

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

Lidiane Pereira dos Reis

**RECONSTRUÇÃO 3D DE IMAGENS MÉDICAS
Uma Proposta para o Enfoque de Metodologias
Generalizadas para Reconstrução 3D de Estruturas
Anatômicas a partir de Tomografias Computadorizadas**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Prof. Dr. rer.nat. Aldo von Wangenheim

Florianópolis, agosto de 2001

Uma Proposta para o Enfoque de Metodologias Generalizadas para Reconstrução 3D de Estruturas Anatômicas a partir de Tomografias Computadorizadas

Lidiane Pereira dos Reis

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração de Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.



Prof. Aldo von Wangenheim, Dr. Orientador



Prof. Luiz Felipe Nobre, MD. Co-orientador



Prof. Fernando A. O. Gauthier - Coordenador do Curso

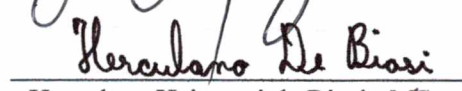
Banca Examinadora



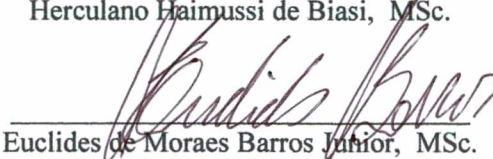
Marino Bianchin, Dr. – Presidente.



Prof. João Bosco Manguera Sobral, Dr.



Herculano Haimussi de Biasi, MSc.



Euclides de Moraes Barros Júnior, MSc.

AGRADECIMENTOS

A Deus.

Aos meus pais e familiares, pelo apoio e incentivo.

Ao meu grande amor e amigo Junior.

Aos colegas e amigos do Cyclops – UFSC.

ÍNDICE

LISTA DE SIGLAS E ABREVIATURAS	vi
LISTA DE FIGURAS	vii
RESUMO	ix
1 INTRODUÇÃO	5
1.1 <i>Definição do problema</i>	5
1.1.1 <i>Imagens Médicas em 3D</i>	6
1.1.2 <i>Justificativas</i>	6
1.2 <i>Objetivo Geral</i>	8
1.3 <i>Objetivos Específicos</i>	8
2 FUNDAMENTAÇÃO TEÓRICA.....	10
2.1 <i>Projeto Cyclops</i>	10
2.2 <i>Tomografia Computadorizada (TC)</i>	10
2.2.1 <i>Fundamentação Física</i>	11
2.2.2 <i>Densidade Radiológica</i>	14
2.2.3 <i>Manipulação de Imagens - Windows</i>	16
2.3 <i>DICOM (Digital Image Communications in Medicine)</i>	17
2.3.1 <i>Histórico</i>	18
2.3.2 <i>Implementação DICOM no Projeto Cyclops</i>	19
2.4 <i>Segmentação de Imagens</i>	20
2.4.1 <i>Thresholding</i>	21
2.4.2 <i>Detecção de Bordas</i>	23
2.4.3 <i>Detector de Bordas Canny</i>	24
2.5 <i>Morfologia Matemática</i>	25
2.5.1 <i>Tipos de Operações Morfológicas</i>	26
2.5.2 <i>Definições Fundamentais</i>	27
2.5.3 <i>Dilatação Binária (\oplus)</i>	27
2.5.4 <i>Erosão Binária (\ominus)</i>	28
2.5.5 <i>Dualidade entre Dilatação e Erosão</i>	28
2.5.6 <i>Abertura (Opening)</i>	29
2.5.7 <i>Fechamento (Closing)</i>	30
2.6 <i>Reconstrução 3D</i>	32
2.6.1 <i>Sistema de Coordenadas</i>	32
2.6.2 <i>Polígonos</i>	33

2.6.3	<i>Triangularização de Delaunay e Diagrama de Voronoi</i>	35
2.7	<i>Representação de Superfícies</i>	39
2.7.1	<i>VRML (Virtual Reality Modeling Language)</i>	39
2.7.2	<i>OpenGL (Open Graphics Library)</i>	41
3	ESTADO DA ARTE	44
3.1	<i>Cyclops StentPlanner</i>	44
3.1.1	<i>Discussão</i>	45
3.2	<i>CyclopsPopCorn</i>	46
3.2.1	<i>Discussão</i>	47
3.3	<i>VPAT – Visualização e interação com Pacientes Virtuais</i>	47
3.3.1	<i>Discussão</i>	48
3.4	<i>3D-DOCTOR</i>	48
3.4.1	<i>Discussão</i>	49
4	DESENVOLVIMENTO DO TRABALHO	50
4.1	<i>Obtenção e Visualização das imagens</i>	50
4.2	<i>Implementação</i>	52
5	METODOLOGIA DE RECONSTRUÇÃO 3D	57
6	TRABALHOS FUTUROS	69
7	CONCLUSÃO	71
8	REFERÊNCIAS BIBLIOGRÁFICAS	72
ANEXO 1	75
ANEXO 2	76

LISTA DE ABREVIATURAS E SIGLAS

2D	- Duas dimensões
3D	- Três dimensões
VRML	- Virtual Reality Modeling Language
TC	- Tomografia Computadorizada
HU	- Unidades de Hounsfield
DICOM	- Digital Image Communications in Medicine
ARC-NEMA	- Digital Imaging and Communications Standards Committee
CTN	- Central Test Node
HTML	- Hyper Text Markup Language
WWW	- World Wide Web
API	- Interface de Programação de Aplicação
GLU	- OpenGL Utility Library

LISTA DE FIGURAS

Figura 01 : <i>Reconstrução Tridimensional</i>	6
Figura 02 : <i>Software para workstations da GE (General Eletrics)</i>	7
Figura 03 : <i>Tomografia Computadorizada</i>	11
Figura 04 : <i>Fatia Transversal</i>	12
Figura 05 : <i>Esquema de Reconstrução</i>	13
Figura 06: <i>Imagem do Fígado</i>	17
Figura 07 : <i>Exemplo de segmentação de uma imagem em tons de cinza e colorida</i>	21
Figura 08 : <i>Histograma</i>	22
Figura 09 : <i>Função $T(r)$</i>	22
Figura 10 : <i>Detecção de Bordas por Canny</i>	24
Figura 11.1 : <i>Imagem Binária Original</i>	29
Figura 11.2 : <i>Imagem Binária aplicada Dilatação</i>	29
Figura 11.3 : <i>Imagem Binária aplicada Erosão</i>	29
Figura 12 : <i>Exemplo de uma Abertura</i>	30
Figura 13 : <i>Exemplo de uma Abertura em Imagem Médica</i>	30
Figura 14 : <i>Exemplo de um Fechamento</i>	31
Figura 15 : <i>Exemplo de um Fechamento em Imagem Médica</i>	31
Figura 16 : <i>Modelo de Coordenas Tridimensional</i>	33
Figura 17 : <i>Objetos formados por polígonos</i>	34
Figura 18 : <i>Diagrama de Voronoi com dois locais</i>	36
Figura 19 : <i>Diagrama de Voronoi com três locais</i>	36
Figura 20 : <i>Diagrama de Voronoi com 12 locais</i>	37
Figura 21 : <i>Diagrama de Voronoi (linhas cheias) e triangulação de Delaunay (linhas pontilhadas)</i>	38
Figura 22 : <i>Triangularização de Delaunay</i>	38
Figura 23 : <i>Reconstrução 3D utilizando VRML</i>	41
Figura 24 : <i>Ilustração da interface do Cyclops Stent Planner</i>	44
Figura 25 : <i>Reconstrução 3D realizada através do Cyclops Stent Planner</i>	45
Figura 26 : <i>Ilustração da interface do Cyclops PopCorn</i>	46
Figura 27 : <i>Reconstrução 3D realizada através do Cyclops PopCorn</i>	47
Figura 28 : <i>3D-DOCTOR</i>	48
Figura 29: <i>Aquisição de Imagens</i>	50
Figura 30: <i>Interface DicomEditor</i>	51

Figura 31: <i>Interface DicomSeriesEditor</i>	52
Figura 32: <i>Interface 3DEditor</i>	53
Figura 33: <i>Interface 3DEditorBrowser (imagens alteradas)</i>	54
Figura 34: <i>Interface 3DEditorBrowser (imagens alteradas por morfologia)</i>	55
Figura 35: <i>Duas imagens a serem reconstruídas em 3D</i>	58
Figura 36: <i>Representação das conexões após a primeira fase</i>	59
Figura 37: <i>Imagem das conexões após a primeira Fase</i>	59
Figura 38: <i>Representação das conexões após a segunda fase</i>	60
Figura 39: <i>Imagem das conexões após a segunda fase</i>	60
Figura 40 : <i>Reconstrução com Bifurcação (método antigo)</i>	61
Figura 41: <i>Ilustração do exemplo da Bifurcação</i>	62
Figura 42 : <i>Ilustração do Ponto de Sela</i>	63
Figura 43 : <i>Ilustração do Processo de Reconstrução da Bifurcação (1)</i>	64
Figura 44 : <i>Ilustração do Processo de Reconstrução da Bifurcação (2)</i>	65
Figura 45 : <i>Ilustração do Processo de Reconstrução da Bifurcação (3)</i>	67
Figura 46 : <i>Reconstrução Bifurcação</i>	68
Figura 47 : <i>Folhas Borda</i>	70
Figura 48 : <i>Pontos centrais das Folhas Borda</i>	70

RESUMO

Atualmente imagens médicas provenientes de aparelhos radiológicos podem trazer grandes avanços na medicina. A visualização de imagens médicas em diferentes planos bidimensionais (axial, coronal, sagital) e em três dimensões (3D) tem se tornado uma grande ferramenta para o auxílio ao diagnóstico médico, e o melhor esclarecimento das relações espaciais entre diferentes estruturas anatômicas de um determinado paciente, pois possibilita um auxílio aos profissionais na análise das imagens radiológicas, fornecendo meios mais precisos de medir a extensão e volume de uma determinada anatomia ou patologia em tais regiões, permitindo ao médico uma visão mais detalhada.

Neste contexto, o presente trabalho consiste no desenvolvimento uma metodologia que possa determinar, parametrizar e coordenar os diversos métodos adequados capazes de manipular e reconstruir imagens médicas, representando anatomias e patologias a partir de Tomografias Computadorizadas.

Desta forma, a disponibilização de uma ferramenta de software dentro deste cenário de radiologia é de fundamental importância para auxiliar diagnóstico médicos.

ABSTRACT

Currently, medical images from radiological equipment can bring a lot of advances to medicine. The visualization of medical images in different bidimensional planes (axial, coronal, sagittal) and three dimensions (3D) has become an important tool in aiding medical diagnosis and in the better clarification of the spatial relationships between different anatomical structures of a particular patient, in this way aiding the professionals in the analysis of radiological images, providing more precise ways to measure the extent and volume of a particular anatomy or pathology in such regions, allowing the medic a more detailed vision.

In this context, this work consist of the development of a methodology that can determine, parameterize and coordinate the various adequate methods capable of manipulating and reconstructing medical images, representing anatomies and pathologies from computerized tomographics.

Thus, the availability of a software tool in the field of radiology is of fundamental importance in aiding medical diagnosis.

1 INTRODUÇÃO

Neste capítulo será abordado os aspectos relevantes para o desenvolvimento de uma ferramenta de software capaz de disponibilizar reconstruções 3D de imagens médicas a partir de Tomografias Computadorizada, bem como o enfoque da necessidade e auxílio que pode ser empregado com a utilização de técnicas de reconstrução tridimensional no diagnóstico médico.

Atualmente a inovação e utilização de aparelhos radiológicos influenciaram de forma significativa a medicina trazendo inúmeros benefícios. A visualização de imagens médicas em diferentes planos bidimensionais (axial, coronal, sagital, oblíquos) e em três dimensões (3D) atualmente tem se tornado uma grande ferramenta para o auxílio ao diagnóstico médico, e o melhor esclarecimento das relações espaciais entre diferentes estruturas anatômicas de um determinado paciente.

O desenvolvimento de uma metodologia capaz de reunir todo um conjunto de técnicas de processamento de imagens, e disponibilizá-la de forma viável e interativa, tornando possível sua utilização independente de plataforma e local, não é uma tarefa simples. Além de toda a pesquisa envolvida, é necessário termos um conhecimento médico adequado, para que se possa ter uma ferramenta confiável.

É neste âmbito que se encontra o projeto *Cyclops*, um projeto que procura reunir profissionais da informática e profissionais da área médica em busca do mesmo objetivo.

