

José Elias Yauri Vidalón

FERRAMENTAS INTERATIVAS DE AUXÍLIO A  
DIAGNÓSTICO POR NEURO-IMAGENS 3D

Campinas  
2012

Universidade Estadual de Campinas  
Faculdade de Engenharia Elétrica e de Computação

José Elias Yauri Vidalón

# FERRAMENTAS INTERATIVAS DE AUXÍLIO A DIAGNÓSTICO POR NEURO-IMAGENS 3D

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Orientadora: Profa. Dra. Wu Shin-Ting

Este exemplar corresponde à versão final da dissertação defendida pelo aluno José Elias Yauri Vidalón e orientado pela Profa. Dra. Wu Shin-Ting

---

Campinas  
2012

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Y28f

Yauri Vidalón, José Elías

Ferramentas interativas de auxílio a diagnóstico por neuroimagens / José Elías Yauri Vidalón. --Campinas, SP: [s.n.], 2012.

Orientador: Wu, Shin-Ting.

Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Sistemas de imagens em medicina. 2. Diagnóstico por imagens - Técnicas digitais. 3. Sistemas de indicação visual tridimensional. 4. Imagem tridimensional. 5. Algoritmos de computador. I. Wu, Shin-Ting, 1958-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Interactive tools for volumetric neuroimage based diagnosis

Palavras-chave em Inglês: Imaging systems in medicine, Diagnostic images - Digital Techniques, Three -dimensional visual display systems, Three-dimensional image, Computer Algorithms

Área de concentração: Engenharia de Computação

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Fernando Cendes, Roberto de Alencar Lotufo

Data da defesa: 29-08-2012

Programa de Pós Graduação: Engenharia Elétrica

## COMISSÃO JULGADORA - TESE DE MESTRADO

**Candidato:** José Elias Yauri Vidalón

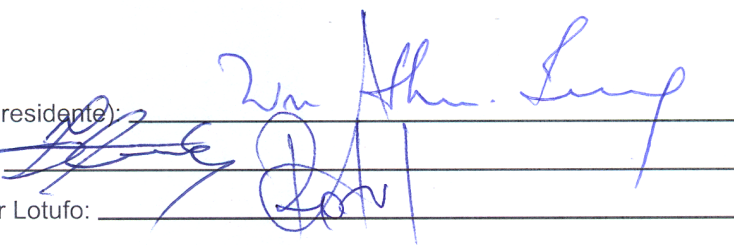
**Data da Defesa:** 29 de agosto de 2012

**Título da Tese:** "Ferramentas Interativas de Auxílio a Diagnóstico por Neuro-Imagens 3D"

Profa. Dra. Wu Shin-Ting (Presidente):

Prof. Dr. Fernando Cendes:

Prof. Dr. Roberto de Alencar Lotufo:



The image shows three horizontal lines representing signature lines. The top line has a signature that appears to be 'Wu Shin-Ting'. The middle line has a signature that appears to be 'Fernando Cendes'. The bottom line has a signature that appears to be 'Roberto de Alencar Lotufo'. The signatures are written in blue ink.

A DEUS, INVISÍVEL, ÚNICO E VER-  
DADEIRO.  
A FÉLIX E RUTH, MEUS PAIS.  
A MARIBEL, MEU AMOR.

# Agradecimentos

A ordem não é importante, apenas a presença realmente importa.

Eu agradeço,

a Deus por me dar forças para seguir sempre em frente;

à Profa. Dra. Wu Shin-Ting pela sua orientação e me encaminhar durante todos estes anos de estudo de mestrado. Suas correções e opiniões contribuíram para o sucesso deste trabalho;

ao Prof. Dr. Yuzo Iano pela oportunidade de chegar até a UNICAMP;

ao Prof. Dr. Fernando Cendes por ter aceitado a participar da banca de defesa e ao Prof. Dr. Roberto de Alencar Lotufo por ter aceitado ser membro da banca de defesa e de qualificação;

aos médicos Ana Carolina Coan, Clarissa Yasuda, Marcia Morita e Felipe Glehn do Laboratório de Neuro-Imagens da UNICAMP pelas sugestões e paciência nos teste de usabilidade e para Fádua Hedjazi e Heliantho Lima do Centro Médico de Campinas pelas sugestões e observações no software.

aos meus amigos da FEEC: Odair, Lionis, Matias, Wallace e Augusto, pela ajuda, convivência, amizade e troca de experiências e para Eloá Nava da FT, minha amiga especial;

à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro concedido durante todo o período de mestrado;

aos meus pais, irmãos e todos meus familiares que me apoiaram constantemente;

de maneira especial para Maribel, quem sempre soube manter seu amor por mim.

Bem-aventurado o homem que acha sabedoria, e o  
homem que adquire conhecimento;  
Porque é melhor a sua mercadoria do que artigos  
de prata, e maior o seu lucro que o ouro mais fino.  
Mais preciosa é do que os rubis, e tudo o que mais  
possas desejar não se pode comparar a ela.  
Vida longa de dias está na sua mão direita; e na  
esquerda, riquezas e honra.  
Os seus caminhos são caminhos de delícias, e todas  
as suas veredas de paz.  
É árvore de vida para os que dela tomam, e são  
bem-aventurados todos os que a retêm.

Provérbios 3:13-18

# Resumo

Por apresentar alta resolução espacial e espectral, é crescente o uso de imagens de ressonância magnética tanto no estudo dos órgãos humanos como também no diagnóstico das anormalidades estruturais e funcionais e no planejamento e treinamento cirúrgico. Junto com a rápida evolução dos algoritmos de processamento de imagens médicas, surgiram na última década aplicativos de diagnósticos assistidos por computador especializados em mamografia, angiografia e imagens da região torácica. A complexidade estrutural do cérebro e as diferenças anatômicas individuais do crânio constituem, no entanto, ainda desafios ao desenvolvimento de um sistema de diagnóstico especializado em neuro-imagens. A intervenção de especialistas é muitas vezes imprescindível na identificação e na interpretação dos achados radiológicos. Nesta dissertação, propomos o uso de três técnicas para auxiliar os especialistas da área médica na busca por achados radiológicos sutis de forma interativa. São apresentados dois objetos de interação, lente móvel e sonda volumétrica, que permitem atualizar continuamente os dados em foco enquanto são manipulados. Com isso, é possível investigar regiões cerebrais de interesse preservando o seu contexto. E, a fim de facilitar a percepção visual das variações funcionais ou estruturais sutis, propomos utilizar um editor de funções de transferência 1D para realçar ou aumentar o contraste entre os *voxels* adjacentes. As ferramentas foram avaliadas por um grupo de especialistas em neuro-imagens do Laboratório de Neuro-imagens da Faculdade de Ciências Médicas da Unicamp.

Palavras-chave: Neuro-imagens de ressonância magnética. Renderização volumétrica. Lente. Sonda. Técnica foco+contexto. Manipulação direta 3D. Diagnóstico assistido por computador.



# Abstract

Because of its high spatial and spectral resolution, it is increasing the use of magnetic resonance images both in the study of human organs as well as in the diagnosis of structural and functional abnormalities and in the surgery planning and training. Along with the rapid evolution of medical image processing algorithms, computer-aided diagnostics systems specialized in mammography, angiography, and computed tomography and magnetic resonance of the thorax have emerged in the last decade. The structural complexity of the brain and individual anatomical shape of skulls are, however, challenges in developing a diagnostic system specializing in neuro-imaging. Expert interventions are still essential both in the identification and in the interpretation of radiological findings. In this dissertation, we propose the use of three techniques to aid the medical experts in the search of subtle findings in an interactive way. We present two widgets, movable lens and volumetric probe, that allow one to update continuously the volume data in focus while are manipulated. In this way, it is possible to investigate brain regions of interest preserving its context. And, in order to facilitate the visual perception of the subtle functional or structural changes, we propose to use an editor of 1D transfer function to enhance or to increase the contrast between adjacent voxels. The tools were assessed by the neuro-imaging experts of the Laboratory of Neuro-Images of the Faculty of Medical Sciences of Unicamp.

Key-words: Magnetic resonance neuroimages. Volume rendering. Lens. Probe. Focus+context technique. 3D direct manipulation. Computer-aided diagnosis.

# Lista de Figuras

2.1	Imagens de CT: (a) Corte axial. (b) Corte Sagital. [fonte dos dados: (Roettger 2006)] . . . . .	7
2.2	Imagens de ressonância magnética: (a) Corte axial. (b) Corte sagital. . . . .	8
2.3	Pipeline gráfico. . . . .	12
2.4	(a) Partição do domínio da integração em vários intervalos. (b) Aproximação pela soma de Riemann. . . . .	14
2.5	Imagem colorida: (a) correspondência independente entre os valores escalares das amostras e os quatro canais R, G, B e A; (b) visualização dos dados volumétricos. [fonte dos dados: (Roettger 2006)] . . . . .	16
2.6	Imagem em escala de cinza: (a) correspondência independente entre os valores escalares e os canais I e A; (b) visualização dos dados volumétricos. [fonte dos dados: (Roettger 2006)] . . . . .	16
2.7	Função de transferência multi-dimensional baseado em gradientes e hessianas: (a) Imagem de CT. (b) Imagem de ressonância magnética. [fonte dos dados: (Roettger 2006)] . . . . .	17
2.8	<i>Ray casting</i> : as amostras ao longo de cada raio são representadas pelos discos cinzentos. . . . .	18
2.9	Renderização direta de volume: (a) Cabeça MRI. (b) Bonsai CT. (c) Motor CT. [fonte dos dados (b) e (c): (Roettger 2006)] . . . . .	21
2.10	Interface do VMTK. Fonte: (Wu, Yauri Vidalon & de Souza Watanabe 2011a)	23
2.11	Arquitetura do VMTK. Fonte: (Wu et al. 2011a). . . . .	24
3.1	Objetos de interação: (a) disco circular; (b) esfera. . . . .	28
3.2	Visualização foco+contexto: (a) e (c) realce global do volume utilizando uma FT; (b) realce localizado utilizando a sonda e FT do foco; e (d) realce localizado utilizando a lente e FT do foco. . . . .	31
3.3	Ilustração da lente móvel. . . . .	33
3.4	Projeção perspectiva em relação à lente móvel. . . . .	34

3.5	Lente móvel sobre corte sagital de uma imagem de RM 3D: (a) $FA = 1.0$ (sem aumento), (b) $FA = 2.5$ , e (c) $FA = 3.5$ . . . . .	35
3.6	Seleção da região focal e contextual utilizando a sonda volumétrica. . . . .	39
3.7	Sonda volumétrica em relação a um volume de dados: (a) livre no espaço 3D, região selecionada (b) com a geometria de sonda visível, e (c) com a geometria desabilitada. . . . .	40
3.8	Disposição espacial da sonda volumétrica em relação a um volume de dados: (a) fora do volume (b) parcialmente fora do volume, e (c) dentro do volume. . . . .	42
4.1	Arquitetura de sistema destacando as novas técnicas integradas ao VMTK. . . . .	46
4.2	Nova interface com 4 janelas sincronizadas no domínio espacial. . . . .	46
4.3	Controle dos objetos de interação: (a) ícone correspondente na barra de ferramenta e (b) janela contendo caixas de controle. . . . .	47
4.4	Editor interativo de função de transferência: (a) ícone correspondente na barra de ferramenta e (b) interface. . . . .	48
4.5	Visualização com: (a) uma função de transferência monotônica e (b) uma função de transferência não-monotônica. . . . .	49
4.6	Visualização de uma lesão sutil com: (a) uma função de transferência monotônica e (b) uma função de transferência não-monotônica. . . . .	50
4.7	Ampliação de uma lesão sutil (a) com uma lente com $FA = 2.6$ : (b) com composição de cores e (c) sem composição de cores. . . . .	51
4.8	Comparação entre (a) imagem em Matlab, (b) lente com composição, e (c) lente sem composição. . . . .	51
4.9	Realce das amígdalas: vista axial (a) amígdala esquerda (b) e amígdala direita(c). . . . .	52
4.10	Comparação em FPS entre algoritmo clássico de <i>ray casting</i> (Raycast.) e o nosso algoritmo de dois passos (FBO+Raycast). . . . .	53
4.11	Comparação em FPS entre o algoritmo de dois passos (FBO+Raycast.) e com sobrecarga da integração da lente móvel (Lente Móvel). . . . .	54
4.12	Comparação entre as taxas de quadros em FPS entre volume de dados com e sem a sonda volumétrica. . . . .	56
4.13	Paciente 1: DCF IIA, localizada no quadrante posterior esquerdo. . . . .	57
4.14	Paciente 2: DCF IIA localizada na parte frontal direita. . . . .	57
4.15	Paciente 3: DCF IIB localizada na parte frontal esquerda. . . . .	58
4.16	Paciente 4: DCF IIB localizada no quadrante posterior direito. . . . .	58
4.17	Paciente 5: DCF IIA localizada na parte frontal direita. . . . .	59
4.18	Paciente 6: DCF IIA localizada na parte frontal direita. . . . .	59
4.19	Concordâncias em identificação das lesões nas neuro-imagens. . . . .	59
4.20	Segmentação manual do hipocampo: (a) vista axial, (b) sagital e (c) coronal. . . . .	60
5.1	Artefatos ficam mais visíveis em regiões com maior fator de ampliação. . . . .	62

# Lista de Tabelas

4.1	Escalabilidade da lente móvel em FPS. . . . .	55
4.2	Escalabilidade da sonda volumétrica. . . . .	55

# Lista de Algoritmos

2.1	<i>Ray casting</i> baseado em GPU. . . . .	20
3.1	Lente móvel . . . . .	37
3.2	Sonda Volumétrica . . . . .	41

# Lista de Acrônimos e Notação

API	Application Program Interface (interface de programa de aplicação)
CAD	Computer-Aided Diagnosis (diagnóstico assistido por computador)
Cg	C for Graphics (C para gráficos)
CPU	Central Process Unit (unidade central de processamento)
CT	Computed Tomography (tomografia computadorizada)
DICOM	Digital Imaging and Communications in Medicine (comunicação de imagens digitais em medicina)
DVR	Direct Volume Rendering (renderização direta de volume)
FBO	Frame Buffer Object (objeto frame buffer)
FPS	Frame Per Seconds (quadros por segundo)
GLSL	OpenGL Shading Language (linguagem de <i>shading</i> de OpenGL)
HLSL	High-Level Shader Language (linguagem de <i>shading</i> de alto nível)
GPU	Graphic Process Unit (unidade de processamento gráfico)
MIP	Maximum Intensity Projection (projeção de intensidade máxima)
MRI	Magnetic Resonance Imaging (imagem por Ressonância Magnética)
NDC	Normalized Device Coordinate (coordenada de dispositivo normalizado)
NPR	Non-Photorealistic Rendering (renderização não foto-realística)
OBJC	Object Coordinate (coordenadas do objeto)
PET	Positron Emission Tomography (tomografia por emissão de pósitrons)
Pixel	Picture element (elemento da imagem)
QDV	Query-Driven Visualization (visualização orientada a consulta)
IR	Conjunto dos números reais
RGB	color space (Red, Green, Blue), espaço de cor (Vermelho, Verde, Azul)
RGBA	color space (Red, Green, Blue, Opacity), espaço de cor (Vermelho, Verde, Azul, Opacidade)
ROP	Raster Operation Processor (processador de operações <i>raster</i> )
SC	Screen Coordinate (coordenada da janela)
SIMD	Single Instruction, Multiple Data (mesma instrução, dados múltiplos)

SPECT	Single-Photon Emission Computed Tomography (tomografia computadorizada por emissão de fóton único)
TMU	Texture Mapping Unit (unidade de mapeamento de textura)
VMTK	Visual Manipulation Tool for 3D MR Neuro-Images (ferramentas de manipulação visual para neuro-imagens de ressonância magnética 3D)
Voxel	Volume element (elemento de volume)

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos da pesquisa . . . . .	2
1.1.1	Objetivo geral . . . . .	2
1.1.2	Objetivos específicos . . . . .	2
1.2	Contribuições . . . . .	2
1.3	Trabalhos relacionados . . . . .	3
1.4	Estrutura da dissertação . . . . .	5
<b>2</b>	<b>Preliminares</b>	<b>6</b>
2.1	Neuro-Imagens . . . . .	6
2.1.1	Imagens Estruturais . . . . .	6
2.1.2	Imagens Funcionais . . . . .	9
2.2	Diagnóstico Auxiliado por Computador Baseado em Imagens . . . . .	10
2.3	A GPU . . . . .	11
2.4	Renderização Direta de Volume . . . . .	13
2.4.1	Modelo Óptico . . . . .	13
2.4.2	Função de Transferência . . . . .	15
2.4.3	Ray casting . . . . .	18
2.4.4	Composição . . . . .	19
	Frente-para-trás . . . . .	19
	Trás-para-frente . . . . .	19
2.4.5	Uma Implementação em GPUs . . . . .	19
2.5	Visualização de Foco+Contexto . . . . .	21
2.6	Consultas Dinâmicas . . . . .	22
2.7	O VMTK . . . . .	22
2.8	Conclusões . . . . .	25
<b>3</b>	<b>Interação Exploratória</b>	<b>26</b>
3.1	Problemas a Resolver . . . . .	27



3.1.1	Especificação de um Objeto de Interação . . . . .	27
3.1.2	Posicionamento Espacial dos Objetos de Interação . . . . .	27
3.1.3	Realce Interativo . . . . .	29
3.1.4	Interatividade . . . . .	30
3.2	Editor de Funções de Transferência 1D . . . . .	32
3.3	Lente Móvel . . . . .	32
3.3.1	Geometria . . . . .	33
3.3.2	Modelo . . . . .	33
3.3.3	Controle dos Movimentos . . . . .	35
3.3.4	Integração com Dados Volumétricos . . . . .	35
3.3.5	Implementação em GPU . . . . .	36
3.4	Sonda Volumétrica . . . . .	38
3.4.1	Geometria . . . . .	38
3.4.2	Modelo . . . . .	38
3.4.3	Controle dos Movimentos . . . . .	39
3.4.4	Integração com Dados Volumétricos . . . . .	40
3.4.5	Implementação em GPU . . . . .	40
3.5	Conclusões . . . . .	42
<b>4</b>	<b>Resultados</b> . . . . .	<b>44</b>
4.1	Ambiente de Desenvolvimento . . . . .	44
4.1.1	Infra-estrutura . . . . .	44
4.1.2	Integração das Ferramentas no VMTK . . . . .	45
	Interface . . . . .	45
	Processamento Computacional . . . . .	48
4.2	Testes de qualidade de imagens . . . . .	49
4.2.1	Função de Transferência . . . . .	49
4.2.2	Lente Móvel . . . . .	50
4.2.3	Sonda Volumétrica . . . . .	51
4.3	Testes de Interatividade . . . . .	52
4.3.1	Renderização em Dois Passos . . . . .	53
4.3.2	Interatividade da Lente . . . . .	54
4.3.3	Interatividade da Sonda Volumétrica . . . . .	55
4.4	Testes de Usabilidade . . . . .	56
4.4.1	Localização de Lesões Cerebrais . . . . .	58
4.4.2	Segmentação do Hipocampo . . . . .	58
4.4.3	Opiniões . . . . .	58
4.5	Conclusões . . . . .	60

<b>5</b>	<b>Análise e Discussão dos Resultados</b>	<b>61</b>
5.1	Qualidade Visual . . . . .	61
5.2	Interatividade . . . . .	63
5.3	Usabilidade . . . . .	64
5.4	Conclusões . . . . .	65
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>66</b>
	<b>Bibliografia</b>	<b>68</b>
<b>A</b>	<b>Apêndice</b>	<b>76</b>

# Introdução

Massivos dados volumétricos, ou dados 3D, são comuns hoje em dia nas áreas da ciência, engenharia, geologia, medicina e biologia. Na medicina neurológica, constituem fontes destes dados as imagens resultantes de exames clínicos com aparelhos de tomografia computadorizada (CT), ressonância magnética (MRI), tomografia por emissão de pósitrons (PET) etc. Estas neuro-imagens são de grande valia para investigação de anormalidades cerebrais, fornecendo informações anatômicas, metabólicas e mesmo funcionais. No entanto, na sua versão original, são de difícil compreensão e manuseio para os médicos e neuro-cientistas. Elas são, portanto, processadas, interpretadas e transformadas em imagens exibíveis em uma tela de computador (Watt 2000, Shirley, Ashikhmin, Gleicher, Marschner, Reinhard, Sung, Thompson & Willemsen 2005, Wright 2007).

Quando se diagnostica uma anormalidade neurológica em estágio inicial, aumenta-se a chance de evitar que esta lesão se agrave, pode-se retardar os sintomas da doença, ou até conseguir uma recuperação plena do paciente. As neuro-imagens podem constituir uma poderosa ferramenta para diagnósticos precoces, se os médicos puderem explorar de forma minuciosa e intuitiva as informações contidas nelas. No entanto, apesar do rápido desenvolvimento tecnológico, com dispositivos de alta precisão e reduzido tempo de aquisição, e com novos e aprimorados algoritmos de visualização, a visualização interativa que permite a médicos e cirurgiões uma visão exploratória da estrutura interna de um paciente ainda se encontra em seu estágio inicial, sendo um campo de pesquisa desafiador.

Dentre as ferramentas que permitem explorar minuciosamente os tecidos cerebrais a partir das neuro-imagens, distinguem-se as técnicas de visualização no modo foco+contexto (Card, Mackinlay & Shneiderman 1999) e os métodos de cortes de volume (Weiskopf, Engel & Ertl 2003). Os primeiros permitem uma visão detalhada da região de interesse (*foco*) sem perda do referencial (*contexto*), e podem ser obtidos por meio de algoritmos de renderização multi-modal (Hauser, Mroz, Italo Bischi & Groller 2001, Zhou, Hinz & Tönnies 2002), funções de transferência (Levoy 1988, Hauser et al. 2001, Engel, Hadwiger, Kniss, Rezk-Salama & Weiskopf 2006) e lentes mágicas de volume (LaMar, Hamann & Joy 2001, Wang, Zhao, Mueller & Kaufman 2005), enquanto as técnicas de cortes em dados volumétricos têm sido

desenvolvidos para remover seletivamente partes do volume de dados para revelar regiões ocultas do volume. Entre os tipos de cortes encontrados na literatura, temos cortes alinhados aos eixos principais (Roden & Brett 2000), cortes em ângulos oblíquos (Rezk-Salama, Engel & Higuera 2002), cortes baseadas em geometrias arbitrárias (Weiskopf et al. 2003) e cortes curvilíneos paralelos ao couro cabeludo da cabeça (Wu, Yasuda & Cendes 2012)).

Estes resultados ainda não são inteiramente satisfatórios para os interesses médicos. Um dos principais empecilhos está na dificuldade de perceber a relação espacial entre os tecidos e manipulá-los apropriadamente. Os médicos e neurocientistas esperam poder experimentar diversas hipóteses manuseando de forma compreensível os dados disponíveis, enfatizando ou mitigando-os, a fim de identificarem vestígios de uma lesão através das imagens exibidas na tela de um monitor. Consideramos neste trabalho a *visualização orientada a consulta dinâmica*, na qual os dados buscados são continuamente atualizados conforme filtros habilitados enquanto o usuário passa o *cursor* sobre as regiões de interesse.

## 1.1 Objetivos da pesquisa

### 1.1.1 Objetivo geral

O objetivo deste trabalho é desenvolver técnicas que auxiliam os médicos no processo de diagnóstico médico de lesões cerebrais sutis através da exploração de neuro-imagens de ressonância magnética 3D (MRI) sem recorrer a uma segmentação prévia.

### 1.1.2 Objetivos específicos

Através deste trabalho, pretendemos desenvolver um protótipo de exploração de volumes de dados cerebrais com as seguintes funcionalidades:

- ferramentas de consulta interativa para agilizar a busca de anomalias cerebrais;
- editor de função de transferência não-monotônica entre os valores escalares das neuro-imagens e níveis de cinza, com o objetivo de enfatizar variações sutis nos dados em tempo interativo; e
- editor interativo de regiões focais com o objetivo de estabelecer funções de transferência individualizadas para regiões de interesse.

## 1.2 Contribuições

Sob o ponto de interface com usuários, desenvolvemos três ferramentas para auxiliar os neurocientistas na busca de achados radiológicos em neuro-imagens:

- dois objetos de interação baseados no paradigma foco+contexto: a *sonda volumétrica* para selecionar regiões específicas do volume em 3D e a *lente móvel* para ampliar uma área de interesse do volume;
- o *editor interativo de funções de transferência 1D* para realçar de forma diferenciada os contrastes dos tecidos selecionados e não selecionados do volume.

O diferencial das nossas ferramentas é que o usuário pode manipular o volume de dados e os objetos exibidos como se tivesse na sua mão utilizando o *mouse*. O *cursor* pode ser deslocado tanto no plano da tela (2D) como entrando e saindo deste plano (3D). Isto permite que a *sonda* seja posicionada em qualquer ponto do espaço 3D e que a *lente* seja deslizada sobre a superfície do volume em investigação. Além disso, pode-se realçar tais regiões de interesse através do *editor de função de transferência* alterando as propriedades ópticas dos *voxels* de forma não monotônica.

Sob o ponto de vista de desenvolvimento de *software*, propomos uma arquitetura unificada de renderização em dois passos para os dados volumétricos e os objetos de interação. No primeiro nível, exploramos a capacidade de renderização *off-screen* para computar eficientemente os mapas de profundidade dos dados. No segundo nível, exploramos a funcionalidade de mistura de cores em unidades de processamento gráfico, GPUs<sup>1</sup>, para sobrepor um objeto de interação com os dados volumétricos, criando o efeito visual do objeto de interação estar deslocando entre estes dados.

Um dos problemas básicos que tivemos que resolver foi conceber uma forma intuitiva de posicionar os objetos de interação do usuário (sonda e lente) no espaço 3D utilizando o *mouse* convencional. Inspirados nos trabalhos de Mesquita & Wu (2001) e Wu et al. (2011a), implementamos um *cursor 3D* de movimento livre. Graficamente, o *cursor* é representado por seis pequenos segmentos alinhados as três eixos principais, permitindo ao usuário explorar o interior do volume.

### 1.3 Trabalhos relacionados

Em relação à visualização em modo foco+contexto, Hauser et al. (2001) utilizaram distintas técnicas de renderização, como a renderização direta de volume (*DVR*<sup>2</sup>) e a projeção de intensidade máxima (*MIP*<sup>3</sup>), combinando diferentes funções de transferência para ressaltar regiões de interesse previamente segmentadas. Outras pesquisas apresentaram soluções baseadas em funções de transferência que permitem ao usuário selecionar qual estrutura enfatizar ou visualizar (Kindlmann & Durkin 1998, Kniss, Kindlmann & Hansen 2002a, Roettger, Bauer & Stamminger 2005, Haidacher, Patel, Bruckner, Kanitsar & Groller 2010, Engel et al. 2006). Mas, como estas soluções realçam as características de interesse levando em

---

<sup>1</sup>Graphics Process Unit

<sup>2</sup>Direct Volume Rendering

<sup>3</sup>Maximum Intensity Projection

conta as intensidades das amostras e suas derivadas, acabam por não permitir uma ótima discriminação espacial ao serem aplicadas em imagens de ressonância magnética. Zhou et al. (2002) adotaram uma abordagem *orientada a geometria* para dividir o volume de dados em duas regiões, uma é chamada foco e outra contexto. Nesta abordagem, a renderização é feita em dois passos: o foco é renderizado primeiro com a técnica de DVR, e o contexto é renderizado em seguida com métodos de renderização não foto-realística (NPR<sup>4</sup>). Viola, Kanitsar & Groller (2005) introduziram o conceito de renderização volumétrica *orientada a importância*. Nesta abordagem, uma função de importância controla a opacidade dos *voxels* independente do ângulo de visão do observador. Tal função é definida com base nas regiões focais e de contexto previamente segmentadas, de forma que assegure que a visibilidade das regiões focais seja independente do ângulo de visão de um observador. Porém, conforme entendemos, a interatividade não é a principal preocupação desses trabalhos.

Recentemente, Sikachev, Rautek, Bruckner & Gröller (2010) apresentaram um modo foco+contexto dinâmico, que destaca as características salientes durante a interação do usuário. No entanto, as interações estão limitadas a transformações afins da geometria *proxy*, ou da geometria dos dados volumétricos, e não da geometria que realmente é exibida. Portanto, o *suporte às consultas dinâmicas* é precário. Neste trabalho, apresentamos uma *sonda volumétrica*, com a qual um usuário pode ajustar a posição do foco de interesse arrastando o ponteiro do *cursor*, e a aparência diferenciada entre o foco e o contexto é ajustada automaticamente. Além disso, projetamos uma arquitetura na qual as regiões tanto focais quanto de contexto são renderizadas em GPUs atuais.

