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

FRANCISCO DE MOURA PINTO

Especificação de Funções de Transferência Unidimensionais e Multidimensionais para Visualização Volumétrica Direta

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

Profa. Dra. Carla Maria Dal Sasso Freitas
Orientadora

Porto Alegre, maio de 2007

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Pinto, Francisco de Moura

Especificação de Funções de Transferência Unidimensionais e Multidimensionais para Visualização Volumétrica Direta / Francisco de Moura Pinto. – Porto Alegre: PPGC da UFRGS, 2007.

82 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2007. Orientadora: Carla Maria Dal Sasso Freitas.

1. Funções de transferência. 2. Visualização volumétrica. 3. Imagens 3D. 4. Computação gráfica. 5. Interação com dados volumétricos. I. Freitas, Carla Maria Dal Sasso. II. Título.

"If we knew what it was we were doing,
it would not be called research, would it?"
— ALBERT EINSTEIN

Dedico minha dissertação à minha namorada, Adriana, por sua paciência nas vezes em que tornei literal meu compromisso de dedicação exclusiva com o programa de mestrado, aos meus colegas de pós-graduação, pela oportunidade de convívio estimulante com mentes cuja perspicácia revela-se em contextos muito mais abrangentes que o da simples ciência da computação, e à minha família e amigos, porque cada passo da minha vida é, pelo menos em parte, reflexo do seu apoio e influência. — FRANCISCO PINTO

AGRADECIMENTOS

Agradeço ao CNPq pelo apoio financeiro, indispensável à minha permanência no programa de mestrado, e a todos os responsáveis pela manutenção e pelo aprimoramento, sob todos os seus aspectos, da Universidade Federal do Rio Grande do Sul por toda a infra-estrutura de que dispus durante o mestrado. Também agradeço aos professores do mestrado pela valiosa instrução que recebi, à Carla, pela orientação agradável e com liberdade, no entanto sem omissão, e aos professores da graduação em Engenharia da Computação, da Fundação Universidade Federal do Rio Grande, onde estudei, pelo conhecimento e experiências adquiridos no caminho que tracei até aqui. Devo ainda reconhecer a importância do sistema educacional público brasileiro, do qual desfrutei quase ininterruptamente desde minha primeira experiência escolar, porque, apesar dos exemplos de permissividade e negligência, ainda existem bons profissionais em todos os níveis de educação.

Por fim, é importante salientar o apoio do grupo de Computação Gráfica da UFRGS na realização desse trabalho. Agradeço a todo o grupo pelas sugestões expostas e pelo conhecimento compartilhado, especialmente aos que se apresentaram como cobaias dos experimentos de avaliação do trabalho, ao João Luis Prauchner, pela cessão da ferramenta desenvolvida em seu trabalho de mestrado, e ao Fausto Richetti Blanco, pela ajuda com a implementação das funções de base radial.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

2D	Bidimensional/Two-dimensional
3D	Tridimensional/Three-dimensional
nD	n-dimensional
BMU	Best Matching Unit
CPU	Central Processing Unit
CT	Computed Tomography
DVR	Direct Volume Rendering
FT	Função de Transferência
GPU	Graphics Processing Unit
ICA	Independent Component Analysis
LH	Low and High
LUT	Lookup Table
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
nf	Neighborhood Function
PCA	Principal Components Analysis
RBF	Radial Basis Function
RGB	Red, Green and Blue
RGBA	Red, Green, Blue and Alpha
SOM	Self-Organizing Maps
TF	Transfer Function

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

O uso de dados volumétricos é bastante comum em diversas áreas da ciência, como Medicina, Física e Meteorologia. São exemplos típicos os dados provenientes de dispositivos de tomografia computadorizada ou ressonância magnética e os obtidos através de estimação de fenômenos físicos pelo uso de sensores diversos ou de simulação numérica. Tais dados apresentam-se, freqüentemente, sob a forma de uma grade tridimensional regular, onde cada elemento possui um valor escalar ou multidimensional (uma tupla de valores). Outras topologias também podem ser usadas para exprimir a disposição espacial dos valores. A visualização de dados volumétricos, importante na compreensão destes, é um processo não-trivial e, em decorrência, diversas técnicas foram propostas para abordar o problema.

Visualização direta de volumes é uma abordagem em crescente popularização que representa visualmente os dados, conservando sua estrutura tridimensional, sem extrair geometrias intermediárias. Esse processo exige o mapeamento dos atributos dos elementos de volume para propriedades ópticas, permitindo a geração de imagens através da aplicação de um algoritmo de visualização, que pode implementar um modelo de iluminação. Tal mapeamento é definido por uma função, conhecida como função de transferência, que determina valores de atributos ópticos para cada valor encontrado no volume. Essa função desenvolve, portanto, um importante papel na visualização, pois define a visibilidade das estruturas presentes no volume — normalmente valendo-se do atributo opacidade — e também o aspecto destas na imagem final. Contudo, a definição de uma boa função de transferência, capaz de produzir imagens informativas, é um processo complexo que deve ser simplificado com o apoio de ferramentas adequadas. A simples especificação manual de uma função de transferência é um processo iterativo de tentativa e erro, em decorrência da dificuldade de compreensão do relacionamento entre a função utilizada e a imagem gerada, especialmente quando se trata de dados multidimensionais, que implicam funções de transferência com maior número de dimensões.

Diante da necessidade de agilizar e simplificar a especificação de funções de transferência, abordagens semi-automáticas e automáticas para geração de funções foram propostas, exigindo do usuário esforço de interação reduzido ou nulo. Entretanto, as propostas existentes deixam a desejar na simplicidade, interatividade ou flexibilidade. O presente trabalho propõe técnicas de especificação de funções de transferência, para volumes escalares e multidimensionais, baseadas na automatização parcial do processo e simplificação do espaço de interação usado na definição das funções. Como principais contribuições, são apresentados uma eficaz combinação de técnicas complementares para especificação de funções de transferência para volumes escalares; e um método de especificação de funções de transferência para volumes multidimensionais que reúne o potencial de classificação dos mapas auto-organizáveis com a capacidade de decisão não-binária acerca da

visibilidade e aspecto de voxels pertinente às funções de transferência tradicionais.

Palavras-chave: Funções de transferência, visualização volumétrica, imagens 3D, computação gráfica, interação com dados volumétricos.

Design of One-Dimensional and Multi-Dimensional Transfer Functions for Direct Volume Rendering

ABSTRACT

Volume data are very often used in several areas of science, such as medicine, physics and meteorology. Typical examples are data provided by computed tomography, magnetic resonance imaging or estimation of physical phenomena through numerical simulation or sensors. Such data are often provided as regular three-dimensional grids where each element has a scalar or higher-dimensional value, though other topologies may also be employed to express the position of the values in the three-dimensional space. Visualizing volume data is very important in understanding the conveyed information, but it is also a hard task. Thus, many approaches to this problem have been developed.

Direct volume rendering is a set of visualization techniques that have become very popular because they can visually represent volume data, keeping their three-dimensional structure, without extracting intermediate geometries. Such processes require a mapping from voxels' attributes to optical attributes, which allows generating images from the data through the application of a visualization algorithm that implements an illumination model, which is often very simple. This mapping, known as transfer function, associates each volume element with values of optical properties. Therefore, transfer functions play an important role in defining the visibility and the aspect of structures inside a volume, typically using opacity and color, respectively, as optical attributes. However, the design of a good transfer function, capable of generating informative images, is a complex task which must be simplified as much as possible through the support of suitable tools. A simple manual design process is a trial-and-error effort, due to the difficulty of understanding the relationship between the transfer function and the generated image, specially when dealing with multi-dimensional volume data, which require transfer functions with a wide domain.

The need to accelerate and simplify the transfer function design led to the development of several automatic and semi-automatic approaches to the problem, which can reduce or eliminate the user's interaction effort. However, the existent proposals lack in simplicity, interactivity or flexibility. This work outlines transfer function design methods for visualization of scalar volume data and multi-dimensional volume data. We propose techniques based on partial automation of the design process and simplification of the interaction space used in TF specification. Our main contributions are an effective combination of complementary techniques for specifying transfer functions for scalar volumes; and a multi-dimensional transfer function design method that brings together the classification capabilities of self-organizing maps and the transfer functions' ability of non-binary decision on voxels' visibility and aspect.

Keywords: transfer functions, volume visualization, 3D images, computer graphics.

1 INTRODUÇÃO

Diversas áreas da ciência valem-se de obtenção e visualização de dados volumétricos — dados cujas amostras possuem, além de valores, o atributo posição no espaço tridimensional. Tem-se como exemplos típicos de dados volumétricos as imagens 3D provenientes de dispositivos de tomografia computadorizada e ressonância magnética e os dados de sensoriamento ou simulação numérica, comuns em Meteorologia, Física e Engenharia. A investigação desses dados, sobretudo sua visualização, proporciona aos pesquisadores dessas áreas um melhor entendimento dos fenômenos e estruturas ali representados. Em decorrência, a visualização de dados volumétricos consagra-se como uma ferramenta de enorme importância na descoberta científica.

É bastante usual a prática da visualização de múltiplas fatias bidimensionais de um dado volumétrico, após serem submetidas a técnicas tradicionais de processamento de imagens — método muito utilizado em diagnóstico médico, por exemplo. Entretanto, a visualização de volumes como um todo tem recebido crescente atenção da comunidade científica, pois a reconstrução tridimensional oferece a clara vantagem de traduzir mais diretamente a real distribuição espacial das estruturas representadas. Contudo, essa forma de visualização faz emergir problemas de oclusão. As soluções existentes para visualização 3D envolvem extração de malhas poligonais ou classificação dos elementos discretos do volume, usualmente com objetivo de definir sua visibilidade.

Dados volumétricos são tipicamente representados como grades tridimensionais regulares, onde cada elemento tem um valor associado, seja ele escalar — caracterizando volumes unidimensionais — ou vetorial — caracterizando volumes multidimensionais. Sob essa forma, um dado volumétrico pode ser entendido como uma matriz tridimensional trivialmente serializável para fins de transmissão e armazenamento, ficando a posição espacial implicitamente definida pela ordem dos elementos após a serialização. Todavia, também é comum a representação de dados volumétricos através de estruturas irregulares ou que não apresentam topologia de grade. Estas dependem de estruturas de dados mais complexas para serem armazenadas, mas podem ser reamostradas para se obter uma representação como grade regular.

Genericamente, a visualização de dados volumétricos, conservando sua geometria tridimensional, se dá por extração de isosuperfícies ou por visualização direta [BRO 2001]. Isosuperfícies (superfícies com o mesmo valor escalar em toda sua extensão) podem ser aproximadas por malhas poligonais, usualmente obtidas utilizando-se algoritmos de particionamento regular do espaço seguido de aproximação local da superfície por polígonos elementares. O principal exemplo é o algoritmo conhecido como *marching cubes* [LOR 87].

A associação de valores de atributos ópticos às amostras de um volume seguida da projeção destas no plano de imagem denomina-se visualização volumétrica direta [LEV 88].

Tem como principal vantagem sobre a visualização por extração de isosuperfícies a melhor representação da incerteza dos valores, por não depender de decisões binárias acerca da existência de segmentos de isosuperfícies em uma determinada porção do volume. Entre as técnicas de visualização volumétrica direta destacam-se a amostragem de textura 3D (facilmente implementada com o apoio do hardware gráfico) [CAB 94, CUL 94, ENG 2001, KRU 2003], *ray casting* (traçado de raios do observador para o volume) [DAN 92, LEV 90] e *cell projection* (projeção ordenada de células do volume, como voxels ou tetraedros).

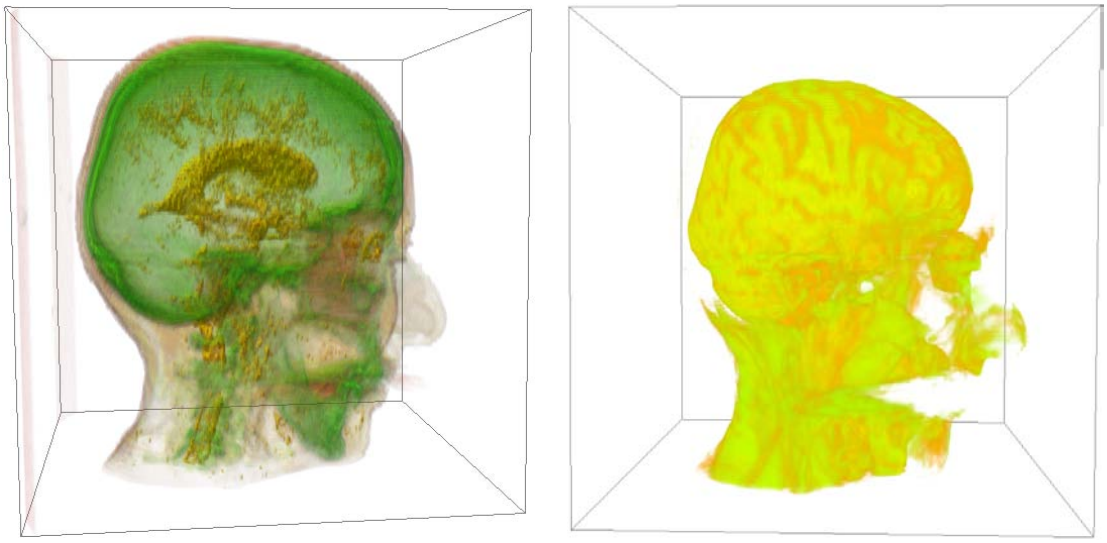
1.1 Visualização Volumétrica Direta e Funções de Transferência

O trabalho pioneiro de Levoy [LEV 88], que introduz a visualização volumétrica direta, fundamentou-se no algoritmo de *ray casting*, que pode ser descrito como segue. O plano da imagem é posicionado em relação ao volume; raios são traçados passando pelo centro de cada pixel, partindo do observador, resultando em projeção perspectiva, ou paralelos entre si, resultando em projeção ortogonal; ao longo de cada raio que intercepta o volume, são extraídas diversas amostras (do volume), com uma freqüência suficientemente grande para capturar as principais variações do campo escalar ou vetorial representado; valores de propriedades ópticas são atribuídos às amostras; e, por fim, compõe-se ordenadamente esses valores, por raio, produzindo a cor final de cada pixel. Maiores detalhes sobre esse processo são fornecidos no Capítulo 2.

O mapeamento dos atributos das amostras do dado volumétrico para atributos visuais, na visualização direta, é chamado de função de transferência (FT). A qualidade de uma visualização é uma medida subjetiva relacionada à quantidade e precisão das inferências permitidas ao usuário e está fortemente relacionada à função de transferência empregada. Conseqüentemente, a especificação de boas funções de transferência (que produzem imagens informativas) é fundamental na exploração de dados volumétricos por visualização direta. Diversas técnicas para especificação de tais funções têm sido propostas na literatura, mas a geração de funções de transferência eficazes permanece como uma tarefa complexa e não-intuitiva, em razão, principalmente, da multiplicidade de FTs e da distância cognitiva entre seu domínio e o domínio espacial — o volume.

Ao definir valores de propriedades ópticas para os elementos de volume, as funções de transferência perfazem classificação e definem a visibilidade das estruturas presentes valendo-se do atributo opacidade, normalmente empregado. Os atributos ópticos mais freqüentemente utilizados são cor e opacidade, porém o uso de quaisquer outros pode ser considerado, desde que seja empregado um algoritmo de geração de imagens correspondente a um modelo de iluminação (normalmente muito simples) para o qual os atributos considerados tenham relevância. Coeficientes ou funções definindo a transmissão e o espalhamento da luz ao interagir com uma amostra do volume [KNI 2003b, KNI 2002a], coeficientes de reflexão especular e difusa da luz [LUM 2004a] e cores espectrais [BER 2005] são exemplos de atributos ópticos empregados em trabalhos anteriores, mediante proposta e implementação de modelos de iluminação consistentes. A figura 1.1 ilustra o emprego da opacidade no destaque ou supressão de estruturas presentes em um dado volumétrico.

Funções de transferência unidimensionais atribuem valores de atributos ópticos às amostras com base apenas em um campo escalar. São as mais comumente utilizadas, mas têm poder de classificação bastante limitado. Por outro lado, FTs multidimensionais consideram campos vetoriais (ou seja, com mais de uma variável), possuindo um voca-



(a) (b)

Figura 1.1: Duas visualizações do Cadaver Head Dataset: (a) baixos níveis de opacidade atribuídos ao crânio, à pele e à porção externa do cérebro, enquanto a porção interna deste apresenta opacidade alta; (b) todo o cérebro recebe opacidade alta, enquanto o crânio e a pele apresentam opacidade nula.

bulário mais amplo para expressar classificações. Campos vetoriais são freqüentemente encontrados em dados de simulação e podem ser produzidos a partir de volumes escalares pelo cálculo de atributos secundários, como curvaturas [KIN 2003, HLA 2000], medidas estatísticas locais [TEM 2001] e derivadas [KNI 2001]. Em contraponto, a complexidade de especificação de uma FT cresce com seu número de dimensões, logo funções de transferência multidimensionais trazem um desafios maiores em decorrência da dificuldade de representação e exploração de seu domínio [KNI 2002b].

1.2 Motivação e Objetivo

A especificação de funções de transferência pode ser considerada um dos maiores problemas na visualização volumétrica [PFI 2001] e, conseqüentemente, um grande limitador da difusão do uso de técnicas de visualização volumétrica direta para fins práticos como, no âmbito da medicina, diagnóstico de tumores, lesões internas e aneurismas e planejamento de cirurgias.

Este trabalho tem como motivação o desafio de proporcionar novos métodos e interfaces para especificação de funções de transferência que sejam mais convenientes aos usuários de ferramentas de visualização volumétrica, permitindo maiores eficácia e eficiência na obtenção de imagens informativas.

A especificação manual de funções de transferência, tipicamente realizada por interação com representações gráficas das funções e manipulação de pontos de controle, é um processo iterativo de tentativa e erro quando o usuário não dispõe de informações adicionais ou conhecimento prévio sobre os dados volumétricos visualizados. Devido à necessidade de reduzir o tempo despendido com a produção de visualizações, é consenso entre os pesquisadores que sistemas de visualização volumétrica devem ser capazes de auxiliar ativamente a especificação de FTs.

Algumas propostas visam à geração de funções de transferência de maneira totalmente

automática [MAR 97, FUJ 2000, FUJ 99, HE 96]; entretanto, o afastamento do usuário do processo de especificação desrespeita a subjetividade da qualidade das visualizações. Abordagens semi-automáticas de especificação têm maior confiabilidade porque auxiliam o usuário sem privá-lo das principais decisões. Técnicas semi-automáticas fornecem informações adicionais, extraídas dos dados volumétricos, provêem, como sugestões, funções de transferência automaticamente geradas ou permitem especificação das FTs em espaços de interação simplificados ou mais intuitivos (ver, como exemplos, [TZE 2004, TZE 2003, TZE 2005, SER 2006a, REZ 2006, TAK 2002, LUM 2004a, SER 2006c]). Técnicas automáticas e semi-automáticas de geração de FTs podem, ainda, ser classificadas como abordagens dirigidas aos dados — quando baseadas na análise das amostras do volume — ou abordagens dirigidas às imagens — quando baseadas na análise das imagens geradas a partir das FTs.

Em trabalho anterior, Prauchner et al. [PRA 2005a, PRA 2005b] propuseram a especificação de funções de transferência unidimensionais utilizando uma abordagem com interação em dois níveis: escolha de funções de transferência dentre um conjunto de FTs gerado automaticamente a partir de informações extraídas do volume — primeiro nível — e refinamento manual da FT escolhida — segundo nível. A primeira parte do trabalho aqui desenvolvido foi motivada pelos resultados de Prauchner.

O objetivo do trabalho detalhado nesta dissertação é propor, implementar e validar técnicas de especificação semi-automática de funções de transferência unidimensionais e multidimensionais, visando oferecer aos potenciais usuários finais maiores praticidade e agilidade na obtenção de visualizações tridimensionais. Mais precisamente, objetiva-se simplificar o projeto de funções de transferência oferecendo abstração da complexidade das FTs, permitindo geração automática de FTs úteis, usadas como sugestões, e proporcionando rápida compreensão das estruturas presentes em um volume e das relações entre o domínio espacial e o domínio da função de transferência.

Para a especificação de funções de transferência unidimensionais, foi proposto um ambiente de visualização volumétrica que combina geração de FTs para ênfase de bordas — uma abordagem dirigida aos dados — e geração estocástica evolutiva usando algoritmos genéticos — abordagem dirigida às imagens. O ambiente também permite múltiplas visualizações simultâneas e refinamento manual de FTs. Dessa forma definiu-se uma ferramenta versátil que combina técnicas complementares de especificação de funções de transferência e utiliza uma interface de dois níveis: visualizações múltiplas sob a forma de *thumbnails* e visualização única com fina resolução. Este trabalho baseia-se, em parte, na proposta de Prauchner et al. [PRA 2005a, PRA 2005b].

Investigou-se a especificação de funções de transferência multidimensionais, sendo proposta uma técnica de redução dimensional baseada em mapas auto-organizáveis [KOH 97, SAN 2002] que permite a definição de FTs em um espaço de interação reduzido. A técnica é aplicável, de maneira uniforme, a dados volumétricos de dimensão arbitrária e consiste em agrupar, por semelhança, as amostras do dado volumétrico representando-as num mapa bidimensional. Visualizações são obtidas pela exploração de funções de transferência definidas no domínio desse mapa. Adicionalmente, a abordagem proposta engloba a geração automática de funções de transferência de cor significativas que traduzem as dissimilaridades entre regiões de um volume.

1.3 Organização do Texto

O próximo capítulo descreve a técnica de visualização volumétrica implementada como plataforma de teste para as abordagens propostas para especificação de funções de transferência. No capítulo 3, é fornecida uma visão geral sobre as técnicas de especificação de FTs descritas na literatura, enfatizando as mais relevantes para o presente trabalho. Após, nos capítulos 4 e 5, estão expostos, sob forma de artigos, os trabalhos desenvolvidos como essência da dissertação, sobre especificação de funções de transferência unidimensionais e multidimensionais, respectivamente. Por fim, no capítulo 6, expõe-se as conclusões alcançadas e discute-se tendências de pesquisas atuais e futuras acerca de funções de transferência para visualização volumétrica direta.

2 VISUALIZAÇÃO VOLUMÉTRICA DIRETA BASEADA EM TEXTURA

Após a proposição inicial de Levoy [LEV 88], várias técnicas de visualização volumétrica direta foram desenvolvidas [LAC 94, LEV 90, WES 90]. Essas técnicas são hoje bastante conhecidas no âmbito da visualização científica e sua descrição pode ser encontrada em diversas publicações [KAU 91, LIC 98, HAN 2005, BRO 2001].

A evolução dos processadores gráficos (GPUs) e a redução do custo das placas aceleradoras de gráficos para computadores pessoais tornaram especialmente popular a visualização volumétrica direta baseada em amostragem de textura tridimensional [CUL 94], um recurso comum nas placas aceleradores modernas.

O hardware dedicado acelera as aplicações gráficas assumindo, e executando com maior rapidez, uma grande parcela das tarefas envolvidas na geração de imagens a partir de modelos tridimensionais. As principais tarefas dos processadores gráficos são transformação geométrica dos vértices dos polígonos do modelo tridimensional, rasterização dos polígonos e processamento de fragmentos. A rasterização subdivide os polígonos projetados no plano de imagem em fragmentos, caracterizando um processo de discretização onde as partículas produzidas (fragmentos) encontram-se perfeitamente alinhadas com os pixels a serem gerados. Os fragmentos então recebem atributos de cor e opacidade (processamento de fragmentos) e são combinados em cada pixel para produzir a cor final, pois freqüentemente há mais de um fragmento para um pixel (sobreposição). A combinação de fragmentos, também tarefa do processador gráfico, pode ser destrutiva, quando um fragmento prevalece sobre os outros (no caso de oclusão por diferença de profundidade, por exemplo), ou aditiva, quando as propriedades dos fragmentos são combinadas para gerar a cor do pixel (no caso de efeitos de transparência, por exemplo).