1.1 Definição do problema

O problema consiste em se desenvolver uma metodologia que possa determinar, parametrizar e coordenar os diversos métodos adequados capazes de manipular e reconstruir imagens médicas, representando anatomias e patologias a partir de Tomografias Computadorizadas.

1.1.1 Imagens Médicas em 3D

Para que serve a visualização tridimensional na medicina diagnóstica e terapêutica?

Para um diagnóstico confiável, a análise da área onde estão localizadas suas principais estruturas anatômicas e possíveis patologias tem que ser interpretada. O processo de visualização em três dimensões possibilita um auxílio aos profissionais na análise das imagens radiológicas, fornecendo meios mais precisos de medir a extensão e volume de uma determinada anatomia ou patologia em tais regiões.

A visão em 3D de imagens médicas pode também trazer benefícios no aprendizado dos estudantes de medicina, já que o processo de análise de estruturas anatômicas a partir de imagens tridimensionais se torna mais dinâmico e interativo ao disponibilizar um meio em que seja possível a navegação destas imagens a partir de uma ferramenta de software.

A figura 01, mostra uma ilustração de uma imagem medica em três dimensões.

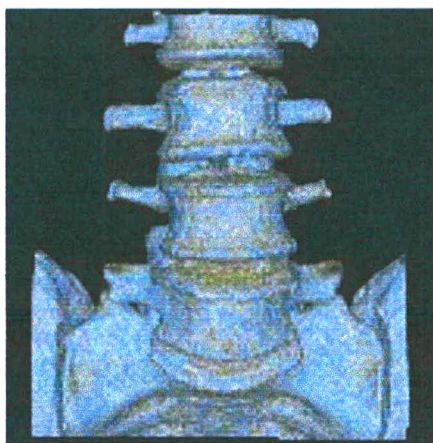


Figura 01 : Reconstrução Tridimensional

1.1.2 Justificativas

Atualmente na medicina podemos encontrar uma série de ferramentas de software gráfico que permitam a reconstrução e visualização de anatomias e patologias em três

dimensões. Porém, normalmente esses softwares muitas vezes não são muito acessíveis no que diz respeito ao custo de processamento e por seqüência o custo financeiro para aquisição dos mesmos.

O desenvolvimento da metodologia proposta consiste na construção de um software multiplataforma, onde seja possível a disponibilidade de módulos de processamento de imagem, reconstrução de imagens médicas em três dimensões e exportação dos resultados obtidos na linguagem VRML.

Neste contexto, a disponibilização multiplataforma permitiria o funcionamento em sistemas operacionais variados, e a exportação em VRML facilitaria o diagnóstico entre radiologistas localizados em ambientes diferentes via internet.

Quanto ao custo de processamento e financeiro, está no sentido de proporcionar a clínicas, hospitais e instituições de ensino que não possuem recursos financeiros para adquirirem equipamentos sofisticados, utilizarem o software.

Normalmente as workstations que acompanham os equipamentos radiológicos (Tomografias Computadorizadas, Ressonâncias Magnéticas, etc), possuem seus próprios softwares embutidos, que dispõe de várias funcionalidades, porém a aquisição destes equipamentos nem sempre são possíveis, pelo seu alto custo financeiro.

A figura 02 mostra uma interface de uma software para workstation da GE (General Eletrics).

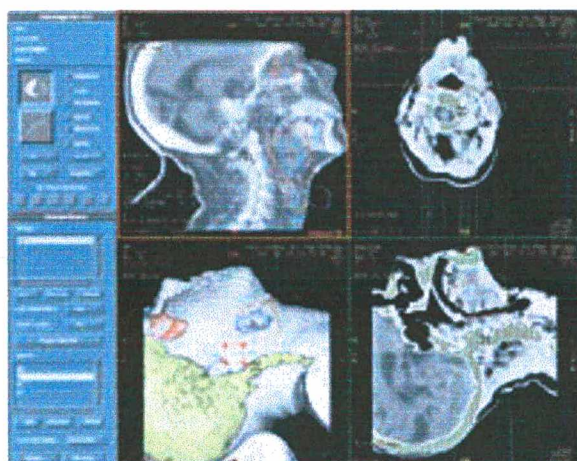


Figura 02 : Software para workstations da GE (General Eletrics)

Outro ponto importante da metodologia proposta está em disponibilizar o fácil acesso à ferramenta através de microcomputadores (PC), onde será possível realizar o diagnóstico através do software em qualquer ambiente. Neste caso, o médico poderá utilizar até mesmo seu computador pessoal.

1.2 Objetivo Geral

O presente projeto objetiva, em linhas gerais, o desenvolvimento de uma ferramenta de software gráfico que seja adaptável para imagens de Tomografia Computadorizada padrão DICOM, onde os médicos radiologistas poderão utilizar módulos de processamento de imagens, recursos de reconstrução tridimensional e exportação do resultado obtido na linguagem VRML.

1.3 Objetivos Específicos

Este trabalho tem como objetivos específicos:

obj. 1 - Desenvolvimento de um software multiplataforma capaz de disponibilizar a seleção de diferentes intervalos de densidades radiológicas (window), em imagens provenientes de exames de tomografia computadorizada no padrão DICOM.

obj. 2 – Disponibilização do recurso de processamento de imagens thresholding, resultando em segmentação da imagem original.

obj. 3 – Disponibilizar os recursos de morfologia matemática Opening e Closing.

obj. 4 – Desenvolvimento de um método para o usuário escolher com o mouse um ou mais objetos de uma imagem para ser reconstruída em 3D.

obj. 5 – Realizar a partir dos dados encontrados a reconstrução 3D da imagem.

obj. 6 – Exportar o resultado obtido da reconstrução 3D no formato VRML tornando possível ao usuário interagir os resultados obtidos com outros especialistas através da Internet;

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem com objetivo descrever os métodos e ferramentas pesquisadas e utilizadas no desenvolvimento da metodologia computacional proposta.

2.1 Projeto Cyclops

O projeto Cyclops, desenvolvido na Universidade Federal de Santa Catarina em cooperação com a Universidade de Kaiserslautern (Alemanha), é laboratório para utilização e desenvolvimento de tecnologia na área de informática médica, especialmente voltado à análise automatizada de imagens médicas. Inúmeros trabalhos têm sido desenvolvidos no âmbito do projeto Cyclops, empregando metodologias de Visão Computacional, Redes Neurais, etc.

2.2 Tomografia Computadorizada (TC)

A Tomografia Computadorizada foi inventada pelo engenheiro eletrônico Godfrey N. Hounsfield (1919), pela qual recebeu o prêmio Nobel [07] em fisiologia e medicina em 1979, juntamente com o sul-africano naturalizado americano, físico Allan McLeod Cormack (1924-). Cormack desenvolveu em 1956 a teoria e a matemática de como múltiplos raios projetados sobre o corpo, em ângulos diferentes, mas em um único plano, forneceriam uma imagem melhor do que o raio único, usado na radiografia.

É uma ferramenta importante no campo da radiologia médica e da indústria, pois ela permitiu visualizar estruturas que não eram identificadas pela radiologia convencional.

A grande vantagem que a Tomografia Computadorizada apresenta sobre as imagens de radiologias convencionais é que ela chega a possuir uma variação de 255 tons de cinza, enquanto que radiologia convencional possui uma variação de 20 a 30 tons de cinza,

apenas. Outra vantagem é poder obter cortes axiais transversos do corpo humano, proporcionando assim, informações que não são conseguidas através de radiologias convencionais, além de eliminar a superposição de estruturas adjacentes que ocorrem na radiologia convencional.

A possibilidade de obtenção de imagens de boa qualidade no plano axial em todo o corpo, e no plano coronal e sagital em segmentos específicos, apresenta como vantagem principal uma resolução espacial das estruturas anatômicas estudadas muito superior.



Figura 03 : Tomografia Computadorizada

2.2.1 Fundamentação Física

A TC é um modo de aquisição de imagens médicas que combina o uso de raios X com tecnologia de computação. Uma série de feixes de raios X é usada partindo de diferentes ângulos para montar imagens de uma seção transversal do paciente, a partir das diferentes densidades radiológicas apresentadas pelas estruturas anatômicas irradiadas. O objetivo é criar imagens onde se possa demonstrar órgãos, ossos e tecidos em grande detalhe, diferenciando espacialmente estas estruturas, tanto em duas como em três dimensões.

Uma imagem de TC é gerada através de um tubo de raios X que gira em torno do eixo do paciente (altura - direção que vai dos pés à cabeça do paciente). Com o auxílio de alta voltagem (120 a 150 KV), o tubo produz um feixe de raios X em leque, que se propaga

através do corpo do paciente em níveis controlados. Dependendo da densidade, da espessura e do número atômico dos tecidos do paciente, os raios serão mais ou menos atenuados. Localizados em posição diametralmente oposta ao tubo emissor de raios X estão os detetores. Os detetores, em geral de 500 a 1000, estão arranjados na forma de um semi-círculo e, dependendo da intensidade de raios X detectada, geram sinais elétricos, que são posteriormente processados por computador para gerar uma projeção transversal da imagem desejada.

A Figura 04 mostra uma representação da anatomia de uma fatia transversal do corpo obtida por várias medidas de atenuação dos raios-X.

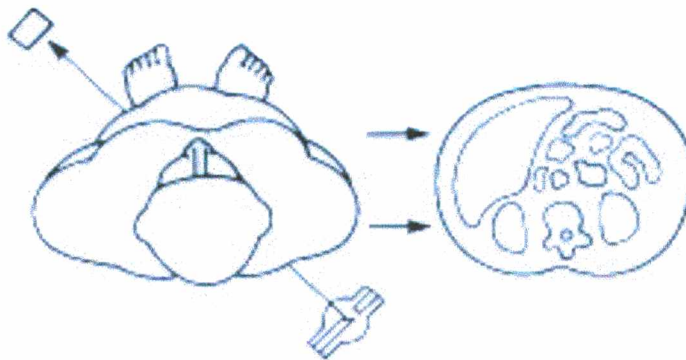


Figura 04 : Fatia Transversal.

Transladando o paciente de maneira incremental, transversalmente ao plano de aquisição, gera-se cada uma das fatias que compõem o exame. A translação é de alguns milímetros após cada fatia. Assim, uma série de imagens em 2D igualmente espaçadas, podem descrever estruturas anatômicas 3D com um nível de detalhe submilimétrico. O resultado de um exame por TC é apresentado como uma série de imagens representando fatias transversais do paciente. Cada fatia representa uma faixa do corpo do paciente com uma espessura entre 1 e 10 milímetros. A maioria dos *scanners* TC tradicionais geram imagens de 512x512 *pixels* para representar cada fatia, com o *pixel* representando porções de 0.5 a 2 milímetros do paciente. Assim, cada *pixel* representa as características de

absorção de um pequeno volume em torno de seu ponto central. Em geral são utilizados 2 bytes para representar a intensidade de cada *pixel*.

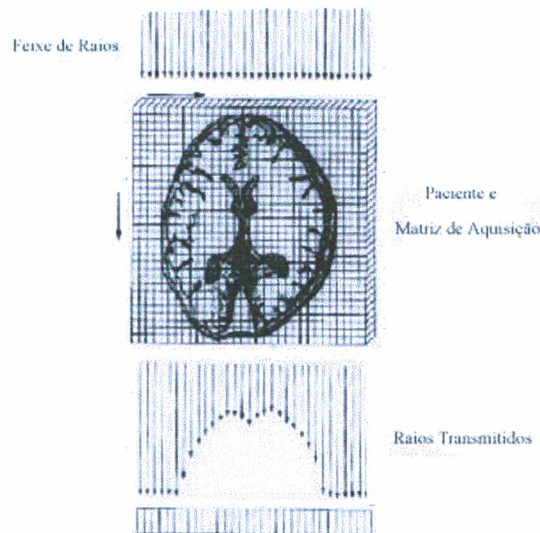


Figura 05 : Esquema de Reconstrução

A intensidade de absorção de cada ponto do corpo do paciente é medida em unidades de Hounsfield (HU). A Tabela 01 representa as HU para algumas características anatômicas importantes.