Devido ao crescente tamanho e resolução dos volumes de dados, outro tipo útil de visualização baseada em foco+contexto é conseguido utilizando-se as lentes mágicas. As lentes são, de fato, filtros espaciais para modificar a aparência visual dos objetos, ou para melhorar a aparência dos dados de interesse ou para suprimir dados que dispersam atenção do usuário (Bier, Stone, Pier, Buxton & DeRose 1993). LaMar et al. (2001) implementaram lentes de volumes baseadas em texturas, enquanto Wang et al. (2005) estenderam a ideia para dar suporte a lentes de aumento de diferentes características e integraram-nas ao fluxo da renderização volumétrica em GPU utilizando *ray casting*. Nestas duas abordagens é necessário segmentar a imagem caso se queira ampliar uma região de interesse específica. Neste trabalho, aplica-se o algoritmo de corte proposto por Wu et al. (2012) para fornecer, sem nenhuma segmentação, vistas que os neurocirurgiões teriam na sala de operação, ou seja, os tecidos expostos. Assim, propomos uma *lente móvel* que visualize somente os *voxels* mais próximos ao observador, evitando a combinação das propriedades ópticas de todos os *voxels* ao longo do raio de visão e, portanto, a alteração do conteúdo da informação original. Além disso, ao invés das funções de transferência multidimensionais (Kindlmann & Durkin 1998, Kniss et al. 2002a, Roettger et al. 2005), adotamos duas funções de transferência 1D para distinguir tanto o foco de interesse quanto a região circundante. Isso garante uma visualização orientada a preservação da informação.

---

<sup>4</sup>Non-Photorealistic Rendering

## 1.4 Estrutura da dissertação

A dissertação começa expondo no Capítulo 2 os principais conceitos para entendimento de nossa proposta. São abordados principalmente os conceitos de neuro-imagens, a renderização direta de volume (DVR) e o algoritmo de *ray casting*, a visualização foco+contexto, a GPU e o protótipo VMTK desenvolvido por um grupo de pesquisa do Laboratório de Controle e Automação Industrial da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas.

No Capítulo 3 apresentam-se as diretrizes de resolução do problema de realce global e localizado do volume 3D. São detalhados o projeto do editor de função de transferência, o projeto da lente móvel e o projeto da sonda volumétrica.

Os resultados alcançados em termos de qualidade de imagem, de interatividade e de usabilidade das ferramentas são apresentados Capítulo 4.

No Capítulo 5, realizamos uma análise e discussão dos resultados alcançados no trabalho. Finalmente, as conclusões e trabalhos futuros são expostos no Capítulo 6.

## Preliminares

Neste capítulo sintetizamos os conceitos necessários para entendimento da nossa proposta a fim de que a leitura desta dissertação seja auto-contida.

### 2.1 Neuro-Imagens

Ao longo da história, o cérebro foi um dos órgãos mais estudados da anatomia humana. O cérebro constitui o centro do sistema nervoso central dos humanos, controlando as atividades vitais e involuntárias como visão, fala, pensamento, respiração, funções cardiovasculares (Simon 1997). Inúmeros estudos foram realizados para fornecer maior conhecimento sobre as estruturas cerebrais e suas funções, como também estudos orientados ao diagnóstico de lesões, doenças e traumas cerebrais (Kalat 2007).

Com o objetivo de estudar o cérebro de forma não invasiva, pesquisadores, médicos e cirurgiões recorrem hoje em dia à neuro-imagem. Diferentes técnicas são usadas para, de forma direta ou indireta, conseguir uma imagem da *estrutura* e do *funcionamento* do cérebro (Zimmerman, Gibby & Carmody 2000). As imagens estruturais fornecem informação para o estudo e diagnóstico de patologias estruturais intracranianas (Drevelegas 2011); enquanto as imagens funcionais apoiam na compreensão da organização do cérebro e os processos mentais que acontecem nele (Jeppard, Matthews & Smith 2001). Estas imagens são genericamente denominadas neuro-imagens.

São expostas sumariamente nesta seção as principais técnicas de produzir neuro-imagens.

#### 2.1.1 Imagens Estruturais

As imagens estruturais, ou anatômicas, são utilizadas para ver a forma, disposição e localização dos órgãos internos do corpo humano.

**Tomografia Computadorizada:** A tomografia computadorizada (CT<sup>1</sup>), introduzida no ano 1972, utiliza raios-X como meio de criação de imagem (Hüsing, Jäncke & Tag 2006).

---

<sup>1</sup>Computed Tomography



Durante um exame de CT, o paciente é colocado no interior do aparelho tomográfico em posição decúbito dorsal, com as faces para acima. O princípio básico é projetar feixes de raios-X através do cérebro e medir no lado oposto as radiações não absorvidas. Para isto, dispõe-se de emissores e detectores de raios-X. Ambos dispositivos são montados opostamente sobre a estrutura do aparelho e vão sendo rotacionados ao redor do cérebro, permitindo que os feixes de raios-X sejam projetados em diferentes direções. Estes feixes são detectados e transformados em imagens 2D ou 3D do cérebro. Dependendo da densidade do tecido cerebral através do qual o feixe de raios-X passou, as amostras das imagens de tomografia construídas variam em níveis de cinza. Materiais de alta densidade como o osso absorvem muita radiação, enquanto materiais de baixa densidade como ar ou sangue absorvem pouca radiação. Como resultado, regiões de alta densidade (osso) são mais claras e regiões de baixa densidade (fluido cérebro espinhal), mais escuras.

Na figura 2.1 são apresentadas duas fatias de tomografia computadorizada: um corte axial (figura 2.1.a) e outro sagital (figura 2.1.b). Observa-se a homogeneidade na região de tecido mole e região de tecido duro, e o alto contraste entre as duas regiões.

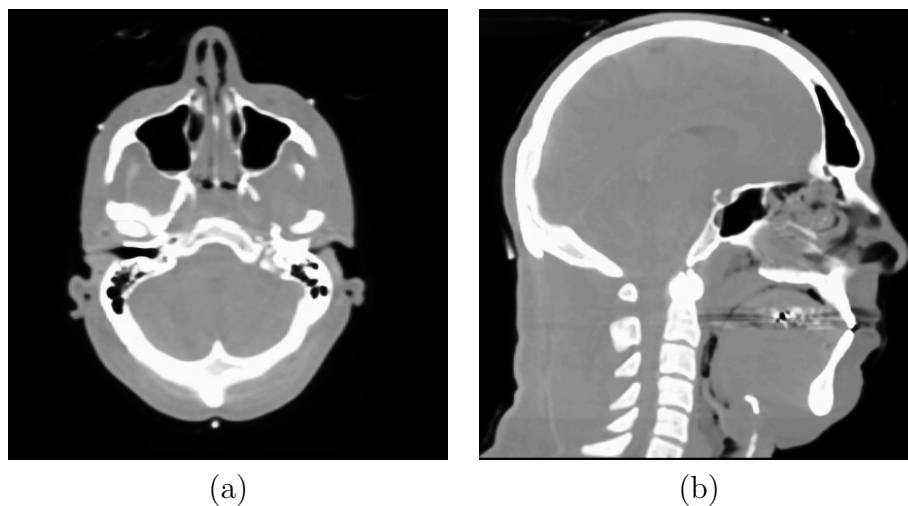


Figura 2.1: Imagens de CT: (a) Corte axial. (b) Corte Sagital. [fonte dos dados: (Roettger 2006)]

**Imagem de Ressonância Magnética:** A técnica de imageamento baseado em ressonância magnética (MRI<sup>2</sup>), desenvolvida no ano 1977, permite o exame não-invasivo do corpo humano sem o uso de radiação ionizante (Hornak 1996, Hüsing et al. 2006).

O princípio básico da imagem de ressonância magnética é submeter o paciente a um forte campo magnético (Weishaupt, Köchli & Marincek 2006). O campo magnético é expressada em unidades chamadas de *Tesla* (T). Aparelhos atuais de ressonância

---

<sup>2</sup>Magnetic Resonance Imaging

magnética de uso clínico são de 1,5 até 3 T; isto indica que o campo magnético gerado é aproximadamente 30.000 a 60.000 vezes mais forte do que força do campo magnético da Terra. Quando o paciente está exposto a esse campo magnético, todos os prótons do núcleo de hidrogênio no seu corpo tendem a se alinhar contra ou a favor desse campo, com uma sutil predominância na direção a favor.

Na direção transversal deste campo são aplicados pulsos de radiofrequência, da ordem de MHz, em intervalos de tempo determinados sobre regiões específicas do corpo. Esses pulsos fazem com que os prótons naquela área absorvem a energia e girem a uma frequência e em uma orientação específica, dependendo do tipo de tecido e da intensidade do campo magnético principal. A mudança do eixo de rotação, ou a precessão, dos prótons produzem sinais que são medidos por detetores específicos em torno da cabeça do paciente. Ao se desligar as ondas de radiofrequência, os prótons voltam para o seu estado inicial de equilíbrio, denominado estado de relaxação, após um intervalo de tempo de relaxação. Destacam-se dois tipos de relaxação: a *relaxação longitudinal* (ou relaxação *spin-lattice*, rotação-quadratura, T1) e a *relaxação transversal* (ou relaxação *spin-spin*, rotação-rotação, T2). O tempo de relaxação T2 é sempre muito menor que o tempo T1. O aparelho de ressonância magnética pode construir, a partir dos sinais induzidos pelo movimento de precessão ou dos tempos de relaxação, imagens do cérebro com alta resolução espacial, da ordem de  $0.5\text{mm}^3$  a  $3\text{mm}^3$ .

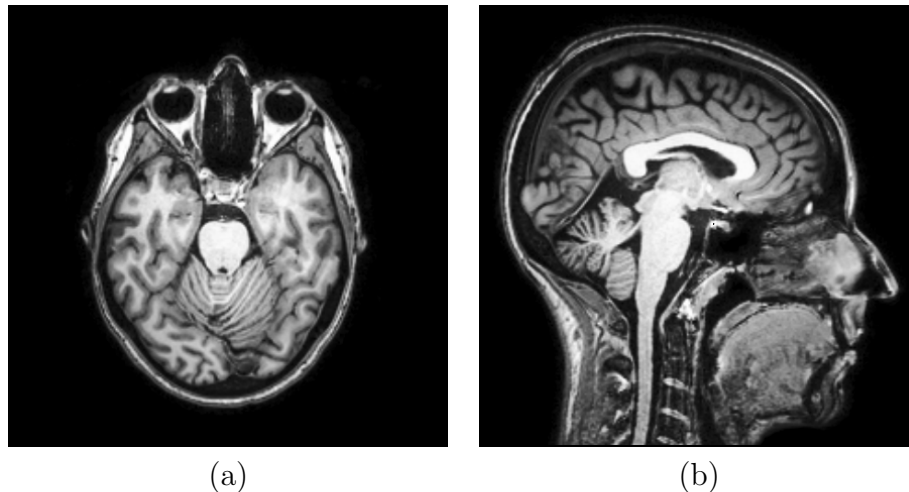


Figura 2.2: Imagens de ressonância magnética: (a) Corte axial. (b) Corte sagital.

Comparadas com as imagens cerebrais baseadas em tomografia computadorizada, as imagens de ressonância magnética fornecem maior nitidez e contraste. A melhora reflete o fato de que as propriedades de relaxação e densidade dos prótons fornecem maiores possibilidades de variações nas respostas, consegue-se distinguir a matéria branca da matéria cinzenta do cérebro. Além disso, a alta resolução espacial permite a percepção de estruturas pequenas e sutis, abrindo muitas possibilidades ao campo de pesquisa,

diagnóstico e exploração de lesões cerebrais. No entanto, a correspondência entre os tipos de tecidos e os tempos de relaxação não é biunívoca e a relação sinal-ruído ainda é relativamente baixa. Tudo isso constitui um desafio para seu processamento (Haidacher et al. 2010).

Na figura 2.2 exibe-se duas fatias de uma imagem de ressonância magnética de tipo T1: um corte axial (figura 2.2.a) e outro sagital (figura 2.2.b). Nota-se o alto contraste entre os tecidos moles: cérebro, medula espinhal e cerebelo.

### 2.1.2 Imagens Funcionais

Em complemento às imagens estruturais, as imagens funcionais são utilizadas para análise da função dos órgãos.

**Eletroencefalografia e Magnetoencefalografia:** A eletroencefalografia (EEG<sup>3</sup>) foi a primeira técnica não-invasiva de neuro-imagem descoberta em 1929 (Hüsing et al. 2006). Utilizando um dispositivo chamado de eletroencefalógrafo, fornece medições em tempo real da atividade do cérebro a partir de sinais captados por eletrodos colocados sobre a cabeça do paciente.

Pesquisadores e neurocientistas ainda utilizam esta técnica para avaliar respostas a estímulos sensoriais e a atividade rítmica do cérebro. Anormalidades encontradas nos dados da eletroencefalografia pode sugerir a presença de epilepsia, tumores ou outras patologias médicas no paciente.

A técnica de magnetoencefalografia (MEE<sup>4</sup>) está intimamente relacionada à eletroencefalografia. No lugar de campos elétricos, a magnetoencefalografia mede campos magnéticos produzidos pela atividade cerebral (Hüsing et al. 2006). A distribuição espacial dos campos magnéticos são analisados para localizar as fontes de atividade dos neurônios e criar uma imagem anatômica que forneça informações sobre a estrutura e função do cérebro.

**Tomografia por Emissão de Pósitrons:** A tomografia por emissão de pósitrons (PET<sup>5</sup>) é uma técnica não-invasiva de diagnóstico e investigação em tempo real, capaz de produzir imagens 3D das atividades funcionais do corpo humano.

No cérebro, os neurônios utilizam uma variedade de moléculas orgânicas e compostos para receber e transmitir mensagens. Estas moléculas e compostos são trazidos pelo sangue ou fabricados no cérebro a partir da matéria-prima presente no sangue. O aparelho de PET captura e cria uma imagem da distribuição de algum tipo específico de moléculas e compostos que circulam no cérebro, refletindo suas variações metabólicas ou seus fluxos sanguíneos. Para detecção destas moléculas e compostos são necessários

---

<sup>3</sup>Electroencephalography

<sup>4</sup>Magnetoencephalography

<sup>5</sup>Positron Emission Tomography

radiotraçadores ou radionuclídeos específicos. No PET estes radiotraçadores são fótons de raios gama que resultam da aniquilação de elétrons e pósitrons. Baseado no movimento de fótons, o aparelho de tomografia por emissão de pósitrons mede os raios gama e cria uma imagem 3D da atividade cerebral em diferentes regiões do cérebro (Hüsing et al. 2006).

O maior benefício da técnica de imageamento por PET é que, por utilizar radiotraçadores que sejam equivalentes ou análogos às substâncias que ocorrem naturalmente no corpo de um paciente, permite reconhecer funções corporais muito específicas. Por exemplo, através do traçador Oxigênio-15 no fluxo sanguíneo cerebral, os cientistas podem localizar regiões cerebrais que entram em atividade enquanto uma pessoa fala, ouve música, sente dor, ou faz uma outra atividade específica. Comparando as imagens obtidas com o paciente em atividade com as tomadas previamente ou depois da tarefa, consegue-se ganhar entendimento sobre a organização e funcionamento do cérebro.

**Imagem de Ressonância Magnética Funcional:** A técnica de imageamento de ressonância magnética funcional (fMRI<sup>6</sup>) foi desenvolvida nos anos 90s (Jezzard et al. 2001). Ela é aplicada no estudo da cognição e na neurociência (Hüsing et al. 2006).

Em um exame de fMRI, o paciente é submetido a estímulos e, em resposta, regiões do cérebro entram em atividade. Com base nas variações químicas, nas alterações do fluxo sanguíneo, ou no consumo extra de oxigênio, determinam-se regiões cerebrais que estão processando informação ou enviando ordens ao corpo. Enquanto um paciente realiza diferentes tarefas, os cientistas tentam associar a cada região cerebral as funções que ela controla. A fMRI tornou-se uma técnica preferida de mapeamento de atividade cerebral, pelo fato que ela não precisa que o paciente seja submetido a uma intervenção cirúrgica, tenha que ingerir substâncias ou seja exposto a radiações ionizantes.

## 2.2 Diagnóstico Auxiliado por Computador Baseado em Imagens

O *diagnóstico auxiliado (ou assistido) por computador (CAD<sup>7</sup>)* consiste de procedimentos computacionais que auxiliam o médico na interpretação de imagens médicas e na detecção e diagnóstico de doenças, atuando como alguém que dá uma segunda opinião (Doi 2006).

Utilizando algoritmos de visão computacional e processamento de imagens (Yoo 2004), o sistema realiza marcações de regiões da imagem com anomalias. Técnicas de aprendizado de máquinas e regras de inferência são aplicadas sobre uma grande base de conhecimentos sobre sintomas e tratamentos possíveis, a fim de detectar e identificar a doença, e posteriormente sugerir um tratamento adequado (Neri, Caramella & Bartolozzi 2007).

---

<sup>6</sup>Funcional Magnetic Resonance Imaging

<sup>7</sup>Computer-Aided Diagnosis

Os inúmeros esforços no desenvolvimento de CADs especializados foram bem sucedidos: a mamografia computadorizada para detecção automática de massas, microcalcificações e tumores na mama da mulher em imagens de mamografia (Tang, Rangayyan, Xu, El Naqa & Yang 2009), a colonografia virtual para detectar crescimentos de pólipos nas suas fases iniciais a fim de serem removidos antes que o câncer se desenvolva (Heiken, Peterson & Menias 2005), CADs para detecção automática de nódulos pulmonares em imagens CT do tórax (Namin, Moghaddam, Jafari, Esmaeil-Zadeh & Gity 2010), sistemas para detecção de calcificações nas artérias coronarianas e aorta, aplicações para determinar se há vazamento de vasos sanguíneos como um indicador precoce de diabetes através da análise de imagens da retina do fundo de olho, sistemas de detecção precoce de câncer da pele mediante inspeção de manchas na pele (Neri et al. 2007), e procedimentos computacionais que detectam lesões Alzheimer em fase inicial no cérebro (Illán, Górriz, López, Ramírez, Salas-Gonzalez, Segovia, Chaves & Puntonet 2011).

Pela complexidade da estrutura interna da cabeça, a presença do cérebro e demais órgãos que controlam funções vitais do corpo humano, o desenvolvimento de técnicas assistidas por computador para detecção de lesões cerebrais ainda é um campo de pesquisa desafiadora.

## 2.3 A GPU

A GPU é a unidade de processamento gráfico responsável pela conversão a partir de modelos 2D e 3D em imagens rasterizadas (Nvidia 2012b). Ela consiste de processadores altamente especializados para processamento de dados paralelos em modo SIMD<sup>8</sup> (Weiskopf 2007).

O processo de conversão de primitivas poligonais em fragmentos de uma imagem *raster* é feito através de uma sequência de etapas de processamento, chamado de *pipeline gráfico de renderização* ou *fluxo gráfico de renderização*. Um fragmento é um meta-*pixel* ao qual é associado uma série de atributos gráficos, como cor, profundidade *z* e opacidade. Cada fragmento corresponde a um *pixel* do *frame buffer* alocado em uma memória interna da GPU. Duas interfaces de programação de aplicativos (API<sup>9</sup>) mais conhecidas destas unidades são: OpenGL<sup>10</sup> (OpenGL 2010) e Direct 3D (Microsoft 2011). Como OpenGL é considerada um padrão gráfico industrial, mantido pelo consórcio Grupo Khronos, ela será utilizada neste projeto de Mestrado.

Na versão OpenGL 1.0 as GPUs provêem somente um fluxo fixo de processamento gráfico, representado pela sequência de blocos em cinza na figura 2.3. A partir da versão OpenGL 2.0 este fluxo tornou programável. Para evitar degradação no desempenho, o conjunto de funções reprogramáveis é bem limitado e os novos trechos de instrução precisam seguir à risca as regras de processamento do fluxo gráfico. Estes novos trechos de instrução executáveis em

---

<sup>8</sup>Single Instruction, Multiple Data

<sup>9</sup>Application Program Interface

<sup>10</sup>Open Graphics Library

GPUs são denominados *shaders*. Os blocos avermelhados na figura 2.3 mostram os pontos que podemos alterar com os *shaders* na versão OpenGL 3.1 (Nvidia 2012b).

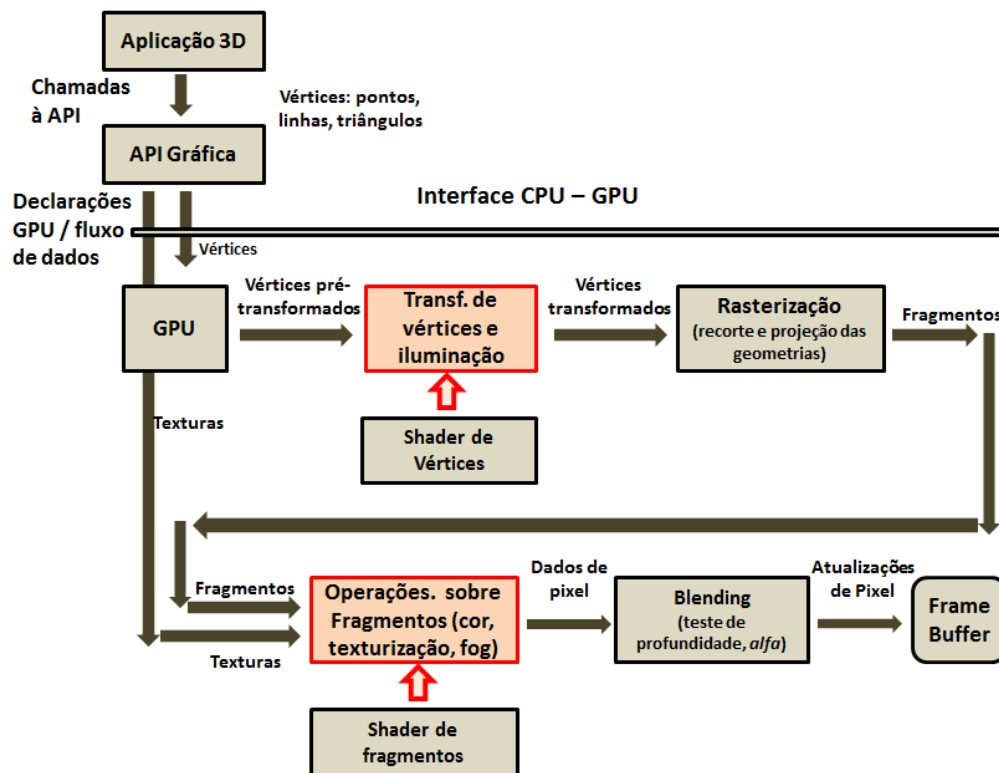


Figura 2.3: Pipeline gráfico.

Essencialmente, o fluxo fixo do *pipeline* gráfico é constituído por seguintes etapas:

1. As primitivas geométricas e os dados de texturas são transferidos pelo aplicativo para GPU.
2. Os vértices e os seus vetores normais são transformados. As coordenadas de texturas são geradas e transformadas. Tudo é de acordo com as matrizes de transformações afins especificadas.
3. Os vértices são conectados conforme as primitivas gráficas definidas pela API de OpenGL. Quando necessário, primitivas gráficas convexas são trianguladas.
4. A cor em cada vértice é determinada com base nos parâmetros do modelo de iluminação da API de OpenGL.
5. O recorte e a rasterização são aplicados. O fluxo de vértices é então transformado em um fluxo de fragmentos.

6. Cada fragmento é texturizado conforme a correspondência definida na API de OpenGL. O efeito *fog*, caso especificado, é adicionado.
7. Antes que os valores dos fragmentos sejam armazenados em *frame buffer*, no estágio chamado de *blending*, vários testes são realizados para determinar se o fragmento de entrada será descartado ou será visualizado na tela. Tipicamente, os testes de transparência ou alfa, de estêncil, de profundidade são realizados nesta etapa. Também é possível determinar se a cor do fragmento de entrada será combinado com a cor armazenada no *frame buffer* na correspondente posição do *pixel*.

Para facilitar a programação das GPUs, foram desenvolvidas linguagens de programação da GPU de alto nível, com sintaxe similar à da linguagem de programação C (Kernighan & Ritchie 1988). Existem três principais linguagens: o GLSL<sup>11</sup> (GLSL 2010), o HLSL<sup>12</sup> (Microsoft 2012) e o Cg<sup>13</sup> (Nvidia 2012a).

## 2.4 Renderização Direta de Volume

As técnicas de *renderização direta de volume* (DVR<sup>14</sup>) permitem gerar imagens diretamente de um volume de dados obtidos pelos equipamentos de aquisição, sem extrair modelos geométricos. A DVR cria uma imagem bidimensional que reflete em cada *pixel* a propriedade óptica de todos os *voxels* ao longo do raio projetor que atravessa aquele *pixel* (Telea 2008). Como todos os *voxels* participam na geração da imagem final, a DVR é adequada para visualização das estruturas internas de um volume de dados (Engel et al. 2006).

### 2.4.1 Modelo Óptico

O modelo óptico da DVR baseia-se na *teoria óptica de transporte da luz*, que tenta ver o volume de dados como uma nuvem de partículas (meio). Cada partícula do meio interage com a luz e as outras partículas (Engel et al. 2006). Consideram-se os seguintes tipos de interação entre a luz e o meio:

1. *Emissão*: As partículas emitem radiações luminosas, como um gás no ambiente quente emitindo luz pela conversão de calor em energia luminosa.
2. *Absorção*: As partículas absorvem radiações luminosas, como a passagem de um feixe luminoso por um meio translúcido. Isso reduz a energia luminosa.
3. *Espalhamento*: As radiações luminosas podem ser espalhadas pelas partículas do meio, mudando sua direção da propagação. Este espalhamento pode tanto reduzir como incrementar a energia luminosa.

---

<sup>11</sup>OpenGL Shading Language

<sup>12</sup>High-Level Shader Language

<sup>13</sup>C for Graphics

<sup>14</sup>Direct volume rendering

O fenômeno de transporte da luz é muito complexo. Por isso, modelos simplificados são adotados para que eles sejam computacionalmente tratáveis, sendo o *modelo de emissão-absorção* a abordagem mais utilizada (Max 1995, Meissner, Huang, Bartz, Mueller & Crawfis 2000). As interações luminosas de emissão-absorção são expressas pela *integral da renderização de volume*:

$$I(D) = I_0 e^{-\int_{s_0}^D \kappa(t) dt} + \int_{s_0}^D q(s), e^{-\int_s^D \kappa(t) dt} ds, \quad (2.1)$$

onde  $I_0$  representa a radiância no ponto de entrada  $s = s_0$  do fluxo luminoso no volume de dados, enquanto  $I(D)$  é a radiância no ponto de saída  $s = D$  do fluxo. Considera-se que seja esta a radiância que alcança o observador. O primeiro termo da equação (2.1) descreve o fluxo luminoso externo atenuado pelo meio constituído pelos *voxels* do volume. O segundo termo representa a contribuição das partículas do próprio volume, mas atenuada pelo meio ao longo do percurso até o observador.

Uma solução numérica para a *integral de renderização de volume* é dividir o domínio da integral em  $n$  intervalos  $s_0 < s_1 < \dots < s_{n-1} < s_n$ , sendo  $s_0$  e  $s_n = D$  os pontos inicial e final do domínio de integração, respectivamente. Para cada intervalo  $[s_{i-1}, s_i]$  a equação 2.1 se reduz em

$$I(s_i) = I(s_{i-1})T(s_{i-1}, s_i) + \int_{s_{i-1}}^{s_i} q(s)T(s, s_i)ds, \text{ com } T(s_1, s_2) = e^{-\int_{s_1}^{s_2} \kappa(t) dt}. \quad (2.2)$$

$T(s_1, s_2)$  é conhecida por transparência do meio no intervalo  $[s_1, s_2]$ . A figura 2.4 ilustra a partição do domínio da integral em  $n$  intervalos.

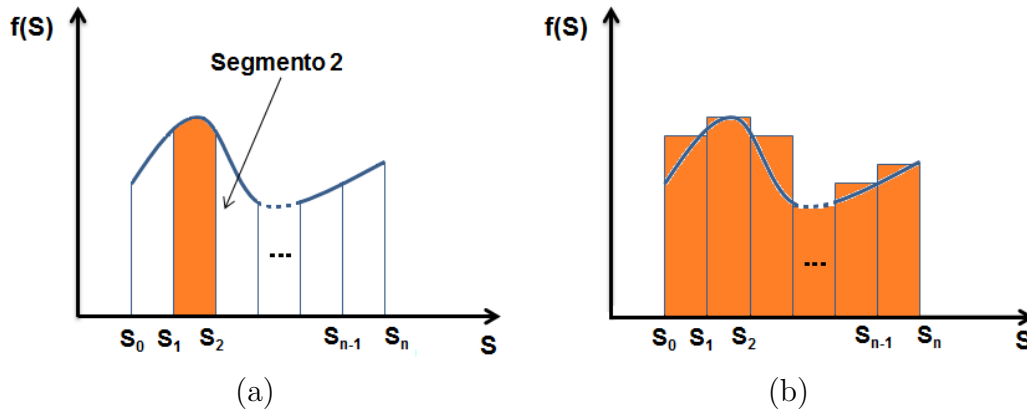


Figura 2.4: (a) Partição do domínio da integração em vários intervalos. (b) Aproximação pela soma de Riemann.