A amostragem de textura 3D foi a técnica escolhida, no presente trabalho, para gerar visualizações, devido à simplicidade, à facilidade de implementação e à possibilidade de aproveitamento dos recursos do GPU, obtendo maior eficiência. Tem como limitação a aplicação restrita a dados volumétricos descritos por uma grade tridimensional regular. Os valores das amostras do dado volumétrico são armazenados em uma textura tridimensional de acordo com a posição das amostras na grade que descreve o volume. Em aplicações que empregam os recursos dos processadores gráficos, por questão de desempenho ou exigência do hardware, a textura armazenada normalmente tem como dimensões potências inteiras de dois, o que pode implicar reamostragem da grade original para obtenção de dimensões apropriadas. Outras formas de representação de volumes também podem ser usadas para produzir grades regulares através de reamostragem. A próxima seção aborda em detalhes a visualização por amostragem de textura tridimensional.

2.1 Descrição Geral da Técnica

Nesta técnica assume-se que a textura 3D, representando o volume, encontra-se confinada em um paralelepípedo sujeito a transformações lineares, geralmente apenas rotação, translação e escala, que posicionam o volume em relação ao observador. Uma vez definida a transformação de projeção e posicionado o paralelepípedo que delimita a textura, esta é amostrada de maneira ordenada.

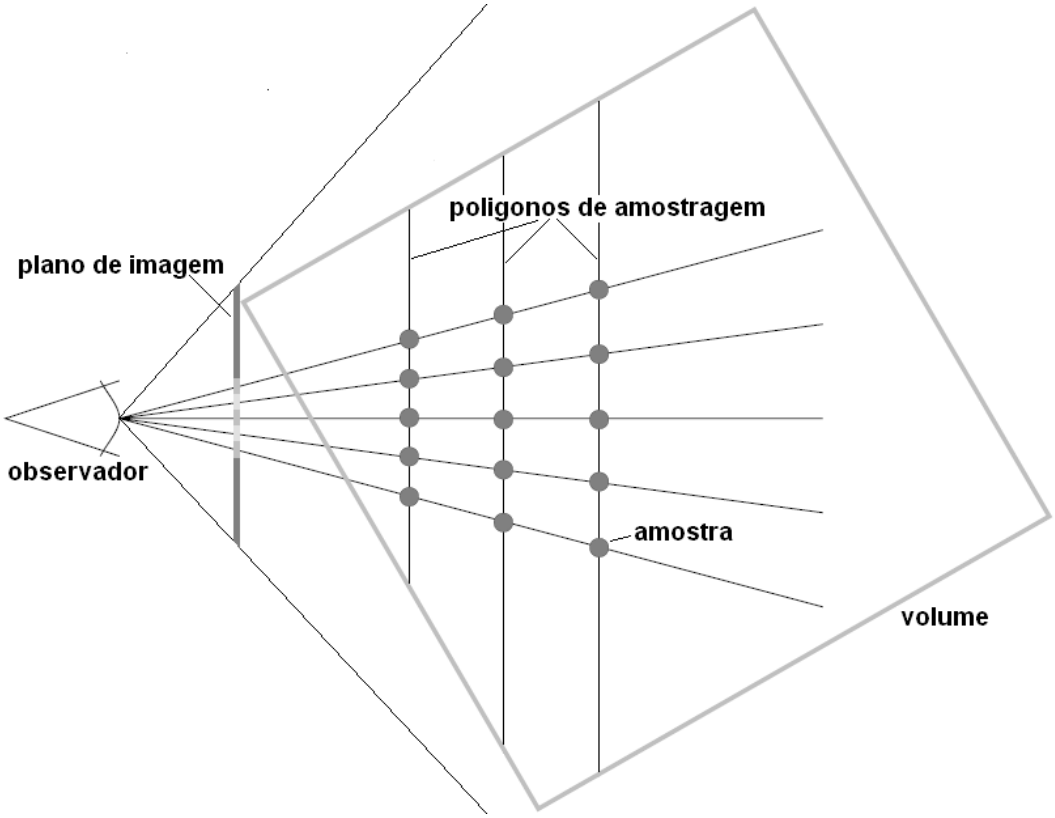


Figura 2.1: Amostragem da textura 3D, confinada em um paralelepípedo, feita por polígonos paralelos ao plano de imagem. A rasterização do polígono define pontos discretos de amostragem, cujos valores ópticos, obtidos pela aplicação da FT, são usados na composição da cor dos pixels, indicada pelos raios convergentes. O processo é similar ao do algoritmo de *ray casting*, no qual os pontos de amostragem não são restritos a polígonos.

No visualizador implementado, a textura é amostrada por polígonos definidos pela intersecção do paralelepípedo com planos perpendiculares à direção de observação, como mostra a figura 2.1. Os polígonos são gerados, igualmente espaçados, em ordem decrescente de profundidade em relação ao observador, sendo rasterizados pelo hardware gráfico. A superposição dos polígonos no plano de imagem é submetida a uma operação de composição que considera os valores de atributos ópticos das amostras, definidos pela função de transferência.

A função de transferência é também armazenada em textura, como uma tabela (*lookup table*), e sua aplicação é realizada por fragmento, valendo-se da possibilidade de programação do *pipeline* gráfico nos GPUs modernos. Para cada valor amostrado do volume pelos fragmentos dos polígonos rasterizados, a função de transferência é aplicada e os valores de atributos ópticos produzidos são utilizados na composição da imagem final.

O diagrama da figura 2.2 sumariza o processo. A operação de composição utilizada

é descrita pela equação 2.1, onde α_s é a opacidade do fragmento rasterizado, C_s é a cor do fragmento rasterizado, C_d é a cor existente no *frame buffer* como resultado parcial do processo de composição e $C_{d\ new}$ é a cor resultante de uma operação de composição.

$$C_{d\ new} = \alpha_s \times C_s + (1 - \alpha_s) \times C_d \tag{2.1}$$

Como pode ser observado, a visualização por amostragem de textura 3D é análoga ao algoritmo de *ray casting* tradicional, porém neste os pontos de amostragem não se encontram confinados em polígonos, mantendo-se restritos apenas aos raios que ligam o observador aos pixels atravessando o plano de imagem. Ainda, na visualização por amostragem de textura 3D com aceleração pelo hardware gráfico, o laço principal do algoritmo itera sobre os polígonos de amostragem, enquanto, no *ray casting*, o laço principal tipicamente itera sobre os raios.

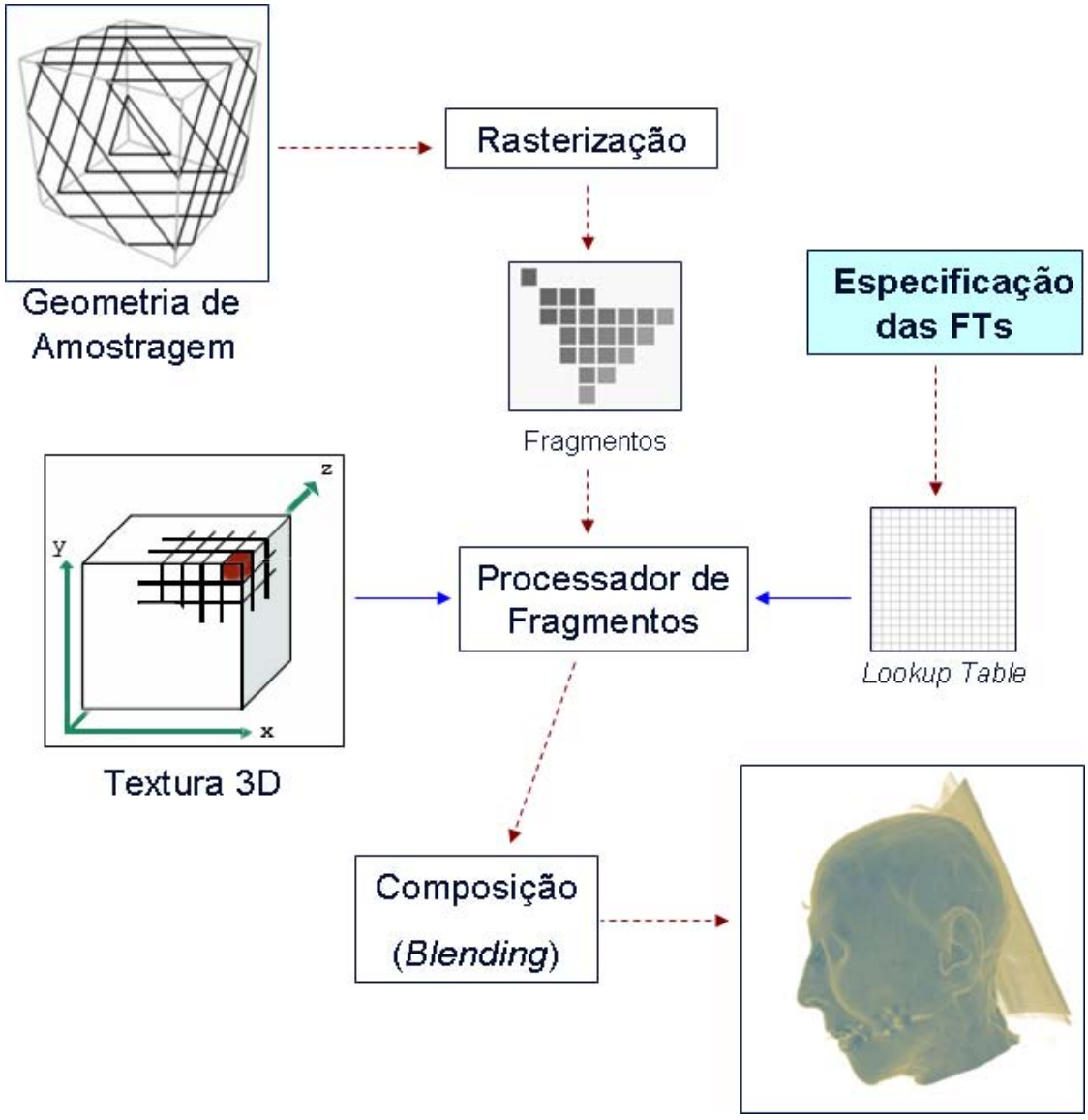


Figura 2.2: Diagrama do processo de geração de visualizações com base em amostragem de textura tridimensional. Adaptação do diagrama da dissertação de mestrado de João Luis Prauchner [PRA 2005b].

Como mencionado, funções de transferência podem ser definidas por tabelas (*lookup tables*), onde existe uma entrada para cada valor discretizado de voxel. Em cada entrada da tabela estão armazenados os valores de atributos ópticos correspondentes ao respectivo valor de voxel. Funções de transferência definidas por tabelas podem ser trivialmente armazenadas como texturas. Neste trabalho foram utilizadas funções de transferência de cor e de opacidade armazenadas na memória do hardware gráfico como texturas RGBA. As FTs unidimensionais foram implementadas como tabelas de 256 entradas, enquanto as multidimensionais foram parcialmente implementadas como tabelas bidimensionais contendo 256 X 256 entradas (maiores detalhes são abordados no capítulo 5). Na amostragem da textura 3D e na aplicação de funções de transferência armazenadas em textura (também realizada através de amostragem), com o objetivo de reduzir imperfeições na imagem final decorrentes da representação discreta do volume e das FTs, pode-se utilizar a interpolação nativa do hardware gráfico: trilinear para a textura 3D e linear, ou bilinear, para a função de transferência.

2.2 Pré-Integração da Contribuição da Função de Transferência

Na visualização volumétrica por amostragem de textura 3D usando polígonos paralelos, ou na visualização por *ray casting*, que também pode ser implementada com aceleração pelo hardware gráfico [KRU 2003], há uma evidente dicotomia entre qualidade da imagem gerada e desempenho, determinados pela freqüência de amostragem do volume. O desempenho é inversamente dependente da freqüência de amostragem, enquanto o comportamento do campo amostrado, quando modulado pela função de transferência, não pode ser satisfatoriamente reconstituído usando-se freqüências de amostragem demasiadamente baixas. Visando à suavização de imperfeições notáveis na imagem final decorrentes do uso de baixas freqüências de amostragem, Engel et al. [ENG 2001] propuseram a pré-integração da contribuição da função de transferência. A composição das amostras depois de submetidas à função de transferência constitui um processo de integração que, normalmente, busca aproximar a integral descrita pela equação 2.2, conhecida como integral de *rendering* volumétrico, que define a cor resultante da avaliação de um raio de visualização, considerando apenas iluminação local.

$$C_{res} = \int_{l_i}^{l_o} C(V(R(l))) \times \exp\left(-\int_{l_i}^{l} \lambda(V(R(j))) \times dj\right) \times dl \qquad (2.2)$$

Na equação, C_{res} é a cor resultante; l_i e l_o são, respectivamente, os pontos de entrada e de saída do raio de visualização, no volume, em sua equação paramétrica; C é a função de transferência de cor, que determina a contribuição de cor por unidade de comprimento do raio; V é a função que retorna o valor da amostra do volume, dada uma posição no espaço; R é a equação paramétrica, com vetor direção normalizado, do raio de visualização; e λ é uma função de transferência, semelhante à de opacidade, que produz o coeficiente de extinção. Na integral, pode-se observar que o coeficiente de extinção determina a atenuação da contribuição de cor de uma amostra do volume em função do caráter semitransparente das amostras que a precedem no raio de visualização. A função de transferência de cor e a de coeficiente de extinção exprimem razões onde o denominador é a unidade de comprimento do raio, mas nas aproximações da integral, essas razões freqüentemente transformam-se em valores absolutos devido à decomposição dos raios de visualização em segmentos de comprimento homogêneo ou muito similar.

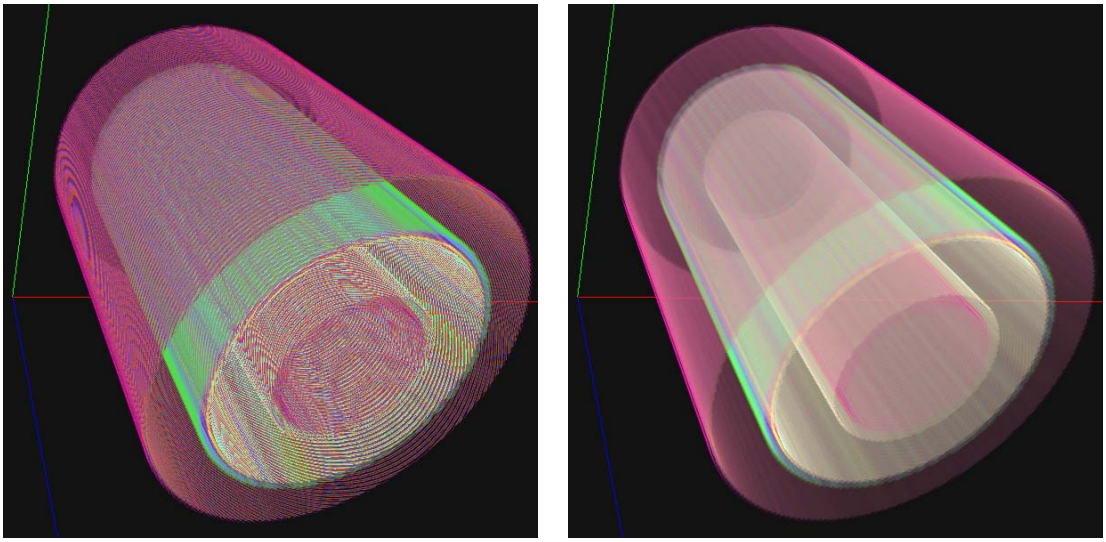
$$\alpha = 1 - \exp\left(-\int_{l_i}^{l_o} \lambda(V(R(l))) \times dl\right) \tag{2.3}$$

A pré-integração desvincula a integração da contribuição da função de transferência do processo de amostragem, permitindo que este se ocupe apenas da reconstrução do volume. Considerando a visualização por *ray casting* e o uso de interpolação linear, um raio define uma porção unidimensional do volume constituída por uma sucessão de segmentos nos quais o valor do campo amostrado varia linearmente. Pode-se, portanto, precomputar, com base na equação 2.2 — sendo l_i e l_o os pontos de início e fim de um segmento —, a contribuição de cor da função de transferência para todos os possíveis segmentos, definidos por seu comprimento e pelo valor do volume nos seus extremos. A contribuição de opacidade de um segmento é dada pela equação 2.3, que deriva da integral de *rendering* volumétrico. O resultado é uma tabela — tridimensional para FTs unidimensionais — que produz a contribuição da FT (cor e opacidade), para um dado segmento, usando-se como entrada (da tabela) os valores amostrados do volume nos pontos inicial e final do segmento e seu comprimento. Na geração da imagem final, para cada raio, as contribuições (cor e opacidade) de seus segmentos são ordenadamente compostas. Em visualizadores mais simples, o comprimento dos segmentos é assumido como constante, o que reduz em um o número de dimensões da tabela. O uso da pré-integração é análogo em visualizadores fundamentados em amostragem de textura 3D por planos paralelos. Os segmentos são definidos entre dois polígonos de amostragem adjacentes, estando o valor de início de segmento sobre um polígono e o de fim sobre o adjacente de maior profundidade [ENG 2001].

A pré-integração não é aplicável, sem restrições, a funções de transferência de dimensão maior que um, devido ao grande consumo de memória e tempo de processamento implicado na construção das tabelas, que têm pelo menos quatro dimensões (no caso das FTs 2D). É possível, entretanto, pré-integrar a contribuição de funções de transferência com grande número de dimensões impondo restrições à sua forma, como demonstrado na proposta das FTs gaussianas, por Kniss et al. [KNI 2003a]. Com o emprego da pré-integração obtém-se imagens de maior qualidade, porém o uso concomitante de modelos de iluminação ou de ferramentas de corte de volume não é trivial. Lum et al. [LUM 2004b] demonstram como combinar iluminação com pré-integração sem causar imperfeições evidentes e, ainda, detalha um algoritmo para cálculo eficiente de tabelas de pré-integração, reduzindo a complexidade em uma ordem de magnitude em relação ao método de força bruta. Uma técnica para uso de pré-integração na visualização de volumes sujeitos a cortes foi delineada por Roettger et al. [ROE 2003]. Um cuidadoso estudo sobre freqüência de amostragem de dados volumétricos pode ser encontrado no trabalho de Bergner et al. [MUR 2006], onde é matematicamente inferido o limite inferior da freqüência de amostragem exigido na obtenção acurada de visualizações.

2.3 Comentários finais

A implementação da ferramenta proposta para visualização volumétrica empregando funções de transferência unidimensionais compreende pré-integração com cômputo de tabelas realizado na CPU, com rapidez suficiente para garantir interatividade em face de alterações em uma única função de transferência, mesmo quando utilizado o cálculo de tabelas por força bruta. Porém, quando múltiplas visualizações são apresentadas simultaneamente, exigindo geração e pré-integração de diversas funções de transferência, a melhor



(a) (b)

Figura 2.3: Imagens obtidas de um volume escalar sintético, por amostragem de textura 3D, utilizando a mesma função de transferência e o mesmo número de polígonos de amostragem, sem uso de pré-integração da contribuição da função de transferência (a) e com uso de pré-integração (b).

opção para garantir interatividade é a pré-integração simplificada, que negligencia a atenuação da contribuição de cor no cálculo da tabela, ou seja, desconsidera atenuação dentro dos segmentos dos raios de visualização [ENG 2001]. O emprego da pré-integração simplificada é penalizado com um desvio no cálculo da contribuição total de cor em cada pixel, mas garante desempenho superior, embora da mesma ordem de magnitude, ao do algoritmo eficiente de Lum et al. [LUM 2004b], que realiza pré-integração plena.

A utilização da pré-integração da contribuição da função de transferência permite a obtenção de imagens de qualidade sensivelmente melhor, sem aumento do número de amostras do volume, ou seja, sem aumento do número de polígonos de amostragem, no caso da técnica de visualização implementada. A figura 2.3 ilustra o benefício propiciado pela pré-integração.

3 ESPECIFICAÇÃO DE FUNÇÕES DE TRANSFERÊNCIA

As bordas presentes em dados volumétricos — regiões de fronteira entre porções predominantemente homogêneas — estão entre as entidades de maior importância, pois determinam a forma e o tamanho de estruturas de interesse. Em imagens médicas as bordas delimitam órgãos; em dados de simulação podem não ser tão bem definidas, mas representam regiões de intensa mudança no valor de uma variável. Na literatura são encontradas diversas técnicas de especificação de funções de transferência para ênfase de bordas, isto é, FTs que atribuem às regiões de separação valores de propriedades ópticas que provocam destaque, como, por exemplo, opacidade elevada. Existem, entretanto, técnicas de especificação de FTs não orientadas à detecção de bordas.

Alguns métodos de especificação de funções de transferência não consideram diretamente o campo escalar ou n-dimensional descrito pelo dado volumétrico, propondo um espaço derivado para especificação das FTs. Este capítulo expõe uma visão geral das técnicas de especificação de funções de transferência que foram julgadas as mais promissoras entre as descritas na literatura atual.

3.1 Especificação de FTs Unidimensionais

Conforme já mencionado, para volumes de dados escalares, são empregadas funções de transferência unidimensionais. O valor utilizado como entrada da função de transferência pode ser um valor do campo escalar descrito pelo volume ou outra medida computada com base neste. Entre as várias as abordagens delineadas na literatura para a especificação do mapeamento que define a FT, destacam-se as descritas a seguir.

Bajaj et al. [BAJ 97] propuseram algumas medidas, e respectivos métodos de cômputo, com base no campo escalar, úteis na escolha dos valores que, mais provavelmente, representam as estruturas de maior importância. Esses valores escalares, através da função de transferência, devem ser associados a valores de atributos ópticos capazes de enfatizar as estruturas correspondentes — normalmente usa-se o atributo opacidade. Para cada valor escalar discretizado, são computados a área da isosuperfície definida por ele, o volume enclausurado pela mesma isosuperfície e a integral do gradiente ao longo desta. Tipicamente, as mais importantes superfícies de separação têm um valor alto para todas as medidas mencionadas. A abordagem proposta compreende ainda a implementação de uma estrutura de grafo dirigido acíclico que representa o comportamento topológico das isosuperfícies definidas, no volume, pelos valores escalares discretizados. As separações e as junções dos segmentos conexos da isosuperfície que ocorrem com a variação monotônica do valor escalar que a define (a superfície) são registradas na estrutura. Essa informação é também útil na identificação dos valores das superfícies de separação, mais bem definidas, e das áreas ruidosas, caracterizadas por grande atividade de junção e sepa-

ração.

Estratégias semelhantes à de Bajaj et al. [BAJ 97] foram propostas por Pekar et al. [PEK 2001] e por Tenginakai et al. [TEM 2001]. A primeira propõe um método bastante eficiente de cômputo de características das isosuperfícies: integral do gradiente ao longo da isosuperfície, gradiente médio, área total, volume confinado e curvatura. A segunda calcula, por voxel, assinaturas estatísticas — média, desvio padrão, etc. considerando os valores dos voxels num pequeno entorno — e cria histogramas relacionando o valor escalar dos voxels com suas respectivas assinaturas. É mostrado como interpretar os histogramas objetivando identificar os valores escalares que representam bordas. As técnicas acima descritas são bastante semelhantes, pois a especificação da função de transferência exige do usuário apenas a interpretação de gráficos simples, com um significado claro.

A abordagem de especificação automática de TFs de Fujishiro et al. [FUJ 2000, FUJ 99] concentra-se na evolução topológica da isosuperfície diante da variação monotônica do valor escalar que a define. Evolução topológica de uma isosuperfície é a alteração da conectividade entre suas porções em face da variação de algum parâmetro que define a isosuperfície. Uma isosuperfície pode evoluir surgindo, extinguindo-se, separando-se em porções desconexas ou combinando tais porções. Na técnica de Fujishiro, os pontos críticos — valores escalares nos quais segmentos de isosuperfícies surgem, separam-se, mesclam-se ou extinguem-se, implicando mudança de topologia — recebem maior opacidade e descontinuidade de cor, com o objetivo de enfatizar as isosuperfícies correspondentes.