Ar	Gordura	Água	Rim	Pâncreas	Músculo	Fígado	Osso Esponjoso	Osso Compacto
-1000	-110 ±15	0	27 ± 15	35 ±10	40 ±10	55 ±10	200-400	>1000

Tabela 01: Absorção de tecidos no exame de CT em HU

Os *scanners* TC podem gerar uma imagem de uma fatia em 1 a 5 segundos, resultando em uma exposição à radiação semelhante às doses da radiografia padrão. Uma tomografia da cabeça leva 10 minutos em média para ser realizada, enquanto a do abdômen gasta em torno de 30 minutos, utilizando-se aparelhagem convencional, não-helicoidal.

Em resumo, a TC possui muitas características que a tornam mais vantajosa que a radiografia simples. Entretanto, também este exame possui seu conjunto de deficiências. As principais deficiências do exame de TC tradicional consistem em: Pequena resolução temporal para movimento cardíaco;

- Presença de artefatos inerentes ao método de aquisição;
- Resolução espacial relativamente pequena;
- Inabilidade de detecção de doenças em estágios incipientes que não tenham resultado ainda em alterações significantes dos coeficientes de densidade dos tecidos;

Outra modalidade de Tomografia Computadorizada foi desenvolvida para evitar algumas desvantagens da técnica original: a Tomografia Helicoidal ou Espiral. O termo TC Helicoidal é derivado da forma da trajetória percorrida pelo emissor de raios X durante o processo de aquisição. Nesta modalidade, o paciente avança a uma velocidade constante, enquanto o conjunto emissor-receptor gira continuamente, transversalmente ao eixo de translação do paciente, percorrendo uma trajetória em espiral. A espiral representa um conjunto de dados volumétricos contíguo, que descreve uma porção da anatomia do paciente sem vazios espaciais ou temporais. A tecnologia de *scanners* TC em espiral permite reduzir a dose de radiação à qual o paciente é exposto, além de resultar na aquisição de imagens com melhor qualidade. Em geral estes exames são 8 a 10 vezes mais rápidos que a TC convencional.

2.2.2 Densidade Radiológica

Um fator importante que afeta a capacidade de um aparelho de tomógrafo descrever precisamente a anatomia é o contraste, ou resolução de densidade, que é a capacidade de diferenciar os coeficientes de atenuação de porções adjacentes de tecido. Como a maioria dos tecidos moles possui densidades quase iguais, a consideração geralmente é a diferenciação de variações de alguns pontos percentuais ou menos.

Na computação, a representação de qualquer valor de pixel possui erro na forma de variação estatística, e é esta variação que limita a resolução de densidade final. Esta variação é denominada “ruído da imagem” e manifesta-se como um fundo granuloso.

O parâmetro usado para avaliar esta variação é o desvio padrão (σ), e o procedimento habitual para avaliação de um sistema é obter um *scan* de uma substância uniforme, como a água, e calcular o desvio padrão pela seguinte fórmula:

$$\text{Desvio-Padrão} = \sigma = \sqrt{\frac{\sum_{i=1}^N (CT_i - CT)^2}{n - 1}}$$

$$CT = 1/n (CT_1 + CT_2 + \dots + CT_n)$$

onde,

n = número de pixels usados na avaliação

CT = valor do pixel médio para n pixels

CT_i = valores de pixel individual

O significado do desvio-padrão é que novos *scans* do mesmo banho de água e um novo cálculo dos mesmos números TC pixel forneceriam valores em uma faixa igual ao valor previamente calculado. Portanto, o desvio-padrão descreve a incerteza em uma medida, e freqüentemente é expresso em termos de percentagem. Esta percentagem é obtida dividindo-se o desvio-padrão computadorizado em unidades de TC pela faixa da escala.

Por exemplo, o desvio-padrão igual a 5 HU em uma escala Hounsfield de ± 1.000 seria traduzido para $5 \text{ HU}/1.000 \text{ HU} = 0,005$, ou 0,5%. A interpretação é que dois pixels adjacentes não são significativamente diferentes se seus números TC variam em 5 HU ou menos.

2.2.3 Manipulação de Imagens - Windows

Vários aspectos de manipulação são incorporados em Tomografia Computadorizada, porém o mais importante, é a capacidade de variar o nível e o número de tons de cinza na imagem.

Normalmente, o olho humano não pode perceber diferenças de contraste menores que cerca de 10%, enquanto os tomógrafos podem facilmente demonstrar diferenças menores que 1%. Assim, a pequena diferença de resolução de densidade medida pelo tomógrafo deve ser exagerada para que o radiologista a veja. Isso é realizado com uma tela de visualização interativa, que permite ao radiologista selecionar uma pequena faixa de tons de cinza de toda a escala de número de TC e reajustar os limites de preto e branco.

Por exemplo, os números TC de tecidos hepáticos situam-se aproximadamente na faixa de 40 a 90 HU em uma escala Hounsfield de ± 1.000 . Um número TC igual a 40 HU ou menor é estabelecido como preto, e um número TC igual a 90 HU ou maior é estabelecida como branco, então a escala de visualização é de 50 HU, e toda a faixa de tecidos hepáticos cobre uma faixa de contraste de 100%. Agora, uma alteração de densidade de 10 HU na escala original, que representa uma alteração de 1% real, é convertida em uma alteração de contraste de 20% (10/50) na faixa ajustada. Isso aumenta significativamente o contraste visual entre as diferentes estruturas estudadas.

A faixa de números de TC selecionados para amplificação da escala de cinza é denominada *Largura da Janela* e a posição nesta escala, onde a imagem é centralizada, é denominada *Nível da Janela*.

A figura 06 mostra uma imagem de fígado com diferentes ajustes da janela.

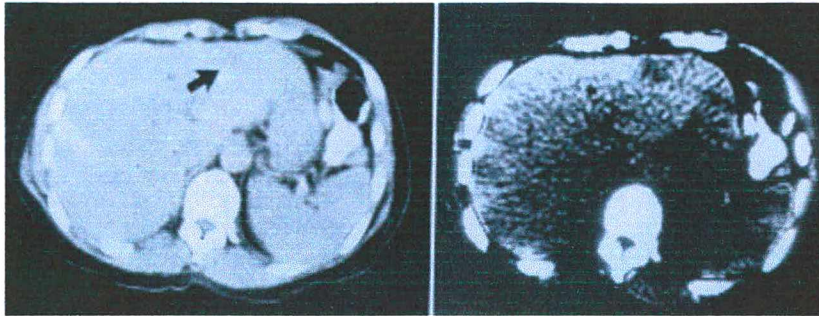


Figura 06: Imagem do Fígado

A importância da exibição interativa e da escolha de ajustes é demonstrada neste exemplo, observando-se que em janelas largas as metástases hepáticas não são visualizadas.

2.3 DICOM (Digital Image Communications in Medicine)

O padrão DICOM é um ponto importante na área de informática médica, pois além de permitir que aparelhos que trabalham com informações médicas digitais comuniquem-se entre si, o padrão DICOM prevê o armazenamento, junto com uma imagem, dos valores de todos os parâmetros de aquisição associados àquela modalidade de imagem e que são necessários à sua correta interpretação.

No âmbito do Projeto Cyclops, a maioria dos trabalhos desenvolvidos relacionam-se com a aquisição e/ou armazenamento de imagens, principalmente no que diz respeito à análise automatizada de imagens médicas digitais, que por sua vez necessita da disponibilidade de imagens a serem analisadas. Atualmente os aplicativos do Projeto Cyclops utilizam um software chamado de *DicomEditor* para ler imagens médicas digitais e informações correlatas que estejam armazenadas em determinados servidores.

Abaixo será descrito um breve histórico do DICOM e em seguida será especificado o trabalho de implementação DICOM existente no projeto *Cyclops*.

2.3.1 Histórico

Em meados da década de 80, a fim de desenvolver um padrão que permitisse aos diversos equipamentos de imagens médicas digitais (Tomógrafo Computadorizado, Raio-X, etc) comunicarem entre si, o Colégio Americano de Radiologia (*ACR*) e a Associação Nacional dos Fabricantes de Equipamentos Elétricos (*NEMA*) formaram um comitê . O objetivo deste grupo, nomeado como *ARC-NEMA (Digital Imaging and Communications Standards Committee)*, era definir e desenvolver um padrão internacional para a comunicação de dados entre equipamentos radiológicos de diversos fabricantes e softwares de informatização hospitalar. Mais tarde este padrão foi denominado DICOM – Digital Image Communications in Medicine.

O DICOM define basicamente três tópicos:

Um protocolo de comunicação entre equipamentos radiológicos e softwares hospitalares e de imagem, tanto em modo ponto-a-ponto como em rede Ethernet sob TCP/IP;

Um formato de dados diferente para cada modalidade de imagem médica existente, como tomografia computadorizada, raio-X ressonância magnética, etc e todos os parâmetros acompanhantes dessa imagem, como parâmetros de ajuste do aparelho durante a aquisição daquela imagem e posição do paciente;

E, um protocolo de serviços, como por exemplo “transmitir dados de exame” ou “procurar exames de paciente X”, que normatiza a comunicação entre equipamentos e softwares DICOM-compatíveis.

O DICOM encontra-se hoje na versão 3.0 e é um padrão aberto, podendo ser estendido para englobar novos serviços ou tipos de imagens ou dados adicionais em um tipo já existente, que é posteriormente homologada.

Atualmente existe uma tendência no sentido de codificar todos o tipos de sinais biológicos, como eletroencefalograma e eletrocardiograma também em DICOM. A utilização de DICOM não está sujeita a nenhum tipo de restrição de direitos ou royalties. A documentação do padrão pode ser encontrada no site do ACR, da NEMA ou RSNA ou

adquirida em CD. Para um equipamento ou um produto de software ser considerado DICOM-compatível é necessária a certificação-DICOM, realizada anualmente. Parte dessa certificação é realizada através de testes de comunicação e compatibilidade com um software de domínio público denominado CTN – Central Test Node, disponível na Internet. Inicialmente o DICOM teve pouca aceitação por parte dos fabricantes pois a intercomunicabilidade entre equipamentos padronizada permitia a hospitais escolher com muito mais liberdade o fornecedor de seus equipamentos. Atualmente o padrão se estabeleceu e todos equipamentos dos principais fabricantes são DICOM-compatíveis.

2.3.2 Implementação DICOM no Projeto Cyclops

Atualmente, no Projeto Cyclops, utiliza-se um cliente DICOM chamado de *DicomEditor*, capaz de carregar imagens médicas digitais no formato DICOM de servidores, como, por exemplo, o CTN. Este cliente DICOM (SCU) suporta os IODs para as modalidades de Ressonância Magnética (MR), Tomografia Computadorizada (CT), Ultra-Som (US), Medicina Nuclear (NM), Raios X (X-RAY) e Secondary Capture (SC). Portanto, a instalação destes aplicativos requer a existência de um servidor de imagens médicas digitais já em funcionamento (ou a instalação de um novo), no respectivo ambiente de utilização.

Para tanto, foi implementado o *CyclopsDicomEditor*¹[16], onde é possível o envio de imagens médicas digitais e outras informações diretamente para a aplicação que as processará, sem a necessidade de armazenamento em outros dispositivos.

Dessa forma, os sistemas para a conversão automática de séries inteiras de imagens de pacientes, no formato DICOM 3, para outros formatos digitais de arquivos de imagem mais compactos e facilmente transmissíveis através de linhas de dados de baixa velocidade, conectadas à Internet podem ser implementados com bastante facilidade, utilizando um software capaz de receber imagens diretamente de um aparelho de Ultra Som ou

¹ O trabalho "*Cyclops DICOM Editor*" está sendo desenvolvido no âmbito do projeto *Cyclops* pelo aluno Paulo Roberto Dellani, mestre do Curso de Ciência da Computação da Universidade Federal de Santa

Tomografia Computadorizada, por exemplo, sem a dependência de soluções de terceiros, que muitas vezes tem limitações ou características de desenvolvimento que dificultam muito a sua expansão.

2.4 Segmentação de Imagens

O objetivo principal da segmentação é dividir uma imagem em partes, obtendo-se como resultante uma forte correlação entre o produto (imagem) obtida, com objetos ou áreas do mundo real contidos na imagem. Existem diversas técnicas de segmentação com pequenas variações entre elas, mas podem ser separadas em dois grupos.

O primeiro grupo é chamado de técnica de *Espaço de Medida*. A imagem é considerada um espaço Euclidiano. Executa-se uma transformação linear para outro espaço vetorial, processa-se a imagem e executa-se a transformação inversa. A segmentação é feita no "Domínio do Espaço" da imagem.

O segundo grupo executa a segmentação na própria imagem, sem utilizar a transformação linear. Esta técnica é mais conhecida como *Domínio Espacial*.

