias previamente processadas podem ser usadas para “pular” as partes vazias do volume. O CVP acrescenta percepção de profundidade à imagem, uma vez que o processamento das amostras ao longo do raio termina quando este deixa o primeiro vaso sanguíneo encontrado.

A técnica de *Confocal Volume Rendering* (CVR), também descrita na seção 3.3, permite que o usuário visualize as estruturas internas de dados volumétricos através do controle de parâmetros físicos (ou medidas). O CVR não necessita de segmentação, tem sido usado para visualizar apenas um volume, e os parâmetros são fornecidos pelo usuário na forma numérica [MUL 2000, SUM 2001].

Da análise das alternativas acima examinadas conclui-se que a abordagem unificada de visualização de estruturas internas, descrita no capítulo 3, fornece uma base sólida para a investigação de volumes de dados multimodais. Exemplos da utilização de ferramentas (definidas segundo esta abordagem) para inspecionar os

volumes de dados multimodais são apresentados na seção 6.3.

6.2 Abordagem Adotada para Visualização de Dados Multimodais

O algoritmo desenvolvido para a visualização de volumes de dados multimodais, que considera que os volumes já estão registrados, é baseado na técnica de integração dos dados para visualização direta de volume [CAI 99] (seção 5.3.4). O algoritmo de *ray casting* foi estendido para permitir a integração dos dados na etapa de acumulação do *pipeline* de visualização de dados multimodais (figura 5.8). Também foi adotada a abordagem recursiva introduzida por Levoy [LEV 90] para implementação da composição na ordem *front-to-back* (equações 2.5 e 2.6).

Para a visualização de dois volumes, V_1 e V_2 , previamente registrados, dois raios são lançados, um para cada volume. Conseqüentemente, duas amostras $V_1(\mathbf{x}_{s_i})$ e $V_2(\mathbf{x}_{s_i})$ são simultaneamente consideradas para o cálculo da cor e da opacidade em cada posição amostrada s_i .

Se os volumes possuem a mesma dimensão, ou seja, a mesma resolução espacial para x , y e z , a posição real \mathbf{x}_{s_i} dentro do espaço abrangido pelo volume é igual em V_1 e V_2 . Porém, se os volumes possuem dimensões diferentes, uma mesma amostra s_i corresponde a posições diferentes nos volumes V_1 e V_2 . Neste caso, é necessário verificar a dimensão do maior volume para determinar o número de amostras que serão processadas de acordo com o passo de amostragem. Assim, por exemplo, se o volume V_1 é maior, primeiro é especificada a posição $\mathbf{x}1_{s_i}$ da amostra neste volume. Depois, é processada a amostra $\mathbf{x}2_{s_i}$ no volume V_2 , de tal maneira que $\mathbf{x}1_{s_i}$ e $\mathbf{x}2_{s_i}$ permitem acessar as células que correspondem à mesma posição espacial na região representada pelo volume. Entretanto, um processamento especial é necessário se as imagens de cada modalidade não foram adquiridas englobando a mesma região do espaço. Por exemplo: em um exame de MRI, são adquiridas imagens de todo encéfalo, resultando em um volume de $256 \times 256 \times 128$; em um exame SPECT, para o mesmo paciente, são adquiridas imagens apenas para a metade superior do encéfalo, resultando em um volume $128 \times 128 \times 32$. Neste caso, além das imagens SPECT terem uma dimensão menor, não foram adquiridas para todo encéfalo. Portanto, neste exemplo, a metade inferior do encéfalo não contém informações funcionais, de maneira que as amostras são processadas apenas no volume MRI.

Resumindo, o número de amostras é determinado a partir da intersecção do raio com o volume de maior dimensão e do passo de amostragem, e o processamento pode ser feito de duas maneiras:

- Se os dados correspondem a uma mesma região do espaço, apesar de possuírem dimensões diferentes, cada posição s_i é mapeada para $\mathbf{x}1_{s_i}$ e $\mathbf{x}2_{s_i}$, que correspondem à amostragem da mesma posição no espaço representado pelos volumes V_1 e V_2 , respectivamente;
- Se, por exemplo, os dados do volume V_2 correspondem à metade da região ocupada pelo volume V_1 , cada posição s_i é mapeada para $\mathbf{x}1_{s_i}$ e para $\mathbf{x}2_{s_i}$ somente quando a amostra for processada dentro da região que também faz parte de V_2 .

Aqui, assume-se que V_1 representa a anatomia ou forma de um órgão ou objeto, sendo usado, então, para determinar o valor de opacidade, e V_2 representa os dados funcionais, sendo usado para obtenção do valor de cor. Desta maneira, é realizada a integração dos dados na etapa de acumulação [CAI 99]. As equações 2.5 e 2.6 são rescritas a seguir (equações 6.1 e 6.2) para enfatizar que as funções de transferência de cor e opacidade são baseadas em dados de diferentes volumes. No caso dos volumes não corresponderem à mesma região do espaço, quando não houverem informações sobre uma das modalidades, em geral dados funcionais, apenas os dados de um volume são considerados, neste caso, os dados anatômicos.

$$C(\mathbf{x}_{s_i}) = \begin{cases} T_{C_\lambda}(\mathbf{x}2_{s_i})T_\alpha(\mathbf{x}1_{s_i}), & i = 0 \\ C(\mathbf{x}_{s_{i-1}}) + T_{C_\lambda}(\mathbf{x}2_{s_i})T_\alpha(\mathbf{x}1_{s_i})(1 - \alpha(\mathbf{x}_{s_{i-1}})), & i > 0 \end{cases} \quad (6.1)$$

$$\alpha(\mathbf{x}_{s_i}) = \begin{cases} T_\alpha(\mathbf{x}1_{s_i}), & i = 0 \\ \alpha(\mathbf{x}_{s_{i-1}}) + T_\alpha(\mathbf{x}1_{s_i})(1 - \alpha(\mathbf{x}_{s_{i-1}})), & i > 0 \end{cases} \quad (6.2)$$

Para exemplificar a visualização de volumes de dados multimodais foram utilizados volumes de dados sintetizados (anexo): as esferas apresentadas nas figuras 6.3a e 6.3b, e um cubo, apresentado na 6.3c. O volume de dados que contém o cubo tem as mesmas dimensões do volume da esfera, porém, seu conteúdo consiste em várias camadas com densidades diferentes. Para ilustrar a visualização integrada nesta seção, e para destacar o resultado obtido com a utilização das ferramentas de corte na seção 6.3, foi utilizada apenas uma função de transferência de opacidade linear e uma tabela de cor.

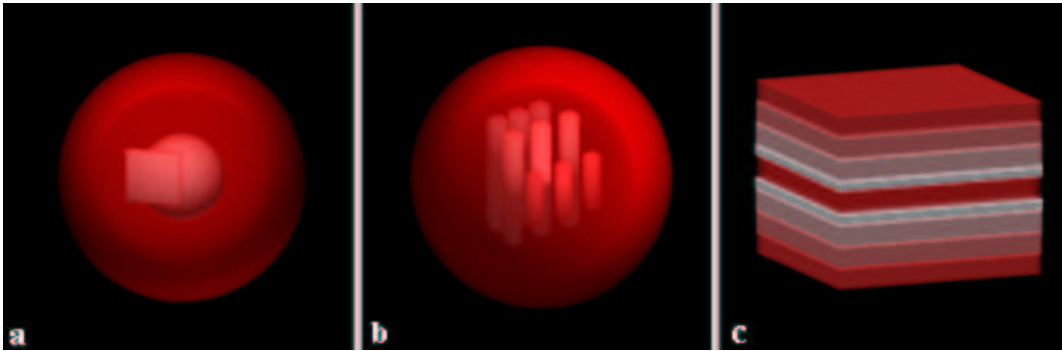


FIGURA 6.3 – Volumes sintetizados usados para exemplificar a visualização de volumes de dados multimodais

A figura 6.4 ilustra a visualização integrada dos volumes de dados sintetizados, considerando que estes possuem as mesmas dimensões e equivalem à mesma região do espaço ($x1_{s_i} = x2_{s_i}$). Na figura 6.4a, a esfera corresponde a V_1 e determina a opacidade, representando a anatomia, e o cubo corresponde a V_2 e é usado para especificar a cor, representando dados funcionais. Na figura 6.4b os volumes são invertidos, fazendo com que V_1 determine a cor e V_2 a opacidade. Assim, o cubo é colorido de acordo com a esfera, conforme mostra a imagem da esquerda da figura 6.4b, onde é usado um plano de corte para exibir o interior do volume. Uma função de transferência de opacidade linear é usada para especificar a opacidade em ambos os volumes. A função de transferência de cor é dada por diferentes tabelas de cor, e as amostras são processadas usando uma interpolação trilinear.

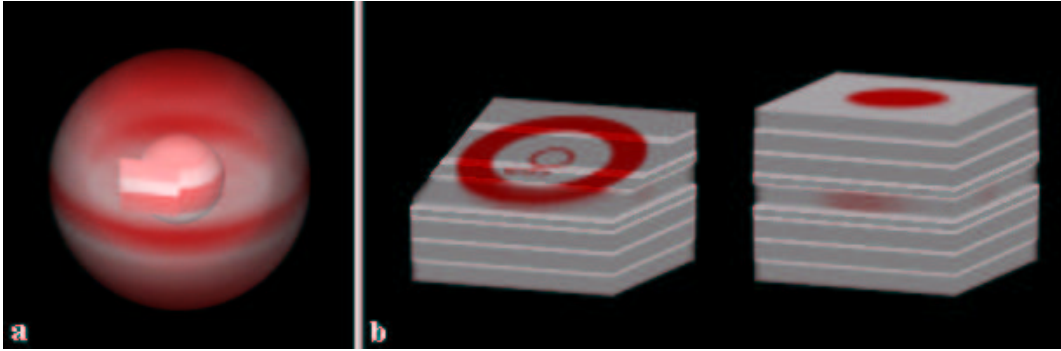


FIGURA 6.4 – Visualização integrada dos volumes sintetizados

Um exemplo de imagem gerada, considerando que o volume sintetizado do cubo corresponde à metade do tamanho do volume da esfera, é exibido na figura 6.5. Neste caso, apesar dos volumes possuírem as mesmas dimensões, assume-se que a região do espaço que o volume do cubo representa é diferente da região do espaço que o volume da esfera representa, de tal forma que $x1_{s_i} \neq x2_{s_i}$. Sendo assim, a metade superior da esfera é colorida de acordo com o cubo, e a metade inferior é exibida em tons de cinza.

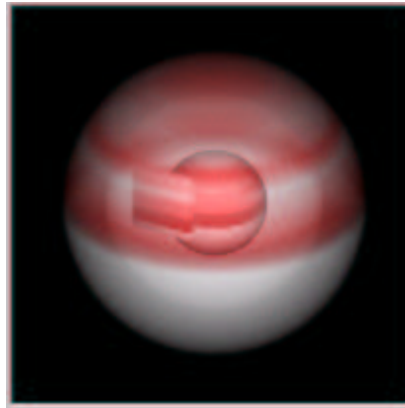


FIGURA 6.5 – Visualização integrada considerando que os volumes sintetizados possuem tamanhos diferentes

6.3 Abordagem para a Visualização de Estruturas Internas em Volumes de Dados Multimodais

Nesta seção, é descrita uma nova abordagem para a visualização de estruturas internas em volumes de dados multimodais, que consiste numa generalização do algoritmo de *ray casting*. A abordagem unificada apresentada para as ferramentas de corte (capítulo 3) e o algoritmo de visualização integrada apresentado na seção 6.2 foram combinados, permitindo, assim, a integração dos volumes de diferentes modalidades e a incorporação de várias ferramentas de corte como um mecanismo de ver através dos volumes. O principal objetivo é fornecer um conjunto de ferramentas genéricas e poderosas que permitam uma fácil inspeção das estruturas internas.

A integração destas ferramentas no *framework* do *ray casting* mostrou-se direta e simples, correspondendo à manipulação de alguns parâmetros.

No caso de corte por geometria, a diferença para a utilização de volumes multimodais está na etapa de composição de cor e opacidade. Considerando a utilização de ferramentas de corte por geometria e conteúdo, ou apenas por conteúdo, com volumes de dados multimodais, surgem novas formas de visualização e especificação dos parâmetros. Uma característica importante e poderosa é a possibilidade de considerar o conteúdo de qualquer um dos volumes para realizar o corte, obtendo, assim, resultados diferentes.

De acordo com a abordagem unificada apresentada no capítulo 3, uma ferramenta de corte é escolhida de acordo com os parâmetros de entrada U_a e U_b , as diferentes implementações das funções f_a e f_b , os resultados esperados e os objetivos do usuário. No caso de volumes de dados multimodais, uma outra “variável” influencia a escolha da ferramenta de corte: quando o corte é baseado na propriedade do voxel, qualquer um dos volumes pode ser usado, resultando em diferentes imagens. Sendo assim, a tabela 3.2 é aqui estendida para mostrar as possíveis combinações que podem ser feitas do ponto de vista do usuário, do tipo de operação e da especificação de a e b .

A tabela 6.1 apresenta algumas possíveis combinações dos valores atribuídos para a e b , considerando que V_1 é o volume usado para determinar a anatomia, representada pelo valor de opacidade, e V_2 é o volume usado para especificar a cor. Da mesma forma que na tabela 3.2, a tabela 6.1 mostra que, teoricamente, existem situações não encontradas na literatura (1..8), mas que são identificadas dentro da abordagem apresentada.

TABELA 6.1 – Diferentes combinações de acordo com a especificação de a e b em volumes de dados multimodais. Os números 1 a 8 representam alternativas não encontradas na literatura

a e b	Corte por Exclusão	Corte por Inclusão
$a \neq b$: geometria	Um ou dois planos de corte, superfícies de corte não-planares, objeto volumétrico convexo de corte	Dois planos de corte, volume de interesse, superfícies de corte não-planares
$a = b$: geometria	1	Seções ortogonais, oblíquas e não-planares
$a = b$: conteúdo (V_1 ou V_2)	2	MIP
$a \neq b$: conteúdo (V_1 ou V_2)	3	4
a : geometria b : conteúdo (V_2)	5	6
a : geometria b : conteúdo (V_1)	7	CVP
a : geometria e conteúdo (V_1) b : geometria	8	CVR

Através da análise da tabela 6.1, observa-se que novas ferramentas podem ser especificadas para corte por conteúdo, pois qualquer um dos volumes pode ser

usado para “cortar”. Na seqüência é feita uma descrição das técnicas de corte para volumes de dados multimodais. Para a geração das imagens que exemplificam as funcionalidades destas técnicas, foi sempre utilizada a mesma tabela de cor, ilustrada na figura 6.6, e uma função de transferência de opacidade linear, com I_l variando entre 40 e 80, de acordo com a equação 4.4:

$$T_\alpha(\mathbf{x}_{s_i}) = \begin{cases} 0, & V(\mathbf{x}_{s_i}) < I_l \\ (V(\mathbf{x}_{s_i}) - I_l)/(I_m - I_l), & V(\mathbf{x}_{s_i}) \geq I_l \end{cases}$$

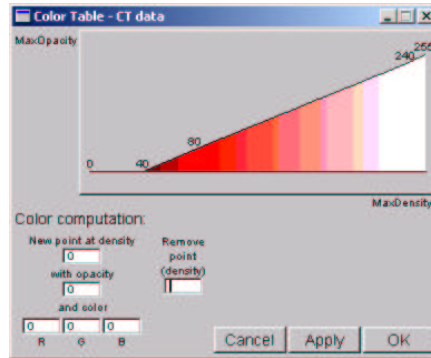


FIGURA 6.6 – Tabela de cor usada na geração das imagens obtidas pela integração dos volumes sintetizados

Corte por Geometria

A figura 6.7 ilustra a aplicação de ferramentas de corte por geometria em conjunto com a visualização integrada. Neste exemplo, o volume representado pela esfera (V_1) foi usado para determinar a opacidade, e o volume representado pelo cubo (V_2) foi usado para especificar a cor. Assim, é possível observar as estruturas internas com a cor identificando seus valores funcionais.

Para ilustrar as situações onde $a \neq b$, foram utilizados planos de corte e um objeto volumétrico convexo de corte. A figura 6.7a mostra o resultado da aplicação de um plano de corte por exclusão ($f_a = c_1$ e $f_b = r_2$). As figuras 6.7b e 6.7c consistem, respectivamente, na aplicação de dois planos de corte por exclusão e por inclusão (em ambos $f_a = c_1$ e $f_b = c_2$).

Um cubo também foi usado como um objeto volumétrico de corte para selecionar e remover voxels localizados na sua intersecção com os volumes multimodais. A figura 6.7f mostra um exemplo onde o cubo é usado como um objeto volumétrico de corte por exclusão. Neste caso, $f_a = r_1$ e $f_b = r_2$, quando o raio não intersecciona o cubo, e $f_a = c_2$ e $f_b = r_2$, caso contrário. A figura 6.7e ilustra a seleção de um volume de interesse a partir deste mesmo cubo (corte por inclusão), situação em que $f_a = c_1$ e $f_b = c_2$.

Um exemplo onde $a = b$ é apresentado na figura 6.7d, que contém uma seção oblíqua ($f_b = f_a = c_1$). Considerando a situação 1 da tabela 6.1, quando $a = b$ e o corte é realizado por exclusão, a imagem resultante não seria útil, uma vez que apenas uma seção seria eliminada do processamento dos volumes.

Corte por Conteúdo

Conforme descrito na seção 3.4, a técnica de MIP pode ser abordada como uma ferramenta de corte por inclusão, onde, ao longo do raio, é considerada apenas

a amostra de maior intensidade. A figura 6.8a exemplifica a sua utilização, sendo que a equação $f_a = f_b = \max(\mathcal{R}_p)$ foi aplicada considerando que o volume da esfera representa a anatomia (V_1). Uma imagem gerada considerando a situação 2 da tabela 6.1 não parece útil, pois, neste caso, por exemplo, apenas uma estrutura que deveria ser bem determinada por um *threshold* não seria considerada no processamento.

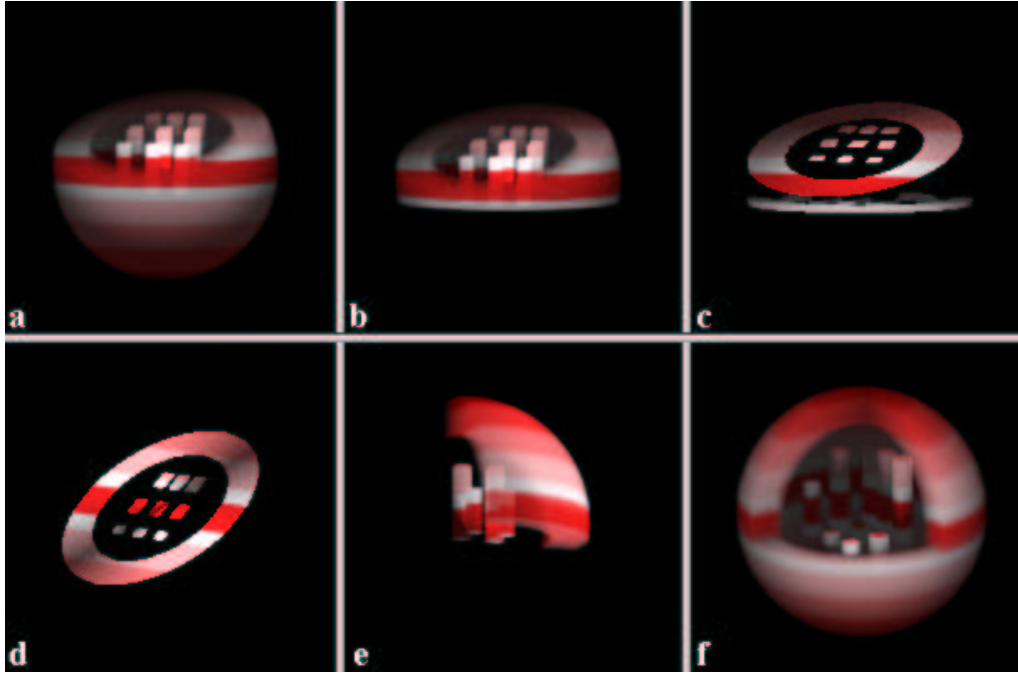


FIGURA 6.7 – Exemplos de corte por geometria em volumes de dados multimodais: (a) um e (b) dois planos de corte por exclusão, (c) dois planos de corte por inclusão, (d) uma seção oblíqua, (e) corte utilizando um cubo por inclusão e (f) por exclusão

As figuras 6.8b e 6.8c ilustram as situações 3 e 4 da tabela 6.1, onde $a \neq b$ e o corte é realizado, respectivamente, por inclusão e por exclusão. No corte por inclusão as equações utilizadas foram $f_a = \text{first}(R_p, v)$, com $v = 100$, e $f_b = a + \text{first}(R_p, v)$, com $v = 255$. Em outras palavras, o processamento das amostras em cada raio começou quando a primeira amostra com intensidade 100 foi encontrada, e terminou quando foi processada a primeira amostra com valor de intensidade 255. Um processamento inverso foi realizado para o corte por exclusão, de tal forma que as duas imagens se complementam. O volume utilizado para análise do conteúdo foi a esfera (V_1).