Outra importante técnica de especificação de funções de transferência unidimensionais objetivando ênfase de bordas foi introduzida por Kindlmann e Durkin [KIN 98]. Os autores propuseram a construção de um histograma tridimensional relacionando valor escalar das amostras do volume (voxels) com os respectivos valores do gradiente e da segunda derivada na direção do gradiente. É demonstrado que, com base nos dois valores derivados citados, pode-se estimar a distância entre a amostra e a borda mais próxima, logo, com base no histograma, estima-se, para cada valor escalar, a distância entre voxels definidos por esse valor e a respectiva borda mais próxima. Dessa forma, cria-se um domínio mais intuitivo para especificação de funções de transferência, onde a distância até a borda é mapeada para atributos ópticos. Em geral, valores escalares com pequena distância associada são convertidos em altos níveis de opacidade.

He et al. [HE 96] desenvolveram um técnica de especificação de funções de transferência unidimensionais, dirigida às imagens, fundamentada em algoritmos genéticos. Nessa abordagem, uma FT é codificada como uma seqüência ordenada de pontos de controle, definidos por um valor escalar e respectivas cor e opacidade. Os pontos de controle são tomados como genes para aplicação de algoritmos genéticos, sendo as FTs os cromossomos (seqüência de genes). As funções de transferência usadas na construção das visualizações são obtidas pela interpolação dos pontos de controle. Na técnica, inicialmente, uma população de FTs é gerada tendo pontos de controle com valores aleatórios de atributos, mas ordenados pelo valor escalar; após, para cada FT da população inicial é gerada uma visualização sob a forma de *thumbnail*; o usuário, então, seleciona as imagens que melhor revelam as estruturas cuja visualização é desejada; a seguir, as FTs responsáveis pela geração das imagens escolhidas são usadas como modelos para a elaboração de novas FTs, através da aplicação de operadores de mutação — que alteram pontos de controle — e cruzamento (*crossover*) — que combinam duas FTs; por fim, as FTs obtidas passam a constituir a nova população e o processo se repete a partir da geração de novos *thumbnails*. A aleatoriedade embutida nos operadores usados em algoritmos ge-

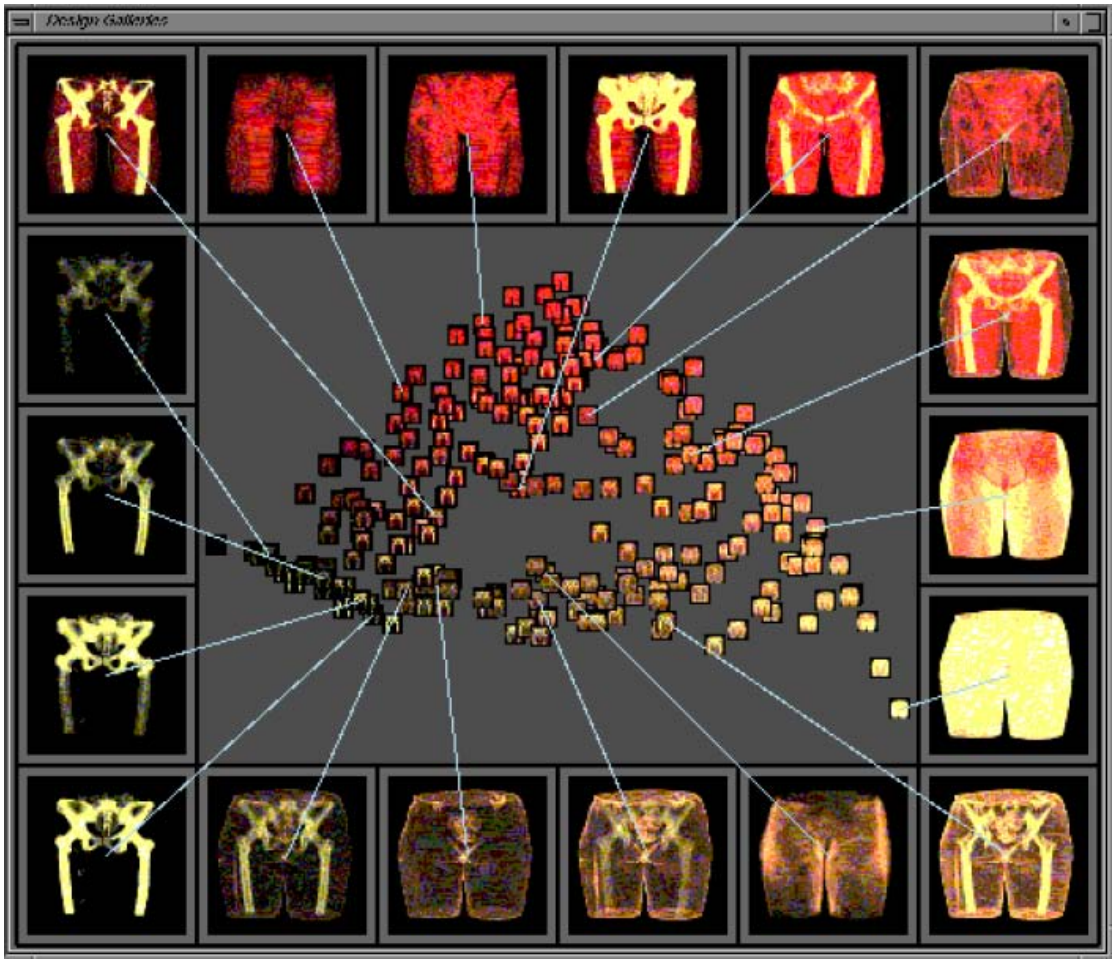
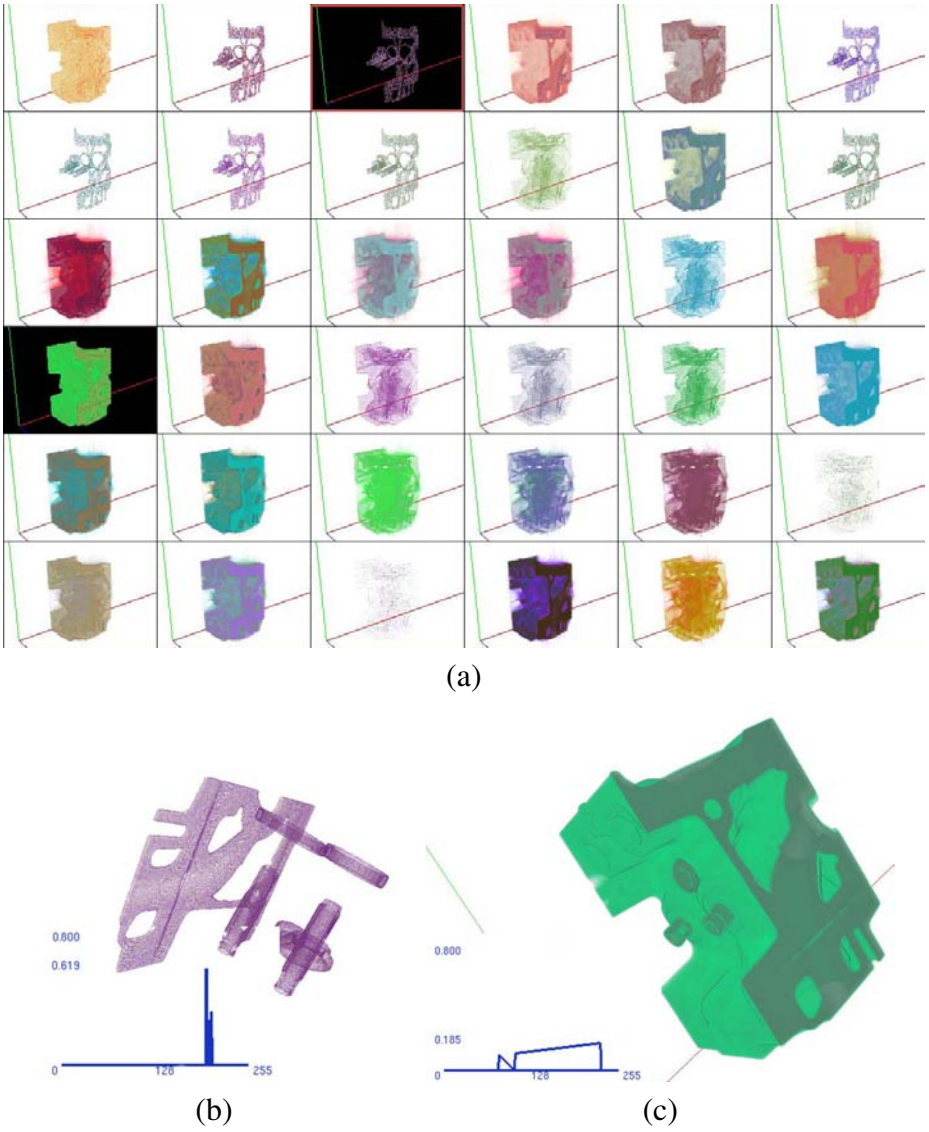


Figura 3.1: Interface do método de *Design Galleries*, onde são apresentados diversos resultados, decorrentes de diferentes funções de transferência, sob a forma de *thumbnails* — imagem reproduzida de [MAR 97].

néticos permite uma busca ampla no espaço de possibilidades, provocando o surgimento de funções de transferência que revelam diversos aspectos do dado volumétrico. Adicionalmente, a técnica tem como vantagens a total abstração da função de transferência e a visualização de múltiplas possibilidades simultaneamente.

Valendo-se também de múltiplas visualizações simultâneas, destaca-se a técnica de *Design Galleries*, de Marks et al. [MAR 97], que produz uma galeria de visualizações distintas, apresentadas como *thumbnails* que podem ser ampliados (ver figura 3.1). A técnica emprega um modelo parametrizável de função de transferência, de forma que cada instância dos parâmetros corresponde a uma FT única. Inicialmente, em uma etapa de pré-processamento, diversas instâncias (valores) dos parâmetros são gerados através de um algoritmo de dispersão, que proporciona uma ampla e bem distribuída cobertura do espaço paramétrico; após, com base no modelo de FT e nos valores produzidos para os parâmetros, são geradas diversas FTs, empregadas na produção de visualizações; por fim, as imagens obtidas são agrupadas por similaridade e apresentadas ao usuário. É uma abordagem que demanda muito tempo de processamento, mas que exige pouco do usuário e que proporciona a compreensão de vários aspectos do dado.

Prauchner et al. [PRA 2005a, PRA 2005b] reuniram numa única ferramenta as vantagens do método de *Design Galleries* [MAR 97], da técnica de Kindlmann e Durkin [KIN 98]



(a)

(b) (c)

Figura 3.2: Interface da ferramenta desenvolvida por Prauchner et al. [PRA 2005a]. O primeiro nível de interação é mostrado em (a), enquanto (b) e (c) ilustram duas visualizações obtidas no segundo nível. Imagens reproduzidas de [PRA 2005a].

e da especificação manual de FTs através do ajuste de pontos de controle. Nessa abordagem, no primeiro nível de interação de uma interface de dois níveis, visualizações múltiplas são fornecidas como uma coleção de *thumbnails*, gerados com FTs especificadas automaticamente pela escolha de pontos de controle através do método de Kindlmann e Durkin. Os pontos de controle da FT têm, em princípio, valores de cor e opacidade aleatórios, mas podem ser ajustados pelo usuário num segundo nível de interação, onde uma única visualização é tratada (ver figura 3.2).

3.2 Especificação de FTs Multidimensionais

Os valores do gradiente e da segunda derivada direcional do campo escalar, além dos próprios valores do campo, são comumente utilizados como domínio de funções de transferência multidimensionais porque bordas podem ser bem caracterizadas com base nesses valores. Voxels em regiões de fronteira têm tipicamente um gradiente de grande

Figura 3.3: Interface da ferramenta desenvolvida por Kniss para definição de funções de transferência tridimensionais. Na parte inferior é apresentado, em tons de cinza, o histograma dos voxels, inseridos de acordo com seu valor escalar e magnitude do gradiente, onde um arco pode claramente ser visto. *Widgets* geométricos são utilizados como meio de definir a FT. Reprodução da figura contida em [KNI 2002b].

magnitude e uma segunda derivada na direção do gradiente próxima de zero. Kniss et al. [KNI 2002b, KNI 2001] descreveram um método de especificação de funções de transferência tridimensionais (escalar, gradiente e segunda derivada) baseado na análise do histograma bidimensional, relacionando valor escalar com magnitude do gradiente, e na interação com *widgets*, representados sobre o histograma, que definem a forma da FT de opacidade (ver figura 3.3). Cada widget define uma FT de opacidade parcial, com uma cor associada, que é combinada com outras FTs parciais para compor a função de transferência final. A terceira dimensão — a segunda derivada — não é graficamente representada por um terceiro eixo coordenado, mas aparece como um parâmetro dos *widgets*. No histograma relacionando escalar e gradiente, as bordas são facilmente identificadas como arcos, o que auxilia no posicionamento dos *widgets*.

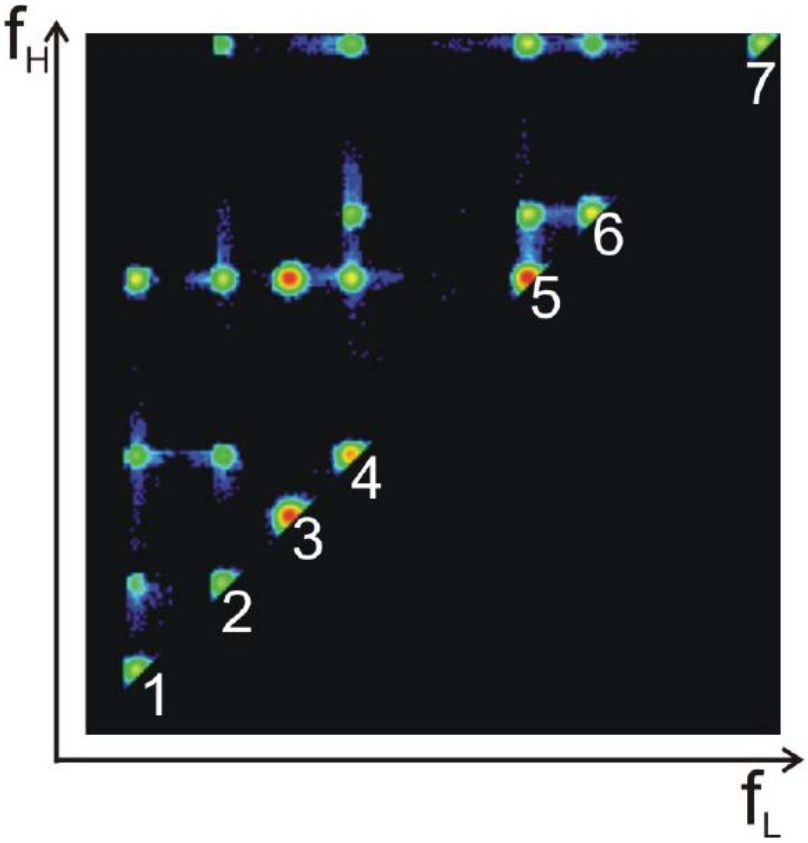


Figura 3.4: Histograma de voxels no espaço LH. Sete materiais podem ser identificados. As bolhas no histograma representam, na diagonal, voxels em regiões homogêneas e, no restante do histograma, regiões de fronteira. Figura retirada do artigo. Reprodução da figura contida em [SER 2006c].

3.3 Especificação de FTs em Espaços Derivados

Entre as técnicas de especificação de funções de transferência voltadas à detecção de bordas, destacam-se os resultados obtidos utilizando-se funções de transferência especificadas no espaço LH, definido por dois eixos coordenados (L e H), proposto por Šereda et al. [SER 2006c]. Nessa abordagem, para cada voxel no volume, a direção do gradiente é seguida, em ambos os sentidos, até que sejam encontradas duas porções homogêneas do volume (valor escalar aproximadamente constante). Os valores escalares das duas porções encontradas definem o voxel como de fronteira, quando são valores distintos, ou como pertencente a uma região homogênea, quando similares. Essa assinatura bidimensional, usada para substituir o valor escalar único, recebe o nome LH porque é descrita pelos valores do par de porções homogêneas entre as quais se situa o voxel, sendo uma porção a de menor valor escalar (low, ou L) e a outra a de maior valor (high, ou H). Constitui-se, dessa forma, um espaço bidimensional (espaço LH) sobre o qual se especifica a função de transferência. Adicionalmente, um histograma de voxels no espaço LH ilustra mais claramente as bordas e as regiões homogêneas, que surgem como pequenas e bem definidas formas arredondadas (ver figura 3.4).

Os mesmos autores publicaram também uma extensão da técnica [SER 2006b] que permite uma classificação mais elaborada dos voxels de borda, fazendo distinção entre os dois lados da fronteira. Da mesma forma, os voxels são representados no espaço LH, sobre o qual se define a função de transferência. Ainda, em outra extensão [SER 2006a],

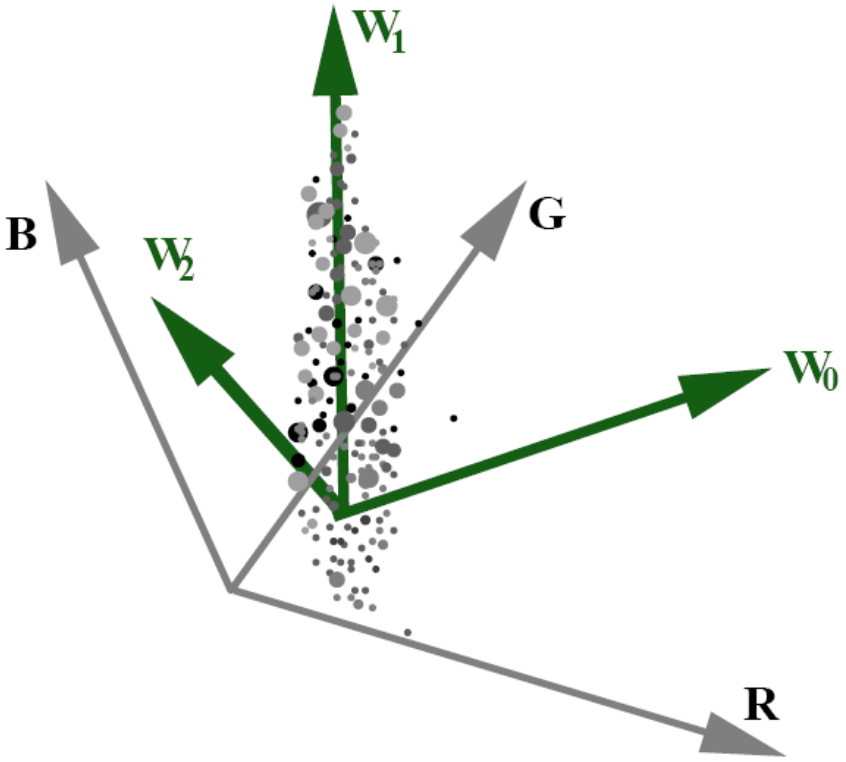


Figura 3.5: A figura ilustra assinaturas de voxels, representadas por pontos no espaço definido pelos vetores R, G e B. Através do método de ICA define-se outra base vetorial — $W0$, $W1$ e $W2$ — na qual as assinaturas passam a ser representadas, mediante transformação linear. Figura retirada do artigo. Reprodução da figura contida em [TAK 2002].

os mesmos autores agruparam voxels hierarquicamente de acordo com suas assinaturas LH, permitindo que classificações do volume sejam expressas selecionando sub-árvores da hierarquia.

Em um trabalho de Lum e Ma [LUM 2004a] é utilizada uma assinatura bidimensional de voxel similar à LH, porém extraída durante a construção da visualização. Para cada amostragem do volume, no processo de geração de imagem, é consultado o gradiente do campo escalar naquele ponto. Duas amostras adicionais do volume escalar são extraídas, de pontos ligeiramente deslocados em relação ao ponto de amostragem original, um deles deslocado em favor do gradiente e outro no sentido oposto. O par de amostras é, então, utilizado para consultar uma função de transferência bidimensional que, em geral, enfatiza bordas, caracterizadas por pares constituídos por dois valores escalares distintos.

Tratando de volumes multidimensionais, destacam-se abordagens de especificação de funções de transferência que proporcionam ao usuário interação em um espaço simplificado, ocultando a complexidade da função. Takanashi et al. [TAK 2002] propuseram a utilização de análise de componentes independentes (ICA) para representar, mediante transformação linear (ver figura 3.5), as assinaturas dos voxels em um espaço onde a classificação manual é mais simples, podendo, em geral, ser satisfatoriamente realizada utilizando-se planos separadores perpendiculares aos eixos coordenados do novo espaço. A ICA é utilizada para inferir os coeficientes da transformação linear mais apropriada para simplificar a representação das assinaturas.

Tzeng et al. [TZE 2003, TZE 2005] implementaram funções de transferência multidimensionais como redes neurais do tipo *back propagation* capazes de decidir sobre a visi-

bilidade dos voxels. Para o treinamento de uma rede, são fornecidos voxels selecionados pelo usuário — sua assinatura multidimensional, na verdade — e também a visibilidade desejada para esses voxels, caracterizando um processo de treinamento supervisionado. O conjunto de voxels para o treinamento é definido por uma interface onde o usuário visualiza as fatias bidimensionais do dado volumétrico e pinta, sobre as fatias, com uma cor, os voxels que devem ser visíveis na imagem final e, com outra cor, os voxels que devem permanecer invisíveis.

Tzeng e Ma criaram também, com o propósito de visualização volumétrica, uma técnica de classificação discreta de voxels contendo assinaturas multidimensionais [TZE 2004]. O usuário alcança a classificação desejada realizando operações de junção ou bipartição sobre as classes existentes e define propriedades visuais para cada classe de voxels. Numa etapa de preprocessamento, uma classificação inicial é automaticamente estabelecida, respeitando critérios de similaridade entre voxels; e, após, nas operações de bipartição, executadas com rapidez, os mesmos critérios são respeitados na geração das classes resultantes.

3.4 Comentários Finais

Existem ainda diversas outras técnicas de especificação de funções de transferência. Merecem destaque aquelas que se baseiam na teoria da probabilidade para classificar voxels considerando as características do processo e do dispositivo de obtenção de imagens tridimensionais, como parâmetros do ruído e da distribuição dos valores para cada material digitalizado. O trabalho de Kniss et al. [KNI 2005] é um dos mais importantes exemplos desse tipo de abordagem. Conhecimento capturado de especialistas também pode ser empregado na geração de funções de transferência, como mostraram Salama et al. [REZ 2006]. Esses autores criaram modelos de funções de transferência que são previamente ajustados por especialistas para diversos dados volumétricos de uma mesma categoria, visando à visualização de uma determinada estrutura. Os especialistas definem as FTs, para os vários volumes, ajustando um amplo e versátil conjunto de parâmetros, mas essa complexidade é ocultada do usuário final. É feita a análise dos componentes principais (PCA — *principal components analysis*) das instâncias dos parâmetros especificadas pelos especialistas na definição das FTs, com objetivo de representar adequadamente esses valores, com muitas dimensões, em um espaço simplificado, com um número muito menor de variáveis. Isso permite ao usuário final definir funções de transferência, para dados volumétricos da mesma natureza, ajustando apenas alguns poucos parâmetros. A análise de componentes principais é uma operação linear que produz, no espaço vetorial em que as amostras estão distribuídas, uma base vetorial onde cada vetor é associado a um grau de importância. Resumidamente, os vetores de maior grau de importância são os componentes principais, indicando as direções em que o conjunto de amostras encontra-se mais distribuído.

4 ESPECIFICAÇÃO DE FUNÇÕES DE TRANSFERÊNCIA UNIDIMENSIONAIS

Este capítulo reproduz, excetuando o *abstract*, o conteúdo do artigo intitulado "Volume Visualization and Exploration through Flexible Transfer Function Design". Trata-se de uma versão estendida do artigo publicado nos anais do SIBGRAPI 2006 [PIN 2006], submetida, por convite aos melhores trabalhos do SIBGRAPI, para a revista Computer Graphics Forum[1], principal revista da Eurographics Association.

