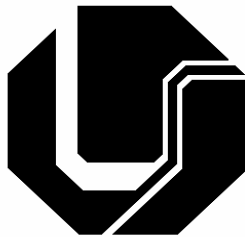


UNIVERSIDADE FEDERAL DE UBERLÂNDIA
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



INTERPOLAÇÃO TRIDIMENSIONAL DE IMAGENS DE
TOMOGRAFIA COMPUTADORIZADA UTILIZANDO
EQUAÇÕES DIFERENCIAIS PARCIAIS

SANDRERLEY RAMOS PIRES

FEVEREIRO
2007

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

INTERPOLAÇÃO TRIDIMENSIONAL DE IMAGENS DE
TOMOGRAFIA COMPUTADORIZADA UTILIZANDO
EQUAÇÕES DIFERENCIAIS PARCIAIS

Tese apresentada por Sandrerley Ramos Pires à Universidade Federal de Uberlândia para obtenção do título de Doutor em Engenharia Elétrica aprovada em 27/02/2007 pela Banca Examinadora:

Professora Edna Lúcia Flores, Dr^a (Orientadora - UFU).

Professora Célia Aparecida Zorzo Barcelos, Dr^a (Co-orientadora - UFU).

Professor João Baptista Tadanobu Yabu-uti, Dr. (UNICAMPI).

Professor Luís Gustavo Nonato, Dr. (USP São Carlos).

Professor Gilberto Arantes Carrijo, Dr. (UFU).

Professor Antônio Cláudio Paschoarelli Veiga, Dr. (UFU).

Professor João Batista Destro Filho, Dr. (UFU).

Dados Internacionais de Catalogação na Publicação (CIP)

P667i Pires, Sandrerley Ramos, 1966-

Interpolação tridimensional de imagens de tomografia computadorizada utilizando equações diferenciais parciais / Sandrerley Ramos Pires. - 2007.

202 f. : il.

Orientadora: Edna Lúcia Flores.

Co-orientadora: Célia Aparecida Zorzo Barcelos.

Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.

Inclui bibliografia.

1. Engenharia elétrica - Teses. 2. Tomografia computadorizada - Teses. I. Flores, Edna Lúcia. II. Barcelos, Célia Aparecida Zorzo. III. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 621.3

INTERPOLAÇÃO TRIDIMENSIONAL DE IMAGENS DE TOMOGRAFIA COMPUTADORIZADA UTILIZANDO EQUAÇÕES DIFERENCIAIS PARCIAIS

SANDRERLEY RAMOS PIRES

Tese apresentada por Sandrerley Ramos Pires à Universidade Federal de Uberlândia
como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica.

Professora Edna Lúcia Flores, Dr.^a
Orientadora

Professor Darizon Alves de Andrade, Dr.
Coordenador do Curso de Pós-Graduação

Dedicatória

Agradeço imensamente a minha esposa e minha mãe, mas gostaria de dedicar este trabalho a meu pai, Natal de Souza Pires.

“Se você pensa que pode ou pensa que não pode, em ambos os casos você está certo”

Henry Ford

Agradecimentos

A Deus, por permitir minha existência.

A meus pais, Natal (que não pôde me acompanhar até aqui) e Alzira, por viabilizar esta existência.

A minha esposa Dulcinéia, que com sua amizade, companheirismo e compreensão, se faz um motivo para minha existência.

A meus filhos, Kênia, Tobias e Bianca, que me fazem perceber a importância da minha existência.

Aos meus familiares, irmãos, primos, cunhados e tios por propiciar um ambiente harmonioso, aspecto importantíssimo para a nossa estabilidade emocional.

A Dênia, grande amiga, que me emprestou o notebook e facilitou muito a estruturação de meu tempo para a escrita desta tese.

A Dra. Rejane Farias e ao Dr. Renato que cederam os resultados de diversos exames de tomografia computadorizada, permitindo a realização dos experimentos desta tese.

Ao grupo de pessoas que embarcaram neste desafio comigo, propiciando-nos momentos inesquecíveis para nossas vidas. São eles: Rubens, Juliano, Cleiber, João Manoel, Élbio, Arquimedes, Lenza e Marco Aurélio.

Aos amigos que me propiciaram algumas discussões, as vezes incompreensíveis, mas necessárias à estruturação de uma contribuição científica. São eles: Marcelo Fernandes, Marco Aurélio Batista, Sirlon Diniz, Márcio Greyck e, novamente, Dulcinéia.

A Faculdade Alves Faria, nas pessoas de Sr. José Alves Faria, Nelson de Carvalho Ramos, Danúsia Arantes e Roberto Pauseiro pelo apoio e compreensão, facilitando muito a construção desta tese.

E, finalmente, as minhas orientadoras, Profa. Dra. Edna Lúcia Flôres e Profa. Dra. Célia Aparecida Zorzo Barcelos, e ao Prof. Dr. Díbio Leandro Borges, que mostraram-me a importância e o prazer da prática do pensamento científico.

Resumo

A visualização de imagens resultantes de exame de tomografia computadorizada (TC) em 3D é um fator importante para o aumento da precisão nos diagnósticos médicos e, conseqüentemente, na eficácia dos tratamentos. Atualmente existem diversos produtos no mercado, que fazem uso de várias técnicas existentes para apresentação de imagens tomográficas em 3D. Contudo, para se obter maior suavidade e precisão nos contornos das estruturas visualizadas em 3D, utiliza-se equipamentos capazes de produzir fatias paralelas do corpo humano muito próximas uma das outras, aumentando a exposição dos pacientes aos raios X. Um método de interpolação de fatias resultantes de exame de TC que forneça bons resultados, pode reduzir a incidência de raios X no paciente, pois esse método pode recuperar a curvatura das estruturas sem a necessidade de uma grande proximidade entre as fatias. Este trabalho propõe um método para a interpolação de imagem em 3D, formada pela justaposição de fatias de resultados de exames de tomografia computadorizada. O objetivo desse método é obter contornos suaves e precisos, melhorando os processos de visualização em 3D. Para isso, esta tese propõe a divisão do processo de interpolação em duas etapas. Na primeira etapa obtém-se uma representação inicial da imagem em 3D composta por fatias reais e por fatias denominadas de fatias virtuais iniciais e, na segunda etapa, restaura-se essas estruturas geradas com um processo de retoque de imagem em 3D. Este trabalho propõe também um método para obtenção da fatia virtual inicial e dois métodos diferentes para a realização do passo de retoque da imagem em 3D resultante da justaposição das fatias reais e virtuais iniciais. Esses métodos são o prolongamento de linhas nas fatias transversais e transporte e difusão de informações. Ambos os métodos utilizam a teoria de equações diferenciais. O método de transporte e difusão de informações demonstrou melhores resultados do que outro método proposto neste trabalho, além de obter melhores resultados do que os métodos de interpolação linear e Goshtasby e outros [1] implementados neste trabalho. Comparações visuais e comparações numéricas utilizando a correlação estatística, a PSNR e a distância de Hausdorff [2] foram realizadas para se obter essas conclusões. O método de transporte e difusão de informações é capaz de gerar contornos mais suaves e precisos que esses outros métodos testados. Além dessa contribuição principal, este trabalho também desenvolveu um KIT para a construção de aplicações visualizadoras de tomografias computadorizadas em 2D e em 3D.

Palavras-chaves: Interpolação de imagem em 3D, transporte e difusão de informações, visualização de tomografia computadorizada, fatia virtual inicial, retoque de imagem em 3D.

Abstract

The visualization of a 3D image obtained from computerized tomography examinations has shown itself to be an important factor for increasing the quality of medical diagnoses and, consequently, treatment efficacy. There already exist on the market, several visualization softwares, which use different techniques to show the 3D tomography image. However, to show a high quality 3D image, sophisticated devices must be used to obtain slices, close to one another, thus increasing the incidence of X-ray given to the patient. An interpolation slice method which resulted from the TC examination produces good results, and is able to reduce the X-ray incidence upon the patient. This method must reconstruct the curvature from the patient's internal structures without using slices in close proximity. This work proposes a method of 3D image interpolation, composed of a juxtaposition of the slices from CT examination results. The goal of this method is to increase the quality of 3D visualization through the production of sharp and precise structure contours. This thesis proposes the division of the interpolation method into two steps. In the first step, the goal is to obtain an initial representation of the image in 3D, which is composed of real slices as well as virtual slices which are referred to in this work as initial virtual slices. In the second step, the empty spaces of the structure are recovered by the 3D image inpainting process. This work also proposes a method to obtain the initial virtual slice and two different methods for inpainting the 3D image. These inpainting methods are the transversal slice line prolongation method and the transportation and diffusion of information. Both methods use the differential equation theory. The transportation and diffusion of information method shows better results than other methods proposed in this work, besides this, this method presents better results than the linear interpolation and Goshtasby et al. [1] methods also implemented in this work. Visual and numerical comparisons are used to obtain this conclusion. The numerical measures used are statistical correlation, the PSNR and the Hausdorff distance [2]. The transportation and diffusion of information method shows itself able to produce better results than all the other tested methods. Besides this principal contribution, this work also developed a KIT to implement 2D and 3D CT visualize applications.

Key-words: 3D Image Interpolation, Transport and Diffusion of Information, Computerized Tomography Visualization, Initial Virtual Slice, 3D image Inpainting.

Sumário

Lista de Figuras

Lista de Tabelas

1	Introdução	p. 1
1.1	– Introdução	p. 1
1.2	– Detalhamento do Problema	p. 7
1.2.1	– Preenchimento de fatias	p. 8
1.2.2	– Contexto para realização do retoque	p. 9
1.2.3	– Performance computacional	p. 11
1.2.4	– Objeto de investigação	p. 12
1.3	– Motivação	p. 12
1.4	– Estrutura da Tese	p. 13
1.5	– Considerações Finais deste Capítulo	p. 15
2	Captura, Armazenamento e Visualização de Imagens de Tomografia Computadorizada	p. 16
2.1	– Introdução	p. 16
2.2	– Como surgiu a Tomografia Computadorizada	p. 16
2.3	– A Tomografia Computadorizada	p. 18
2.4	– O Padrão DICOM	p. 20
2.5	– Visualização de Imagens em 2D	p. 21
2.6	– Visualização de Imagens em 3D	p. 23

2.6.1 – Representação matemática de imagem em 2D e em 3D	p. 24
2.6.2 – Algoritmo Marching Cubes	p. 24
2.6.3 – Visualização direta da imagem em 3D	p. 27
2.6.3.1 – Algoritmo de Ray-Casting	p. 27
2.6.3.2 – Funções de transferência	p. 29
2.6.3.3 – Visualização direta em toda imagem em 3D	p. 31
2.6.4 – Bibliotecas para a visualização em 3D	p. 35
2.6.4.1 – O Open/GL	p. 35
2.6.4.2 – O visualization toolkit (VTK)	p. 36
2.7 – O KIT de Visualização de Imagens de Tomografia Computadorizada (KVICT)	p. 39
2.8 – Considerações Finais deste Capítulo	p. 41
3 Interpolação de Imagens em 3D	p. 42
3.1 – Introdução	p. 42
3.2 – Interpolação	p. 43
3.2.1 – Interpolação de Imagens em 2D	p. 44
3.2.1.1 – A função interpoladora ideal	p. 45
3.2.1.2 – Algumas técnicas de interpolação de imagens em 2D	p. 46
3.3 – Interpolação de Imagens em 3D	p. 52
3.3.1 – Reconstrução da estrutura em 3D	p. 53
3.4 – Exemplos de Técnicas de Interpolação de Imagens em 3D	p. 54
3.4.1 – Interpolação linear	p. 54
3.4.2 – Interpolação de Goshtasby e outros	p. 55
3.5 – Outras Abordagens em Interpolação de Imagens em 3D	p. 58
3.6 – Considerações Finais deste Capítulo	p. 60

4	Equações Diferenciais Aplicadas a Processamento de Imagens	p. 61
4.1	– Introdução	p. 61
4.2	– Equações Diferenciais Parciais	p. 62
4.3	– Aplicação de Equações Diferenciais	p. 63
4.3.1	– A equação do calor	p. 63
4.4	– Equações Diferenciais e Processamento de Imagem	p. 64
4.4.1	– Aplicações de equações diferenciais a processamento de imagem	p. 65
4.4.1.1	– Detecção de bordas	p. 65
4.4.1.2	– Morfologia matemática	p. 66
4.4.1.3	– Difusão	p. 67
4.4.1.4	– Suavização com conservação de bordas	p. 71
4.4.1.5	– Retoque de imagens (<i>inpainting</i>) usando EDP	p. 74
4.4.1.6	– Tratando funções em um domínio discreto	p. 79
4.5	– Considerações Finais deste Capítulo	p. 83
5	Obtenção de uma Fatia Virtual Inicial com e sem Suavização da Imagem a Partir de Duas Fatias Reais de Tomografia Computadorizada Utilizando Equações Diferenciais	p. 84
5.1	– Introdução	p. 84
5.2	– Fatia Virtual Inicial	p. 85
5.3	– Similaridade Entre as Fatias Consecutivas	p. 86
5.3.1	– Determinação do melhor coeficiente K	p. 90
5.4	– Suavização de Fatias com Conservação de Bordas	p. 97
5.5	– Fatia Virtual Inicial com Suavização Prévia das Fatias Reais	p. 102
5.6	– Suavização de Fatias com Conservação de Bordas em 3D	p. 107

5.6.1 – Geração da fatia virtual inicial após a suavização em 3D	p. 112
5.7 – Conclusões Sobre os Resultados Obtidos	p. 115
6 Interpolação de Imagens 3D Utilizando Inpainting	p. 120
6.1 – Introdução	p. 120
6.2 – Experimento Padrão	p. 121
6.2.1 – Exames utilizados no experimento padrão	p. 122
6.2.2 – Medidas realizadas no experimento padrão	p. 122
6.3 – Retoque Pelo Método de Prolongamento de Linhas	p. 124
6.3.1 – Desenvolvimento do método	p. 124
6.3.1.1 – O algoritmo de inpainting	p. 127
6.3.1.2 – Resultados dos testes realizados	p. 131
6.4 – Interpolação de Imagens Utilizando Transporte e Difusão em 3D	p. 137
6.4.1 – Retoque pelo transporte e difusão em 3D	p. 138
6.4.2 – Resultados Obtidos	p. 142
6.5 – Análise Comparativa dos Métodos Utilizados	p. 146
6.5.1 – Análise visual da reconstituição de todo conjunto	p. 147
6.5.2 – Análise das medidas de similaridade entre as imagens	p. 152
6.5.3 – Primeiro conjunto de dados	p. 152
6.5.4 – Segundo conjunto de dados	p. 154
6.5.5 – Terceiro conjunto de dados	p. 155
6.5.6 – Análise das medidas da distância de Hausdorff	p. 160
6.6 – Tempo de Processamento para Transporte e Difusão	p. 161
6.7 – Análise Visual de Outros Exames de TC	p. 166
6.8 – Conclusões	p. 174
7 Conclusões, Contribuições deste Trabalho e Sugestões para Futuros	

Trabalhos	p. 175
7.1 – Conclusões	p. 175
7.2 – Contribuições deste Trabalho	p. 177
7.3 – Sugestões para Futuros Trabalhos	p. 178
7.3.1 – Para o processo de interpolação de imagens	p. 178
7.3.2 – Sugestões para a obtenção de fatias virtuais iniciais	p. 179
Referências	p. 180