Corte por Geometria e Conteúdo

Os cortes por exclusão e por inclusão quando o parâmetro a é definido por geometria e o parâmetro b é definido pelo conteúdo do volume de dados, são exemplificados nas figuras 6.8d e 6.8e, respectivamente. Estes exemplos ilustram as situações 5 e 6 da tabela 6.1, com $f_a = r_1$ e $f_b = \text{first}(R_p, v)$, com $v = 200$, e com o volume do cubo (V_2) usado para realizar o corte.

Closest Vessel Projection [ZUI 95] é uma técnica que foi aqui estendida para utilização com volumes de dados multimodais. Foram realizados alguns testes, como

ilustra a figura 6.9, onde apenas as informações do volume de dados anatômico (V_1), que neste caso é a esfera com os paralelepípedos, foram utilizadas para especificar até onde o raio deveria ser processado. Assim, $f_a = r_1$ e $f_b = \text{second}(\mathcal{R}_p, v)$, sendo v o valor de densidade dos paralelepípedos. O detalhe ampliado na figura 6.9a mostra que o uso da técnica de CVP possibilita uma melhor definição das estruturas, permitindo verificar a relação de profundidade entre elas.

A situação 7 da tabela 6.1 corresponde ao complemento da técnica de CVP. Assim, por exemplo, todas as amostras que estiverem entre $f_a = r_1$ e $f_b = \text{second}(\mathcal{R}_p, v)$ não são processadas.

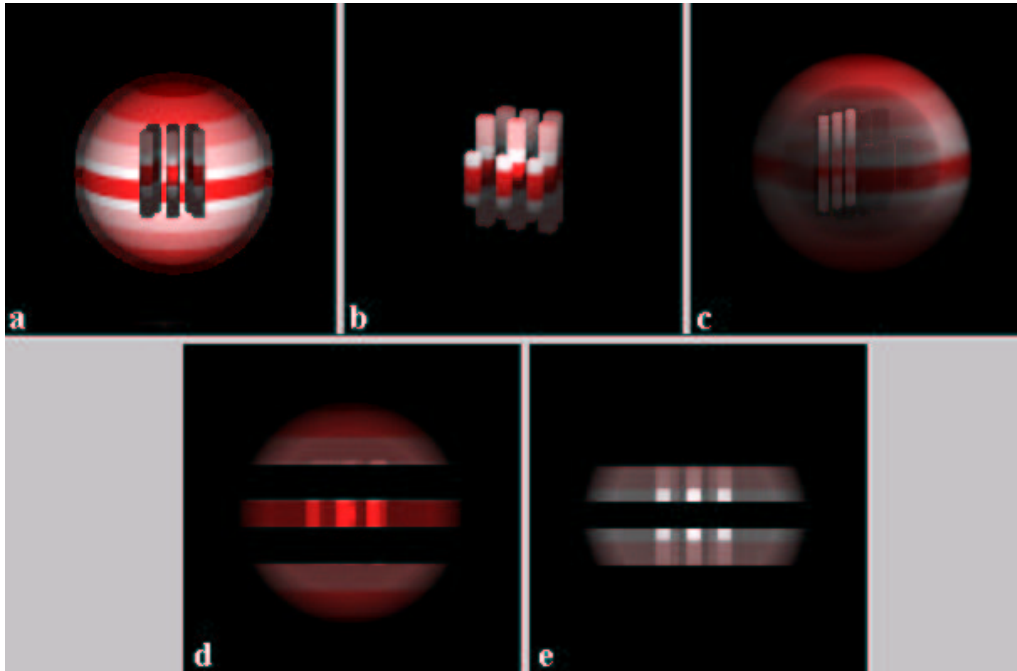


FIGURA 6.8 – Exemplos de corte (a,b,c) por conteúdo e por (d,e) geometria e conteúdo em volumes de dados multimodais

Confocal Volume Rendering [MUL 2000] aplicada aos volumes de dados sintetizados está exemplificada na figura 6.10. Neste caso, somente o volume da esfera está sendo usado para especificar os parâmetros S-Band e T-Band. O cubo é usado para a atribuição de cor aos voxels. A diferença entre as figuras 6.10a e 6.10b está apenas nos valores dos parâmetros S-Band e T-Band, de maneira que diferentes estruturas que estão no interior do volume podem ser visualizadas na imagem final.

Considerando o algoritmo CVR original [MUL 2000], foram realizadas duas extensões quando este foi incorporado ao contexto da abordagem apresentada:

1. Possibilidade de utilização de volumes de dados multimodais;
2. Manipulação interativa dos parâmetros S-Band e T-Band, bem como da escolha da função de escala.

A extensão do CVR para processar volumes multimodais consiste, basicamente, na especificação dos parâmetros S-Band e T-Band considerando um volume, por exemplo V_1 , e na especificação das cores na etapa de classificação de acordo com outro volume, no caso V_2 . A imagem final é gerada de acordo com o algoritmo descrito na seção 6.2. Apesar de nos exemplos aqui apresentados ter sido usado apenas

o volume V_1 , que caracteriza a anatomia, para determinar S-Band e T-Band, nada impede que o volume V_2 seja usado no seu lugar.

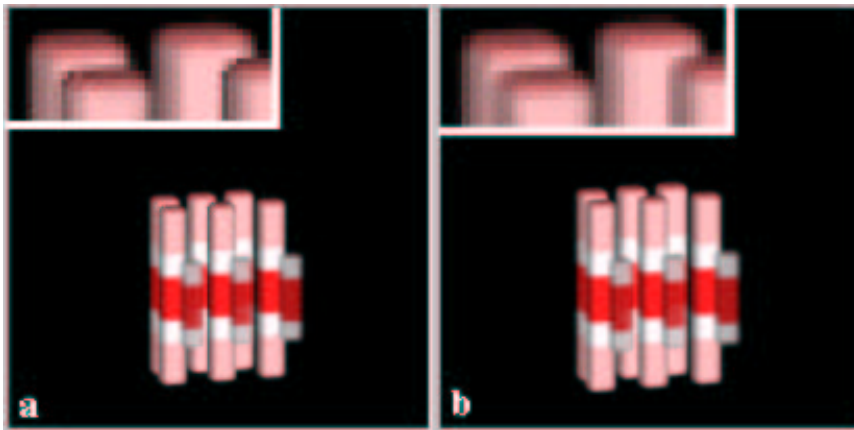


FIGURA 6.9 – Visualização dos volumes multimodais (a) com a técnica de CVP e (b) com o algoritmo de *ray casting* convencional

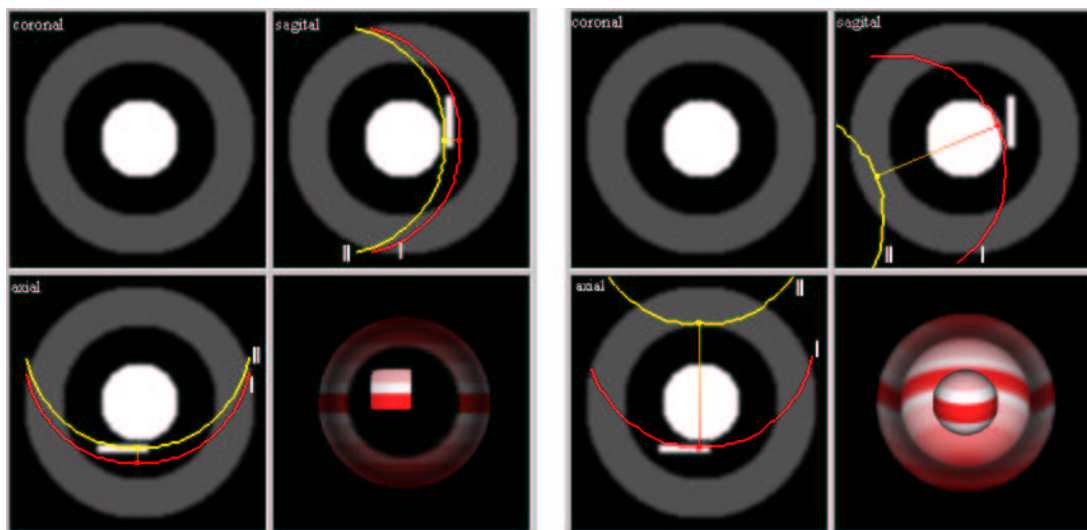


FIGURA 6.10 – Visualização dos volumes multimodais através da técnica de CVR

Na figura 6.10 foi selecionada uma função de transferência linear de acordo com o valor de opacidade do voxel, e sem considerar a sua profundidade, conforme sugerido pelo autor [MUL 2000]. Já no caso dos parâmetros S-Band e T-Band, ao invés do usuário fornecer valores numéricos em mm, é necessário apenas movimentar, respectivamente, os pontos centrais das linhas I e II que correspondem a estes valores e estão representados nos cortes axial e sagital do volume. A partir dos pontos S-Band e T-Band, que são exibidos no momento que esta ferramenta é ativada, e alterados interativamente pelo usuário, são desenhadas as linhas I e II ilustradas na figura 6.10.

A linha I é definida a partir dos seguintes parâmetros: ponto S-Band, posição de observação, alvo (centro do objeto) e número da fatia selecionada no respectivo plano (axial ou sagital). Assim, ao longo da fatia 2D são lançados raios paralelos ao

vetor que vai do observador ao alvo, e, para cada um deles, primeiro é identificada a superfície do objeto em estudo, e depois é definido o ponto no qual começa a região de interesse. Depois que toda a fatia é percorrida, os pontos processados são conectados para formar a linha I. A linha II é processada de forma análoga, sendo que se parte do ponto onde começa a região de interesse para, usando o valor de T-Band, definir onde esta região termina. Desta forma, as linhas mostram, no plano, a porção do volume que será visualizada. Cada vez que a posição de observação ou a fatia selecionada é alterada, as linhas são automaticamente re-exibidas. Além disso, todo o processamento é realizado no espaço do volume, sendo, portanto, completamente independente da interface do sistema, o que torna o método portátil.

Resumindo, os pontos S-Band e T-Band correspondem a posições no espaço do volume que são independentes da posição de observação. Estes pontos representam constantes de deslocamento que identificam a porção do volume ao longo do raio, onde as amostras são processadas e podem ser alteradas interativamente. As vantagens em relação ao algoritmo CVR original é que não há a necessidade de saber a profundidade em mm, além de possibilitar, através dos planos axial e sagital, uma fácil identificação da porção do volume que está sendo considerada no processamento.

Um complemento da técnica de CVR corresponde a situação 8 da tabela 6.1. Neste caso, são eliminadas todas as amostras que estiverem entre $f_a = first(\mathcal{R}_p, v) + d_1$ e $f_b = a + d_2$.

Outras combinações de parâmetros não contempladas na tabela 6.1 também podem ser realizadas. Por exemplo, de maneira análoga à figura 3.15, CVR poderia ser alterado para que as amostras fossem processadas entre $f_a = first(\mathcal{R}_p, v)$, com $v = 230$ e $f_b = r2$. A figura 6.11 ilustra como seria a visualização se amostras fossem processadas somente depois de encontrar a primeira amostra com valor 230 ao longo da raio, considerando o volume da esfera.

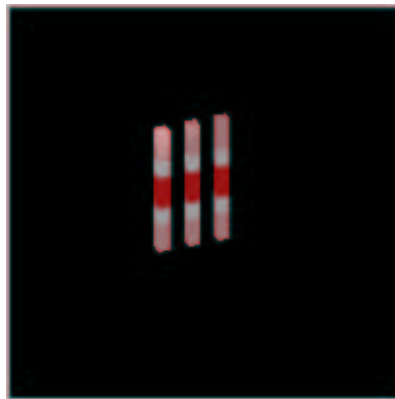


FIGURA 6.11 – Corte por inclusão (a=conteúdo, b=geometria)

Apesar dos exemplos apresentados até aqui considerarem que um volume anatômico é usado para especificar a opacidade e outro, funcional, para especificar a cor, nada impede a utilização de dois volumes anatômicos ou que um dos volumes seja usado para especificar cor e opacidade e o outro seja usado apenas para determinar a e b . Por exemplo, na figura 6.12a foram usados os dois volumes da esfera (anatômicos) para gerar a imagem, sendo que um foi usado para especificar a cor e o outro a opacidade. Na figura 6.12b foi usado o volume sintetizado da esfera

para definir a cor e a opacidade, e o volume do cubo foi usado apenas para fazer o corte por conteúdo. Esta figura é análoga à figura 6.8d, que consiste em um corte por exclusão com $f_a = r1$ e $f_b = first(R_p, v)$, onde $v = 200$.

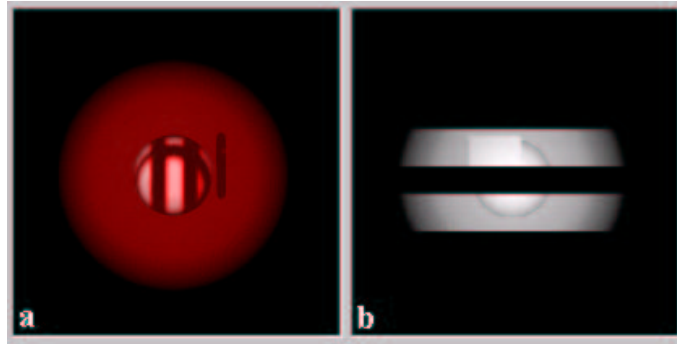


FIGURA 6.12 – (a) Um volume da esfera usado para definir a cor e o outro a opacidade, e (b) cor e opacidade definidas pelo volume da esfera e corte por conteúdo especificado através do volume do cubo

6.4 Discussão

A integração de volumes de dados multimodais em uma única imagem requer o desenvolvimento de técnicas de visualização aprimoradas que combinem dados de natureza diferente e que forneçam representações significativas. A abordagem apresentada na seção 6.2 foi desenvolvida tomando como base a técnica de *Data Intermixing*, com a integração realizada na etapa de acumulação. Assim, obtém-se uma representação mais precisa, gerando uma imagem de melhor qualidade, do que na integração realizada na etapa de imagem. Além disso, apesar do tempo para geração das imagens ser maior, a implementação é mais simples, sem perder em qualidade, do que com a integração na etapa de iluminação [CAI 99].

A grande diferença da abordagem adotada para visualização de dados multimodais em relação às técnicas de *Normal Projection* e *Normal Fusion* (seção 5.3), é que não há a necessidade de extrair uma superfície, permitindo, assim, a visualização de estruturas internas, uma vez que todos os voxels são usados no processamento. Já a técnica de *Spectral Volume Rendering* não utiliza a abordagem $RGB\alpha$ introduzida por Levoy [LEV 88] e usada no desenvolvimento deste trabalho, mas trabalha com outras bandas espectrais e propriedades de iluminação para combinar os volumes.

Para a visualização de estruturas internas em volumes de dados multimodais é possível utilizar funções de transferência e técnicas de corte e segmentação, que podem ser combinadas ou usadas como abordagens alternativas. Técnicas interativas de corte são necessárias para remover partes do volume que estejam obstruindo a visualização de outras porções do seu interior que sejam de interesse [JON 95]. Na seção 6.3 foi apresentada a extensão e utilização da abordagem unificada descrita no capítulo 3 como forma de explorar o interior de volumes de dados multimodais.

A abordagem desenvolvida permite definir os parâmetros de corte baseados na propriedade do voxel tanto a partir do volume de dados “anatômico”, como do “funcional”. Na maioria dos exemplos apresentados na seção 6.3, tais parâmetros foram especificados considerando apenas o volume de dados anatômico, que no caso

é a esfera, porque em muitas situações os volumes sintetizados não se mostraram adequados para a especificação destes parâmetros a partir do cubo.

Na literatura, técnicas como CVR e CVP não são tratadas como ferramentas de corte. Na abordagem unificada apresentada, que pode ser utilizada com um único volume ou com volumes de dados multimodais, tais técnicas são consideradas como ferramentas que podem realizar o corte baseado em uma geometria e no conteúdo dos voxels. Desta maneira, são fornecidas poderosas ferramentas de inspeção do volume cujos parâmetros são de fácil manipulação e a implementação torna-se trivial.

Os exemplos apresentados também mostram que, no caso da visualização de volumes de dados multimodais, a aplicação de ferramentas de corte baseadas na propriedade do voxel fornece uma variedade de possibilidades para a seleção da ROI. Para ilustrar, no caso de um plano de corte frontal, a e b terão os mesmos valores para ambos os volumes, enquanto que na técnica de CVR a pode ser especificado de acordo com um valor v (equação 3.4) obtido do volume V_1 ou do volume V_2 , gerando, assim, resultados diferentes.

As figuras 6.10 e 6.9 mostram que a combinação das técnicas de CVR [MUL 2000] e CVP [ZUI 95] com a integração dos dados baseada no algoritmo de *ray casting* [CAI 99] para permitir a visualização de estruturas internas em volumes de dados multimodais, é uma alternativa muito interessante. A combinação de ferramentas de corte flexíveis com a integração de volumes de dados multimodais, torna a abordagem apresentada bastante poderosa. Suas principais vantagens são:

- Permitir a análise de duas informações diferentes, funcional e anatômica, e fornecer alternativas para eliminar regiões que não contribuem para o diagnóstico usando objetos geométricos ou informações do voxel;
- Dispensar o uso de uma técnica de segmentação;
- Reduzir o tempo de geração das imagens através da redução do número de amostras que devem ser processadas, por exemplo, quando é usado um plano de corte ou as técnicas de CVP e CVR;
- Permitir, através da abordagem unificada, um melhor entendimento das ferramentas de corte descritas na literatura, bem como a identificação de alternativas ainda não exploradas.

A extensão da abordagem unificada para exploração de volumes de dados multimodais possibilita a elaboração de várias combinações que usam as propriedades dos voxels de mais de um volume, que são, então, usadas como parâmetros para as ferramentas de corte. Tais ferramentas são muito úteis e importantes na visualização de volumes e geralmente são descritas na literatura de maneira simples, referenciadas apenas como *slicing tools*. Entretanto, quando considera-se que o corte pode ser feito tomando como base não apenas informações geométricas, mas também as propriedades do voxel, chega-se a uma maneira mais natural e intuitiva de interagir com os dados, principalmente quando volumes multimodais são utilizados. Neste caso, por exemplo, os parâmetros U_a e U_b em uma ferramenta de corte podem ser especificados, respectivamente, como uma propriedade do voxel obtida do volume anatômico ou funcional.

Outros autores já desenvolveram técnicas para visualização de dados multimodais que incluem ferramentas de corte convencionais, tais como seções planares

e objetos volumétricos de corte. Todas estas ferramentas podem ser facilmente modeladas usando a abordagem unificada apresentada, que permite, ainda, o desenvolvimento de novas ferramentas.

Finalmente, apesar dos exemplos apresentados neste capítulo incluírem apenas dois volumes para visualização, a abordagem desenvolvida também possibilita que três ou mais volumes sejam utilizados. Por exemplo, na figura 6.13 o volume da esfera com paralelepípedos foi usado para determinar os parâmetros a e b , a outra esfera foi usada para especificar o valor de opacidade e o cubo foi usado para determinar a cor. Neste caso, foi realizado um corte por conteúdo, no qual foram processadas apenas as amostras com valores de densidade entre 200 e 255.

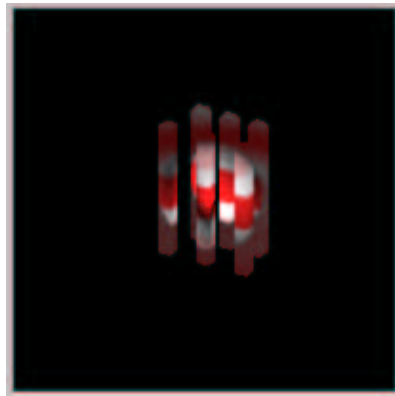


FIGURA 6.13 – Visualização de três volumes de dados multimodais: corte, opacidade e cor determinados por três volumes diferentes

7 *Framework* para Visualização de Imagens Médicas

A abordagem proposta para visualização de volumes envolvendo ferramentas de corte e funções de transferência foi implementada em um *framework* voltado para a área médica e concebido durante o desenvolvimento deste trabalho. Essencialmente os objetivos desta concepção foram:

- Desenvolver um sistema independente de plataforma, que possua uma interface amigável e que possa ser usado tanto nas estações de trabalho médicas, como nos computadores pessoais dos médicos;
- Permitir a integração de ferramentas de segmentação, registro, quantificação e visualização;
- Disponibilizar diversas formas de visualização e ferramentas interativas para a exploração de estruturas internas, bem como permitir uma fácil inclusão de novos módulos.

Inicialmente, foram identificadas as funcionalidades necessárias para tais sistemas [MAN 2000]. Considerando, então, as questões em aberto identificadas, foi desenvolvido um *framework* para um sistema de visualização de imagens médicas, mono e multimodais, que é apresentado neste capítulo.

7.1 Arquitetura do *Framework*

A construção de um sistema de visualização de dados multimodais envolve a integração de várias ferramentas, tais como segmentação, registro e visualização interativa. Nesta seção é apresentada a arquitetura de um *framework* para visualização interativa de dados multimodais que integra estas ferramentas, e pode incluir diversos algoritmos de visualização, tal como para visualização de estruturas internas.

O *framework* para visualização de dados multimodais foi desenvolvido como parte do projeto VPat (*Visualization and Interaction with Virtual Patients*). O objetivo principal do VPat é gerar modelos de representação de seres humanos virtuais para uso em aplicações de computação gráfica na área médica [FRE 2002]. Por isso, muitas decisões sobre a arquitetura descrita a seguir foram tomadas levando em consideração as especificações do VPat, pois várias classes são usadas por outros integrantes do projeto [MAC 2002].