Resumidamente, o artigo apresenta uma ferramenta interativa para visualização volumétrica que provê especificação semi-automática de funções de transferência de cor e de opacidade. A ferramenta é capaz de gerar funções de transferência para ênfase de bordas, oferece geração estocástica evolutiva de FTs e permite especificação manual através de interação no domínio da FT e, também, no domínio espacial. Propõe-se uma interface de dois níveis que, no primeiro nível, permite múltiplas visualizações simultâneas, de um mesmo dado volumétrico, com diferentes funções de transferência, como no trabalho de Prauchner et al. [PRA 2005a, PRA 2005b]. O segundo nível apresenta uma única visualização do volume, com melhor resolução, e uma representação gráfica detalhada da FT empregada, que pode ser manualmente modificada. As técnicas de especificação de funções de transferência combinadas na ferramenta proposta são complementares, permitindo a construção fácil e rápida de TFs úteis, bem como exploração eficiente de dados volumétricos.

4.1 Introduction

Volume rendering is widely known as a set of methods for the visualization of large three-dimensional (3D) scalar or vector fields, mainly in medical and scientific data exploration. In these areas, one often deals with 3D images, like those obtained from CT and MR devices or from numerical simulation data. Volume rendering techniques and algorithms are well described in the literature [BRO 2001], and can be classified as isosurface extraction based methods and direct volume rendering (DVR) methods. The former methods extract polygonal meshes representing isosurfaces in the volume, and then use the traditional rendering pipeline to display the meshes (see [LOR 87] as an example). On the other hand, DVR methods display volume data without building an intermediate geometry. DVR and its advantages were first described by Levoy [LEV 88]. Modern graphics hardware allows volume rendering at interactive rates in both approaches.

To obtain useful images through direct volume rendering, voxels need to be classified in order to determine which ones must be displayed. This classification is typically

[1] http://www.eg.org/EG/Publications/CGF

performed through transfer functions (TFs), which associate values of optical attributes to voxels according to their values. Opacity and color are the most common optical properties used in TFs. The degree of opacity can make a voxel more or less visible, and is normally used to emphasize voxels in the boundary between different homogeneous regions of the volume [KIN 98], which can be displayed with different colors. Other optical properties may also be used, like spectral reflectance [LUM 2004a], specular reflection coefficients [BER 2005] and light scattering coefficients [KNI 2003b]. The information conveyed by the image built from volume data is, therefore, highly dependent on the quality of the transfer function. However, TF design is a non-trivial and unintuitive task, and has been referred as one of the top ten problems in volume visualization [PFI 2001].

One-dimensional transfer functions take into account only scalar voxel values, and are the most common TFs, although having a limited classification power. On the other hand, multi-dimensional TFs allow more freedom in voxel classification by taking as arguments vectorial values or combinations of local measures of scalar fields, like derivative values [KNI 2002b, KNI 2001], neighborhood, position [TZE 2005], curvature [KIN 2003, HLA 2000] and statistical signatures [TEM 2001]. Notwithstanding, the design complexity grows with the size of the transfer function domain [KNI 2002b], and the memory required to implement truly multi-dimensional TFs restricts their application [KNI 2003a]. In this work, we adopted one-dimensional transfer functions due to their simplicity and low memory requirements, since they can be implemented as small lookup-tables. Furthermore, the pre-integrated volume rendering technique, proposed by Engel et al. [ENG 2001], allows high quality direct volume rendering at interactive rates using one-dimensional TFs.

Designing transfer functions with no assistance leads to a trial-and-error approach. Therefore, several automatic and semi-automatic techniques for specifying TFs have been proposed [BAJ 97, FAN 98, FUJ 99, HE 96, KIN 98, MAR 97, PEK 2001, TEM 2001]. They can be guided by the analysis of volumetric data (*data-driven*) or of the generated images (*image-driven*) [PFI 2001]. In any case, to make the process less frustrating and less time-consuming, a rapid feedback must be given to the user at real-time rendering frame rates.

The main contribution of our work is a two-level interface that combines a set of useful tools for semi-automatic one-dimensional transfer function design and fast data exploration in an interactive workspace. The first level of our interface presents several thumbnails of the volume data rendered with different TFs, allowing immediate insight of the main structures in the data set. The second level shows a detailed visualization of a single TF as well as the resulting rendering. TFs can be easily generated or refined using semi-automatic boundary emphasis, stochastic evolutive search, and manual design aided by dual domain interaction [KNI 2001]. These three approaches are complementary and were successfully combined in this work, improving the idea of two-level interaction proposed by Prauchner et al. [PRA 2005a].

This chapter extends our previous work [PIN 2006] with an improved interface and an experimental evaluation of our approach. We implemented a history tree that keeps track of the transfer function evolution and allows the user to go back to a previously specified TF. The evaluation was performed with the help of subjects who have experience in 3D scientific visualization. This way we tested the usability of our methods with potential users. We also implemented a set of geometric tools for volume inspection inspired by the work of Dietrich et al. [DIE 2004], but their description is beyond the scope of this article.

The chapter is organized as follows. Section 4.2 discusses the closest related work. Section 4.3 describes the proposed interface and the TF design techniques provided within it. Implementation details are addressed in Section 4.4, while in Section 4.5 we present the evaluation of our tools. At last, in Section 4.6, we draw some conclusions and point directions for future work.

4.2 Related Work

The transfer function specification problem has received much attention from researchers. Traditional approaches rely on the user's effort in adjusting control points of a graphic plot of the transfer function [KON 2000]. The control points — scalar values associated to values of optical attributes — are then interpolated in order to build the TF. However, with no clues or prior knowledge about the data, this is a "blind process". Some data-driven approaches provide to the user higher-level information [BAJ 97, PEK 2001] that help in obtaining insight about the data distribution and support manual TF design. Some methods hide the TF from the user through abstractions [TZE 2003, HE 96], while others offer a simplified space for TF specification [KIN 98, REZ 2006].

Kindlmann and Durkin [KIN 98] proposed a derived space for specification of opacity transfer functions in which the user assigns opacity levels to voxels as a function of the distance between the voxel and the nearest boundary. The authors assume that boundaries are smoothed by a Gaussian filtering process due to the Gaussian frequency response of 3D scanners. Informative histograms are built relating voxel scalar values with first and second derivative values. From these histograms, the mean first and second derivative values associated to each voxel value are used to estimate the distance to the nearest border. Boundaries often need to be emphasized, thus voxel values with small estimated distances are associated to large opacity values.

The Design Galleries method [MAR 97] is based on the generation of a wide set of transfer functions through a dispersion algorithm, in a pre-processing phase. Then, the obtained TFs are used to render thumbnails that are grouped by similarity and presented in an interface where the user can pick and zoom in the most appealing thumbnails. The stochastic approach for TF design proposed by He et al. [HE 96] also uses galleries of thumbnails. The authors represent TFs as vectors of control points and the smooth interpolation of the control points produces the actual transfer function. The method employs genetic algorithm operators to create new transfer functions from previous ones. The mutation operator changes control points, while the crossover operator produces new TFs by concatenating sub-vectors of control points from different TFs. Given a set of TFs, and their respective rendered images presented as thumbnails, the "best" ones are selected as parents, either by the user or through automatic evaluation based on objective metrics. The initial TFs are randomly specified and the TF population evolves as the parents are selected and new transfer functions are generated by applying the operators mentioned above to the current parents.

Prauchner et al. [PRA 2005a] used Kindlmann's method to classify the voxel values by the estimated distance to the nearest border. The voxel scalar values with the smallest distances are selected and random subsets of these values are then built. The values in the subsets are used as control points which receive random color and random opacity value. Each transfer function is obtained by interpolating the control points in one of those subsets. The generated TFs are used to render a gallery of thumbnails of the volume data, similarly to the Design Galleries method. This is the first level of the two-level interaction

interface proposed by Prauchner et al. in [PRA 2005a]. In the second level, the user can visualize a selected thumbnail in high resolution and refine its TF by manually adjusting the control points. The thumbnails can be re-generated at any time using new TFs.

Our method [PIN 2006] improves the transfer function specification process by combining features from the approaches referred above into a general-purpose interactive workspace that implements image-driven and data-driven TF design through a two level interface. The set of features provided by our interface is described in the next section.

4.3 Transfer Function Specification

Most researchers agree that transfer function specification methods should not overload the users nor exclude them from the process [PFI 2001]. The quality of a transfer function depends on the amount of information conveyed by the generated image — a subjective metric. Therefore, it is hard to automatically evaluate how "good" a TF is. Fully automatic TF specification methods may miss important features of the volume while completely manual TF design may demand a lot of effort and time, mainly from users who do not have prior knowledge about the data. In this section we outline some features that are desired in TF design tools and then we present our method.

4.3.1 Requirements for Transfer Function Design Tools

A TF design interface should offer useful information about the data as well as guidelines for the TF building process. The interface may also suggest sets of TFs based on heuristics or mathematical criteria, but the choice between the alternatives must be made by the user. Abstractions of the TF specification process can also be applied with success as shown by Tzeng et al. [TZE 2005] and Rezk-Salama [REZ 2006]. In any case, the user makes decisions and sees the results; hence immediate feedback is needed to make the process continuous, with minimum cognitive effort.

Experienced users working with known data can find good transfer functions with relatively few interactions with a simple manual TF editing interface. However, when the data is unknown, a trial-and-error approach may not be a good choice. In these cases, the two-level interface for TF specification proposed by Prauchner et al. [PRA 2005a] is very useful. The different thumbnails presented in the first level allow an immediate insight of several volume structures. However, the potential benefits of a two-level interface were not thoroughly explored in that work. The TFs of the most appealing thumbnails out of the set presented in the first level of interaction could also be used as a basis to generate other transfer functions. Moreover, the manual refinement should be more flexible and the interface must allow interaction in both transfer function domain and spatial domain in order to improve the understanding of their correlation. Changes in the transfer function must have immediate effect in the rendered volume, and voxel values queried in the spatial domain must be also shown in the transfer function domain [KNI 2001]. We try to fulfill all these requirements with our two-level interface.

4.3.2 Two-Level Interaction Overview

Our method combines three techniques for transfer function design: boundary emphasis; stochastic search; and manual specification with dual domain interaction. The two-level interaction is summarized in Figure 4.1.

User interaction starts at the first level, where thumbnails generated initially through the boundary emphasis technique are shown. At this level, the user can generate new TFs

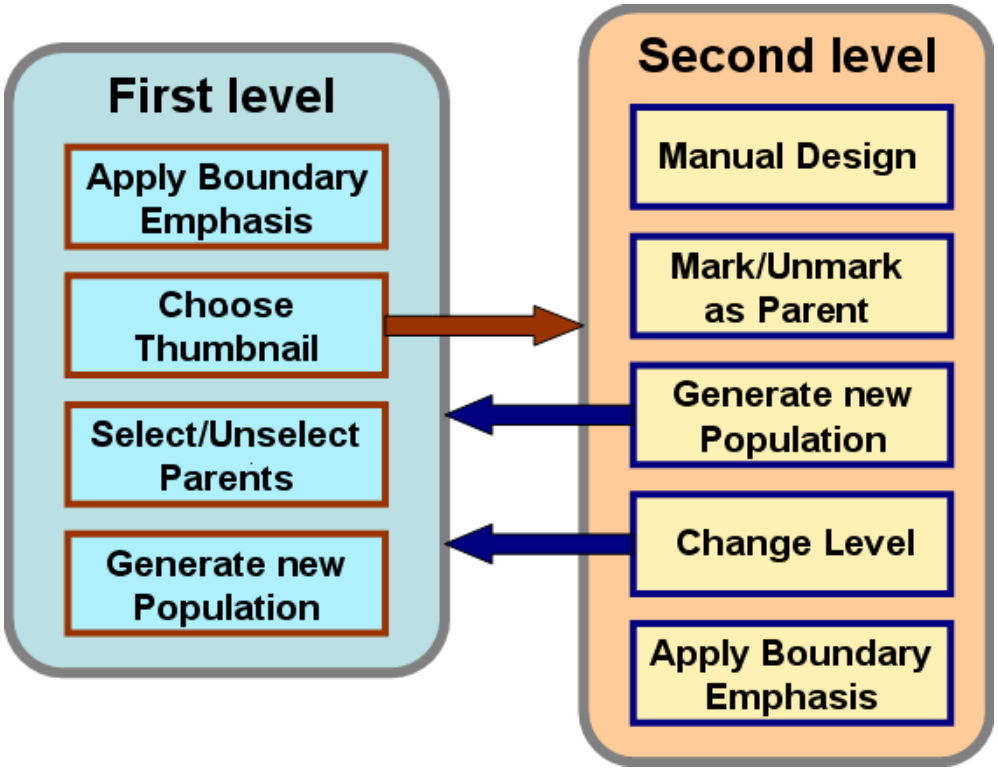


Figura 4.1: The main actions allowed in each level of the interface, the arrows representing level changes.

(new thumbnails) either reapplying boundary emphasis or selecting thumbnails as parents and then applying stochastic search. The user can also select a single thumbnail, action that leads to the second level of interaction, where he/she can apply boundary emphasis and manual design only in the selected TF. Any changes in the TF selected for the second level do not affect the other TFs exhibited in the first level. The user can change back to the first level anytime. The TF design techniques combined in the proposed interface are described in the next subsections.

4.3.3 Boundary Emphasis

The initial set of transfer functions for rendering the thumbnails in the first level is generated using a boundary emphasis technique, based on a 3D histogram that relates voxel values with the first and second derivatives, following Kindlmann and Durkin [KIN 98]. Like Prauchner et al. [PRA 2005a], we use the histogram approach to choose control points for the transfer functions. The control points correspond to voxel values that are probably in a boundary region, and receive opacity values chosen randomly between a maximum and a minimum value. When building the TF lookup table, an interval of voxel values around each control point is assigned with non-zero opacities in a decreasing way depending on the distance from the control point, which builds triangular shapes in the opacity TF. The width of the intervals is adjustable. The color transfer functions are also designed to emphasize boundaries. Each control point is associated to two different random colors: one for voxel values larger than the control point's value, and another for the smaller ones. Between two control points, the color values are linearly interpolated. This way, boundary regions present color discontinuities, as suggested by Fujishiro et al. [FUJ 2000]. Figure 4.2 shows the benefits of this method for color TF generation as

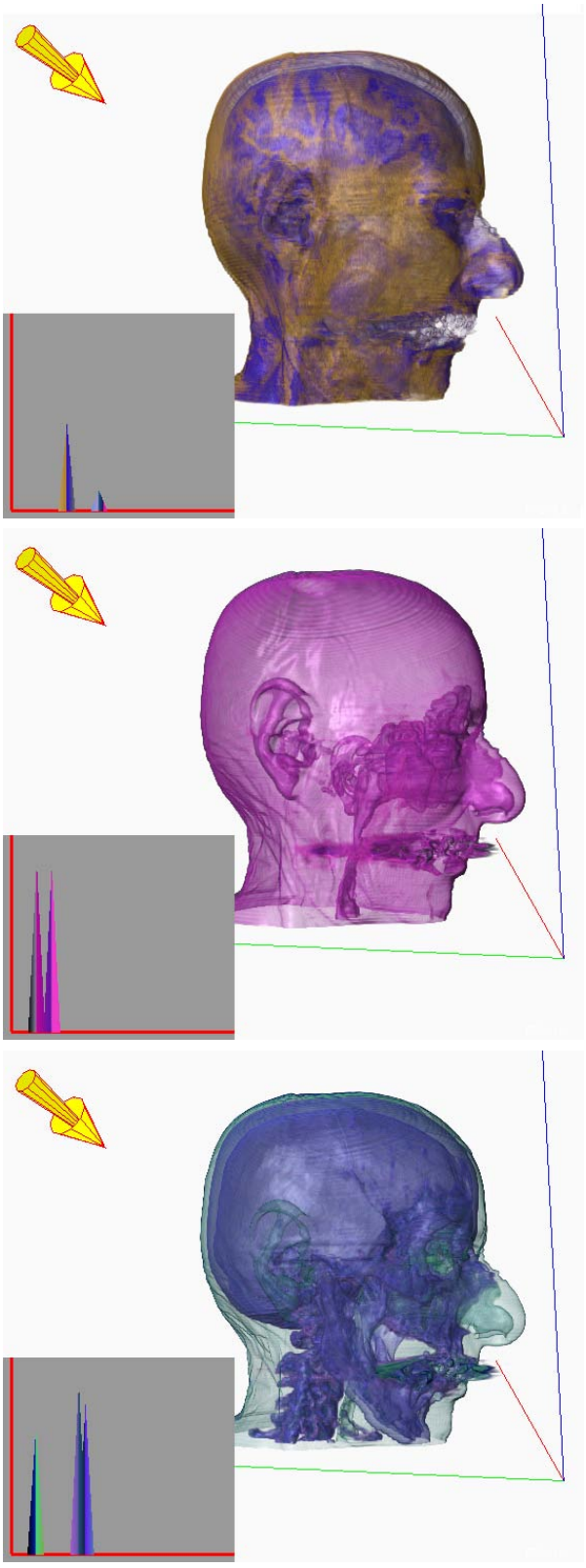


Figura 4.2: The cadaver data set rendered using TFs generated by our boundary emphasis algorithm. The employed TF is displayed in the lower left corner of each image. The triangular shapes in the TF are responsible for the color discontinuities at specific voxel values, which help to emphasize boundaries. In the top image, the color discontinuity allows clear visualization of the saliencies in the outermost part of the brain despite the similarity between the voxel values. The arrow indicates the light incidence direction.

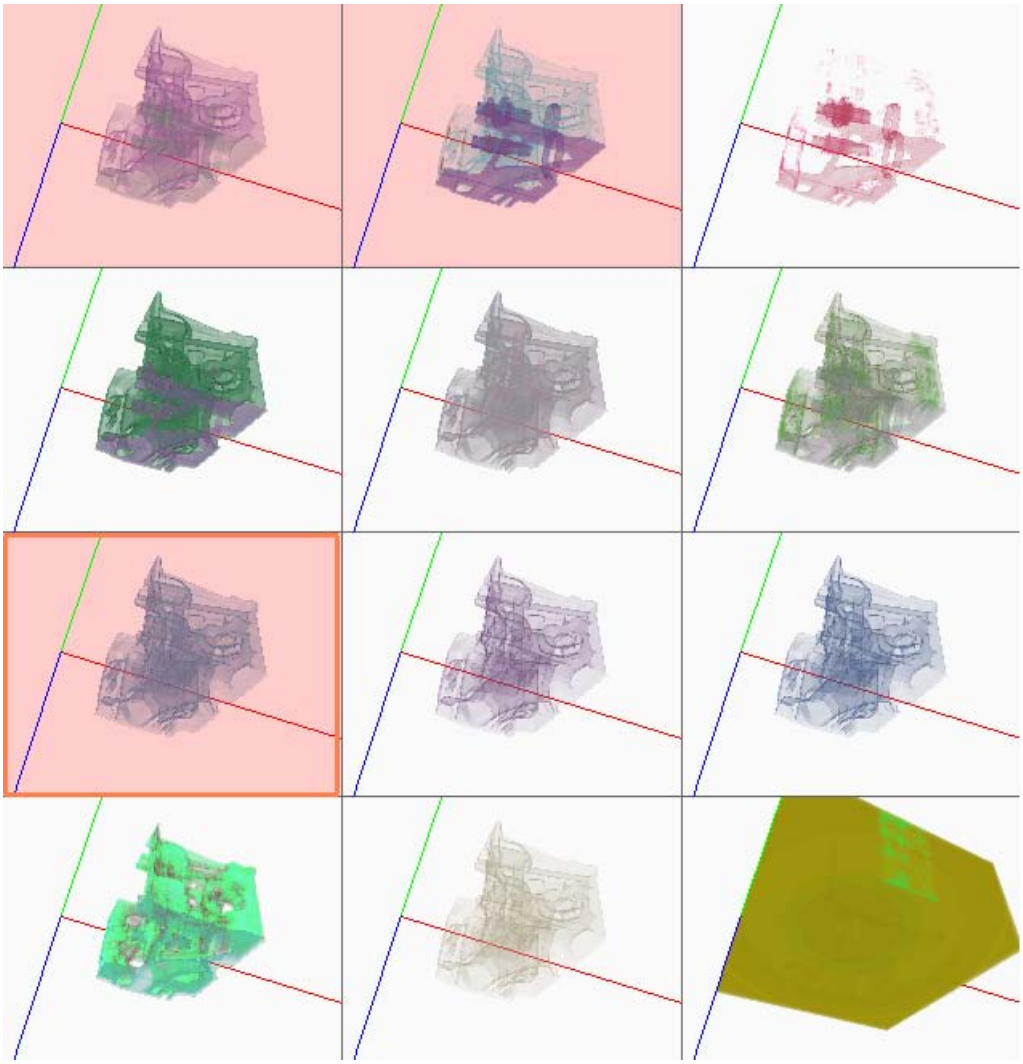


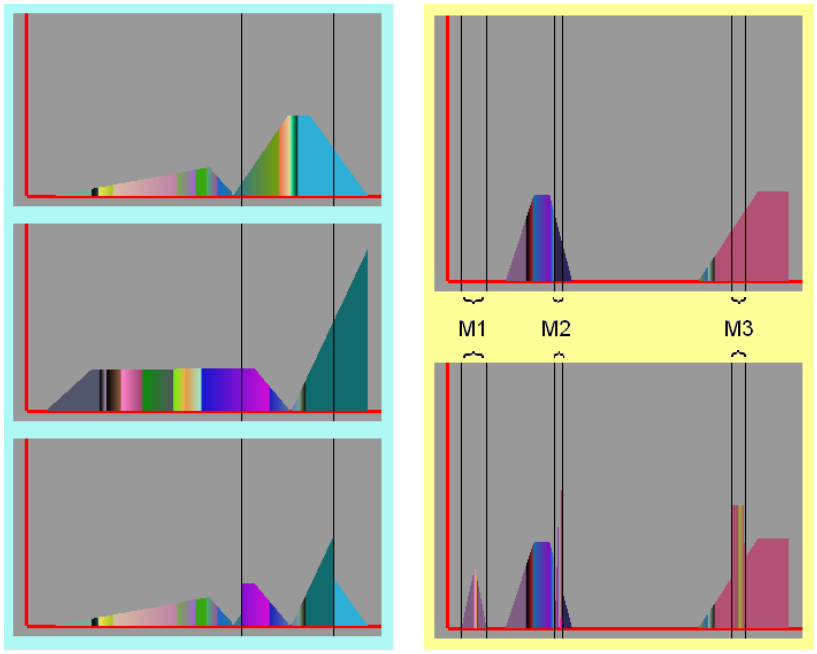
Figura 4.3: Selection of the parents for a new population of TFs in the first level of interaction. The selected thumbnails are presented in red background. The volume is the well-known engine data set.

well as the triangular shapes in the opacity TF centered at boundary values. In the TF graphic plots, voxel values are represented as the horizontal axes, while opacity values are represented as the vertical axes. The color TF is directly represented.

4.3.4 Stochastic Evolutive Transfer Function Design

In order to generate the TFs applied to render thumbnails in the first level, we not only use the histogram approach by Kindlmann and Durkin [KIN 98], but also a more flexible version of the technique proposed by He et al. [HE 96]. While their implementation of genetic algorithms codifies TFs as vectors of control points taken as genes, our codification of a transfer function is the TF lookup table itself, where the entries are the genes. This way we can easily combine the stochastic search with any other method for one-dimensional transfer function design.

The method works as follows. Given a set of thumbnails and associated TFs, the user selects the best ones to be the parents of a new population, as shown in Figure 4.3. Then, the parent TFs produce descendants by mutation, crossover or both, in a random way, and



(a) (b)

Figura 4.4: (a) The crossover operator. The topmost two TFs are the parents and the bottom one is the resulting TF. (b) Three types of mutations applied to a TF. The bottom TF is the result.

a new set of TFs (and thumbnails) are obtained, preserving only the parents. By selecting and unselecting members of the new set (population), the user can define the parents of the next TF population. If the quality of a particular thumbnail could not be properly evaluated due to its low resolution, the image can be better inspected in the second level of interaction, which shows a high resolution rendering of the volume. The crossover and mutation operators are described below.