Lista de Figuras

1.1	Aparelho de Tomografia Computadorizada	p. 2
1.2	Visualização em 3D da Imagem Gerada pelo Método Proposto	p. 4
1.3	Fatias Consecutivas de um Exame de Tomografia	p. 4
1.4	Fatias Justapostas para Reconstrução	p. 8
1.5	Semelhança entre Fatias Consecutivas	p. 9
1.6	Fatias Idênticas com Diferentes Interpretações	p. 10
1.7	Imagens Interpoladas a partir de Contexto Local	p. 11
2.1	Experimentos com Raios-X	p. 17
2.2	Primeiro Tomógrafo Criado	p. 18
2.3	Resultado Impresso de uma Tomografia Computadorizada	p. 22
2.4	Visualizador de Imagem no Formato DICOM - AMIDE	p. 23
2.5	Relação de Cubos Utilizados no Algoritmo de Marching Cubes	p. 25
2.6	Resultados de Visualização Direta Impresso em Volume	p. 28
2.7	Esquema de Geração de Quadro de uma Imagem no Algoritmo Ray-Casting	p. 29
2.8	Histograma Utilizado para definir Tabela de uma Função de Transferência	p. 31
2.9	Exemplo de Uso de Funções de Transferência	p. 32
2.10	Elementos envolvidos na Obtenção da Reflexão da Luz	p. 32
2.11	Figuras com e sem Efeito de Sombra	p. 34
2.12	Trecho de Código OpenGL e Resultado	p. 37
2.13	Elementos Principais do VTK	p. 38
2.14	Estrutura de Classes do KIT Desenvolvido	p. 40
3.1	Uso de Interpolação de Imagem - Distorção	p. 44

3.2	Interpolação Vizinho Mais Próximo	p. 47
3.3	Aproximações da Interpolação Vizinho Mais Próximo	p. 47
3.4	Interpolação Bilinear - Preparação	p. 49
3.5	Interpolação Bilinear - Resultados	p. 49
3.6	Aproximações da Interpolação Linear	p. 50
3.7	Aproximações da Interpolação Cúbica	p. 51
3.8	Comparação de Resultados de Interpolação 2D	p. 52
3.9	Resultado de um Processo de Interpolação de Imagens	p. 54
3.10	Resultado de Interpolação 3D - Linear	p. 55
3.11	Resultado de Interpolação 3D - Goshtasby	p. 57
3.12	Resultado da comparação entre a Interpolação Linear e Goshtasby . . .	p. 58
4.1	Resultado de um Processo de Detecção de Bordas	p. 66
4.2	Resultado de um Processo de Erosão e Dilatação	p. 68
4.3	Resultado do Processo de Suavização	p. 72
4.4	Resultado de um Processo de Erosão e Dilatação	p. 74
4.5	Vetor Normal e de linha Isocrômica	p. 78
4.6	Representação do Domínio Discreto de uma Imagem	p. 81
5.1	Semelhança entre Fatias	p. 86
5.2	Semelhança entre Fatias em Métodos Distintos	p. 88
5.3	Semelhança entre Fatias em Métodos Distintos	p. 90
5.4	Semelhança entre Fatias - Imagens Originais	p. 91
5.5	Similaridade e Correlação por K	p. 94
5.6	Semelhança entre Fatias - Resultados	p. 96
5.7	Detalhes para Imagens não Similares - Resultados	p. 97
5.8	Segmentação com Conservação de Bordas	p. 101
5.9	Similaridade e Correlação por K	p. 104

5.10	Semelhança entre Fatias - Resultados	p. 105
5.11	Detalhes em Fatias Não Suavizadas e Suavizadas	p. 106
5.12	Similaridade e Correlação por K	p. 114
5.13	Semelhança entre Fatias - Resultados	p. 116
5.14	Detalhes Ampliados de Fatias Virtuais Iniciais pelos Três métodos . . .	p. 117
6.1	Fatias para o Experimento Padrão	p. 122
6.2	Fatia Virtual Inicial Restaurada	p. 125
6.3	Seções Transversais da imagem em 3D	p. 125
6.4	Ampliação de Áreas Vazias em uma Seções Transversal	p. 126
6.5	Curva Chegando ao Domínio de Restauração	p. 129
6.6	Retoque em imagens Normais	p. 132
6.7	Seções Transversais Restauradas	p. 133
6.8	Fatia Virtual Restauradas - Método de Prolongamento de Linhas . . .	p. 134
6.9	Visão 3D - Óssea - Método Prolongamento de Linhas	p. 135
6.10	Visão 3D - Externa - Método de Prolongamento de Linhas	p. 136
6.11	Vetor normal e isocrômico em uma Estrutura	p. 138
6.12	Fatia Virtual Restauradas - Método de Transporte Difusão	p. 143
6.13	Fatia Reais e Virtuais pelo Método de Transporte Difusão	p. 143
6.14	Visão 3D - Óssea - Método Transporte e Difusão	p. 144
6.15	Visão 3D - Externa - Método de Transporte e Difusão	p. 145
6.16	Resultado para comparação visual dos métodos testados - primeiro paciente	p. 149
6.17	Resultado para comparação visual dos métodos testados - segundo paciente	p. 150
6.18	Resultado para comparação visual dos métodos testados - terceiro paciente	p. 151
6.19		p. 157
6.20		p. 158
6.21	Gráfico de comparação entre os métodos - Primeiro paciente	p. 159

6.22	Aplicação Parcial de Transporte e Difusão	p. 164
6.23	Aplicação Parcial de Transporte e Difusão	p. 165
6.24	Gráfico de tempo do método de Transporte e Difusão	p. 166
6.25	Resultado para comparação visual dos métodos testados - primeiro conjuntop.	168
6.26	Resultado para comparação visual dos métodos testados - segundo conjuntop.	170
6.27	Resultado para comparação visual dos métodos testados - terceiro conjuntop.	172
6.28	Resultado para comparação visual dos métodos testados - quarto conjuntop.	173

Lista de Tabelas

1.1	Recursos computacionais utilizados neste trabalho	p. 12
5.1	Comparação Entre Métodos de Verificação de Similaridade Entre Fatias	p. 88
5.2	Resultado do Processo de Avaliação de Similaridade Entre Fatias	p. 93
5.3	Suavização com Conservação de Bordas - Diversos Parâmetros	p. 100
5.4	Resultado do Processo de Avaliação de Similaridade Entre Fatias Suavizadasp.	103
5.5	Suavização com Conservação de Bordas em 3D	p. 112
5.6	Resultado do Processo de Avaliação de Similaridade entre Fatias Suavizadasp.	113
6.1	Tempo de Execução dos métodos	p. 148
6.2	Correlação estatística para os métodos implementados	p. 153
6.3	Correlação estatística para os métodos implementados	p. 155
6.4	Correlação estatística para os métodos implementados	p. 156
6.5	Distância de Hausdorff para os Métodos Implementados	p. 161
6.6	Tempo de execução do algoritmo de transporte e difusão	p. 165

1 Introdução

1.1 — Introdução

Desde que as primeiras civilizações começaram a se desenvolver a mais de 4.000 anos atrás, a atividade da medicina também apareceu [3]. Desde então, a prática da medicina subdividiu-se em duas etapas principais: a determinação do mal que afligia o paciente, o diagnóstico, e um processo de tratamento que pudesse fornecer a ele o retorno à saúde.

A forma de obtenção de um diagnóstico pode ser realizada a partir do relato dos sintomas que o paciente está sentindo em decorrência do mal; pelo exame de amostras de tecidos, secreções ou dejetos dele e pela análise da morfologia dos órgãos externos e internos do paciente. Nesta última, busca-se perceber, seja pelo toque ou por exames que mostram uma visão dos órgãos do corpo do paciente, qual o órgão responsável pelos sintomas relatados.

A anatomia é a área da medicina que estuda e mapeia as estruturas internas do corpo humano [4]. Ela ampara a tarefa do diagnóstico, pois o órgão que apresenta uma discrepância da morfologia padrão, pode ser o causador do mal que acomete um paciente. Entretanto, um problema ainda persiste: a dificuldade de analisar as estruturas internas do corpo humano. A aplicação de técnicas invasivas causam, as vezes, ao paciente males mais graves que o mal diagnosticado com a invasão [5, 6, 7]. Assim, a busca de processos não invasivos para a obtenção de uma visão das estruturas internas do corpo humano é um dos principais objetos de estudo das áreas médica, computação, engenharia e outras [7].

No início da década de 70 Hounsfield [8, 9], construiu o primeiro aparelho de tomo-

grafia computadorizada (TC), ilustrado na Figura 1.1 (a). O resultado obtido por esse aparelho, apesar dos raios X incidirem frontalmente no paciente, como mostrado na Figura 1.1 (b), é a visão de um corte transversal, fatia, de uma parte do corpo humano, como pode-se observar na Figura 1.1 (c). O conjunto de fatias permite uma visão dos órgãos sem a necessidade de abordagens invasivas, fornecendo ao médico condições de visualizar detalhes minuciosos do corpo do paciente e tornar o diagnóstico mais preciso.

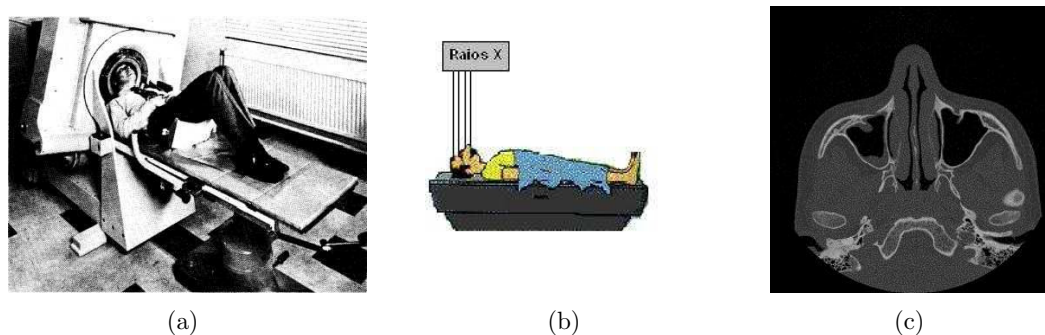


Figura 1.1: (a) Primeiro aparelho de tomografia computadorizada; (b) raios X atravessando frontalmente o corpo do paciente; (c) uma fatia transversal de um exame de TC.

Atualmente, as imagens resultantes dos exames de tomografia computadorizada podem ser armazenadas em computadores para posteriormente serem visualizadas por softwares específicos. Os visualizadores como eFilm [10], SLICE [11], BLOX [12] e o AMIDE [13], oferecem ao médico diversos recursos para a visualização e análise das imagens obtidas nos exames, além de permitir o armazenamento e a recuperação dos resultados de exames de uma maneira mais ágil. O formato *Digital Imaging and Communications in Medicine* (DICOM) [14], é utilizado na maioria dos aparelhos que produzem esse tipo de exames. Ele foi proposto para permitir o armazenamento tanto da imagem do exame quanto outros dados do próprio exame e do paciente examinado.

Além da visualização em 2D dos exames, existem os softwares para a visualização em 3D. Esses softwares exigem computadores com maior capacidade de processamento. A partir de um conjunto de fatias, normalmente distanciadas em alguns milímetros uma das outras, é possível produzir a visão em 3D, sendo necessário para isso posicionar adequadamente cada fatia e preencher os espaços vazios entre elas até chegar a uma

imagem em 3D.

Potencialmente, a visão em 3D não mostra nenhuma informação a mais do que pode ser visto a partir do conjunto de fatias resultantes de um exame de TC. No entanto, ela apresenta as estruturas de uma forma mais natural, aumentando a confiabilidade do diagnóstico médico. A visão das fatias em 2D exige do médico a capacidade de reconstruir mentalmente a imagem de um órgão com a finalidade de saber se ele está dentro dos padrões.

A qualidade da imagem 3D depende de diversos fatores. Destaca-se como um dos principais, o processo de preenchimento dos espaços vazios entre as fatias: a interpolação de imagens. Atualmente vários métodos de interpolação de imagens são utilizados, entre eles o de Goshtasby e outros [1], Bors e outros [15], Wang e outros [16] e Prado e outros [17], alguns desses métodos são detalhados e implementados no Capítulo 3 deste trabalho. Normalmente eles exigem, para a obtenção de um bom nível de qualidade, fatias muito próximas, que tem como consequência uma maior exposição do paciente à incidência de raios X.

Este trabalho propõe um método de interpolação de imagens [18] que visa obter um contorno adequado das estruturas interpoladas mesmo utilizando fatias relativamente distanciadas. A Figura 1.2 mostra a visualização em 3D dos resultados obtidos pelo método desenvolvido nesta tese.

O método proposto neste trabalho é baseado nas seguintes hipóteses:

1. As estruturas internas do corpo dos seres humanos não possuem variações abruptas em sua forma. Assim, pode-se considerar que as regiões vão modificando-se de maneira gradual, e;
2. As estruturas internas do corpo dos seres humanos obedecem a uma harmonia na composição de seus limites [1].

A Figura 1.3 apresenta três fatias consecutivas obtidas de um exame de tomografia computadorizada. Pode-se observar que as imagens são muito parecidas e todas elas

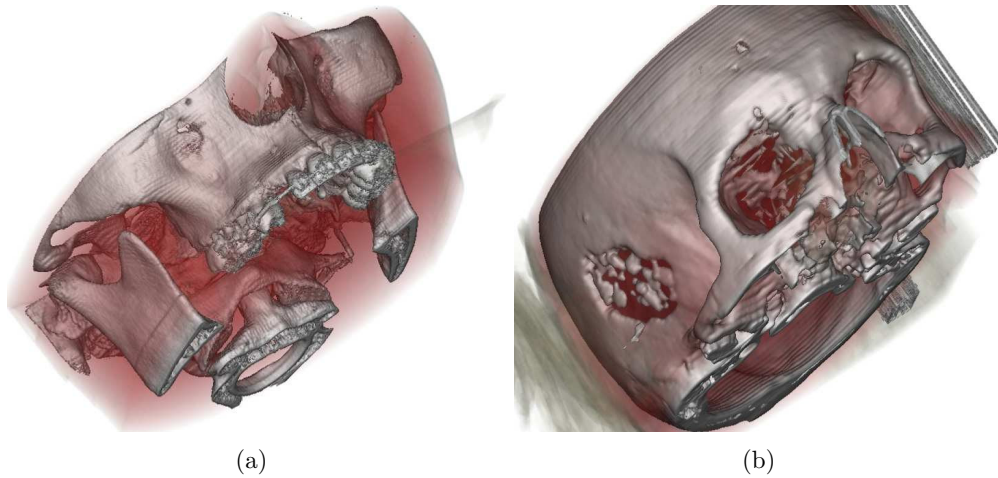


Figura 1.2: (a) e (b) Resultado da visualização em 3D aplicando-se o método desenvolvido neste trabalho.

apresentam quase que as mesmas regiões. Como citado por Goshtasby e outros [1] isso pode ajudar no processo de interpolação.

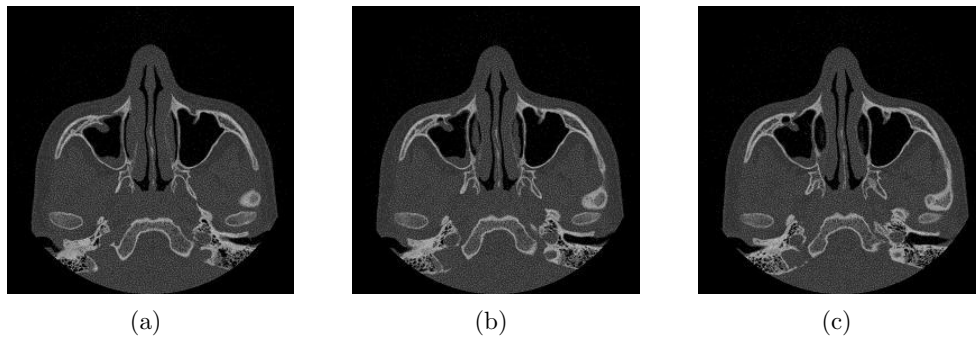


Figura 1.3: Três fatias consecutivas obtidas de um exame de tomografia computadorizada.

Os vários métodos de interpolação de imagens em 3D já existentes não consideram, no processo de interpolação, uma harmonia mais global do contorno da estrutura reconstruída. Eles utilizam normalmente, para a interpolação, somente as informações constantes nas fatias reais consecutivas que limitam o espaço a ser interpolado.

Este trabalho justifica-se pelo fato de que obtém estruturas com contornos mais harmônicos devido à utilização não somente de duas fatias reais consecutivas e sim um contexto maior envolvendo também as fatias vizinhas das fatias reais interpoladas.