Inicialmente, os seguintes requisitos foram considerados para o desenvolvimento do *framework*: características que um sistema de visualização interativa multimodal deve ter (seção 5.1); possibilidade de utilização de alguns algoritmos otimizados já apresentados na literatura; reutilização de *software*; e o fato de que aplicações interativas devem ser fáceis de usar e, conseqüentemente, projetadas para uma aplicação específica ou grupo de usuários. Baseando-se nestas suposições, foi desenvolvido o modelo conceitual de um *framework* para permitir o projeto e a implementação de sistemas de visualização e exploração de imagens médicas de maneira fácil e rápida [MAN 2000a, MAN 2001].

O modelo é baseado no padrão *Model-View-Controller* (MVC) [BUS 96] e é descrito neste trabalho usando a *Unified Modeling Language* (UML) [LAR 97]. MVC consiste em uma tríade de classes extremamente utilizadas em sistemas interativos para construção de interfaces com o usuário, que mantém o núcleo funcional do sistema independente da interface. O núcleo do sistema normalmente permanece estável, uma vez que é baseado nos requisitos funcionais. A interface com o usuário, entretanto, geralmente está sujeita a alterações e adaptações. Assim, é essencial o desenvolvimento de uma arquitetura que dê suporte às alterações da interface sem causar efeitos colaterais nas funções específicas da aplicação ou do modelo de dados [BUS 96]. Além disso, como a tendência atual é criar sistemas com um alto grau de interação com o usuário, a maioria das classes foi modelada considerando tal interação. Por exemplo, a segmentação manual é uma tarefa difícil e cansativa, mas sem o conhecimento do usuário é muito difícil selecionar as estruturas corretamente. Esta é a razão porque a segmentação interativa é considerada como uma das mais eficientes [ZUI 95, NOO 97, OLA 99].

Num primeiro passo, foi elaborado o diagrama de caso de uso apresentado na figura 7.1. De acordo com este diagrama o sistema tem dois atores: o usuário (por exemplo, um médico radiologista), que vai manipular os dados e ferramentas disponíveis; e o sistema de aquisição, que vai fornecer ao sistema os volumes de dados. O caso de uso modelado contempla algumas funcionalidades listadas como questões em aberto em um trabalho anterior [MAN 2000].

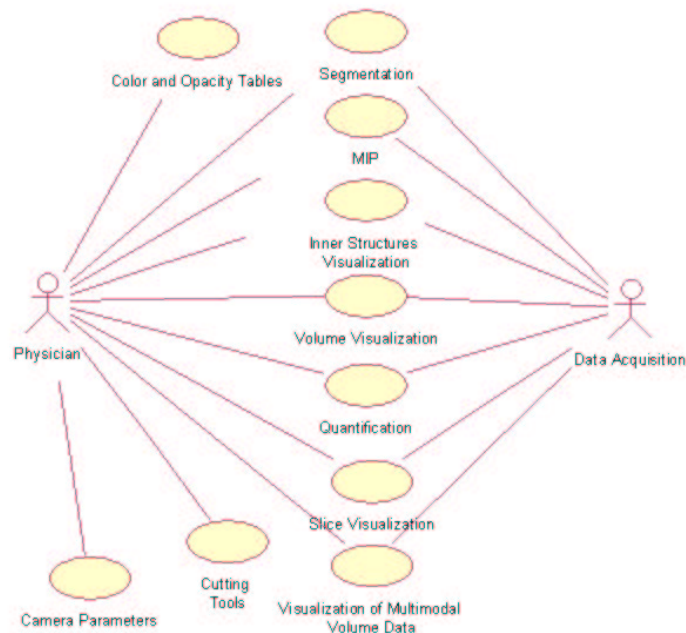


FIGURA 7.1 – Diagrama de caso de uso

A arquitetura proposta possibilita a integração de ferramentas de registro, segmentação e visualização interativa de volumes de dados multimodais. A figura 7.2 mostra uma descrição simplificada do modelo conceitual em UML, onde é possível verificar que o paradigma orientado a objetos permite uma fácil integração das ferramentas existentes, bem como uma fácil extensão para inclusão de novas ferramentas.

A biblioteca de classes foi projetada de tal maneira que os métodos responsáveis pelo gerenciamento da interface estão concentrados nas classes *UserInterface* e *View*, e as outras ferramentas e funções permanecem completamente independentes da interface. Assim, de acordo com a figura 7.2, e considerando o padrão MVC, a classe *UserInterface* corresponde ao controle do sistema, responsável pelo gerenciamento dos eventos, como por exemplo quando o usuário clica em um ícone. A classe *View* tem como função exibir os dados. Desta maneira, o modelo, representado aqui pela classe *Scene*, que é composta por listas das classes *Camera*, *Light* e *GraphicObject*, é totalmente independente de uma plataforma específica. Usando estas classes, a organização estrutural fundamental para um sistema interativo de visualização de dados mono ou multimodais é fornecida.

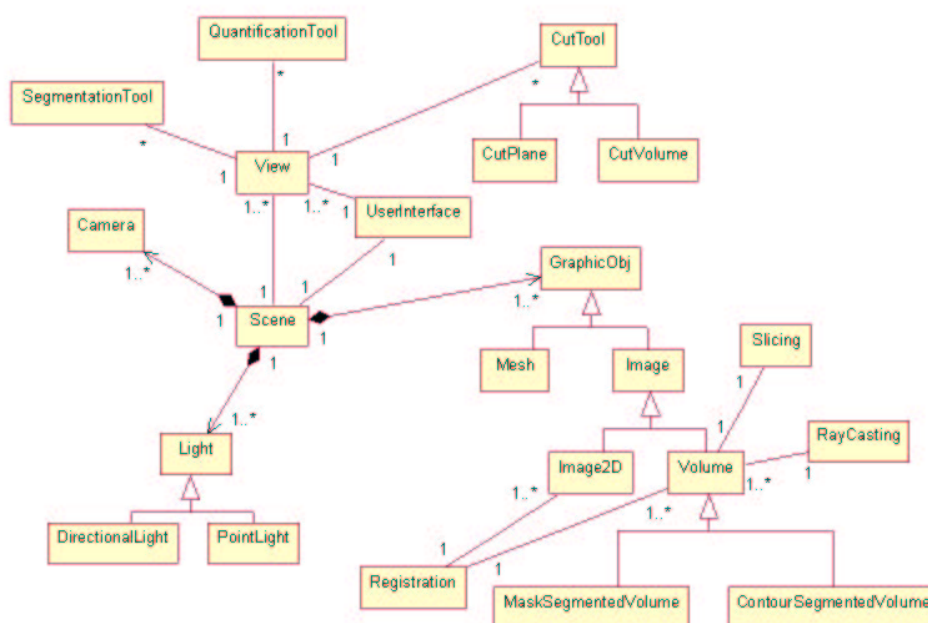


FIGURA 7.2 – Modelo conceitual simplificado

Aplicações de visualização de dados médicos geralmente necessitam de mais de uma janela de visualização. Um exemplo clássico consiste na exibição dos planos axial, coronal e sagital, juntamente com o órgão reconstruído a partir de dados de CT ou MRI. Como neste caso há uma dependência “um-para-muitos” entre os objetos, quando um objeto troca de estado, todos os seus dependentes devem ser notificados e atualizados automaticamente [GAM 95]. No *framework* desenvolvido, o objeto *UserInterface* e todas as instâncias da *View* são dependentes das modificações do modelo. Tanto os objetos da *View*, como da *UserInterface*, agem como um observador: eles estão constantemente olhando o modelo, isto é, um objeto *Scene*. Quando o modelo se modifica, os objetos são, então, atualizados. Na verdade, há um mecanismo de propagação das alterações que mantém um registro dos componentes dependentes do modelo. Alterações no estado do modelo provocam o mecanismo de propagação, que é a única ligação entre as classes *Scene/GraphicObject*, *UserInterface* e *View* [BUS 96].

Para ilustrar a interação entre os objetos, um diagrama de colaboração é apresentado na figura 7.3. Neste caso, o usuário troca o tipo de visualização para MIP,

por exemplo, ao clicar em um botão. Primeiro, o evento é detectado (1); depois, uma mensagem é enviada para *GraphicObject*, mais especificamente para uma instância de *Volume* (2), para notificá-lo da necessidade da sua modificação. Após a realização das alterações, uma mensagem é enviada para uma instância da classe *Scene* (3), que então notifica um objeto da classe *View* (4), que é responsável por verificar o novo estado do objeto (5) para re-exibição da imagem, e um objeto da classe *UserInterface* (6), que também pode ser modificado caso uma característica específica esteja ou não disponível depois das alterações do objeto (7).

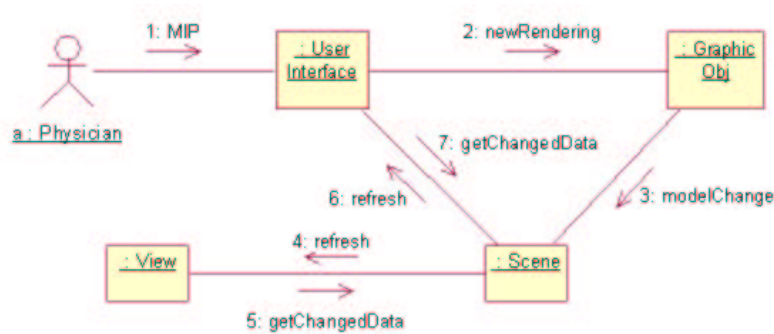


FIGURA 7.3 – Diagrama de colaboração

Os motivos que levaram à associação das ferramentas de interação, tais como *CutTool*, *SegmentationTool* e *QuantizationTool*, com a classe *View* são: necessidade de uma representação gráfica para estas ferramentas que é manipulada pelo usuário (por exemplo: um plano de corte); alteração do modelo a partir da manipulação das ferramentas; e, conseqüentemente, nova geração de imagens. Na verdade, através da manipulação destas ferramentas são obtidos os valores que servem como parâmetros de entrada para a geração das imagens.

A classe *View* pode, por exemplo, requisitar a exibição de um plano de corte e fornecer uma lista de objetos gráficos ativos para uma ferramenta de interação. Como, depois da utilização de uma ferramenta de segmentação, o volume segmentado resultante deve ser armazenado, a classe *Volume* pode ser estendida de tal forma que duas novas classes são criadas (figura 7.2): *MaskSegmentedVolume* e *ContourSegmentedVolume*. De acordo com a técnica de segmentação aplicada, através de uma máscara volumétrica ou com contornos marcados, estas classes podem ser instanciadas ou não.

A classe *RayCasting* tem um conjunto de métodos que implementam variações do algoritmo de *ray casting*, tal como MIP, visualização de dados multimodais e de estruturas internas. A classe *Slicing* é responsável pela geração de imagens de fatias ortogonais ao volume, isto é, imagens dos planos axial, coronal e sagital. A classe *Registration* está associada com as classes *Volume* e *Image2D*, pois fornece métodos para realizar o registro entre imagens 2D ou entre volumes. Instâncias das classes que implementam tabelas de cor e opacidade são atributos da classe *Volume*. A classe *Mesh* é útil para visualização híbrida, que inclui técnicas para a exibição simultânea de dados de volume e modelos geométricos.

Para simplificar o diagrama apresentado, algumas classes básicas implementadas não foram exibidas. Exemplos destas classes são *Point*, *Vector*, *Line*, *Matrix* e *Color*. As classes *View* e *UserInterface* são específicas da interface do protótipo

desenvolvido. Já as classes *Scene*, *Camera*, *Light* e *GraphicObject* são usadas em outras aplicações dentro do projeto VPat [MAC 2002].

7.2 Ambiente de Implementação e Interface do Protótipo

Para possibilitar o desenvolvimento de sistemas flexíveis, portáteis e fáceis de estender, que são propriedades desejáveis em aplicações de visualização de dados multimodais [STO 98], o protótipo foi desenvolvido usando apenas *software* livre. A linguagem de programação utilizada foi o C++ padrão, que tem se mostrado como a mais adequada, uma vez que é possível usar somente as bibliotecas ANSI e compilar o mesmo código em plataformas diferentes, sempre com a garantia de uma boa performance. Zuiderveld et al. [ZUI 96] apresentaram uma avaliação da utilização desta linguagem no desenvolvimento de sistemas de visualização de volumes de dados multimodais.

A biblioteca OpenGL foi utilizada apenas para a exibição das imagens. OpenGL é uma poderosa e sofisticada API (*Application Programming Interface*) que possui rotinas gráficas e de modelagem, 2D e 3D. Sua grande vantagem é tirar proveito das placas aceleradoras gráficas existentes hoje em dia.

Para o desenvolvimento da interface gráfica com o usuário, foi utilizado o *toolkit* FLTK [SPI 2002]. FLTK é portátil, multiplataforma, fácil de usar, desenvolvido sob a Licença Pública Geral GNU e otimizado, tanto em tamanho de código como em performance.

Para a visualização dos volumes de dados, apenas o algoritmo de *ray casting* tradicional foi implementado e estendido, para testar as técnicas de visualização de estruturas internas e de visualização integrada de volumes multimodais. Não foram desenvolvidos algoritmos híbridos, onde volumes são interceptados por objetos geométricos e técnicas de otimização [SIL 2000]. O custo computacional do algoritmo não foi considerado como um fator limitante porque este problema tem sido reduzido com o desenvolvimento de arquiteturas paralelas, placas aceleradoras gráficas e placas de *hardware* projetadas especialmente para manipulação de dados de volume.

A figura 7.4 apresenta uma imagem da interface com o usuário projetada para o protótipo. A interface é composta de duas janelas principais: a janela “a” contém somente algumas funções do sistema, representadas por seis ícones, enquanto a janela “b” (figura 7.4b) é dedicada à visualização 2D e 3D. A idéia é que ao usuário clicar com o mouse sobre um dos ícones da janela “a”, uma tarefa específica seja executada (da esquerda para a direita): segmentação, visualização 2D, visualização de um volume monomodal, visualização de dois volumes (multimodais, ou não), ajuda e saída do protótipo. Por exemplo, quando o terceiro ícone é selecionado (visualização monomodal) na janela “a”, uma caixa de diálogo é aberta para permitir a seleção de um arquivo de dados. Depois, esta caixa é fechada e a janela de visualização “b” é aberta. Esta janela apresenta uma visualização *default* e um conjunto de ícones que permite a seleção de outras funcionalidades do sistema.

A interface do protótipo foi inicialmente projetada com o objetivo de ser simples e fácil de utilizar. Dependendo das funcionalidades selecionadas, novas janelas são abertas para especificação de parâmetros, e os resultados são sempre exibidos na(s) mesma(s) janela(s) de visualização. Para a visualização *default*, o usuário pre-

cisa fornecer apenas o nome do arquivo que contém o volume de dados. Entretanto, depois desta primeira etapa, os parâmetros de visualização podem ser alterados através da seleção dos ícones apropriados ou através da interação com o mouse na janela de visualização. Os ícones que aparecem na janela “b” são (da esquerda para a direita, de cima para baixo): salvar a imagem gerada; abrir um novo volume de dados; ativar as ferramentas de corte; permitir a visualização MIP; colocar linhas nos planos ortogonais indicando a posição de cada um; ativar a ferramenta de visualização de estruturas internas; alterar parâmetros de *rendering*; especificar detalhes de iluminação; manipular tabela de cores; manipular tabela de opacidades; extrair informações funcionais; extrair medidas; ajuda; fechar a janela.

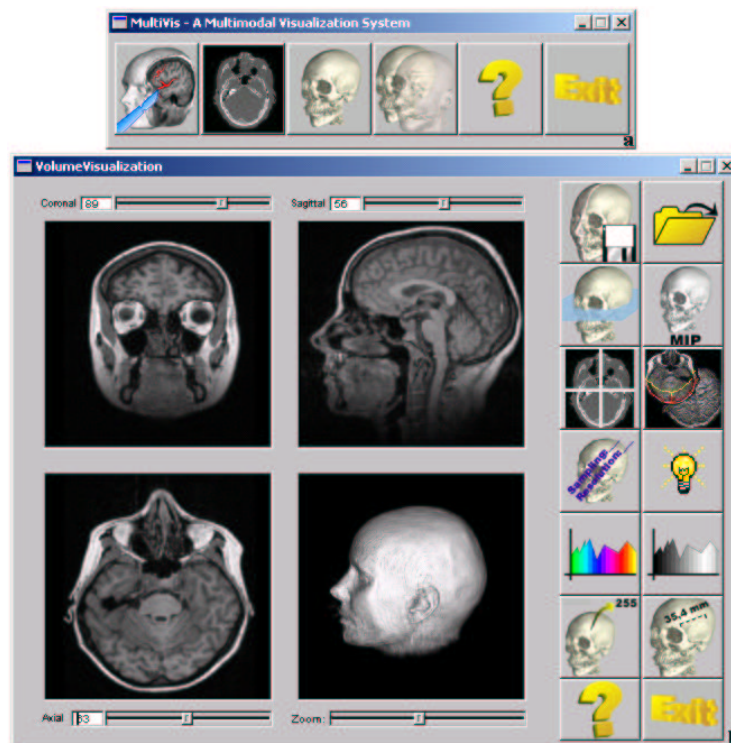


FIGURA 7.4 – Interface do protótipo

Para permitir, por exemplo, a geração da imagem apresentada na figura 7.4 instâncias das seguintes classes são criadas:

- *UserInterface*, responsável pelos botões, barras de rolagem e exibição das janelas de visualização de imagens;
- *View*, três instâncias para os planos ortogonais e uma para a visualização do volume;
- *Scene*, composta por uma instância da classe *Volume* e quatro instâncias da classe *Camera* (uma para cada *View*).

As classes *UserInterface*, *View* e *Scene* compõem o padrão MVC. A primeira delas é responsável por todo tratamento de eventos, que corresponde à identificação de cada ação do usuário e conseqüente notificação ao modelo das ações que devem ser realizadas. A classe *View* tem como função exibir uma imagem, 2D ou 3D, e possui

uma instância do modelo (*Scene*). Assim, cada vez que é necessário redesenhar a imagem, o modelo é notificado.

No exemplo da figura 7.4, a instância da classe *Scene* contém apenas um *GraphicObject*, que corresponde a um volume de dados médicos de MRI. As quatro instâncias da classe *Camera* são necessárias porque os dados são visualizados de maneiras distintas em cada janela. A classe *Volume* é uma das mais importantes, pois além de ler os dados do arquivo (*.raw* ou *.vol*) e armazená-los em uma estrutura de dados, possui instâncias das classes *Slicing* e *RayCasting*, responsáveis pela geração das imagens.

A classe *Slicing* contém apenas os métodos utilizados para a geração da imagem de um plano ortogonal. No momento que são invocados, estes métodos recebem as informações sobre o volume de dados e a câmera. A classe *RayCasting* contém os métodos necessários para a visualização 3D. Entre estes métodos, os mais importantes são: processamento da interpolação trilinear; configuração dos parâmetros de iluminação (luz ambiente, difusa e especular); processamento do gradiente por Sobel ou pelo método das diferenças centrais [LIC 98]; classificação, de acordo com as informações contidas nas instâncias da classe *Table*, que é responsável pela configuração de tabelas de cor e opacidade; implementação do algoritmo de *ray casting* propriamente dito, considerando a parametrização de a e b e a utilização de um ou mais volumes de dados.

Atualmente, os valores de a e b são fornecidos através de um arquivo de configuração que é lido pela instância da classe *RayCasting*. Entretanto, a técnica de CVR, conforme descrito na seção 6.3, e a ferramenta de corte para extração de uma seção oblíqua permitem a especificação destes parâmetros através de uma interface interativa. Para extração de uma seção oblíqua (corte por inclusão, onde $f = f_a = f_b$) é possível manipular interativamente um plano, que consiste no parâmetro $U = U_a = U_b$, conforme ilustra a figura 7.5. Neste caso, é possível movimentar as linhas para posicionar o plano e a imagem resultante é automaticamente atualizada.



FIGURA 7.5 – Extração de um plano de corte oblíquo - as linhas brancas são usadas para posicionar o plano

Conforme descrito na seção 7.1, o *framework* foi desenvolvido com o objetivo de facilitar a inclusão de algoritmos já descritos na literatura. Para ilustrar, na implementação do protótipo, foram utilizadas as funções que implementam o cálculo do gradiente, o modelo de sombreamento e algumas funções de transferência, que estão disponíveis em Lichtenbelt et al. [LIC 98].

Uma parte do *framework* foi compilada e executada no ambiente operacional Linux, como mostra a figura 7.6. Apesar do desenvolvimento de um sistema portátil não ser o foco desta tese, devido a escolha do ambiente de implementação, foi possível observar que o protótipo desenvolvido é portátil.