- *crossover* - the crossover operator randomly takes two transfer functions among the ones selected as parents and cuts them at two points, also randomly determined. The interval between the two points of one TF is combined with the two outer intervals (defined by the same two points) of the other TF, as shown in Figure 4.4a. In the graphic plot, the horizontal axis represents voxel values and the vertical one represents opacity. The vertical black lines are the two cutting points.

- *mutation* - the mutation operator can be applied repeatedly to a TF. Figure 4.4b shows a TF and three types of mutation (M1, M2 and M3) applied to regions defined by pairs of points marked as vertical black lines. M1 is a triangular mutation: an opacity value is defined for the center of the mutation and a linear interpolation is applied between the central opacity value and the opacity values at each border of the mutation. M2 is a random mutation: the opacities are randomly set for the voxel values around the mutation center. M3 is a rectangular mutation. The center and the range of the mutations as well as the colors and the opacities are randomly set between limited intervals. There is also a fourth type of mutation: a random global scale applied to all opacity values.

The genetic operators produce high frequency features in the transfer function domain, which might cause artifacts in the images. Since we implemented Engel's pre-integrated

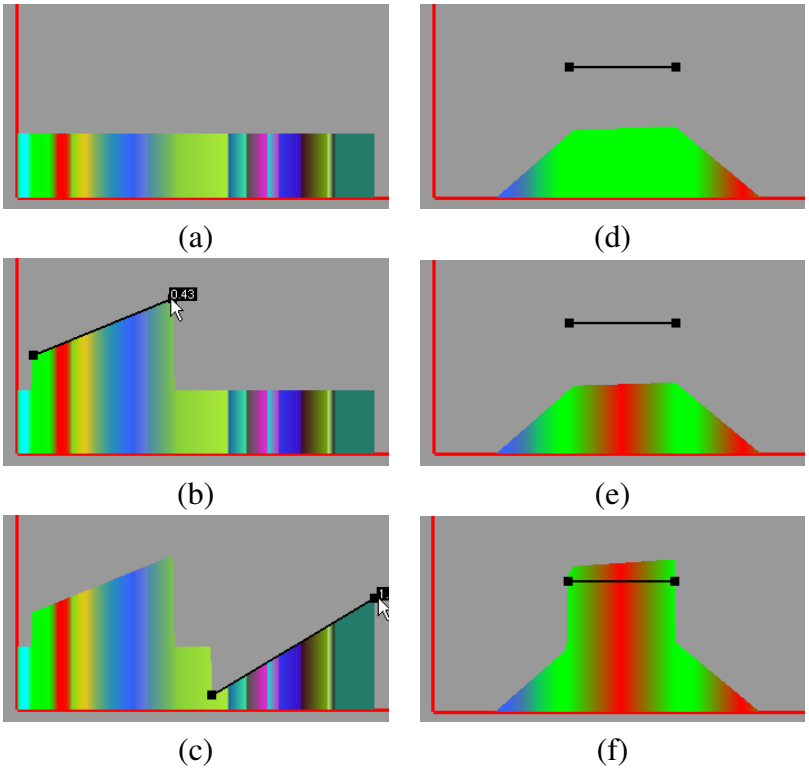


Figura 4.5: A TF (a) and the results of two design steps (b and c). A second TF (d) with a selected interval of influence; red is assigned to the interval (e); and the opacities for the same interval are scaled (f).

volume rendering method [ENG 2001] that deals very well with high frequencies, these features do not affect the images rendered using our interface.

4.3.5 Manual Specification

We propose a simple interface for manual specification of TFs at the second level of interaction. The user draws directly on the transfer function graphic plot. By dragging the mouse with the left button pressed, the user draws a line, setting the shape of the opacity transfer function, as shown in Figure 4.5(a-c). Using the mouse right button, instead, the user sets an interval of influence in the voxel value axis. The selected interval is affected by color selection or scale changes in the opacity values. Once an interval is set, the user can select a color from a color picker. The interval is painted with this color linearly interpolated with the existing ones, as shown in figure 4.5(d-e). The center of the interval receives the selected color and the contribution of that color decreases toward the borders of the interval. Using the keyboard, the user can scale the opacities in the selected interval (see figure 4.5f). During the manual TF modification, the voxel value under the cursor is shown in a small box (fig. 4.5(b-c)).

When the user is able to make correlations between the transfer function domain and the spatial domain, manual TF design becomes simpler. The presentation of colors in the TF graphic plot helps in this way. The colors used to render the volume are also used to paint the TF plot, thus providing information about the distribution of voxel values in the volume (spatial domain). In addition, we implemented a volume investigation scheme that can also be used to edit the transfer function, as a dual domain interaction approach. The user can use a clipping plane to sweep the volume, as shown in Figure 4.6. The slice

sampled by the clipping plane is displayed at the top right as a gray level image directly showing the voxel values. As the mouse pointer moves on the clipping plane or on the slice, the value of the pointed voxel is presented, and a small white square is displayed on the TF graphic plot indicating the position of that voxel value in the transfer function domain. By clicking using the left mouse button, the opacity value associated to that voxel value is increased. This scheme allows direct volume inspection and is very useful to emphasize isosurfaces passing through a specific point in the volume.

4.3.6 History Tree

As discussed by Bavoil et al. [BAV 2005], it is important to keep track of the evolution of a visualization. In their work, changes made to the pipeline and parameters during the process of building an expressive visualization are recorded to allow retrieving previous versions as well as to reproduce results later on. This way they also provide a manner to reuse configurations for visualizing different data sets.

In our work we have implemented a persistent history tree of the transfer functions used to produce visualizations. In the second level of interaction the user is able to save the current transfer function. The TF is then stored as a node of the tree, together with a snapshot of the respective rendered volume. The history tree is displayed in a separate window, each node graphically represented by the snapshot taken from the visualization. Any saved transfer function can be retrieved by clicking on the respective tree node. The tree (see Figure 4.7) can be scaled and translated by dragging the mouse and its layout is automatically maintained. The user can also delete branches of the tree when it becomes too large.

4.4 Implementation Details

Our volume rendering tool was implemented in C++ using the GLUT and GLUI libraries for the interface (see Figure 4.8), and OpenGL and CG for volume visualization. The rendering algorithm runs in GPU and is based on 3D texture sampling using view-aligned slices as proxy geometry. The number of slices can be changed by the user, and is automatically reduced when the volume is being rotated, to guarantee interactive rates in both levels of interaction. It is worthy to mention that Engel's pre-integrated volume rendering method [ENG 2001] allows high-quality rendering even with a relatively small number of proxy planes.

In our implementation, the volume data is stored in the GPU memory as a 3D texture with eight bits of precision. Transfer functions are represented as lookup tables with 256 entries and pre-integrated tables are stored as two-dimensional RGBA textures. We can pre-integrate color and opacity contributions neglecting color attenuation inside volume slabs, which is much faster than full pre-integration. We apply this simplified pre-integration in the first level, since an action can modify several transfer functions (36 TFs with the respective thumbnails in our implementation). In the second level of the interface a single transfer function is under modification; then, we apply full pre-integration without compromising the interactivity. The thumbnails are replaced by a single rendered volume, like in Figure 4.2, which can be edited or inspected using the dual domain interaction tool (Figure 4.6). Volume shading can be enabled or disabled by the user and is based on the per-fragment Phong lighting model using the gradient of the scalar field as normal vectors.

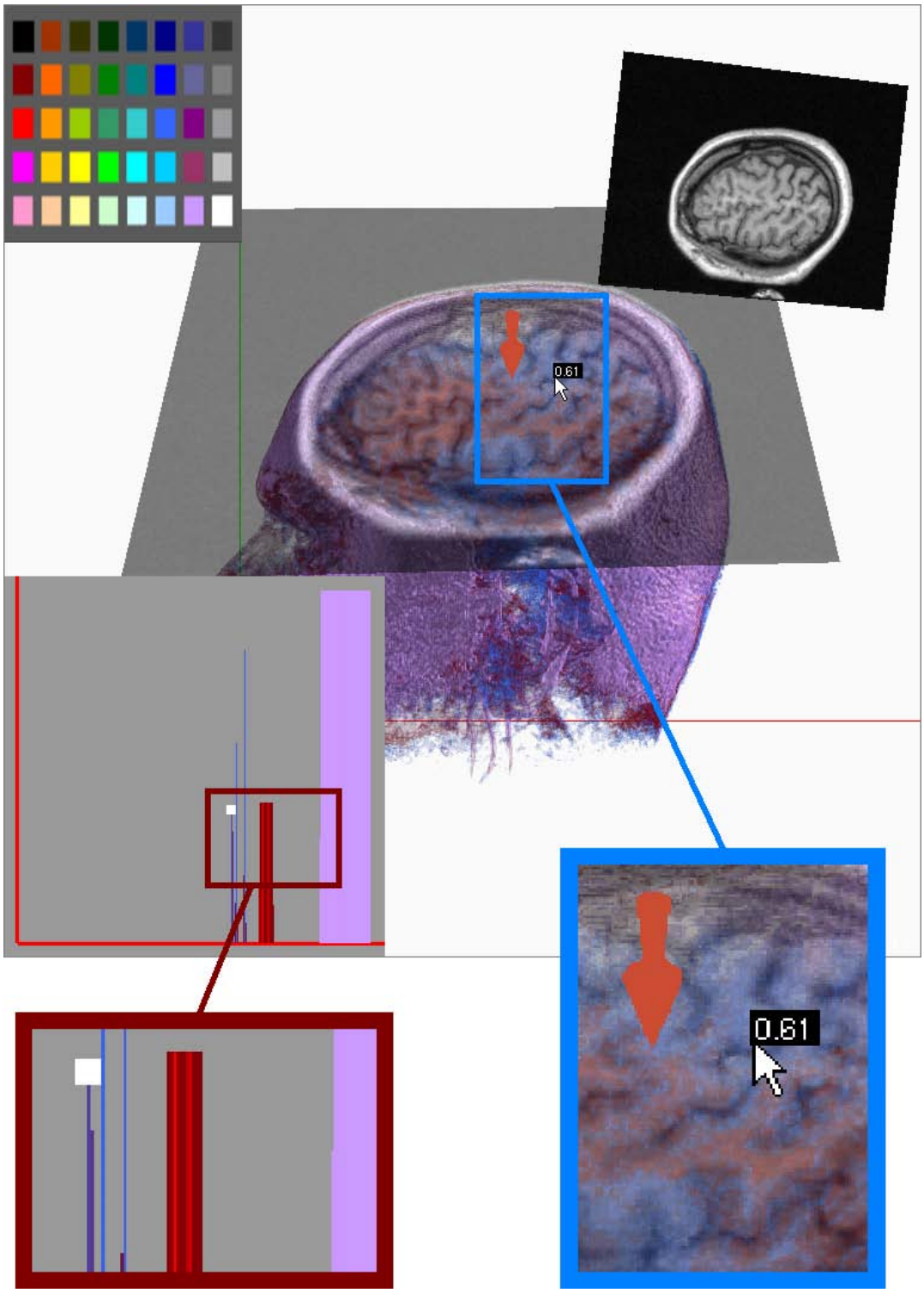


Figura 4.6: A clipping plane is sweeping the volume, the red arrow representing its normal. The voxels on the clipping plane are displayed as a grayscale slice at the top right. The mouse pointer is querying a voxel on the plane (blue rectangle). The white square in the TF graphic plot (see the enlarged red rectangle) represents the value of the queried position in TF domain.

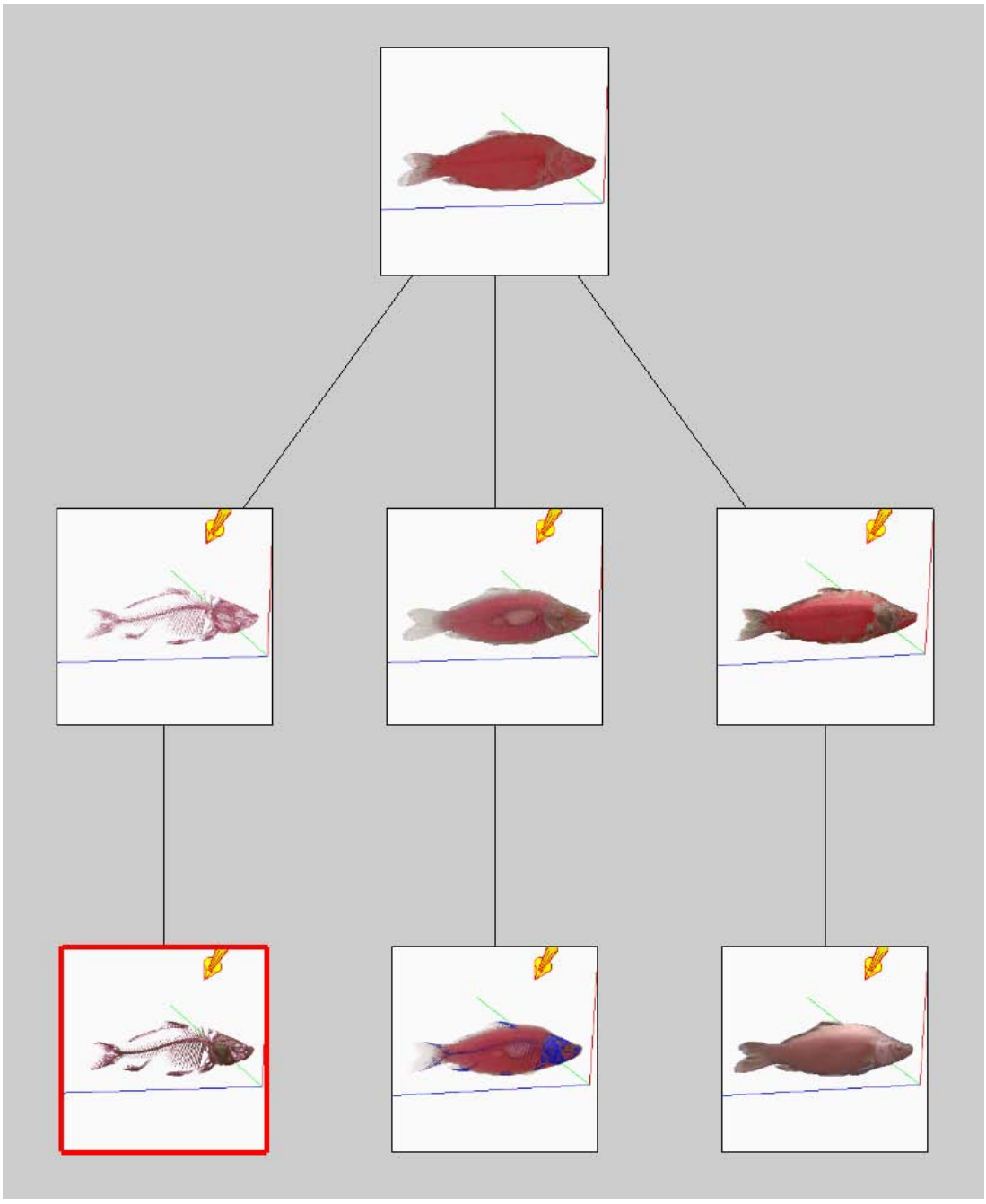


Figura 4.7: The tree maintains the history of modifications along the TF design process. The red square indicates the current transfer function.

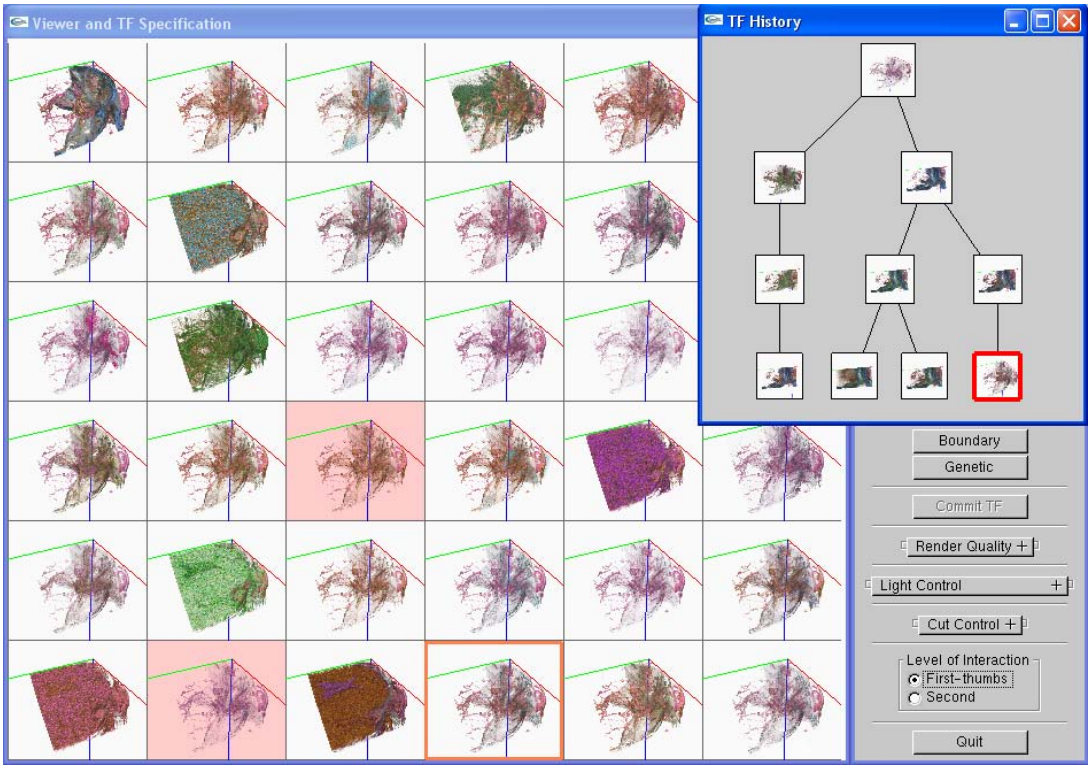


Figura 4.8: A snapshot of the interface of our framework. Three windows shown, the rendering window; the history window; and the control window.

4.5 Evaluation and Discussion

In this section we describe the experiment performed in order to evaluate our interface and then we present the results. Assuming that our manual TF design tool with dual domain interaction would be at least as good as the traditional interfaces based on adjustment of control points, we wanted to prove the usefulness of our interface by comparing the time needed to build suitable visualizations of selected data sets using all resources of the workspace with the time spent when using only the manual design tool.

Our subjects were four students with experience with scientific 3D visualization. Two of them have already developed their own volume visualization tools as well as transfer functions design schemes; the others are only familiarized with volume visualization. The experiment consisted of a training step, in which the subjects used our framework to visualize the Visible Male Head data set (Figure 4.9, a and b), followed by four visualization tasks, two using all resources of the interface and two using only the manual design tool. All the tasks were to obtain visualizations of two data sets (Chest and Cadaver Head), revealing the volume structures shown in Figures 4.9, c, d, e and f. For each task we presented one of these four images and ask the users to produce a visualization of the depicted structure by designing a proper transfer function. The subjects had never visualized those two data sets and they stopped the design process when a good visualization, according to their judgment, was achieved.

The experiment was performed in two days — one visualization of each data set in each day — to reduce the effect of accumulating experience during the first task. We also prevented orientation clues by omitting the axes from the images presented to the users. The combination of data set, structure to be visualized, order of tasks and the TF design tools allowed was different for each subject. During the experiments we recorded

all interaction steps performed by the subjects, and after they answered some questions about the interface as a subjective evaluation of its usefulness.

We had a very good feedback from the users. All subjects agreed that the interface's first level provides immediate understanding of the most important structures in the data sets. According to them, the boundary emphasis tool produces, at least, a good starting point, and the stochastic search is very useful for combining TFs that emphasize different structures; exploring new possibilities; and refining transfer functions. The subjects also agreed that our framework is simple to use and fast, due to real-time rendering and interaction.

Based on the answers given by the subjects through the questionnaire and the list of actions recorded during the tests, we concluded that the combination of TF design tools is useful and the different techniques were well explored by them. All subjects first used boundary emphasis to obtain appealing thumbnails, which were often inspected in the second level and selected as parents, and then used for combination or refinement of TFs by applying stochastic search. As a final step, the users applied the manual design tool to further refine the transfer function of a chosen thumbnail to get the target image. The dual domain interaction tool (the clipping plane) was often used during manual design.

	Manual Design		Free Design	
	Task 1	Task 2	Task 1	Task 2
Subject 1	13:47	05:38	11:06	10:42
Subject 2	17:53	10:33	17:38	10:09
Subject 3	10:00	02:47	08:03	03:41
Subject 4	07:31	07:19	05:11	02:20

Tabela 4.1: Time spent in the visualization tasks (minutes : seconds). Subjects 3 and 4 are the most experienced users.

Based on the data recorded while the subjects were performing the visualization tasks, we could also verify that manual transfer function refinement is still a bottleneck, especially for the two less experienced users (subjects 1 and 2). For these two subjects, the time spent to perform the tasks using all resources was not significantly different from the time taken using only manual design, because even using all resources the time for the manual design step was high. Table 4.1 shows the time spent by the subjects to perform the tasks. However, an interesting finding was that the patterns of the design sessions performed by the two most experienced subjects were very different. Subject 3 preferred to use only manual design after a single application of boundary emphasis and a single use of stochastic search with only one or two parents. On the other hand, subject 4, when all design tools were allowed, employed only boundary emphasis and stochastic search, without manual refinement. In one session he applied boundary emphasis five times and stochastic search two times, selecting and unselecting parents 32 times. In the other session he applied boundary emphasis two times and stochastic search only one time, selecting and unselecting parents 11 times. Using all resources of the framework this user took only around five minutes to accomplish one task and around two minutes to accomplish the second one. Using only the manual design tool the times were higher (07min31secs and 07min19secs, respectively). In this case the framework turned the design process nearly two times faster. However, to validate our hypothesis that the TF design process using our framework is faster, we need to perform more tests.

At the end of the experiment we asked for suggestions to improve our framework.

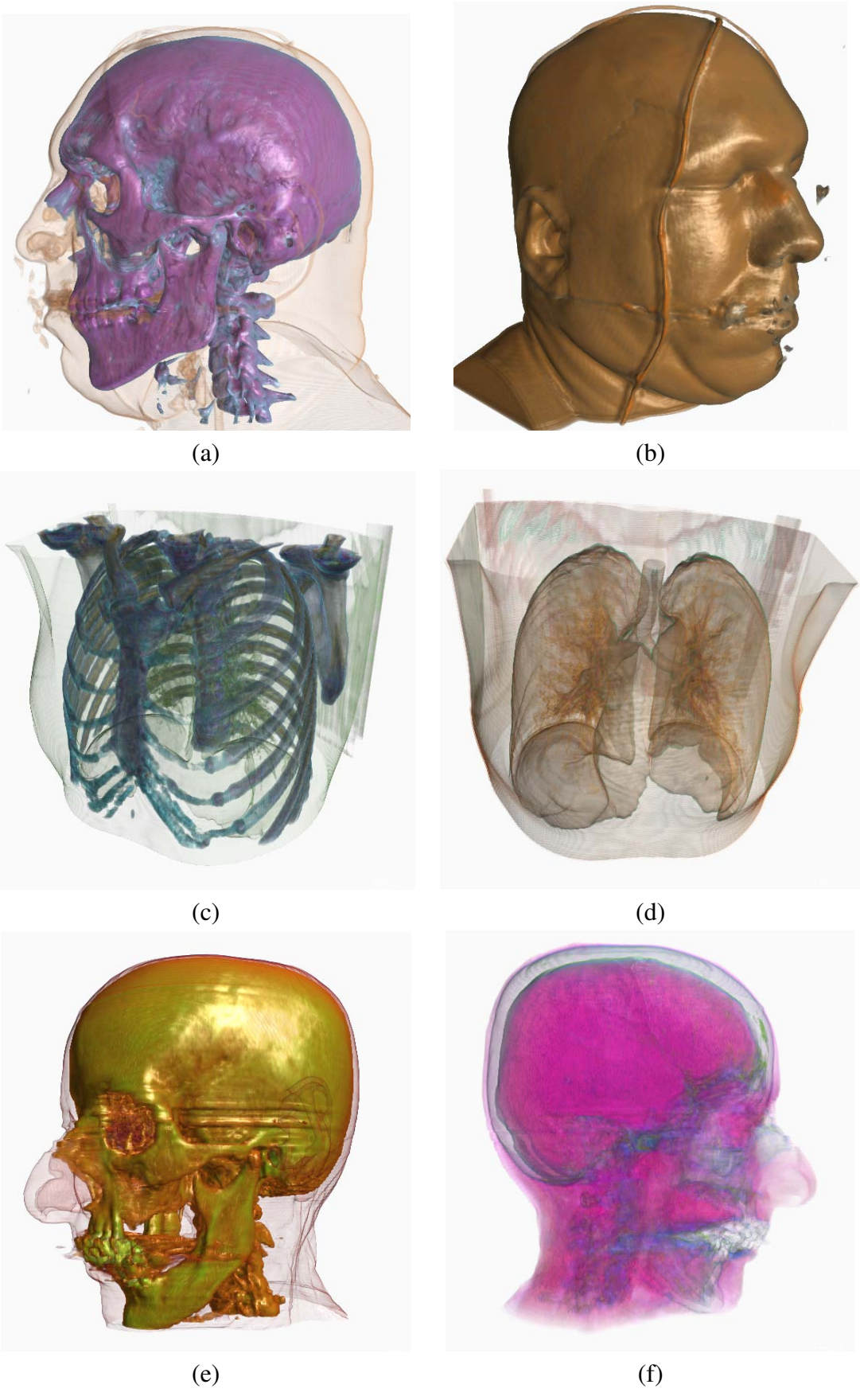
The subjects approved our manual design tool, but they agreed that it should also allow the use of opacity transfer functions with predefined shapes and adjustable control points. Two of the subjects also suggested “pan” and “zoom” operations on the transfer function graphic plot.