Com isso a Eq. 2.1 pode ser aproximada por

$$I(D) = I(s_n) = I(s_{n-1})T_n + c_n = (I(s_{n-2})T_{n-1} + c_{n-1})T_n + c_n = \dots,$$



o que pode ser re-escrito como um somatório

$$I(D) = \sum_{i=0}^n c_i \prod_{j=i+1}^n T_j, \text{ com } c_0 = I(s_0), \quad (2.3)$$

que é uma *forma discretizada da integral de renderização de volume*. Para a tecnologia disponível, é mais conveniente utilizar o conceito de opacidade  $\alpha$  ao invés de transparência, substituindo-se  $T_i$  por  $(1 - \alpha_i)$ , o que conduz à seguinte equação

$$I(D) = \sum_{i=0}^n c_i \prod_{j=i+1}^n (1 - \alpha_j), \text{ com } c_0 = I(s_0). \quad (2.4)$$

## 2.4.2 Função de Transferência

Para resolver a equação do modelo óptico de emissão-absorção apresentado na seção 2.4.1, seria necessário conhecer os coeficientes de emissão e absorção em cada ponto do volume. Como não há ainda um modelo matemático que represente tais coeficientes em termos dos valores escalares das amostras, usa-se por simplicidade uma correspondência entre os valores escalares das amostras e as propriedades ópticas. As propriedades óptica compreendem a cor, definida pelos componentes vermelho (R), verde (G) e azul (B), e a opacidade (A). Usualmente é o usuário quem atribui estas propriedades, decidindo diretamente a aparência das diferentes estruturas contidas no volume. Este mapeamento é chamado de *função de transferência* (Drebin, Carpenter & Hanrahan 1988, Levoy 1988).

Definição de uma função de transferência (FT) é crítica no processo de renderização volumétrica, pois ela permite identificar estruturas internas no volume e torná-las visíveis (Kniss, Kindlmann & Hansen 2001). O processo de especificação da função de transferência é frequentemente conhecido como *classificação*, por associar distintas propriedades ópticas a diferentes padrões de tecidos (Engel et al. 2006, Telea 2008). A forma mais simples e direta para representar uma função de transferência é através de gráficos bi-dimensionais. Sobre um plano cartesiano os valores escalares das amostras do volume são representados no eixo  $X$  e as propriedades ópticas (cor e opacidade) no eixo  $Y$ , sendo chamada de *função de transferência unidimensional* (1D).

Através da inserção e remoção dos pontos de controle na função de transferência, o usuário pode manipular a relação entre intervalos de valores escalares e os atributos (R,G,B,A) e melhorar os contrastes ou realçar as regiões de interesse. A figura 2.5 mostra uma interface para quatro funções de transferência, do domínio de valores escalares para as quatro componentes do vetor (R,G,B,A). Quando as três componentes (R,G,B) assumem um mesmo valor, denominado intensidade  $I$ , temos uma imagem em escala de cinza, como ilustra a figura 2.6. Nota-se que uma amostra será totalmente opaca, se  $\alpha = 1.0$ , ou totalmente transparente se  $\alpha = 0$ .

Editar funções de transferência por meio de pontos de controle foi considerado um processo difícil, tedioso e demorado, envolvendo inúmeras tentativas e erros, até chegar ao resultado

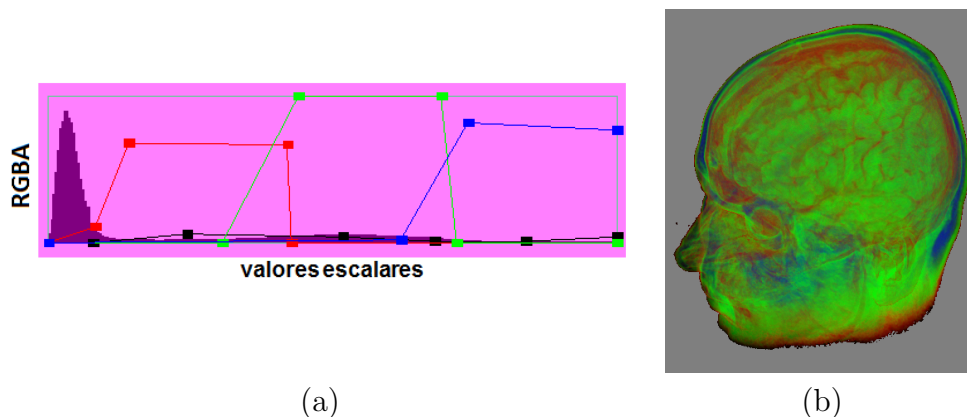


Figura 2.5: Imagem colorida: (a) correspondência independente entre os valores escalares das amostras e os quatro canais R, G, B e A; (b) visualização dos dados volumétricos. [fonte dos dados: (Roettger 2006)]

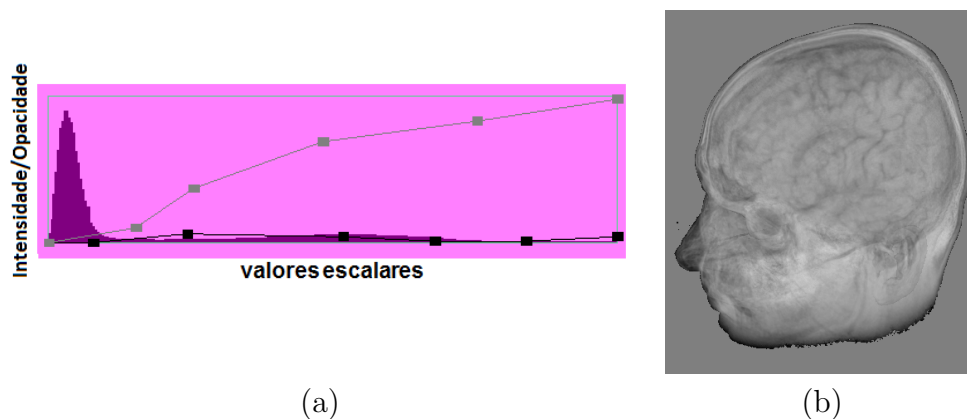


Figura 2.6: Imagem em escala de cinza: (a) correspondência independente entre os valores escalares e os canais I e A; (b) visualização dos dados volumétricos. [fonte dos dados: (Roettger 2006)]

desejado (Kniss et al. 2002a). Nas tentativas de prover mais elementos para suportar a edição de funções de transferência, introduziram-se outras variáveis no domínio da função de transferência: os gradientes e as hessianas dos valores de intensidade (Kindlmann & Durkin 1998, Kniss et al. 2002a), informação de vizinhança e de posição (Roettger et al. 2005), tamanho e visibilidade (Correa & Ma 2009), medidas estatísticas (Haidacher et al. 2010). Tais funções são conhecidas como *funções de transferência multidimensionais* (FTM).

O uso dos gradientes e das hessianas dos valores escalares mostrou muito útil para descri-

minar as regiões em imagens CT. Porém, para imagens RM os resultados não são conclusivos. Na figura 2.7 são apresentadas imagens de CT e RM geradas utilizando funções de transferência baseadas em magnitudes de gradientes e hessianas. As figuras da coluna esquerda exibem as imagens, as figuras da coluna central seus correspondentes histogramas entre valores escalares do volume e magnitude de gradiente e as figuras da coluna direita, os histogramas entre valores escalares e magnitude de hessiana. Observa-se nos histogramas a formação de arcos entre grupos de amostras de CT que correspondem à região de transição entre dois tecidos (figura 2.7.a), enquanto, para a imagem RM, o histograma de gradientes não apresenta um arco bem definido entre tecidos (figura 2.7.b), sendo difícil ou até impossível detectar as transições entre tecidos (Roettger et al. 2005). A galeria de imagens do projeto Simian desenvolvido por Kniss, Kindlmann & Hansen (2002b) corrobora a nossa observação.

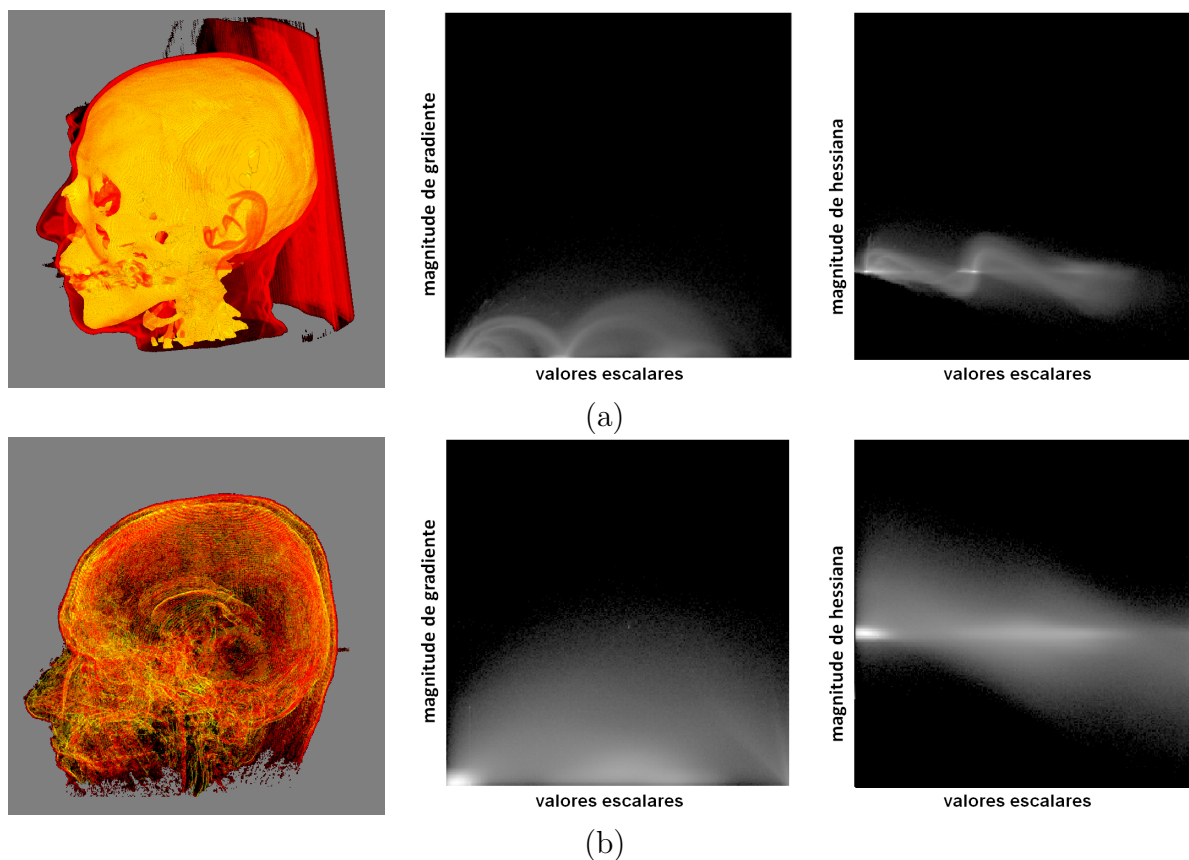


Figura 2.7: Função de transferência multi-dimensional baseado em gradientes e hessianas: (a) Imagem de CT. (b) Imagem de ressonância magnética. [fonte dos dados: (Roettger 2006)]

Além do mais, acrescentar a dimensionalidade da função de transferência incrementa complexidade de manipulação da mesma, pois adicionar mais variáveis no domínio da função de transferência requer uma etapa de pré-processamento, geralmente custosa, podendo comprometer o quesito de interatividade. Vale a pena mencionar que, embora tenham sido desenvolvidas técnicas automáticas para especificação de funções de transferência (He, Hong, Kaufman & Pfister 1996, Marks, Andalman, Beardsley, Freeman, Gibson, Hodgins, Kang,

Mirtich, Pfister, Ruml, Ryall, Seims & Shieber 1997, Haidacher et al. 2010), ainda preferem-se os métodos manuais ou semi-automáticos (Kindlmann & Durkin 1998), pois estes mantêm no usuário a decisão do que se deseja ver (Pfister, Lorensen, Bajaj, Kindlmann, Schroeder, Avila, Raghu, Machiraju & Lee 2001).

Em vista disso, apesar de todas as desvantagens, o paradigma de edição de funções de transferência 1D na base de pontos de controle é ainda o mais popular. Em aplicações médicas, ele é conhecido por janelamento das imagens. Do nosso conhecimento, esta abordagem é, no entanto, aplicada sobre um volume como todo e não fornece suporte para uma exploração mais discriminada dos dados mediante variação das propriedades ópticas somente das amostras de interesse.

### 2.4.3 Ray casting

O traçado de raio, ou *ray casting*, é a técnica de renderização volumétrica orientada a imagem (Levoy 1988, Engel et al. 2006).

Para cada *pixel* da imagem, no mínimo um raio é lançado dentro do volume. Este raio é amostrado num espaçamento  $\Delta d$  para determinar o valor escalar em alguns pontos e calcular a *integral da renderização de volume* segundo a aproximação dada em (Eq. 2.4). Para cada amostra é calculada suas propriedades ópticas com base na função de transferência especificada. Estas são, então, integradas para gerar a cor final no *pixel*. O processo de *ray casting* é ilustrado na figura 2.8. Vale observar que nem sempre o ponto amostrado  $P$  ao longo de um raio coincide com as amostras obtidas por um equipamento de aquisição. Neste caso, é feita uma reamostragem na vizinhança de  $P$ .

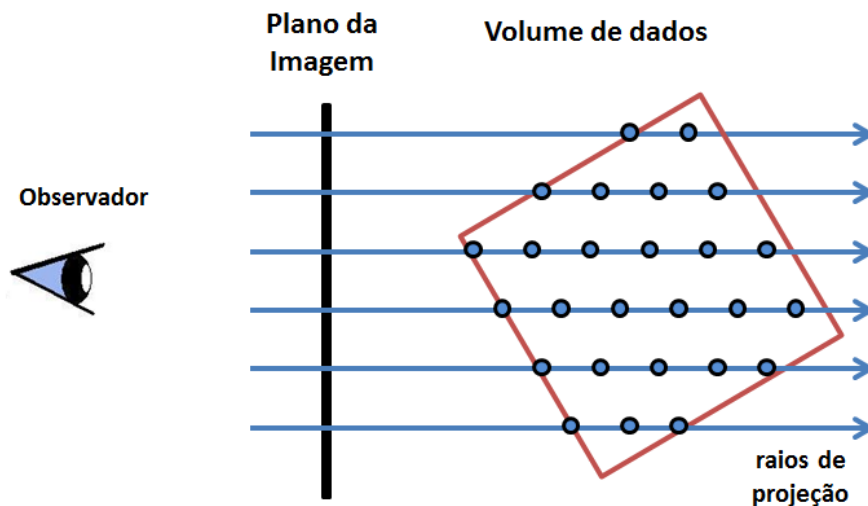


Figura 2.8: *Ray casting*: as amostras ao longo de cada raio são representadas pelos discos cinzentos.

### 2.4.4 Composição

A composição é uma solução recorrente para computar a Eq. (2.4). Ela envolve operações simples e totalmente independentes para cada *pixel*.

Dependendo da origem de onde os raios de projeção saem, existem dois esquemas de composição: *frente-para-trás* e *trás-para-frente*.

#### Frente-para-trás

Na composição *frente-para-trás*, os raios de projeção partem da câmera ou observador.

A expressão algébrica da composição de *frente-para-trás* é dada por em cada iteração:

$$\begin{aligned} C_{dst} &= C_{dst} + (1 - \alpha_{dst})C_{src} \\ \alpha_{dst} &= \alpha_{dst} + (1 - \alpha_{dst})\alpha_{src}, \end{aligned} \tag{2.5}$$

onde  $C$  representa uma cor em RGB já ponderada pela sua opacidade  $\alpha$ . O sub-índice *src* denota as propriedades ópticas da amostra corrente, e o sub-índice *dst*, as propriedades ópticas até então acumuladas no *pixel*. Nesta recorrência, iniciamos com a amostra mais próxima do observador,  $C_{dst} = C_n$  e  $\alpha_{dst} = \alpha_n$ .

#### Trás-para-frente

Neste esquema os raios de projeção atravessam o volume em direção inversa, de *trás-para-frente*. A iteração começa em um ponto na face do fundo do volume e termina em um ponto da face mais próxima do observador.

Esta composição de *trás-para-frente* é expressa por

$$\begin{aligned} C_{dst} &= (1 - \alpha_{src})C_{dst} + C_{src} \\ \alpha_{dst} &= (1 - \alpha_{src})\alpha_{dst} + \alpha_{src}, \end{aligned} \tag{2.6}$$

ponderando a cor acumulada no *pixel*,  $C_{dst}$ , com a opacidade da amostra em corrente  $\alpha_{src}$

A principal desvantagem desta abordagem é que não é possível terminar o percurso de um raio de forma antecipada (Levoy 1990). Mesmo que  $\alpha_{dst}$  alcance um valor de opacidade  $\alpha_{dst} > 0.99$ , precisaria ainda cumprir o condicional de atravessar o volume para determinar a cor final do pixel, que usualmente é dominada pela cor da amostra mais próxima do observador (Engel et al. 2006).

### 2.4.5 Uma Implementação em GPUs

Com o aparecimento da GPU<sup>15</sup> programável, tornou-se factível executar a técnica de *ray casting* em tempo interativo com uso dos *shaders* de fragmentos (Stegmaier, Strengert, Klein & Ertl 2005).

---

<sup>15</sup>Graphics Process Unit

Em termos das instruções da API de OpenGL, para imagear um volume de amostras obtidas pelo equipamento de aquisição médica, carrega-se este volume como uma textura 3D e desenha-se um cubo texturizado com esta textura. O cubo é denominado *proxy geometry*. Importantes são as modificações que devem ser feitas no fluxo de fragmentos do *pipeline* gráfico e a transferência dos dados para a memória de vídeo da GPU antes de iniciar o *shader* de fragmento.

Um algoritmo de *ray casting* baseado na GPU é apresentado no algoritmo 2.1 que requerem a transferência dos seguintes dados:

- o volume de dados 3D representado em uma textura 3D, *VolumeDeDados*,
- a função de transferência em uma textura 1D, *FuncaoDeTransferencia*,
- tamanho de passo de amostragem ao longo do raio de projeção, *tamanhoDePassos*.

Ao passar para o *shader*, a variável de  $cor_{dest}$  em formato  $RGBA\alpha$  é inicializada em zero (Alg. 2.1, linha 1). Logo determina-se a posição  $(x, y, z)$  de entrada do raio no volume de dados (Alg. 2.1, linha 2) e sua direção (Alg. 2.1, linha 3). Durante o percurso do raio dentro do volume (Alg. 2.1, da linha 4 até 10), para cada posição ao longo do raio é recuperado o valor escalar do volume de dados (linha 5). Através da função de transferência é obtida a propriedade óptica daquele valor escalar (linha 6). De acordo com o esquema de composição especificado (de frente-para-trás, neste caso), são atualizadas a cor e a opacidade (linhas 7 e 8). Em seguida, avança-se um passo igual a *tamanhoDePassos* na direção do raio (linha 9). Quando se sai fora do volume de dados, acaba o laço. A cor acumulada em  $cor_{dest}$  é retornada e exibida no *pixel* de início do raio (Alg. 2.1, linha 11). O tempo de execução do algoritmo pode ser ainda reduzido pelas técnicas de terminação antecipada do raio, amostragem adaptativa e desvio de espaços vazios (Engel et al. 2006).

---

**Algoritmo 2.1** *Ray casting* baseado em GPU.

---

**Require:** *VolumeDeDados*

**Require:** *FuncaoDeTransferencia*

**Require:** *tamanhoDePasso*

```

1:  $cor_{dest}[r][g][b][a] \leftarrow (0, 0, 0, 0)$ 
2:  $posicao(x, y, z) \leftarrow$  coordenadas da textura 3D do fragmento
3: determine a direção do raio direcaoRaio
4: while  $posicao(x, y, z) \in$  VolumeDeDados do
5:    $escalar \leftarrow$  VolumeDeDados(posicao)
6:    $cor_{src} \leftarrow$  FuncaoDeTransferencia(escalar)
7:    $cor_{dest}[r][g][b] \leftarrow cor_{dest}[r][g][b] + (1.0 - cor_{dest}[a]) * cor_{src}[r][g][b]$ 
8:    $cor_{dest}[a] \leftarrow cor_{dest}[a] + (1.0 - cor_{dest}[a]) * cor_{src}[a]$ 
9:    $posicao = posicao + direcaoRaio * tamanhoDePasso$ 
10: end while
11: return  $cor_{dest}$ 

```

---

Na figura 2.9 são exibidos alguns exemplos de imagens geradas com este algoritmo de *ray casting*.

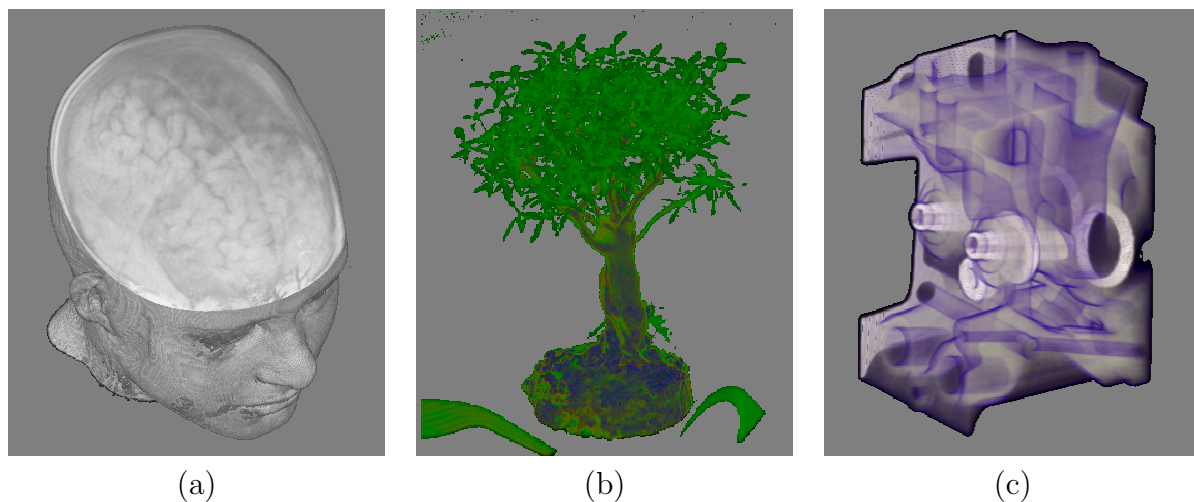


Figura 2.9: Renderização direta de volume: (a) Cabeça MRI. (b) Bonsai CT. (c) Motor CT. [fonte dos dados (b) e (c): (Roettger 2006)]

## 2.5 Visualização de Foco+Contexto

A ideia fundamental da visualização foco+contexto é permitir ao usuário, observar ao mesmo tempo, o objeto de interesse apresentado em maior detalhe e os objetos circundantes em menor detalhe.

De acordo com Card et al. (1999), o paradigma foco+contexto baseia-se em três premissas:

1. O usuário precisa tanto da visão geral (contexto) como das informações detalhadas (foco) simultaneamente.
2. A informação necessária no contexto pode ser diferente do que é necessária no foco.
3. Os dois tipos de informação (foco+contexto) podem ser combinados e exibidos dentro de uma vista (dinâmica) simples.

Nos dados volumétricos em geral e, particularmente, em imagens médicas, frequentemente encontra-se regiões de interesse que são ocluídas pelo tecido circundante. Isto é um dos problemas com as técnicas de DVR (Roettger et al. 2005). A abordagem típica para tratamento deste problema é utilizar uma função de transferência que oculte o tecido circundante da região de interesse. Neste caso, perderíamos a visão do contexto. Abordagens de visualização de foco+contexto, como lentes mágicas (Bier et al. 1993, LaMar et al. 2001, Wang et al. 2005), renderização não-fotorealística (Lu, Morris, Ebert, Rheingans & Hansen 2002, Bruckner & Gröller 2005), renderização com múltiplas técnicas (Hauser et al. 2001, Zhou et al. 2002) e modulação de opacidade (Levoy 1988, Ebert & Rheingans 2000, Zhou, Döring & Tönnies 2004),

foram desenvolvidas para fornecer uma visão conjunta do foco e contexto, tornando-se uma área importante de pesquisa durante os últimos anos. No entanto, vale indicar que a principal limitação das técnicas de foco+contexto é que os dados têm que passar por algum processo de segmentação prévia que ainda é um problema em aberto para neuro-imagens.

## 2.6 Consultas Dinâmicas

As *consultas dinâmicas* atualizam continuamente os dados que são visualizados de acordo com as ações do usuário, tipicamente via controles e botões (*sliders*, *radio button*, *checkbox*) (Williamson & Shneiderman 1992). Eles são aplicados no contexto da busca de informação visual (Shneiderman 1994). Grandes porções de dados desnecessários são filtradas, permitindo que subconjuntos relevantes, em tamanho menor, sejam analisados e visualizados de forma eficiente (Spence 2007). A realimentação visual é imediata, dentro de alguns milissegundos.

Apesar das consultas dinâmicas estarem intimamente ligadas às tecnologias de banco de dados, elas foram estendidas para o campo da *visualização de informação* recebendo o nome de *visualização orientada a consulta* (QDV<sup>16</sup>). A QDV consiste em limitar/realçar a visualização dos dados que o usuário considera importantes ou de seu interesse (Gosink, Garth, Anderson, Bethel & Joy 2011). A QDV fornece uma abordagem geral para a exploração visual com realce das regiões de interesse guiado pelo usuário, demonstrando ser adequada à análise e à visualização de grandes conjuntos de dados (Stockinger, Shalf, Wu & Bethel 2005) e dados multivariados (Glatter, Huang, Ahern, Daniel & Lu 2008).

## 2.7 O VMTK

O VMTK<sup>17</sup> é um protótipo de visualização interativa munido de ferramentas para auxiliar os médicos no diagnóstico de lesões corticais desenvolvido por Batagelo et al. (Batagelo & Wu 2008, Wu, Yauri Vidalon & de Souza Watanabe 2011b). Lesões corticais são anormalidades estruturais localizadas na superfície mais externa do cérebro (Swenson 2012). A principal característica deste *software* é que ela permite que um neurocirurgião descasque a região cortical em camadas paralelas ao couro cabeludo em busca daquelas lesões (Wu et al. 2012).

Na figura 2.10 é exibida a interface de usuário do VMTK. Consiste de 4 janelas de visualização dos dados: uma vista 3D e três vistas 2D (axial, sagital e coronal) do volume sob investigação. Na imagem 3D é apresentada a região descascada mostrando uma lesão cortical. Na parte inferior central é exibida o editor de função de transferência 1D. Todas as vistas são coordenadas entre si: operações numa são refletidas imediatamente nas outras. Com relação às interações, três funcionalidades são disponíveis: recortes, remoção de ruídos e medições.

---

<sup>16</sup>Query-Driven Visualization

<sup>17</sup>Visual Manipulation ToolKit



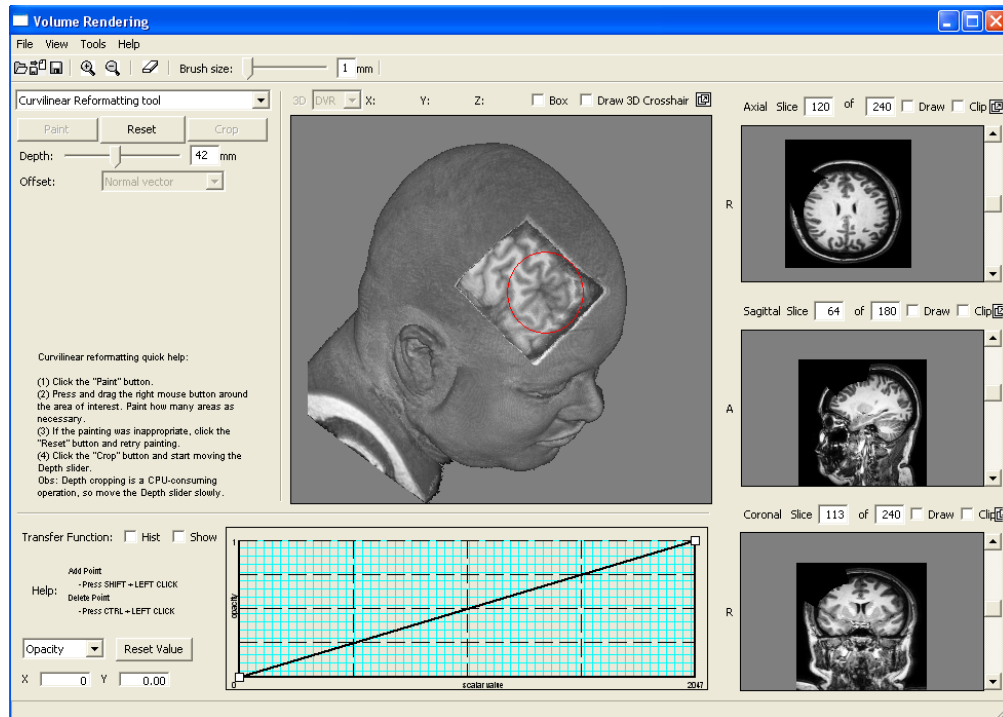


Figura 2.10: Interface do VMTK. Fonte: (Wu et al. 2011a)

Na figura 2.11 é apresentada uma visão geral da arquitetura do VMTK, a qual consiste de 4 módulos funcionais:

**Importação:** Realiza a leitura de imagens médicas 3D em formato DICOM<sup>18</sup>, um padrão de armazenamento de imagens médicas de uso clínico (NEMA 2012). Neste módulo, os dados são organizados num reticulado 3D de valores escalares. Além disso, este módulo permite salvar e recuperar sessões de trabalho de usuário.