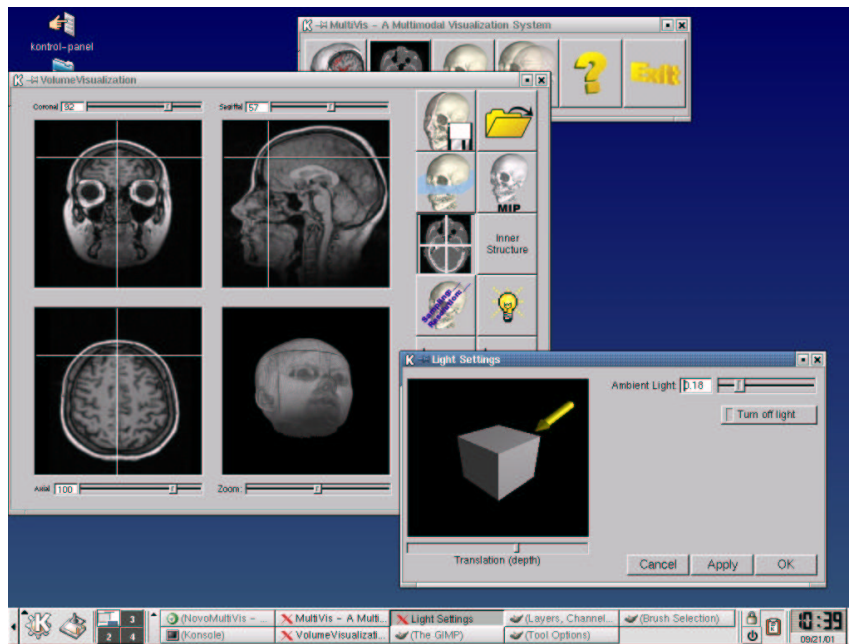


FIGURA 7.6 – Protótipo em execução no Linux

7.3 Comentários Finais

Neste capítulo foi apresentado o *framework* para visualização de dados multimodais que suporta a integração de várias ferramentas e foi desenvolvido no contexto projeto VPat [FRE 2002].

Alguns sistemas para a visualização de volumes de dados multimodais têm sido desenvolvidos e são apresentados na literatura, conforme descrito no capítulo 5. O sistema ANALYZETM [ROB 94, ROB 98] fornece um conjunto de ferramentas de visualização, manipulação e extração de medidas em imagens médicas. Obtenção de seções oblíquas e curvas e seleção de subvolumes são exemplos de ferramentas de exploração disponíveis, inclusive para volumes de dados multimodais. Apesar de atualmente estar disponível para várias plataformas, este sistema não é de domínio público. Muitas das suas funcionalidades são facilmente modeladas utilizando a abordagem apresentada na seção 3.1. Por exemplo, seções e subvolumes consistem apenas em um ajuste dos parâmetros a e b . Além disso, o ANALYZETM não fornece ferramentas de corte baseadas nas propriedades dos voxels e a técnica de visualização de estruturas internas implementada depende de segmentação.

O VROOM (*Volume Rendering by Object-Oriented Methods*) [ZUI 95] é um sistema portátil e orientado a objetos que permite a visualização integrada de volumes de dados multimodais, incluindo uma técnica para visualização de informações vasculares em volumes monomodais (CVP). Esta técnica foi facilmente implementada através da abordagem unificada, onde o parâmetro b é definido de acordo com as propriedades do voxel. Além disso, técnicas de segmentação e registro não são incluídas na arquitetura do VROOM, mas os dados precisam obrigatoriamente passar por um pré-processamento. O *framework* aqui apresentado fornece meios para a integração das várias ferramentas necessárias, segmentação, registro e visualização, além de permitir o desenvolvimento de um conjunto de ferramentas para visualizar estruturas internas, incluindo a técnica de CVP.

O pacote 3DBench fornece rotinas genéricas para manipular e visualizar volumes, ferramentas para extração de medidas e um *slicer* interativo para a obtenção de planos de corte ortogonais e oblíquos [NOO 97]. Vários tipos de volumes de dados podem ser utilizados, como por exemplo, volumes segmentados e multimodais. Entretanto, o 3DBench não dá suporte à segmentação interativa e a única ferramenta para visualização de estruturas internas disponível é o *slicer*, ao contrário do *framework* apresentado, que fornece um extenso conjunto de ferramentas de corte que são facilmente implementadas e permitem a visualização do interior de um volume de dados.

Uma nova abordagem que possibilita a fusão de dois volumes foi desenvolvida por Hastreiter e Ert [HAS 98] para a visualização integrada e interação com os volumes de dados registrados. Neste caso, é utilizado o mapeamento de textura por *hardware* disponível nas placas aceleradoras gráficas. Open Inventor [WER 94] e OpenGL foram utilizados na implementação deste trabalho.

O uso de volumes tratados como texturas não foi explorado no trabalho aqui descrito, mas a extensibilidade do *framework* facilitaria o seu desenvolvimento. Esta extensibilidade é evidenciada pelos demais trabalhos realizados e em andamento no contexto do projeto VPat. Maciel [MAC 2002], por exemplo, utilizou o mesmo *framework* introduzindo classes para modelar articulações e desenvolveu uma classe *View* baseada no Open Inventor.

8 Estudo de Caso

Neste capítulo é apresentado um estudo de caso com volumes obtidos por tomografia computadorizada e por ressonância magnética, onde é utilizado o *framework* descrito no capítulo 7 em conjunto com as abordagens apresentadas nos capítulos 3 e 6.

8.1 Introdução

Ferramentas de corte são muito utilizadas com volumes de dados médicos, pois permitem eliminar estruturas que dificultam a visualização de regiões de interesse. Tais ferramentas podem ser utilizadas como uma alternativa para visualizar estruturas internas em volumes de dados multimodais, conforme descrito no capítulo 6. Por exemplo, uma aplicação da ferramenta de corte em volumes de dados multimodais é a visualização volumétrica do cérebro a partir de dados MRI com um plano de corte representando a informação funcional [PAY 90]. A partir da visualização 3D de imagens MRI do cérebro, Stokking et al. [STO 94] apresentaram os dados funcionais SPECT com as respectivas características anatômicas das imagens MRI em um plano de corte multimodal. Silva [SIL 2000] também implementou a visualização de planos de corte, onde os dados MRI que caracterizam a anatomia são usados na visualização do volume, e dados funcionais PET são usados somente para adicionar cor ao plano de corte que secciona o volume. O trabalho desenvolvido por Noordmans [NOO 97] inclui um *slicer* interativo que possibilita a extração de cortes ortogonais e oblíquos nos volumes de dados multimodais.

Para o estudo de caso relatado neste capítulo, foram utilizados dois volumes de dados de pacientes humanos e o volume sintetizado do cubo (anexo). Um dos volumes de dados foi gerado na *GE Aircraft Engines*, em Evendale, Ohio (EUA). Este volume consiste em uma tomografia computadorizada de um dente humano. As fatias axiais foram ordenadas de baixo para cima, uma em cada arquivo, com espaçamento $1 \times 1 \times 1$ mm [PFI 2001]. Estes dados estão disponíveis para uso não comercial em <http://visual.nlm.nih.gov/>.

O outro volume de dados foi obtido por ftp da Universidade de Carolina do Norte (EUA) e consiste em imagens de MRI de um encéfalo. Este volume, que é composto de 128 fatias de 113×128 , foi modificado gerando dois volumes sintetizados: um semelhante a um volume de dados de fMRI e o outro semelhante a um volume de dados de medicina nuclear. O motivo que levou à alteração do volume MRI, gerando dados sintéticos com conteúdo semelhante a outras modalidades, foi a garantia que os dados estariam registrados. No caso dos dados sintetizados de fMRI, foi possível mostrar a área de ativação de acordo com uma tarefa hipotética realizada pelo paciente. O volume sintetizado de medicina nuclear possibilitou a exploração das ferramentas de corte, uma vez que o seu conteúdo abrange todo encéfalo, o que não ocorre com a fMRI.

8.2 Imagens Médicas

As imagens médicas podem ser obtidas, basicamente, de duas maneiras: através de métodos invasivos ou não-invasivos. Idealmente, o diagnóstico definitivo de uma doença deve ser obtido usando métodos não traumáticos, bem como não-invasivos. A aquisição através de métodos não-invasivos inclui as imagens obtidas através de raio X, ultra-som, ressonância magnética, etc. São considerados métodos invasivos aqueles nos quais existe a inserção de algum equipamento ou líquido dentro do corpo humano para obtenção das imagens. Exemplos destas modalidades de aquisição de imagens são a angiografia, a artrografia (raio X com contraste), a colonoscopia, a endoscopia, entre outros [GRE 90].

Ainda considerando as modalidades de aquisição, duas categorias podem ser identificadas: anatômicas e funcionais. Conforme descrito no capítulo 1, as modalidades anatômicas retratam principalmente a morfologia, e são usadas, por exemplo, para identificar corretamente toda a região espacial que um tumor ocupa. Imagens anatômicas pode ser obtidas por exames de raio X, CT, CTA, MRI, MRA e ultra-som. As modalidades funcionais possibilitam a representação de informações funcionais e metabólicas e englobam imagens obtidas por cintilografia, SPECT e PET, que correspondem às imagens de medicina nuclear, e fMRI.

Os diversos tipos de imagens médicas são usados pelos especialistas para visualizar as estruturas internas e as funções do corpo humano de maneira indireta. Na última década, aumentou consideravelmente a utilização de diferentes modalidades de imagens digitais para o diagnóstico clínico e para o planejamento de terapia e cirurgia. Algumas delas, tais como tomografia computadorizada, ressonância magnética e ultra-som, são capazes de produzir imagens espaciais da anatomia interna de alta precisão [GRE 90]. Além disso, novos desenvolvimentos na aquisição de volumes de dados, como por exemplo a tomografia computadorizada helicoidal, estão criando - ou até mesmo exigindo - uma necessidade crescente de visualização tridimensional.

A tomografia computadorizada (CT) revolucionou o uso de imagens médicas, pois permitiu a visualização de estruturas previamente indistinguíveis em diagnósticos radiológicos convencionais. Imagens de CT consistem em um conjunto de imagens de seções do corpo (“fatias”), equivalentes a radiografias de suas camadas internas, obtidas através de incrementos na direção do plano de aquisição de imagens com alguns milímetros de distância entre cada fatia. Através da CT houve um aperfeiçoamento na capacidade de distinguir regiões com diferentes índices de absorção de raios X e na capacidade de separar estruturas encobertas, possibilitando, assim, a demonstração radiológica de muitos órgãos e tecidos doentes [MAT 93, LIC 98].

A CT é indicada para auxiliar no diagnóstico da cabeça, da coluna vertebral, do tórax, do abdome e da pélvis. O paciente corre riscos quando há a necessidade de injeção de material de contraste intravenoso. Algumas de suas vantagens são: rapidez; ótima resolução espacial e possibilidade de avaliação simultânea de múltiplos órgãos do sistema [WAL 94].

Existem, basicamente, dois tipos de CT: por transmissão e por emissão. Na CT por transmissão a imagem de uma fatia do corpo humano é reconstruída a partir de uma série de varreduras com um feixe de raio X. Este feixe atravessa o plano do paciente em diferentes ângulos e a radiação não absorvida pelo corpo do paciente é medida por detectores alinhados com emissores de raio X. Como resultado, é obtida

uma imagem que representa uma fatia plana do corpo examinado, sem sobreposições.

Na CT de emissão (ECT) o objetivo é fazer uma imagem seccional da distribuição de isótopos radioativos no corpo humano, considerando que um isótopo pode ser administrado a um paciente na forma de radiofármacos, tanto por injeção como por inalação. Isótopos radioativos são caracterizados pela emissão de fótons de raio-gama, um produto do decaimento nuclear. Fótons de raio-gama são indistinguíveis dos fótons de raio X, os diferentes termos utilizados são para diferenciar sua origem. A concentração de um isótopo em qualquer secção muda com o tempo, devido ao decaimento radioativo e a cibernética bioquímica do corpo. Isto implica que todos os dados para uma imagem seccional sejam coletados em um pequeno tempo constante, associado às mudanças de concentração. Porém, este aspecto dá à ECT mais potencial e utilidade nos diagnósticos médicos, sendo, assim, usada para a medição do metabolismo, revelando o funcionamento fisiológico do corpo, como fluxo sanguíneo ou consumo de oxigênio.

As técnicas de CT por emissão são divididas em PET e SPECT. Em ambas as aplicações o objetivo é um pouco diferente dos scanners de CT. A origem da radiação eletromagnética é agora um radioisótopo que é distribuído dentro do corpo. O objetivo é gerar um mapa espacial ou uma imagem desta distribuição, que, usualmente, consiste em conjunto de fatias bidimensionais. Dessa forma, enquanto as técnicas de CT geralmente mostram as propriedades de absorção, as quais são relacionadas com a anatomia, as técnicas de emissão mostram a distribuição radioativa, podendo indicar vários aspectos da função fisiológica. Estes traçadores radioativos também podem ser usados em CT na forma de agentes de contraste rádio-opaco. Os resultados dos exames PET e SPECT não apresentam bem as estruturas anatômicas, mas, sim, níveis de atividade metabólica. Imagens PET são usadas principalmente para a detecção de tumores no cérebro, seios, coração, pulmões e cólon, e para a avaliação de anomalias em artérias coronárias. Doenças como epilepsia e mal de Alzheimer também podem ser estudadas através de imagens PET do encéfalo.

Os exames por ressonância magnética (MRI), ao contrário de CT, permitem um bom contraste entre tecidos de baixa densidade. Uma das grandes características da MRI é o seu potencial para realizar análises químicas de regiões do corpo de maneira não-invasiva, pois o tecido que não é identificado em exames de raio X ou scanners CT é claramente distinguível com técnicas MRI. Por exemplo, como um dos átomos mais fáceis de visualizar através da MRI é o de hidrogênio, um dos mais comuns do corpo, é possível observar o metabolismo e caracterizar tecidos, diferenciando, em alguns casos, tecidos normais dos cancerígenos. A tecnologia MRI tem sido uma área de pesquisa ativa, com muitas aplicações clínicas [GRE 90].

A MRI é indicada para a visualização dos tecidos moles da região da cabeça, do pescoço, do tórax, do abdome, da pélvis, do sistema muscular e ósseo e da espinha. Este exame possui alguns riscos e contra-indicações, como para pessoas cardíacas ou com aneurisma. Algumas de suas limitações incluem a necessidade de contraste intravenoso em casos especiais, a sensibilidade ao movimento e a detecção de calcificação. Entre as vantagens que possui, pode-se citar a não utilização de radiação e a possibilidade de demonstrar o fluxo sanguíneo e de detectar a má-formação intracraniana [WAL 94].

O exame de MRI funcional, ou *functional* MRI (fMRI), consiste em uma variação da MRI utilizada para gerar imagens que mostram a atividade metabólica de um órgão. fMRI é usada principalmente quando há interesse no mapeamento das

alterações do fluxo sanguíneo no cérebro, que correspondem à realização de uma atividade neurológica, para as imagens de MRI. Por exemplo, a fMRI pode detectar mudanças na quantidade de oxigênio de uma região do cérebro responsável por uma atividade específica [UDU 2000]. Durante a aquisição das imagens, solicita-se ao paciente a realização desta atividade, que quando é realizada, faz com que a região do cérebro envolvida necessite de oxigênio, que é, então, mapeado para uma cor naquela região. Em geral, as cores variam de vermelho, menor atividade, a amarelo, maior atividade. Cada aquisição é composta por diferentes tarefas, começando com repouso e depois intercalando entre períodos de repouso e de realização de tarefas.

Algumas das modalidades descritas nesta seção são exploradas no estudo de caso relatado nas próximas seções.

8.3 Visualização Monomodal

Para ilustrar as funcionalidades da abordagem unificada apresentada no capítulo 3 com um volume de dados de CT, foi utilizado o volume do dente ilustrado na figura 8.1. Para gerar esta imagem foi utilizada a função de transferência de opacidade proposta por Levoy (equação 4.10) [LEV 88] e uma tabela de cores onde o valor de densidade zero foi mapeado para a cor vermelha e os valores de densidade entre 180 e 155 foram mapeados para a cor branca.

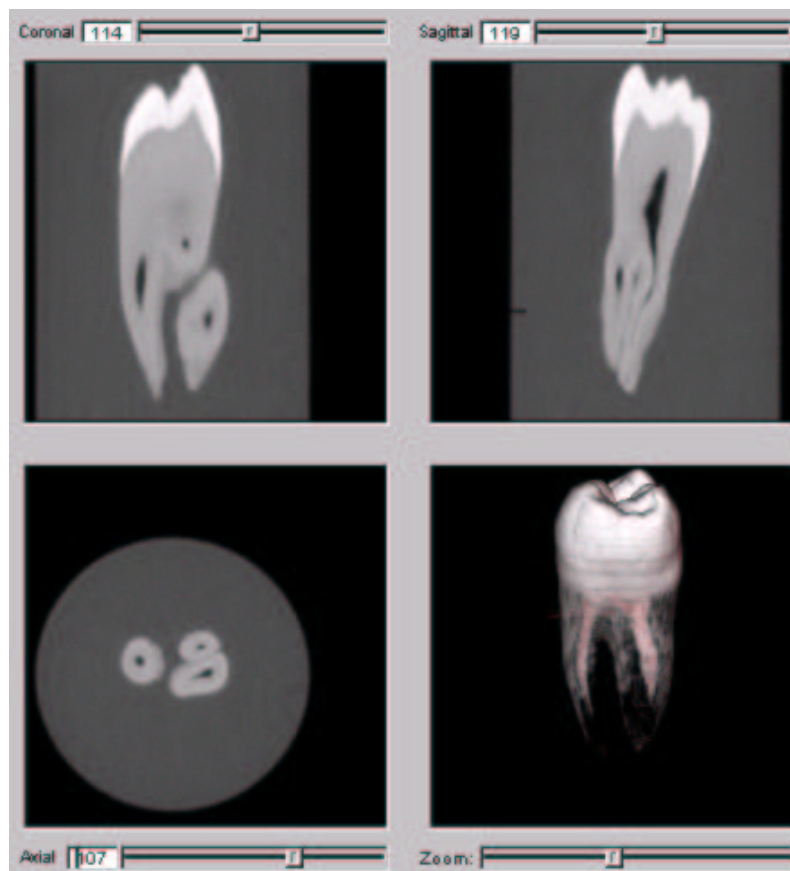


FIGURA 8.1 – Visualização do volume de dados de um dente humano (<http://visual.nlm.nih.gov/>)

Algumas ferramentas de corte, tanto baseadas em geometria como no conteúdo, foram aplicadas neste volume de dados. A figura 8.2a mostra o resultado da utilização de uma função de transferência de opacidade linear a partir de 100 com um plano de corte. Na figura 8.2b foi utilizada a função de transferência de opacidade proposta por Levoy para extração de uma isosuperfície (equação 4.10) [LEV 88] também com um plano de corte. A imagem da figura 8.2a parece ter uma qualidade inferior à imagem da figura 8.2b, porém, com o uso da função de transferência proposta por Levoy, algumas estruturas, no caso o interior do dente, acabam sendo eliminadas, o que não acontece se for usada a função linear. Por exemplo, como na figura 8.2b é extraída uma isosuperfície, no momento que é utilizado um plano de corte, parece que o interior do dente está “vazio”. A utilização de um volume de corte também usando a função proposta por Levoy é ilustrada na figura 8.2c.



FIGURA 8.2 – Visualização com plano de corte (a) com uma função de transferência linear, (b) com a função de transferência de opacidade de Levoy e (c) visualização com volume de corte

Uma dificuldade para a visualização 3D deste volume de dados é a necessidade de eliminar o “cilindro” que envolve o dente, que pode ser observado nos planos axial, sagital e coronal da figura 8.1. Neste caso, pode ser aplicada uma função de transferência de opacidade que atribua opacidade zero para as amostras que possuem a densidade do “cilindro”, conforme ilustrado na figura 8.2, ou pode ser realizado o corte por conteúdo. A figura 8.3 ilustra o resultado obtido quando $a \neq b$, onde $f_a = first(\mathcal{R}_p, v)$, com $v = 200$, e $f_b = a + first(\mathcal{R}_p, v)$, com $v = 250$.

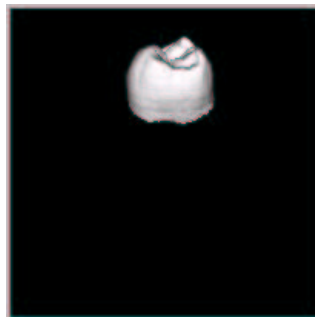


FIGURA 8.3 – Exemplo de corte por conteúdo

O corte aplicando a técnica de CVR neste volume de dados também é uma alternativa para visualizar a ROI. A figura 8.4 apresenta um exemplo de imagem onde foi selecionada uma ROI. Para gerar esta imagem foi utilizada uma função de

transferência de opacidade linear a partir de 100, que permite exibir todo interior sem ressaltar estruturas internas.