Considerando a hipótese de que as estruturas internas do corpo humano obedecem um padrão natural de harmonia em suas formas, a proposta deste trabalho permite que as estruturas possam ser reconstruídas a partir de um número menor de fatias. A consequência do uso de menos fatias no processo são a redução da incidência de radiação de raios X nos pacientes examinados, a diminuição do tempo de realização dos exames e a redução do custo final dos exames. Por outro lado, a manutenção do mesmo número de fatias obtidas atualmente, leva à obtenção de contornos mais suaves e precisos, aumentando a eficácia da análise da estrutura visualizada.

Outra justificativa para este trabalho é que a implementação de algoritmos para a plataforma PC torna mais acessível as aplicações visualizadoras em 3D. Atualmente a grande maioria das aplicações só estão disponíveis para plataformas mais potentes de processamento, aumentando o custo e inviabilizando a utilização dessas aplicações para um número maior de profissionais da saúde.

Um processo de interpolação que utilize de um contexto maior do que duas fatias reais consecutivas, conforme já citado anteriormente neste capítulo, pode ser obtido com o uso de equações diferenciais. Trabalhos como o de Bertalmío e outros [19], Martins [20], Barcelos e outros [21], e Barcelos e Batista [22] utilizam equações diferenciais para transporte de informações de uma região para outra em um processo evolutivo. Na forma discreta, por exemplo, o cálculo de uma derivada terceira em uma imagem em 3D composta pelas fatias justapostas pode envolver informações de três fatias distintas, tornando o domínio de atuação do processo mais abrangente do que três fatias.

Duas observações a respeito do uso de equações diferenciais em processamento de imagem motiva a sua utilização neste trabalho: a primeira é a observação citada por Gabor: “A diferença entre a imagem borrada e a imagem original é proporcional ao seu laplaciano” [23], isto é, o processo de borrar uma imagem comporta-se como a equação do calor descrita pela teoria de equações diferenciais [24]. A segunda é a utilização de equações diferenciais como ferramentas para algumas técnicas de retoque de imagens [20, 22].

A grande semelhança entre duas fatias reais consecutivas de um exame de tomografia computadorizada aliada a utilização de técnicas de retoque de imagem [19, 20, 22], permite tratar o processo de interpolação de imagens como um processo de retoque de imagem. Essa grande semelhança entre as fatias permite um preenchimento inicial das áreas vazias entre as fatias e, seguido de um processo de retoque, pode-se realizar o completo preenchimento das informações não constantes na imagem.

Com o objetivo de reduzir a área de retoque entre duas fatias consecutivas, e consequentemente aumentar o desempenho computacional do mesmo, o método proposto cria diversas fatias virtuais equidistantes entre duas fatias reais a partir das similaridades existentes entre elas. Neste trabalho essas fatias virtuais parcialmente preenchidas são denominadas de fatias virtuais iniciais. O Capítulo 5 desta tese detalha o processo de criação dessas fatias.

Assim, o processo de interpolação das fatias reais será realizado em duas etapas. A primeira etapa é a construção de uma imagem em 3D composta pelas fatias reais justapostas e por fatias virtuais iniciais contendo algumas regiões não preenchidas. E a segunda etapa, consiste em retocar a imagem em 3D com a finalidade de preencher os espaços sem informação preenchidos, tornando essa imagem completa. O uso da teoria de equações diferenciais parciais (EDPs) ocorre nas duas etapas, sendo que a segunda etapa, a de retoque da imagem em 3D, é completamente fundamentada nas EDPs. Os resultados obtidos dessa abordagem estão apresentados no Capítulo 6 desta tese. Também são realizadas comparações dos resultados obtidos nesta tese com as técnicas consagradas na literatura, tais como: Goshtasby e outros [1] e interpolação Linear.

O processo de retoque realizado após a criação e o preenchimento da imagem em 3D com as fatias virtuais iniciais é realizado por duas abordagens distintas. Sendo elas:

1. Prolongamento de linhas [20]: consiste em analisar a borda da região a ser retocada e determinar qual é a melhor linha a ser prolongada para dentro do domínio de retoque. Repete-se esse processo até o preenchimento total da região a ser retocada. A região de retoque é obtida por seções transversais da imagem em 3D, isto é,

imagens em 2D;

2. Transporte e difusão de informações [19, 22]: o método consiste em efetuar alternadamente duas ações na região de retoque, uma de transporte de informações para a região de retoque, seguida da difusão das informações contidas nessa região. Repete-se essa iteração um certo número de vezes até se obter o completo retoque da região. A região de retoque é toda porção indefinida da imagem em 3D.

Os métodos de prolongamento de linhas [20] e transporte e difusão de informações [19, 22] propostos são implementados e comparados por análise visual e numérica. Na análise numérica são utilizadas a correlação estatística, a medida da relação sinal/ruído de pico (PSNR) e a distância de Hausdorff [2]. O Capítulo 6 detalha o processo de desenvolvimento desses métodos e o resultado das comparações efetuadas.

Este capítulo mostra um resumo das atividades envolvidas nesta tese. Descreve as hipóteses, as justificativas e os objetivos deste trabalho. Explicita detalhes dos problemas tratados nesta tese. Apresenta a motivação para realizá-la. E, finalmente, apresenta uma descrição de cada capítulo que compõe este trabalho.

1.2 – Detalhamento do Problema

Caracteriza-se como a principal contribuição deste trabalho o desenvolvimento de um método que obtenha uma imagem em 3D de um conjunto de fatias de um exame de tomografia computadorizada (TC) e que consiga aliar:

- uma boa qualidade dos contornos para as estruturas constantes na imagem tridimensional que representa as estruturas internas do corpo de um paciente;
- a capacidade de obter uma imagem em 3D (com menos fatias) para que seja possível minimizar a quantidade de raios X recebidos pelo paciente no processo de obtenção das imagens dessas estruturas.

1.2.1 – Preenchimento de fatias

Com o objetivo de obter a imagem em 3D, deve-se considerar que as fatias obtidas da TC contém simplesmente a visão de um corte transversal em uma determinada posição do corpo do paciente. Todas as fatias que compõem o exame quando justapostas umas às outras, conforme mostrado na Figura 1.4, levam a uma imagem em 3D com áreas vazias entre as elas, isto é, faltam fatias para preencher corretamente a imagem.

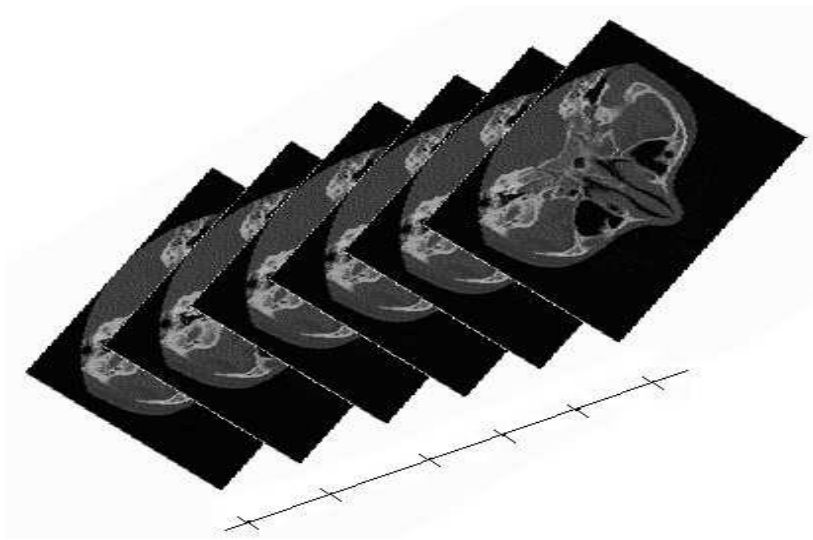


Figura 1.4: Conjunto de fatias reais consecutivas de um exame de TC justapostas umas às outras, demonstrando os espaços vazios entre cada fatia real.

O processo de interpolação visa completar a imagem em 3D, obtendo-se fatias virtuais entre as fatias reais obtidas do exame. Esse processo de obtenção das fatias virtuais é realizado, neste trabalho, em duas etapas com o objetivo de transferir à fatia virtual as informações das fatias reais adjacentes. Inicialmente, para obter uma fatia virtual inicial entre duas fatias reais consecutivas, deve-se considerar que quanto maior for o nível de semelhança entre essas fatias reais, maiores serão as regiões preenchidas na fatia virtual inicial. Isso ocorre devido ao processo de preenchimento da fatia virtual utilizar uma etapa de comparação entre as fatias reais consecutivas seguida do preenchimento das fatias virtuais nas regiões similares. A primeira parte do processo de interpolação utilizado neste trabalho pode gerar diversas fatias virtuais não preenchidas completamente, isto é, fatias virtuais iniciais.

Apesar da grande semelhança visual entre duas fatias reais consecutivas de um determinado exame de TC, como mostrado na Figura 1.5 (a) e (b), pode ocorrer ruído no processo de aquisição e reconstrução da imagem desse exame. Tais diferenças diminuem a semelhança entre as fatias reais consecutivas, como mostrado na Figura 1.5 (c). Nesta figura, a cor vermelha indica os *pixels* cujos valores de intensidade não são coincidentes entre as duas imagens. Embora possa ser percebido em uma análise visual uma grande semelhança, ao comparar *pixel a pixel* essa semelhança é muito menor.

Este trabalho utiliza no método desenvolvido processos de suavização de imagens utilizando equações diferenciais baseado nos trabalhos de Barcelos e outros [21]. O objetivo dessa suavização é a redução de ruídos das fatias para aumentar a similaridade entre elas. Além da suavização, propõe-se nesta tese também um processo utilizando informações estatísticas para determinar as similaridades entre as fatias, pois desta forma a similaridade entre duas fatias reais consecutivas é melhor reconhecida. A Figura 1.5 (d) mostra os resultados obtidos com esse processo, reduzindo significativamente as regiões não similares.

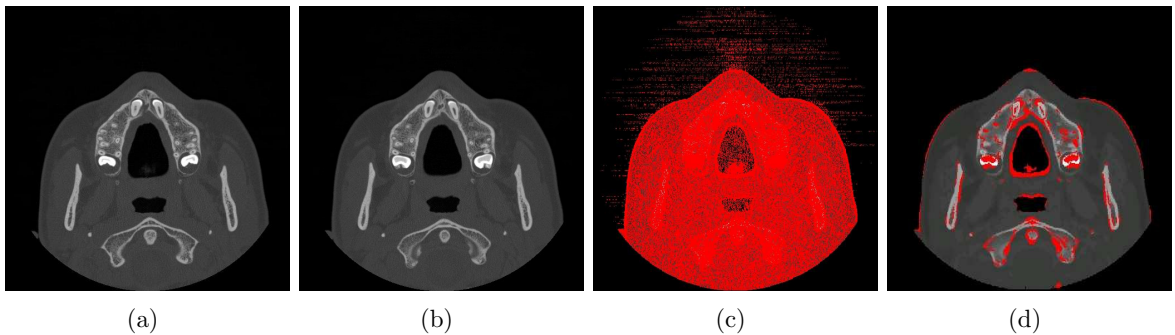


Figura 1.5: (a) e (b) Semelhança entre duas fatias consecutivas; (c) diferença *pixel a pixel* entre as duas fatias consecutivas das imagens das letras (a) e (b) desta figura; (d) diferença utilizando o processo desenvolvido neste trabalho entre as duas fatias consecutivas das imagens das letras (a) e (b) desta figura.

1.2.2 – Contexto para realização do retoque

Para o processo de obtenção das fatias virtuais não se deve considerar apenas as fatias reais consecutivas entre elas, pois isso limita o contexto da origem das informações

necessárias para a interpolação devido ao uso de um contexto restrito para o processo. O resultado de tal abordagem pode levar a determinação de contornos incorretos que não representam a curvatura real da estrutura retratada nas fatias, conforme mostrado nas Figuras 1.6 e 1.7.

Somente em um contexto local mais abrangente, isto é, considerando mais do que duas fatias reais, é que se pode efetuar um processo de interpolação mais adequado, pois dessa forma a curvatura da estrutura pode ser obtida com maior precisão. Para ilustrar a diferença dos resultados do retoque considerando por um lado, duas fatias reais consecutivas, e por outro lado um contexto que considere mais do que essas duas fatias, a Figura 1.6 (a) mostra duas estruturas com formas bastante diferentes. Retirando quatro fatias dessas estruturas, conforme pode-se observar na Figura 1.6 (b), percebe-se que dependendo da posição onde foram retiradas, ambas as estruturas produzem a mesma imagem como mostrado nas fatias 2 e 3 dessa figura, apesar de terem uma forma geral diferente.

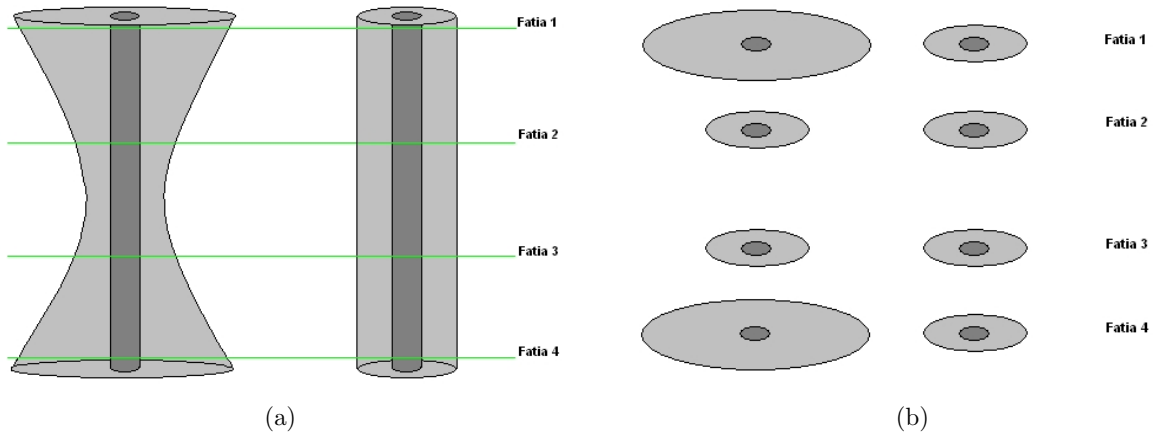


Figura 1.6: (a) Duas estruturas com formas diferentes; (b) fatias reais consecutivas idênticas geradas das duas estruturas diferentes.

Pode-se observar na Figura 1.6 (a) que a imagem com a forma de ampulheta possui uma curvatura que em um processo de interpolação considerando somente as fatias reais consecutivas 2 e 3, ocorrerá a geração de estruturas incorretas, como pode-se verificar na Figura 1.7 (a). Entretanto, na forma cilíndrica pode-se observar na Figura 1.7 (b) que o processo de interpolação obtém bons resultados.

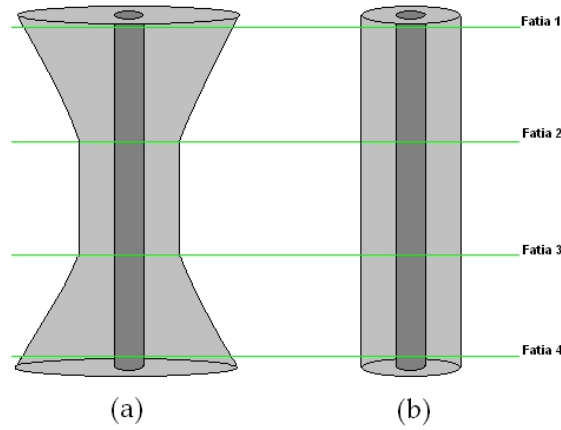


Figura 1.7: Resultado da interpolação das fatias obtidas das duas estruturas diferentes, utilizando o contexto entre duas fatias reais consecutivas.

Considerar somente duas fatias reais consecutivas no processo de interpolação, sem considerar as outras fatias reais vizinhas a essas, não permite solucionar o problema de reconstrução da estrutura original corretamente. O método de interpolação neste caso obterá resultados diferentes da estrutura real.