Os algoritmos de segmentação de imagens monocromáticas geralmente são baseadas em duas propriedades básicas dos valores de nível de cinza : Descontinuidade e Similaridade.

Segmentação por Similaridade : é um processo baseado na classificação de "pixels", por limiarização (thresholding), crescimento de regiões (region growing) e união e separação de regiões (region merging and splitting). Assim as técnicas confiam em padrões comuns valores de intensidade dentro de um agrupamento de "pixels" vizinhos. O agrupamento de "pixels" é chamado de região, e a meta do algoritmo de segmentação em regiões é agrupar regiões de acordo com seus papéis funcionais.

Segmentação por Descontinuidade : é um processo pelo qual a partição da imagem é baseada na mudança abrupta do nível de cinza. A principal área de interesse dentro desta

categoria é detecção de pontos isolados, detecção de linhas e contornos (edges) na imagem. Assim estas técnicas confiam em descontinuidades dos valores da imagem nas regiões distintas, e a meta do algoritmo de segmentação por bordas é demarcar o limite que separa estas regiões com precisão.

O conceito de segmentação de uma imagem baseado em descontinuidade ou em similaridade dos valores de nível de cinza de seus pixels pode ser aplicado tanto em imagens estáticas como em imagens dinâmicas (que variam com o tempo).

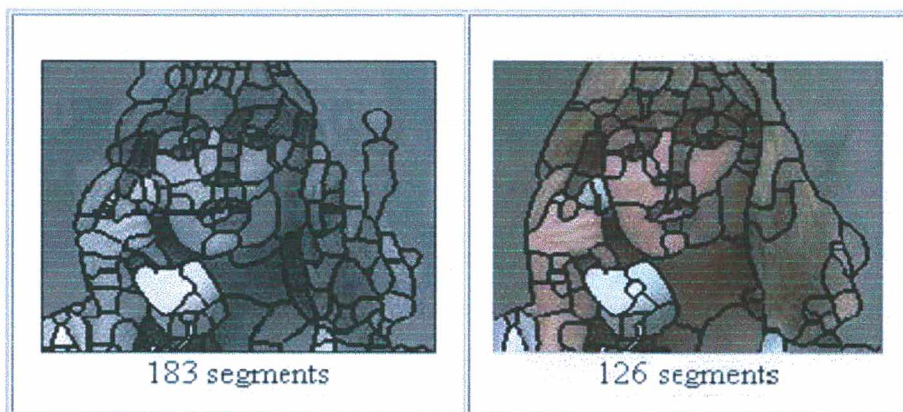


Figura 07 : Exemplo de segmentação de uma imagem em tons de cinza e colorida

2.4.1 Thresholding

A técnica de limiarização (Thresholding) é um tipo de segmentação que consiste em definir um valor limiar para a cor (densidade radiológica) de um pixel. Pixels com valores de densidade acima do limiar se tornam pretos e de valor abaixo se tornam brancos.

A conversão dos níveis de cinza de uma imagem com representação binária (dois tons) é importante para uma série de objetivos, tais como:

- Identificar objetos e separá-los do fundo da imagem;
- Analisar a forma da imagem, sendo mais importante que a intensidade dos pixels;

- Apresentar a imagem em um dispositivo de saída que tem somente um bit de resolução de intensidade, ou seja, um dispositivo de dois níveis.

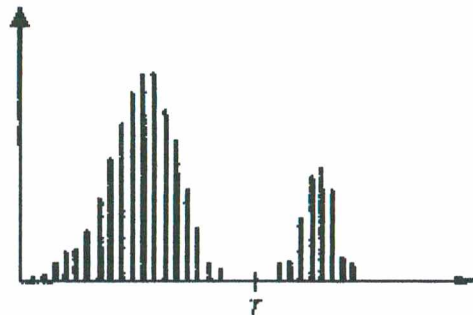


Figura 08 : Histograma

A figura 08 representa a um histograma típico de uma imagem de cor mais clara sob uma funda mais escuro. Uma transformação da intensidade da imagem para que ela passe a ter somente dois níveis distintos, pode separar o objeto do fundo. Esta transformação é chamada binarização, e pode ser descrita através da aplicação da função: $s = T(r)$.

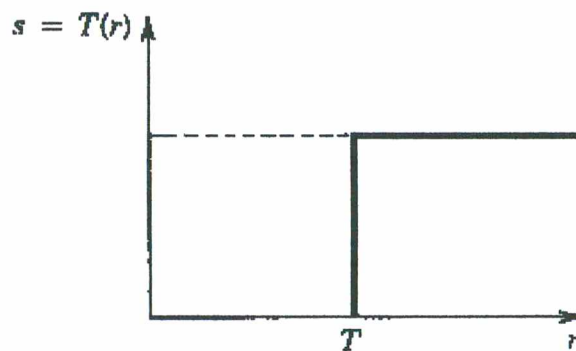


Figura 09 : Função $T(r)$

A função $T(r)$ compara o sinal de entrada com um valor de *threshold* (T), escolhido como referência para a separação dos níveis de cinza. O sinal de saída, apresentado é obtido pela relação:

$$s = \begin{cases} 1 & \text{para } r > T \\ 0 & \text{para } r < T \end{cases}$$

O histograma da imagem, após sua binarização, terá apenas dois tons com número de pixels diferentes de zero.

2.4.2 Detecção de Bordas

A Detecção de bordas é uma das técnicas básicas utilizadas pela visão humana no reconhecimento de objetos. É o processo de localização e realce dos pixels de borda, aumentando o contraste entre a borda e o fundo. Este processo verifica a variação dos valores de luminosidade de uma imagem.

Uma Borda é o *contorno* entre um objeto e o fundo indicando o limite entre objetos sobrepostos. São definidas como picos da magnitude do gradiente, ou seja, são variações abruptas que ocorrem ao longo de curvas baseadas nos valores do gradiente da imagem. As bordas são regiões da imagem onde ocorre uma mudança de intensidade em um certo intervalo do espaço, em uma certa direção. Isto corresponde a regiões de alta derivada espacial, que contém alta frequência espacial.

A detecção de bordas é parte do processo de segmentação e, tecnicamente, pode ser descrita como sendo o processo de localização dos *pixels* da borda, assim, com o realce da borda, o contraste entre ela e o segundo plano será incrementado, tornando-a mais visível. Na prática, tem-se que este processo procura determinar se uma borda passa sobre ou perto de um determinado *pixel*. Isto é feito examinando a taxa de troca de intensidade próxima ao *pixel* – alterações agudas são boas evidências de uma borda, alterações suaves irão sugerir o contrário.

Há um certo número de algoritmos de detecção de bordas, cada qual pode produzir um resultado de maior ou menor qualidade, dependendo das características da imagem na qual foi aplicado.

2.4.3 Detector de Bordas Canny

O detector de Canny, elaborado por Jonh Canny em 1986, é amplamente usado nos sistemas de visão computacional modernos.

O detector de bordas de Canny é um filtro de convolução f que uniformizaria o ruído e localizaria as bordas. O problema é identificar um filtro que otimize os três critérios do detector de bordas. Se considerarmos uma borda de uma dimensão variando no contraste e então convolucionando a borda com a função de uniformização de Gauss, o resultado será uma variação contínua do valor inicial ao final, com uma inclinação máxima no ponto onde existe um "degrau". Se esta continuidade é diferenciada em relação a x , esta inclinação máxima será o máximo da nova função em relação a original.

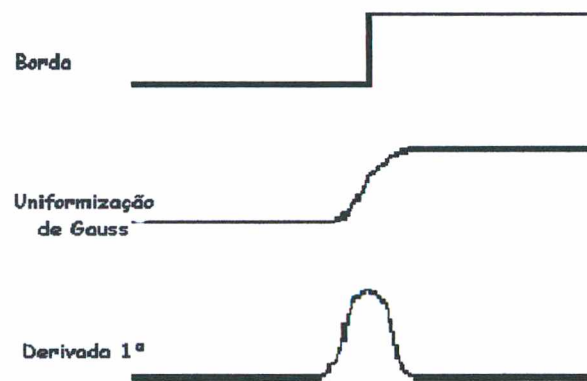


Figura 10 – Detecção de Bordas por Canny

Os máximos da convolução da máscara e da imagem indicarão bordas na imagem. Este processo pode ser realizado através do uso de uma função de Gauss de 2-Dimensões

na direção de x e y . Os valores das máscaras de Gauss dependem da escolha do sigma na equação de Gauss:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$G'(x) = \frac{x}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma^2}}$$

Fórmula 1 : Equação de Gauss

A aproximação do filtro de Canny para detecção de bordas é G' . Convolucionando a imagem com G' obtemos uma imagem I que mostrará as bordas, mesmo na presença de ruído. A convolução é relativamente simples de ser implementada, mas é cara computacionalmente, especialmente se for em 2-Dimensões. Entretanto, uma convolução de Gauss de 2-dimensões pode ser separada em duas convoluções de Gauss de 1-Dimensão.

A intensidade computacional do detector de bordas de Canny é relativamente alta, e os resultados são geralmente pós-processados para maior clareza. Entretanto, o algoritmo é mais eficiente no processamento de imagens com ruídos ou com bordas difusas.

2.5 Morfologia Matemática

A Morfologia matemática é uma técnica de análise de imagem que se desenvolveu a partir dos anos sessenta. Seu domínio de aplicação inicial foi a análise de imagens microscópicas. Com a evolução dos métodos, houve a diversificação das imagens analisadas (imagens de satélites, análise de cenas, reconhecimento de padrões...).

Em outras palavras, a Morfologia é a forma e estrutura de um objeto ou os arranjos e inter-relacionamentos entre as partes de um objeto.

Os mais antigos usos desta palavra estão relacionados com linguagem e biologia. Em lingüística, morfologia é o estudo da estrutura das palavras. Em biologia, morfologia

esta relacionada mais diretamente a forma de um organismo. A forma de uma folha pode ser usada para identificar uma planta ou a forma de uma colônia de bactérias pode ser usada para identificar sua variedade.

O princípio básico da Morfologia Matemática consiste em extrair informações relativas à geometria e à topologia de um conjunto desconhecido de uma imagem, a partir de transformações de formas, realizadas através de operadores elementares denominados *Erosão*, em que os pixels que não atendem a um dado padrão são apagados da imagem, e *Dilatação*, em que uma pequena área relacionada a um pixel é alterada para um dado padrão. Todavia, dependendo do tipo de imagem sendo processada (preto e branco, tons de cinza ou colorida) a definição destas operações muda, assim cada tipo deve ser considerado separadamente.

Morfologia digital é um caminho para descrever ou analisar a forma de um objeto digital, é uma ciência relativamente recente, pois só os computadores digitais permitiram seu uso na prática. Por outro lado os matemáticos a consideram uso da teoria de conjuntos que é uma área bem estudada.

A idéia de morfologia digital é que uma imagem consiste de um conjunto de “picture elements” (pixels) que são reunidos em grupos tendo uma estrutura bidimensional (forma).

Certas operações matemáticas em conjuntos de pixels podem ser usadas para ressaltar aspectos específicos das formas permitindo que sejam contadas ou reconhecidas.

2.5.1 Tipos de Operações Morfológicas

Operações morfológicas estão divididas em operações morfológicas binárias e operações morfológicas sobre imagens coloridas ou de tons de cinza. Há uma diferença de enfoque nos dois grupos.

Operações morfológicas binárias são definidas em imagens com pixels pretos e brancos somente. Um objeto é considerado sendo um conjunto matemático de pixels pretos,

cada pixel é identificado pelos seus índices de linha e coluna sendo chamado de um ponto no espaço bidimensional.

Nas operações morfológicas sobre imagens com tons de cinza ou cores, as operações de conjunto realizadas não implicam na retirada ou inclusão de um pixel (substituição de seu valor por preto ou branco), mas sim na modificação parcial de seu valor.

2.5.2 Definições Fundamentais

As operações fundamentais associadas com um objeto são o conjunto padrão de operações: *União* $\{ \cup \}$, *Interseção* $\{ \cap \}$, *Diferença* $\{ - \}$, *Disjunção*, *Reflexão* e *Complemento* $\{^c\}$ com *Translação* ;

2.5.3 Dilatação Binária (\oplus)

A dilatação, também às vezes chamada de *dilação*, é uma transformação morfológica que combina dois conjuntos usando adição vetorial. Seu símbolo é \oplus . Como o nome diz, o resultado será uma imagem “engordada”.