4.6 Conclusions and Future work

Despite the considerable attention received by the transfer function specification problem, TF design is still a hard task. We are far away from an ideal solution, but several TF specification methods have proved to be useful. We developed an interactive general purpose volume visualization tool with high-quality volume rendering by adapting, extending and combining known TF specification techniques. Compared to other methods, ours successfully combines two different classes of approaches (image-driven and data-driven), allowing abstractions during the TF design process without preventing user control, since a friendly manual editing interface is also provided. The subjective evaluation of our framework gave us a very good feedback, but we still need to test our approach with more potential users to prove that the transfer function design can be accelerated by our tool.

A particularly interesting future work is to extend our framework to design multi-dimensional transfer functions. TF domains with more than two dimensions are very difficult to visually represent and explore, thus abstractions of TF representation are an attractive solution. With this in mind, image-driven approaches, like stochastic evolutive design, deserve special attention. Besides, our two-level interaction interface is suitable for fast exploration of complex transfer function domains due to the simultaneous visualization of multiple possibilities.

As a limitation, our method suffers from noise in volume data, and thus another straightforward future work is to apply a pre-processing step of filtering. This step could not only remove noise, but also enhance features present in the volume as shown by Fang [FAN 98].



(a) (b) (c) (d) (e) (f)

Figura 4.9: Visualizations produced with our framework: the Visible Male Head data set (a and b), the Chest data set (c and d) and the Cadaver Head data set (e and f).

5 ESPECIFICAÇÃO DE FUNÇÕES DE TRANSFERÊNCIA MULTIDIMENSIONAIS

Este capítulo reproduz, excetuando o *abstract*, o conteúdo do artigo intitulado "Design of Multi-dimensional Transfer Functions Using Dimensional Reduction", aceito para apresentação e publicação na conferência EuroVis — Eurographics/IEEE Symposium on Visualization[1].

Resumidamente, este artigo propõe uma técnica de especificação de funções de transferência multidimensionais que utiliza mapas auto-organizáveis para perfazer redução dimensional das assinaturas dos voxels. A abordagem dá tratamento uniforme a dados volumétricos de dimensão arbitrária e permite a utilização de qualquer tipo de atributo de voxel como parte de sua assinatura. A especificação da função de transferência é realizada em um espaço simplificado através de uma interface simples, que permite a obtenção rápida de uma variedade de visualizações.

5.1 Introduction

In direct volume rendering (DVR), transfer functions (TFs) are used for emphasizing regions of interest inside volumes. The most common type of transfer function is the one-dimensional TF, which assigns optical properties (usually color and opacity) to voxels based only on their scalar value. Notwithstanding, one-dimensional TFs have a very limited classification power because they can not make distinction between volume regions defined by scalar values within the same range. On the other hand, multi-dimensional transfer functions can perform better classification because they take into account not only the scalar value of a voxel [KNI 2002b], but also other attributes such as gradient magnitude, directional second derivative, curvature [KIN 2003] and statistical measures [TEM 2001]. This way, sets of attributes are interpreted as multi-dimensional voxel signatures.

Designing an appropriate transfer function, even a one-dimensional TF, is a difficult task and much attention has been given to this issue in the literature [PFI 2001]. As its domain increases, the interaction with and the visualization of the transfer function become more difficult. Thus, specifying multi-dimensional TFs is a very difficult problem.

In this chapter we present a dimensional reduction method based on self-organizing maps (SOMs) and radial basis functions (RBFs) to simplify the design of multi-dimensional (nD) transfer functions. In a first step, nD voxel signatures calculated from the volume data are used to create a two-dimensional map; then, map coordinates are defined for all voxels. The transfer function design is performed in the two-dimensional map space (see details in Section 5.3). We explore the use of two types of SOMs: Koho-

[1] http://www.nvis.itn.liu.se/eurovis2007/

nen maps [KOH 97] and spherical self-organizing maps [SAN 2002]. Since we reduce the dimension of nD signatures, volume data of any dimension can be treated uniformly, and any type of voxel attribute can be part of the signature, allowing selecting those that provide meaningful visualizations. We also propose a simple and effective user interface based on dual domain interaction [KNI 2002b] as part of a visualization framework that uses graphics hardware to provide interactive volume rendering.

This chapter is organized as follows. Next section presents the closest related works, including a brief review of some concepts. Section 5.3 describes our approach in detail, while main implementation aspects are discussed in Section 5.4. Section 5.5 analyzes results we obtained with our method, while Section 5.6 draws some conclusions and points directions for future work.

5.2 Related Work

5.2.1 Multi-dimensional Transfer Function Design

The design of multi-dimensional transfer functions brings two main challenges: exploration of the TF domain and actual representation of the TF for rendering purpose. It is possible to explicitly define a multi-dimensional transfer function by interacting in its domain with proper tools. Kniss et al. [KNI 2002b] proposed a volume rendering environment containing a set of direct manipulation widgets for volume inspection, visualization of data distribution and design of three-dimensional transfer functions, using dual domain interaction.

However, the difficulty of exploring the transfer function domain increases with its dimensionality; therefore some approaches for transfer function design provide interfaces based on interaction in a simplified space. Region growing techniques were used by Huang and Ma [HUA 2003] to segment volume data from seed points specified by the user; voxel signatures of the segmented region were used to automatically design a transfer function. Tzeng and Ma [TZE 2004] clusterized voxel signatures by similarity allowing the user to specify the desired classification by successively splitting and merging the clusters. The user sees the results by associating visual properties to each material class. The same authors [TZE 2005] implemented multi-dimensional classification functions using neural networks and support vector machines. These functions were learned from training sets selected through a slice painting interface. The user paints the voxels of interest with a specific color, and the undesired ones with a different color. This way they implemented binary classification of voxels. Šereda et al. [SER 2006a] used hierarchical clustering to group voxels according to their LH signatures [SER 2006c]. The user navigates through the hierarchy searching for the branches corresponding to regions of interest. Takanashi et al. [TAK 2002] used independent component analysis (ICA) of multi-dimensional voxel signatures in order to represent them in a space where the classification is performed by moving axis aligned separation planes. Rezk-Salama et al. [REZ 2006] created models of transfer functions that are carefully adjusted by specialists for several data sets of the same type in order to reveal the desired structures. Then, they applied PCA to represent the parameter set of each model by a single variable with an associated semantic. The models can be reused for new data sets by setting only that variable. Ma et al. [MA 98] used SOMs to extract a reduced number of representative multi-dimensional voxel signatures from volume data. They built probabilistic classification functions based on these signatures.

Regarding the actual representation of TFs for rendering, Kniss et al. [KNI 2003a]

discussed the high memory requirements to store high-dimensional TF lookup tables and proposed Gaussian multi-dimensional transfer functions, which can be analytically evaluated. In another work [KNI 2005], Kniss et al. performed volume visualization by separating data classification from the transfer function. Classification is done by using a probabilistic approach, while the TF assigns optical properties to classes considering uncertainty.

One may think of our method as a non-discrete classification scheme (dimensional reduction) combined with transfer function design in an easily manageable space. Since we use self-organizing maps to reduce the dimension of the interaction space, we briefly review this topic in the next subsection.

5.2.2 Self-Organizing Maps

Kohonen's maps [KOH 97] are regular structures containing cells and neighborhood relations between them (in this work, a 2D grid topology is employed). Each cell contains an nD weight vector that represents a subset (class) of the mapped data. The map is built through an iterative unsupervised learning process in which the training cases are the multi-dimensional values to be mapped. For each presented training case, the winner cell (BMU) — the cell with the most similar weight vector — is identified (Equation 5.1). Then, the current weight vectors of this cell ($\vec{W}_t$) and of the cells in a limited neighborhood are modified. Each new weight ($\vec{W}_{t+1}$) vector is obtained through interpolation between the training case and the respective current weight vector (Equation 5.2). The coefficient of this interpolation for each cell depends on a predefined constant (η) and the neighborhood function (nf), which is one for the winner cell and decreases for the other cells according to their topological distance to the winner cell.

$$\begin{aligned} BMU &= C_i \mid \forall j \in [1,n], \\ &\quad d(\vec{W}(C_i),\vec{T}) \leq d(\vec{W}(C_j),\vec{T}) \end{aligned} \tag{5.1}$$

$$\begin{aligned} \vec{W}_{t+1}(C_i) &= (1-a) \times \vec{W}_t(C_i) + a \times \vec{T} \\ a &= \eta \times nf(td(BMU, C_i)) \end{aligned} \tag{5.2}$$

BMU is the winner cell (or best matching unit), C_i and C_j are map cells, n is the number of cells, d is a distance metric between vectors, $\vec{W}$ is a function that produces the weight vector of a cell and $\vec{T}$ is a training case. $\vec{W}_{t+1}$ is the new weight vector of a cell, $\vec{W}_t$ is the current weight vector, η is a constant between zero and one, nf is the neighborhood function and td is the topological distance between two cells. Usually, the distance metric is the weighted Euclidean distance, i. e., vector elements are multiplied by weights before the evaluation of Euclidean distance. The weights define the importance of each vector element.

The weight vectors are initialized with random values between zero and one. Then, the training cases are presented to the map in random order until it converges, so each training case is often presented to the map more than once. The balance between convergence time and map stability is determined by the value of η. The weight vectors must have the same dimension of the input data.

Spherical maps have the same features, except that they present a spherical topology that is built by subdividing the triangular faces of an icosahedron in order to obtain more nodes. In this work we use self-organizing maps to represent all voxels of a volume

data set in a two-dimensional space. We build the SOMs using, as training cases, multi-dimensional voxel signatures obtained from the data set.

5.3 Design of Multi-dimensional Transfer Functions

Our multi-dimensional transfer function design technique consists basically of three processes: building of a two-dimensional map from the voxels' multi-dimensional signatures; dimensional reduction of the nD signatures, where each signature is replaced by its coordinates in the map space; and specification of a transfer function in the reduced space — the 2D map. Therefore, the actual multi-dimensional transfer function is the composition of two functions: the dimensional reduction function and the transfer function in the map space.

Self-organizing maps have interesting properties for dimensional reduction. They can represent a set of multi-dimensional values (for example, a set of voxel signatures) in a compact way as well as group the values by similarity, according to a distance metric. Consequently, similar signatures have similar map coordinates. This feature facilitates the map exploration and understanding. Additionally, since the screen is a two-dimensional space, the exploration of 2D maps for TF design can be very natural.

5.3.1 Building of maps

The map building process starts with a preprocessing phase, when complex voxel signatures (such as derivatives and statistical measures) are extracted from the volume data and normalized. This way, each voxel has an nD signature (a set of scalar values represented as a vector) that can be used as a training case for the map building algorithm. It is important to mention that, depending on the source of volume data, there are many background voxels, which do not carry useful information (air around scanned objects in CT/MRI volume data, for example), and would influence the map due to their high occurrence. Upon user decision, they can be partially removed from the input set of the training process by a very simple region growing technique using as seeds the voxels identified as background in the most exterior regions of the volume.

The signatures of all non-background voxels are employed as training cases and presented in random order to the self-organizing map, and two types of neighborhood functions are applied. In a first stage we define the overall aspect of the map by training it using a Gaussian neighborhood function (Equation 5.3), which depends on the topological distance from the winner cell to the cell in focus. Next, we continue the map training with a modified neighborhood function (Equation 5.4) that also depends on the distance (d) between the training case and the weight vector of the winner cell (refer to Subsection 5.2.2). This modified neighborhood function is designed in order to allow a voxel with a signature far from the weight vector of the corresponding winner cell (according to the distance metric) to have more influence on the map. Without this strategy, large homogeneous regions of the volume would tend to dominate the map, while important regions with fewer voxels would be badly represented (signatures far from their respective winner cells).

$$nf1(td) = \exp\left(-\frac{td^2}{3}\right) \tag{5.3}$$

$$nf2(td, d) = \min(d, 1) \times \exp\left(-\frac{td^2}{3}\right) \tag{5.4}$$

We use as topological distance the Euclidean distance between the integer 2D coordinates of two cells in the map grid. For spherical maps, the topological distance is the number of edges in the shortest path connecting two cells. At the end of the process, we have a Kohonen (or spherical) map where each cell has an associated weight vector that represents a class of voxels, being the most similar weight vector for all elements in that class.

5.3.2 Dimensional Reduction

Dimensional reduction is motivated by the need to provide a simplified space for the design of multi-dimensional transfer functions. When using Kohonen maps, two-dimensional map space coordinates, in the interval from zero to one, can be associated to cells according to their position in the 2D grid. Dimensional reduction can be performed by replacing each voxel signature by the coordinates of its respective winner cell, which has the most similar vectorial signature. However, this would cause unnecessary discretization. To avoid this, we create two multiquadric radial basis functions (multiquadric RBFs) [BUH 2003], for x and y map space coordinates, based on the weight vectors of all cells. For spherical maps we adopt x, y and z position coordinates ranging from -1 to 1, and use three RBFs to obtain the coordinates of voxel signatures. Thus, the RBFs support the final step in dimensional reduction of voxel signatures by producing, through interpolation, the proper x and y (and z) map coordinates for each nD voxel signature.

Dimensional reduction normally implies loss and distortion of information, but volumetric data usually have properties that reduce this problem. The voxels signatures are usually not uniformly distributed in their domain (they form clusters, which are well represented in the map), and elements of the voxel signatures are often not completely independent [TAK 2002]. Moreover, voxel signatures that are not present in the training set do not require space in the map, since SOMs are able to perform non-linear dimensional reduction. This ability and the well-defined topology and shape of SOMs have guided our choice for this particular dimensional reduction scheme.

5.3.3 Transfer Function Specification

After the dimensional reduction step, the continuous map space defined by the RBFs becomes the TF domain. The user can interactively define the mapping from map coordinates (which represent voxel signatures) to optical properties. We propose an interface for specification of color and opacity transfer functions that provides dual domain interaction [KNI 2002b] as well as visualizations of the transfer function and of the voxel signatures.

5.3.3.1 Interaction in the TF Domain

The visualization of voxel signatures in our interface is obtained by directly mapping up to three elements of the weight vectors of the map cells to the three color channels. The user decides which elements of the nD signatures must be mapped to each basic color. One element can be associated to more than one color channel and a color channel may have no elements mapped to it. This interface feature illustrates the distribution of voxel signatures on the map and can be used to build color TFs as described below. Figure 5.1 (a and b) shows the distribution of voxel signatures of the well-known engine data set. The same regions (clusters of signatures) can be found in both maps.

The transfer function (color and opacity) is represented as an RGBA image and dis-

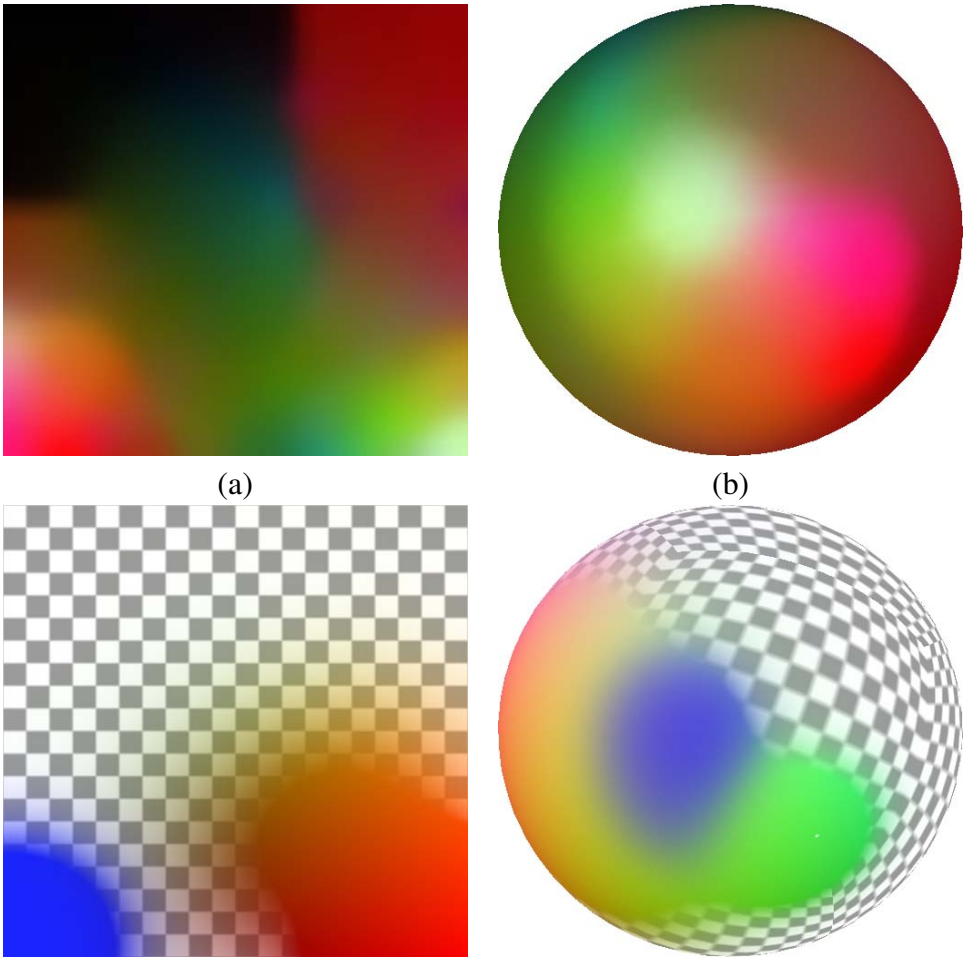


Figura 5.1: Maps of three-dimensional voxel signatures — a Kohonen map (a) and a spherical map (b). Scalar value is mapped to red, gradient magnitude to green and directional second derivative to blue. Transfer functions displayed on a Kohonen map (c) and on a spherical map (d).

played by blending it with a checkerboard pattern according to the equation $C_{out} = \alpha_{blending} \times C_{TF} + (1 - \alpha_{blending}) \times C_{checker}$, where colors are represented by C and opacity by α, and $\alpha_{blending} = \log(1 + s \times \alpha_{TF})/\log(1 + s)$. s is a constant greater than one (we empirically use 200) that allows TFs with small opacities to be clearly visualized. Figure 5.1 displays TFs on a Kohonen map (c) and on a spherical map (d) generated for visualizing the engine data set. The rendering of the data set using the TF in Figure 5.1c is shown in Figure 5.4a.

Transfer functions are composed by blending several 2D Gaussian opacity TFs, each one having an associated 2D color TF. We provide three types of color TFs that can be associated to a Gaussian opacity function: a constant color chosen from a colorpicker; map coordinates directly mapped to color channels; and elements of weight vectors of map cells mapped to color channels (for each map coordinate the weight vectors of the near cells are interpolated and mapped to colors). At each step a new Gaussian TF is specified and then blended with the current TF according to Equations 5.5 and 5.6, for opacity and color, respectively. The result becomes the current transfer function and the composition continues until the desired TF is reached. At start, the current TF has zero

opacity and RGB colors for all the map space.

$$\alpha_{new} = 1 - (1 - \alpha_{current}) \times (1 - \alpha_{Gaussian}) \tag{5.5}$$

$$C_{new} = \frac{\alpha_{current} \times C_{current} + \alpha_{Gaussian} \times C_{Gaussian}}{\alpha_{current} + \alpha_{Gaussian}} \tag{5.6}$$

In our interface, by clicking or dragging the mouse on the map representation, the user moves a circle whose center is the peak of a Gaussian function and whose radius is its standard deviation (σ). The Gaussian opacity TF — defined by Equation 5.7, where d is the distance to the center of the circle — is scaled by a constant k, between zero and one, which is linearly mapped to the circle color, with blue being zero and red being one. The parameters σ and k can be increased or decreased using the keyboard. In order to fully explore the spherical maps, they can be rotated by dragging the mouse using the right button.

$$\alpha_{Gaussian} = k \times \exp\left(-\frac{d^2}{2\sigma^2}\right) \tag{5.7}$$

The transfer function used for rendering is the composition of the current color and opacity TF and the Gaussian TF represented by the circle. This scheme provides interactive previewing of the effect of the composition while the user explores the map by moving the circle on it. When the desired effect is reached, the user can set the composition as the current TF using the space bar, and other Gaussian function can be further experimented. Our interface keeps track of all transfer functions defined during a session, and provides a tree representation of this evolution using static thumbnails of the volume rendered with the corresponding TF. This allows simple recovering of previous TFs by clicking on the thumbnails.

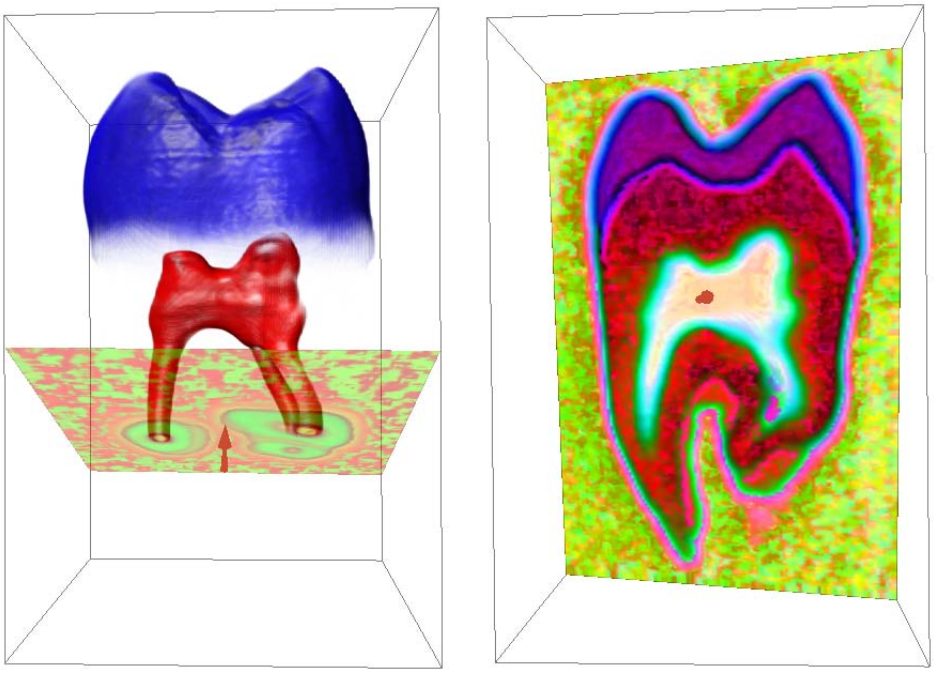
5.3.3.2 Interaction in the Spatial Domain

At any time the user can rotate and translate the volume and place a clipping plane to better explore inner structures. The volume slice defined by the clipping plane is textured by mapping to color channels the map coordinates of the voxels sampled by the slice. This causes an interesting coloring effect that helps in inspecting the volume. The slice is blended with the rendered volume using an opacity value controlled by the user, as shown in Figure 5.2. When Kohonen maps are employed, the x and y map space coordinates of the voxels are mapped to red and green. When using a spherical map, the x, y and z map space coordinates are mapped to RGB colors. The user can also click on the clipping plane to set the position of the Gaussian function peak to the map coordinates of the voxel pointed by the mouse cursor, emphasizing this region. By moving the mouse on the clipping plane, the user can see the position of the pointed voxel depicted as a white cross in the map graphical representation. This spatial domain interaction mapped to TF domain helps in understanding the relationship between both domains.