1.2.3 – Performance computacional

Outro fato a ser considerado no processo de reconstrução em 3D é a baixa capacidade dos processadores atuais para tratarem, em tempo real, a quantidade de informações que são geradas. Atualmente uma TC gerada com 512×512 *pixels*, quando ela é obtida em 3D, pode produzir um cubo $512 \times 512 \times 512$ *pixels* com um total de 134.217.728 *pixels* a serem processados, isto inviabiliza a maioria das aplicações atuais.

Assim, o método desenvolvido neste trabalho concentrou o consumo computacional de tempo no processo de interpolação, permitindo ao usuário a visualização da imagem em 3D de maneira rápida e natural. Os recursos computacionais utilizados neste trabalho são mostrados na Tabela 1.1 deste capítulo.

Tabela 1.1: Recursos Computacionais utilizados neste trabalho

Recurso Computacional	Especificação do Recurso
CPU	Intel Centrino 1.73Ghz
Memória RAM	1 Gbytes, 795Mhz
HD	80Gb, 7200rpm
Placa de Vídeo	Mobile Intel(R) 915GM/GMS,910GML, 128Mbytes
Sistema Operacional	Windows XP home Edition
Ambiente de Desenvolvimento	Microsoft Visual Studio 6.0, Visual C++

1.2.4 – Objeto de investigação

As imagens utilizadas nos testes realizados neste trabalho são imagens obtidas de um aparelho de tomografia computadorizada do Instituto Goiano de Radiologia (IRG) [25] e do Centro de Reabilitação e Readaptação Dr. Henrique Santillo [26]. Para a realização dos testes foram selecionados um conjunto de imagens da mandíbula, da região dos olhos e da porção superior do crânio de alguns pacientes, além de vários outros exames retratando diversas porções do corpo, cujos resultados estão apresentados no Capítulo 6 desta tese.

As imagens utilizadas nos testes realizados neste trabalho estão armazenadas no padrão *DICOM* e elas são convertidas para o padrão *bitmap* (bmp) antes da realização do processo de interpolação. O uso do padrão bmp foi escolhido por manter a matriz de pontos da imagem sem nenhuma compactação, aumentando o desempenho computacional do processo como um todo, embora esse padrão consuma maior memória real. Com o objetivo de avaliar o desempenho computacional dos algoritmos desenvolvidos neste trabalho, as imagens utilizadas são 512 x 512 *pixels*.

1.3 – Motivação

A realização deste trabalho permitiu o aumento do domínio de conhecimento em várias áreas, tais como:

- padrões de armazenamento de imagens médicas;
- equações diferenciais aplicadas a processamento de imagem;

- técnicas de segmentação de imagens utilizando morfologia matemática;
- computação gráfica, e;
- projeto orientado a objetos para aplicações computacionais.

Os conhecimentos adquiridos neste trabalho, permitirão o desenvolvimento de uma série de outros trabalhos ligados a visualização de imagens médicas, a computação gráfica e processamento de imagem. Os resultados de tais trabalhos poderão contribuir de alguma forma para o progresso dos pesquisadores envolvidos e para a área de pesquisa propriamente dita.

A abordagem proposta neste trabalho para a interpolação de imagens utilizando a teoria de equações diferenciais insere uma nova forma de suavização para as fatias de tomografia computadorizada, além de permitir a interpolação da estrutura considerando um contexto mais global. A interpolação será tratada como um processo de retoque de imagens em 3D. Essa originalidade, aliada à possibilidade de trazer bons resultados aos profissionais médicos, foram as maiores motivações para a realização desta tese.

1.4 – Estrutura da Tese

Neste trabalho, a partir do estudo e da apresentação do que é a tomografia computadorizada, dos processo de visualização, de algumas técnicas existentes de interpolação e do uso de equações diferenciais em processamento de imagem, segue a proposta de interpolação de imagem em 3D em duas etapas: geração das fatias virtuais iniciais e retoque na imagem em 3D, ambos os passos utilizam a teoria de equações diferenciais. Os conteúdos descritos acima encontram-se distribuídos nos próximos capítulos da seguinte forma:

O Capítulo 2 mostra uma visão geral do que é a tomografia computadorizada e o seu histórico. Apresenta o padrão *DICOM* para o armazenamento digital de imagens tomográficas e alguns visualizadores bidimensionais e tridimensionais em 2D e em 3D existentes no mercado. Descreve algumas técnicas para o processo de visualização em 3D e apresenta algumas bibliotecas para a construção de aplicações visualizadoras. E

finalmente, são realizadas as considerações finais desse capítulo.

O Capítulo 3 apresenta os principais métodos de interpolação de imagens, inicialmente para imagens em 2D em seguida para imagens em 3D. Com o objetivo de fornecer mais informações à proposta de interpolação de imagens em 3D, são mostrados alguns trabalhos que utilizaram métodos de interpolação em 3D em fatias de tomografia computadorizada. E finalmente, são realizadas considerações finais sobre esse capítulo.

O Capítulo 4 mostra uma fundamentação matemática sobre equações diferenciais, apresenta algumas abordagens que utilizam as equações diferenciais parciais aplicadas a processamento de imagem, exemplificando com alguns trabalhos que utilizaram essas abordagens. E finalmente, são realizadas considerações finais sobre esse capítulo.

O Capítulo 5 apresenta um método de obtenção de uma fatia virtual inicial utilizando medidas estatísticas e também usando a suavização da imagem com conservação de bordas utilizando EDPs. Nesse capítulo realiza-se a extensão do processo de suavização com conservação de bordas em 2D para o espaço tridimensional. E finalmente, são mostrados os resultados obtidos nos testes realizados e as conclusões sobre esses resultados.

No capítulo 6 são apresentados dois métodos de restauração de imagens, considerando como ponto de partida a imagem tridimensional formada pelas fatias reais e virtuais iniciais. Esses métodos utilizam o processo de retoque, o primeiro em 2D e o segundo em 3D. São mostrados os resultados obtidos nos testes utilizando-se esses dois métodos, bem como a comparação com outros métodos já existentes. E finalmente, são realizadas conclusões sobre os resultados obtidos nesses testes.

Finalmente, o Capítulo 7 apresenta as conclusões, as contribuições deste trabalho e as sugestões para futuros trabalhos.

1.5 – Considerações Finais deste Capítulo

Este capítulo mostrou um resumo das atividades desenvolvidas nesta tese. Descreveu as hipóteses, a justificativa e os objetivos deste trabalho. Explicitou detalhes dos proble-

mas tratados nesta tese. Apresentou a motivação para realizá-lo. E finalmente, apresentou uma descrição de cada capítulo que compõe esta tese.

O próximo capítulo mostra uma visão geral do que é a tomografia computadorizada, seu histórico e a fundamentação matemática de como se chegou a obtenção de imagens tomográficas. Apresenta o padrão *DICOM* para o armazenamento digital de imagens tomográficas e alguns visualizadores bidimensionais e tridimensionais em 2D e em 3D existentes no mercado. Descreve algumas técnicas para o processo de visualização em 3D e apresenta algumas bibliotecas para a construção de aplicações visualizadoras. E finalmente, são realizadas considerações finais desse capítulo.

2 Captura, Armazenamento e Visualização de Imagens de Tomografia Computadorizada

2.1 — Introdução

Este capítulo mostra uma visão geral do que é a tomografia computadorizada (TC) e o seu histórico. Apresenta o padrão DICOM para armazenamento digital de imagens tomográficas e alguns visualizadores bidimensionais (2D) e tridimensionais (3D) existentes no mercado. Descreve algumas técnicas para o processo de visualização em 3D e apresenta algumas bibliotecas para a construção de aplicações visualizadoras. E, finalmente, são realizadas considerações finais deste capítulo.

Para fundamentar o conteúdo apresentado neste capítulo, são utilizados os conhecimentos de engenharia, matemática e computação. A utilidade da TC sob o ponto de vista médico não é a ênfase deste capítulo. O objetivo é tornar claro cada um dos passos que envolve o domínio de conhecimento para a criação e o uso do recurso da tomografia computadorizada.

2.2 — Como surgiu a Tomografia Computadorizada

No final do século XIX, Rontgên descobriu, por acaso, a existência de um tipo de raio, o raio X, que tinha a capacidade de atravessar o corpo humano e registrar uma projeção (chapa) das estruturas internas do corpo humano. A Figura 2.1 mostra o aparelho utilizado por Rontgên (a) e a imagem obtida com o raio X (b). Essa descoberta permitiu

a observação de estruturas internas do corpo do paciente sem uma abordagem invasiva. No entanto, logo após a descoberta de Rontg en, observou-se uma grande difus o de seu uso, e esse elevado uso permitiu a r pida observ o dos males causados pelos raios X, principalmente nas pessoas que operavam os aparelhos em sua rotina de trabalho.

O raio X, aliado a outras t cnicas de aplica o de contraste, permitiu no decorrer do s culo XX um avan o das t cnicas de diagn stico nos mais diversos ramos da medicina, como ortopedia, neurologia, urologia, pneumologia e outras [27, 28].

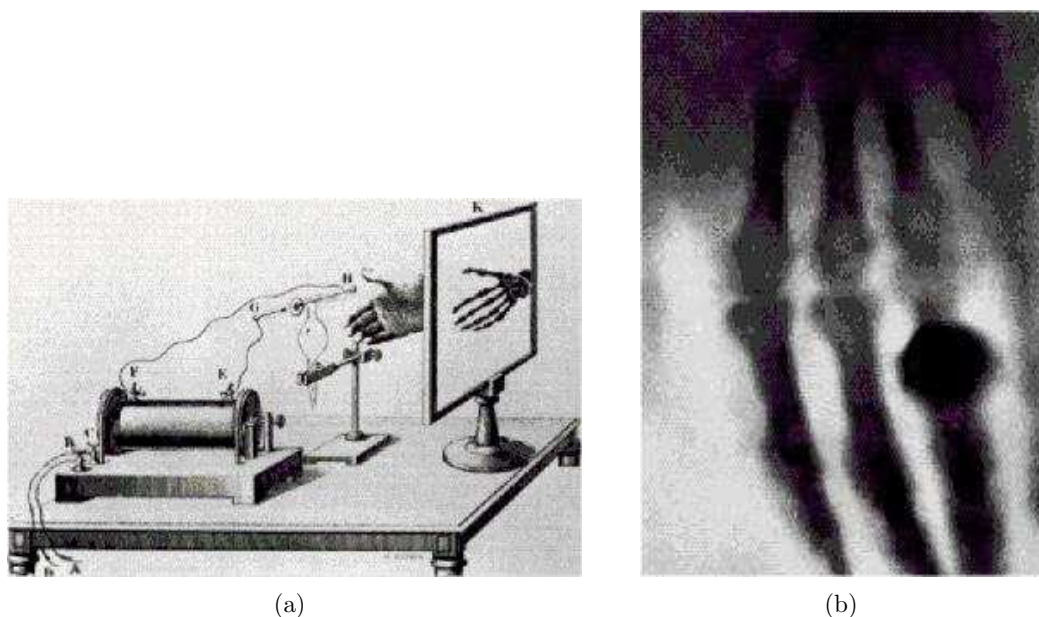


Figura 2.1: (a) Aparelho constru do por Rontg en; (b) chapa resultante do experimento realizado por ele.

Sem os raios X, muito do conhecimento cient fico ainda n o estaria mapeado, mas sabe-se que o seu uso deve ser controlado devido  s conseq  ncias danosas da radia o nos seres humanos.

Alguns anos depois, no in cio do s culo XX, John Radon, um matem tico alem o, demonstrou matematicamente que   poss vel reconstruir uma estrutura em tr s dimens es (3D) a partir de um conjunto infinito de proje  es em duas dimens es (2D) dessa estrutura [29]. Para a  poca, essa prova matem tica teve pouco valor. Em termos de fundamentos

teóricos, as chapas de raios X, que são projeções em 2D, poderiam permitir a reconstrução dos órgãos em 3D aplicando as idéias propostas por Radon. Entretanto, na época, o elevado número de cálculos exigidos para isso tornava essa iniciativa tecnicamente inviável.

2.3 – A Tomografia Computadorizada

Com a evolução dos computadores, dos equipamentos de raios X e utilizando as idéias propostas por Radon, foi possível o surgimento da tomografia computadorizada (TC). A tomografia computadorizada representou para a medicina um avanço tão importante quanto o raio X surgido quase um século antes, proporcionando inclusive o prêmio Nobel de Medicina em 1979 para os seus inventores [8]. A Figura 2.2 mostra o primeiro deles, o protótipo de Hounsfield [30].

O produto do aparelho de Hounsfield é a visão de um corte transversal (fatia) de uma parte do corpo humano. Pela captura dos raios X atenuados ao atravessar o corpo humano e utilizando-se o método de retroprojeção [30][29], pode-se obter as imagens como as mostradas na Figura 1.3 do Capítulo 1 deste trabalho.

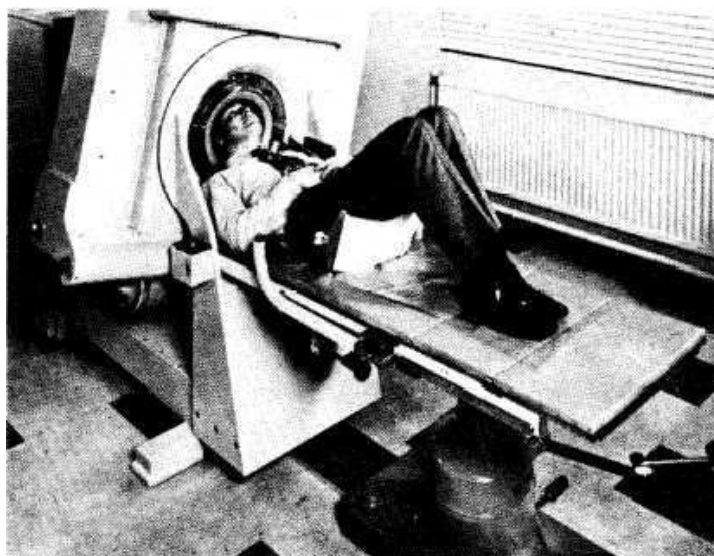


Figura 2.2: Protótipo do tomógrafo criado por Hounsfield.

Antes do protótipo de Hounsfield, houve na década de sessenta algumas outras ini-

ciativas visando a produção de um tomógrafo. Em 1961, o neurólogo William H. Oldendorf, construiu um protótipo do tomógrafo utilizando uma fonte radioativa de iodo e um detector cintilador [31]. Ele utilizou o método da retroprojeção para reconstruir a imagem de uma estrutura feita de pregos. Esse equipamento foi considerado comercialmente inviável. Oldendorf ouviu o seguinte comentário de um fabricante de equipamentos de raios X: “Mesmo funcionando como você sugere, não podemos imaginar um mercado significativo para um aparelho tão caro, que não faz nada além de gerar seções transversais da cabeça” [31].

Também na década de sessenta, o físico Allan M. Cormack, atuando como supervisor do uso de isótopos radioativos em um hospital, percebeu que deveria existir alguma maneira de determinar as não homogeneidades de um tecido a partir das medições realizadas na região exterior. Cormack realizou uma série de estudos teóricos e experimentais sobre o assunto, publicando os resultados em artigos no início da década de sessenta [32], esses artigos tratavam das mesmas idéias de Radon.

Os aparelhos de tomografia computadorizada desde então, tornaram-se um equipamento fundamental na busca de precisão dos diagnósticos médicos. Utilizando outros processos para se obter as projeções, mas mantendo o mesmo processo de reconstrução de imagens a partir de projeções, vários outros recursos surgiram para a análise da estrutura interna do corpo humano. Entre esses vários outros recursos, pode-se citar a ressonância magnética, a ultrasonografia e a Tomografia por emissão de Positron (PET) [30]. Todos esses recursos permitem a visualização dos tecidos internos do corpo, sendo cada técnica indicada para a visualização de determinados tipos de tecidos.

Com a evolução dos aparelhos para a visualização interna do corpo humano, surgiram novas necessidades de um padrão para garantir a integração entre os vários modelos de aparelhos. O padrão definido e em uso na maioria desses equipamentos é o padrão *Digital Imaging and Communications in Medicine* (DICOM) [14, 33, 34].