**Mapeamento:** Os valores escalares do reticulado são mapeados de acordo com a função de transferência 1D ou alterados com os rótulos dos *voxels* presentes em um volume de controle. Rótulos implementados são: transparente, pintado, recortado, ponto de interesse, ponto de referência.

**Interação:** O sistema depende das ações do usuário para executar determinada tarefa. Este módulo captura a ação do usuário e, para retornar para ele uma resposta em tempo interativo, armazena os resultados em um segundo volume denominado o volume de controle.

**Renderização:** Os valores escalares, mapeados com a função de transferência e alterados pelos valores no volume de controle, são imageados, utilizando o algoritmo de *ray*

<sup>18</sup>Digital Imaging and Communications in Medicine

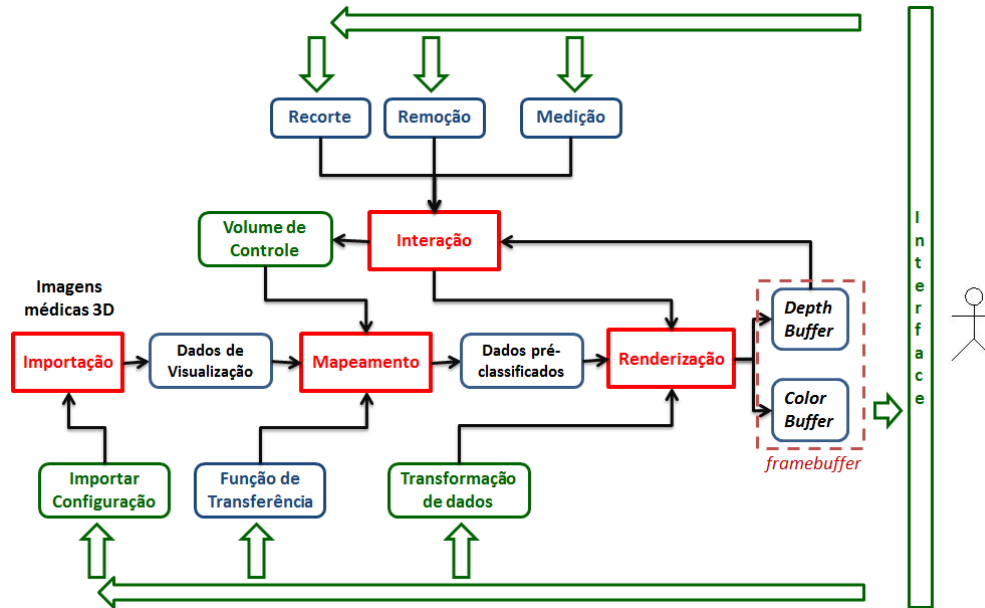


Figura 2.11: Arquitetura do VMTK. Fonte: (Wu et al. 2011a).

*casting*. Todos os *voxels* atravessados por um raio são processados, armazenando-se a cor resultante no *frame buffer*, enquanto a profundidade do primeiro *voxel* visível é armazenada no *buffer* de profundidade. Uma vez calculado o mapa de profundidade da cena, ele é retornado à CPU a fim de suportar as operações de interação de usuário. As operações de corte curvilíneo, medições e remoção de ruídos utilizam a técnica de mapeamento de 2D-para-3D (de coordenadas da tela para coordenadas do mundo) baseado no mapa de profundidade do volume de dados (Wu et al. 2011a).

Algumas características de implementação do VMTK:

**Linguagem C++.**

**API gráfica** OpenGL e GLSL versão 3.x.

**GDCM** biblioteca multiplataforma desenvolvida em C++ e de código livre para leitura, escrita e manipulação de imagens médicas DICOM (*GDCM, Grass roots DiCoM Version 2.0.18* 2011).

**wxWidgets** biblioteca em C++ apropriada para desenvolvimento de interfaces gráficas no ambiente Windows, MacOS, Linux e UNIX, de arquiteturas 32-bit e 64-bit, bem como em várias plataformas móveis, incluindo Windows Mobile, iPhone SDK e GTK+ embutido (*wxWidgets, Cross-Plataform GUI Library Version 2.8.12* 2011).

## 2.8 Conclusões

Neste capítulo foram introduzidos os conceitos de neuro-imagem, que tornou um poderoso meio não invasivo para estudo do cérebro. Na prática médica de diagnóstico e exploração de lesões cerebrais, destacam-se as neuro-imagens de ressonância magnética como imagens anatômicas de alta resolução espacial e alto contraste entre os tecidos moles da cabeça.

Pela complexidade da estrutura interna do cabeça e pela presença de órgãos vitais, o desenvolvimento de técnicas auxiliadas por computador para detecção de lesões cerebrais ainda estão no seus estágios iniciais, especialmente nos casos de lesões sutis no córtex cerebral.

A renderização volumétrica é um método efetivo para criar imagens 2D a partir de volume de dados 3D. Utilizando o algoritmo de *ray casting* baseado em GPU, uma neuro-imagem 3D pode ser visualizada em tempo interativo, sendo apropriado para exploração das imagens no espaço 3D. No entanto, a função de transferência é crítica para uma adequada visualização. Embora as funções de transferência multidimensionais forneçam melhor diferenciação das estruturas em imagens de tomografia computadorizada, elas não são de utilidade para visualização de imagens de ressonância magnética (Kniss et al. 2002b). Isso nos levou a propor um tratamento alternativo: ao invés de investir esforços em funções de transferência de elevadas dimensões e de pré-processamento mais custoso, preferimos adotar o modo de visualização *contexto+foco* para restringir interativamente as regiões de interesse baseada na forma de uma sonda esférica e uma lente circular. As regiões de foco e de contexto são renderizadas simultaneamente utilizando *ray casting* enquanto as propriedades ópticas são determinadas pelas funções de transferência 1D, do foco e do contexto, respectivamente.

Para desenvolvimento e implementação das técnicas, escolhemos o protótipo VMTK (seção 2.7) como plataforma de testes das técnicas desenvolvidas, pois esta plataforma dispõe de ferramentas de corte diferenciadas das ferramentas que temos conhecimento (Wu et al. 2011a), propiciando uma exploração diferenciada das amostras.

## Interação Exploratória

Com os avanços nas técnicas de visualização de dados volumétricos, tornou-se também possível que os dados sejam reconstruídos no seu espaço 3D nativo e visualizados na tela do computador. Para não perder o contexto tri-dimensional enquanto se analisa uma região de interesse, tornou-se necessário o uso de ferramentas de interação que permitem a um especialista manusear interativamente os atributos ópticos dos dados volumétricos em diferentes níveis de granularidade e sob distintos ângulos até encontrar achados anormais.

Tais interações fazem ainda mais sentido quando se trata de neuro-imagens 3D adquiridas a partir dos aparelhos de ressonância magnética, pois a complexidade da estrutura da cabeça e as diferenças anatômicas individuais relacionadas à área do cérebro dificultam o processamento e análise computacionais. Neste capítulo apresentamos o projeto de três ferramentas de interação que podem assistir um radiologista na exploração de neuro-imagens 3D em um ambiente convencional de interações: monitor com tela plana 2D, *mouse* e teclado.

Na seção 3.2 é apresentado o projeto do editor de funções de transferência 1D para atribuir propriedades ópticas aos *voxels* do volume de dados, sendo úteis na compreensão das estruturas do volume sem recorrer a algoritmos de segmentação que geralmente possuem um tempo de processamento elevado. Em seguida, apresentamos dois objetos de interação baseados no paradigma de visualização foco+contexto. Na seção 3.3 descreve-se o projeto da lente móvel com a qual podemos inspecionar áreas de interesse do volume de dados na sua resolução original e ampliá-las, mesmo em telas pequenas, o que pode propiciar descoberta de alterações anatômicas sutis. E na seção 3.4 detalha-se o projeto da sonda volumétrica como uma alternativa para selecionar interativamente regiões do volume em 3D e segmentá-la utilizando funções de transferência 1D. Veremos que este último projeto nasceu como resposta a uma série de insucessos na nossa tentativa de classificar automaticamente os tecidos cerebrais através dos valores escalares das neuro-imagens e das suas derivadas de primeira e segunda ordem.

## 3.1 Problemas a Resolver

Para atender às características desejadas das duas ferramentas ou objetos de interação, identificamos quatro problemas comuns:

1. como especificar os objetos de interação que selecionam regiões de interesse?
2. como movimentar espacialmente os objetos de interação?
3. como realçar os *voxels* da região de interesse?
4. como garantir que as tarefas sejam dinâmicas e interativas?

Para cada uma das questões expostas, apresentamos nossas soluções à luz dos recursos gráficos computacionais disponíveis. Estas soluções serviram de diretrizes para o desenvolvimento do nossas técnicas e ferramentas.

### 3.1.1 Especificação de um Objeto de Interação

Como as neuro-imagens 3D consistem de um arranjo de amostras, perguntamo-nos como devemos modelar os objetos de interação de forma que a sua integração com os dados volumétricos, a sua edição e a sua manipulação sejam simples, passíveis de serem realizadas em tempo interativo. Com base nas soluções de integração de uma geometria da cena com os dados volumétricos apresentadas em Engel et al. (2006), propomos que os objetos de interação sejam de geometrias convexas e analiticamente descritíveis com poucos parâmetros a fim de facilitar a sua representação e edição. Com isto, esperamos ter uma interface com um número mínimo de ações para controle das dimensões das regiões de interesse pelo usuário.

Selecionamos duas geometrias de interação que cumprem os requisitos acima expostos, a esfera e o círculo. Estas duas figuras geométricas podem ser definidas por dois parâmetros: o centro  $C = (x, y, z)$  e o raio  $r$ . A geometria da *lente móvel* que propomos é um disco circular através do qual especificamos a área de ampliação, enquanto a geometria da *sonda volumétrica* é uma esfera que nos permite destacar um sub-volume de interesse. Na figura 3.1 são ilustradas as geometrias utilizadas neste trabalho para interagir com o volume de dados.

### 3.1.2 Posicionamento Espacial dos Objetos de Interação

O posicionamento espacial dos objetos de interação propostos utilizando o *cursor* do *mouse* requer o mapeamento de coordenadas de um ponto da tela (2D) para coordenadas do mundo (3D) (Foley 1996). O uso de geometrias radialmente simétricas (seção 3.1.1) reduz o problema de posicionamento dos objetos de interação a um problema de *deslocamento* espacial controlado por *mouse* e teclado.

Em vista da funcionalidade dos dois objetos de interação, consideramos dois tipos de deslocamentos espaciais: restrito à superfície visível (Wu et al. 2011a) e movimentos decompostos em dois planos (Mesquita & Wu 2001).

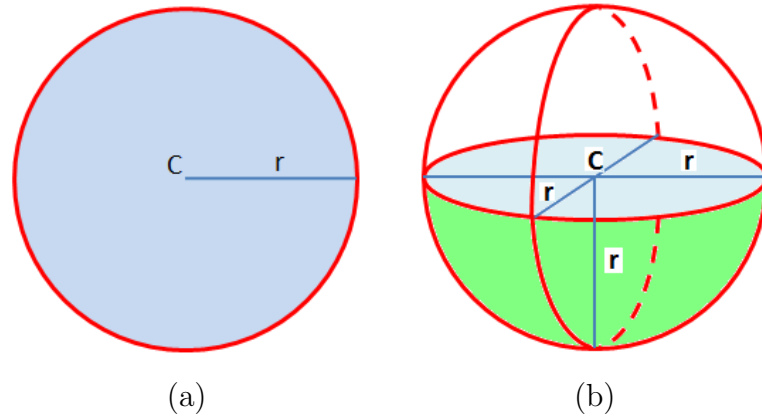


Figura 3.1: Objetos de interação: (a) disco circular; (b) esfera.

No caso da *lente móvel*, nossa proposta limita o posicionamento da geometria da lente sobre a superfície visível do volume de dados. Com base na proposta de Wu et al. (2011a), mapeamos a posição  $(x_m, y_m)$  do *cursor* do *mouse* em coordenadas 3D  $(x_{oc}, y_{oc}, z_{oc})$  de um ponto sobre a superfície visível do volume de dados. Posicionamos sobre este ponto o centro da lente CL. Com isso, a posição da lente é atualizada continuamente conforme o usuário arrasta o *cursor* do *mouse*.

A fim de deslocar livremente a *sonda volumétrica*, decompomos um movimento espacial 3D do seu centro C em movimentos em dois planos perpendiculares entre si (Mesquita & Wu 2001):

1. Movimentos normais do *cursor* sobre a tela do monitor correspondem a movimentos do centro C da sonda sobre o plano XY no espaço 3D.

Como as coordenadas do *cursor* do *mouse* correspondem ao espaço do dispositivo ( $DC^1$ ), enquanto a coordenada da sonda é dada em coordenadas do objeto ( $OC^2$ ), será necessário considerar o mapeamento do espaço do objeto para o espaço do dispositivo, ou vice-versa.

Sejam  $T_{DC \rightarrow OC}$  a transformação do espaço DC para OC,  $(x_{dc_{i+1}}, y_{dc_{i+1}})$  as coordenadas do *cursor* do *mouse* sobre a janela e  $S_{dc_i} = (x_{dc_i}, y_{dc_i}, z_{dc_i})$  as coordenadas do *cursor* do *mouse* na iteração anterior, então as coordenadas do centro correspondentes à nova posição do *cursor* do *mouse* são obtidas por

$$S_{oc_{i+1}} = T_{DC \rightarrow OC} \begin{bmatrix} x_{dc_{i+1}} \\ y_{dc_{i+1}} \\ z_{dc_i} \end{bmatrix}$$

<sup>1</sup>Device Coordinate – coordenada de dispositivo

<sup>2</sup>Object Coordinate – coordenada do objeto

2. O movimento em profundidade ao longo do eixo  $z$  é realizado mapeando-se o movimento do *cursor* do *mouse*,  $(x_{dc_{i+1}}, y_{dc_{i+1}})$ , sobre o plano  $XZ$  no espaço 3D.

Para esta interação, propomos utilizar a tecla “z” pressionada enquanto o usuário move o *cursor* do *mouse* sobre a janela.

Durante a interação, calculam-se as variações discretas dos deslocamentos do *cursor* do *mouse* sobre a tela:  $\Delta x_{dc_{i+1}} = x_{dc_{i+1}} - x_{dc_i}$  e  $\Delta y_{dc_{i+1}} = y_{dc_{i+1}} - y_{dc_i}$ . Com base nestas variações, obtivemos efeitos visuais plausíveis estimando a variação na profundidade  $\Delta z_{dc_{i+1}}$  de acordo com

$$\Delta z_{dc_{i+1}} = \frac{\sqrt{\Delta x_{dc_{i+1}}^2 + \Delta y_{dc_{i+1}}^2}}{4H},$$

sendo  $H$  a magnitude da linha diagonal imaginária sobre a janela de exibição. Assim a coordenada  $z_{dc_{i+1}}$  na iteração seguinte passa a ser

$$z_{dc_{i+1}} = z_{dc_i} \pm \Delta z_{dc_{i+1}},$$

onde o sinal, que indica incremento ou decremento, depende do sentido do movimento do *mouse*: se o *mouse* é arrastado horizontalmente da esquerda para direita ou verticalmente de cima para baixo, o sinal é +; do contrário, o sinal é -.

Tendo as coordenadas  $(x_{dc_{i+1}}, y_{dc_{i+1}}, z_{dc_{i+1}})$ , podemos obter a nova posição do centro em coordenadas  $OC$

$$S_{oc_{i+1}} = T_{DC \rightarrow OC} \begin{bmatrix} x_{dc_{i+1}} \\ y_{dc_{i+1}} \\ z_{dc_i} \end{bmatrix}$$

Com fins de compatibilidade e integração de nossas técnicas ao VMTK, atribuímos funções específicas aos botões do *mouse* e do teclado para interações 3D simultâneas com o volume de dados e com os objetos de interação propostos. O *mouse wheel* controla o *zoom* da cena, o botão esquerdo controla a rotação do volume de dados, o botão direito controla os movimentos da sonda e da lente em relação ao plano  $XY$  da janela. O movimento em profundidade, consegue-se mantendo pressionada a tecla  $z$  enquanto se movimenta o *mouse*.

### 3.1.3 Realce Interativo

Como diferentes composições espectrais podem produzir a mesma percepção no sistema visual humano, a simples correspondência de distintos valores de intensidade nas neuroimagens em diferentes cores não necessariamente produz sensações diferenciáveis em todos os indivíduos. Para contornar estas limitações naturais é fundamental prover mecanismos dinâmicos de ajustes na correspondência em um sistema de análise baseada em percepção de cores para que o que se vê seja condizente com os valores numéricos.

Vimos que as técnicas de renderização direta de volume (seção 2.4), no lugar de modelos ópticos analíticos, fazem uso da função de transferência para mapear os valores escalares do volume em propriedades ópticas. Sendo nossa proposta uma visualização foco+contexto utilizando funções de transferência (FT) 1D, temos duas questões a resolver: (1) com base nas geometrias de interação propostas, como determinar quais *voxels* correspondem ao foco e quais ao contexto? e (2) qual função de transferência utilizar?

A primeira questão é reduzida simplesmente à classificação de pertinência dos *voxels* do volume de dados em relação à geometria dos objetos de interação. O *disco circular* especifica uma região focal que consiste de uma superfície circular de *voxels* não transparentes mais próximos ao observador. Os *voxels* fora da região circular correspondem à região do contexto. Nossa abordagem é similar à proposta de Wang et al. (2005) na definição do foco, exceto pela característica de movimento da região focal, a ser detalhado na seção 3.3. A *esfera* permite selecionar uma região focal que consiste de um sub-volume de *voxels* no espaço 3D. Os *voxels* que não estão contidos dentro da geometria esférica pertencem à região do contexto. A abordagem adotada é similar à proposta por Zhou et al. (2002) e Weiskopf et al. (2003), que chamamos de *sonda volumétrica*, a ser exposto na seção 3.4).

Em relação à segunda questão, propomos utilizar duas funções de transferência para atribuição de propriedades ópticas dos *voxels* do volume após se estabelecer a região focal e contextual: uma para a região do foco e a outra para a região do contexto. Através da figura 3.2 pode-se perceber a diferença nos efeitos de contraste produzidos pela nossa proposta de realce localizado na região de interesse (duas funções de transferência) e os gerados com o realce global (uma função de transferência). Observa-se como funções de transferência 1D diferenciadas permitem realçar as regiões de interesse, facilitando a percepção de anormalidades em imagens de ressonância magnética.

### 3.1.4 Interatividade

Para assegurar a interatividade das técnicas propostas, propomos explorar os recursos gráficos e as capacidades de paralelismo disponíveis nas GPUs. Tendo como base o protótipo VMTK, descrito na seção 2.7, a ideia inicial era a extensão do algoritmo básico de *ray casting* para dar suporte às técnicas de seleção, de classificação e de atribuição de cores e opacidade aos *voxels* propostas neste trabalho.

No entanto, nos testes preliminares percebemos dois problemas: (1) acesso aleatório aos mapas de profundidade, tanto dos *voxels* visíveis quanto dos objetos de interação, compromete a interatividade do sistema; e (2) a regeneração integral das imagens em cada atualização do *cursor* do *mouse* pode degradar o desempenho do sistema em resoluções elevadas.

Após a análise dos novos requerimentos de mapas de profundidade da lente e da sonda, propomos a reestruturação da arquitetura de renderização volumétrica em dois passos como solução do primeiro problema. Tirando proveito dos objetos FBO<sup>3</sup> das GPU atuais (OpenGL 2010), geramos

---

<sup>3</sup>Frame Buffer Object



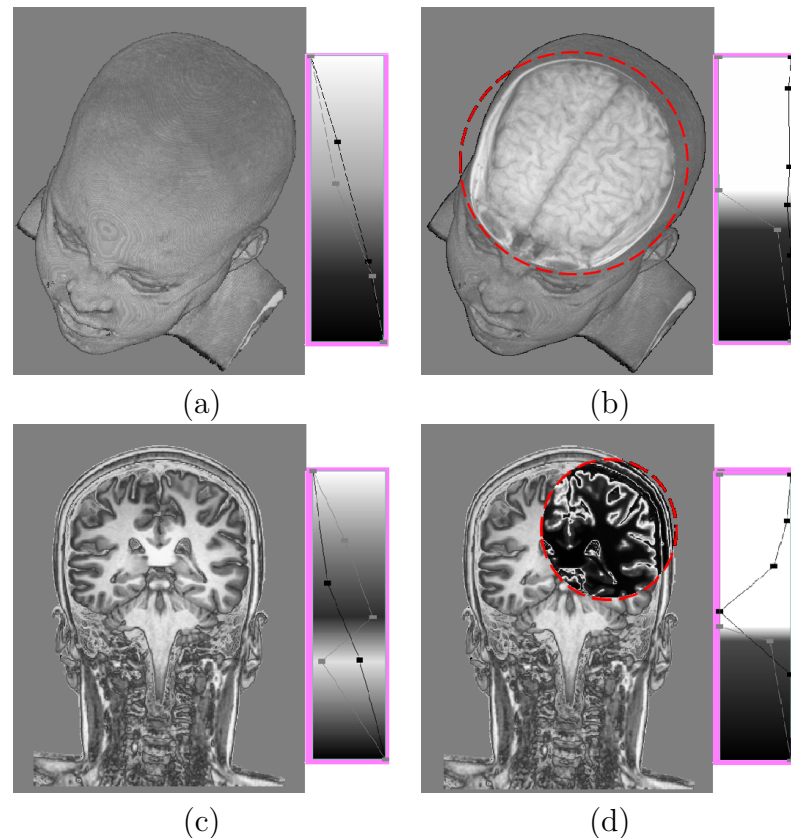


Figura 3.2: Visualização foco+contexto: (a) e (c) realce global do volume utilizando uma FT; (b) realce localizado utilizando a sonda e FT do foco; e (d) realce localizado utilizando a lente e FT do foco.

1. no *modo off-screen* ou *fora da tela* os mapas de profundidade do volume de dados e dos objetos de interação, lente e sonda. Para resolver nosso requerimento de vários *depth buffers*, os mapas de profundidade calculados (um para o volume, um para a lente, e dois para a sonda, totais 04) são escritos em *color buffers* representados por texturas 2D ligadas a um objeto FBO. Observa-se que o sistema sempre gera antecipadamente os mapas de profundidade em cada regeneração, preparando-o para processamento dos eventos gerados pelo usuário; e
2. no *modo normal* ou visível as imagens do volume de dados integrado com os objetos de interação a fim de criar uma imagem 2D exibível na tela do monitor. Este modo disponibiliza as interações entre o sistema gráfico e o usuário.

Vale a pena comentar que no *modo normal*, o volume de dados é processado utilizando o *shader* de *ray casting*, enquanto as geometrias de interação utilizam os recursos do *pipeline* gráfico de funcionalidade fixa para serem imageadas.

## 3.2 Editor de Funções de Transferência 1D

Tanto para realce global do volume de dados como para realce localizado de regiões de interesse, propomos utilizar duas funções de transferência 1D. Conjeturamos que um editor interativo de função de transferência baseado em pontos de controle é suficiente para realçar regiões de interesse selecionadas pelos objetos de interação fornecidos, desde que o usuário tenha controle interativo no que deseja ver. Lesões sutis que dificilmente seriam visíveis poderiam ser descobertas pelo especialista através das variações interativas das propriedades ópticas dos *voxels* selecionados.

A função transferência 1D estabelece uma correspondência entre os valores escalares dos *voxels* do volume e as propriedades ópticas de cor e opacidade, representadas em RGBA (seção 2.4.2). Nosso editor de função de transferência fornece dois ambientes para atribuição de propriedades ópticas:

1. colorida, que permite a edição dos 4 canais de cor RGBA; e
2. tons de cinza, que permite a edição de 2 canais de cor, I (intensidade) e A (opacidade), quando se assume que  $R=G=B=I$ .

A atribuição de propriedades ópticas em tons de cinza é o ambiente preferido pelos especialistas da área médica, por terem sido treinados na análise de imagens médicas monocromáticas durante a sua formação. Utilizando o editor de função de transferência, o usuário insere ou remove pontos de controle livremente. Enquanto ele vai arrastando um ponto de controle, altera-se a função de mapeamento linear por partes. E a atualização das imagens ocorre em tempo interativo. Ressaltamos que uma característica peculiar do nosso editor é que a função de transferência editada não é necessariamente monotônica. Isso permite melhorar localmente os contrastes entre as regiões de interesse, apesar das possíveis ambiguidades na correspondência entre os valores escalares e as propriedades ópticas.

## 3.3 Lente Móvel

A *lente* é um “objeto transparente de vidro” localizado entre o objeto de interesse a inspecionar e o usuário. As lentes incorporam filtros visuais para alterar a representação dos objetos examinados, revelar informação escondida ou remover dados redundantes. Elas melhoram a qualidade da percepção visual dos objetos, permitindo investigar seus detalhes sem perda de informação contextual.

Nossa *lente móvel* permite ao usuário interativamente selecionar, ampliar e explorar áreas de interesse, as quais podem ser exibidas somente com as propriedades ópticas dos *voxels* visíveis e mais próximos ao observador. Ao exibir os *voxels* sob a lente com suas informações originais, contornamos um dos problemas que percebemos na técnica de composição adotada no processo de renderização direta do volume (seção 2.4): atenuação dos detalhes nas regiões de interesse. Ao combinarmos o modo de composição com o modo de exibição por camada,

podemos aumentar a fidelidade dos dados nas regiões de interesse sem perda da percepção 3D que a técnica de composição nos proporciona. Isto, somado ao fato de ser móvel, faz da ferramenta um meio de investigação e auxílio ao especialista no diagnóstico de lesões sutis em neuro-imagens.

### 3.3.1 Geometria

Nossa lente consiste de um disco circular, de acordo com a explicação dada na seção 3.1.1, e o seu centro está posicionado sobre a superfície visível do volume de dados, conforme mostra a seção 3.1.2). Além dos dois parâmetros geométricos, o centro  $CL = (x_{CL}, y_{CL}, z_{CL})$  em coordenadas do dispositivo e o raio  $RL$ , incluímos um parâmetro óptico para controlar o fator de ampliação: o ponto focal  $PF$ . A lente é sempre paralela ao plano de imagem passando pelo centro  $CL$ . Na visualização foco+contexto, a região circular da lente corresponde ao foco e a restante ao contexto. Uma vista lateral da lente, em cor amarela, é apresentada na figura 3.3.

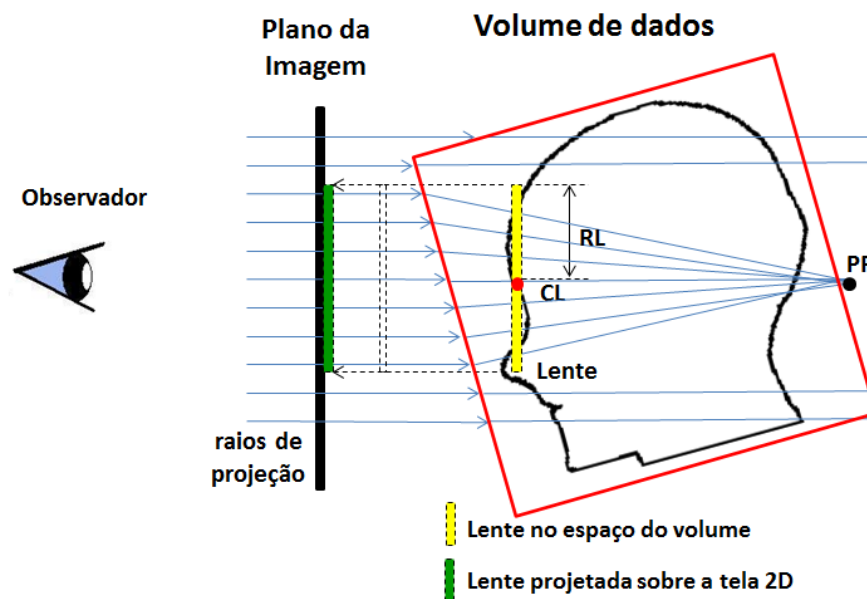


Figura 3.3: Ilustração da lente móvel.

Para integrar a lente no fluxo de renderização baseado em *ray casting* utilizando GPU de maneira similar à proposta de Wang et al. (2005), projetamos nossa lente sobre a tela, como mostra a figura 3.3, a fim de determinar os *pixels* que correspondem à região da lente.