5.4 Implementation Aspects

We implemented map training and dimensional reduction as offline processes, but rendering and transfer function specification demand interactive rates, which are achieved through an intensive use of the GPU.

The map coordinates of the voxels are stored in a 3D RGB texture, which is sampled using view-aligned slices as proxy geometry. When using a Kohonen map, the transfer



(a) (b)

Figura 5.2: Visualizations of the tooth data set: a semi-transparent slice blended with the tooth image rendered using a transfer function specified in a 2D space built from a Kohonen map (a); and a fully opaque slice of the tooth colored according to voxel coordinates in a spherical map (b). The noisy regions can be clearly seen. The red arrow is the plane normal.

function is stored in a 2D RGBA texture which is accessed by using the R and G components (the x and y map coordinates) of the sampled 3D RGB texture. The blue component is used to identify background (zero) or non-background (one) voxels. Background voxels must receive zero opacity during rendering since they are not well represented in the map. Nevertheless, due to hardware interpolation, the blue component can assume values between zero and one. With this in mind, the opacity is actually modulated by a smooth step function ($f(x) = 0$, if $x <= a; (x - a)/(b - a)$, if $a < x < b; 1$, if $x >= b$, where $a < b$) of the blue component. When using a spherical map, the TF is stored in the GPU memory as an RGBA cube map and is accessed using the RGB values of the 3D texture, taken as vectors (the value of each color channel is first converted to the interval $[-1, 1]$. Background voxels have null vectors and the opacity is modulated by a smoothed step function of the $L2$-norm of the vectors. The blending of TFs and the evaluation of Gaussian opacity functions also run in GPU.

When sampling the three-dimensional texture for rendering, interpolation must be performed. The hardware can automatically interpolate the map coordinates stored in the 3D texture and generally this produces good results. However, in our approach, it is more correct to interpolate color and opacity associated to voxels (see [HAD 2003] for better understanding). In our implementation, we use the GPU to create another 3D texture, with the same size, containing the RGBA values that result from the evaluation of the transfer function for each voxel, and this texture is sampled for rendering. When the transfer function changes, this texture must be recomputed, but this effort does not compromise interactivity. We also calculate another 3D RGB texture to store the gradient field of the opacity. This is done in GPU by applying central differences on each voxel.

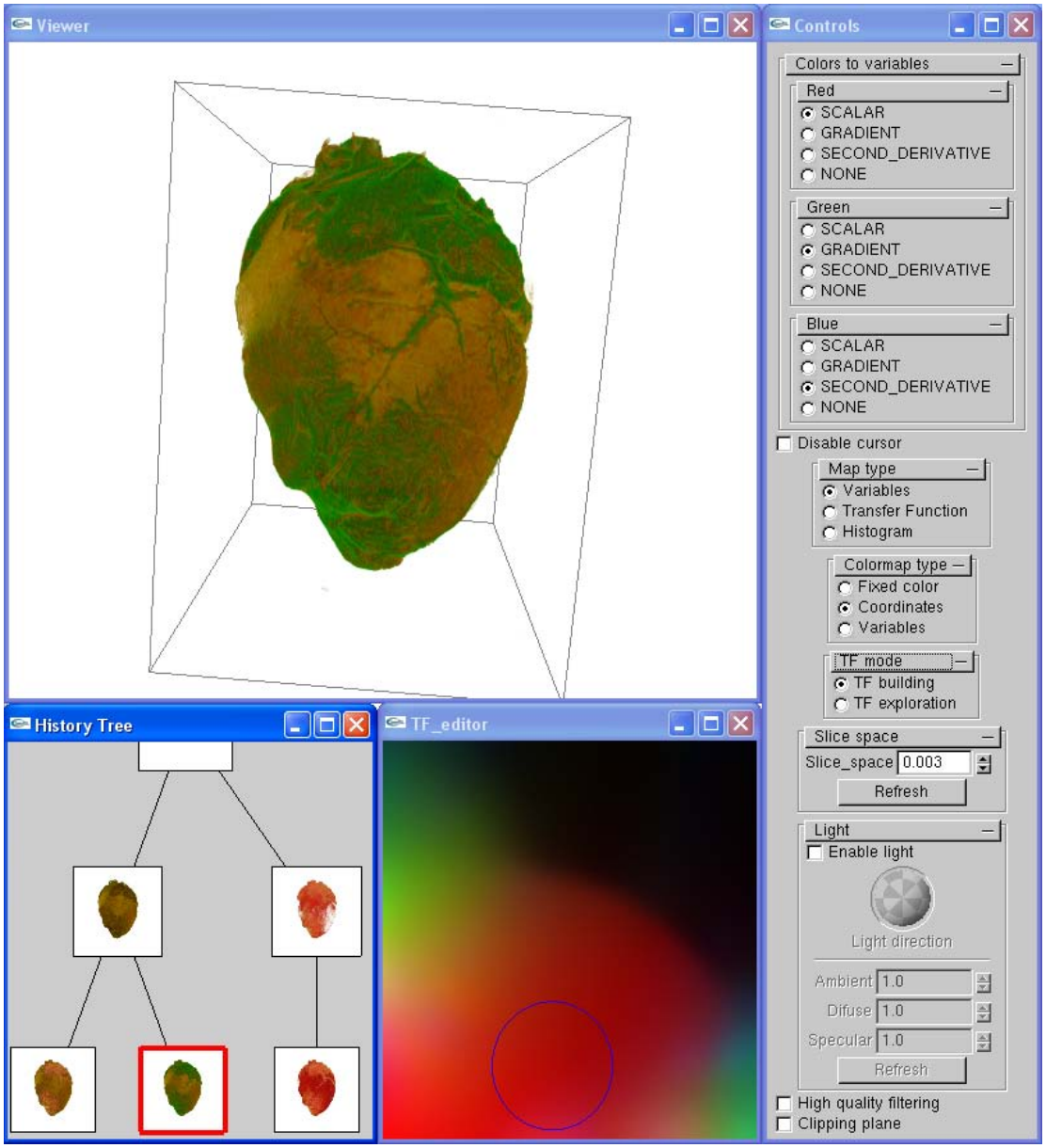


Figura 5.3: The interface of the tool we have implemented: the rendering window (top left); the control panel (right); the transfer function history tree (bottom-left corner); and the window for transfer function design (bottom middle).

The opposite vector of the gradient of the opacity is used as surface normal for shading. Since we are using complex signatures for each voxel, this scheme for evaluating normals is more accurate than sampling a 3D texture containing the precomputed normals of the scalar field. Additionally, the normals of the opacity field do not have ambiguity in their orientation (see [LUM 2004a]). In our implementation, we set hardware interpolation of map coordinates and precomputed normals as default options, but the user can select color and opacity interpolation and normals computed on the fly as high-quality rendering options.

As for the RBF design, we solve the systems of equations with the Lapack library [AND 99]. GLUT and the GLUI libraries are used for the interface, while the rendering is based on OpenGL and Cg, with the OpenGL's Framebuffer-Objects extension used in hardware-accelerated computing.

The developed interface is depicted by Figure 5.3, which shows a few windows: the rendering window, which displays an image of the data set rendered using the current transfer function; the control window, where a control panel allows setting several options, such as the type of colormap, light coefficients and what is shown in the map window; the history window, where the transfer function history tree formerly described can be explored through "pan" and "zoom"; and the transfer function design window, which displays either the map or the transfer function or a histogram of voxels and provides the interface to design transfer functions.

5.5 Results and Discussion

We tested our method using well-known data sets (see Figure 5.4), comprising scalar and multivariate volume data. Similar results were obtained using Kohonen and spherical maps. Most of the data sets were successfully visualized using voxels signatures based on scalar value, gradient magnitude and directional second derivative. For noisy scalar data, however, we achieved better results using statistical signatures, such as mean scalar value, standard deviation and cubic root of the third-order statistical moment, taken from a small subvolume centered at the voxel under focus. Figure 5.4 shows visualizations of test data sets obtained using different sets of voxel attributes as signatures. In all these renderings, we used the automatic generated color transfer functions (see Subsection 5.3.3.1). For the hurricane data set we used only the colormap which assigns voxel attributes to color channels, since the attributes carry a clear physical meaning: temperature was mapped to red, pressure to green and wind speed to blue. However, the tooth data set (Figure 5.2) was visualized using manually chosen colormaps and statistical signatures, achieving a very good separation of the pulp. Since self-organizing maps group similar voxel signatures, the automatic generated color transfer functions produce very good results because they assign different colors to different regions of the map, which correspond to voxels with considerably different attribute's values. Regarding our shading using normals computed on the fly, results can be seen in Figure 5.4 (c, d and e). It is worth to mention that for the multivariate data sets, such as the hurricane, we can not use meaningful precomputed normals.

The importance of each voxel attribute is defined by weights (see Subsection 5.2.2). We suggest associating smaller weights with higher-order voxel attributes. The visualizations presented in this chapter were produced using weights of 1.3, 1.0 and 0.7 for the statistical variables formerly mentioned, respectively, and the same weights for scalar and first and second derivative values, respectively. For the hurricane data set the weights

were 1.0 for wind speed and pressure, and 0.5 for temperature.

Due to the loss and distortion of information usually caused by dimensional reduction, our method can not provide accurate quantitative information about the volume data during the transfer function specification. However, our approach is well suitable for revealing qualitative aspects such as shape of structures and dissimilarity between regions. By moving the Gaussian opacity function on the map space (see Subsection 5.3.3.1), the user quickly obtains an overview of the main structures in the data volume. After, by carefully tuning the parameters of the Gaussian functions and combining them, meaningful visualizations can be built. Automatically generated color transfer functions are usually a good choice, which turns the design process less difficult (for example, the blood vessels in the Sheep Heart data set can be easily emphasized using our colormaps). The history tree, briefly described in Subsection 5.3.3, provides a powerful mechanism for exploring the transfer function domain, allowing not only "undo" and "redo" operations, but navigation in the whole history of TF modifications.

Regarding the building of the self-organizing maps for dimensional reduction, it involves setting of parameters and selection of voxel attributes to be mapped. In the following we provide guidelines for choosing appropriate parameters values and discuss their impact on processing time.

Self-organizing maps easily converge, becoming stable along the training, for small values of η (a constant between zero and one), and large number of training steps, but one generally has to assume a compromise between stability and training time. Fortunately, in our method, map stability is not a critical point. The lower bound of the number of training steps needed to guarantee stability depends on the map size, the neighborhood function, the value of η used in training (see Subsection 5.2.2), and the characteristics of the mapped data. Experimentally we found that good results can be achieved using Kohonen maps with more than 20×20 cells or spherical maps with more than seven subdivisions by icosahedron edge. For building the map, first we set η to 0.3 and perform training using the neighborhood function described in Equation 5.3 (see Subsection 5.3.1). Then we refine the map by training it using 0.5 as η value and Equation 5.4 as neighborhood function. The values for η were also chosen based on experiments.

To illustrate the voxels' distribution over the map space, our interface presents a histogram built from the coordinates obtained from dimensional reduction of the voxel signatures. The histogram depicts the occurrence of voxels over the map and is represented in logarithmic scale, being white the highest occurrence. We can also use this histogram to illustrate the importance of training the maps using two different types of neighborhood functions in order to better use the map space, as shown in Figure 5.5. Both maps were obtained through the same total number of training steps. The top map was built using only a Gaussian neighborhood function, while the bottom one was built in two stages, using two different types of neighborhood functions: the Gaussian one and our modified one. The data set is the Sheep Heart and the mapped signatures are scalar value, gradient and second derivative in gradient direction. In the first map (Figure 5.5a), the dark region, representing the noisy background which could not be properly removed using our simple approach, is larger than the same region in the second map (Figure 5.5c). The respective histograms (Figure 5.5, b and d) show that the background concentrates in a smaller part of the map when the two-stage training is employed. The same effect is obtained for any other large group of similar voxels.

The time needed to train the maps is quadratically related to the map size (number of cells) and is proportional to the size of the weight vectors. At each training step, the

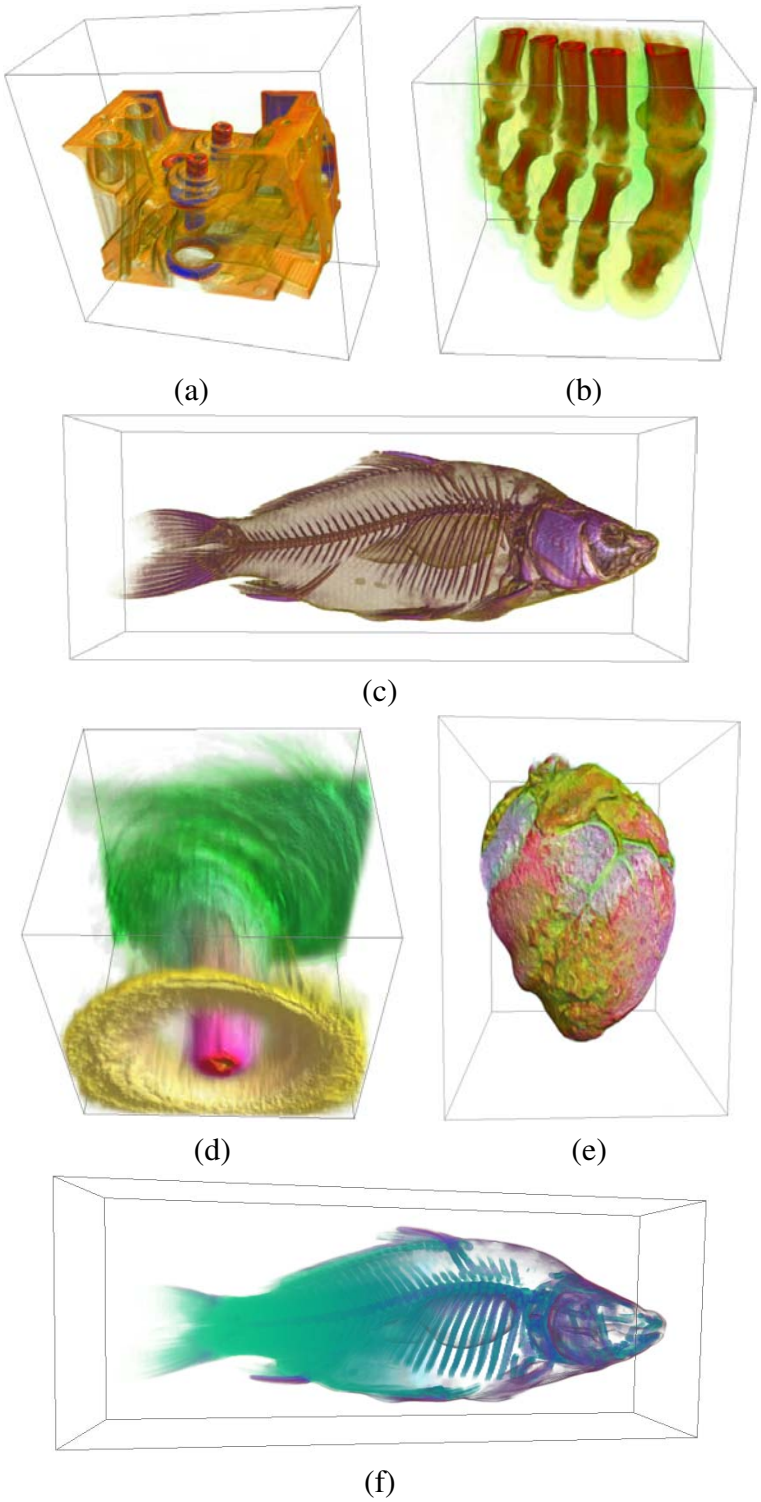


Figura 5.4: Visualizations obtained using Kohonen maps — (a) engine data set and (c) carp data set, both using scalar and derivative values (gradient magnitude and directional second derivative) as voxel signature; (b) foot data set, using statistical signatures (mean scalar value, standard deviation, cubic root of the third-order statistical moment); and (f) carp data set, using the normalized z coordinate of the voxels and same three statistical signatures, thus composing a four-dimensional voxel signature (z axis is horizontally represented) — and Spherical maps — (d) hurricane data set at the 24th time step, using wind speed, pressure and temperature as voxel signature; and (e) sheep heart data set, using the same statistical signatures as (b).

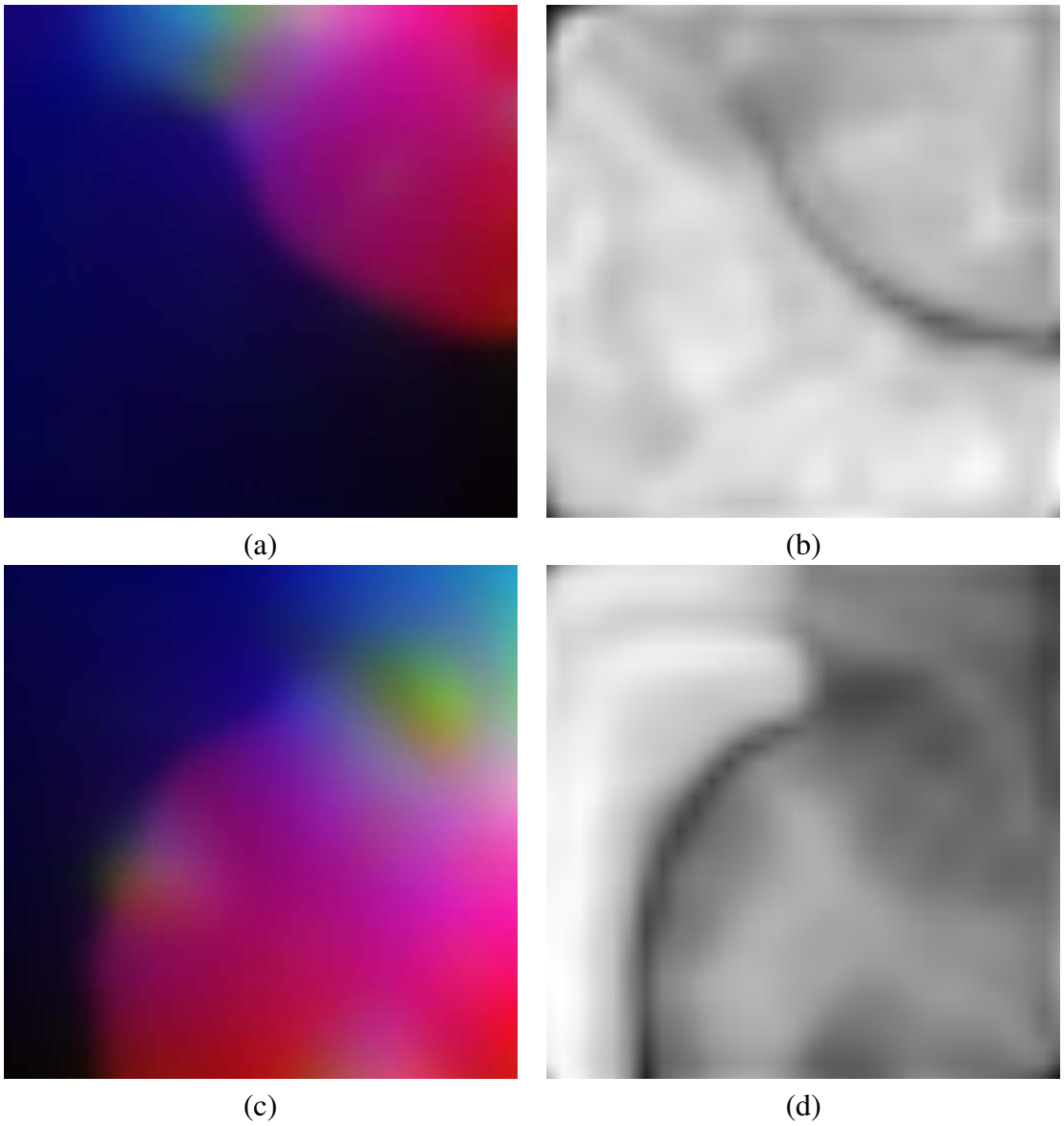


Figura 5.5: Maps of voxels' three-dimensional signatures of the Sheep Heart data set (a and c) — scalar (red), gradient (green) and directional second derivative (blue) — and respective histograms of voxels (b and d), in logarithmic scale, obtained using only the traditional neighborhood function (a and b) and using our modified neighborhood function in a second training stage (c and d).

Map type	Training steps/Time	
	Neighborhood function 1	Neighborhood function 2
2D map	3,000,000/57	4,500,000/87
Spherical map	3,000,000/151	5,250,000/264

Tabela 5.1: Number of training steps and training time (seconds) for both types of maps in the two stages of training, which use different neighborhood functions.

Data set	Size	Non-Background voxels	Time
Engine	$256^2 \times 128$	5297265	140
Tooth	$128^2 \times 256$	4193888	110
Foot	256^3	4890753	133
Carp	$256^2 \times 512$	32806605	845
Hurricane	$256^2 \times 128$	8388608	220
Sheep heart	256^3	16776721	375

Tabela 5.2: Size, number of non-background voxels and time (seconds) spent in the dimensional reduction step, using Kohonen maps and three voxel attributes, for the data sets presented in this chapter.

search for the winner cell has linear complexity, being dependent on the number of cells; the number of training steps needed is also directly related to the map size. The size of the weight vectors have linear influence on the time spent to search for the winner cell and update weight vectors (see Subsection 5.2.2). The time spent in the dimensional reduction step (see Subsection 5.3.2) is proportional to the number of non-background voxels (see Subsection 5.3.1), the size of the map, and the size of weight vectors. The RBF evaluation is performed for each non-background voxel and its time consumption is proportional to the number of cells and the size of weight vectors.

The results presented herein were obtained using Kohonen maps with 32×32 cells and spherical maps with ten subdivisions by icosahedron edge. Table 5.1 shows the number of performed training steps and the training time for both map types. Spherical maps usually require little more training; and the training steps are slower, mainly because the search for neighbors of the winner cell, for weight update, is more complex. Table 5.2 shows size, number of non-background voxels and dimensional reduction time, using Kohonen maps, of the visualized data sets. Using spherical maps the times were about thirteen percent greater. The time needed to extract high-order signatures from scalar volumes is short and is not considered because this is done only once for each data set. Our results were generated using an AMD Athlon 64 3700+, 2.21GHz CPU and 2GB of RAM.

5.6 Conclusions and Future Work

In this chapter we presented a new approach for the design of multi-dimensional transfer function that uses self-organizing maps to perform dimensional reduction of the voxel attributes. The strongest points of our technique are simplicity and flexibility. Our approach allows building multi-dimensional transfer functions through the exploration of a simplified (reduced) space where traditional interaction techniques can be employed. A simple and effective interface for transfer function design is provided, and the user can interact with the system in both spatial and TF domains.

Self-organizing maps have the ability of representing clusters of voxel signatures in a compact way, and this helps to understand the data distribution. All relevant voxel signatures are represented in the map and every region of the map has voxels mapped to it. Moreover, to explore two-dimensional maps is easier and faster than to navigate in a class hierarchy. The proposed dimensional reduction scheme requires a preprocessing step, but it has clear advantages in relation to volume segmentation techniques because it performs a non-discrete classification which can represent uncertainty. In addition, with simple interaction, the user can change the transfer function defined in the map space, interactively obtaining new visualizations. Our automatically generated color TFs are able to emphasize details that are hard to be visualized using other approaches. When compared to other visualization techniques based on transfer functions, ours has the advantage of having a powerful built-in classification scheme. The existence of classes and class properties is, however, kept abstract. As a drawback, our approach suffers from loss of information, caused by the dimensional reduction process.