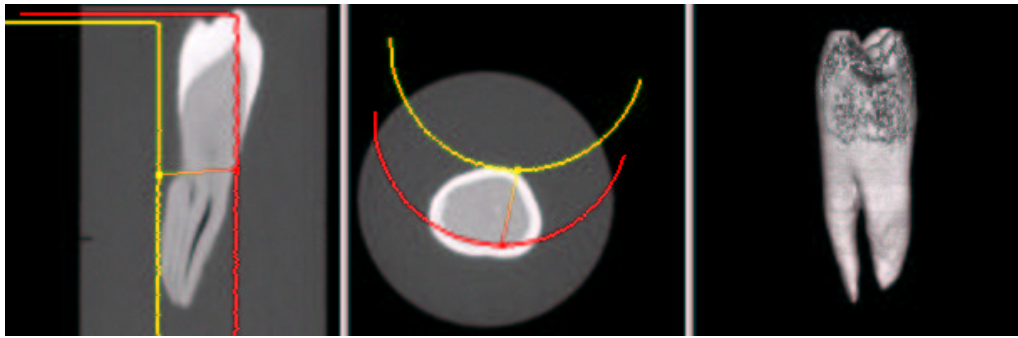


FIGURA 8.4 – Utilização da técnica de CVR com o volume de dados do dente

8.4 Visualização de Volumes de Dados Multimodais

Inicialmente, os dados MRI do encéfalo foram integrados com o volume sintetizado do cubo para ilustrar a utilização da abordagem desenvolvida com dados médicos, como mostra a figura 8.5. Para a geração destas imagens foi usado o algoritmo apresentado na seção 6.2, trocando os valores dos parâmetros a e b para simular as diferentes ferramentas de corte, conforme descrito na seção 6.3. Por exemplo, f_a e f_b foram especificadas considerando uma geometria nas figuras 8.5b e 8.5c, onde foram utilizados, respectivamente, um volume e um plano de corte. Já na figura 8.5d foi usada a técnica de CVR, onde o corte é realizado através da combinação de informações geométricas com propriedades dos voxels, com o objetivo de exibir o encéfalo. Neste caso, os parâmetros S-Band e T-Band foram definidos considerando apenas os dados anatômicos (MRI), através do posicionamento das linhas nos planos axial e sagital. O volume sintetizado foi usado apenas para o processamento da cor.

A figura 8.6 apresenta imagens de fMRI que foram cedidas pelo Centro de Diagnóstico por Imagens do Hospital São Lucas. As marcas que aparecem sobre o encéfalo correspondem a uma região ativada quando o paciente realiza uma determinada tarefa motora, visual, de linguagem ou de memória. Nesta imagem foi realizada uma segmentação manual, e as marcas que aparecem fora dos limites do encéfalo correspondem a artefatos, causados, por exemplo, por uma movimentação do paciente. Como existem padrões conhecidos, para determinadas tarefas, tal como movimentar os dedos, que permitem identificar a área que deve ser ativada, é possível combinar o padrão de ativação de voluntários normais com pacientes que apresentam uma patologia, como mal de Alzheimer e epilepsia.

A figura 8.7 apresenta imagens dos planos axial, sagital e coronal que ilustram como foi sintetizado um exame de fMRI a partir dos dados MRI. As imagens da figura 8.6 foram usadas como modelo para a especificação deste volume. As regiões com a cor branca na figura 8.7 equivalem às marcas que aparecem na figura 8.6, e, conseqüentemente, identificam a área de ativação. Como estas regiões foram manualmente alteradas nas imagens MRI originais, a etapa de registro torna-se desnecessária, neste caso, para a visualização integrada destes dois volumes.

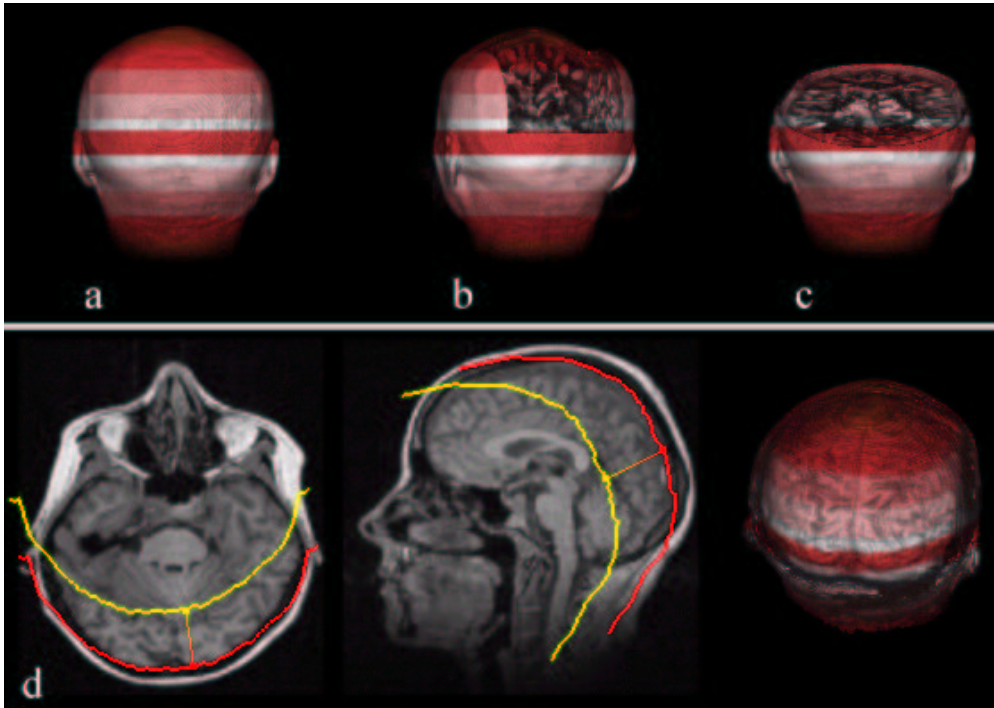


FIGURA 8.5 – Exemplo da visualização integrada de dados MRI com o volume sintetizado

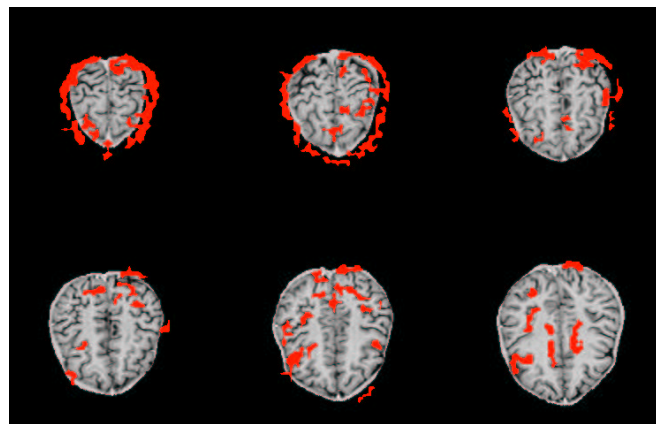


FIGURA 8.6 – Imagens geradas por um exame de fMRI (imagem cedida pelo Centro de Diagnóstico por Imagens do Hospital São Lucas)

Exemplos da visualização integrada dos volumes de MRI e de fMRI sintetizado são apresentados nas figuras 8.8 e 8.9. Observando estas imagens é possível verificar que a visualização integrada em conjunto com as ferramentas de corte permite visualizar as estruturas internas e analisar as suas localizações dentro do volume. Planos e volumes de corte são usados para remover regiões que não são de interesse e revelar o interior do volume de dados (figura 8.9). Já a técnica de CVR permite visualizar apenas uma “camada” do encéfalo, salientando, assim, as áreas de maior ativação. A combinação da técnica de CVR [MUL 2000] com a visualização integrada (figura 8.8) é uma alternativa muito interessante, principalmente pelo fato de não precisar de segmentação e de reduzir o tempo de geração das imagens, resultado da redução do cálculo do número de amostras.

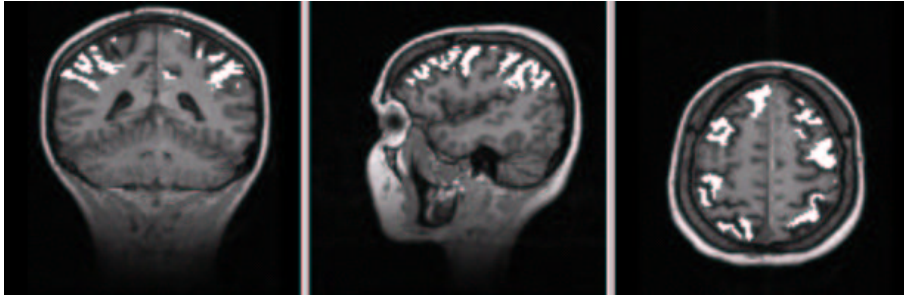


FIGURA 8.7 – Imagens dos dados de fMRI que foram sintetizados

A figura 8.10 contém imagens de medicina nuclear, obtidas por um exame de SPECT. As cores são usadas para identificar quanto o material radioativo foi absorvido pelo tecido. Sendo assim, quanto mais claro, mais material radioativo foi absorvido pelo tecido, de tal maneira que regiões mais permeáveis parecem mais “brilhantes”. A figura 8.11 mostra a região na qual houve uma maior absorção. Para as regiões onde não há ativação, destacada pelo quadrado na figura 8.11, é atribuída a cor azul. Exames de medicina nuclear são usados para vários tipos de diagnóstico. Por exemplo, um tumor tem mais atividade que um tecido normal, por isso precisa de mais sangue e vai absorver mais radiofármaco. Neste caso, também é passada uma tarefa para o paciente, o que vai fazer com que uma determinada região precise de mais sangue e, conseqüentemente, absorva mais radiofármaco. Desta maneira, é possível diferenciar a região do encéfalo onde está o tumor da região responsável pela execução da tarefa. Se estas regiões forem sobrepostas, a retirada do tumor deve ser estudada [ANE 2002].

A alteração do volume de dados de MRI para criar um volume de dados sintetizado de medicina nuclear foi feita de acordo com as imagens apresentadas na figura 8.10. Neste caso, um conjunto de fatias do volume MRI foi manualmente alterado para que o seu conteúdo ficasse similar às imagens axiais exibidas na figura 8.10. Para a visualização integrada deste volume sintetizado com o volume MRI, foi utilizada a tabela de cor apresentada na figura 8.12 e uma função de transferência de opacidade linear a partir do valor de intensidade 10 (equação 4.4). A tabela de cor foi gerada considerando as cores existentes nas imagens da figura 8.10.

Para possibilitar uma exploração das estruturas existentes nos volumes de dados integrados, várias técnicas de corte foram aplicadas. Inicialmente, foram realizados alguns testes com a aplicação de ferramentas de corte geométrico. A figura 8.13 ilustra a utilização de planos (8.13a) e volumes (8.13b) de corte, por inclusão e por exclusão. As cores representam a atividade funcional no interior do volume de dados. Como as imagens sintetizadas de medicina nuclear correspondiam apenas à região do encéfalo, e não a todo volume de dados MRI, a parte do volume onde não existe informação funcional é exibida em tons de cinza.

A utilização da técnica de CVR (figura 8.14) apresenta resultados mais interessantes da exploração do volume de dados. Neste caso, tanto a como b são especificados com base numa informação de geometria combinada com valores dos voxels do volume de dados anatômico (MRI), e o volume de dados funcional é usado apenas para atribuição de cor. O que muda nas imagens de cada linha da figura 8.14, é a seleção de diferentes intervalos para realizar a amostragem. O mapeamento da informação de medicina nuclear para cor permite identificar como está a atividade

funcional em cada região.

Exemplos onde um corte é realizado com base no conteúdo do volume de dados funcional são apresentados na figura 8.15. Para gerar as imagens da figura 8.15a, são processadas apenas as amostras que estiverem depois da primeira amostra no volume funcional com valor de intensidade 150. Sendo assim, $f_a = first(\mathcal{R}_p, v)$ e $f_b = r2$, com $v = 150$ (seção 3.3). Observa-se, neste caso, que não foram encontradas amostras com este valor de intensidade em alguns raios, pois as imagens da figura 8.15a apresentam regiões “vazias”. A figura 8.15b foi gerada de forma semelhante à 8.15a, onde a única diferença é o valor de v , que é igual a 100. A grande vantagem do corte por conteúdo é a possibilidade de identificar melhor as regiões onde há uma maior absorção de radiofármaco. Por exemplo, na figura 8.15b foi eliminada toda região onde não há ativação (destacada pelo quadrado na figura 8.11), exibindo apenas o cérebro. Portanto, o corte por conteúdo permite isolar as regiões onde há alteração de absorção, sem a necessidade de utilizar um algoritmo de segmentação.

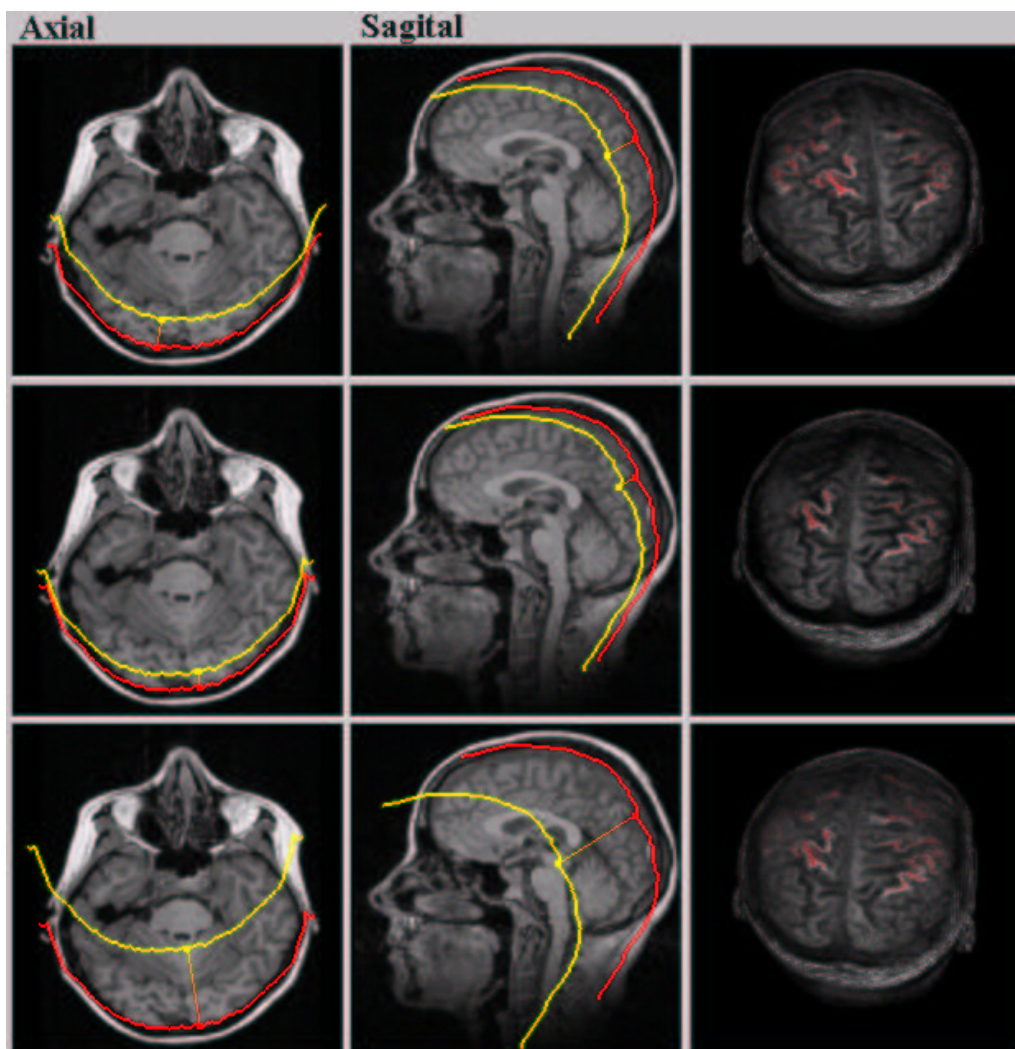


FIGURA 8.8 – Visualização dos dados multimodais (MRI e fMRI sintetizado) através da técnica de CVR

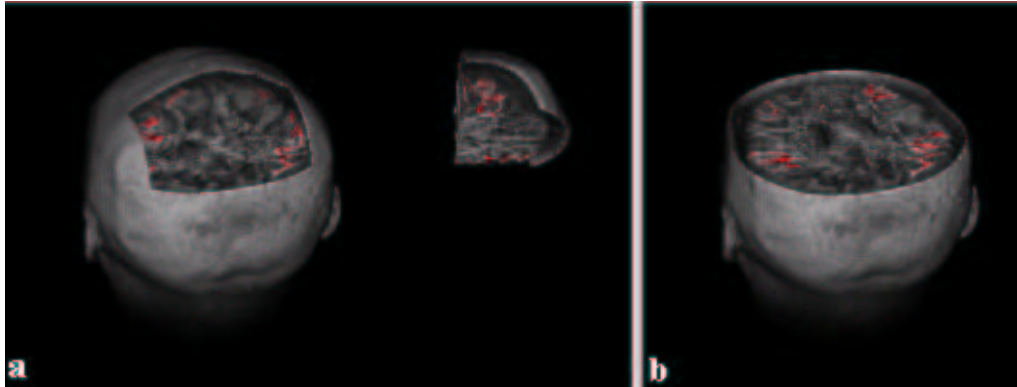


FIGURA 8.9 – Utilização das ferramentas de (a) volume de corte, por exclusão e por inclusão, e (b) plano de corte com volumes de dados multimodais (MRI e fMRI)

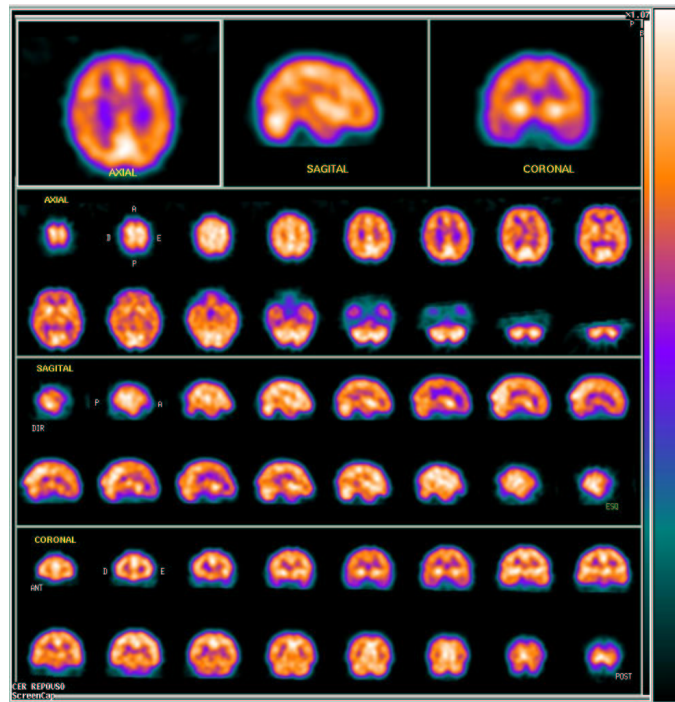


FIGURA 8.10 – Imagens geradas por um exame de medicina nuclear (imagem cedida pelo Centro de Diagnóstico por Imagens do Hospital São Lucas)

8.5 Comentários Finais

Os resultados obtidos com a visualização de estruturas internas em imagens médicas multimodais foram descritos neste capítulo. Neste contexto, como muitas vezes os dados são de natureza complementar, a abordagem apresentada fornece poderosas ferramentas de exploração que ajudam a revelar informações adicionais para o diagnóstico. As ferramentas de corte permitem, por exemplo, exibir os volumes integrados usando até mesmo um terceiro volume para especificar os parâmetros a e b , ou isolar diferentes regiões de interesse apenas através da especificação das

funções f_a e f_b . Por exemplo, na técnica de CVR, ao invés das distâncias d_1 e d_2 serem definidas por valores geométricos fixos, poderiam ser determinadas através de propriedades do voxel. Para ilustrar, as linhas que especificam as regiões T-Band e S-Band na figura 8.14 poderiam acompanhar a anatomia do tecido cerebral. Para alcançar este objetivo na abordagem apresentada seria necessário apenas redefinir as funções f_a e f_b .