2.4 – O Padrão DICOM

Normalmente é utilizado o padrão DICOM [14, 33, 34], para armazenar as imagens e as informações obtidas de exames médicos de vários dispositivos, como: raios X, tomografia computadorizada, ressonância magnética, ultrasonografia e outros.

A *American College of Radiologists* [35] e a *National Electrical Manufacturers Association* [14] buscaram a criação de um padrão que permitisse a conectividade entre os equipamentos médicos e os diversos softwares para a comunicação em rede e visualização de imagens médicas. A primeira versão do DICOM foi proposta em 1985 e após isso ele vem sendo atualizado. Entre 1992 e 1993 foi feita uma reformulação desse padrão, embora tenha sido mantida a compatibilidade com as versões anteriores. Atualmente o padrão DICOM está na versão 3.2 de 2003 [14].

O DICOM é um padrão que define além do mapa de bits da imagem (as fatias), as informações gerais do paciente e os dados da localização espacial das fatias dentro do contexto do exame. As informações espaciais das fatias são importantes para que o médico possa fazer medidas quando estiver analisando o resultado do exame e para programas de visualização em computador, para obter a visualização em 3D.

Programas de computadores que permitem a visualização dos exames e apresentam as medidas dos tecidos contidos nos exames também utilizam as informações de localização espacial contidas no arquivo armazenado no formato DICOM. Para a visualização em 3D, as informações dessa localização são importantes para que se possa reconstruir a estrutura original sem distorção das proporções das estruturas a serem apresentadas.

No arquivo armazenado no padrão DICOM, a imagem é armazenada no formato de um mapa de bytes que pode ou não ser compactado. Como um dos objetivos deste trabalho é interpolar as fatias de um exame com a finalidade de apresentar as imagens médicas em 3D, neste caso as principais informações do cabeçalho DICOM que são obtidas são as seguintes:

- **Acquisition Date:** data de aquisição das imagens armazenadas no seguinte for-

mato: ano(com 4 algarismos).mês(com 2 algarismos).dia(com 2 algarismos), (yyyy.mm.dd);

- **Acquisition Time:** horário de aquisição das imagens armazenadas no seguinte formato: horas.minutos.segundos.fração de segundos, (hh.mm.ss.frac);
- **Patient Name:** nome do paciente que está sendo examinado;
- **Patient Sex:** sexo do paciente;
- **Patient Age:** idade do paciente;
- **Slice Thickness:** distância em milímetros entre duas fatias. Esta informação é importante para o processo de reconstrução em 3D;
- **Slice Location:** posição da fatia em milímetros em relação a um referencial inicial;
- **Rows:** quantidade de linhas que a imagem possui;
- **Columns:** quantidade de colunas da imagem; e
- **Pixel Size:** distância em milímetros entre o centro de dois pixels.

As imagens no formato DICOM não são lidas pela maioria dos visualizadores gráficos utilizados atualmente, exceto pelo fato da adição de *plug-ins* especiais em alguns deles, mas normalmente eles não fazem parte do pacote original do produto. Mesmo os visualizadores capazes de mostrar as imagens DICOM, alguns não carregam as informações do paciente e das localizações espaciais das fatias contidas na imagem.

A manipulação de imagens no padrão DICOM em um software exige um tratamento detalhado desse padrão [14], aumentando a complexidade do software. O padrão DICOM está em constante evolução, dificultando também a manutenção dos softwares que tratam da visualização das imagens obtidas nos exames.

2.5 – Visualização de Imagens em 2D

A apresentação do resultado de um exame de tomografia computadorizada é normalmente feita a partir da impressão em filmes fotográficos, como as radiografias tradicionais,

ou em papéis fotográficos. A Figura 2.3 mostra essas opções de apresentação de resultados de exames. Essa prática torna mais fácil a manipulação dos resultados do exame, pois não exige do profissional médico nenhuma estrutura adicional para visualizar o resultado do exame. Por outro lado, o resultado impresso em filmes ou em papel, restringe as possíveis análises que o profissional poderia fazer.



Figura 2.3: (a) Resultados da tomografia impressos em filme; (b) resultados de exames impressos em papel fotográfico.

Como os aparelhos atuais de tomografia computadorizada armazenam as imagens no formato DICOM (em um CD, por exemplo), torna-se possível analisar os resultados de um exame sem a necessidade da impressão deles. Para tanto, é necessário a existência de um equipamento, um computador, e softwares adequados.

Com os recursos computacionais existentes nos PCs atuais (256 Mb de RAM, CPU com clock acima de 1.6 Ghz e placa aceleradora de vídeo) é possível utilizar um software visualizador, como mostrado na Figura 2.4. O software mostrado nessa figura é o *Medical Imaging Data Examiner* (AMIDE) [13]. Este software oferece diversos recursos adicionais como ampliações e reduções das imagens, bem como diversos tipos de segmentações que tornam mais visíveis algumas estruturas que normalmente não poderiam ser percebidas a partir de uma análise visual dos resultados impressos em filmes.

Existem vários visualizadores disponíveis no mercado como a linha de produtos da

General Electric Medical System (GE) [36] ou o eFilm da Merge eFilm [10]. Existem também produtos gratuitos e abertos como o Slice [11], o Blox [12] e o Amide [13]. A interface desse último é mostrada na Figura 2.4. Todos esses produtos permitem a visualização das fatias resultantes dos exames, oferecendo uma série de outros recursos já citados anteriormente neste capítulo.

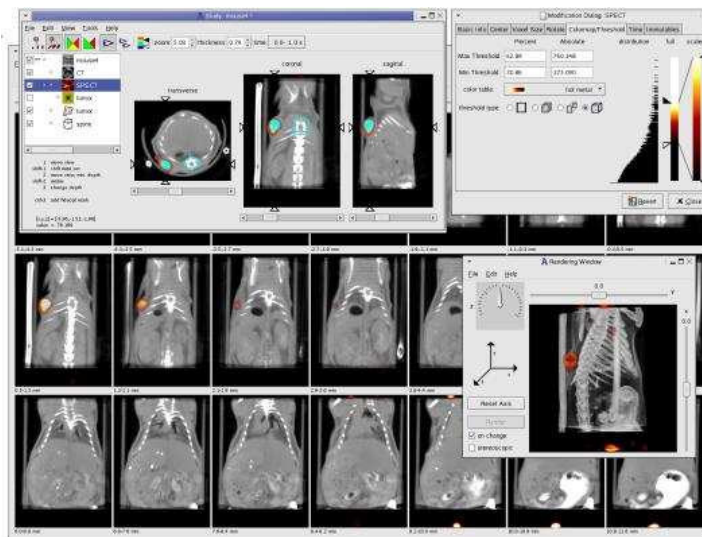


Figura 2.4: Software visualizador de imagens médicas, AMIDE.

2.6 – Visualização de Imagens em 3D

A visualização da imagem em 3D pode ser feita utilizando-se vários métodos existentes [37, 38, 39, 40]. O processo de visualização de uma imagem em 3D passa necessariamente pela geração de quadros, imagem em 2D, que representa uma determinada visão (perspectiva) da imagem em 3D. Navegar, isto é, visualizar várias regiões da imagem em 3D, exige a geração de uma sequência desses quadros para transmitir a sensação de movimento nessa imagem. O custo computacional de tais processos é sempre muito grande, exigindo estruturas sofisticadas de computação para gerar resultados satisfatórios.

Os processos de visualização em 3D podem ser realizados por duas abordagens principais, a visualização utilizando primitivas geométricas e a visualização pelo processamento direto no volume de dados [41]. A primeira abordagem de visualização utilizando primi-

tivas geométricas, apesar de possuir vários métodos, neste trabalho detalha-se apenas o algoritmo de Marching Cubes proposto por Lorensen e Cline [37]. A segunda abordagem de visualização consiste na geração de uma imagem em 2D contendo uma visão em perspectiva da imagem em 3D. Essa imagem é gerada a partir do acesso direto à imagem em 3D. Neste trabalho será detalhado o algoritmo de *Ray-Casting* [42]. Adicionalmente, são apresentadas também as funções de transferência, recurso utilizado para implementar a geração de imagens em 2D diretamente a partir da imagem em 3D, isolando estruturas específicas com a finalidade de permitir somente a visualização destas.

2.6.1 – Representação matemática de imagem em 2D e em 3D

Os algoritmos apresentados neste capítulo necessitam da utilização de uma notação matemática para a representação de imagens. Neste trabalho são utilizadas as seguintes notações para imagens em 2D e em 3D, respectivamente:

- **Imagem em 2D:** uma imagem em 2D é definida como uma função $I : \Omega \subset \mathbf{R}^2 \longrightarrow [a, b] \subset \mathbf{R}$, representada por $I(x, y) = k$, onde k é a informação do *pixel* (nível de cinza) da imagem no ponto (x, y) .
- **Imagem em 3D:** uma imagem em 3D é definida como uma função $I : \Omega \subset \mathbf{R}^3 \longrightarrow [a, b] \subset \mathbf{R}$, representada por $I(x, y, z) = k$, onde k é a informação do *voxel* (nível de cinza) da imagem em 3D no ponto (x, y, z) .

2.6.2 – Algoritmo Marching Cubes

O algoritmo *Marching Cubes* foi proposto por Lorensen e Cline [37]. Esse algoritmo descreve um processo que visa extrair de uma imagem em 3D, um conjunto de superfícies triangulares que definem o contorno de uma determinada estrutura constante nessa imagem. A renderização¹ desse conjunto de triângulo permite a visualização da imagem 3D da estrutura mapeada.

¹**Renderização:** produção de imagens virtuais em dispositivos computacionais gráficos

Um cubo de *pixels*² possui oito vértices e devido a isso podem ocorrer 256 combinações diferentes de triângulos formados por esses vértices. Entretanto, apenas as 16 combinações apresentadas na Figura 2.5 são suficientes, porque as demais combinações são similares a essas. Diferindo apenas por rotações dos vértices. Assim, basta o algoritmo implementado fazer todas as rotações possíveis do cubo de *pixels* e tentar reconhecê-los entre os 16 padrões mostrados nessa figura.

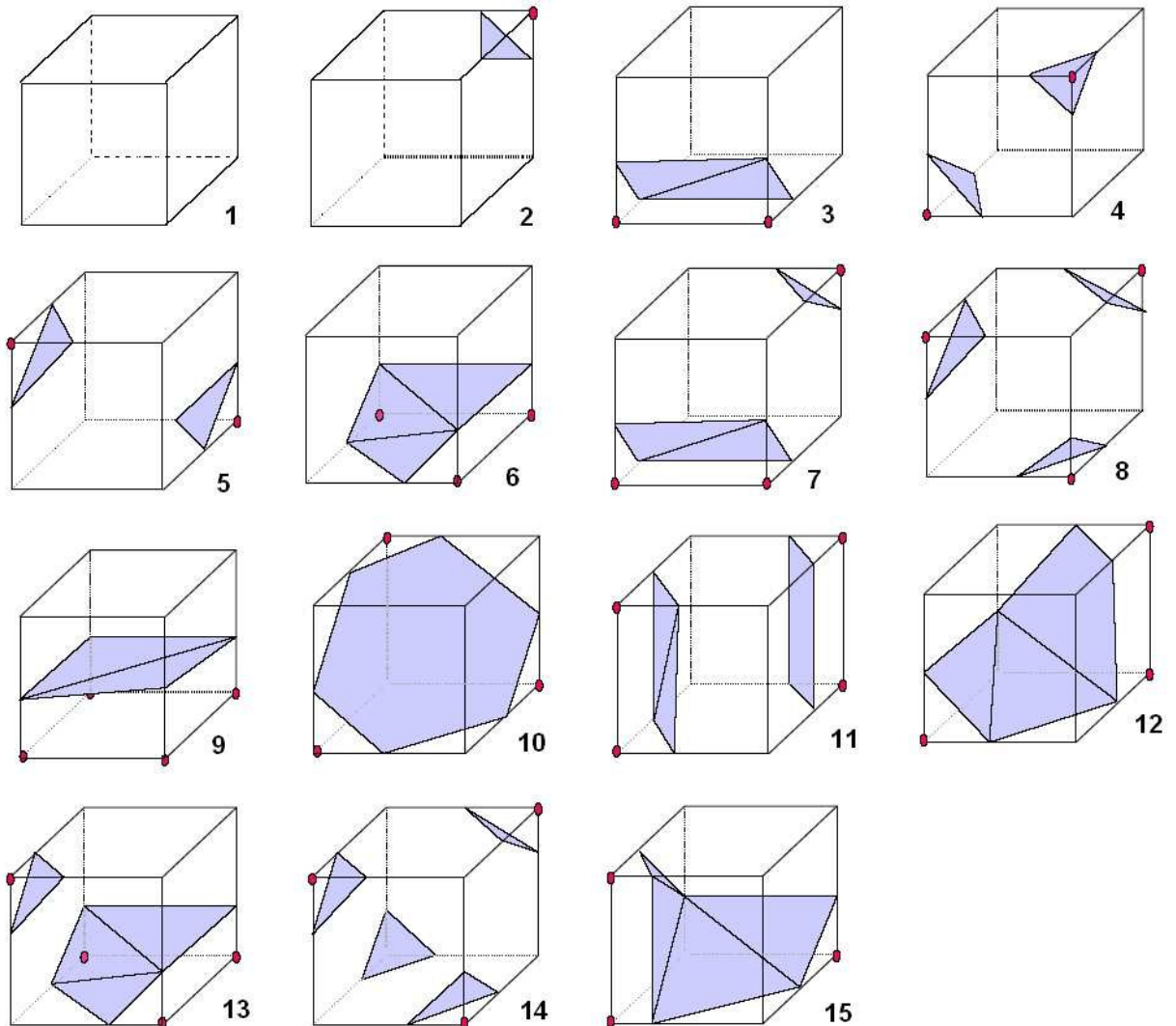


Figura 2.5: Combinações possíveis de um cubo de *pixels* utilizado pelo algoritmo Marching Cubes.

²**Cubo de *Pixel*:** neste trabalho, define-se como cubo de pixel, o conjunto formado por oito pixels postados na forma de um cubo. Esse conjunto é formado de dois subconjuntos de quatro pixels, onde cada subconjunto encontra-se em fatias distintas e consecutivas.

A obtenção dos triângulos que modelam a estrutura a ser visualizada, utilizando o algoritmo Marching Cubes, é realizada da seguinte forma:

- isolar na imagem em 3D apenas as intensidades de cores ou níveis de cinza que caracterizam a estrutura a ser aproximada pelos triângulos;
- para cada cubo de *pixels* da imagem em 3D, determina-se qual dos cubos mostrados na Figura 2.5 mais se assemelha com o cubo de *pixels* em questão; e
- armazenar as informações necessárias para a construção do triângulo no momento da visualização. Normalmente são armazenadas as coordenadas dos vértices e da cor, ou o nível de cinza do triângulo.

Os triângulos representam a superfície das estruturas da imagem em 3D que serão visualizadas, permitindo a navegação nas partes externas delas e resultando em uma visualização completa da estrutura. Devido ao fato que os triângulos delimitam apenas as superfícies da estrutura, se ocorrer a necessidade de navegar no interior da estrutura, com essa técnica não será possível. Outras abordagens podem ser utilizadas para tratar esse problema.

Os algoritmos descritos por Schroeder e outros [37] e por Wyvill e outros [43] não tratam de algumas inovações que foram adicionadas a eles ao longo do tempo. Uma delas é a redução do número de triângulos para representar as superfícies, desenvolvido por Durst e outros [44]. Essa redução aumenta a eficiência computacional do processo de renderização da imagem. Outros trabalhos como de Hoppe [45] também visou a redução do número de triângulos, mas nesse caso pela formação de malhas que aproximam às formas definidas pelo conjunto de triângulos.

A construção de uma malha triangular reduz a quantidade de informações que são tratadas no processo de apresentação e de navegação pela estrutura, tornando computacionalmente mais eficiente a abordagem quando comparada com a visualização direta no volume [37, 44]. Após a conclusão do processo de construção da malha triangular, uma aplicação visualizadora desenha o conjunto de triângulos que irá representar a estrutura

a ser analisada. Esse processo permite a redução do montante de informações a serem manipuladas, como por exemplo, em uma imagem em 3D de $256 \times 256 \times 256$ *pixels*, é reduzida de algumas unidades de milhões de *pixels* para algumas dezenas de milhares de triângulos.