A dilatação de um conjunto A pelo conjunto B é definida por:

$$A \oplus B = \{ c \mid c = a + b, a \in A, b \in B \}$$

Onde A representa a imagem sendo operada e B é um segundo conjunto onde é chamado *elemento estrutural* e sua composição define a natureza específica da dilatação, sendo assim a dilatação expande uma imagem.

Ela pode ser representada pela união $A \oplus B = \cup B$.

Seja o conjunto $A = \{ (0,1), (1,1), (2,1), (2,2), (3,0) \}$ e $B = \{ (0,0), (0,1) \}$ então o resultante da dilatação é :

$$A \oplus B = \{ A + \{(x \in B)\} \cup A + \{(x \in B)\} \}$$

$$A \oplus B = \{ (0,1), (1,1), (2,1), (3,0), (0,2), (1,2), (2,2), (2,3), (3,1) \}$$

2.5.4 Erosão Binária (θ)

A erosão basicamente encolhe uma imagem e pode ser vista como uma transformação morfológica que combina dois conjuntos usando vetores de subtração. Ela é expressa como a interseção de A e B. Assim é definido $A \theta B = B \cap A$.

A erosão da imagem A pelo elemento estrutural B pode ser definida como:

$$A \theta B = \{ x \mid x + b \in A \text{ para todo } b \in B \} \quad \text{ou}$$

$$A \theta B = \{ c \mid B' \subseteq A \}.$$

Assim define-se que a erosão é o conjunto de todos os pixels, e o *elemento estruturante* B e transladado pelo c corresponde a um conjunto de pixel em A.

Podemos visualizar a erosão através do exemplo dado a seguir. Assim verifica-se que cada elemento de B e um elemento de A, ou seja B esta contido em A.

2.5.5 Dualidade entre Dilatação e Erosão

A erosão não é o inverso da dilatação, somente em alguns casos de erosão isso ocorre. Erosão e dilatação são duais no seguinte sentido:

$$(A \theta E)' = A \oplus E'$$

Isso significa que o complemento de uma erosão é o mesmo que uma dilatação do complemento da imagem pelo elemento estrutural refletido.

Quando os operadores são duais de outros, estes meios podem ser escritos em termos de um outro. A Dilatação e Erosão são relatados em seguida.

Se B denota a reflexão de B.

$$(A \theta E)' = A' \oplus E$$

$$(A \oplus E)' = A' \theta E$$

Em outras palavras, dilatação a primeiro plano esta a mesmo como erosão a plano fundo, mas o *elemento estruturante* reflete entre estas elas. Igualmente, erosão a plano fundo e dilatação a primeiro plano.

Aplicação de Dilatação e Erosão em uma imagem binária;

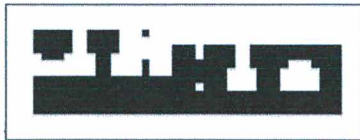


Figura 11.1) Imagem Binária
Original

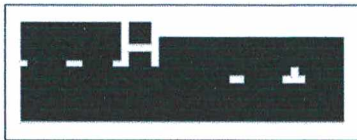


Figura 11.2) Imagem Binária
aplicada Dilatação

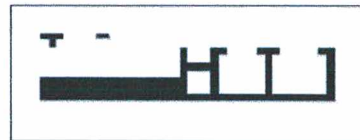


Figura 11.3) Imagem Binária
aplicada Erosão

2.5.6 Abertura (Opening)

A abertura em geral suaviza o contorno de uma imagem, quebra estreitos e elimina proeminências delgadas, a operação de abertura e usada também para remover ruídos da imagem. A abertura de um conjunto A por *elemento estruturante* B e denotado $A \circ B$ e definida como:

$$A \circ B = (A \ominus B) \oplus B$$

A aplicação de uma erosão imediatamente seguida de uma dilatação usando o mesmo elemento estrutural é uma operação de abertura, ela tende a abrir pequenos vazios ou espaços entre objetos próximos numa imagem. Ou seja, em outras palavras uma abertura e uma erosão seguida de uma dilatação usando um mesmo *elemento estruturante*, relembrando que a erosão acha todos os lugares onde o ajuste do *elemento estruturante* esta dentro a imagem, mas isto somente marca esta posição a origem de um elemento.

Porém, uma erosão seguida por uma dilatação, é possível enchermos de brancos os lugares onde o ajuste do *elemento estruturante* estiver dentro do objeto. Na abertura podem se considerar para ser a união de todos, as copias traduzidas de um ajuste do *elemento estruturante* que possa estar dentro do objeto.



Figura 12 : Exemplo de uma Abertura



Figura 13 : Exemplo de uma Abertura em Imagem Médica

2.5.7 Fechamento (Closing)

O fechamento funde pequenas quebras e alargas golfos estreitos elimina pequenos orifícios, se uma abertura cria pequenos vazios na imagem, um fechamento irá preencher ou fechar os vazios, estas operações podem remover muitos dos pixels brancos com ruídos, ou seja basicamente ele é igual a abertura só que primeiramente é feita a dilatação e após é feita a erosão assim ela se define como:

$$A \bullet B = (A \oplus B) \ominus B$$

Em outras palavras o fechamento trabalha de um modo oposto ao método abertura, onde ela remove todos os pixels onde o ajuste do *elemento estruturante* não está dentro da imagem (primeiro plano), fechamento enche todos os lugares onde o *elemento estruturante* não iria ajustar na imagem (plano fundo).

Entretanto operações inversas, abertura e fechamento não irão estabelecer a imagem original.

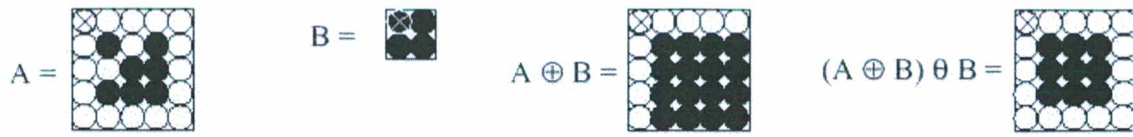


Figura 14 : Exemplo de um Fechamento

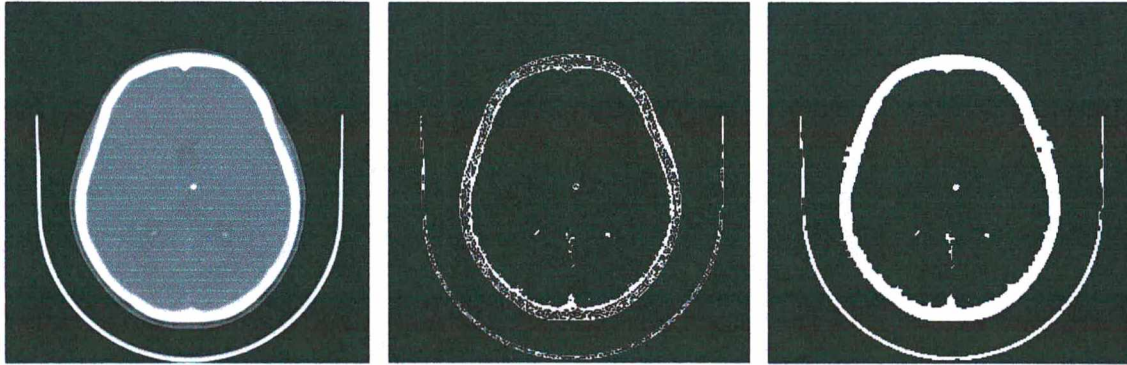


Figura 15 : Exemplo de um Fechamento em Imagem Médica

2.6 Reconstrução 3D

Os objetos do mundo físico possuem formas altamente diversificadas. Seria impossível criar um programa gráfico que tivesse cada uma destas possíveis formas. Por isso, é necessário que se disponha de recursos que permitam a construção de um objeto qualquer a partir de primitivas e propriedades comuns entre os objetos. Entre estas primitivas, a geração de polígonos permite representar uma série de objetos com forma.

A representação tridimensional pode ainda ser realizada de duas maneiras distintas:

- Através da representação de fronteira: os objetos são descritos como um conjunto de superfícies que separam seu interior do meio-ambiente;
- Através da representação de subdivisão de espaço: os objetos são representados através das propriedades de cada ponto no seu interior.

Computacionalmente, os pontos do interior são agrupados geralmente em pequenos cubos, cada um com característica própria. Juntos eles definem a aparência e as propriedades do objeto como um todo. Este tipo de representação é utilizado quando deseja-se analisar o interior de objetos, como por exemplo o corpo humano.

2.6.1 Sistema de Coordenadas

Um desenho é composto de elementos básicos combinados: pontos, segmentos de reta e áreas (espaços preenchidos), cada um com determinadas características como cor, dimensionamento, tipo de traçado (tracejado, pontilhado, etc...).

A adoção de um sistema de coordenadas faz-se importante, pelo fato de possibilitar a descrição matemática do posicionamento relativo dos elementos que compõem a desenho. O sistema de coordenadas, também denominado sistema de referências define uma referência (ou origem) em relação à qual se descrevem todos os posicionamentos.

Existem os sistemas de coordenadas cartesianas e o de coordenadas polares. O sistema de coordenadas cartesianas divide-se em bidimensional (coordenadas X e Y) e o tridimensional (coordenadas X, Y e Z). Este trabalho usa o sistema de coordenadas cartesianas tridimensionais, pois objetos serão animados em três dimensões.

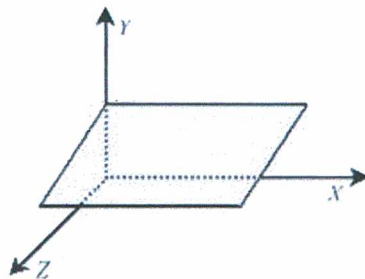


Figura 16 : Modelo de Coordenadas Tridimensional

2.6.2 Polígonos

A forma mais comum de representação tridimensional é através de um conjunto de superfícies poligonais que delimitam o interior de um objeto. Os programas gráficos geralmente traduzem a descrição do objeto em termos de superfícies poligonais. A vantagem deste método é que as superfícies planas formadas permitem um tratamento através de equações lineares, o que acelera o cálculo.

Um polígono pertencente a um dado objeto é caracterizado pelo conjunto de vértices que o constitui, e pelos atributos da sua superfície, como por exemplo a refletividade, a transparência, etc.

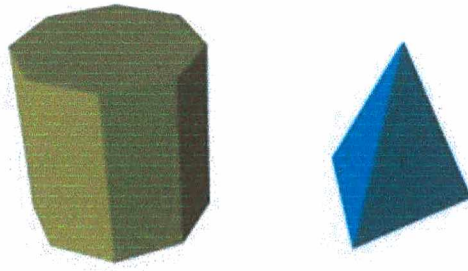


Figura 17 : Objetos formados por polígonos

2.6.3 Triangularização de Delaunay e Diagrama de Voronoi

Uma triangularização de um conjunto de pontos consiste em encontrarmos segmentos de reta que conectam estes pontos de tal modo que nenhum desses segmentos cruze com nenhum outro e que cada ponto seja vértice de pelo menos um triângulo formado por esses segmentos. Esses segmentos particionam o conjunto de pontos em triângulos, daí o nome Triangularização.

O grafo dual de um Diagrama de Voronoi constitui uma triangularização cujos pontos são os pontos construtores do diagrama de Voronoi. A esta triangularização particular dá-se o nome de Triangularização de Delaunay. Para uma melhor compreensão sobre o método de Triangularização de Delaunay, inicialmente será explicado o Diagrama de Voronoi.

O *diagrama de Voronoi* é uma estrutura geométrica proposta no início do século e que é capaz de responder uma grande variedade de perguntas a respeito de proximidade em um conjunto de pontos: qual ponto está mais próximo, qual o mais distante, entre outras.

A construção do diagrama pode ser melhor compreendida observando o que ocorre quando o número de locais vai sendo aumentado gradativamente. Inicialmente, considere-se apenas dois locais, $p1$ e $p2$. O diagrama de Voronoi consiste na reta que secciona ao meio o segmento $p1p2$ e é perpendicular a este, a *mediatriz* do segmento (Figura 19).

Todos os pontos da reta são igualmente próximos a $p1$ e a $p2$. Pontos no semiplano que contém $p1$ constituem o polígono de Voronoi correspondente a $p1$, e analogamente o outro semiplano corresponde a $V(p2)$.