### 3.3.2 Modelo

Todos os raios disparados que incidem sobre os *pixels* da região da lente sofrem deflexão ao cruzar com o envoltório do volume de dados e convergem no ponto focal  $PF$ , formando

um cone de raios. Quando o raio cruza o primeiro *voxel* não transparente, o percurso é interrompido e a propriedade óptica daquele *voxel* é exibido no *pixel* de onde partiu o raio.

Segundo a óptica geométrica (Szeliski 2012), as estruturas e objetos contidos dentro do cone são ampliados em função da posição de  $PF$ , enquanto os outros objetos fora do cone são projetados paralelamente. Como a projeção das amostras do volume dentro do cone é perspectiva, o fator de ampliação  $FA$  da lente

$$FA = \frac{\text{tamanho da imagem}}{\text{tamanho do objeto}} \quad (3.1)$$

pode ser obtido através do deslocamento do ponto focal  $PF$  ao longo do eixo óptico.

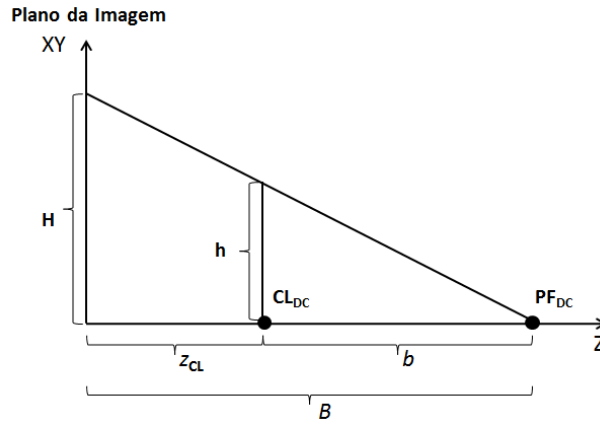


Figura 3.4: Projeção perspectiva em relação à lente móvel.

A figura 3.4 mostra a projeção perspectiva de um ponto a uma altura  $h$  em relação ao eixo óptico perpendicular ao plano de imagem com o centro da lente localizado a uma distância  $z_{CL}$  em relação ao plano de imagem. Dado um fator de ampliação  $FA$ , podemos obter pela semelhança de triângulos a distância  $B$  de  $PF$  em relação ao plano de imagem.

$$\begin{aligned} B &= \frac{bH}{h} = \frac{(B - z_{CL})H}{h} \\ &= (B - z_{CL})FA \\ (1 - FA)B &= -z_{CL}FA \\ B &= -z_{CL} \frac{FA}{1 - FA}, \quad FA \neq 1.0 \\ &= z_{CL} \left(1 - \frac{1}{1 - FA}\right) \\ &= z_{CL} + \frac{z_{CL}}{FA - 1}. \end{aligned} \quad (3.2)$$

Esta distância corresponde exatamente à coordenada  $z$  do ponto focal no espaço do dispositivo e para obter as coordenadas de  $PF$  no espaço do objeto, podemos aplicar a transformação definida na seção 3.1.2 ao ponto  $(x_{CL}, z_{CL}, B)$ .

### 3.3.3 Controle dos Movimentos

Como já foi explicado na seção 3.1.2 sobre posicionamento das geometrias de interação, o deslocamento da lente é controlado mantendo pressionado o botão direito do *mouse* e arrastado-o na direção dos eixos  $x$  e  $y$  do plano da imagem.

O raio  $RL$  da lente, configurado pelo usuário, define o tamanho da região focal. O centro da lente  $CL$  é atualizado em cada deslocamento do *cursor*. A posição do ponto focal  $PF$  é calculada automaticamente conforme o fator de ampliação  $FA$  especificado. Conseqüentemente, o usuário tem a percepção de que a lente se desliza suavemente sobre a superfície do volume, ampliando o conteúdo da região focal do objeto de interesse.

Vale enfatizar que o eixo óptico é sempre perpendicular ao plano da imagem e passa pelo centro da lente. E o movimento do centro da lente é sempre restrito à superfície visível do volume de dados, sendo possível girar o volume com o objetivo de inspecionar as superfícies que estiverem ocultas. Quando isto acontece, os valores dos parâmetros  $CL$  e  $PF$  são atualizados de acordo com a nova superfície visível do volume. Como consequência, tem-se a percepção de que a lente desliza suavemente sobre a superfície visível.

Na figura 3.5 apresentam-se imagens produzidas durante a interação com a lente móvel calibrada em distintos fatores de ampliação: 1.0 (sem aumento), 2.5 e 3.5, respectivamente. Vale observar a diferença entre a região focal e o contexto em termos de contraste.

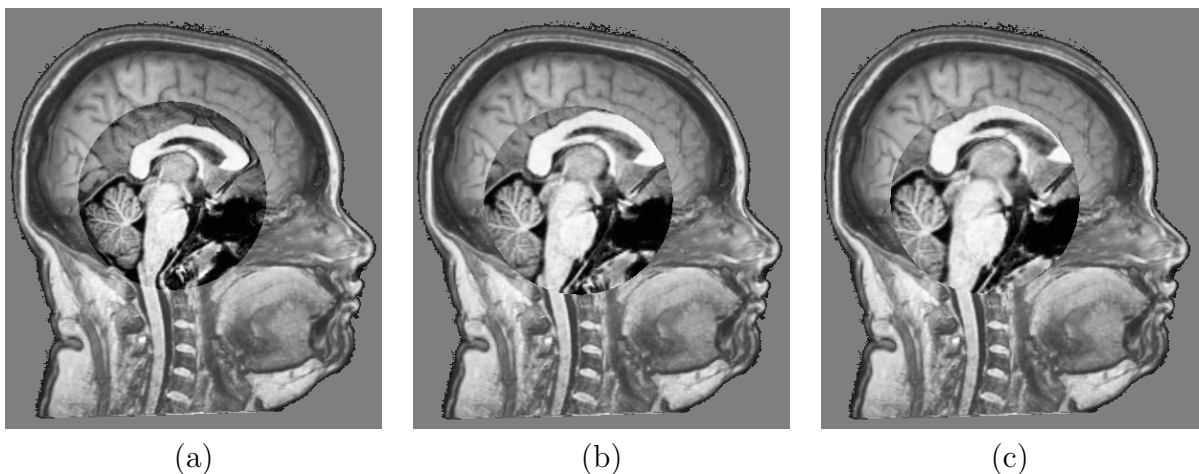


Figura 3.5: Lente móvel sobre corte sagital de uma imagem de RM 3D: (a)  $FA = 1.0$  (sem aumento), (b)  $FA = 2.5$ , e (c)  $FA = 3.5$ .

### 3.3.4 Integração com Dados Volumétricos

A integração da geometria da lente aos dados volumétricos ocorre implicitamente quando tratamos de forma diferenciada os raios que partem e que não partem da região da lente. Os que partem da região da lente são convergentes e os que não partem da região da lente são paralelos. Para isso, é necessário conhecer quais *pixels* correspondem à lente para poder realizar um tratamento diferenciado dos raios convergentes, conforme ilustra a figura 3.3.

Ao invés de projetar a lente sobre o plano de imagem, calculamos no modo *off-screen* a profundidade dos fragmentos da geometria da lente e armazenamo-la em uma textura 2D, com dimensões similares às da janela atual de visualização. Como os fragmentos que tiverem profundidade diferente de 1.0 correspondem aos *pixels* sobre os quais a lente projetaria, utilizamos este mapa para diferenciar os raios paralelos e os raios convergentes.

Na renderização volumétrica normal com *ray casting*, determina-se se o raio começa da região focal, ou não, com base na informação da textura 2D. Somente os raios que partem da região da lente sofrem deflexões a fim de convergir no ponto focal  $PF$ . Cada raio convergente em  $PF$  é amostrado em intervalos regulares até atingir o primeiro *voxel* visível ao observador. Neste caso, o laço do raio acaba e as propriedades ópticas daquele *voxel*, que nós chamamos de “informação original”, são utilizadas para exibí-los no plano da imagem. Alternativamente, fornecemos a possibilidade de o raio atravessar todo o volume. Neste caso, a cor final no *pixel* corresponde à composição das cores de todos os *voxels* amostrados ao longo do raio projetor.

Observa-se que a atribuição de cores aos *voxels* está baseada na função de transferência. Em nossa proposta, é possível utilizar uma função de transferência global para todo o volume de dados, ou estabelecer uma função de transferência local para realçar diferenciadamente a região sob investigação. É importante destacar que não foram incluídos tratamentos especiais para compensar a deformação gerada na transição entre a região focal e contextual, uma vez que esta deformação é pouco perceptível quando o usuário movimentava continuamente a lente.

### 3.3.5 Implementação em GPU

Assim como foi discutido na seção 3.1.4 sobre interatividade, a especificação da geometria da lente e sua integração com os dados volumétricos são realizados em dois passos:

1. Ativando o teste de profundidade *depth test* do OpenGL (OpenGL 2010), em modo *off-screen*, renderiza-se o disco circular para calcular seu mapa de profundidade em uma textura 2D de igual tamanho à resolução da tela. Em decorrência disso, os *texels* com valores diferentes de 1.0 correspondem à região da lente no plano da imagem.
2. no modo normal é gerado a imagem do volume de dados aplicando a projeção perspectiva para os fragmentos contidos na área circular da lente e a projeção paralela para o restante dos fragmentos. O teste de pertinência dos fragmentos é conduzido com uso da textura de profundidade gerada no primeiro passo. Para isso, precisamos passar adicionalmente para o *shader* de *ray casting* os seguintes parâmetros:
  - textura 2D do mapa de profundidade da lente,
  - o ponto focal  $PF$ , e
  - função de transferência 1D da região focal.

O algoritmo modificado de *ray casting* que suporta a lente móvel é apresentado no algoritmo 3.1. Inicializa-se *shader* zerando as variáveis de destino de cor (Alg. 3.1, linha 1) e

posição de entrada do raio no volume de dados (Alg. 3.1, linha 2). Logo é lido o valor de profundidade da lente (Alg. 3.1, linha 3). Se a profundidade for diferente de 1.0 (Alg. 3.1, linha 4) então o raio incidente está sobre a região da lente. Portanto, uma nova direção de raio convergente em *PontoFocal* é calculada para percorrer o volume (Alg. 3.1, da linhas 5–14). Caso contrário, realiza-se o percurso na direção perpendicular ao plano de imagem (Alg. 3.1, linhas 15–17).

O percurso dos raios convergentes ocorre em um laço (Alg. 3.1, linhas 6–14). Dentro do laço, avança-se a passo *tamanhoDePasso* na direção do raio, estima-se o valor escalar no ponto com base nos valores escalares dos *voxels* na vizinhança, e a partir deste valor acessam-se as cores correspondentes (linhas 7 e 8). E assim sucessivamente até atingir o primeiro *voxel* não transparente (Alg. 3.1, linhas 9–12). Neste caso, retorna-se a cor daquele *voxel* (Alg. 3.1, linha 10) e interrompe o percurso (Alg. 3.1, linha 11).

---

**Algoritmo 3.1** Lente móvel
 

---

**Require:** *VolumeDeDados*

**Require:** *FuncaoDeTransferenciaFoco*

**Require:** *FuncaoDeTransferenciaContexto*

**Require:** *RegiaoLente*

**Require:** *tamanhoDePasso*

**Require:** *PontoFocal*

```

1:  $cor_{dest}[r][g][b][a] \leftarrow (0, 0, 0, 0)$ 
2:  $posicao(x, y, z) \leftarrow$  coordenadas da textura 3D do fragmento
3:  $profundidade = RegiaoLente[x_{fragmento}][y_{fragmento}]$ 
4: if profundidade  $\neq$  1.0 then
5:    $lentedirecao \leftarrow normalizado(PontoFocal - posicao)$ 
6:   while  $posicao(x, y, z) \in VolumeDeDados$  do
7:      $escalar \leftarrow VolumeDeDados(posicao)$ 
8:      $cor_{src} \leftarrow FuncaoDeTransferenciaFoco(escalar)$ 
9:     if  $cor_{src}[a]$  nao-e-transparente then
10:       $cor_{dest} \leftarrow cor_{src}$ 
11:      return
12:     end if
13:      $posicao = posicao + lentedirecao * tamanhoDePasso$ 
14:   end while
15: else
16:   composição descrita no algoritmo 2.1 aplicando FuncaoDeTransferenciaContexto
17: end if
18: return  $cor_{dest}$ 

```

---

## 3.4 Sonda Volumétrica

Em geral, a *sonda* é um objeto de manipulação com objetivo de reportar informação sobre o alvo examinado. Na área médica, ela é considerada como um dispositivo para exploração de feridas, cavidades, tecidos ou órgãos.

Consideramos neste trabalho uma *sonda volumétrica* o objeto de manipulação direta para selecionar regiões volumétricas específicas no espaço 3D. Ela permite uma exploração de dados em modo foco+contexto: regiões focais podem ser realçadas de forma a melhorar a sua qualidade e regiões irrelevantes podem se tornar transparentes. Para tal efeito, fornecemos uma interação com duas funções de transferência, uma para a região selecionada e outra para a região circundante. Esta abordagem melhora a orientação do usuário, pois permite uma exploração dos detalhes sem perda de informação contextual.

### 3.4.1 Geometria

A geometria da sonda volumétrica é uma esfera aproximada por uma malha triangular, cuja definição é dada na seção 3.1.1. O seu centro é posicionado livremente no espaço 3D, utilizando movimentos combinados do *mouse* e teclado, conforme detalhado na seção 3.1.2. Sempre que o usuário arrasta o *mouse* enquanto pressiona o botão direito, a posição da esfera é atualizada. Isso permite varrer todo o volume mantendo-o fixo.

Conforme vimos anteriormente, dois parâmetros são necessários para especificar a sonda: seu centro  $C = (x, y, z)$  no espaço 3D e o raio  $r$ . Uma vista da sonda em cor amarela é ilustrada na figura 3.6. Nota-se que a região esférica da sonda (cor verde) corresponde ao foco, e a região circundante ao contexto (cor azul).

### 3.4.2 Modelo

Para selecionar os *voxels* da região de interesse, utilizamos um procedimento similar ao proposto por Zhou et al. (2002):

- em modo *off-screen*, calculamos o mapa de profundidade das faces da geometria de sonda: face frontal `MapaFaceFrontal` e face oculta `MapaFaceOculta`.
- baseado nos mapas de profundidades das faces da sonda, cada raio disparado é dividido em três intervalos durante a renderização em modo normal:
  1. antes do valor de `MapaFaceFrontal` (semi-esfera de cor vermelha na figura 3.6),
  2. após o valor de `MapaFaceOculta` (semi-esfera de cor rosa na figura 3.6), e
  3. entre os dois valores de profundidade.

Aplicamos a função de transferência do foco para todos os *voxels* do último intervalo, enquanto para o restante aplicamos a função de transferência do contexto, sendo a



cor em cada *pixel* o resultado da composição das propriedades ópticas obtidas a longo de cada raio. Optamos pela composição, em detrimento de maior contraste, porque acreditamos que na fase exploratória pode ser mais interessante visualizar a disposição espacial da sonda em relação ao volume de dados, mesmo que esteja no interior deste.

Observa-se que, na renderização volumétrica, a profundidade de cada *voxel* é avaliada em relação aos valores nos mapas de profundidade da sonda. Um *voxel* pertence à região focal, se a sua profundidade  $z_{voxel}$  estiver contida entre a face frontal e a face oculta da sonda, ou seja,

$$(z_{voxel} \geq z_{FaceFrontal}) \wedge (z_{voxel} \leq z_{FaceOculta}) \quad (3.3)$$

Em nossa proposta, a renderização da região focal e contextual é feito em um único passo, enquanto o raio atravessa o volume. A figura 3.6 apresenta uma vista que ilustra a classificação dos *voxels* em regiões de foco e contexto.

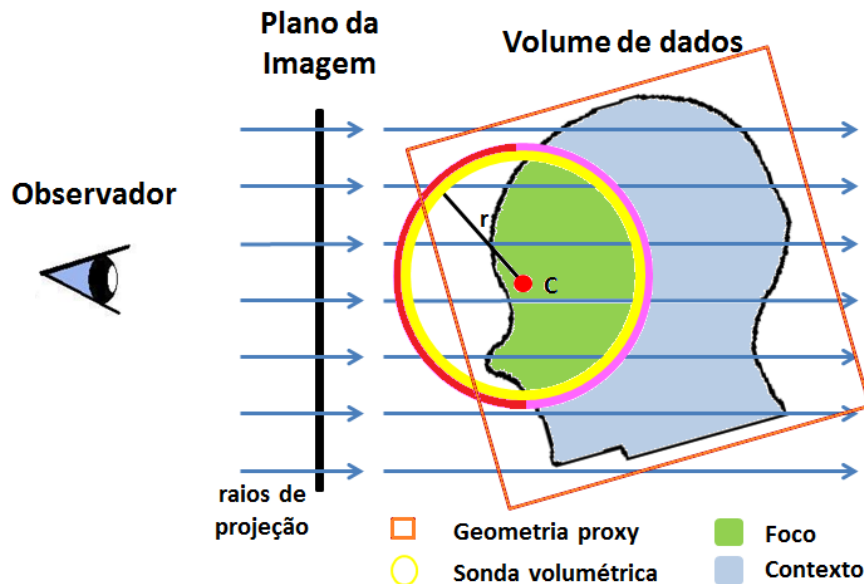


Figura 3.6: Seleção da região focal e contextual utilizando a sonda volumétrica.

Após a seleção da região de interesse, ela pode ser explorada e realçada via uma função de transferência 1D distinta, com o objetivo de fornecer mais informação sobre a região de interesse.

### 3.4.3 Controle dos Movimentos

Como já foi explicado na seção 3.1.2, os deslocamentos do centro da sonda são relativos ao plano  $XY$  e  $XZ$  da janela de visualização, controlados pelo *mouse* mantendo, ou não, pressionada a tecla “z”. Assim, o usuário tem a percepção de que a sonda segue o movimento do *cursor* do *mouse*, deslocando-se suave e livremente no espaço 3D.

### 3.4.4 Integração com Dados Volumétricos

Baseado na proposta de Engel et al. (2006) para integração de geometrias com os dados volumétricos, calculamos em modo *off-screen* os dois mapas de profundidade das faces da sonda utilizando a funcionalidade fixa do *pipeline* gráfico. Em seguida, em modo normal de renderização do volume, as inequações 3.3 são aplicadas para decidir qual função de transferência a ser aplicada para cada *voxel*: a do foco ou a de contexto. A figura 3.7(c) ilustra o resultado desta estratégia de integração.

Para visualizar a geometria da sonda volumétrica, como na figura 3.7(b), imageamos em primeiro lugar o volume de dados e determinamos os valores de profundidade de cada *voxel* visível, e depois é que imageamos a geometria da malha com o teste de profundidade habilitado. Na figura 3.7 são mostradas três situações distintas de interação com a sonda volumétrica.

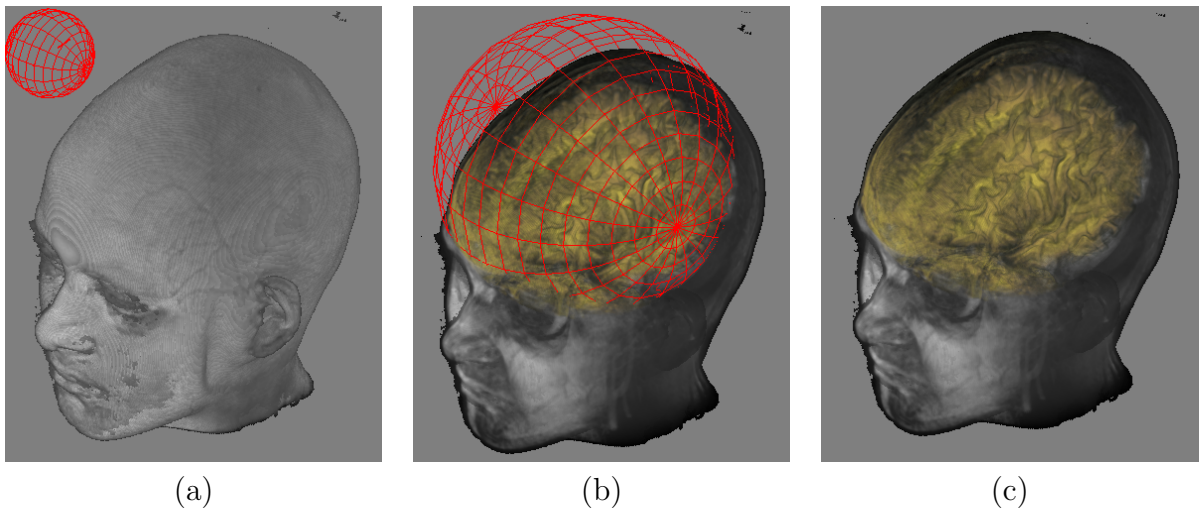


Figura 3.7: Sonda volumétrica em relação a um volume de dados: (a) livre no espaço 3D, região selecionada (b) com a geometria de sonda visível, e (c) com a geometria desabilitada.

### 3.4.5 Implementação em GPU

Conforme foi explicado na seção 3.1.4 de interatividade, a renderização da sonda volumétrica e sua integração com os dados volumétricos são realizados em dois passos:

1. ativando o teste de profundidade *depth test* do OpenGL (OpenGL 2010), em modo *off-screen*, a geometria da sonda é renderizada duas vezes a fim de calcular os mapas de profundidade da face frontal e da oclusa da sonda. Estas informações são salvas em texturas 2D de dimensões igual tamanho à resolução da janela de visualização.
2. no modo normal é gerado a imagem do volume de dados aplicando a inequação 3.3 para decidir a função de transferência a ser aplicada. Para isso, precisamos passar adicionalmente para o *shader* de *ray casting* os seguintes dados:

- duas texturas 2D contendo os mapas de profundidade das faces da sonda,
- função de transferência 1D da região focal.

O algoritmo modificado de *ray casting* que suporta a sonda volumétrica é apresentado no algoritmo 3.2. Inicializa-se o *shader* zerando a variável de destino de cor (Alg. 3.2 linha 1) e calcula-se a posição de entrada do raio (Alg. 3.2 linha 2). Em seguida, são acessados os valores de profundidade das faces da sonda (Alg. 3.2 linha 3 e 4). Enquanto a posição  $(x, y, z)$  estiver dentro do volume de dados (linas 5–15), interpola-se o valor escalar na posição  $(x, y, z)$  com base nos valores escalares das amostras na vizinhança (Alg. 3.2 linha 6) e calcula-se a profundidade daquela amostra (Alg. 3.2 linha 7). Baseado no valor  $z$  da amostra, avalia-se se ela pertence à região focal ou à região contextual (Alg. 3.2 linha 8–12), atribuindo à amostra valores de cor e opacidade conforme a função de transferência correspondente (Alg. 3.2 linha 9 ou linha 11). A seguir, efetua-se a composição de cores (Alg. 3.2 linha 13) e avança-se para a próxima amostra do volume (Alg. 3.2 linha 14). Uma vez finalizado o laço de percurso do raio, a cor resultante é exibida no *pixel* que originou aquele raio (linha 16).

---

**Algoritmo 3.2** Sonda Volumétrica
 

---

**Require:** *VolumeDeDados*

**Require:** *MapaFaceFrontal*

**Require:** *MapaFaceOculto*

**Require:** *FuncaoDeTransferenciaFoco*

**Require:** *FuncaoDeTransferenciaContexto*

**Require:** *direcaoRaio*

**Require:** *tamanhoDePasso*

```

1:  $cor_{dest}[r][g][b][a] \leftarrow (0, 0, 0, 0)$ 
2:  $posicao(x, y, z) \leftarrow$  coordenadas da textura 3D do fragmento
3:  $z_{Frontal} \leftarrow$  MapaFaceFrontal $[x_{fragmento}][y_{fragmento}]$ 
4:  $z_{Oculto} \leftarrow$  MapaFaceOculto $[x_{fragmento}][y_{fragmento}]$ 
5: while  $posicao(x, y, z) \in$  VolumeDeDados do
6:    $escalar \leftarrow$  VolumeDeDados( $posicao$ )
7:    $z_{Voxel} \leftarrow$  calculaProfundidadeDoEscalar( $posicao$ )
8:   if  $(z_{Voxel} \geq z_{Frontal})$  and  $(z_{Voxel} \leq z_{Oculto})$  then
9:      $cor_{src} \leftarrow$  FuncaoDeTransferenciaFoco( $escalar$ )
10:  else
11:     $cor_{src} \leftarrow$  FuncaoDeTransferenciaContexto( $escalar$ )
12:  end if
13:   $cor_{dest} =$  composição ( $cor_{dest}, cor_{src}$ )
14:   $posicao = posicao + direcaoRaio * tamanhoDePasso$ 
15: end while
16: return  $cor_{dest}$ 

```

---

Para incluir a geometria da sonda na imagem, exploramos o recurso de *blending* disponível no fluxo fixo de OpenGL (OpenGL 2010). Atribuímos à cada amostra das arestas um fator

de opacidade que varia conforme a sua distância em relação à superfície visível do volume de dados. Quanto mais afastada estiver a amostra em relação à superfície, menor será o seu fator de opacidade. Em seguida, combinamos linearmente o resultado da sua cor com a cor da imagem do volume de dados gerada pelo algoritmo 3.2. Figura 3.8 ilustra os efeitos visuais desta proposta. A fim de proporcionar uma melhor percepção das diferenças na cor das linhas da sonda, adotamos a mesma função de transferência para a região focal e a contextual.

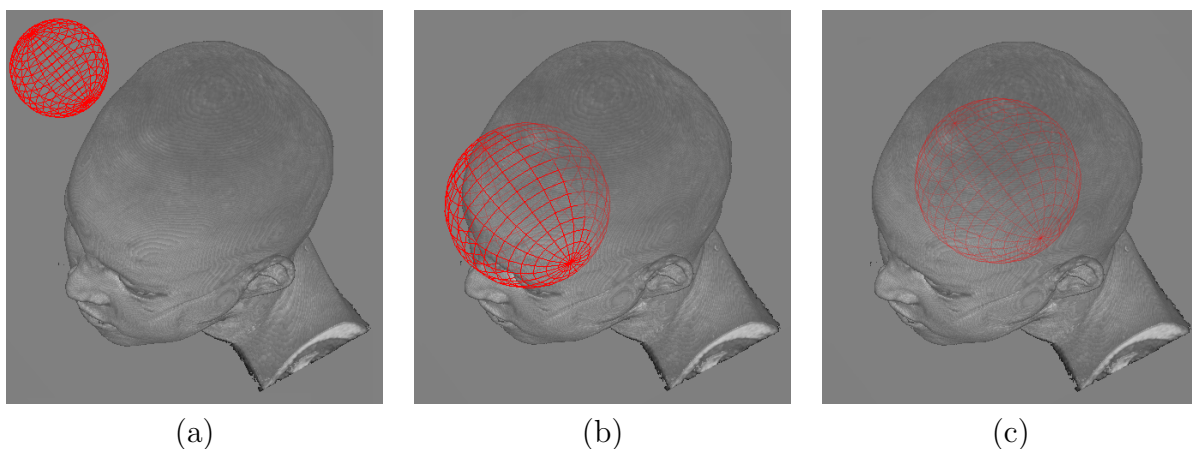


Figura 3.8: Disposição espacial da sonda volumétrica em relação a um volume de dados: (a) fora do volume (b) parcialmente fora do volume, e (c) dentro do volume.

### 3.5 Conclusões

Neste capítulo foram apresentadas três técnicas para exploração interativa de anomalias cerebrais em neuro-imagens de ressonância magnética:

1. o *editor função de transferência* para realçar de forma diferenciada os contrastes dos tecidos selecionados, e não selecionados, em neuro-imagens, a fim de facilitar a percepção de variações estruturais como lesões sutis no cérebro.
2. a *lente móvel* para inspecionar e ampliar áreas do volume preservando sua informação original, e
3. a *sonda volumétrica* para destacar sub-volumes de dados sem perda do seu contexto.

O diferencial de nossa proposta está na resposta interativa das ferramentas, permitindo que os usuários explorem e experimentem suas hipóteses de forma contínua, sem longos tempos de espera.

Sob o ponto da arquitetura do sistema, foi proposto um modelo de renderização em dois passos, que aproveita a capacidade de renderização *off-screen* das GPUs para recuperar as informações de profundidade necessárias para interações.

---

Sob o ponto de vista da interface gráfica, foi redesenhada a interface gráfica de usuário do VMTK (seção 2.7) com o objetivo de integrar as novas técnicas. Procurou-se aumentar o espaço de visualização e melhorar a usabilidade. Todos os resultados obtidos são apresentados no Capítulo 4.

## Resultados

Nossas técnicas foram implementadas e integradas ao protótipo VMTK (seção 2.7). Tais técnicas foram submetidas a três classes de testes para sua validação, sendo os resultados apresentados neste capítulo:

1. testes de qualidade das imagens produzidas em termos de realce de regiões de interesse e percepção de lesões sutis;
2. testes de desempenho em termos de interatividade das ferramentas; e
3. testes de usabilidade em termos de eficiência em assistir aos especialistas em identificar achados radiológicos nas neuro-imagens.