2.6.3 – Visualização direta da imagem em 3D

Os processos de visualização direta em imagens em 3D podem ser divididos em quatro modos principais [41]: *X-Ray Rendering*, *Maximum Intensity Projection (MIP)*, *iso-surface rendering* e *full volume rendering*. Os quatro modos possuem duas características principais em comum:

- eles geram uma imagem em 2D disparando um raio que corta um grid (que será a própria imagem em 2D) em direção ao volume de dados (imagem em 3D); e
- eles obtêm os valores de cada elemento do grid por interpolação, utilizando também as funções de transferências, para definir a cor e o nível de transparência dos tecidos a serem visualizados.

O modo *X-Ray Rendering* determina o valor dos *pixels* da imagem em 2D apenas por um somatório gerando o resultado mostrado na Figura 2.6 (a), enquanto o modo *MIP*, considera apenas os elementos de maior valor que são computados gerando o resultado ilustrado na Figura 2.6 (b). Os outros dois modos podem ser considerados um único modo, pois o *iso-surface rendering* é um caso particular do *full volume rendering* [41].

Neste capítulo, este trabalho detalha a abordagem *full volume rendering*; após a apresentação do algoritmo de Ray-Casting proposto por Levoy [42] e das funções de transferência [41] que subsidiam o entendimento dessa abordagem.

2.6.3.1 – Algoritmo de Ray-Casting

O algoritmo de *Ray-Casting* [42] é usado para a visualização dos volumes quando se busca uma alta qualidade na imagem em 3D. O custo computacional desse algoritmo é

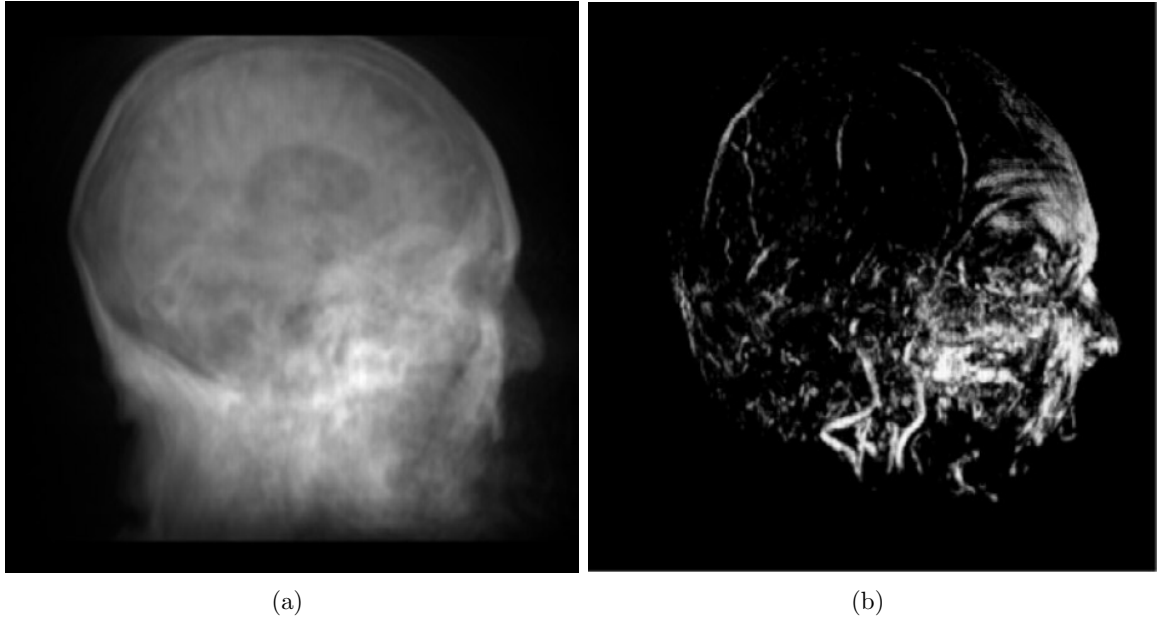


Figura 2.6: (a) Resultado do método *X-Ray*; (b) resultado do método MIP.

alto pois todos os *voxels* do volume são percorridos para determinar a cor e a opacidade de cada um deles em cada quadro da imagem resultante.

Para obter a cor e a opacidade de cada *pixel* de cada quadro da imagem a ser apresentada, um raio é disparado de cada ponto do plano de observação da imagem em direção ao volume de pixels, como mostrado na Figura 2.7.

O cálculo do valor da cor e da opacidade de cada *pixel* do plano de observação é obtido pela Equação (2.1).

$$I(u, v) = \sum_{s_i=a}^b f(x_{s_i}, y_{s_i}, z_{s_i}) \quad (2.1)$$

onde:

- $f(x_{s_i}, y_{s_i}, z_{s_i})$: intensidade de um *voxel* da imagem em 3D, pertencente à reta R_p .
- S_i : ponto da reta R_p pertencente à imagem em 3D.

A intensidade de cada *pixel* da imagem $I(u, v)$ é obtida pelo raio R_p direcionado contra

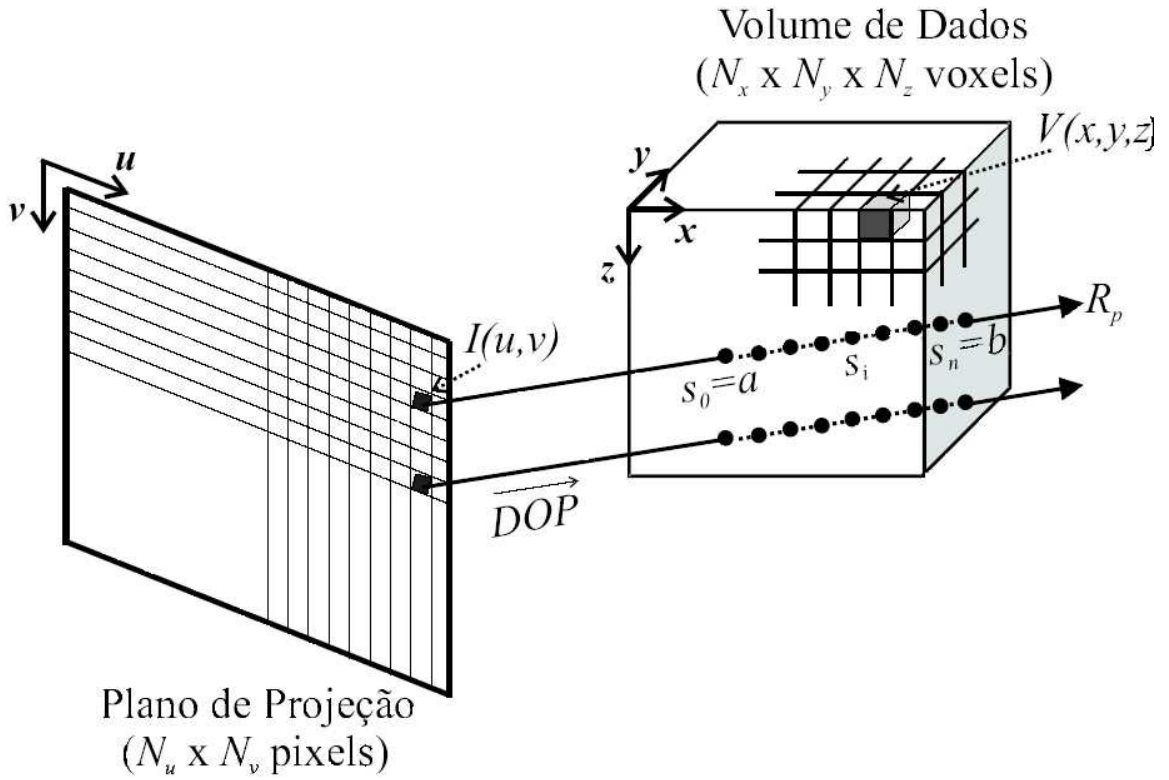


Figura 2.7: Esquema que ilustra os elementos do algoritmo de Ray-Casting.

o volume dos pixels na direção $\overrightarrow{DOP}$, como mostrado na Figura 2.7.

A obtenção de um quadro da imagem é feita pela determinação do valor da cor e da opacidade dependentes de u e v , isto é, da linha e da coluna da imagem obtida. As opacidades e as cores encontradas ao longo do raio são acumuladas até serem determinadas a cor e a opacidade final do pixel. Após ter obtido a opacidade máxima em um ponto (u, v) , isto é, a estrutura representada, no ponto (u, v) tornou-se totalmente opaca, Torna-se desnecessário prosseguir o processo de cálculo de $I(u, v)$, pois todas as estruturas seguintes da imagem não serão mais visualizadas, pois a visão delas estará obstruída pelas estruturas anteriores.

2.6.3.2 – Funções de transferência

A tarefa de visualização de volumes consiste em apresentar porções desse que contenha a região ou fenômeno de interesse a ser observado. Para que o processo de visualização seja correto é necessário que os objetos que não sejam de interesse sejam eliminados da

imagem a ser formada.

As funções de transferência alteram algumas características específicas da estrutura do volume orientando-se pelas cores e nível de transparência dos *voxels* da imagem em 3D. Essa propriedade permite que se faça a visualização de estruturas específicas da imagem em 3D, gerando cores e transparências específicas aos diversos elementos que compõem a estrutura.

Na sua forma geral, uma função de transferência possui tabelas internas de conversão que permitem a conversão de porções específicas do volume para uma determinada cor e para um determinado grau de transparência.

A escolha da função de transferência, por se tratar de uma tarefa difícil [41], é mais conveniente de ser realizada após conhecer a estrutura da imagem em 3D a ser visualizada. A determinação dos parâmetros, tabela de conversão, é também uma tarefa importante, pois ela determina a qualidade da visualização bem como do processo de visualização. A Figura 2.8 mostra o uso do histograma de uma imagem médica para escolher as cores e as transparências dos tecidos mapeados pelos tons de cinza constantes no histograma. Pode-se observar nessa figura que os diversos tipos de tecidos apresentam intensidades bem específicas, facilitando a composição da tabela de conversão da função de transferência.

O uso de funções de transferência em tarefas de visualização direta em volumes pode abranger diversas aplicações como interpolação [46, 47] e segmentação e análise de volumes onde se tenta localizar estruturas de formatos específicos como turbilhões ou linhas com formatos específicos [48, 49].

Alguns trabalhos utilizando funções de transferências em conjunto com processos de visualização direta em imagens em 3D são:

1. No Interactive Transfer Function Control for Monte Carlo Volume Rendering [50], foi apresentado uma proposta de uso de um processo interativo para a definição dos melhores parâmetros para a função de transferência. Esse processo interativo

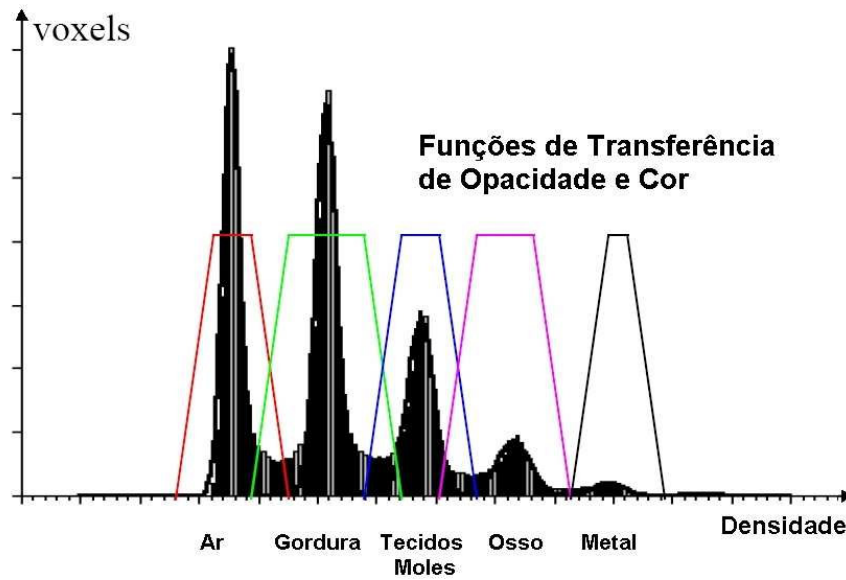


Figura 2.8: Histograma de uma imagem médica, onde cada tipo de tecido é caracterizado por um pico diferente.

permite que os parâmetros da função de transferência sejam modificados durante o processo de renderização, facilitando a determinação da função de transferência. A definição dos parâmetros das funções de transferência é realizada, na maioria das aplicações, por exaustivos experimentos até atingir a visualização desejada; e

2. Na Visualização de Estruturas Internas em Volumes de Dados Multimodais [38], as funções de transferências foram utilizadas para permitir a visualização e a integração de dados multimodais dos exames de um paciente. Além da definição de cores e transparência, as funções utilizadas, tornam-se ativas em intervalos específicos, sendo capazes de mapear cortes na estrutura visualizada. A Figura 2.9 [38] ilustra os resultados desse trabalho.

2.6.3.3 – Visualização direta em toda imagem em 3D

Para o detalhamento da abordagem de visualização direta em toda a imagem (*full volume rendering*), considere a Figura 2.10 que mostra o esquema de visualização de um ponto S de uma estrutura. Esse observador terá um visão em perspectiva da estrutura iluminada, isto é, uma imagem em 2D I . Um raio de luz de frequência λ atinge o ponto

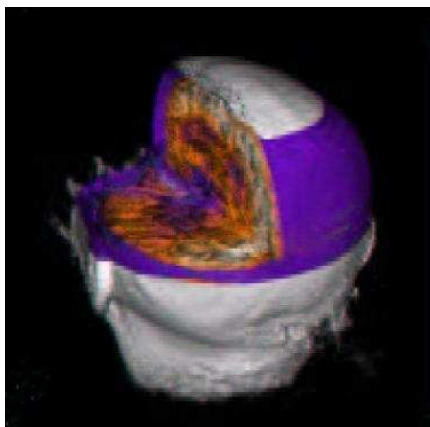


Figura 2.9: Resultado da aplicação de funções de transferência em dados multimodais, realçando um corte específico da estrutura visualizada.

S e é refletido em um ponto $I_\lambda(x_i)$ cujo raio refletido é representado pela reta $r(x_i)$ de comprimento L . Da mesma forma, existem diversos outros pontos s_i na estrutura que podem de alguma forma obstruir a visão do ponto S observado.

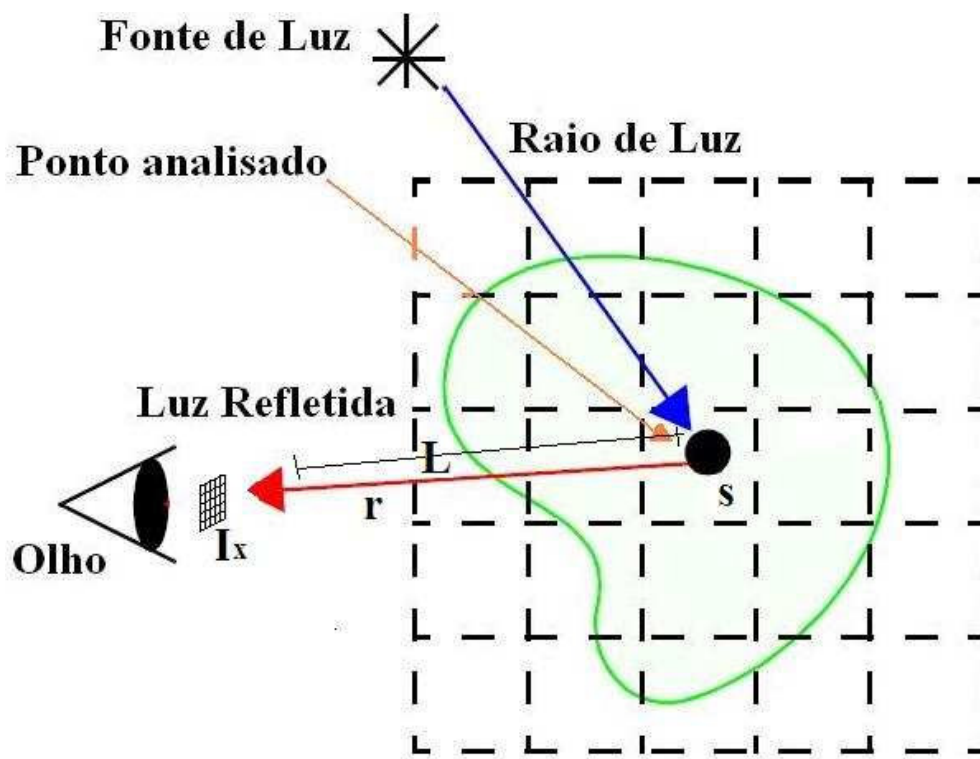


Figura 2.10: Esquema de elementos envolvidos no processo de visualização de um ponto em uma estrutura em 3D.