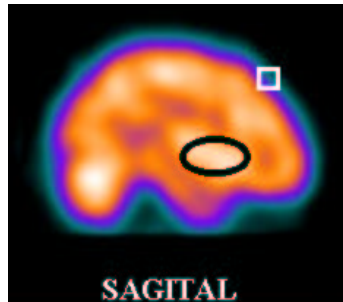


FIGURA 8.11 – Imagem SPECT: a elipse e o círculo exemplificam, respectivamente, regiões onde há e não há ativação (imagem cedida pelo Centro de Diagnóstico por Imagens do Hospital São Lucas)

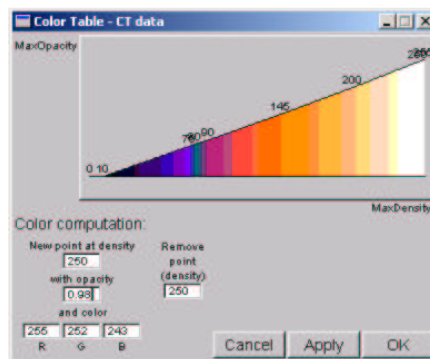


FIGURA 8.12 – Tabela de cor utilizada na geração das imagens onde o volume sintetizado de medicina nuclear é integrado com o volume de MRI

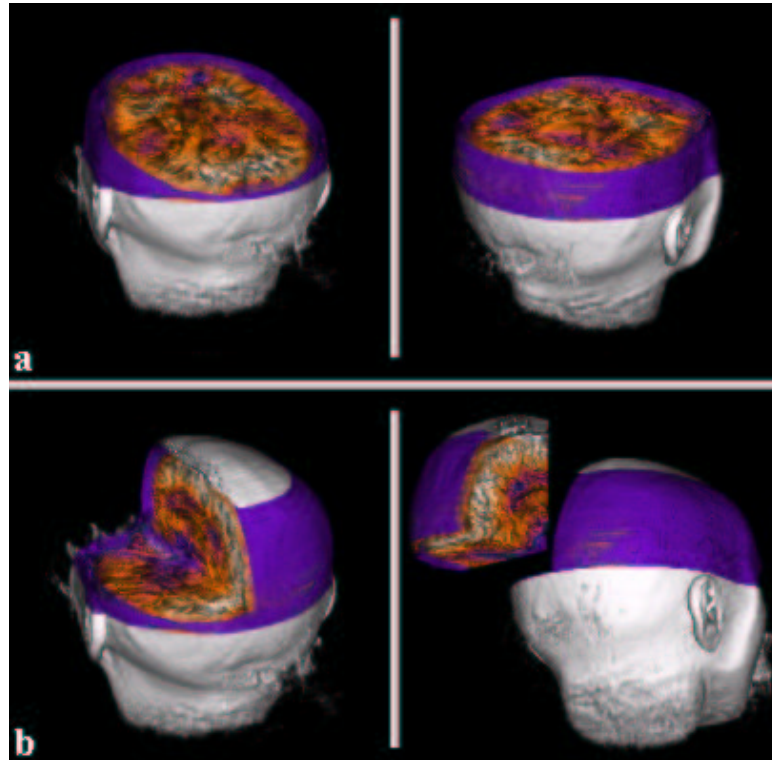


FIGURA 8.13 – Visualização dos volumes de dados de MRI e medicina nuclear com a aplicação de (a) planos e (b) volumes de corte

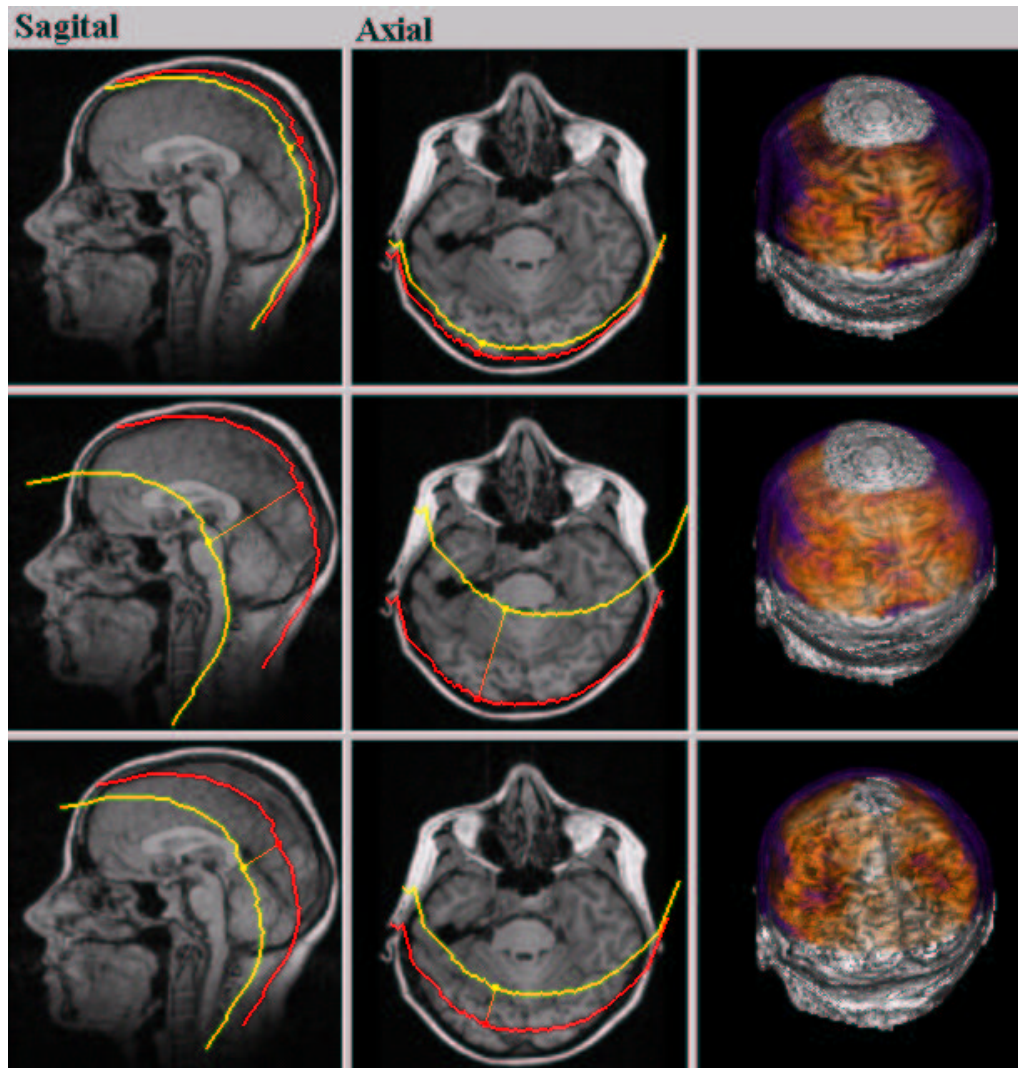


FIGURA 8.14 – Utilização da técnica de CVR em conjunto com a visualização integrada dos volumes de dados de MRI e medicina nuclear

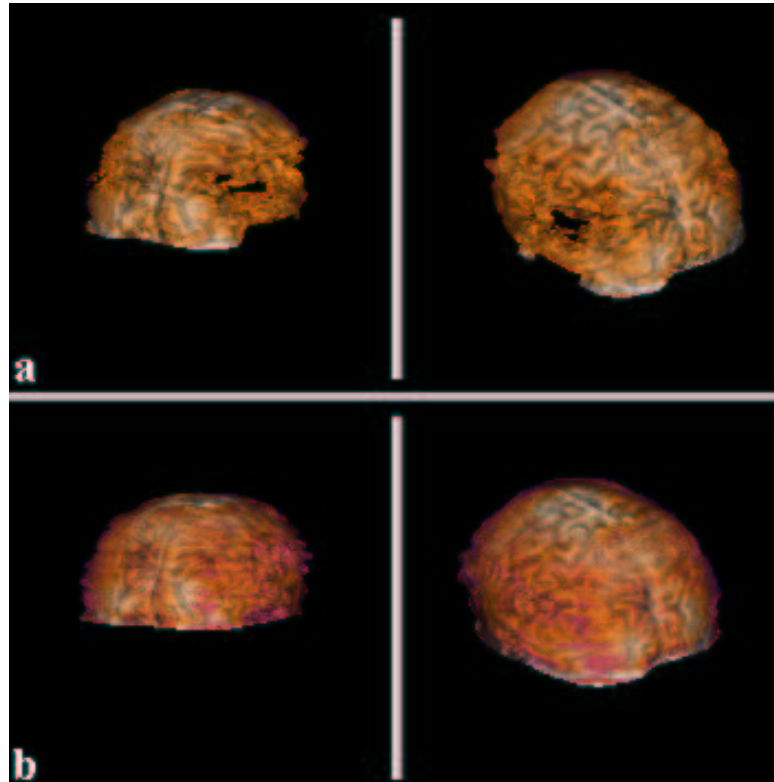


FIGURA 8.15 – Exemplos da utilização de técnicas de corte baseadas no conteúdo do volume de dados

9 Conclusões e Trabalhos Futuros

9.1 Introdução

A área de visualização de dados multimodais tem como principal objetivo desenvolver estratégias para integrar características importantes de mais de um volume de dados em uma única imagem. Entre as várias questões em aberto identificadas nesta área de pesquisa [MAN 2000] destacam-se: (a) o desenvolvimento de sistemas independentes de plataforma com uma interface de fácil utilização; (b) integração de técnicas de segmentação, registro e visualização; (c) elaboração de novos algoritmos para visualização integrada, bem como para a (d) visualização de estruturas internas em volume de dados multimodais; (e) geração de imagens em tempo real. As quatro primeiras questões foram tratadas neste trabalho, e os resultados são avaliados na seção 9.2. A seção 9.3 discute trabalhos futuros.

9.2 Avaliação Geral e Contribuições

As principais contribuições desta tese são uma nova alternativa para a visualização integrada de volumes de dados multimodais e a especificação e formalização de uma abordagem unificada para o uso de ferramentas de corte baseada em geometria ou no conteúdo do voxel, com o objetivo de visualizar estruturas internas, suprimindo alternativas para as questões (a), (b), (c) e (d).

Através do estudo das técnicas de visualização volumétrica foi possível identificar a existência de um grande número de aplicações que beneficiam-se desses sistemas e que necessitam de ferramentas que possibilitem tanto a combinação de informações disponíveis em mais de um volume, como a visualização de suas estruturas internas. O trabalho culminou no desenvolvimento de uma abordagem para visualização de estruturas internas em mais de um volume de dados e de um *framework* para integração das diversas ferramentas envolvidas neste processo.

A alternativa proposta para a visualização de estruturas internas consiste, basicamente, numa interpretação e formalização unificadas das ferramentas de corte, tanto geométricas, como baseadas no conteúdo dos voxels. Neste caso, cada ferramenta de corte pode ser representada através da atribuição de valores diferentes para os parâmetros que determinam o intervalo de amostragem no algoritmo de *ray casting*. Técnicas de visualização volumétrica com mecanismos sofisticados para eliminar estruturas que não são de interesse podem ser enquadradas nesta abordagem, mesmo que o conceito de corte não esteja explícito no método proposto.

A adoção desta abordagem unificada facilitou o entendimento do funcionamento de cada ferramenta de corte e das suas aplicabilidades para inspecionar de forma interativa os volumes de dados. Estas ferramentas podem ser facilmente implementadas de forma direta nos sistemas de visualização, além de abrirem um leque de alternativas que possibilitam eliminar estruturas que não são de interesse, de maneira simples e flexível.

A visualização integrada [CAI 99] dos volumes de dados multimodais fornece informações complementares que podem facilitar, por exemplo, o diagnóstico médico. Além disso, a combinação deste tipo de visualização com as técnicas de corte genéricas para identificação de estruturas internas fornece novas maneiras de

explorar os volumes. A utilização destas ferramentas, além de evitar o uso de uma técnica de segmentação, reduz automaticamente o tempo de geração das imagens, pois o corte implica numa diminuição do número de amostras a serem processadas. O estudo realizado revela que várias ferramentas de corte podem ser criadas a partir da combinação de múltiplos volumes de dados e das estratégias de corte encontradas nos métodos existentes. Por exemplo, a abordagem unificada permitiu a fácil integração da técnica de CVR [MUL 2000] com a visualização de dados multimodais, sendo que os parâmetros para o corte podem ser definidos em qualquer um dos volumes, abrindo novas possibilidades para estudos de caso com estas técnicas. No contexto da visualização de imagens médicas, são fornecidas ferramentas poderosas para explorar os volumes multimodais, na maioria das vezes de natureza complementar, revelando informações adicionais para o diagnóstico médico. A formulação genérica para as ferramentas de corte permite, por exemplo, exibir os volumes integrados usando até mesmo um terceiro volume para especificar os parâmetros a e b para o corte.

Na literatura encontram-se vários trabalhos dedicados ao desenvolvimento de algoritmos otimizados para visualização e registro de volumes. Para a geração de imagens e a interação com o volume de dados em tempo real, a tendência atual é realizar a visualização através do mapeamento de textura por *hardware*. O principal motivo é o avanço da tecnologia, a popularização, e conseqüentemente, o baixo custo das placas aceleradoras gráficas. O objetivo do *framework* é possibilitar a utilização destes algoritmos, em conjunto com as ferramentas disponíveis, sem a necessidade de implementar todos os módulos necessários. Além disso, entre os sistemas estudados que permitem a visualização de volumes de dados multimodais, observou-se que existe uma carência no desenvolvimento de técnicas para integrar ferramentas de registro, quantificação, segmentação interativa e visualização. Segundo Johnson [JOH 99], um sistema de visualização interativa deve ser:

- modular e fácil de estender, como proposto no projeto do *framework* orientado a objetos (seção 7.1);
- adaptável ao *hardware*, que pode variar de grandes supercomputadores a estações de trabalho gráficas e PC's, o que foi alcançado com o uso do padrão MVC, da linguagem de programação C++ padrão, da biblioteca gráfica OpenGL e do *toolkit* FLTK.

A flexibilidade e a portabilidade fornecidas pelo *framework* é, em grande parte, conseqüência da utilização do padrão MVC [BUS 96]. Por exemplo, a extensão do algoritmo de CVR [MUL 2000] inclui a especificação interativa dos parâmetros S-Band e T-Band. Esta especificação é completamente independente da interface com o usuário, pois faz parte do modelo. Assim, mesmo que outra interface seja implementada, ou que o *framework* seja utilizado em outra plataforma, apenas utilizando as classes do modelo se tem acesso à definição interativa dos parâmetros. Conforme já mencionado, a portabilidade é desejável em sistemas de visualização.

9.3 Trabalhos Futuros

Através do trabalho desenvolvido, surgiram várias perspectivas de desdobramento da pesquisa realizada. A descrição destes futuros trabalhos foi separada em

dois grupos: trabalhos que podem ser feitos a curto prazo (seção 9.3.1) e trabalhos mais elaborados, que podem ser desenvolvidos como dissertações de mestrado (seção 9.3.2).

9.3.1 Curto Prazo

A configuração dos parâmetros de visualização para destacar uma região de interesse não é trivial. Além dos parâmetros de iluminação, também é necessário especificar funções de transferência de cor e de opacidade, o que ainda é um tópico de pesquisa e investigação [PFI 2001], conforme descrito no capítulo 4. Portanto, é desejável que um conjunto de parâmetros pré-configurados seja fornecido para que seja necessário apenas selecionar tal conjunto de parâmetros de acordo com a informação que se deseja extrair do volume de dados.

A pesquisa realizada revelou que novas ferramentas podem ser criadas a partir da combinação das estratégias de corte definidas na abordagem apresentada. A tabela 6.1 comprova que ainda existe espaço para a exploração desta nova abordagem de ferramentas de corte combinadas com a visualização de dados multimodais. Por exemplo, a realização de um estudo de caso, em conjunto com um médico especialista, utilizando dois volumes de dados médicos reais, um anatômico e um funcional, de um mesmo paciente, é interessante. Com o acompanhamento do usuário final, neste caso um médico, seria possível identificar extensões e aperfeiçoamentos que poderiam ser realizados. O desenvolvimento e/ou integração de novas técnicas de corte, que consistem na especificação das funções f_a e f_b , podem se tornar necessários durante o estudo da sua aplicabilidade para a investigação de volumes de dados multimodais, considerando que o corte pode ser realizado baseado no conteúdo do voxel de qualquer um dos volumes.

9.3.2 Médio Prazo

Uma proposta de trabalho consiste no desenvolvimento de uma interface interativa para a abordagem unificada das técnicas de corte, de tal forma que o usuário pudesse especificar os valores de a e b de maneira amigável e intuitiva. Tal interface facilitaria a exploração do volume de dados através das ferramentas de corte. Além disso, como a maioria dos sistemas de visualização de dados médicos possuem uma interface complexa para os seus usuários finais (médicos), é necessária uma avaliação de usabilidade [NIE 94] que contribua para o projeto de uma interface amigável para tais sistemas. A consideração da usabilidade durante o processo de desenvolvimento da interface pode evitar vários problemas na utilização destes sistemas.

Também é tópico de futura investigação a combinação da visualização integrada e da abordagem unificada com a exploração de diferentes formas de especificar as funções de transferência de opacidade, e a sua adaptação para aplicação com os algoritmos de *splatting*, *shear-warp* e com a visualização através de mapeamento de textura. Meissner et al. [MEI 2000a] descreveram estes quatro algoritmos de visualização direta de volume (*ray casting*, *splatting*, *shear-warp* e mapeamento de textura 3D por hardware) em um mesmo *framework* conceitual, o que mostra a viabilidade de extensão da abordagem unificada. Neste caso, seria necessário apenas criar extensões da classe *ray casting* para implementação dos novos algoritmos, pois existem etapas comuns.

Outro trabalho a ser desenvolvido consiste na análise de desempenho do al-

goritmo implementado em termos de custo computacional e no desenvolvimento de novos algoritmos que permitam uma interação em tempo real com os volumes de dados. A tendência, neste caso, é a utilização de técnicas de textura 3D ou de multi-texturas 2D, que, aproveitando a capacidade das placas aceleradoras gráficas, possibilitam uma rápida visualização e manipulação do volume de dados. Assim, após o armazenamento do volume de dados em memória, os parâmetros U_a e U_b , em conjunto com as funções f_a e f_b , poderiam ser utilizados para inicializar o *stencil buffer*, que é usado para fazer o corte nesta técnica de visualização [MEI 2000a].

Finalmente, uma última sugestão de trabalho é o desenvolvimento ou utilização de critérios descritos na literatura para analisar a qualidade das imagens obtidas, tanto com as ferramentas de corte, como empregando diferentes funções de transferência.

Anexo Volumes de Dados Sintetizados

Para ilustrar as diferenças nas aplicações de cada técnica de corte, função de transferência e visualização, descritas nos capítulos 3, 4, 5 e 6, foram desenvolvidos alguns experimentos usando como entrada volumes de dados sintetizados, gerados através de um procedimento, de forma controlada e com conteúdo conhecido. A utilização destes volumes possibilita identificar se as imagens geradas estão corretas ou não, e qual é a diferença entre as técnicas empregadas.

Por simplicidade, foi definido que as dimensões dos volumes seriam $100 \times 100 \times 100$, com um byte usado para armazenar um valor de intensidade que varia de 0 a 255. Em todos os volumes foram aplicados filtros de suavização, com o objetivo de reduzir o *aliasing* no momento da visualização. Os volumes são apresentados considerando cortes clássicos usados com imagens médicas (figura A.1).

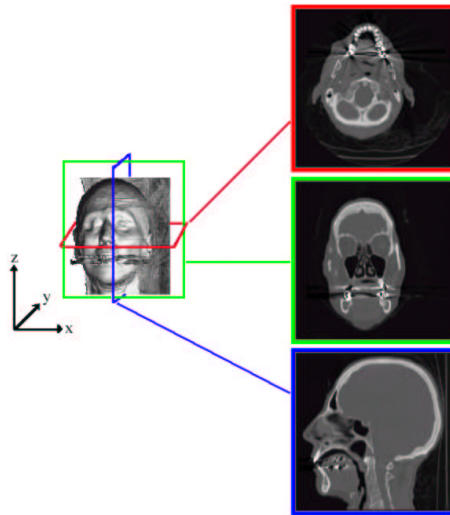


FIGURA A.1 – Cortes axial, coronal e sagital (de cima para baixo)

O conteúdo do primeiro volume consiste em uma esfera cujo raio tem a extensão de 45 voxels, e que possui três camadas diferentes: a mais externa com valor de intensidade 80, a do meio com valor de intensidade zero, e a mais interna com valor de intensidade 255. Além disso, no seu interior também foi colocado um paralelepípedo com três voxels de espessura e intensidade 255. A figura A.3 mostra um exemplo de visualização desta esfera. As duas janelas superiores, da esquerda para a direita, mostram os cortes coronal e sagital, respectivamente, e as janelas inferiores mostram um corte axial e a visualização 3D. As linhas brancas que aparecem nas imagens dos planos ortogonais, equivalem à posição selecionada para os planos. Na visualização 3D foi utilizada apenas uma tabela de cor e uma função de transferência de opacidade linear a partir de 70 (equação 4.4), ou seja, os valores de intensidade menores que 70 foram desprezados (opacidade=0.0). Na tabela de cor o valor de intensidade 0 foi mapeado para a cor preta, 80 foi mapeado para a cor vermelha, e as intensidades entre 240 e 255 foram mapeadas para a cor branca. Valores de intensidade intermediários foram interpolados, conforme ilustra a figura A.2.

O volume de dados ilustrado na figura A.4 também é utilizado nos experimentos. A dimensão e o conteúdo deste volume é semelhante ao volume apresentado

na figura A.3, e os parâmetros de visualização utilizados são os mesmos. A diferença está apenas no interior da esfera, que neste caso contém nove pequenos paralelepípedos centralizados no seu interior, onde seis possuem o mesmo tamanho e três são menores. A densidade dos três paralelepípedos menores é 150, dos paralelepípedos centrais é 225 e dos paralelepípedos de trás, que não estão totalmente alinhados em x com os demais, é 255.

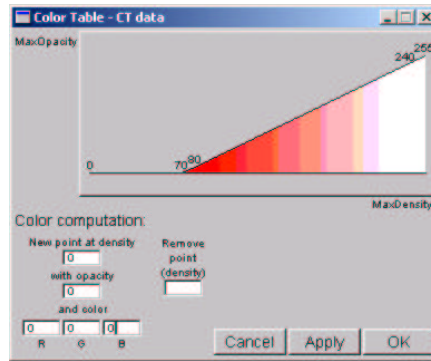


FIGURA A.2 – Tabela de cor usada para gerar as imagens dos volumes sintetizados

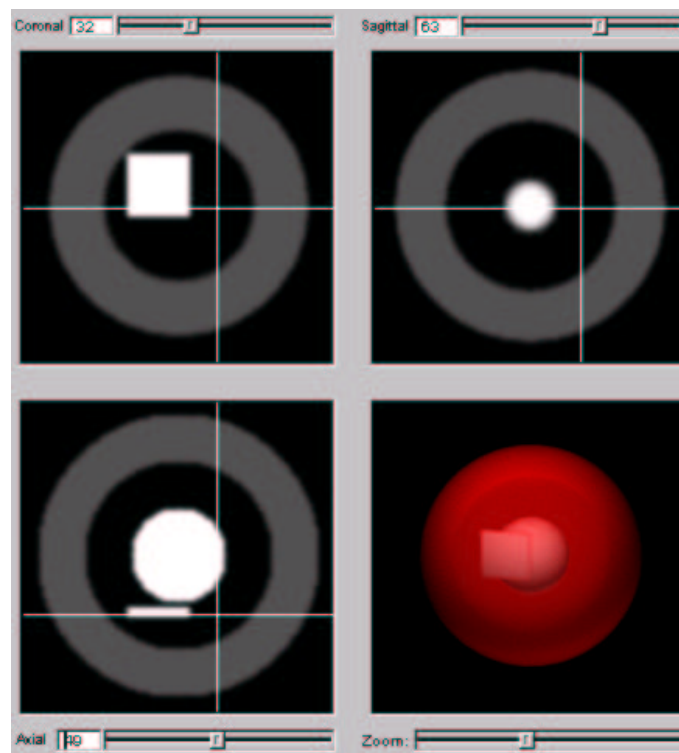


FIGURA A.3 – Visualização do volume de dados que contém uma esfera

Para ilustrar as técnicas de visualização de volumes de dados multimodais (capítulo 5), foi necessário gerar um terceiro volume de dados. Este volume de dados possui as mesmas dimensões dos volumes descritos anteriormente, mas o seu conteúdo é equivalente a um “cubo”, isto é, ele possui várias camadas, cada uma com valores de densidades diferentes. A figura A.5 mostra a visualização deste volume usando os mesmos parâmetros de visualização das figuras A.3 e A.4.