Nos testes foram utilizadas as neuro-imagens de ressonância magnética, de modalidade T1, de dimensões  $240 \times 240 \times 180 \times 12$  bits, fornecidas pelo Laboratório de Neuro-Imagem da Faculdade de Ciências Médicas da Unicamp (LNI-UNICAMP 2012).

### 4.1 Ambiente de Desenvolvimento

As ferramentas propostas, editor de função de transferência, sonda volumétrica e lente móvel, foram incorporadas ao protótipo VMTK com a finalidade de obtermos um ambiente de exploração mais rico, reutilizando as funcionalidades de *recorte*, *medição* e *remoção de ruídos* já disponíveis.

Nesta seção descrevemos a plataforma de desenvolvimento, as suas principais funcionalidades e a integração das propostas apresentadas na seção 3.

#### 4.1.1 Infra-estrutura

O recurso de *hardware* utilizado para o desenvolvimento foi um computador Intel<sup>®</sup>, processador Core2 Duo E7500 de 2.936 GHz, 2GB RAM, GPU NVIDIA<sup>®</sup> GeForce GT240 de

1GB VRAM, com suporte a OpenGL 3.1 (OpenGL 2010). Os códigos foram implementados em C++ e testados nos sistemas operacionais de Microsoft Windows e Linux Ubuntu. Os componentes da biblioteca WxWidgets (*wxWidgets, Cross-Platform GUI Library Version 2.8.12* 2011) foram utilizados para implementar a interface gráfica, cujo desenho foi confeccionado com o auxílio do aplicativo wxFormBuilder (*wxFormBuilder, An Open-Source GUI Builder for wxWidgets Version 3.1.70* 2010). As funções da biblioteca GDCM (*GDCM, Grass roots DiCoM Version 2.0.18* 2011) foram utilizadas para importar os volumes de dados médicos no formato digital DICOM (NEMA 2012).

### 4.1.2 Integração das Ferramentas no VMTK

Para integrar as técnicas propostas foram realizadas as seguintes alterações no fluxo de trabalho:

**Mapeamento:** Neste módulo são processadas duas funções de transferência 1D, uma para a região selecionada (foco) e outra para a região não selecionada (contexto).

**Interação:** Preservando a modularidade do VMTK, foi adicionado um novo ambiente de interação para tratamento de eventos da sonda e da lente. Uma nova interface gráfica de edição de função de transferência foi implementada para atender às demandas por visualização foco+contexto.

**Renderização:** O processo de renderização volumétrica é desdobrado em dois passos (no modo *off-screen* e no modo normal), de forma que seja possível visualizar a geometria do objeto de interação integrada ao volume de dados.

Uma visão global da arquitetura do sistema com as novas técnicas integradas é mostrada na figura 4.1. Foram alterados os códigos dos módulos de Renderização e de Interação da versão original do VMTK (seção 2.7).

#### Interface

A interface do VMTK foi redesenhada com o objetivo de melhorar a sua usabilidade, procurando aumentar o espaço de exibição das imagens. Na figura 4.2 é apresentada a nova interface de usuário. São preservadas a janela de volume de dados em 3D e as janelas de exibição de cortes planares 2D: axial, sagital e coronal. E a coordenação entre elas é mantida, de forma que a ação em uma delas é refletida em todas as outras. Isso reforça a percepção 3D.

Só para efeito ilustrativo, são mostrados os dois objetos de interação na janela 3D da figura 4.2. Para não competir a área da tela com as quatro janelas de exibição das imagens, decidimos criar uma janela extra (figura 4.3(b)), que se abre ao clicar um botão disponível na barra de ferramenta (figura 4.3(a)), para controlar os novos objetos de interação. Nesta

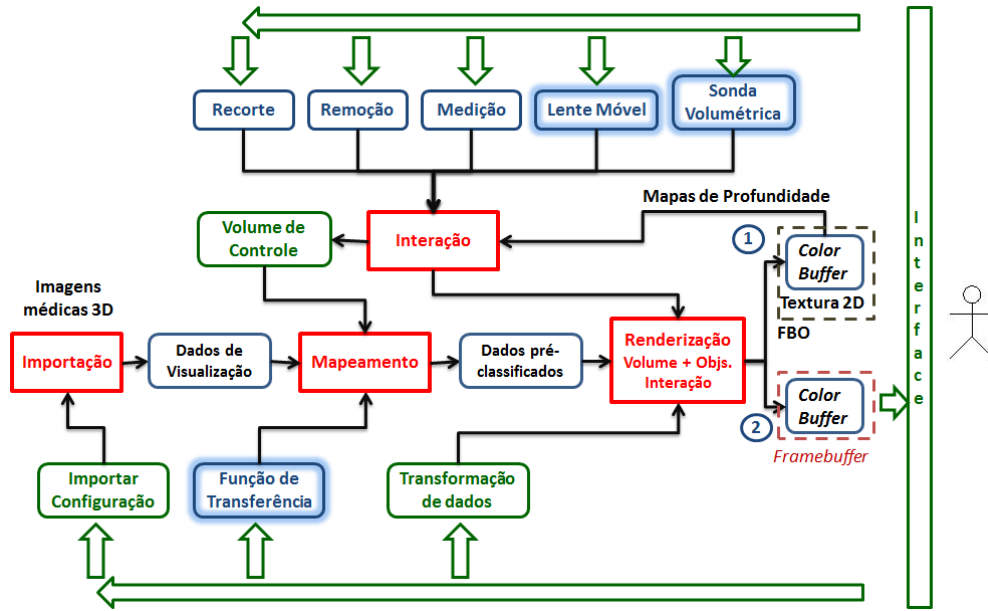


Figura 4.1: Arquitetura de sistema destacando as novas técnicas integradas ao VMTK.

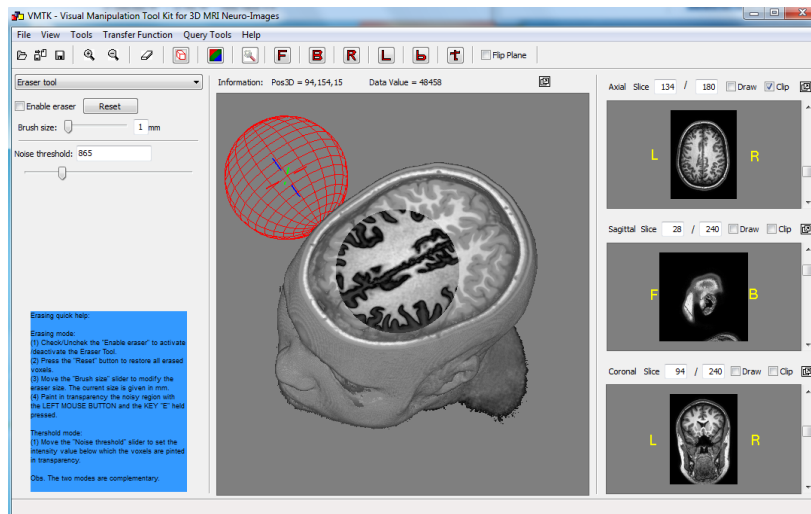


Figura 4.2: Nova interface com 4 janelas sincronizadas no domínio espacial.

janela, o usuário pode optar por ativar/desativar o uso da *sonda volumétrica* ou da *lente móvel*, como também configurar a geometria e as propriedades dos objetos de interação.

Quando ativado um objeto de interação, ele é exibido automaticamente nas janelas de exibição das imagens. O usuário pode deslocá-lo na janela 3D movendo, de acordo com as convenções dadas na seção 3.1.2, utilizando o *cursor* do *mouse* com o seu botão direito pressionado. E nas janelas 2D, os movimentos do *cursor* do *mouse* podem ser controlados pelo movimento do *cursor* do *mouse* ou pelas barras de rolagem que ficam nos respectivos



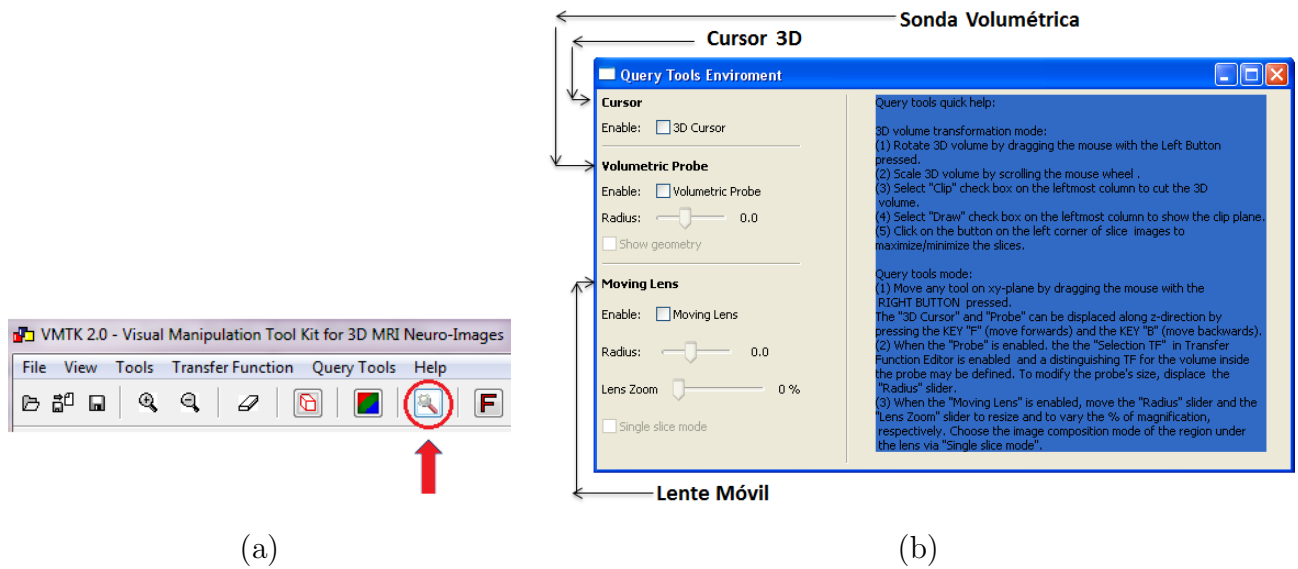


Figura 4.3: Controle dos objetos de interação: (a) ícone correspondente na barra de ferramenta e (b) janela contendo caixas de controle.

lados direitos. Como realimentação visual, utilizamos dois recursos para reforçar a percepção 3D: as vistas nas janelas 2D são sempre coordenadas com o centro dos objetos de interação e as cores das partes dos objetos de interação que ficam no interior do volume são mais difusas.

Por mesmo motivo de priorizar as áreas de exibição das imagens, separamos ainda a janela do editor de função de transferência da janela principal, de forma que ela passa a ser uma janela extra (figura 4.4(b)) que aparece quando se clica um ícone na barra de ferramenta (figura 4.4(a)). Nesta janela, o usuário pode ajustar a cor e a opacidade dos *voxels* manipulando diretamente nas funções de transferência dos quatro canais R (vermelho), B (azul), G (verde) e A (opacidade). Cada uma dessas funções é editável independentemente, através da inclusão, do arrasto ou da remoção dos pontos de controle em cada curva da função de transferência.

É possível ainda selecionar qual função de transferência a ser editada: quando Selection TF é selecionada, edita-se a função de transferência da região focal; caso contrário, é alternada para a função de transferência da região contextual. As imagens resultantes podem ser totalmente coloridas ou em tons de cinza e elas seguem, em tempo interativo, as alterações feitas nas funções de transferência. Isso proporciona uma sensação de controle sobre as ferramentas e facilita análises exploratórias. Em especial, quando o objeto de interação é a sonda volumétrica, temos ainda a alternativa de ver simultaneamente o contexto e a região focal.

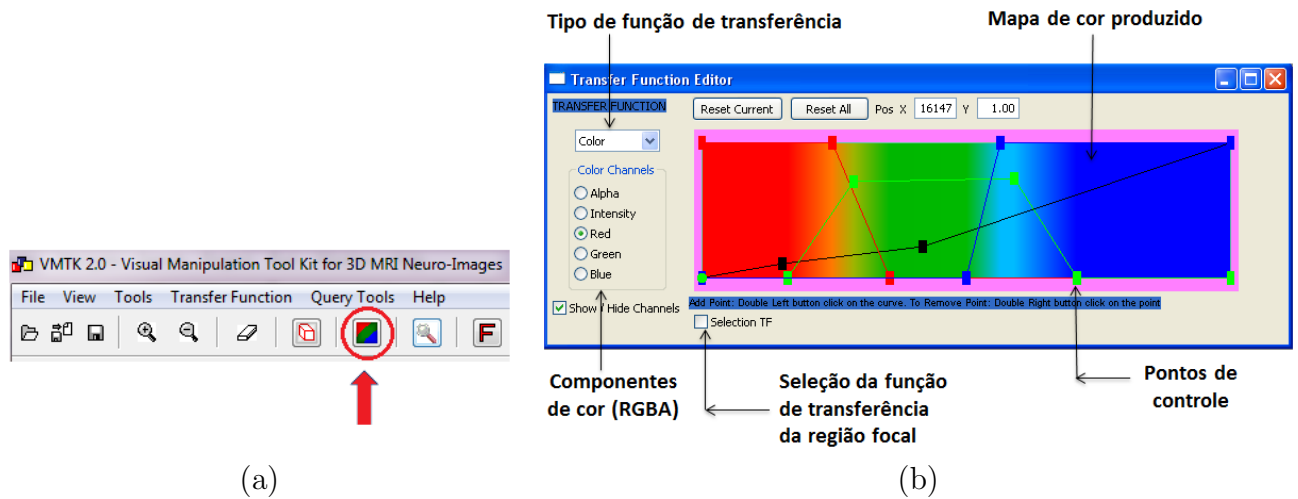


Figura 4.4: Editor interativo de função de transferência: (a) ícone correspondente na barra de ferramenta e (b) interface.

## Processamento Computacional

Para suportar as ações de maneira coordenada e interativa, foi essencial reduzir o tráfego entre CPU e GPU durante as interações. A nossa solução foi pré-processar e manter alguns dados temporários na GPU, através dos quais o sistema consegue responder em tempo aceitável:

1. é alocado um objeto FBO (OpenGL 2010), reservando-se memória interna na GPU para quatro texturas 2D de igual resolução à janela de visualização 3D. As texturas possuem tipo de dados `GL_RGBA` e elas são, de fato, o mapa de profundidade dos *voxels* visíveis do volume de dados e os mapas de profundidade dos objetos de interação, um da lente e dois da sonda, para controlar e realimentar os seus movimentos.
2. é alocado mais um vetor unidimensional que é transferido como textura 1D para a memória da GPU. Este vetor representa a correspondência entre os valores escalares nas regiões focais das imagens e as propriedades ópticas a serem exibidas.

O primeiro conjunto de dados é atualizados dinamicamente na GPU em cada interação e utilizados para gerar realimentações visuais apropriadas. O segundo conjunto de dados é, por sua vez, editado na CPU, transferido para GPU e aplicado na renderização das imagens.

Outra solução eficaz que adotamos para realimentar visualmente os movimentos dos objetos de interação dentro e fora do volume de interação foi a utilização da combinação linear das cores do volume de dados e da sonda geométrica.

## 4.2 Testes de qualidade de imagens

Sendo o principal objetivo deste trabalho o fornecimento de ferramentas de exploração para auxílio ao diagnóstico de anomalias cerebrais, conduzimos experimentos envolvendo avaliação da qualidade das imagens produzidas utilizando a função de transferência, a lente móvel e a sonda volumétrica.

### 4.2.1 Função de Transferência

Foram realizados dois testes para avaliar a qualidade visual das neuro-imagens produzidas com a ferramenta da função de transferência. Utilizando o editor interativo de função de transferência, o usuário adiciona, arrasta e remove um conjunto de pontos de controle para definir correspondências entre os valores escalares do volume e as cores nas imagens.

No primeiro teste, o objetivo foi realçar os detalhes de uma lesão cerebral de forma a torná-la facilmente perceptível pelo usuário. A figura 4.5 apresenta duas imagens geradas com diferentes funções de transferência 1D. Na figura 4.5(a), nota-se a região lesionada no lado esquerdo do cérebro sendo identificável com um mapeamento da função de transferência monotônica padrão. Na figura 4.5(b), após alterá-la para uma função não-monotônica, o contraste nos detalhes das bordas da massa branca do cérebro passam a ser mais acentuado, sendo mais perceptível a magnitude da lesão.

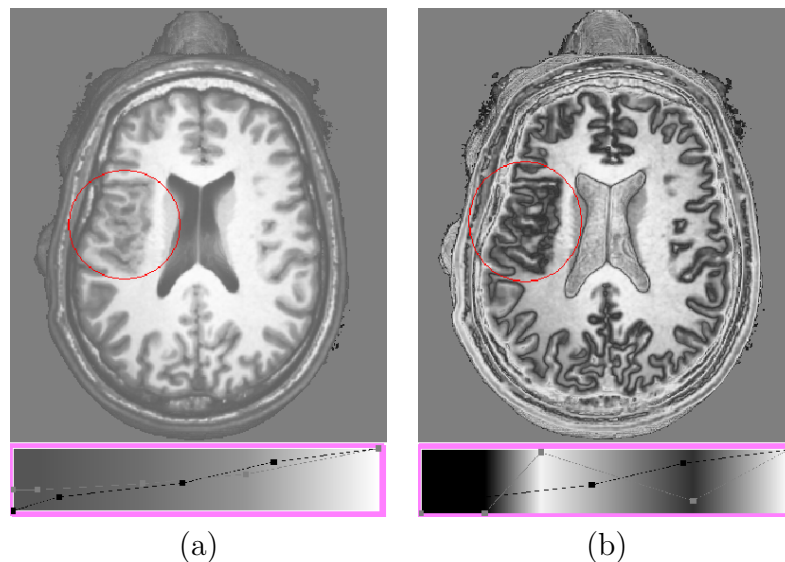


Figura 4.5: Visualização com: (a) uma função de transferência monotônica e (b) uma função de transferência não-monotônica.

Em nosso segundo teste, uma lesão cortical focal sutil, dificilmente perceptível, é realçada utilizando a ferramenta de função de transferência. A figura 4.6 apresenta duas imagens geradas com diferentes funções de transferência 1D. Na figura 4.6(a), foi marcada uma pequena região suspeita de displasia cortical focal quase imperceptível. Podemos, no entanto,

augmentar o contraste da imagem utilizando uma função de transferência não-monotônica. Com isso, detalhes da região lesionada tornaram-se mais perceptíveis, conforme mostra a figura 4.6(b).

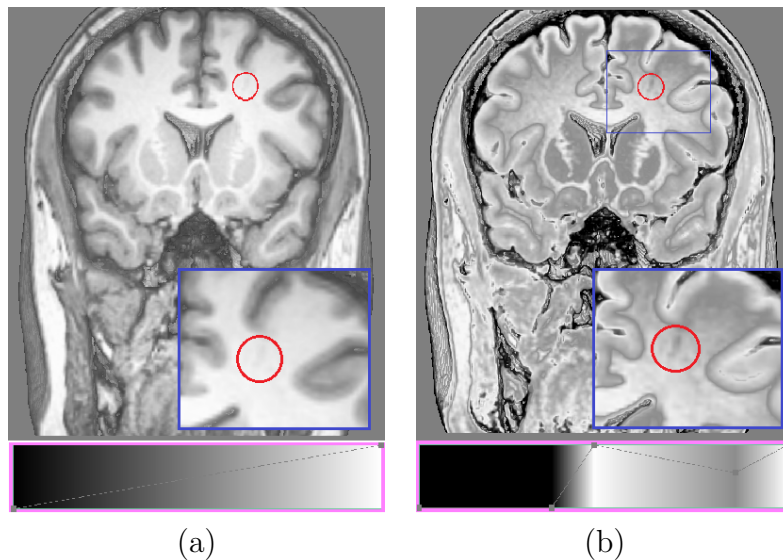


Figura 4.6: Visualização de uma lesão sutil com: (a) uma função de transferência monotônica e (b) uma função de transferência não-monotônica.

## 4.2.2 Lente Móvel

Foram realizados dois testes para avaliar a qualidade visual das neuro-imagens produzidas utilizando a ferramenta de lente móvel.

No primeiro teste, o objetivo foi verificar a qualidade gerada no processo de ampliação de resolução de uma lesão cerebral sutil, figura 4.7. A região de interesse é destacada com uma linha vermelha tracejada na figura 4.7(a). Nas figuras 4.7(b) e 4.7(c), a região de ampliação da lente, com fator  $FA = 2.6$ , é circulado por uma linha vermelha. No primeiro caso, a cor em cada *pixel* na região da lente se resulta da composição de cores das amostras do volume ao longo do raio de projeção, enquanto no segundo caso a cor corresponde à cor do primeiro *voxel* não transparente ao longo do raio de visão (sem composição), conforme explicado na seção 3.3.4. Observe que o contraste no primeiro caso é menor que no segundo.

Sendo Matlab<sup>®</sup> uma ferramenta de amplo uso pelos neurocientistas e pesquisadores de processamento de imagens, o segundo teste foi comparar as imagens geradas em Matlab<sup>®</sup> (MATLAB 2010) e as obtidas através da lente proposta. A figura 4.8 mostra comparativamente uma imagem imageada em Matlab, a mesma imagem filtrada com a lente setada em  $FA = 1.0$  com composição das propriedades ópticas dos *voxels* e, depois, somente com a propriedade óptica do primeiro *voxel* visível. Visualmente, o contraste nas imagens filtradas com a lente sem composição se equipara com o das imagens geradas em Matlab, com a vantagem de podermos “desfocar” os tecidos circundantes.

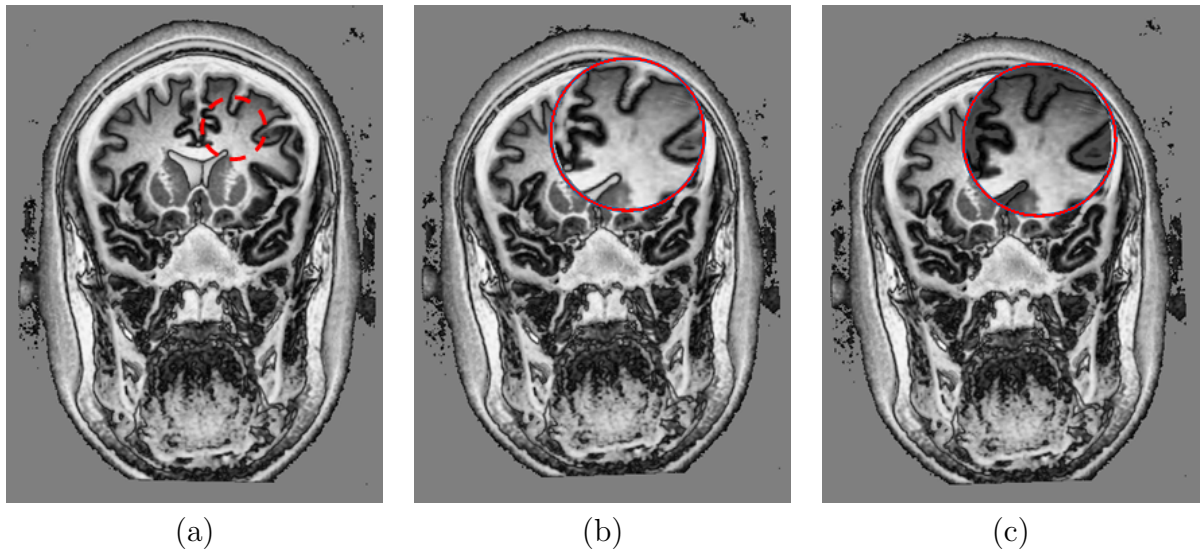


Figura 4.7: Ampliação de uma lesão sutil (a) com uma lente com  $FA = 2.6$ : (b) com composição de cores e (c) sem composição de cores.

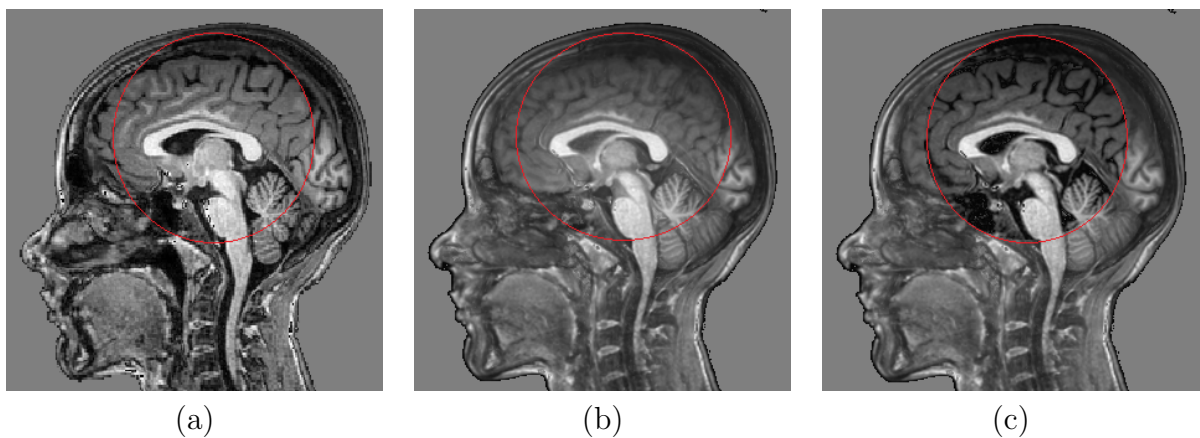


Figura 4.8: Comparação entre (a) imagem em Matlab, (b) lente com composição, e (c) lente sem composição.

### 4.2.3 Sonda Volumétrica

Foi conduzido um experimento com o objetivo de realçar uma região de interesse e destacar um órgão específico. Um conjunto de tarefas foi executado utilizando a *sonda volumétrica* para destacar a amígdala:

1. seleção da região da amígdala,
2. redução da opacidade da região circundante para torná-la semi-transparente,
3. realce dos contrastes da região focal via função de transferência, até alcançar a segmen-

tação visual desejada.

Na figura 4.9 apresentam-se três imagens do resultado obtido. A primeira, exibe uma vista axial das amígdalas apontada pelas setas azuis (figura 4.9.a), enquanto as outras exibem uma vista lateral da amígdala esquerda e direita (figura 4.9.b e figura 4.9.c)

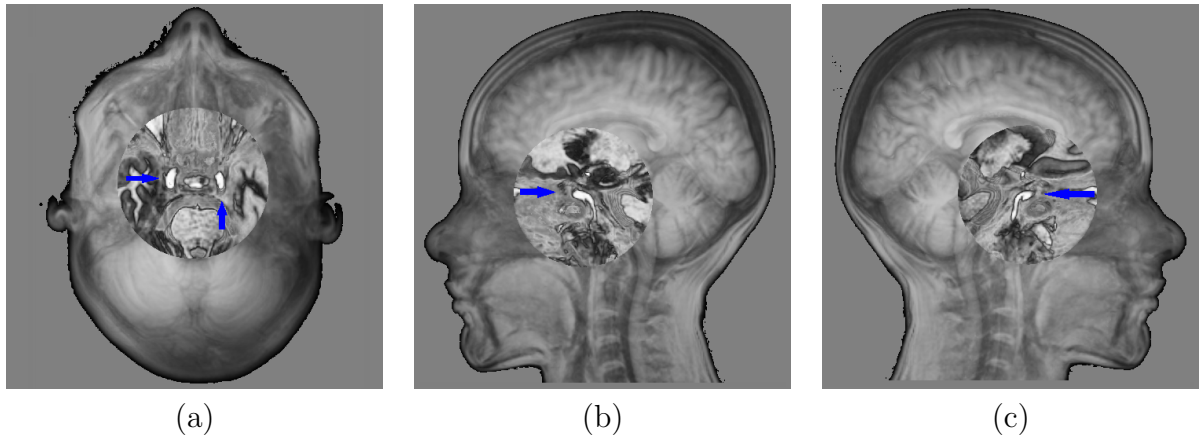


Figura 4.9: Realce das amígdalas: vista axial (a) amígdala esquerda (b) e amígdala direita(c).

### 4.3 Testes de Interatividade

De acordo com (Gnome 2012), o tempo de resposta aceitável em um sistema interativo varia com as tarefas:

- para eventos básicos de *mouse* e de teclado, como movimento ou redimensionamento da janela, o tempo é até 0,1s,
- para eventos correspondentes às tarefas corriqueiras, ou quando são mostrados indicadores de progresso, até 1,0s,
- para eventos que disparem processamento de tarefas mais complexas cuja complexidade o usuário tem consciência, até 10,0s, e
- para eventos que aguardem respostas do usuário, 10,0s.

Com o objetivo de verificar se as nossas ferramentas funcionam dentro dos limites indicados acima, medimos os tempos de execução médios da etapa mais custosa: renderização. Os tempos foram medidos utilizando a função `clock()` disponível na biblioteca `ctime` de C++ e transformados em FPS<sup>1</sup>.