Assim, pode-se representar a imagem observada como $I_\lambda(X, r(X))$, onde X representa

o conjunto de pontos x_i da imagem em 2D vista pelo observador. O modo *full volume rendering* descrito por Johnson [41] descreve como elemento principal da formação da imagem, o resultado da Equação (2.2).

$$I_\lambda(x_i, r(x_i)) = \int_0^L C_\lambda(s) \mu(s) e^{-\int_0^s \mu(t) dt} ds \quad (2.2)$$

onde:

L - tamanho do raio $r(x_i)$, entre o ponto S da estrutura e o ponto x_i da imagem em 2D observada.

$\mu(s)$ - função que descreve a densidade dos elementos existentes na imagem em 3D.

s - representa os pontos da estrutura entre o observador e o ponto S observado.

$C_\lambda(s)$ - função que retorna a cor e a transparência no ponto s .

A função $C_\lambda(s)$ é composta na realidade por três componentes: emissão(E), reflexão(R) e transmissão de luz (T). Essa função é descrita pela Equação (2.3).

$$C_\lambda(s) = E_\lambda(s) + R_\lambda(s) + T_\lambda(s) \quad (2.3)$$

onde:

λ - comprimento de uma determinada onda de cor.

Dentro do contexto deste trabalho, as estruturas das imagens em 3D visualizadas, não são emissoras nem transmissoras de luz, particularizando a função $C_\lambda(s)$ apenas à reflexão de luz.

A Equação (2.2) modela a atenuação da luz incidida no ponto S , mas essa luz é também atenuada pela densidade das estruturas presentes no caminho percorrido pelo raio de luz antes de atingir o ponto S , levando à modelagem de sombras. Assim, o termo

$C_\lambda(s)$ da Equação (2.3), é dependente da posição do ponto s e é obtido pela Equação (2.4).

$$C'_\lambda(s) = C_\lambda(s)e^{\int_s^T \mu(t)dt} \quad (2.4)$$

onde:

C_l - cor da fonte de luz.

T - distância entre a fonte de luz e o ponto s .

A inserção da Equação (2.4) na Equação (2.2) produz a percepção de sombra. Isso ocorre porque estruturas opacas irão bloquear, isto é, atenuar totalmente $C_\lambda(s)$. As Figuras 2.11 (a) e (b) [41] mostram a diferença de uma imagem utilizando a Equação (2.2), sem sombra, e uma imagem utilizando a Equação (2.2), mas alterando $C_\lambda(s)$ por $C'_\lambda(s)$, com sombra.

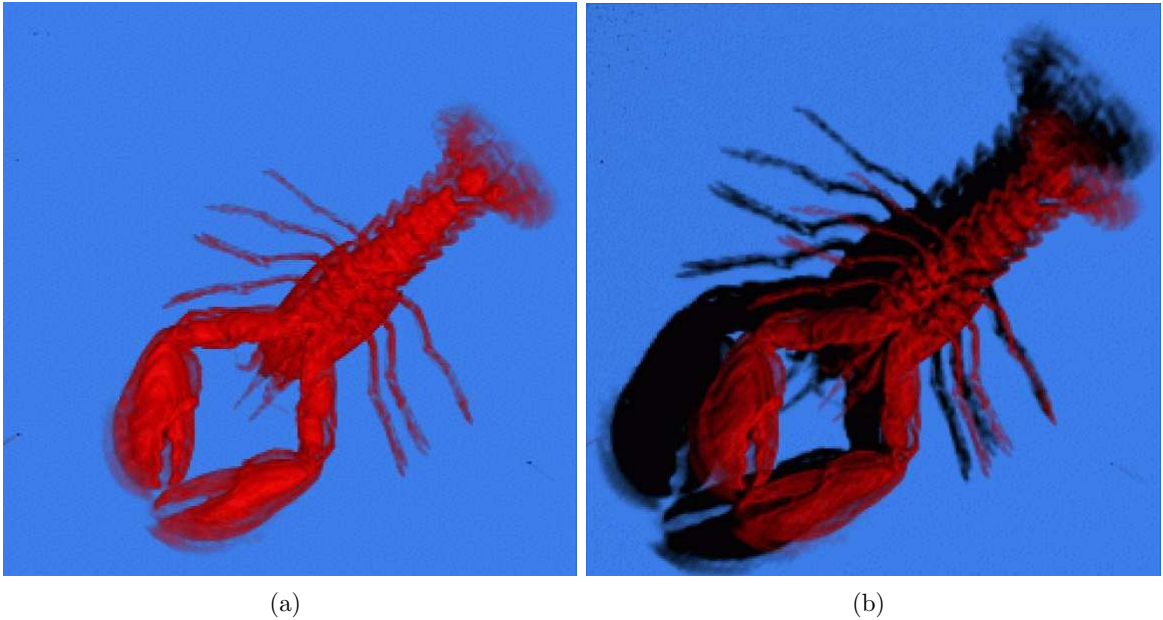


Figura 2.11: (a) Imagem de uma lagosta sem o efeito de sombra; (b) imagem de uma lagosta com o efeito de sombra.

Para o processamento computacional das funções descritas nas Equações (2.2) e (2.4),

é necessário que se definam essas funções em um domínio discreto, tornando possível a implementação de algoritmos computacionais para a determinação das imagens. Uma aproximação da Equação (2.2) apresentada por Johnson e outros [41] é mostrada na Equação (2.5).

$$I_{\lambda}(x, r) = \sum_{i=0}^{L/\Delta s-1} C_{\lambda}(i\Delta s) \mu(i\Delta s) \Delta s \prod_{j=0}^{i-1} e^{(-\mu(j\Delta s)\Delta s)} \quad (2.5)$$

Johnson e outros [41], citaram outras aproximações para se obter algoritmos computacionalmente mais eficientes para os processos de visualização, embora reduzindo a qualidade da imagem gerada.

2.6.4 – Bibliotecas para a visualização em 3D

A implementação de aplicações que permitem a visualização em 3D pode ser uma tarefa complexa, visto que existem no mercado diversos tipos de equipamentos e cada um desses equipamentos possui particularidades específicas. É indicado ao desenvolvedor de software a utilização de bibliotecas que tornem essa tarefa menos complexa, pois elas tratam das especificidades dos diversos dispositivos gráficos existentes e oferecem a esse desenvolvedor uma interface única. Desta forma, as aplicações visualizadoras podem se tornar mais portáteis aos diversos ambientes de *hardware* existentes hoje.

2.6.4.1 – O Open/GL

O OpenGL, desenvolvido pela Silicon Graphics, Inc. [51], é uma biblioteca de funções que permite a criação de gráficos em 2D e em 3D. A *Application Programming Interface* (API) do OpenGL pode ser utilizada para a construção de aplicações como CAD, jogos eletrônicos e aplicações visualizadoras de imagens reais ou virtuais. As aplicações escritas ou baseadas no OpenGL tem a vantagem de serem facilmente portáteis, visto que existem versões dessa biblioteca para os diversos ambientes disponíveis atualmente como Windows, Linux, Unix, Mac e outros.

O OpenGL permite o desenho de primitivas gráficas como linhas e polígonos, fornece suporte para o tratamento de iluminação, colorização, texturas e transparência, e permite a programação de animações, além de outros efeitos especiais. Quando uma aplicação chama a API OpenGL, ela vai formatando um buffer com todas as informações necessárias para a geração da imagem. Em seguida, é realizada a aplicação das transformações geométricas e de iluminação e, finalmente, é realizada a rasterização, isto é, a geração das imagens no dispositivo gráfico. Após a geração de um buffer com a imagem final, ela é colocada no *frame buffer*, que é a memória do dispositivo gráfico [52].

O uso do OpenGL exige do desenvolvedor de software o cumprimento de alguns padrões dentro da aplicação. Deve-se inicializar a interface e configurar diversos parâmetros com a finalidade de tornar o desenho dos gráficos possíveis. Os principais parâmetros a serem configurados na biblioteca são: posicionamento e escalas dos eixos cartesianos, posicionamento e intensidade das fontes de iluminação, definição do grau de transparência, da capacidade de reflexão e das cores das estruturas que serão desenhadas no ambiente.

Além das funções de configuração do ambiente e do desenho das estruturas, o OpenGL oferece uma série de recursos que facilitam o trabalho do desenvolvedor de software para a implementação de facilidades na navegação pelas estruturas apresentadas.

A Figura 2.12 (a) mostra um trecho de uma aplicação que desenha uma série de triângulos com o uso da API do OpenGL. Esse trecho permite, por exemplo, a apresentação dos resultados obtidos de um processo de triangularização utilizando o algoritmo Marching Cubes [37], como mostrado na Figura 2.12 (b).

2.6.4.2 – O visualization toolkit (VTK)

O VTK [53] é uma ferramenta, estruturada como uma hierarquia de classes, onde um conjunto de serviços é oferecido para a construção de aplicações gráficas em geral, fornecendo suporte a gráficos em 3D/2D, visualização e navegação, processamento de imagem e geração de volumes. O VTK é um software aberto produzido e disponibilizado gratuitamente pela Kitware Inc [53, 54]. Ele é portátil a várias plataformas diferentes de

```

for (i=0 ; i<pDoc->iqtTriangulos ; i++) {
    // Desenha o triangulo
    x1 = pDoc->fPonto[pDoc->iTriangulo[i][0]][0] * 5;
    y1 = pDoc->fPonto[pDoc->iTriangulo[i][0]][1] * 5;
    z1 = pDoc->fPonto[pDoc->iTriangulo[i][0]][2] * 5;

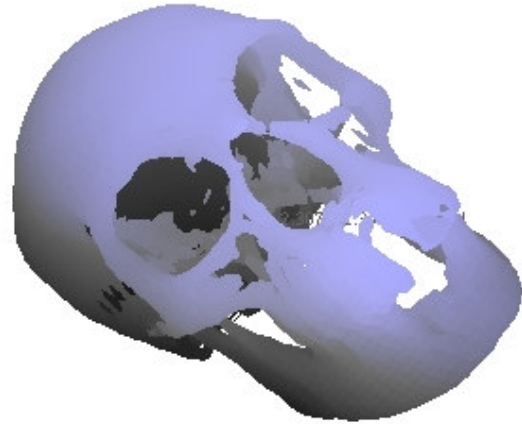
    x2 = pDoc->fPonto[pDoc->iTriangulo[i][1]][0] * 5;
    y2 = pDoc->fPonto[pDoc->iTriangulo[i][1]][1] * 5;
    z2 = pDoc->fPonto[pDoc->iTriangulo[i][1]][2] * 5;

    x3 = pDoc->fPonto[pDoc->iTriangulo[i][2]][0] * 5;
    y3 = pDoc->fPonto[pDoc->iTriangulo[i][2]][1] * 5;
    z3 = pDoc->fPonto[pDoc->iTriangulo[i][2]][2] * 5;

    // Calcula a Normal
    xn = x1+x2+x3;
    yn = y1+y2+y3;
    zn = z1+z2+z3;

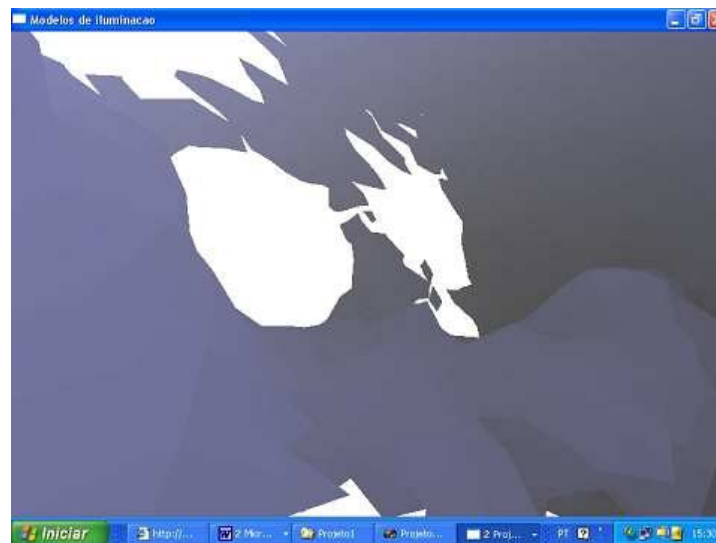
    glBegin(GL_TRIANGLES);
        glNormal3f(xn*10, yn*10, zn*10);
        glVertex3f(x1*10, y1*10, z1*10);
        glVertex3f(x2*10, y2*10, z2*10);
        glVertex3f(x3*10, y3*10, z3*10);
    glEnd();
}

```



(a)

(b)



(c)

Figura 2.12: (a) Trecho da programação OpenGL; (b) imagem resultante da execução do trecho do código da letra (a); (c) ampliação de parte da imagem da letra (b).

hardware e software.

A Figura 2.13 ilustra a estrutura básica de utilização do VTK, mostrando os principais elementos de uma aplicação escrita com a hierarquia de classes do VTK, a janela principal,

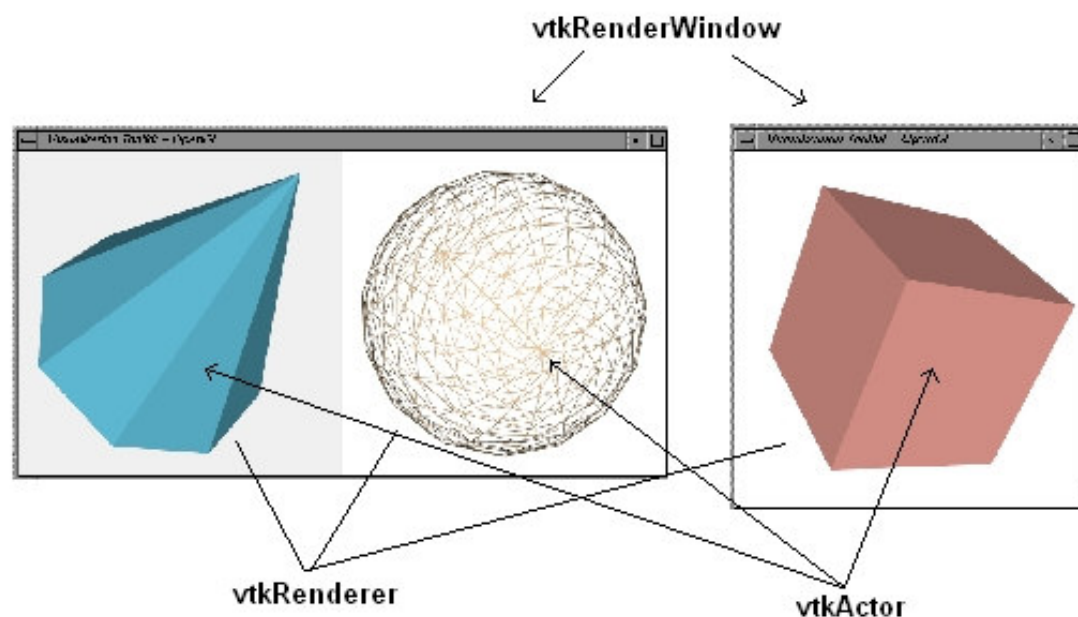


Figura 2.13: Principais elementos a serem tratados em uma aplicação escrita em VTK.

as áreas em que serão apresentadas as imagens e, finalmente, a própria imagem. O uso da interface, de maneira mais resumida, pode ser descrito como:

1. Criação de uma janela principal, onde são apresentadas as imagens. A classe `vtkRenderWindow` deve ser instanciada para tal finalidade;
2. Definição das áreas específicas da janela, onde são apresentadas as imagens. As classes `vtkRenderer` devem ser instanciadas e associadas com a classe `vtkRenderWindow` para efetivar a definição de tais áreas. Diversos parâmetros podem ser utilizados nesse momento com o objetivo de definir cores de fundo, textura, iluminação e outras propriedades, e;
3. Definição dos elementos gráficos que compõem a cena. As classes `vtkActor` devem ser instanciadas para cada um dos elementos gráficos. Diversos tipos de elementos gráficos podem ser associados à classe `vtkActor`. O VTK oferece vários desses elementos em sua biblioteca.