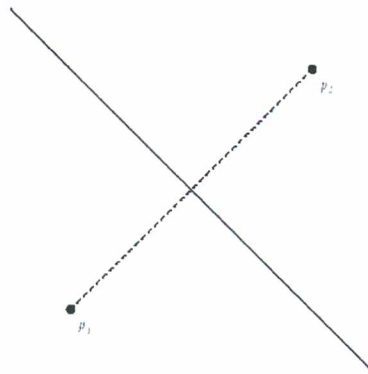


Figura 18 – Diagrama de Voronoi com dois locais

Expandindo para três locais, é fácil perceber que o diagrama de Voronoi será formado pelas semi-retas que cortam as arestas de $p_1p_2p_3$ ao meio e segundo uma perpendicular, portanto as mediatrizes das arestas, partindo do *circuncentro* do triângulo (Figura 20). O circuncentro é o centro do círculo definido pelos vértices do triângulo, e é possível que ele não pertença ao triângulo.

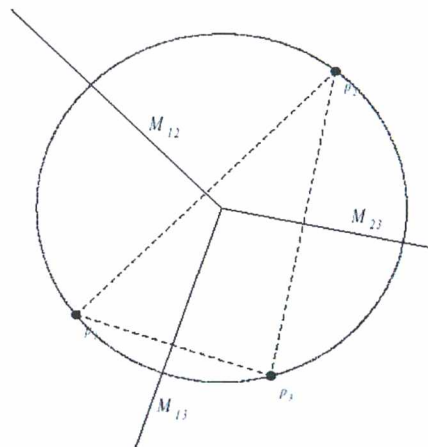


Figura 19 – Diagrama de Voronoi com três locais

Generalizando para um número maior de locais, fica claro que o processo de construção deve levar em conta as mediatrizes dos segmentos definidos entre cada par de locais. A mediatriz entre os locais p_i e p_j será denotada como M_{ij} . Seja S_{ij} o semiplano definido por M_{ij} e que contém p_i . Então S_{ij} contém todos os pontos do plano que estão mais

próximos de p_i do que de p_j . Para obter o polígono de Voronoi de p_i , é necessário combinar todos os semiplanos S_{ij} com $i \neq j$, e portanto

$$V(p_i) = \bigcap_{i \neq j} S_{ij}$$

Como semiplanos são, por definição, convexos (não existe nenhum segmento definido entre dois pontos do semiplano e que contém pontos que não pertençam a ele). A interseção de conjuntos convexos é também um conjunto convexo [REF]. Portanto, pode-se concluir que qualquer polígono de Voronoi é convexo também.

A Figura 21, mostra um exemplo de diagrama de Voronoi com 12 locais. Observe-se que existem 12 polígonos, um para cada local.

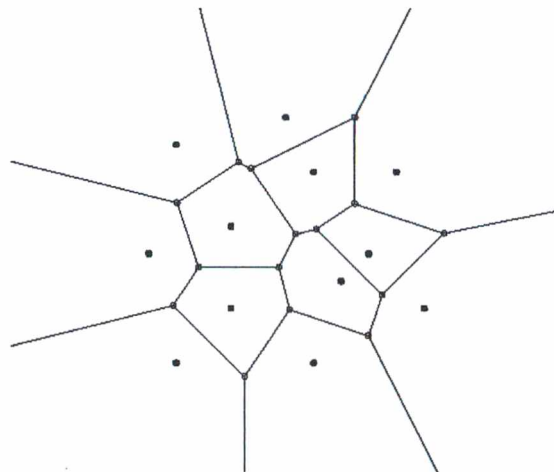


Figura 20 - Diagrama de Voronoi com 12 locais

Assim, considerando que o máximo grau de um vértice seja igual a três, define-se o dual de um diagrama de Voronoi através de um grafo G , cujos nós correspondem aos locais e cujos arcos conectam locais cujos polígonos de Voronoi compartilham uma aresta do diagrama.

Traçando G com linhas retas entre os vértices (locais), é produzida uma *triangulação* de P , denominada *Triangulação de Delaunay*. A Figura 22 mostra a triangulação de Delaunay sobre o diagrama de Voronoi da Figura 21.

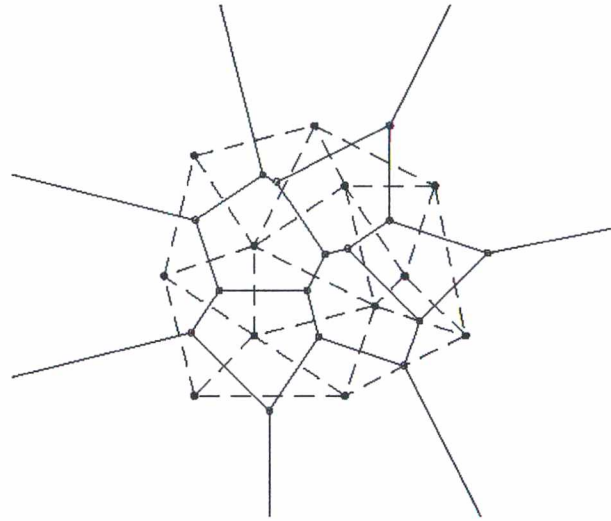


Figura 21 : Diagrama de Voronoi (linhas cheias) e triangulação de Delaunay (linhas pontilhadas)

Um triângulo da Triangularização de Delaunay tem a seguinte propriedade: ele determina um círculo cujo interior não contém nenhum outro ponto do conjunto de pontos a não ser os três pontos que determinam o triângulo. Você pode visualizar uma Triangularização de Delaunay para um conjunto de 10 pontos na figura abaixo.

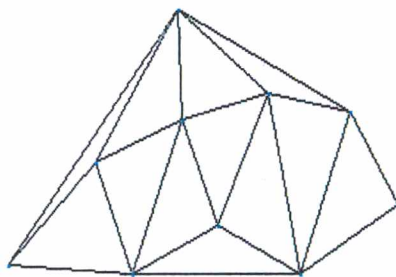


Figura 22 : Triangularização de Delaunay

2.7 Representação de Superfícies

2.7.1 VRML (Virtual Reality Modeling Language)

Em meados de 1994, Tim Berners-Lee, numa conferência, falou sobre a necessidade de um padrão para uma *Web* 3D. Dessa forma, foi criado o nome VRML como um acrônimo ao termo *HTML* (*Hyper Text Markup Language*), logo sendo alterado para *Virtual Reality Modeling Language*, buscando refletir a ênfase em mundos virtuais ao invés de páginas de texto.

Assim, com o auxílio de uma comunidade de artistas, engenheiros e visionários, a Silicon Graphics lançou, no final de 1994, a versão 1 da linguagem. Porém sua especificação só ficou totalmente pronta em 1995.

VRML é abreviação de ou linguagem para modelagem em Realidade Virtual. É uma linguagem independente de plataforma que permite a criação de cenários em 3D, visualizar objetos por ângulos diferentes e interagir com eles através de um *browser*.

A linguagem foi concebida para descrever simulações interativas de múltiplos participantes, em mundos virtuais disponibilizados na *Internet* e ligados com *WWW* (*World Wide Web*), mas a primeira versão da linguagem não possibilitou muita interação do usuário com o mundo virtual. Nas versões futuras foram acrescentadas características como animação, movimentos de corpos, som e interação com o usuário. A última versão é a 2.0, chamada *Moving Worlds VRML 2.0*.

O código de VRML é um subconjunto de formato ASCII do *Open Inventor*, da *Silicon Graphics*, com características adicionais para navegação na *WEB*. Esta característica é semelhante às âncoras em HTML. Assim, pode-se criar âncoras em um ambiente virtual que levem a outros ambientes virtuais.

A linguagem, na sua versão 1.0, trabalha com geometria 3D, permitindo a criação de objetos baseados em polígonos, possuindo alguns objetos pré-definidos como triângulo, esfera, cilindro, cubo e cone, suportando transformações como rotação, translação e escala.

Permite também a aplicação de texturas, luz, sombreamento, etc. Outra característica importante da linguagem é o Nível de Detalhe (LOD, *Level of Detail*), que permite o ajustamento da complexidade dos objetos, dependendo da distância, em que eles se encontram, do observador.

A versão 2.0 permite a criação de objetos tridimensionais estáticos e dinâmicos em uma cena, bem como comportamentos para esses objetos. Também possibilita a prototipação, onde as partes de uma mesma cena podem ser criadas separadamente.

O principal benefício da utilização de VRML para a criação de mundos virtuais é a possibilidade de se manipular e interagir com os objetos presentes em cenas localmente ou remotamente através da Internet. Existem atualmente muitos visualizadores dos gráficos gerados pela linguagem.

Tudo que se precisa para criar uma cena em VRML é um editor de textos. Uma vez que os arquivos são salvos em formato texto com a extensão **.wrl*. A linguagem apenas descreve como os ambientes tridimensionais devem ser apresentados, pois os arquivos não são compilados. A versão 1.0 é basicamente uma simplificação do *Open Inventor*, com propriedades de materiais, transformações, visões de câmera, texturas mapeadas e iluminação. Há pouca possibilidade de interação, tendo como principal objetivo a criação de mundos virtuais estáticos e criação de âncoras para outros ambientes.

A versão 2.0 permite que os objetos do mundo virtual possam se movimentar e responder a eventos, baseados nas iniciativas do usuário e comportamentos baseados em *scripts*. É permitido também a utilização de objetos multimídia, como sons e filmes. As principais características desta versão são: melhoria dos mundos estáticos, interação, animação, comportamento baseado em *scripts*, prototipação de novos objetos VRML e criação de *links* interligando os mundos virtuais.

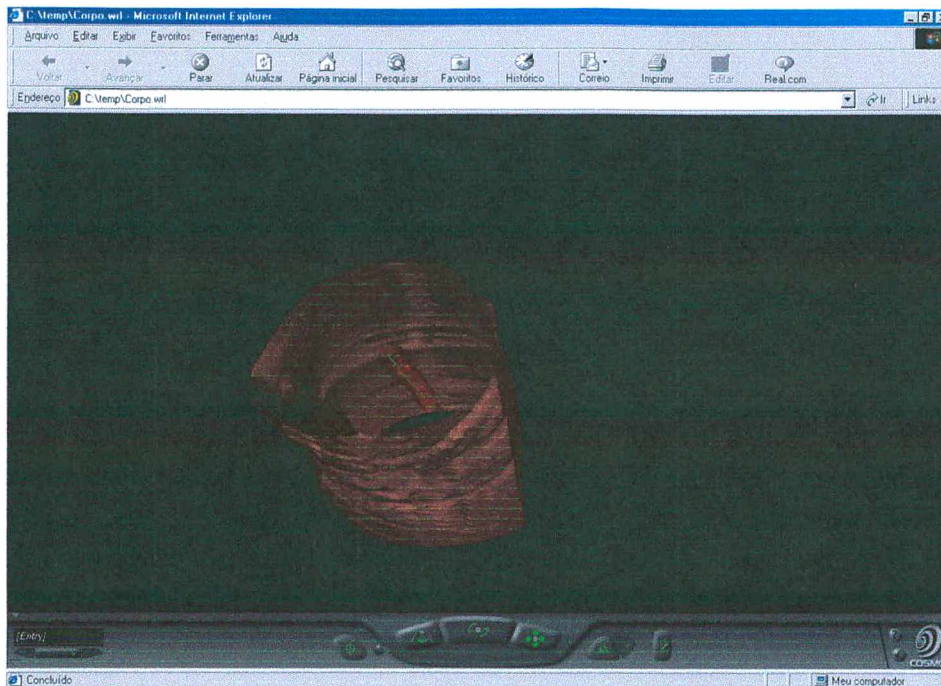


Figura 23 : Reconstrução 3D utilizando VRML

2.7.2 OpenGL (Open Graphics Library)

OpenGL é uma biblioteca de rotinas gráficas de modelagem, manipulação de objetos e exibição tridimensional que permite a criação de aplicações que usam Computação Gráfica. Seus recursos permitem ao usuário criar objetos gráficos com qualidade, de modo rápido, além de incluir recursos avançados de animação, tratamento de imagens e texturas é possível ter visualização em vários ângulos.

A biblioteca OpenGL foi introduzida em 1992 pela Silicon Graphics, no intuito de conceber uma API (Interface de Programação de Aplicação) gráfica independente de dispositivos de exibição. Com isto, seria estabelecida uma ponte entre o processo de modelagem geométrica de objetos, situadas em um nível de abstração mais elevado, e as rotinas de exibição e de processamento de imagens implementadas em dispositivos (hardware) e sistemas operacionais específicos.