<sup>1</sup>Frame Per Seconds (quadros por segundo)

### 4.3.1 Renderização em Dois Passos

Nas seções 3.3.4 e 3.4.4 vimos que para integrar a geometria da lente e a da sonda geométrica ao volume de dados, utilizamos respectivos mapas de profundidade. E na seção 4.1.2 mostramos que estes mapas são computados durante o passo de renderização em modo *off-screen* e armazenados em 4 texturas atreladas a um *FBO*. No passo em modo normal eles são utilizados na geração de imagens dos objetos de interação integrados ao volume de dados. Portanto, o tempo total  $t_{FBO+Raycast.}$  para geração de uma imagem é dado pela soma

$$t_{FBO+Raycast.} = t_{off-screen} + t_{normal}, \quad (4.1)$$

onde  $t_{off-screen}$  e  $t_{normal}$  correspondem, respectivamente, aos tempos de renderização em modo *off-screen* e em modo *normal*.

Para avaliar a sobrecarga que o primeiro passo em modo *off-screen* gera em relação ao procedimento típico de *ray-casting*, conduzimos um experimento que mede, para diferentes resoluções de imagens finais, os tempos de execução do nosso algoritmo, que denominamos *FBO+Raycast.*, e o tempo do algoritmo 2.1, identificado simplesmente por *Raycast.*. Vale comentar que desabilitamos neste experimento os objetos de interação, de forma que somente o mapa de profundidade da superfície visível do volume de dados é computado.

Na figura 4.10 os tempos obtidos são mostrados. A curva azul indica os tempos do nosso algoritmo e a verde, os do algoritmo clássico. O eixo  $x$  do gráfico representa a resolução das imagens e o eixo  $y$  representa a taxa de quadros gerados por segundo em FPS (*frames per second*).

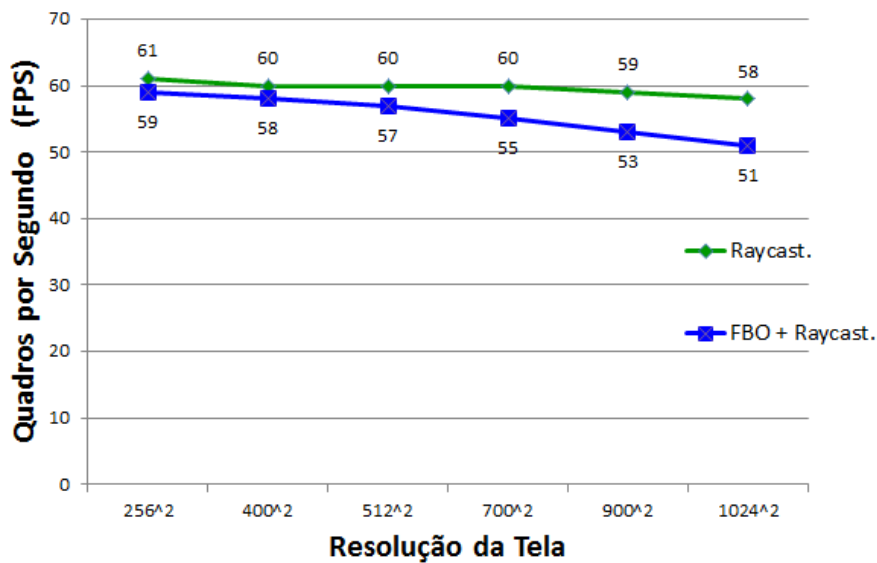


Figura 4.10: Comparação em FPS entre algoritmo clássico de *ray casting* (Raycast.) e o nosso algoritmo de dois passos (FBO+Raycast.).

### 4.3.2 Interatividade da Lente

Com o mapa de profundidade da superfície visível do volume de dados, controlamos os movimentos da lente conforme o procedimento apresentado na seção 3.3.3. A cada evento de movimento do *mouse*, determinamos a partir desse mapa de profundidade e das coordenadas do *cursor* do *mouse* o centro da lente no espaço do objeto. Com base neste centro e no raio da lente, geramos o mapa de profundidade da lente de acordo com a explicação dada na seção 3.3.4. Após determinar a posição do ponto focal com a Equação 3.2, o algoritmo 3.1 é aplicado para gerar uma imagem com a lente integrada nela. E assim, sucessivamente, são geradas as imagens com a lente posicionada em diferentes posições. Isso proporcionaria uma percepção de movimento sobre a superfície desde que o intervalo de tempo entre duas imagens subsequentes seja aceitável. Conduzimos duas séries de medições de tempos para avaliar o quanto a integração da *lente móvel* ao volume de dados pode impactar na interatividade do sistema.

No primeiro experimento medimos a taxa por segundo de quadros com a *lente móvel* integrada, ao variarmos a resolução da imagem de saída. O raio da lente e o fator de ampliação foram fixados em 96 *voxels* e em  $FA = 1.4$ , respectivamente. Na figura 4.11 são apresentados os resultados em FPS. Para facilitar a comparação, plotamos no mesmo gráfico a curva de taxa de quadros do volume de dados prontos para interações (FBO+Raycast.) e a curva taxa de quadros da lente móvel integrada ao volume de dados (Lente Móvel).

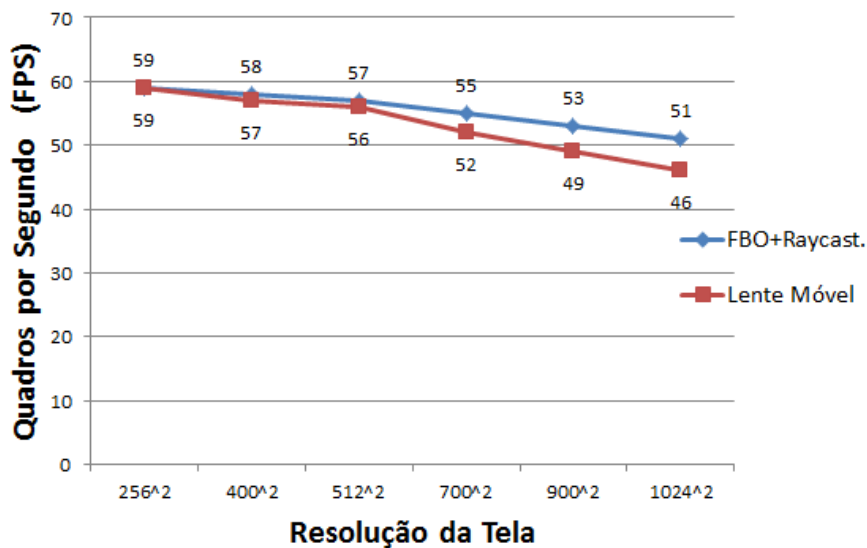


Figura 4.11: Comparação em FPS entre o algoritmo de dois passos (FBO+Raycast.) e com sobrecarga da integração da lente móvel (Lente Móvel).

No segundo experimento para verificar o quanto o tamanho da *lente móvel* pode afetar na sua interatividade, mantemos a resolução da tela em 512×512 e o fator de ampliação em  $FA = 1.0$  e medimos a variação da taxa de quadros sem composição na região focal em



função do raio da lente. Variamos o raio da lente de 20% até 100% em relação ao tamanho da imagem de saída. Repetimos o procedimento para  $FA = 1.5$  e  $FA = 2.0$ . Os resultados em FPS são apresentados na tabela 4.1.

Tabela 4.1: Escalabilidade da lente móvel em FPS.

<b>Tamanho da lente</b>	$FA = 1.0$	$FA = 1.5$	$FA = 2.0$
20%	56	56	56
40%	56	56	55
60%	56	55	55
80%	56	55	55
100%	56	55	55

### 4.3.3 Interatividade da Sonda Volumétrica

De forma análoga à lente móvel, uma sobrecarga no processamento é necessária para integrar a sonda ao volume de dados nas imagens finais. A cada movimento do *mouse* precisamos atualizar o seu centro e gerar dois mapas de profundidade correspondentes à superfície mais próxima e mais distante do volume de dados, segundo a seção 3.4.2. Dois experimentos foram conduzidos para avaliar o custo desta sobrecarga.

O primeiro experimento foi conduzido para medir a taxa de quadros por segundo para distintas resoluções da imagem de saída, quando se integra ao volume de dados a *sonda volumétrica*. O raio da sonda foi mantido fixo em 120 *voxels*. Novamente, para facilitar a comparação, apresentamos no gráfico da Figura 4.12 a curva de taxa de quadros com a sonda (Sonda Volumétrica) e sem a sonda (FBO+Raycast.).

Para verificar o quanto a variação no tamanho da *sonda volumétrica* pode afetar no desempenho do sistema, medimos no segundo experimento os tempos de execução em FPS para diferentes tamanhos, sem composição. Em relação ao primeiro experimento, a única diferença foi variar o raio da sonda entre 20% até 100% em relação a um valor inicial, que fixamos em 120 *voxels*. Na tabela 4.2 são apresentados os resultados.

Tabela 4.2: Escalabilidade da sonda volumétrica.

<b>Volume da Sonda</b>	<b>FPS</b>
20%	55
40%	55
60%	55
80%	55
100%	55

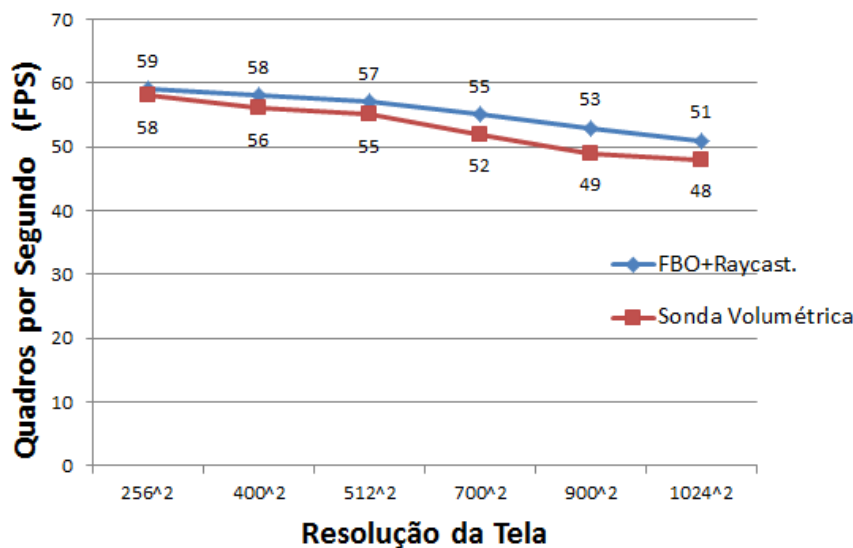


Figura 4.12: Comparação entre as taxas de quadros em FPS entre volume de dados com e sem a sonda volumétrica.

## 4.4 Testes de Usabilidade

Elaboramos um questionário de usabilidade para verificar se os objetos de interação desenvolvidos atendem a expectativa dos especialistas em neuro-imagens. No apêndice A encontra-se uma cópia do questionário.

Contamos com a ajuda das Dras. Ana Carolina Coan e Juliana Porto, ambas do Laboratório de Neuro-imagem da nossa universidade (LNI-UNICAMP 2012), na seleção das imagens de teste. As neuro-imagens foram selecionadas de forma que contemplem lesões grandes e sutis (displasias corticais focais (DCF) tipo IIA e tipo IIB), com localização em hemisfério esquerdo e direito. Observamos que ambos tipos são anomalias com neurônios disfórmicos, diferindo apenas na presença (IIB) ou não (IIA) de células em balão (Bronen, Vives, Kim, Fulbright, Spencer & Spencer 1997). Nas figuras 4.13–4.18 são exibidas as regiões cerebrais lesionadas dos 6 pacientes selecionados.

Essencialmente os testes de usabilidade consistem em:

1. Localização de regiões lesionadas nas 6 neuro-imagens sem nenhuma informação clínica ou histológica, a fim de verificar a eficácia das nossas ferramentas em distinguir tecidos lesionados.
2. Segmentação do tálamo ou hipocampo de um paciente, para avaliar as ferramentas de seleção e realce localizado.
3. Avaliação qualitativa das ferramentas baseada na experiência de teste.

Como os voluntários escolhidos para os testes não tem contato prévio com o nosso protótipo, planejamos dar uma breve explicação das ferramentas e um tempo para eles se familia-

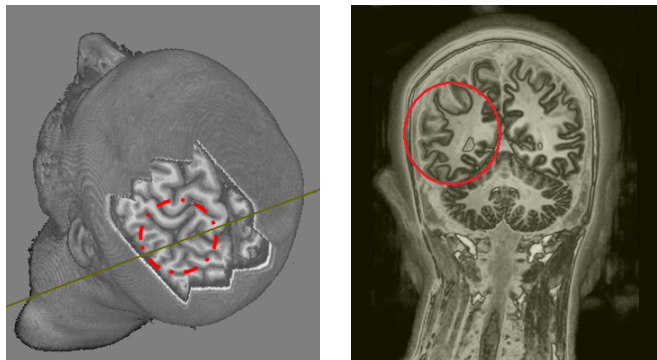


Figura 4.13: Paciente 1: DCF IIA, localizada no quadrante posterior esquerdo.

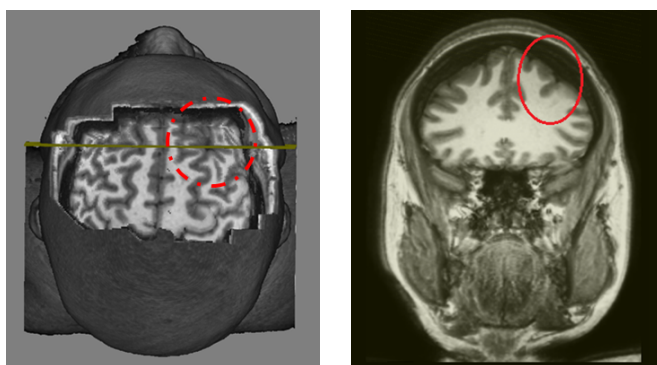


Figura 4.14: Paciente 2: DCF IIA localizada na parte frontal direita.

rizarem com elas. Em seguida, serão realizadas as atividades designadas. E, após finalizá-las, os especialistas devem responder o conjunto de questões referentes à avaliação objetiva e subjetiva das ferramentas testadas e do software em geral.

Tivemos como voluntários 4 neuro-cientistas do Laboratório de Neuro-Imagens e 2 radiologistas do Hospital do Centro Médico de Campinas. Na apresentação do protótipo, todos mostraram grande interesse e deram sugestões valiosas, mas somente 3 deles do Laboratório de Neuro-Imagens aceitaram a realizar os testes. Os outros não se sentiram à vontade para “diagnosticar” sem conhecer o prontuário do paciente. Isso decorreu principalmente do nosso desconhecimento em relação ao protocolo de diagnóstico utilizado pelos médicos para localizar um achado. A avaliação pré-cirúrgica de pacientes com epilepsia parcial refratária é complexa e depende de uma abordagem multidisciplinar (Cendes 2007a). Além das neuro-imagens, a anamnese com o paciente e familiares e o exame EEG são fundamentais. Para não perder as valiosas realimentações que tivemos de forma espontânea, optamos por anotar as observações positivas e negativas à medida que os médicos foram experimentando o nosso protótipo, sem o compromisso de finalizar os testes.

Sintetizamos nesta seção os principais resultados.

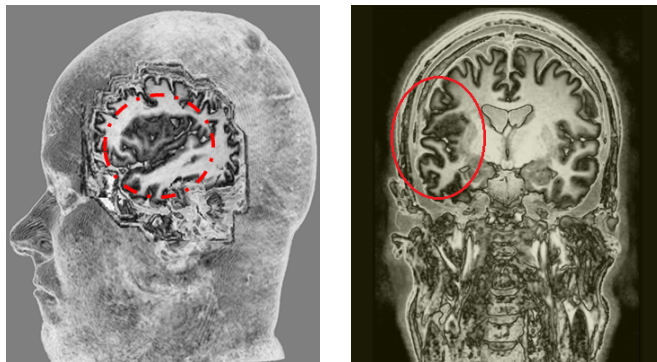


Figura 4.15: Paciente 3: DCF IIB localizada na parte frontal esquerda.



Figura 4.16: Paciente 4: DCF IIB localizada no quadrante posterior direito.

#### 4.4.1 Localização de Lesões Cerebrais

A primeira atividade consistiu na identificação de regiões de regiões lesionadas nos exames de seis pacientes. A figura 4.19 apresenta o gráfico da concordância entre os especialistas na identificação das áreas lesionadas em cada paciente. Houve 83% de concordância.

#### 4.4.2 Segmentação do Hipocampo

A segunda tarefa do usuário foi segmentar o hipocampo de um paciente, mostrados na figura 4.20. Foi sugerido aos voluntários que utilizem a *sonda volumétrica* para selecionar a região de interesse e o editor de função de transferência para realçar localmente os tecidos da região.

#### 4.4.3 Opiniões

Embora tenhamos elaborado um conjunto de questões para avaliação qualitativa das ferramentas testadas, as opiniões orais que registramos foram, a nosso ver, mais valiosas. Estas foram espontâneas e comparativas em relação às ferramentas profissionais. Transcrevemos

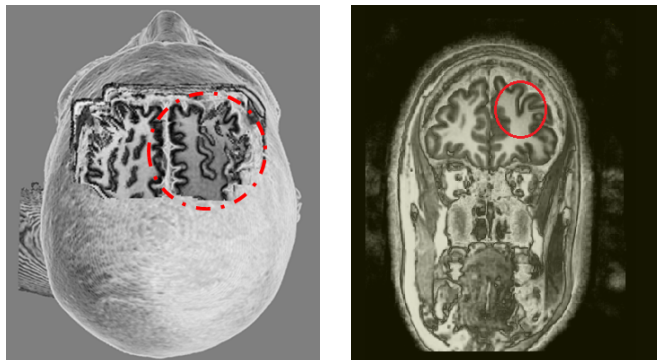


Figura 4.17: Paciente 5: DCF IIA localizada na parte frontal direita.

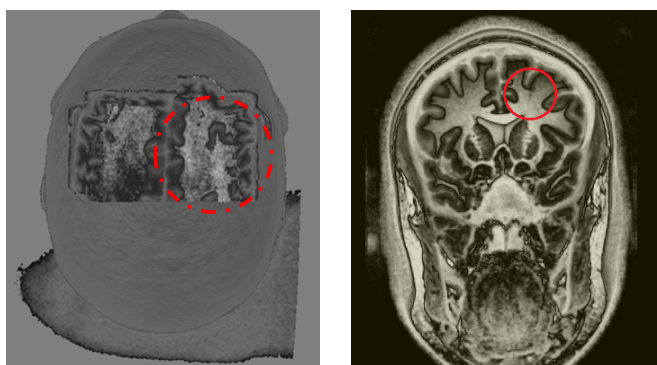


Figura 4.18: Paciente 6: DCF IIA localizada na parte frontal direita.

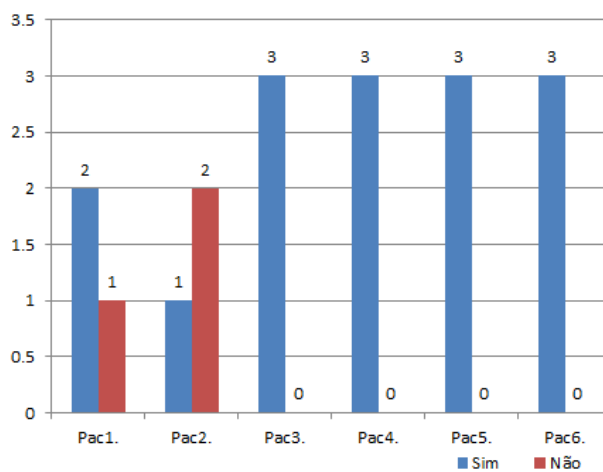


Figura 4.19: Concordâncias em identificação das lesões nas neuro-imagens.

aqui as mais relevantes.

Os pontos positivos registrados foram: visualização 3D; janelas coordenadas permitindo visualizar operações em fatias 2D no espaço do objeto 3D, ou vice-versa; manipulações diretas

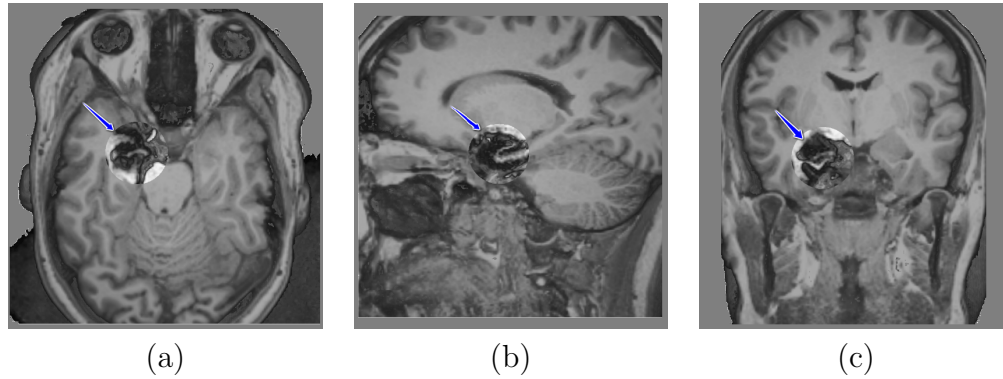


Figura 4.20: Segmentação manual do hipocampo: (a) vista axial, (b) sagital e (c) coronal.

sobre as superfícies visíveis do volume de dados; funções de transferência não-monotônicas que proporcionam realces diferenciados, mesmo que a interpretação do conteúdo da imagem torne mais difícil.

Destacamos entre os pontos negativos: área de visualização das imagens 2D muito pequena, uma vez que para fins de diagnóstico médico, análises por fatia é ainda essencial; ausência de cortes oblíquos, pois há lesões que só conseguem ser identificadas em cortes diferentes do corte axial, sagital e coronal.

## 4.5 Conclusões

Neste capítulo foram apresentados os resultados dos testes realizados a fim de avaliar as três ferramentas propostas e implementadas na nova versão do VMTK (editor interativo de função de transferência, lente móvel e sonda volumétrica).

Em primeiro lugar, testes de qualidade de imagem foram realizadas para detectar lesões sutis nos tecidos cerebrais. Logo, foram realizados testes para medir o desempenho das ferramentas em termos de interatividade. Finalmente, para verificar a utilidade das ferramentas na exploração e diagnóstico médico, foi realizado um teste de usabilidade com alguns especialistas.

## Análise e Discussão dos Resultados

Neste capítulo os resultados alcançados são analisados e discutidos sob três pontos de vista:

1. qualidade visual das imagens,
2. interatividade das ferramentas, e
3. usabilidade.

### 5.1 Qualidade Visual

Entendemos como qualidade visual a capacidade das nossas ferramentas em discriminar visualmente os tecidos sem comprometer a interatividade do sistema. Nesta seção apresentamos uma análise e discussão para as três ferramentas propostas:

**Editor de função de transferência:** A função de transferência é a componente fundamental da renderização volumétrica, pois determina a visibilidade dos dados através da atribuição de propriedades ópticas aos *voxels* do volume.

Embora as funções não-monotônicas tenham mostrado serem eficazes no aumento dos contrastes, em detrimento da univocidade nas correspondências, acreditamos que ele não é o ingrediente principal na identificação das lesões sutis. O principal fator é, a nosso ver, a flexibilidade que o usuário tem para variar os contrastes nas imagens, como foi apresentado na seção 4.2.1, permitindo que este experimente interativamente diferentes padrões de configuração até que encontre um padrão através do qual os tecidos suspeitos se destaquem do restante. Conjeturamos que são estas variações interativas no contraste das imagens que de fato ajudam um especialista a perceber mais facilmente lesões cerebrais.

Por outro lado, percebemos que, para serem efetivas tais interações, é interessante que os especialistas tenham informações clínicas e outros exames do paciente para reduzir regiões de perscrutação. Caso contrário, o processo pode ser longo e infrutífero.

**A lente:** A lente fornece um aumento da área para visualizar as regiões selecionadas da imagem a fim de torná-las mais perceptíveis ao olho humano. Ao se exibir sob a lente a informação original dos *voxels* mais próximos ao observador, consegue-se examinar minuciosamente cada *voxel*, sem perda da percepção 3D que a técnica de renderização volumétrica proporciona (figura 4.8 da seção 4.2.2). Isso pode ajudar na formulação de hipóteses.

Ao fornecer ao usuário controle da atribuição de propriedades ópticas das amostras selecionadas com a lente, ainda é possível explorar as diferenças entre os tecidos através através das variações dos contrastes. Isso propicia a descoberta de lesões sutis. Além disso, diferentes regiões podem ser exploradas de forma interativa em razão da característica móvel da lente (figura 4.7 da seção 4.2.2).

Alguns artefatos podem aparecer em decorrência da resolução das imagens originais, como mostra a figura 5.1. A causa está diretamente relacionada com o problema de *aliasing* na reconstrução da imagem original a partir das suas amostras. Este problema é detalhadamente discutido e uma solução é apresentada em (Engel et al. 2006), que consiste em pré-integrar os valores escalares. No entanto, ainda não temos uma posição clara se tais alterações nos valores originais das amostras são desejáveis em aplicações médicas. Tememos que elas possam induzir erros na interpretação de uma imagem.

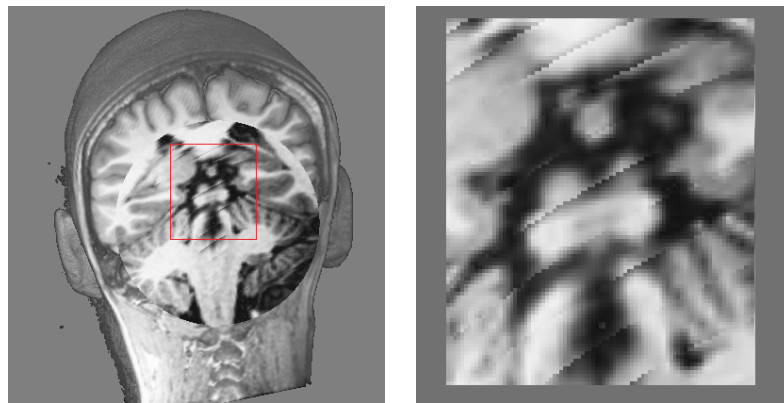


Figura 5.1: Artefatos ficam mais visíveis em regiões com maior fator de ampliação.

**A sonda:** A sonda permite a seleção de regiões correspondentes a órgãos e tecidos cerebrais no espaço 3D, fornecendo um modo de exploração baseado no paradigma foco+contexto. Através da interação com duas funções de transferência, regiões de interesse podem ser realçadas a fim de melhorar a sua percepção visual, enquanto regiões não importantes podem se tornar mais transparentes. A nossa expectativa era que ela auxilie na segmentação manual de algum órgão cerebral.

Mas, como os órgãos da cabeça possuem geometria complexa e muitos deles entrelaçados, a seleção de regiões utilizando uma geometria volumétrica fixa não permite



segmentar apropriadamente uma região específica. Talvez, a nossa sonda possa ser aplicada em uma etapa preliminar de uma segmentação automática, delimitando os *voxels* que serão processados.

## 5.2 Interatividade

Nesta seção analisamos e discutimos os resultados das medições que obtivemos à luz das recomendações listadas na seção 4.3. Considerando que todas as nossas interações envolvem eventos básicos, portanto são aceitáveis respostas em até  $t = 0,1s$ , utilizamos como referência de comparação 6 FPS.

**Renderização em dois passos:** Com o objetivo de aproveitar as técnicas do VMTK e manter a compatibilidade e funcionalidade durante a integração das novas ferramentas, foram avaliados os tempos de renderização volumétrica da nossa proposta de dois passos em relação à renderização utilizando *ray casting* simples. Para resoluções até  $1024 \times 1024$ , obtivemos um desempenho maior ou igual a 51 FPS conforme mostra a figura 4.10, mesmo que o sistema tenha que gerar antecipadamente os mapas de profundidade dos objetos de interação e do volume de dados em cada regeneração (seção 3.1.4). Este resultado demonstra que as respostas do nosso sistema atendem as recomendações para eventos básicos.

**Integração da lente:** Variações de tempo durante a interação com a lente móvel foi avaliada em relação aos tempos de desempenho obtidos na renderização volumétrica em dois passos (seção 4.3.2). Observou-se uma ligeira queda no desempenho ao aumentarmos a resolução da tela, mesmo que a área da lente tenha aumentada (figura 4.11). Pois, para poder selecionar os *voxels* sob a lente e parar o percurso, tivemos que adicionar duas instruções condicionais *if* no *shader* de *ray casting*. Um desempenho de 46 FPS para uma resolução  $1024 \times 1024$  está dentro dos limites estabelecidos para eventos corriqueiros.

Nas medições dos tempos de execução variando somente o fator de ampliação (Tabela 4.1), observou-se alterações quase imperceptíveis no desempenho do sistema ( $\approx 55$  FPS). Esperava-se que, ao adotar a estratégia de parar no primeiro *voxel* o percurso ao longo do raio de projeção, melhoraria o desempenho quando se aumenta a quantidade de *voxels* sob a lente. No entanto, acreditamos que a sobrecarga em decorrência do cômputo da nova direção do raio em cada *pixel* seja mais dominante.