Além do processo básico do VTK descrito neste item, ele oferece classes específicas

para tratar as características importantes de uma aplicação gráfica. Algumas delas são:

- `vtkLigth` para estabelecer os critérios de iluminação da cena com a finalidade de permitir o tratamento adequado da reflexão, contribuindo para o realismo em 3D das cenas;
- `vtkCamera` para estabelecer as referências do observador com a finalidade de orientar a classe responsável pela navegação na cena; e
- `vtkRenderWindowInteractor` e `vtkInteractorStyle` para estabelecerem os critérios de navegação a serem realizados pelo usuário, isto é, o que será oferecido via mouse e teclado para o usuário navegar pela cena.

O VTK utiliza as primitivas do OPENGL [51] para acessar os recursos dos dispositivos gráficos, tornando-o portátil aos mesmos ambientes do OPENGL. A utilização do VTK na construção das aplicações visualizadoras propicia ao desenvolvedor do software maior facilidade de implementação e mais produtividade, visto que diversas tarefas já vem embutidas na hierarquia das classes.

Neste trabalho para a visualização de uma imagem em 3D, foram utilizadas duas classes do VTK: o `vtkMarchingCubes` e `vtkRayCasting` que implementam respectivamente os algoritmos de Marching Cubes [37] e de Ray-Casting [42], tornando mais simplificada a implementação da geração das imagens a partir da imagem em 3D e da navegação.

2.7 – O KIT de Visualização de Imagens de Tomografia Computadorizada (KVICT)

O KVICT [55, 56] é um KIT de desenvolvimento de aplicações visualizadoras de tomografia computadorizada que contém facilidades de operação da aplicação em sua interface usuário-máquina, a capacidade de ler exames no formato DICOM [14], apresentar os resultados dos exames em 2D e oferecer recursos para a interpolação e a visualização dos exames em 3D. Além de oferecer um conjunto de outras funções que permitem acompanhar passo a passo o resultado da pesquisa.

A premissa da modelagem desse KIT é a utilização da abordagem orientada a objetos [57] e a estruturação de software em camadas [58, 59], cuja finalidade é permitir a construção de aplicações organizadas em porções específicas e independentes que interagindo entre si, pelas suas interfaces ³, atinge os objetivos da aplicação. Essa abordagem permite a utilização dos pacotes já existentes com a finalidade de facilitar a sua estruturação e também torná-lo mais estável quanto à evolução dos padrões tratados por ele.

O conjunto das classes que compõem o KVITC [55, 56] realiza todas as tarefas de captura das imagens no formato DICOM, conversão dessas imagens para o formato bmp e apresentação delas em 2D. Ele também oferece funções para obter uma representação que permita a apresentação da estrutura em 3D.

A Figura 2.14 mostra o diagrama de classes do KVITC [55, 56]. Para construir uma aplicação, o desenvolvedor de software deve apenas instanciar as classes desse KIT e fazer uso de seus serviços.

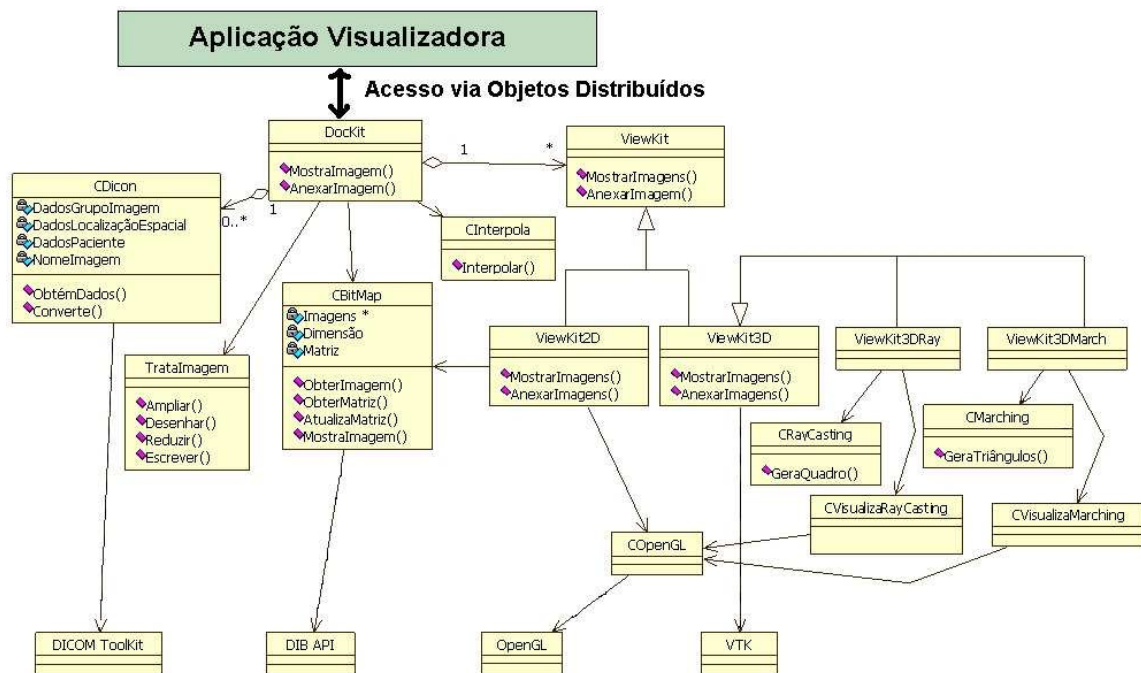


Figura 2.14: Hierarquia de classes do KIT desenvolvido neste trabalho.

³Interface: Nesse contexto, interface significa o conjunto de serviços públicos (funções) oferecidos por uma porção do software, que suas funcionalidades podem ser realizadas em outros softwares.

Todos as aplicações construídas para a realização dos experimentos deste trabalho utilizaram o KVICT [55, 56] como ambiente básico para o desenvolvimento de aplicações.

2.8 – Considerações Finais deste Capítulo

Este capítulo mostrou uma visão geral do que é a tomografia computadorizada (TC) e o seu histórico. Apresentou o padrão DICOM para armazenamento digital de imagens tomográficas e alguns visualizadores em 2D e em 3D existentes no mercado. E, finalmente, descreveu algumas técnicas para o processo de visualização em 3D e apresentou algumas bibliotecas para a construção de aplicações visualizadoras.

O próximo capítulo apresenta os principais métodos de interpolação de imagens, inicialmente para imagens em 2D em seguida para imagens em 3D. Com o objetivo de fornecer mais informações à proposta de interpolação de imagens em 3D, são mostrados alguns trabalhos que utilizaram métodos de interpolação em 3D em fatias de tomografia computadorizada. E, finalmente, são realizadas considerações finais sobre esse capítulo.

3 *Interpolação de Imagens em 3D*

3.1 — Introdução

Os métodos de interpolação vem sendo utilizados em várias áreas de conhecimento como a matemática, a estatística e também a área de processamento de imagem. Informalmente, pode-se dizer que interpolar é completar informações não existentes em uma região a partir de informações conhecidas dessa região. A operação que busca determinar valores além dos limites do conjunto conhecido é denominada de extrapolação.

No contexto de processamento de imagem, o objetivo dos processos de interpolação é completar regiões da imagem que estejam com informações indefinidas, defeituosas ou oclusas. Os processos de interpolação de um modo geral buscam transpor informações das porções conhecidas da imagem para a região que está sendo interpolada. A literatura atual apresenta diversos métodos de interpolação de imagens em 2D como Keys [60], Borges [61] e Meijering [62].

A interpolação aplicada a imagens em 3D, processo a ser utilizado neste trabalho, consiste da mesma forma que a interpolação da imagem em 2D [62, 63]. Na complementação de pontos que definem a estrutura da imagem a partir de pontos já conhecidos nela. Contudo, neste contexto os processos deverão promover a complementação de informações nas três dimensões, permitindo, por exemplo, a obtenção de uma imagem em 3D utilizando um conjunto de imagens em 2D, como as fatias das tomografias computadorizadas.

Este capítulo apresenta uma definição formal de interpolação, seguida da apresentação de alguns métodos de interpolação de imagens, inicialmente para imagens em 2D e em

seguida para imagens em 3D. São mostrados também alguns trabalhos que utilizaram métodos de interpolação em 3D com o objetivo de fornecer mais informações à proposta de interpolação de imagens em 3D a ser realizada nesta tese. E finalmente, são realizadas considerações finais sobre este capítulo.

3.2 – Interpolação

Considere um conjunto de pontos conhecidos e distintos $x_1, x_2, \dots, x_N \in R^q$ e $\lambda_1, \lambda_2, \dots, \lambda_N$ números reais quaisquer. Define-se como um problema de interpolação a determinação de uma função $F : R^q \rightarrow R$, tal que:

$$F(x_\mu) = \lambda_\mu, 1 \leq \mu \leq N \quad (3.1)$$

Uma forma de se obter F é por combinações lineares de translações de funções fixadas [64], ficando F da seguinte forma:

$$F(x) = \sum_{v=1}^N c_v h(x - x_v), x \in R^q \quad (3.2)$$

Impondo à Equação 3.2 as condições definidas na Equação (3.1), resulta-se em um sistema de N equações lineares. Se h é uma função estritamente positiva definida em R^q , a matriz gerada no sistema é positiva definida e, conseqüentemente, inversível [64]. Isso garante que o sistema terá solução independente do valor de N (número de pontos conhecidos) e dos escalares $\lambda_1, \lambda_2, \dots, \lambda_N$.

Uma classe especial de funções geradoras de funções interpoladoras, são as funções *radiais*, com a seguinte condição para uma função h ser dita *radial*:

$$h(x) = g(|x|) \quad (3.3)$$

onde:

$|\cdot|$ – alguma norma em R^q , como, por exemplo, a euclidiana; e

g – função contínua em R^q , definida em $[0, \infty)$.

Assim uma forma da função interpoladora é:

$$F(x) = \sum_{v=1}^N c_v g(|x - x_v|), x \in R^q \quad (3.4)$$

3.2.1 – Interpolação de Imagens em 2D

A interpolação para imagens em 2D pode ser utilizada para diversas finalidades, entre elas o retoque, *zoom*, rotações de imagens e até correção de distorções de imagem, conforme mostrado na Figura 3.1 [65].

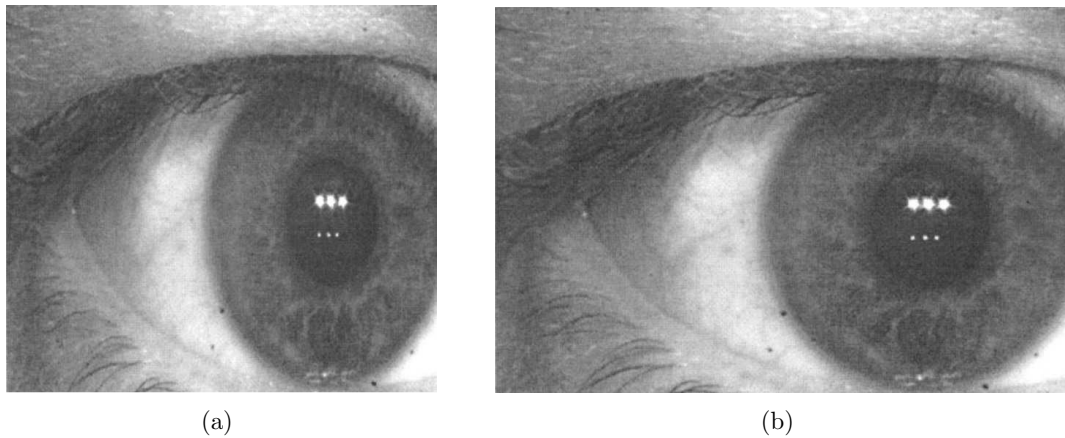


Figura 3.1: (a) Imagem distorcida; (b) imagem com distorção corrigida pela interpolação.

Para a determinação de métodos de interpolação de imagem digital deve-se inicialmente considerar que a imagem analógica é um sinal bidimensional $f(x, y)$ e sua transformada de Fourier é uma representação contínua e limitada na frequência. A imagem digital é um sinal discreto amostrado em intervalos de tempo τ , e sua transformada de Fourier

corresponde a reprodução da transformada de Fourier do sinal analógico em períodos de frequência $\frac{2\pi}{\tau}$.

O problema da interpolação corresponde a recuperar o sinal analógico a partir do sinal amostrado [65], e se a frequência de amostragem for suficientemente elevada de modo a evitar os efeitos de aliasing, é possível recuperar o sinal analógico convoluindo o sinal digital com um filtro retangular *rect*, representado pelo *kernel* $H(u, v) = \Im h(x, y)$, conforme mostrado na Equação (3.5).

$$H(u, v) = \begin{cases} 1, & |u| \leq \frac{\pi}{\tau}, |v| \leq \frac{\pi}{\tau} \\ 0, & \text{caso contrário} \end{cases} \quad (3.5)$$

Usualmente, são utilizados *kernels* simétricos e linearmente separáveis com o objetivo de reduzir a complexidade computacional [65], assim:

$$H(u, v) = H(u)H(v); H(w) \begin{cases} 1, & |w| \leq \frac{\pi}{\tau} \\ 0, & \text{caso contrário} \end{cases} \quad (3.6)$$

3.2.1.1 – A função interpoladora ideal

A função interpoladora do filtro *rect* é $h(x) = \text{sinc}(x)$. O problema dessa função é o fato de que ela não é espacialmente limitada, não sendo possível realizar-se a convolução de maneira completamente correta. Entretanto, pode-se utilizar aproximações da função $\text{sinc}(x)$ em intervalos limitados.

A função $\text{sinc}(x)$ é simétrica. Ela também é positiva entre 0 e 1, negativa entre 1 e 2, positiva entre 2 e 3, e assim por diante. Essas características fazem com que o processo de convolução dela com a imagem não provoque modificações na imagem, caso se utilize

o mesmo período para a realização da amostragem da imagem, além de não provocar o borramento da imagem e preservar as suas bordas.

Os métodos apresentados a seguir para a interpolação de imagens são apenas processos que se diferenciam pela sua capacidade de aproximar a função $\text{sinc}(x)$ com maior precisão. Para cada um dos métodos apresentados serão mostrados os resultados da imagem resultante e do espectro de Fourier da aproximação da função sinc obtido.

3.2.1.2 – Algumas técnicas de interpolação de imagens em 2D

1. Vizinho mais próximo:

O método do vizinho mais próximo consiste em determinar o valor do *pixel* interpolado pela obtenção do valor do *pixel* vizinho cuja distância do *pixel* interpolado é a menor distância absoluta entre todos os vizinhos analisados. O algoritmo mostrado abaixo detalha esse processo:

Algoritmo VizinhoMaisProximo;

Para $x_1 = 1 : M$

Para $y_1 = 1 : M$

Determinar o ponto (x_0, y_0) tal que $f(x_0, y_0)$ corresponda à tonalidade do ponto

$f(x_1, y_1) = f(x_0, y_0)$ (x, y) da imagem mais próximo de (x_1, y_1)

Fim;

Fim;

Fim.

O processo do vizinho mais próximo pode ser realizado pela convolução da imagem I com o *kernel* descrito na Equação (3.7).

$$h(w) = \begin{cases} 1, 0 \leq |w| < 0,5 \\ 0, \text{ caso contrário} \end{cases} \quad (3.7)$$