As future work, we intend to assess the usability of our technique and to experiment our method on visualizing time-varying multivariate volume data. Coherence of time-varying transfer functions could be maintained by incremental training of the SOMs for each time step; and the evaluation time of the RBFs can be reduced using approximating RBFs. We also want to transport transfer functions designed in map space to the actual multi-dimensional space using the Gaussian multi-dimensional TFs proposed by Kniss [KNI 2003a]. Another promising future work is the semi-automatic search for important structures in the map. This search could be aided by an interface that would provide additional information about the spatial distribution of voxels. At last, it is important to emphasize that we have built a map of voxel signatures for each visualized data set. The success of performing dimensional reduction of a data set using a map built from another but similar data set needs to be assessed, and this is also intended as future work.

6 DISCUSSÃO E CONCLUSÕES

6.1 Especificação de Funções de Transferência Unidimensionais

Com foco na especificação de funções de transferência unidimensionais, o trabalho de Prauchner et al. [PRA 2005a, PRA 2005b] demonstrara uma estratégia promissora e inédita: a combinação de visualizações múltiplas simultâneas com geração automática interativa de FTs. Contudo, a proposta tinha um evidente potencial inexplorado. Esse foi o ponto de partida para o presente trabalho com FTs unidimensionais, que, além de implementar melhorias significativas na qualidade das visualizações, na interface e no desempenho da ferramenta criada por Prauchner et al., agregou novas funcionalidades que conferiram maiores versatilidade e praticidade à proposta anterior. As visualizações múltiplas interativas propiciam, através de FTs diversas geradas automaticamente para ênfase de bordas, compreensão imediata de vários aspectos do dado volumétrico investigado. A geração estocástica evolutiva permite refinamento de FTs, com total abstração destas, e também uma cobertura ampla do espaço de possibilidades em função da aleatoriedade inserida no processo. A ferramenta de modificação manual de funções de transferência com interação em dois domínios garante ao usuário, quando necessário, controle simples e preciso da especificação.

A principal contribuição desse trabalho é a adaptação e a combinação de técnicas de especificação de FTs unidimensionais conhecidas, escolhidas cuidadosamente para que a relação entre elas seja de complementação, não de redundância. Os argumentos em favor da relação de complementação são fortes, evidenciando o perfil e o potencial distintos de cada uma das técnicas reunidas. Corroboram tais argumentos os experimentos com os usuários, que rapidamente dominaram a ferramenta de especificação, empregando de maneira plena e eficaz o conjunto de técnicas disponível, demonstrando sua versatilidade. Os registros dos experimentos detalham os passos de interação efetuados pelos usuários e evidenciam que, intuitivamente, eles utilizaram todas as técnicas de especificação reunidas, seguindo um perfil de interação bem definido: obtenção de pontos de partida promissores utilizando visualizações múltiplas e geração automática de FTs; refinamento de FTs com geração estocástica evolutiva; e, após, refinamento manual de uma FT escolhida. Os resultados dos experimentos, contudo, apesar de encorajadores, foram obtidos de estudantes-pesquisadores com experiência em visualização volumétrica. Ainda é necessária uma avaliação do método com a presença de usuários finais de aplicações específicas com fins práticos.

Métodos de especificação de funções de transferência encontrados na literatura oferecem ao usuário poucas opções de interação e poucas operações. Alguns, sobretudo os automáticos, permitem pouco controle; outros exigem muito do usuário, fundamentando-se, total ou parcialmente, na especificação manual. O emprego de vários modos de interação,

interativos e intuitivos, e vários tipos de operação, além das visualizações simultâneas interativas já introduzidas por Prauchner et al. [PRA 2005a, PRA 2005b], constituem as vantagens da presente proposta sobre trabalhos anteriores.

6.2 Especificação de Funções de Transferência Multidimensionais

Funções de transferência têm a vantagem de representar decisões não-binárias acerca da visibilidade e aspecto de elementos de volume, mas são de difíceis especificação, representação e armazenamento em contextos multidimensionais. Alternativamente, algoritmos de segmentação podem ser aplicados a volumes multidimensionais de maneira natural, mas geralmente possuem tempo de processamento elevado — um obstáculo à interatividade na reorganização das classes de voxels, necessária para produzir novas visualizações. Além disso, tais algoritmos, excetuando os que realizam classificação probabilística, baseiam-se em decisões binárias sobre a pertinência dos voxels às classes definidas. Valores de propriedades ópticas distintos podem ser atribuídos aos voxels de acordo com a classe a que pertencem, mas isto caracteriza um indesejável compromisso entre excessiva discretização dos atributos ópticos dos voxels e excessivo número de classes para as quais se deve atribuir valores de atributos.

A principal contribuição do presente trabalho, ao abordar as funções de transferência multidimensionais, é um método geral de especificação que combina o potencial da segmentação de volumes com o mapeamento flexível permitido pelas FTs. Foi proposta uma técnica de especificação de FTs em um espaço simplificado, onde os voxels são agrupados por similaridade. O mapeamento de voxels para esse espaço fundamenta-se numa conhecida técnica não-linear de redução dimensional e de classificação: os mapas auto-organizáveis. A utilização de RBFs construídas sobre os mapas auto-organizáveis permite redução dimensional para um espaço contínuo em vez de discreto, processo que pode ser comparado à classificação não-discreta, como a obtida por algoritmos de segmentação probabilística. Entretanto, a existência de classes fica abstraída na abordagem aqui descrita. Como contribuição secundária, a técnica compreende geração automática de funções de transferência de cor significativas, funcionalidade encontrada, na literatura, apenas no trabalho de Fujishiro et al. [FUJ 2000, FUJ 99], que é restrito a FTs unidimensionais, bem mais simples de ser especificadas.

A abordagem proposta é de mais simples implementação e utilização que a classificação probabilística, que também sofre da dificuldade em definir um número apropriado de classes e suas características. Outras técnicas de classificação permitem apenas decisões binárias, limitação superada pela visualização volumétrica através de funções de transferência. Entretanto, a literatura não dispõe de técnicas de especificação de FTs multidimensionais que oferecem interação em espaço simplificado via redução dimensional não-linear. Técnicas de redução dimensional linear já foram propostas ([TAK 2002, BOR 2006]), mas estas tendem a sub-aproveitar o espaço simplificado construído para representação dos valores multidimensionais.

Julga-se de simples utilização o método de especificação de FTs desenvolvido, mas a sustentação dessa asserção depende de uma avaliação de usabilidade, pretendida como trabalho futuro.

6.3 Tendências na Especificação de Funções de Transferência

Duas tendências estão claras na pesquisa de estratégias para definição de funções de transferência: utilização de domínios multidimensionais e abstração da função no processo de especificação. As severas limitações, amplamente reconhecidas na literatura, das FTs unidimensionais as desqualificam como um meio geral de obtenção de visualizações acuradas para fins práticos, os quais freqüentemente dependem de dados com grande quantidade de ruído. Ainda, como constatação óbvia, FTs unidimensionais são insuficientes para produzir visualizações de dados não-escalares. As funções multidimensionais, por outro lado, têm seu potencial superior de classificação reconhecido pelos pesquisadores, embora estes corroborem a dificuldade de especificação em domínios amplos, o que tem levado à busca de mecanismos de abstração do processo.

Além do desenvolvimento de abordagens que ocultam a complexidade das FTs, promovendo especificação por interação em espaços derivados simplificados, tem-se buscado extrair, dos dados volumétricos, assinaturas de voxels com significado de mais alto nível, como a assinatura LH ([SER 2006c]) ou a probabilidade de um voxel pertencer a um determinado material representado no dado volumétrico (ver seção 3.3). A agregação de conhecimento de especialistas, proposta no trabalho de Salama et al. [REZ 2006], embora incipiente, pode tornar-se também uma tendência forte na pesquisa sobre especificação de FTs para fins práticos.

As funções de transferência unidimensionais, embora pobres em poder de classificação, são bastante úteis na compreensão rápida das características gerais dos dados volumétricos, portanto o trabalho aqui desenvolvido acerca desse tipo de FT visa especialmente à agilidade na especificação das funções e à apresentação imediata de múltiplas visualizações. O presente trabalho sobre FTs multidimensionais segue as tendências supracitadas, oferecendo um espaço de interação reduzido para a especificação das funções de transferência, que podem ter, como domínio, um número arbitrário de atributos de voxel. Adicionalmente, o espaço de interação proposto tem um significado de alto nível, traduzindo a similaridade entre voxels.

REFERÊNCIAS

[AND 99] ANDERSON, E. et al. **LAPACK Users' Guide**. 3rd ed. Philadelphia: SIAM, 1999.

[BAJ 97] BAJAJ, C. L.; PASCUCCI, V.; SCHIKORE, D. R. The contour spectrum. In: IEEE VISUALIZATION, VIS, 8., 1997, Phoenix. **Proceedings...** Los Alamitos: IEEE Computer Society, 1997. p. 167-173.

[BAV 2005] BAVOIL, L. et al. Vistrails: Enabling interactive multiple-view visualizations. In: IEEE VISUALIZATION, VIS, 16., 2005, Minneapolis. **Proceedings...** Los Alamitos: IEEE Computer Society, 2005. p. 135-142.

[BER 2005] BERGNER, S. et al. A practical approach to spectral volume rendering. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, v. 11, n. 2, p. 207-216, Mar./Apr. 2005.

[BOR 2006] BORDIGNON, A. L. et al. Exploratory visualization based on multidimensional transfer functions and star coordinates. In: BRASILIAN SYMPOSIUM ON COMPUTER GRAPHICS AND IMAGE PROCESSING, SIBGRAPI, 19., 2006, Manaus. **Proceedings...** Los Alamitos: IEEE Computer Society, 2006. p. 213-280.

[BRO 2001] BRODLIE, K.; WOOD, J. Recent advances in volume visualization. **Computer Graphics Forum**, Blackwell, v. 20, n. 2, p. 125-148, June 2001.

[BUH 2003] BUHMANN, M. D. **Radial Basis Functions**: Theory and Implementations. Cambridge: Cambridge University, 2003.

[CAB 94] CABRAL, B.; CAM, N.; FORAN, J. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In: IEEE SYMPOSIUM ON VOLUME VISUALIZATION, VOLVIS, 4., 1994, Tysons Corner. **Proceedings...** New York: ACM Press, 1994. p. 91-98.

[CUL 94] CULLIP, T. J.; NEUMANN. U. **Accelerating volume reconstruction with 3d texture hardware**. Chapel Hill: University of North Carolina, 1994. Technical report.

[DAN 92] DANSKIN, J.; HANRAHAN, P. Fast algorithms for volume ray tracing. In: WORKSHOP ON VOLUME VISUALIZATION, 2., 1992, Boston. **Proceedings...** New York: ACM Press, 1992. p. 91-98.

[DIE 2004] DIETRICH, C. A.; NEDEL, L. P.; OLABARRIAGA, S. D.; COMBA, J. L. D.; ZANCHET, D. J.; SILVA, A. M. M.; MONTERO, E. F. de S. Real-time interactive visualization and manipulation of the volumetric data using gpu-based methods. **Progress in Biomedical Optics and Imaging**, Bellingham, v. 5, n. 21, p. 181-192, 2004. Trabalho apresentado no International Society Optics and Imaging, 2004.

[ENG 2001] ENGEL, K.; KRAUS, K.; ERTL, T. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In: ACM SIGGRAPH/EUROGRAPHICS WORKSHOP ON GRAPHICS HARDWARE. EGGH, 5., 2001, Los Angeles. **Proceedings...** New York: ACM Press, 2001. p. 9-16.

[FAN 98] FANG, S; BIDDLECOME, T; TUCERYAN, M. Image-based transfer function design for data exploration in volume visualization. In: IEEE VISUALIZATION, VIS, 9., 1998, Research Triangle Park. **Proceedings...** Los Alamitos: IEEE Computer Society, 1998. p. 319-326.

[FUJ 99] FUJISHIRO, I.; AZUMA, T.; TAKESHIMA, Y. Automating transfer function design for comprehensible volume rendering based on 3d field topology analysis (case study). In: IEEE VISUALIZATION, VIS, 10., 1999, San Francisco. **Proceedings...** Los Alamitos: IEEE Computer Society, 1999. p. 467-470.

[FUJ 2000] FUJISHIRO, I. et al. Volume data mining using 3d field topology analysis. **IEEE Computer Graphics and Applications**, Los Alamitos, v. 20, n. 5, p. 46-51, Sept./Oct. 2000.

[HAD 2003] HADWIGER, M.; BERGER, C.; HAUSER, H. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In: IEEE VISUALIZATION, VIS, 14., 2003, Seattle. **Proceedings...** Los Alamitos: IEEE Computer Society, 2003. p. 40-47.

[HAN 2005] HANSEN, C. D.; JOHNSON, C. R. (Ed.). **Visualization Handbook**. Amsterdam: Elsevier Butterworth Heinemann, 2005.

[HE 96] HE, T. et al. Generation of transfer functions with stochastic search techniques. In: IEEE VISUALIZATION, VIS, 7., 1996, San Francisco. **Proceedings...** Los Alamitos: IEEE Computer Society, 1996. p. 227-236.

[HLA 2000] HLADUVKA, J.; KÖNIG, A.; GRÖLLER, E. Curvature-based transfer functions for direct volume rendering. In: SPRING CONFERENCE ON COMPUTER GRAPHICS, SCCG, 16., 2000. **Proceedings...** New York: ACM Press, 2000. p. 58-65.

[HUA 2003] HUANG, R.; MA, K. Rgvis: Region growing based techniques for volume visualization. In: PACIFIC CONFERENCE ON COMPUTER GRAPHICS AND APPLICATIONS, PG, 11., 2003, Canmore. **Proceedings...** Los Alamitos: IEEE Computer Society, 2003. p. 355-363.

[KAU 91] KAUFMAN, A. **Volume Visualization**. Los Alamitos: IEEE Computer Society, 1991.

[KIN 98] KINDLMANN, G.; DURKIN, J. W. Semi-automatic generation of transfer functions for direct volume rendering. In: IEEE SYMPOSIUM ON VOLUME VISUALIZATION, VOLVIS, 6., 1998, Research Triangle Park. **Proceedings...** New York: ACM Press, 1998. p. 79-86.

[KIN 2003] KINDLMANN, G. et al. Curvature-based transfer functions for direct volume rendering: Methods and applications. In: IEEE VISUALIZATION, VIS, 14., 2003, Seattle. **Proceedings...** Los Alamitos: IEEE Computer Society, 2003. p. 513-520.

[KOH 97] KOHONEN, T. **Self-Organizing Maps**. 2nd ed. Secaucus: Springer-Verlag, 1997.

[KON 2000] KÖNIG, A.; GRÖLLER, M. E. **Mastering transfer function specification by using volumepro technology**. Vienna: Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2000. (Technical Report TR-186-2-00-07).

[KNI 2001] KNISS, J. M.; KINDLMANN, G.; HANSEN, C. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In: IEEE VISUALIZATION, VIS, 12., 2001, San Diego. **Proceedings...** Los Alamitos: IEEE Computer Society, 2001. p. 255-262.

[KNI 2002a] KNISS, J. M. et al. Interactive translucent volume rendering and procedural modeling. In: IEEE VISUALIZATION, VIS, 13., 2002, Boston. **Proceedings...** Los Alamitos: IEEE Computer Society, 2002. p. 109-116.

[KNI 2002b] KNISS, J. M.; KINDLMANN, G.; HANSEN, C. Multidimensional transfer functions for interactive volume rendering. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, v. 8, n. 3, p. 270-285, July/Sept. 2002

[KNI 2003a] KNISS, J. M. et al. Gaussian transfer functions for multi-field volume visualization. In: IEEE VISUALIZATION, VIS, 14., 2003, Seattle. **Proceedings...** Los Alamitos: IEEE Computer Society, 2003. p. 497-504.

[KNI 2003b] KNISS, J. M. et al. A model for volume lighting and modeling. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, v. 9, n. 2, p.150-162, Apr./June 2003.

[KNI 2005] KNISS, J. M. et al. Statistically quantitative volume visualization. In: IEEE VISUALIZATION, VIS, 16., 2005, Minneapolis. **Proceedings...** Los Alamitos: IEEE Computer Society, 2005. p. 37-44.

[KRU 2003] KRUGER, J.; WESTERMANN, R. Acceleration techniques for gpu-based volume rendering. In: IEEE VISUALIZATION, VIS, 14., 2003, Seattle. **Proceedings...** Los Alamitos: IEEE Computer Society, 2003. p. 38-43.

[LAC 94] LACROUTE, P.; LEVOY, M. Fast volume rendering using a shear-warp factorization of the viewing transformation. **Computer Graphics**, New York, v. 28, p. 451-458, July 1994.

[LEV 88] LEVOY, M. Display of surfaces from volume data. **IEEE Computer Graphics and Applications**, Los Alamitos, v. 8, n. 3, p. 29-37, May/June 1988.

[LEV 90] LEVOY, M. Efficient ray tracing of volume data. **ACM Transactions on Graphics**, New York, v. 9, n. 3, p. 245-261, July 1990.

[LIC 98] LICHTENBELT, B.; CRANE, R.; NAQVI, S. **Introduction to volume rendering**. Upper Saddle River, NJ: Prentice Hall PTR, 1998.

[LOR 87] LORENSEN, W. E.; CLINE, H. E. Marching cubes: a high resolution 3d surface construction algorithm. **Computer Graphics**, New York, v. 21, n. 4, p. 163-169, July 1987. Trabalho apresentado na Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, 1987.

[LUM 2004a] LUM, E. B.; MA, K. Lighting transfer functions using gradient aligned sampling. In: IEEE VISUALIZATION, VIS, 15., 2004, Austin. **Proceedings...** Los Alamitos: IEEE Computer Society, 2004. p. 289-296.

[LUM 2004b] LUM, E. B.; WILSON, B.; MA, K. High-quality lighting and efficient pre-integration for volume rendering. In: JOINT EUROGRAPHICS-IEEE TVCG SYMPOSIUM ON VISUALIZATION, VISSYM, 6., 2004, Konstanz. **Proceedings...** Los Alamitos: IEEE Computer Society, 2004. p. 25-34.

[MA 98] MA, F. et al. Probabilistic segmentation of volume data for visualization using som-pnn classifier. In: IEEE SYMPOSIUM ON VOLUME VISUALIZATION, VOLVIS, 6., 1998, Research Triangle Park. **Proceedings...** New York: ACM Press, 1998. p. 71-78.

[MAR 97] MARKS, J. et al. Design galleries: a general approach to setting parameters for computer graphics and animation. **Computer Graphics**, New York, v. 31, n. 4, p. 389-400, July 1997. Trabalho apresentado na Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, 1997.

[MUR 2006] MURAKI, D. J. et al. A spectral analysis of function composition and its implications for sampling in direct volume visualization. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, v. 12, n. 5, p. 1353-1360, Sept./Oct. 2006.

[PEK 2001] PEKAR, V.; WIEMKER, R.; HEMPEL, D. Fast detection of meaningful isosurfaces for volume data visualization. In: IEEE VISUALIZATION, VIS, 12., 2001, San Diego. **Proceedings...** Los Alamitos: IEEE Computer Society, 2001. p. 223-230.

[PFI 2001] PFISTER, H. et al. The transfer function bake-off. **IEEE Computer Graphics and Applications**, Los Alamitos, v. 21, n. 3, p. 16-22, May/June 2001.

[PIN 2006] PINTO, F. de M.; FREITAS, C. M. D. S. Two-level interaction transfer fuction design combining boundary emphasis, manual specification and evolutive generation. In: BRASILIAN SYMPOSIUM ON COMPUTER

GRAPHICS AND IMAGE PROCESSING, SIBGRAPI, 19., 2006, Manaus. **Proceedings...** Los Alamitos: IEEE Computer Society, 2006. p. 281-288.

[PRA 2005a] PRAUCHNER, J. L.; FREITAS, C. M. D. S.; COMBA, J. L. D. Two-level interaction approach for transfer function specification. In: BRASILIAN SYMPOSIUM ON COMPUTER GRAPHICS AND IMAGE PROCESSING, SIBGRAPI, 18., 2005, Natal. **Proceedings...** Los Alamitos: IEEE Computer Society, 2005. p. 265-272.

[PRA 2005b] PRAUCHNER, J. L. **Especificação de Funções de Transferência para Visualização Volumétrica**. 2005. Dissertação (Mestrado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre.

[REZ 2006] REZK-SALAMA, C.; KELLER, M.; KOHLMANN, P. High-level user interfaces for transfer function design with semantics. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, v. 12, n. 5, p. 1021-1028, Sept./Oct. 2006.

[ROE 2003] ROETTGER, S. et al. Smart hardware-accelerated volume rendering. In: JOINT EUROGRAPHICS-IEEE TVCG SYMPOSIUM ON VISUALIZATION, VISSYM, 5., 2003, Grenoble. **Proceedings...** Los Alamitos: IEEE Computer Society, 2003. p. 231-238.

[SAN 2002] SANGOLE A.; KNOPF. G. K. Representing high-dimensional data sets as closed surfaces. **Information Visualization**, Palgrave Macmillan, v. 1, n. 2, p. 111-119, June 2002.

[SER 2006a] ŠEREDA, P.; BARTROLI, A. V.; GERRITSEN, F. A. Automating transfer function design for volume rendering using hierarchical clustering of material boundaries. In: JOINT EUROGRAPHICS-IEEE TVCG SYMPOSIUM ON VISUALIZATION, EUROVIS, 8., 2006, Lisboa. **Proceedings...** Los Alamitos: IEEE Computer Society, 2006. p. 243-250.

[SER 2006b] ŠEREDA, P.; BARTROLI, A. V.; GERRITSEN, F. A. Mirrored lh histograms for the visualization of material boundaries. In: VISION, MODELING AND VISUALIZATION, VMV, 11., 2006, Aachen. **Proceedings...** Erlangen: Aka GmbH, 2006. p. 237-244.

[SER 2006c] ŠEREDA, P. et al. Visualization of boundaries in volumetric data sets using LH histograms. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, v. 12, n. 2, p. 208-218, Mar./Apr. 2006.

[TAK 2002] TAKANASHI, I. et al. Ispace: Interactive volume data classification techniques using independent component analysis. In: PACIFIC CONFERENCE ON COMPUTER GRAPHICS AND APPLICATIONS, PG, 10., 2002, Beijing. **Proceedings...** Los Alamitos: IEEE Computer Society, 2002. p. 366-374.

[TEM 2001] TENGINAKAI, S.; LEE, J.; MACHIRAJU, R. Salient iso-surface detection with model-independent statistical signatures. In: IEEE VISUALIZATION,

VIS, 12., 2001, San Diego. **Proceedings...** Los Alamitos: IEEE Computer Society, 2001. p. 231-238.

[TZE 2003] TZENG, F.; LUM, E. B.; MA, K. A novel interface for higher-dimensional classification of volume data. In: IEEE VISUALIZATION, VIS, 14., 2003, Seattle. **Proceedings...** Los Alamitos: IEEE Computer Society, 2003. p. 66-73.

[TZE 2004] TZENG, F.; MA, K. A cluster-space visual interface for arbitrary dimensional classification of volume data. In: JOINT EUROGRAPHICS-IEEE TVCG SYMPOSIUM ON VISUALIZATION, VISSYM, 6., 2004, Konstanz. **Proceedings...** Los Alamitos: IEEE Computer Society, 2004. p. 25-34.

[TZE 2005] TZENG, F.; LUM, E. B.; MA, K. An intelligent system approach to higher-dimensional classification of volume data. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, v. 11, n. 3, p. 273-284, May/June 2005.

[WES 90] WESTOVER, L. Footprint evaluation for volume rendering. **Computer Graphics**, New York, v. 24, n. 4, p. 367-376, August 1990. Trabalho apresentado na Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, 1990.