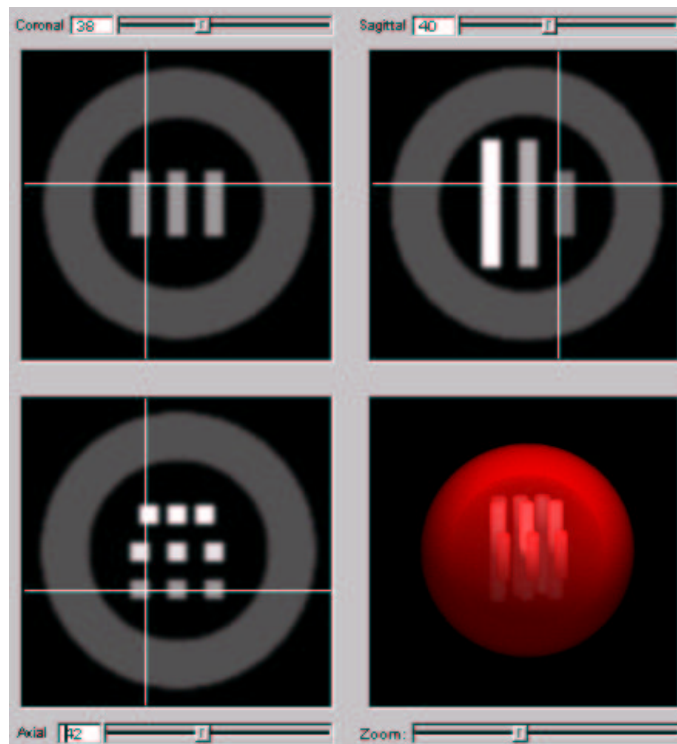


FIGURA A.4 – Visualização do volume de dados que contém vários paralelepípedos no interior de uma esfera

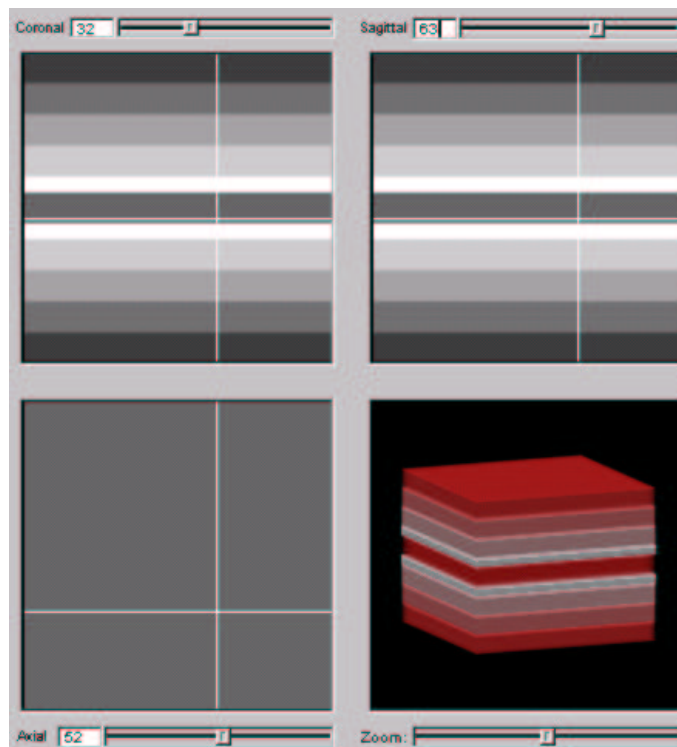


FIGURA A.5 – Visualização do volume de dados que contém um “cubo”

Bibliografia

- [ANE 2002] ANES, M. **Imagens médicas**. Porto Alegre, 01 ago. 2002. Entrevista concedida a Isabel Harb Manssour.
- [APP 68] APPEL, A. **Some techniques for shading machine renderings of solids**. [S.l.]: SJCC, 1968. p.37–45.
- [BÆR 98] BÆRENTZEN, A. Octree-based volume sculpting. In: IEEE VISUALIZATION CONFERENCE, 1998, Research Triangle Park, NC, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1998. Disponível em: <<http://www.imm.dtu.dk/~jab/publications.html>>. Acesso em: ago. 2002.
- [BAJ 97] BAJAJ, C.; PASCUCCI, V.; SCHIKORE, D. The contour spectrum. In: IEEE VISUALIZATION CONFERENCE, 1997, Phoenix, AZ, USA. **Proceedings...** Los Alamitos: IEEE Computer Society and ACM, 1997. p.167–173, 539.
- [BAN 98] BANSAL, R. et al. A novel approach for the registration of 2D portal and 3D CT images for treatment setup verification in radiotherapy. In: MICCAI - INTERNATIONAL CONFERENCE ON MEDICAL IMAGE COMPUTING AND COMPUTER-ASSISTED INTERVENTION, 1998, Cambridge, MA. **Proceedings...** Berlin: Springer-Verlag, 1998. p.1075–1086. (Lecture Notes in Computer Science, v.1496).
- [BER 95] BERGMAN, L.; ROGOWITZ, B.; TREINISH, L. A rule-based tool for assisting colormap selection. In: IEEE VISUALIZATION CONFERENCE, 1995, Atlanta, Georgia, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1995. p.118–125, 444.
- [BOT 2002] BOTHA, C.; POST, F. Interactive previewing for transfer function specification in volume rendering. In: JOINT EUROGRAPHICS - IEEE TCVG SYMPOSIUM ON VISUALIZATION, 2002, Barcelona, Spain. **Proceedings...** [S.l.]: ACM SigGraph, 2002.
- [BRO 2001] BRODLIE, K.; WOOD, J. Recent advances in volume visualization. **Computer Graphics Forum**, Amsterdam, v.20, n.2, p.125–148, 2001.
- [BRO 88] BROGEFORS, G. Hierarchical chamfer matching: a parametric edge matching algorithm. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Los Alamitos, v.10, n.6, p.849–865, 1988.
- [BRO 92] BROWN, L. A survey of image registration techniques. **ACM Computing Surveys**, New York, v.24, n.4, p.325–376, 1992.

- [BUL 2001] BULLITT, E.; AYLWARD, S. Volume rendering of segmented tubular objects. In: MICCAI, 2001. **Proceedings...** [S.l.: s.n.], 2001. p.161 – 168.
- [BUS 96] BUSCHMANN, F. et al. **Pattern-oriented software architecture, a system of patterns**. West Sussex, UK: John Wiley & Sons, 1996.
- [CAI 99] CAI, W.; SAKAS, G. Data intermixing and multi-volume rendering. **Computer Graphics Forum**, Amsterdam, v.18, n.3, p.C359 – C368, 1999. Trabalho apresentado no EUROGRAPHICS, 20., 1999, Milano, Italy.
- [COL 98] COLLIGNON, A. **Multi-modality medical image registration by maximization of mutual information**. 1998. PhD thesis — Katholieke Universiteit Leuven, Belgium.
- [COL 95] COLLIGNON, A. et al. Automated multi-modality image registration based on information theory. In: INFORMATION PROCESSING IN MEDICAL IMAGING, 1995, Dordrecht. **Proceedings...** [S.l.]: Kluwer, 1995. p.263–274.
- [DRE 88] DREBIN, R.; CARPENTER, L.; HANRAHAN, P. Volume rendering. **ACM Computer Graphics**, New York, v.22, n.4, p.125–134, Aug. 1988.
- [EHR 93] EHRICKE, H.; DAIBER, G.; STRASSER, W. The vision camera: an interactive tool for volume data exploration and navigation. In: IEEE VISUALIZATION CONFERENCE, 1993, San Jose, CA, USA. **Proceedings...** [S.l.: s.n.], 1993. p.25–29.
- [ELV 92] ELVINS, T. A survey of algorithms for volume visualization. **ACM Computer Graphics**, New York, v.26, n.3, p.194–201, Aug. 1992.
- [ENG 2001] ENGEL, K.; KRAUS, M.; ERT, T. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In: EUROGRAPHICS / SIGGRAPH WORKSHOP ON GRAPHICS HARDWARE, 2001. **Proceedings...** [S.l.]: Addison-Wesley, 2001. p.9. Disponível em: <<http://wwwvis.informatik.uni-stuttgart.de/eng/research/pub/pub2001/GH2001.pdf>>. Acesso em: ago. 2002.
- [FAN 98] FANG, S.; BIDDLECOME, T.; TUCERYAN, M. Image-based transfer function design for data exploration in volume visualization. In: IEEE VISUALIZATION CONFERENCE, 1998, Research Triangle Park, NC, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1998. p.319–326, 546.
- [FOL 90] FOLEY, J. et al. **Computer graphics: principles and practice**. 2nd ed. New York: Addison Wesley, 1990.

- [FON 97] FONSECA, L.; REIS, L.; MARTHA, L. Uma arquitetura para construção de ferramentas de manipulação para visualização interativa de dados volumétricos. In: SIBGRAPI - SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DE IMAGENS, 1997, Campos do Jordão, SP. **Anais...** [S.l.: s.n.], 1997.
- [FRE 2002] FREITAS, C.; NEDEL, L. **Vpat - visualizing and interacting with virtual patients**. Disponível em: <<http://www.inf.ufrgs.br/cg/vpat/>>. Acesso em: ago. 2002.
- [FUC 77] FUCHS, H.; KEDEM, Z.; USELTON, S. Optimal surface reconstruction from planar contours. **Communications of the ACM**, New York, v.20, n.10, p.693–702, Oct. 1977.
- [FUJ 99] FUJISHIRO, I.; AZUMA, T.; TAKESHIMA, Y. Automating transfer function design for comprehensible volume rendering based on 3D field topology analysis. In: IEEE VISUALIZATION CONFERENCE, 1999, San Francisco, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1999. p.467–563.
- [GAM 95] GAMMA, E. et al. **Design patterns: elements of reusable object-oriented software**. Reading: Addison-Wesley, 1995.
- [GER 2001] GERING, D. et al. An integrated visualization system for surgical planning and guidance using image fusion and an open MR. **Journal of Magnetic Resonance Imaging**, Baltimore, p.967–975, 2001.
- [GON 93] GONZALEZ, R.; WOODS, R. **Digital image processing**. Reading, MA: Addison-Wesley, 1993.
- [GRA 96] GRAÇA, G.; OLABARRIAGA, S. Sistema interativo para localização de cortes em planos arbitrários em volumes tomográficos. In: SEMINARIO INTEGRADO DE SOFTWARE E HARDWARE, 1996, Recife, PE. **Anais...** [S.l.: s.n.], 1996. p.555–562.
- [GRE 90] GREENES, R.; BRINKLEY, J. Radiology systems. In: SHORTLIFFE, E.; PERREAULT, L. (Ed.). **Medical informatics**. USA: Addison-Wesley, 1990. p.324–365.
- [HAS 98] HASTREITER, P.; ERTL, T. Integrated registration and visualization of medical image data. In: COMPUTER GRAPHICS INTERFACE, CGI, 1998, Hannover, Germany. **Proceedings...** [S.l.: s.n.], 1998. Disponível em: <<http://www9.informatik.uni-erlangen.de/Persons/Hastreiter/deutsch.html>>. Acesso em: ago. 2002.
- [HAS 98a] HASTREITER, P. et al. Interactive and intuitive visualization of small and complex vascular structures in MR and CT. In: THE ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGI-

- NEERING IN MEDICINE AND BIOLOGY SOCIETY, 20., 1998, Hong Kong. **Proceedings...** [S.l.: s.n.], 1998. v.2, p.532–535.
- [HAU 2002] HAUSER, H. **Two-level volume rendering**. Disponível em: <<http://www.VRVis.at/vis/research/two-level/>>. Acesso em: ago. 2002.
- [HAU 2000] HAUSER, H. et al. Two-level volume rendering - fusing mip and dvr. In: IEEE VISUALIZATION CONFERENCE, 2000, Salt Lake City, Utah, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2000. p.211–218.
- [HAU 2001] HAUSER, H. et al. Two-level volume rendering. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, CA, v.7, n.3, p.242–252, July/Sept. 2001.
- [HE 96] HE, T. et al. Generation of transfer functions with stochastic search techniques. In: IEEE VISUALIZATION CONFERENCE, 1996, San Francisco, CA, USA. **Proceedings...** [S.l.: s.n.], 1996. p.227–234, 489.
- [HLA 2000] HLADÚVKA, J.; KÖNIG, A.; GRÖLLER, M. Curvature-based transfer functions for direct volume rendering. In: SPRING CONFERENCE ON COMPUTER GRAPHICS, 2000. **Proceedings...** [S.l.: s.n.], 2000. p.58–65. Disponível em: <<http://www.cg.tuwien.ac.at/research/vis/vismed/CurvatureTF/>>. Acesso em: ago. 2002.
- [HLA 2001] HLADÚVKA, J.; KÖNIG, A.; GRÖLLER, M. Salient representation of volume data. In: DATA VISUALIZATION, 2001. **Proceedings...** [S.l.: s.n.], 2001. p.203–211, 351. Disponível em: <<http://www.cg.tuwien.ac.at/research/vis/vismed/SalientRepresentation/>>. Acesso em: ago. 2002.
- [PRO 98] IEEE SYMPOSIUM ON VOLUME VISUALIZATION, 1998. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1998.
- [INC 2002] INC, A. V. S. **AVS in the medical industry: designing new applications for medicine**. Disponível em: <<http://www.addlink.es/docs/avs/>>. Acesso em: jul. 2002.
- [JAN 2001] JANKUN-KELLY, T.; MA, K.-L. Visualization exploration and encapsulation via a spreadsheet-like interface. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, v.7, n.3, p.275–287, July-Sept. 2001. Disponível em: <<http://muldoon.cipic.ucdavis.edu/research/Spreadsheets.shtml/>>. Acesso em: ago. 2002.
- [JEA 92] JEAN, Y.; HODGES, L. A method for interactive manipulation and animation of volumetric data. In: SPIE - VISUALIZATION

IN BIOMEDICAL COMPUTING, 1992, Chapel Hill, NC, USA. **Proceedings...** [S.l.: s.n.], 1992. p.453–461.

- [JOH 99] JOHNSON, C. et al. Interactive simulation and visualization. **Computer**, Los Alamitos, v.32, n.12, p.59–65, 1999.
- [JON 95] JONES, M.; CHEN, M. Fast cutting operations on three dimensional volume datasets. **Visualization in Scientific Computing**, Berlin, p.1–8, 1995.
- [KAN 2001] KANITSAR, A. et al. Computed tomography angiography: a case study of peripheral vessel investigation. In: IEEE VISUALIZATION CONFERENCE, 2001, San Diego, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2001. p.477–480.
- [KAU 91] KAUFMAN, A. **Volume visualization**. Los Alamitos, CA: IEEE Computer Society Press, 1991.
- [KAU 2000] KAUFMAN, A. **Volume visualization: principles and advances** (course notes: volume graphics). Bonn-Röttgen, Germany: [s.n.], 2000. Disponível em: <<http://www.merl.com/people/pfister/courses/Bonn2000/Syllabus.htm>>. Acesso em: ago. 2002.
- [KAU 93] KAUFMAN, A.; COHEN, D.; YAGEL, R. Volume graphics. **Computer**, Los Alamitos, v.26, n.7, p.51–64, 1993.
- [KAU 90] KAUFMAN, A. et al. A survey of architectures for volume rendering. **IEEE Engineering in Medicine and Biology Magazine**, New York, v.9, n.4, p.18–23, Dec. 1990.
- [KEP 75] KEPPEL, E. Approximating complex surfaces by triangulation of contour lines. **IBM Journal of Research and Development**, New York, v.19, n.1, p.2–11, 1975.
- [KIM 2001] KIM, K.; WITTENBRINK, C. M.; PANG, A. Extended specifications and test data sets for data level comparisons of direct volume rendering algorithms. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, v.7, n.4, p.299–317, Oct.-Dec. 2001.
- [KIN 2001] KINDLMANN, G. **Transfer functions for direct volume rendering (tutorial presented in IEEE visualization conference 2001)**. Disponível em: <<http://visual.nlm.nih.gov/tutorials/vis2001/>>. Acesso em: jul. 2002.
- [KIN 98] KINDLMANN, G.; DURKIN, J. Semi-automatic generation of transfer functions for direct volume rendering. In: IEEE SYMPOSIUM ON VOLUME VISUALIZATION AND GRAPHICS, 1998,

- Research Triangle Park, NC, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1998. p.79–86.
- [KIN 2001a] KING, D.; WITTENBRINK, C. M.; WOLTERS, H. J. An architecture for interactive tetrahedral volume rendering. In: VG01 - INTERNATIONAL WORKSHOP ON VOLUME GRAPHICS, 2001, Stony Brook, USA. **Proceedings...** Wien: Springer-Verlag, 2001. p.163–180.
- [KNI 2001] KNISS, J.; KINDLMANN, G.; HANSEN, C. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In: IEEE VISUALIZATION CONFERENCE, 2001, San Diego, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2001. p.255–262. Disponível em: <<http://www.cs.utah.edu/~jmk/papers/vis01/>>. Acesso em: ago. 2002.
- [KNI 2002] KNISS, J.; KINDLMANN, G.; HANSEN, C. Multi-dimensional transfer functions for interactive volume rendering. **IEEE Transactions on Visualization and Computer Graphics**, Piscataway, 2002. Disponível em: <http://www.cs.utah.edu/~jmk/kniss_tv02-small.pdf>. Acesso em: ago. 2002.
- [KÖN 2001] KÖNIG, A.; GRÖLLER, M. Mastering transfer function specification by using VolumePro technology. In: SPRING CONFERENCE ON COMPUTER GRAPHICS, 2001. **Proceedings...** [S.l.: s.n.], 2001. p.279–286. Disponível em: <<http://www.cg.tuwien.ac.at/research/vis/vismed/ATFSpec/>>. Acesso em: ago. 2002.
- [KRE 98] KREEGER, K. et al. Adaptive perspective ray casting. In: IEEE SYMPOSIUM ON VOLUME VISUALIZATION AND GRAPHICS, 1998, Research Triangle Park, NC, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1998. p.19–20.
- [LAC 94] LACROUTE, P.; LEVOY, M. Fast volume rendering using a shear-warp factorization of the viewing transformation. In: SIGGRAPH - INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 1994. **Proceedings...** [S.l.: s.n.], 1994. p.451–458.
- [LAR 97] LARMAN, C. **Applying uml and patterns - an introduction to object-oriented analysis and design**. Upper Saddle River, NJ: Prentice-Hall, 1997.
- [LEV 97] LEVENTON, M. **A registration, tracking, and visualization system for image-guided surgery**. 1997. Dissertação (Mestrado em Ciência da Computação) — MIT - Massachusetts Institute of Technology, Cambridge, Massachusetts - US. Disponível

em: <<http://www.ai.mit.edu/people/leventon/Research/SM-Thesis/thesis.pdf>>. Acesso em: ago. 2002.

- [LEV 98] LEVENTON, M.; GRIMSON, W. Multi-modal volume registration using joint intensity distributions. In: MICCAI - INTERNATIONAL CONFERENCE ON MEDICAL IMAGE COMPUTING AND COMPUTER-ASSISTED INTERVENTION, 1998, Cambridge, MA. **Proceedings...** Berlin: Springer-Verlag, 1998. p.1057–1066. (Lecture Notes in Computer Science, v.1496).
- [LEV 89] LEVIN, D. et al. The brain: integrated three-dimensional display of MR and pet images. **Radiology**, Syracuse, v.172, n.3, p.783–789, 1989.
- [LEV 91] LEVKOWITZ, H. et al. Color vs. black-and-white in visualization. In: IEEE VISUALIZATION CONFERENCE, 1991. **Proceedings...** [S.l.: s.n.], 1991. p.336–339.
- [LEV 88] LEVOY, M. Volume rendering - display of surfaces from volume data. **IEEE Computer Graphics and Applications**, Los Alamitos, v.8, n.3, p.29 – 37, 1988.
- [LEV 90] LEVOY, M. Efficient ray-tracing of volume data. **ACM Transactions on Graphics**, New York, v.9, n.3, p.245–261, 1990.
- [LEV 90a] LEVOY, M. Volume rendering by adaptive refinement. **The Visual Computer**, New York, v.6, n.1, p.2–7, Feb. 1990.
- [LEV 2000] LEVOY, M. Error in volume rendering paper was in exposition only. **IEEE Computer Graphics and Applications**, Los Alamitos, v.20, n.4, p.6, July/Aug. 2000.
- [LIC 98] LICHTENBELT, B.; CRANE, R.; NAQVI, S. **Introduction to volume rendering**. Upper Saddle River, NJ: Prentice Hal, 1998.
- [LIU 99] LIU, B. et al. A simplified energy projection and applications of transfer function in isometric volume rendering. In: IEEE CANADIAN CONFERENCE ON ELECTRICAL AND COMPUTER ENGINEERING, 1999. **Proceedings...** [S.l.: s.n.], 1999. v.3, p.1728–1733.
- [LOR 93] LORENSEN, W. Geometric clipping using boolean textures. In: IEEE VISUALIZATION CONFERENCE, 1993, San Jose, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1993. p.268–274.
- [LOR 87] LORENSEN, W.; CLINE, H. Marching cubes: a high resolution 3D surface construction algorithm. **Computer Graphics**, New York, v.21, n.4, p.163–169, July 1987.