As função utilizada pelo OpenGL para desenhar um ponto na tela, por exemplo, possui os mesmos nome e parâmetros em todos os sistemas operacionais nos quais OpenGL foi implementada, e produz o mesmo efeito de exibição em cada um destes sistemas.

Diante das funcionalidades providas pelo OpenGL, tal biblioteca tem se tornado um padrão amplamente utilizado na indústria de desenvolvimento de aplicações. Este fato tem sido adotado também pela facilidade de aprendizado, pela estabilidade das rotinas e pelos resultados visuais consistentes para qualquer sistema de exibição concordante com este padrão.

Diversos jogos, aplicações científicas e comerciais tem utilizado OpenGL como ferramenta de apresentação de recursos visuais, principalmente com a adoção deste padrão por parte dos fabricantes de placas de vídeo destinadas aos consumidores domésticos.

Todas as rotinas do OpenGL são implementadas na linguagem C, tornando fácil sua utilização em qualquer programa escrito em C ou C++.

Entre os recursos gráficos disponíveis pelo OpenGL, podem ser destacados os seguintes:

- Modos de desenho de pontos;
- Ajuste de largura de linhas;
- Aplicação de transparência;
- Ativação/desativação de serrilhamento (aliasing);
- Mapeamento de superfícies com textura;
- Seleção de janela de desenho;
- Manipulação de fontes/tipos de iluminação e sombreadamento;
- Transformação de sistemas de coordenadas;
- Transformações em perspectiva;
- Combinação de imagens (blending);

As implementações do OpenGL geralmente provêm bibliotecas auxiliares, tais como a GLU (OpenGL Utility library), utilizada para realizar tarefas comuns, tais como manipulação de matrizes, geração de superfícies e construção de objetos por composição.

As especificações do OpenGL não descrevem as interações entre OpenGL e o sistema de janelas utilizado (Windows, X Window etc). Assim, tarefas comuns em uma aplicação, tais como criar janelas gráficas, gerenciar eventos provenientes de mouse e teclado, e apresentação de menus ficam a cargo de bibliotecas próprias de cada sistema operacional.

3 ESTADO DA ARTE

Alguns trabalhos estão sendo desenvolvidos no sentido da disponibilização de reconstruções 3D de imagens radiológicas, onde podemos citar:

3.1 Cyclops StentPlanner

O "*Cyclops Stent Planner*"² [14], tem por finalidade auxiliar no planejamento do tratamento cirúrgico endoluminal da doença de aneurisma de aorta abdominal. Para tanto, foi desenvolvido um software que é possível reconstruir e mensurar a aorta e o aneurisma abdominal a partir de tomografias computadorizadas, helicoidais ou não.

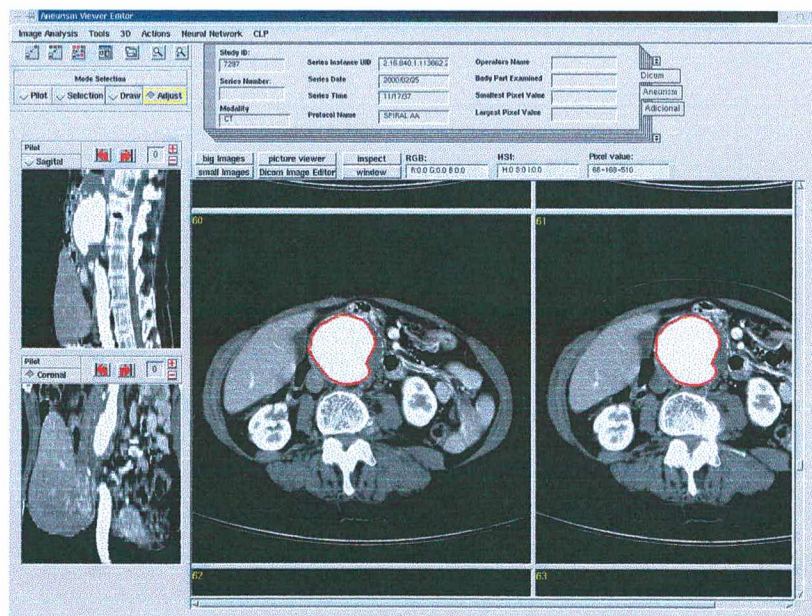


Figura 24 : Ilustração da interface do Cyclops Stent Planner

² O trabalho "*Cyclops Stent Planner*" está sendo desenvolvido no âmbito do projeto *Cyclops* pelo aluno Herculano Haymussi De Biasi, mestre do Curso de Ciência da Computação da Universidade Federal de Santa Catarina.

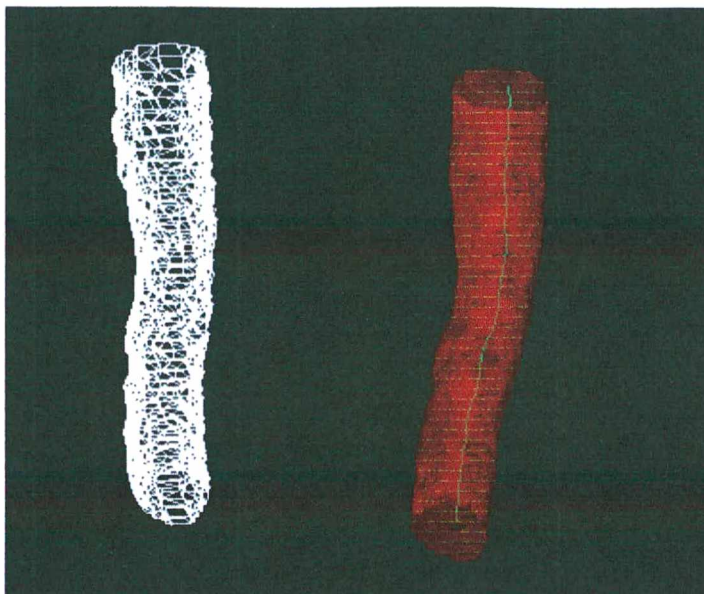


Figura 25 : Reconstrução 3D realizada através do Cyclops Stent Planner

3.1.1 Discussão

Neste projeto é utilizado, entre outras técnicas, o reconhecimento de imagens, em especial da artéria aorta e do aneurisma abdominal, segmentando as estruturas, sendo possível fazer a correta mensuração dos mesmos gerando dados mais confiáveis para a produção de endopróteses necessárias ao tratamento da patologia. Também é possível a reconstrução tridimensional das estruturas anatômicas.

Este trabalho é parte integrante do Projeto Cyclops de desenvolvimento de ferramentas inteligentes para análise de imagens médicas

3.2 CyclopsPopCorn

O "*Cyclops PopCorn*"³ [15], tem por objetivo auxiliar os médicos radiologistas em suas tarefas diárias de análise, através de um ferramenta de software gráfico. Dentre suas funcionalidades podemos citar, a identificação, contagem e mensuração de calcificações relacionadas com Neurocisticercose (NC) em imagens de Tomografia Computadorizada, através de uma visualização 3D dos achados anatomopatológicos e simplificar a identificação de áreas do cérebro afetadas.

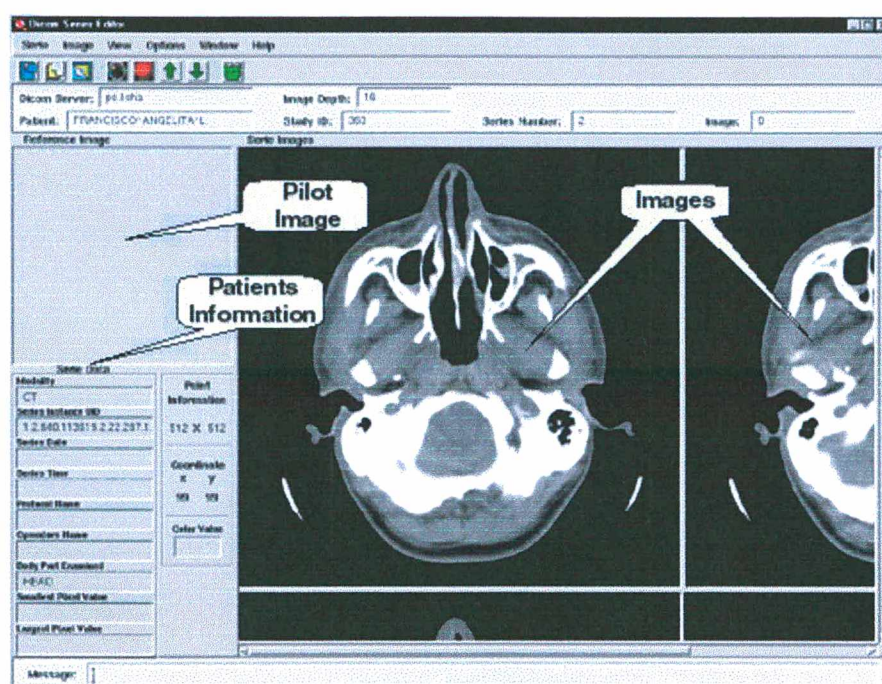


Figura 26 : Ilustração da interface do Cyclops PopCorn

³ O trabalho "*Cyclops PopCorn*" está sendo desenvolvido no âmbito do projeto *Cyclops* pelo aluno Eros Comunello, mestre do Curso de Ciência da Computação da Universidade Federal de Santa Catarina.

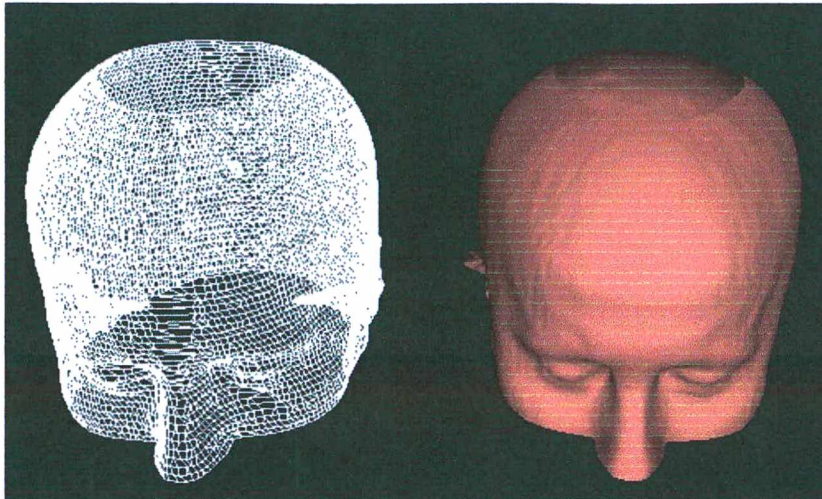


Figura 27 : Reconstrução 3D realizada através do Cyclops PopCorn

3.2.1 Discussão

Este trabalho também é parte integrante do Projeto Cyclops de desenvolvimento de ferramentas inteligentes para análise de imagens médicas.

3.3 VPAT – Visualização e interação com Pacientes Virtuais

O “VPAT – Visualização e interação com Pacientes Virtuais” é um projeto da Universidade Federal do Rio Grande do Sul (UFRGS) e tem por objetivo a criação de "pacientes virtuais", ou melhor, a geração de modelos de representação de seres humanos virtuais para uso em aplicações de computação gráfica na área médica.

Este projeto envolve reconstrução tridimensional de partes da anatomia humana partir de imagens médicas como a síntese de corpos virtuais e seus movimentos.

3.3.1 Discussão

Projeto somente no papel e conta com uma equipe de seis pessoas para seu desenvolvimento.

3.4 3D-DOCTOR

O 3D-DOCTOR é um software para fins comerciais que consiste na manipulação, mensuração e reconstrução de imagens radiológicas.

Utilizando imagens radiológicas providas de aparelhos de Tomografia Computadorizada e Ressonância Magnética o software 3D-DOCTOR cria superfícies em 3D e realiza mensuração de volume para uma série de cortes de imagens em 2D e 3D.

Os modelos gerados após a reconstrução 3D podem ser salvos em formato 3D, e visualizados no próprio software ou exportados em VRML.