**Integração da sonda:** De forma análoga à lente, houve uma ligeira queda no desempenho quando aumentou a resolução da saída. Mesmo assim, os tempos estão dentro dos limites recomendados para eventos corriqueiros.

Como esperado, não houve variações no desempenho quando se variou o volume da sonda. Pois, em termos de instruções, os raios que atravessam a sonda e os raios que

não atravessam compartilham o mesmo procedimento de composição. A diferença só está na função de transferência utilizada.

### 5.3 Usabilidade

Como foi mencionado na seção 4.4, somente três neuro-cientistas responderam os questionários de usabilidade. Conforme a figura 4.19, tivemos 83% em concordância sem que os voluntários tenham conhecimento do quadro clínico de cada paciente. Quanto à segmentação do hipocampo, todos conseguiram concluir com orientações ao lado. Atribuímos este sucesso à experiência dos voluntários e a facilidade de uso do nosso protótipo.

Em relação às avaliações subjetivas, segue-se um resumo:

1. A *interface gráfica de usuário* é considerada *intuitiva*. Os processos seguem uma sequência de atividades organizadas em ambientes ou janelas independentes, porém coordenadas. No entanto, pelo fato de que os médicos estão mais familiarizados com análises por fatias, eles sugeriram ampliar as áreas de visualização das imagens 2D. Além do mais, eles observaram uma série de detalhes que saem do padrão dos aplicativos médicos. Um é a indicação explícita dos lados da cabeça de um paciente. Isso requer um redesenho da nossa interface gráfica seguindo os padrões médicos.
2. Os especialistas consideram que as ferramentas testadas *auxiliam no diagnóstico da área lesionada*. Eles conseguiram identificar displasias corticais focais de tipo IIA e tipo IIB com relativa facilidade. Como já mencionamos anteriormente, a nosso ver, este sucesso se deve à facilidade de variar os contrastes para experimentar distintas situações.
3. Há lesões que não são visíveis em cortes planares seguindo a angulação dos planos de referência, axial, sagital e coronal (Cendes 2007b). Sugeriram que seja incluída a funcionalidade de cortes oblíquos. Dispomos de todas as funções para implementar esta classe de cortes. Só nos falta uma forma mais amigável de torná-las acessíveis ao usuário.
4. Ajustando os contrastes, a segmentação “visual” de uma região específica tornou uma tarefa simples. No entanto, a forma rígida da sonda apresentou limitações na separação da região do restante dos *voxels* do volume. É, portanto, válido experimentar modelar as sondas com modelos deformáveis.
5. Os médicos foram unânimes em ter uma interface amigável para estimar as dimensões e o volume de uma lesão. Em princípio, VMTK suporta tais funcionalidades. Mas, não estão apresentadas em formas familiares aos médicos. Vale adequar, em parceria com os nossos colegas médicos, a interface atual ao contexto médico.
6. Os especialistas preferem as imagens monocromáticas para estudos e diagnósticos, pois eles estão habituados com a interpretação desta classe de imagens. Talvez seja interessante aprofundarmos mais a linguagem da cor médica para provermos um mapeamento

entre as densidades dos tecidos e tons de cinza mais próximo possível da realidade médica.

## 5.4 Conclusões

Neste capítulo foi apresentada uma análise dos resultados obtidos em termos de qualidade de imagem, interatividade e usabilidade de nossas ferramentas de interação propostas. Também foram discutidas suas principais limitações e possíveis soluções.

Em relação à qualidade de imagem, pelo fato que nossas técnicas e ferramentas estarem baseadas no algoritmo de *ray casting*, as imagens geradas são de qualidade aceitável para estudos e diagnósticos. Além disso, através da variação interativa das propriedades ópticas dos *voxels* do volume, detalhes sutis da imagem ficaram mais perceptíveis.

Em relação à interatividade, os tempos de processamento das imagens com as ferramentas de interação integradas estão acima de 46 FPS, podendo certamente serem considerada interativas as nossas ferramentas de acordo com Gnome (2012).

Em relação à usabilidade, três usuários fizeram os testes e outros três só emitiram comentários construtivos, baseados em suas experiências com os aplicativos profissionais. De modo geral, os resultados foram animadores. Aqueles que fizeram os testes, concluíram-nos com sucesso. E os outros ficaram muito interessados com a possibilidade de visualizar e manipular os dados volumétricos diretamente no espaço nativo das imagens. No entanto, percebemos que os especialistas ainda utilizam as fatias monocromáticas como referência. Portanto, é necessário redesenhar a nossa interface onde as imagens 2D tenham no mínimo o mesmo peso de importância das imagens 3D.

## Conclusões e Trabalhos Futuros

Nesta trabalho foram apresentadas três técnicas de auxílio para os especialistas da área médica na busca lesões sutis em imagens de ressonância magnética:

1. um editor interativo de funções de transferência 1D para facilitar a percepção visual de unidades estruturais e lesões sutis no cérebro. Ao invés de funções de transferência multidimensionais, a adoção de funções de transferência 1D produziram melhor percepção devido à alta sensibilidade das imagens de ressonância magnética para tecidos moles;
2. uma sonda volumétrica para selecionar e manusear regiões de interesse, permitindo uma pré-classificação das amostras condicionado à sua disposição espacial;
3. uma lente móvel para inspecionar áreas de interesse é ampliá-las, preservando as intensidades originais dos *voxels* mais próximos ao observador. Contornamos a atenuação de detalhes da imagem produzidas pelos algoritmos de renderização direta de volumes, propiciando a descoberta de alterações anatômicas sutis.

Nossas ferramentas de sonda e lente estão baseadas no modo de visualização foco+contexto: o usuário manipula de maneira interativa as propriedades ópticas das amostras selecionadas, em diferentes níveis de granularidade e diferentes ângulos, até encontrar achados radiológicos anormais.

As características das GPUs foram exploradas para assegurar a interatividade das ações do usuário, sendo isto um requisito essencial para prover um ambiente de exploração de imagens 3D.

As técnicas desenvolvidas foram integradas no VMTK 2.7 a fim de ter um ambiente de exploração e testes mais rico. No teste de usabilidade realizado com neuro-cientistas do Laboratório de Neuro-Imagem de nossa universidade, alcançou-se um sucesso de 83% na detecção de lesões cerebrais sutis (displasias corticais focais tipo IIA e IIB), sendo os resultados animadores.

Vários temas apresentados neste trabalho podem ser estendidos visando acrescentar novas facilidades ou aumentar o desempenho dos algoritmos propostos.

A interface gráfica de usuário ainda deve ser aprimorada seguindo as convenções de tamanho de janelas, navegação e de *short-cuts* dos aplicativos médicos. Também rotações em relação a pontos variáveis e recortes oblíquos devem ser explorados para facilitar a percepção e identificação de quaisquer tipos de lesão.

Como a especificação de uma boa função de transferência pode ser trabalhosa, há possibilidade de integrar algoritmos de reconhecimento de padrões e segmentação para ajustar automaticamente a função de transferência.

Para reforçar a compreensão e orientação das imagens cerebrais, pode-se explorar o uso de técnicas de renderização não foto-realísticas e ilustrativas, em parceria com os nossos colegas médicos.

Sendo a principal limitação da sonda volumétrica sua geometria fixa para seleção de tecidos, sondas de geometrias flexíveis e deformáveis devem ser avaliadas para seleção de estruturas cerebrais.

Um estudo mais abrangente das lesões cerebrais requer neuro-imagens de distintas modalidades (CT, MRI, PET, fMRI). Um dos temas de pesquisa, denominado *visualização multimodal*, é combinar estas modalidades em uma única imagem para visualizar simultaneamente as características estruturais, anatômicas e funcionais de um cérebro. Gostaríamos de aplicar as ferramentas desenvolvidas neste trabalho no contexto de visualização multimodal.

# Bibliografia

- Batagelo, H. C. & Wu, S.-T. (2008). A framework for gpu-based application-independent 3d interactions, *Vis. Comput.* **24**(12): 1003–1012.  
URL: <http://dx.doi.org/10.1007/s00371-008-0296-y>
- Bier, E. A., Stone, M. C., Pier, K., Buxton, W. & DeRose, T. D. (1993). Toolglass and magic lenses: the see-through interface, *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, ACM, New York, NY, USA, pp. 73–80.  
URL: <http://doi.acm.org/10.1145/166117.166126>
- Bronen, R. a., Vives, K. P., Kim, J. H., Fulbright, R. K., Spencer, S. S. & Spencer, D. D. (1997). Focal cortical dysplasia of Taylor, balloon cell subtype: MR differentiation from low-grade tumors, *AJNR. American journal of neuroradiology* **18**(6): 1141–1151.
- Bruckner, S. & Gröller, M. E. (2005). Volumeshop: An interactive system for direct volume illustration, in H. R. C. T. Silva, E. Gröller (ed.), *Proceedings of IEEE Visualization 2005*, pp. 671–678.  
URL: <http://www.cg.tuwien.ac.at/research/publications/2005/bruckner-2005-VIS/>
- Card, S. K., Mackinlay, J. D. & Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think*, Academic Press, London.
- Cendes, F. (2007a). Neuroimagem na investigação das síndromes hemisféricas. [Acessado em Fevereiro 2013].  
URL: [http://www.lasse.med.br/mat\\_didatico/lasse1/textos/fernando02.html](http://www.lasse.med.br/mat_didatico/lasse1/textos/fernando02.html)
- Cendes, F. (2007b). Volumetry and diagnostic MRI evaluation of hippocampal sclerosis. [Acessado em Fevereiro 2013].  
URL: [http://www.lasse.med.br/mat\\_didatico/lasse1/textos/fernando02.html](http://www.lasse.med.br/mat_didatico/lasse1/textos/fernando02.html)
- Correa, C. & Ma, K.-L. (2009). Visibility-driven transfer functions, *Visualization Symposium, 2009. Pacific Vis '09. IEEE Pacific*, pp. 177–184.

- Doi, K. (2006). Diagnostic imaging over the last 50 years: research and development in medical imaging science and technology, *Physics in Medicine and Biology* **51**(13): R5–R27.  
URL: <http://stacks.iop.org/0031-9155/51/i=13/a=R02>
- Drebin, R. A., Carpenter, L. & Hanrahan, P. (1988). Volume rendering, *SIGGRAPH Comput. Graph.* **22**: 65–74.
- Drevelegas, A. (2011). *Imaging of Brain Tumors With Histological Correlations*, Springer.
- Ebert, D. & Rheingans, P. (2000). Volume illustration: non-photorealistic rendering of volume models, *Proceedings of the conference on Visualization '00, VIS '00*, IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 195–202.  
URL: <http://dl.acm.org/citation.cfm?id=375213.375241>
- Engel, K., Hadwiger, M., Kniss, J., Rezk-Salama, C. & Weiskopf, D. (2006). *Real-Time Volume Graphics*, AK Peters.
- Foley, J. (1996). *Computer graphics: principles and practice*, The systems programming series, Addison-Wesley.
- GDCM, *Grass roots DiCoM Version 2.0.18* (2011). [Acessado em Janeiro 2012].  
URL: <http://www.creatis.insa-lyon.fr/software/public/Gdcm>
- Glatter, M., Huang, J., Ahern, S., Daniel, J. & Lu, A. (2008). Visualizing temporal patterns in large multivariate data using modified globbing, *IEEE Transactions on Visualization and Computer Graphics* **14**(6): 1467–1474.  
URL: <http://dx.doi.org/10.1109/TVCG.2008.184>
- GLSL (2010). The OpenGL Shading Language (language version 4.10). [Acessado em Janeiro 2012].  
URL: <http://www.opengl.org/documentation/glsl/>
- Gnome (2012). Gnome human interface guidelines 2.2.2. acceptable response times. [Acessado em Maio 2012].  
URL: <http://developer.gnome.org/hig-book/3.2/feedback-response-times.html.en>
- Gosink, L., Garth, C., Anderson, J., Bethel, E. & Joy, K. (2011). An application of multivariate statistical analysis for query-driven visualization, *Visualization and Computer Graphics, IEEE Transactions on* **17**(3): 264–275.
- Haidacher, M., Patel, D., Bruckner, S., Kanitsar, A. & Groller, M. (2010). Volume visualization based on statistical transfer-function spaces, *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pp. 17–24.

- Hauser, H., Mroz, L., Italo Bisch, G. & Groller, M. (2001). Two-level volume rendering, *Visualization and Computer Graphics, IEEE Transactions on* **7**(3): 242–252.
- He, T., Hong, L., Kaufman, A. & Pfister, H. (1996). Generation of transfer functions with stochastic search techniques, *Proceedings of the 7th conference on Visualization '96, VIS '96*, IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 227–.
- URL: <http://portal.acm.org/citation.cfm?id=244979.245572>
- Heiken, J. P., Peterson, C. M. & Menias, C. O. (2005). Virtual colonoscopy for colorectal cancer screening: current status, *Cancer Imaging: The Official Publication of the International Cancer Imaging Society* **5**(October): S133–S139.
- URL: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1665314&tool=pmcentrez&rendertype=abstract>
- Hornak, J. (1996). The basics of MRI. [Acessado em Maio 2012].
- URL: <http://www.cis.rit.edu/htbooks/mri/index.html>
- Hüsing, B., Jäncke, L. & Tag, B. (2006). *Impact Assessment of Neuroimaging: Final Report*, TA-SWISS, IOS Press.
- Illán, I., Górriz, J., López, M., Ramírez, J., Salas-Gonzalez, D., Segovia, F., Chaves, R. & Puntonet, C. (2011). Computer aided diagnosis of alzheimer's disease using component based svm, *Applied Soft Computing* **11**(2): 2376–2382.
- URL: <http://linkinghub.elsevier.com/retrieve/pii/S1568494610002449>
- Jezzard, P., Matthews, P. & Smith, S. (2001). *Functional MRI: an introduction to methods*, Oxford medical publications, Oxford University Press.
- Kalat, J. (2007). *Biological Psychology*, Thomson/Wadsworth.
- Kernighan, B. & Ritchie, D. (1988). *C Programming Language*, 2nd edn, Prentice Hall.
- Kindlmann, G. & Durkin, J. W. (1998). Semi-automatic generation of transfer functions for direct volume rendering, *Proceedings of the 1998 IEEE symposium on Volume visualization, VVS '98*, ACM, New York, NY, USA, pp. 79–86.
- URL: <http://doi.acm.org/10.1145/288126.288167>
- Kniss, J., Kindlmann, G. & Hansen, C. (2001). Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets, *Proceedings of the conference on Visualization '01*, IEEE Computer Society, Washington, DC, USA, pp. 255–262.
- Kniss, J., Kindlmann, G. & Hansen, C. (2002a). Multidimensional transfer functions for interactive volume rendering, *IEEE Transactions on Visualization and Computer Graphics* **8**: 270–285.



- Kniss, J., Kindlmann, G. & Hansen, C. (2002b). Simian, a volume rendering tool for scientific visualization and medical imaging. [Acessado em Outubro 2010].  
URL: <http://www.cs.utah.edu/~jmk/simian/index.htm>
- LaMar, E., Hamann, B. & Joy, K. (2001). A magnification lens for interactive volume visualization, *Computer Graphics and Applications, 2001. Proceedings. Ninth Pacific Conference on*, pp. 223–232.
- Levoy, M. (1988). Display of surfaces from volume data, *IEEE Comput. Graph. Appl.* **8**(3): 29–37.
- Levoy, M. (1990). Efficient ray tracing of volume data, *ACM Trans. Graph.* **9**(3): 245–261.  
URL: <http://doi.acm.org/10.1145/78964.78965>
- LNI-UNICAMP (2012). Laboratório de neuro-imagem - faculdade de ciências médicas – unicamp. [Acessado em Junho 2012].  
URL: <http://www.hc.unicamp.br/laboratorios/lab-neuroimagem.shtml>
- Lu, A., Morris, C. J., Ebert, D. S., Rheingans, P. & Hansen, C. (2002). Non-photorealistic volume rendering using stippling techniques, *Proceedings of the conference on Visualization '02, VIS '02*, IEEE Computer Society, Washington, DC, USA, pp. 211–218.  
URL: <http://dl.acm.org/citation.cfm?id=602099.602131>
- Marks, J., Andalman, B., Beardsley, P. A., Freeman, W., Gibson, S., Hodgins, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., Ryall, K., Seims, J. & Shieber, S. (1997). Design galleries: a general approach to setting parameters for computer graphics and animation, *Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH '97*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 389–400.  
URL: <http://dx.doi.org/10.1145/258734.258887>
- MATLAB (2010). *version 7.11.0 (R2010b)*, The MathWorks Inc., Natick, Massachusetts.
- Max, N. (1995). Optical models for direct volume rendering, *Visualization and Computer Graphics, IEEE Transactions on* **1**(2): 99–108.
- Meissner, M., Huang, J., Bartz, D., Mueller, K. & Crawfis, R. (2000). A practical evaluation of popular volume rendering algorithms, *Proceedings of the 2000 IEEE symposium on Volume visualization, VVS '00*, ACM, New York, NY, USA, pp. 81–90.  
URL: <http://doi.acm.org/10.1145/353888.353903>
- Mesquita, L. A. G. & Wu, S.-T. (2001). Cursors 3D using 2D devices (on line in portuguese), *Technical report*, Department of Computer Engineering and Industrial Automation, State University of Campinas. [Acessado em Junho 2012].

- URL: <ftp://ftp.dca.fee.unicamp.br/pub/docs/techrep/2001/lmesq-2001-cursor3d.pdf>
- Microsoft (2012). High Level Shading Language. [Acessado em Junho 2012].  
URL: [http://msdn.microsoft.com/en-us/library/windows/desktop/bb509561\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/bb509561(v=vs.85).aspx)
- Microsoft (2011). Programming Guide for Direct3D 11). [Acessado em Junho 2012].  
URL: [http://msdn.microsoft.com/en-us/library/ff476080\(v=vs.85\)](http://msdn.microsoft.com/en-us/library/ff476080(v=vs.85))
- Namin, S., Moghaddam, H., Jafari, R., Esmaeil-Zadeh, M. & Gity, M. (2010). Automated detection and classification of pulmonary nodules in 3D thoracic ct images, *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pp. 3774–3779.
- NEMA (2012). Digital imaging and communications in medicine. [Acessado em Junho 2012].  
URL: <http://medical.nema.org/>
- Neri, E., Caramella, D. & Bartolozzi, C. (2007). *Image Processing in Radiology: Current Applications*, Medical Radiology: Diagnostic Imaging, Springer.
- Nvidia (2012a). C for Graphics. [Acessado em Janeiro 2012].  
URL: [http://http.developer.nvidia.com/CgTutorial/cg\\_tutorial\\_chapter01.html](http://http.developer.nvidia.com/CgTutorial/cg_tutorial_chapter01.html)
- Nvidia (2012b). What is GPU computing? [Acessado em Julho 2012].  
URL: <http://www.nvidia.com/object/what-is-gpu-computing.html>
- OpenGL (2010). OpenGL, the cross-platform graphics API. [Acessado em Agosto 2011].  
URL: <http://www.opengl.org>
- Pfister, H., Lorensen, B., Bajaj, C., Kindlmann, G., Schroeder, W., Avila, L., Raghu, K., Machiraju, R. & Lee, J. (2001). The transfer function bake-off, *Computer Graphics and Applications, IEEE* **21**(3): 16–22.
- Rezk-Salama, C., Engel, K. & Higuera, F. V. (2002). The OpenQVis Project. [Acessado em Maio 2012].  
URL: <http://openqvis.sourceforge.net/>
- Roden, C. & Brett, M. (2000). Stereotaxic display of brain lesions, *Behavioural Neurology* **12**: 191–200.
- Roettger, S. (2006). The Volume Library. [Acessado em Outubro 2011].  
URL: <http://www9.informatik.uni-erlangen.de/External/vollib>
- Roettger, S., Bauer, M. & Stamminger, M. (2005). Spatialized Transfer Functions, *Euro-Vis05: Joint Eurographics - IEEE VGTC Symposium on Visualization*, pp. 271–278.

- Shirley, P., Ashikhmin, M., Gleicher, M., Marschner, S., Reinhard, E., Sung, K., Thompson, W. & Willemssen, P. (2005). *Fundamentals of Computer Graphics*, 2nd edn, A. K. Peters, Ltd., Natick, MA, USA.
- Shneiderman, B. (1994). Dynamic queries for visual information seeking, *IEEE Softw.* **11**(6): 70–77.
- Sikachev, P., Rautek, P., Bruckner, S. & Gröller, M. E. (2010). Dynamic focus+context for volume rendering, *Proceedings of Vision, Modeling and Visualization 2010*, University of Siegen, Germany, pp. 331–338.
- Simon, S. (1997). *The brain: our nervous system*, Morrow Junior Books.
- Spence, R. (2007). *Information visualization: design for interaction*, Pearson/Prentice Hall.
- Stegmaier, S., Strengert, M., Klein, T. & Ertl, T. (2005). A simple and flexible volume rendering framework for graphics-hardware-based raycasting, *Proceedings of the Fourth Eurographics / IEEE VGTC conference on Volume Graphics*, VG'05, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 187–195.  
URL: <http://dx.doi.org/10.2312/VG/VG05/187-195>
- Stockinger, K., Shalf, J., Wu, K. & Bethel, E. (2005). Query-driven visualization of large data sets, *Visualization, 2005. VIS 05. IEEE*, pp. 167–174.
- Swenson, R. S. (2012). Review of clinical and functional neuroscience.  
URL: <http://www.dartmouth.edu/~rswenson/NeuroSci/index.html>
- Szeliski, R. (2012). *Computer Vision, Algorithms and Applications*, 3.
- Tang, J., Rangayyan, R., Xu, J., El Naqa, I. & Yang, Y. (2009). Computer-aided detection and diagnosis of breast cancer with mammography: Recent advances, *Information Technology in Biomedicine, IEEE Transactions on* **13**(2): 236–251.
- Telea, A. C. (2008). *Data Visualization: Principles and Practice*, A K Peters, Ltd., Wellesley, MA, USA.  
URL: <http://www.akpeters.com/product.asp?ProdCode=3066>
- Viola, I., Kanitsar, A. & Groller, M. (2005). Importance-driven feature enhancement in volume visualization, *Visualization and Computer Graphics, IEEE Transactions on* **11**(4): 408–418.
- Wang, L., Zhao, Y., Mueller, K. & Kaufman, A. (2005). The magic volume lens: an interactive focus+context technique for volume rendering, *Visualization, 2005. VIS 05. IEEE*, pp. 367–374.
- Watt, A. (2000). *3D Computer Graphics*, 3rd edn, Addison-Wesley.

- Weishaupt, D., Köchli, V. & Marincek, B. (2006). *How Does Mri Work?: An Introduction to the Physics And Function of Magnetic Resonance Imaging*, Lecture notes in mathematics, Springer.
- Weiskopf, D. (2007). *GPU-Based Interactive Visualization Techniques*, Mathematics and Visualization, Springer.
- Weiskopf, D., Engel, K. & Ertl, T. (2003). Interactive clipping techniques for texture-based volume visualization and volume shading, *Visualization and Computer Graphics, IEEE Transactions on* **9**(3): 298–312.
- Williamson, C. & Shneiderman, B. (1992). The dynamic homefinder: evaluating dynamic queries in a real-estate information exploration system, *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92, ACM, New York, NY, USA, pp. 338–346.  
URL: <http://doi.acm.org/10.1145/133160.133216>
- Wright, H. (2007). *Introduction to scientific visualization*, Springer.
- Wu, S.-T., Yasuda, C. L. & Cendes, F. (2012). Interactive curvilinear reformatting in native space, *IEEE Transactions on Visualization and Computer Graphics* **18**(2): 299–308.  
URL: <http://dx.doi.org/10.1109/TVCG.2011.40>
- Wu, S.-T., Yauri Vidalon, J. & de Souza Watanabe, L. (2011a). Snapping a cursor on volume data, *Graphics, Patterns and Images (Sibgrapi), 2011 24th SIBGRAPI Conference on*, pp. 109–116.
- Wu, S.-T., Yauri Vidalon, J. & de Souza Watanabe, L. (2011b). VMTK: Visual Manipulation Tool for 3D MR Neuro-Images. [Acessado em Julho 2012].  
URL: <http://www.dca.fee.unicamp.br/projects/mtk/wu/vmtk2.html>
- wxFormBuilder, An Open-Source GUI Builder for wxWidgets Version 3.1.70* (2010). [Acessado em Janeiro 2012].  
URL: <http://sourceforge.net/projects/wxformbuilder>
- wxWidgets, Cross-Plataform GUI Library Version 2.8.12* (2011). [Acessado em Janeiro 2012].  
URL: <http://wxwidgets.org>
- Yoo, T. (2004). *Insight Into Images: Principles and Practice for Segmentation, Registration, and Image Analysis*, Ak Peters Series, A K Peters.
- Zhou, J., Döring, A. & Tönnies, K. D. (2004). Distance based enhancement for focal region based volume rendering, *Proceeding of Bildverarbeitung Für die Medizin*, pp. 199–203.

- 
- Zhou, J., Hinz, M. & Tönnies, K. (2002). Focal region-guided feature-based volume rendering, *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pp. 87–90.
- Zimmerman, R., Gibby, W. & Carmody, R. (2000). *Neuroimaging: Clinical and Physical Principles*, Springer.

Apêndice **A**

Apêndice



# Teste de Usabilidade

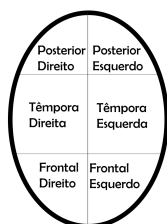
Este teste tem como principal objetivo avaliar a eficácia das quatro funções propostas no trabalho de Mestrado do José Elías Yauri Vidalón para diagnóstico e segmentação das neuro-imagens sob controle dos médicos especialistas.

O protótipo *VMTK (Visual Manipulation Tool for 3D MR Neuro-Images)* consegue carregar neuro-imagens no formato DICOM e exibí-las simultaneamente em 4 vistas: 1 vista 3D, 1 vista 2D sagital, 1 vista 2D coronal e 1 vista 2D axial. Ele dispõe de uma série de funcionalidades que permitem o usuário explorar as neuro-imagens sob diversos ângulos. O teste é dividido em duas partes. A primeira parte do teste consiste na **localização da área lesionada** do paciente e na **segmentação interativa** de um órgão do cérebro. A segunda parte elaboramos algumas perguntas para a avaliação do uso do *software*.

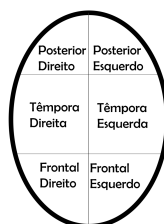
## Primeira Parte

Por favor, localize a localização da área lesionada dos pacientes:

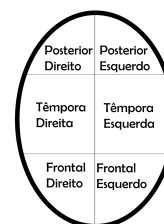
Paciente 1



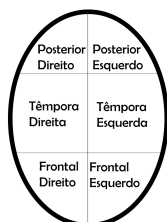
Paciente 2



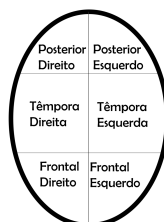
Paciente 3



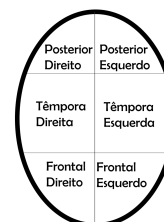
Paciente 4



Paciente 5



Paciente 6



Por favor, segmente o cerebelo, o tálamo e o hipocampo do Paciente ( )

## Segunda Parte

Esse questionário tem a finalidade de avaliar o software *VMTK*

**1. O quanto intuitivo é a interface do *software*?**

Nada intuitivo ○—○—○—○—○ Muito intuitivo

**2. O *software* ajuda no diagnóstico de uma área lesionada?**

Pouco ajuda ○—○—○—○—○ Muito ajuda

**3. A segmentação de partes do cérebro (hipotálamo, hipocampo, cerebelo, etc) foi uma tarefa**

Muito difícil ○—○—○—○—○ Muito fácil

**4. Qua(is) ferramenta(s) te ajudou na identificação de uma área lesionada?**

- Sonda volumétrica.
- Lente de aumento.
- Função de transferência.
- Cursor 3D
- Outra: \_\_\_\_\_

**5. Qua(is) ferramenta(s) te ajudou na segmentação, ou pré-segmentação, de uma parte do cérebro?**

- Sonda volumétrica.
- Lente de aumento.
- Função de transferência.
- Cursor 3D
- Outra: \_\_\_\_\_



6. Qual ferramenta você sentiu mais dificuldade em utilizar?

- Sonda volumétrica.
- Lente de aumento.
- Função de transferência.
- Cursor 3D
- Outra: \_\_\_\_\_

7. Você utilizaria algum outro *software* para auxiliar no diagnóstico?

- Sim. Qual: \_\_\_\_\_
- Não.

8. Você utilizaria algum outro *software* para auxiliar na segmentação?

- Sim. Qual: \_\_\_\_\_
- Não.

9. Qual funcionalidade adicional você gostaria que o *software* tivesse para os objetivos propostos?

---

---

10. Qual funcionalidade que você acha supérflua para os objetivos propostos?

---

---

11. Qual sua opinião sobre o *software*? Deixe por favor suas sugestões.

---

---

---

---

---

---