- [LUC 92] LUCAS, B. A scientific visualization renderer. In: IEEE VISUALIZATION CONFERENCE, 1992, Boston, MA, USA. **Proceedings...** [S.l.: s.n.], 1992. p.227–234.
- [MA 99] MA, K.-L. Image graphs: a novel approach to visual data exploration. In: IEEE VISUALIZATION CONFERENCE, 1999, San Francisco, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1999. p.81–88.
- [MA 99a] MA, K.-L. **Image graphs:** a novel approach to visual data exploration. Disponível em: <<http://graphics.cs.ucdavis.edu/research/Image-Graphs.shtml>>. Acesso em: jul. 2002.
- [MAC 95] MACHIRAJU, R.; YAGEL, R. Accuracy control of reconstruction errors in volume slicing. In: FRONTIERS IN BIOMEDICAL VISUALIZATION SYMPOSIUM, 1995, Atlanta, GA. **Proceedings...** [S.l.: s.n.], 1995. p.50–57.
- [MAC 2002] MACIEL, A.; NEDEL, L.; FREITAS, C. Anatomy based joint models for virtual humans skeletons. In: IEEE COMPUTER ANIMATION, 2002, Geneva, Switzerland. **Proceedings...** [S.l.: s.n.], 2002. p.110–116.
- [MAE 98] MAES, F. **Segmentation and registration of multimodal medical images.** 1998. PhD thesis — Katholieke Universiteit Leuven, Belgium.
- [MAE 97] MAES, F. et al. Multimodality image registration by maximization of mutual information. **IEEE Transactions on Medical Imaging**, New York, v.16, n.2, p.187–198, 1997.
- [MAG 91] MAGUIRE, G. et al. Graphics applied to medical image registration. **IEEE Computer Graphics and Applications**, Los Alamitos, v.11, n.2, p.20–28, Mar./Apr. 1991.
- [MAI 96] MAINTZ, J. **Retrospective registration of tomographic brain images.** 1996. PhD thesis — Universiteit Utrecht, Netherlands.
- [MAI 98] MAINTZ, J.; VIERGEVER, M. Survey of medical image registration. **Medical Image Analysis**, Amsterdam, v.2, n.1, p.1–36, 1998.
- [MAN 94] MANSSOUR, I. **Algoritmo de ray-casting para visualização volumétrica de dados obtidos por tomografia computadorizada.** Porto Alegre, RS: Instituto de Informática, UFRGS, 1994. Trabalho Individual (Mestrado em Ciência da Computação).
- [MAN 2000] MANSSOUR, I. et al. An architecture for interactive multimodal visualization system. In: CGIM - INTERNATIONAL CON-

- ERENCE COMPUTER GRAPHICS AND IMAGING, 2000, Las Vegas, USA. **Proceedings...** [S.l.: s.n.], 2000. p.26–31.
- [MAN 2000a] MANSSOUR, I. et al. A multimodal visualization framework for medical data. In: SIBGRAPI - SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DE IMAGENS, 2000, Gramado/RS, Brasil. **Anais...** Los Alamitos: IEEE Computer Society Press, 2000. p.356.
- [MAN 2001] MANSSOUR, I. et al. A framework to visualize and interact with multimodal medical images. In: VG01 - INTERNATIONAL WORKSHOP ON VOLUME GRAPHICS, 2001, Stony Brook, USA. **Proceedings...** Wien: Springer-Verlag, 2001. p.385–398.
- [MAN 2002] MANSSOUR, I. et al. Visualizing inner structures in multimodal volume data. In: SIBGRAPI - SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DE IMAGENS, 2002. **Anais...** Los Alamitos: IEEE Computer Society Press, 2002.
- [MAN 94a] MANSSOUR, I.; FREITAS, C. Utilização do iris explorer para visualização de dados tomográficos. In: SUPERCOMP - SEMINÁRIO DE SUPERCOMPUTAÇÃO APLICADA, 1994, Porto Alegre, RS. **Anais...** [S.l.: s.n.], 1994. p.324–327.
- [MAN 2002a] MANSSOUR, I.; FREITAS, C. Visualização volumétrica. **Revista de Informática Teórica e Aplicada**, Porto Alegre, v.9, n.2, 2002.
- [MAN 92] MANSSOUR, I.; JUNGBLUT, V. **Sistema para visualização volumétrica de imagens de tomografia computadorizada e ressonância magnética**. Porto Alegre: Faculdade de Informática, PUC-RS, 1992. Trabalho de Conclusão (Bacharel em Informática).
- [MAR 97] MARKS, J.; ANDALMAN, B.; BEARDSLEY, P. A. Design galleries: a general approach to setting parameters for computer graphics and animation. In: SIGGRAPH - INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 1997. **Proceedings...** [S.l.: s.n.], 1997. p.389–400. Disponível em: <<http://visual.nlm.nih.gov/tutorials/vis2000/pdf/dg.pdf>>. Acesso em: jul. 2002.
- [MAT 93] MATTOS, P. de. **Um estudo sobre reconstrução tomográfica a partir de projeções**. Porto Alegre: Instituto de Informática, UFRGS, 1993. Trabalho Individual (Mestrado em Ciência da Computação).
- [MAY 2002] MAYO, C. **Software:** analyze program. Disponível em: <<http://www.mayo.edu/bir/Software/Analyze/Analyze.html>>. Acesso em: ago. 2002.

- [MEI 2000] MEISSNER, M. et al. A practical evaluation of popular volume rendering algorithms. In: IEEE SYMPOSIUM ON VOLUME VISUALIZATION AND GRAPHICS, 2000, Salt Lake City, Utah, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2000.
- [MEI 2000a] MEISSNER, M. et al. **Volume visualization and volume rendering techniques (tutorial presented in eurographics conference 2000)**. Disponível em: <<http://www.gris.uni-tuebingen.de/~meissner/tutorials/tutorial.pdf>>. Acesso em: ago. 2002.
- [MEI 99] MEISSNER, M.; HOFFMANN, U.; STRASSER, W. Enabling classification and shading for 3D texture mapping based volume rendering using opengl and extensions. In: IEEE VISUALIZATION CONFERENCE, 1999, San Francisco, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1999. p.207–214. Disponível em: <<http://www.gris.uni-tuebingen.de/~meissner/publications/pdf/vis99.pdf>>. Acesso em: ago. 2002.
- [MÜE 2001] MÜELLER, K.; KAUFMAN, A. In: VG01 - INTERNATIONAL WORKSHOP ON VOLUME GRAPHICS, 2001. **Proceedings...** Wien: Springer-Verlag, 2001. p.429.
- [MÜE 99] MÜELLER, K.; MÖLLER, T.; CRAWFIS, R. Splatting without the blur. In: IEEE VISUALIZATION CONFERENCE, 1999, San Francisco, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1999. p.363–370.
- [MUL 2000] MULLICK, R.; BRYAN, N.; BUTMAN, J. Confocal volume rendering: fast segmentation-free visualization of internal structures. In: INTERNATIONAL CONFERENCE ON INFORMATION PROCESSING IN MEDICAL IMAGING, 16., 2000, Information processing in medical imaging. **Proceedings...** [S.l.: s.n.], 2000. p.70–76.
- [NED 99] NEDEL, L.; MANSSOUR, I.; FREITAS, C. **Computer graphics and medicine**. Campinas/SP: [s.n.], 1999. Tutorial apresentado no SIBGRAPI, 1999. Disponível em: <<http://www.inf.ufrgs.br/manssour/tutorial-cg&m.pdf>>. Acesso em: ago. 2002.
- [NIE 94] NIELSEN, J.; MACK, R. **Usability inspection methods**. New York, NY: John Wiley & Sons, 1994.
- [NIE 90] NIELSON, G.; HAMANN, B. Techniques for the interactive visualization of volumetric data. In: IEEE VISUALIZATION CONFERENCE, 1990, San Francisco, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1990. p.45–50.

- [NOO 97] NOORDMANS, H. **Interactive analysis of 3D microscope images**. 1997. PhD thesis — Universiteit Utrecht, Netherlands.
- [NOO 2002] NOORDMANS, H. **Physically realistic visualization of embedded volume structures for medical image data**. Disponível em: <<http://www.isi.uu.nl/Research/Visualization/visualization-frame.html>>. Acesso em: ago. 2002.
- [NOO 2001] NOORDMANS, H. et al. Volume rendering for neurosurgery. In: IMIVA - WORKSHOP ON INTERACTIVE MEDICAL IMAGE VISUALIZATION AND ANALYSIS, 2001, Utrecht. **Proceedings...** [S.l.: s.n.], 2001.
- [NOO 2000] NOORDMANS, H.; VOORT, H. van der; SMEULDERS, A. Spectral volume rendering. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, CA, v.6, n.3, p.196–207, July/Sept. 2000.
- [OLA 99] OLABARRIAGA, S. **Human-computer interaction for the segmentation of medical images**. 1999. PhD thesis — Universiteit Utrecht, Netherlands.
- [OSB 97] OSBORNE, R. et al. Em-cube: an architecture for low-cost real-time volume rendering. In: SIGGRAPH/EUROGRAPHICS WORKSHOP ON GRAPHICS HARDWARE, 1997, Los Angeles, CA. **Proceedings...** [S.l.: s.n.], 1997. Disponível em: <<http://www.merl.com/people/pfister/pubs/ws97.pdf>>. Acesso em: ago. 2002.
- [PAT 98] PATTEN, J.; MA, K.-L. A graph based interface for representing volume visualization results. In: GRAPHICS INTERFACE, 1998, Vancouver, B.C. **Proceedings...** [S.l.: s.n.], 1998. Disponível em: <<http://www.graphicsinterface.org/proceedings/1998/121/>>. Acesso em: ago. 2002.
- [PAY 90] PAYNE, B.; TOGA, A. Surface mapping brain function on 3D models. **IEEE Computer Graphics and Applications**, Los Alamitos, v.10, n.5, p.33–41, 1990.
- [PEK 2001] PEKAR, V.; WIEMKER, R.; HEMPEL, D. Fast detection of meaningful isosurfaces for volume data visualization. In: IEEE VISUALIZATION CONFERENCE, 2001, San Diego, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2001. p.223–230.
- [PET 96] PETERS, T. et al. Three-dimensional multimodal image-guidance for neurosurgery. **IEEE Transactions on Medical Imaging**, New York, v.15, n.2, p.121–128, Apr. 1996.
- [PFI 99] PFISTER, H. et al. The volumepro real-time ray-casting system. In: SIGGRAPH - INTERNATIONAL CONFERENCE ON COM-

- PUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 1999, Los Angeles, CA. **Proceedings...** [S.l.: s.n.], 1999. p.251–260.
- [PFI 2001] PFISTER, H. et al. The transfer function bake-off. **IEEE Computer Graphics and Applications**, Los Alamitos, CA, v.21, n.3, p.16–22, May/June 2001.
- [PFL 2000] PFLESSER, B. et al. Volume based planning and rehearsal of surgical interventions. In: CARS - COMPUTER ASSISTED RADIOLOGY AND SURGERY, 2000. **Proceedings...** Amsterdam: Elsevier, 2000. p.607–612. Disponível em: <<http://www.uke.uni-hamburg.de/institute/imdm/idv/publikationen/cars2000/>>. Acesso em: ago. 2002.
- [PLU 2002] PLUIM, J.; MAINTZ, J. **Registration of multimodal medical images**. Disponível em: <<http://www.isi.uu.nl/Research/Registration/registration-frame.html>>. Acesso em: ago. 2002.
- [RAY 99] RAY, H. et al. Ray casting architectures for volume visualization. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, CA, v.5, n.3, p.210–223, July/Sept. 1999.
- [REZ 2000] REZK-SALAMA, C. et al. Interactive volume rendering on standard pc graphics hardware using multi-textures and multi-stage rasterization. In: SIGGRAPH/EUROGRAPHICS WORKSHOP ON GRAPHICS HARDWARE, 2000, Interlaken, Switzerland. **Proceedings...** [S.l.: s.n.], 2000. p.21–22.
- [RHE 2001] RHEINGANS, P.; EBERT, D. Volume illustration: nonphotorealistic rendering of volume models. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, CA, v.7, n.3, p.253–264, July/Sept. 2001.
- [RHO 80] RHODES, M.; GLENN, W.; AZZAWI, Y. Extracting oblique planes from serial CT sections. **Journal of Computer Assisted Tomography**, Philadelphia, v.5, n.5, p.649–657, Oct. 1980.
- [ROB 94] ROBB, R. Visualization methods for analysis of multimodality images. In: THATCHER, R. et al. (Ed.). **Functional neuroimaging: technical foundations**. San Diego, CA: Academic Press, 1994. p.324–365.
- [ROB 98] ROBB, R. **Three-dimensional biomedical imaging: principles and practice**. New York, NY: John Wiley & Sons, 1998.
- [ROB 89] ROBB, R.; BARILLOT, C. Interactive display and analysis of 3-d medical images. **IEEE Transactions on Medical Imaging**, New York, v.8, n.3, p.217–226, Sept. 1989.

- [SAN 96] SANTOS, B. et al. On assessing the error introduced in volume data visualization by direct volume rendering methods. In: ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY, 18., 1996. **Proceedings...** [S.l.: s.n.], 1996. v.2, p.646–647.
- [SCH 2002] SCHIEMANN, T.; HÖHNE, K. **Research projects - volume segmentation**. Disponível em: <<http://www.uke.uni-hamburg.de/institute/imdm/idv/forschung/segmentation/>>. Acesso em: ago. 2002.
- [SCH 2001] SCHUSSMAN, G.; MAX, N. Hierarchical perspective volume rendering using triangle fans. In: VG01 - INTERNATIONAL WORKSHOP ON VOLUME GRAPHICS, 2001, Stony Brook, USA. **Proceedings...** Wien: Springer-Verlag, 2001. p.307–320.
- [SIL 2000] SILVA, M. **Ray casting based volume visualization techniques**. 2000. 146p. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.
- [SPI 2002] SPITZAK, B. **Fltk - the fast light tool kit home page**. Disponível em: <<http://ftk.easysw.com>>. Acesso em: jul. 2002.
- [STE 98] STEWART, N.; LEACH, G.; JOHN, S. An improved z-buffer csg rendering algorithm. In: EUROGRAPHICS/SIGGRAPH WORKSHOP ON GRAPHICS HARDWARE, 1998, Lisbon, Portugal. **Proceedings...** [S.l.: s.n.], 1998. p.25–30. Disponível em: <<http://www.nigels.com/research/egsggh98.pdf>>. Acesso em: ago. 2002.
- [STO 98] STOKKING, R. **Integrated visualization of functional and anatomical brain images**. 1998. PhD thesis — Universiteit Utrecht, Netherlands.
- [STO 94] STOKKING, R. et al. Integrated visualization of SPECT and MR images for frontal lobe damaged regions. In: SPIE - VISUALIZATION IN BIOMEDICAL COMPUTING, 1994. **Proceedings...** [S.l.: s.n.], 1994. p.282–290.
- [SUM 2001] SUMMERS, R. et al. Confocal volume rendering of the thorax. In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 2001, Thessaloniki, Greece. **Proceedings...** [S.l.: s.n.], 2001. v.2, p.297–299.
- [TCL 2002] TCL. **Tcl developer xchange**. Disponível em: <<http://www.tcl.tk/>>. Acesso em: ago. 2002.
- [TEN 2001] TENGINAKAI, S.; JINHO, L.; MACHIRAJU, R. Salient iso-surface detection with model-independent statistical signatures. In: IEEE VISUALIZATION CONFERENCE, 2001, San Diego, CA,

- USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2001. p.231–238.
- [TIE 96] TIEDE, U.; SCHIEMANN, T.; HÖHNE, K. Visualizing the visible human. **IEEE Computer Graphics and Applications**, Los Alamitos, CA, v.16, n.1, p.7–9, 1996.
- [UDU 2000] UDUPA, J.; HERMAN, G. **3D imaging in medicine**. 2nd ed. Danvers, MA: CRC Press LLC, 2000.
- [UDU 93] UDUPA, J.; ODHNER, D. Shell rendering. **IEEE Computer Graphics and Applications**, Los Alamitos, v.13, n.6, p.58–67, 1993.
- [UPS 89] UPSON, C. et al. The application visualization system: a computational environment for scientific visualization. **IEEE Computer Graphics and Applications**, Los Alamitos, v.9, n.4, p.30–42, 1989.
- [UPS 88] UPSON, C.; KEELER, M. V-buffer: visible volume rendering. **Computer Graphics**, New York, v.22, n.4, p.59–64, Aug. 1988.
- [VEM 97] VEMURI, B. et al. A robust and efficient algorithm for image registration. In: IPMI - INTERNATIONAL CONFERENCE ON INFORMATION PROCESSING IN MEDICAL IMAGING, 15., 1997, Poultney, Vermont. **Proceedings...** Berlin: Springer-Verlag, 1997. p.465–470. (Lecture Notes in Computer Science, v.1230).
- [VIO 95] VIOLA, P.; III, W. W. Alignment by maximization of mutual information. In: INTERNATIONAL CONFERENCE ON COMPUTER VISION, 1995. **Proceedings...** [S.l.: s.n.], 1995. p.16–23.
- [WAL 94] WALL, S. Imaging. In: TIERNEY, L. M.; MCPHEE, S. J.; PAPADAKIS, M. A. (Ed.). **Current medical diagnosis & treatment**. Norwalk, Connecticut: Appleton & Lange, 1994. p.23–30.
- [WEI 2000] WEINSTEIN, D. Scanline surfacing: building separating surfaces from planar contours. In: IEEE VISUALIZATION CONFERENCE, 2000, Salt Lake City, Utah, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2000. p.283–289.
- [WER 94] WERNECKE, J. **The inventor mentor**: programming object-oriented 3D graphics with open inventor. [S.l.]: Addison-Wesley, 1994.
- [WES 98] WESTERMANN, R.; ERTL, T. Efficiently using graphics hardware in volume rendering applications. In: SIGGRAPH - INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 1998. **Proceedings...** [S.l.: s.n.], 1998. p.169–177. Disponível em: <<http://wwwvis.informatik.uni->

stuttgart.de/ger/research/pub/pub1998/SIGGRAPH98.ps.gz>.
Acesso em: ago. 2002.

- [WES 89] WESTOVER, L. Interactive volume rendering. In: WORKSHOP ON VOLUME VISUALIZATION, 1989, Chapel Hill, NC. **Proceedings...** Chapel Hill: University of North Carolina Press, 1989. p.9–16.
- [WES 90] WESTOVER, L. Footprint evaluation for volume rendering. In: SIGGRAPH - INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 1990. **Proceedings...** [S.l.: s.n.], 1990. p.367–376.
- [WIL 91] WILHELME, J.; GELDER, A. V. A coherent projection approach for direct volume rendering. In: SIGGRAPH - INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 1991, Las Vegas, Nevada. **Proceedings...** [S.l.: s.n.], 1991. p.275–284.
- [WIL 96] WILLIAMS, P.; USELTON, S. **Foundations for measuring volume rendering quality**. [S.l.]: Nasa Ames Research Center, 1996. NAS-96-021.
- [WIT 98] WITTENBRINK, C.; MALZBENDER, T.; GOSS, M. Opacity-weighted color interpolation for volume sampling. In: IEEE SYMPOSIUM ON VOLUME VISUALIZATION AND GRAPHICS, 1998, Research Triangle Park, NC. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1998. p.135–142.
- [YEN 96] YEN, S. Y.; NAPEL, S.; RUBIN, G. Fast sliding thin slab volume visualization. In: IEEE SYMPOSIUM ON VOLUME VISUALIZATION AND GRAPHICS, 1996, San Francisco, CA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1996. p.79–86.
- [YOO 91] YOO, T. et al. Achieving direct volume visualization with interactive semantic region selection. In: IEEE VISUALIZATION CONFERENCE, 1991. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1991. p.58–65, 410.
- [ZUI 95] ZUIDERVELD, K. **Visualization of multimodality medical volume data using object-oriented methods**. 1995. 184p. PhD thesis — Universiteit Utrecht, Netherlands.
- [ZUI 96] ZUIDERVELD, K. et al. Multimodality visualization of medical volume data. **Computer and Graphics**, Oxford, v.20, n.6, p.775–791, 1996.