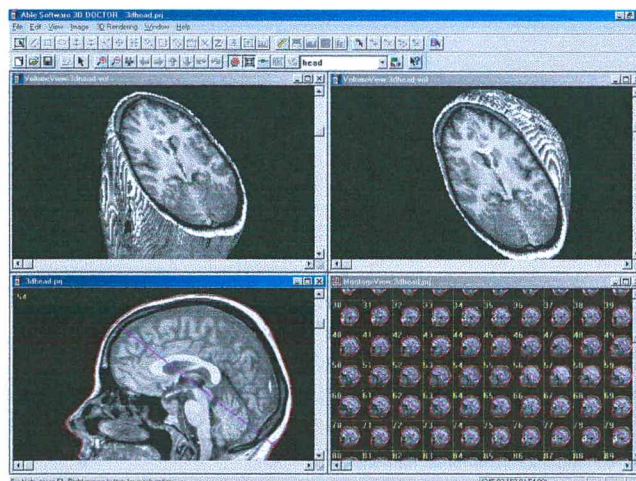


Figura 28 : 3D-DOCTOR

3.4.1 Discussão

A partir das pesquisas realizadas no software 3D-DOCTOR foi possível descrever algumas características.

O software está disponível somente para o sistema operacional Windows 9X e NT, e seu requisito mínimo de memória é de 32MB RAM, apesar de recomendar 128MB RAM para que haja uma melhor performance na reconstrução 3D de imagens, devido o tamanho da imagem após este procedimento.

4 DESENVOLVIMENTO DO TRABALHO

Neste capítulo será explanado o desenvolvimento do trabalho, especificando as fases da elaboração e desenvolvimento do mesmo.

4.1 Obtenção e Visualização das imagens

As imagens utilizadas para este trabalho foram adquiridas na clínica DMI – Diagnóstico Médico por Imagem, que por sua vez participa no Projeto Cyclops auxiliando no fornecimento de imagens e esclarecimentos médicos, sempre que necessário.

As imagens adquiridas foram importadas para um banco de dados DICOM para serem visualizados através do aplicativo *DicomSeriesEditor* e gerenciadas pelo *DicomEditor*.

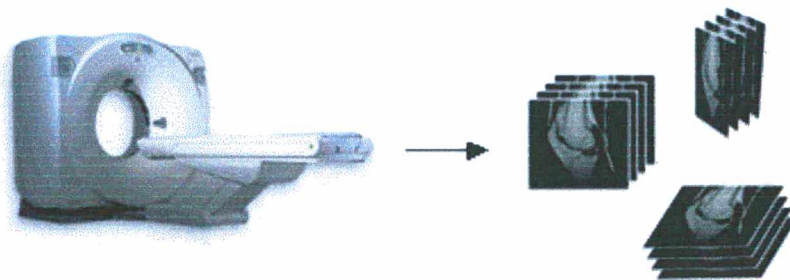


Figura 29: Aquisição de Imagens

O *DicomEditor* é um aplicativo cliente do banco de dados DICOM responsável pela visualização e gerenciamento dos dados dos pacientes e suas respectivas séries de imagens. Trata-se de um módulo desenvolvido anteriormente dentro do projeto Cyclops, que é encarregado de fazer a comunicação com o banco de dados para trazer as informações solicitadas pelo usuário. Essas informações podem ser relativas ao paciente (nome, idade, sexo, data de nascimento, etc.), aos exames que ele realizou (data e hora do exame) e à

série de imagens de um determinado exame (tipo de contraste utilizado, data de aquisição da imagem, entre outros).

Na Fig.30 é mostrado o *DicomEditor*, o qual apresenta os pacientes e seus respectivos estudos e séries. Na janela à direita são exibidos vários dados relacionados ao paciente, estudo, série, equipamento, etc. Na janela inferior tem-se a visualização de miniaturas dos exames.

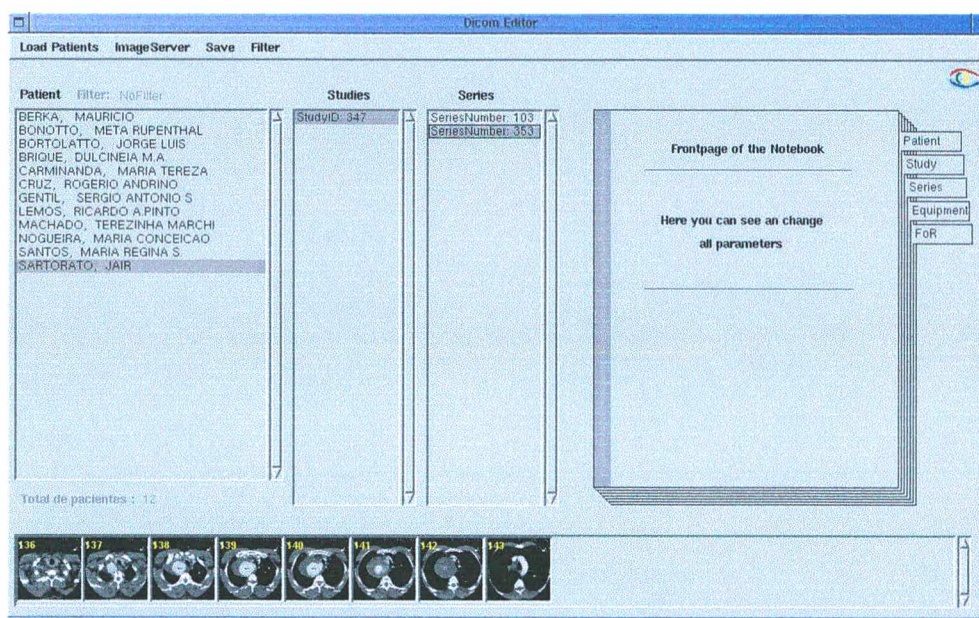


Figura 30: Interface DicomEditor

Para a visualização e manipulação de imagens é utilizado o *DicomSeriesEditor*, onde tem-se a possibilidade de observar uma série de imagens e executar tarefas relacionadas à análise e ao tratamento destas imagens.

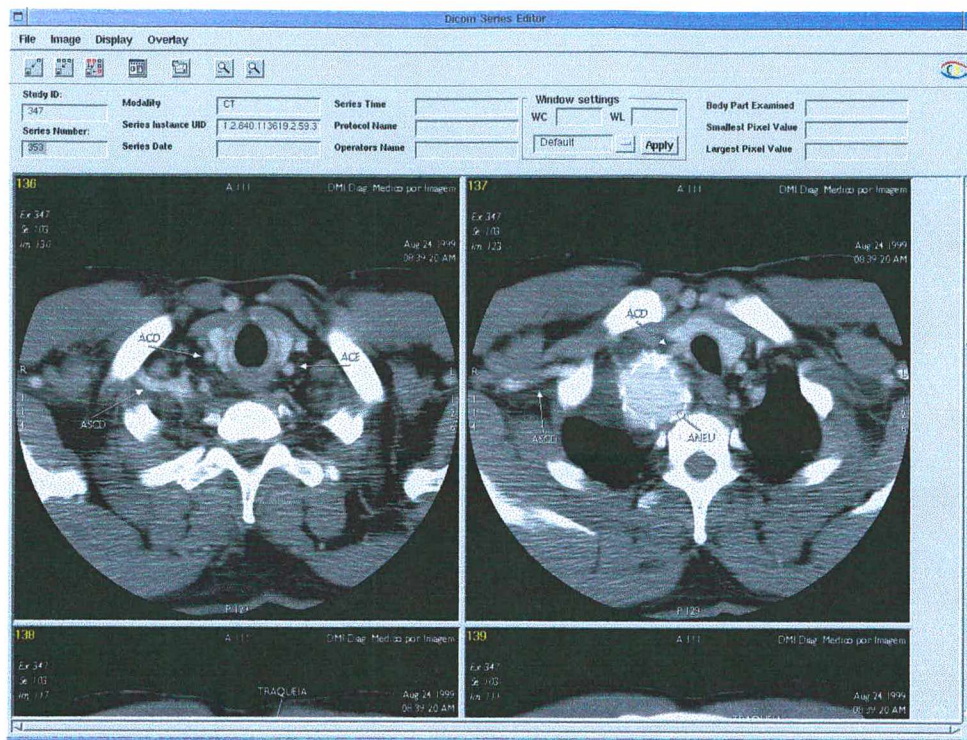


Figura 31: Interface DicomSeriesEditor

Neste projeto, foi necessário ampliar a funcionalidade deste aplicativo para o sistema conseguir realizar o propósito requerido. Para isso, foi criado um novo aplicativo denominado 3DEditor, responsável pela manipulação de imagens e execução de rotinas de processamento de imagens e reconstrução 3D.

4.2 Implementação

No âmbito do desenvolvimento, logo após de obter as imagens, o primeiro passo foi o desenvolvimento de uma interface, onde seria possível visualizar essas imagens e algumas informações referentes à identificação das mesmas.

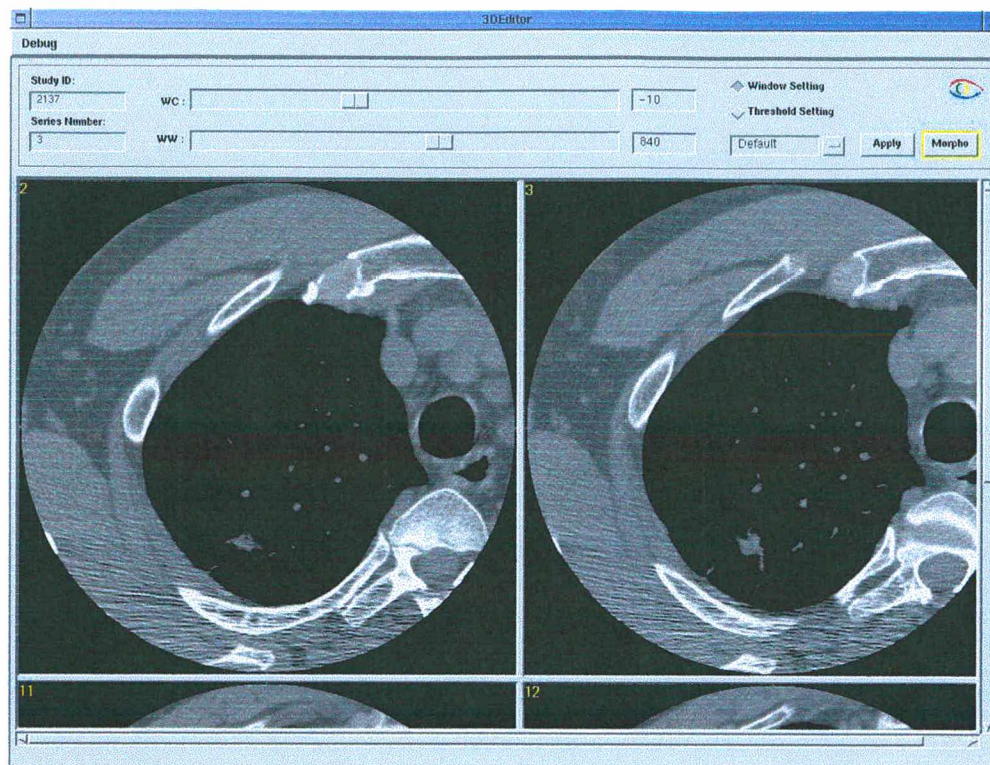


Figura 32: Interface 3DEditor

Dentro desta interface, foi criado dois “sliders” referentes aos valores de *WindowWidth* e *WindowLevel* da imagem, onde o médico radiologista pode manipular os parâmetros de densidade e assim obter o resultado desejado, para o propósito requerido. Tais valores indicam a largura da paleta (quantidade de diferentes tons de cinza) e o seu centro na escala. Estes parâmetros não indicam valores na paleta e sim valores na escala Hounsfield. Esta escala define valores entre -1000 (corresponde ao preto) e +1000 (corresponde ao branco).

Após obter a visualização da imagem requisitada é possível aplicar a técnica baseada em segmentação por limiarização (Thresholding).

A técnica de limiarização consiste em definir um valor limiar para a cor de um pixel. Pixels com valores de cor acima do limiar se tornam pretos e de valor abaixo se tornam brancos.

A conversão de cores corresponde na modificação dos valores de pixels que compõem a imagem. Estes valores são convertidos de acordo com uma regra de três simples. Uma imagem é representada por uma matriz de pixels, onde cada elemento da matriz possui um valor que representa o índice da paleta de cores da imagem. A paleta de cores é um vetor, onde cada posição contém um valor da intensidade da cor. Este valor é formado por um conjunto de três valores, um que representa a intensidade de azul, um que representa o vermelho e outro que representa o verde. Um valor de cinza é representado quando os três valores são iguais. Por exemplo, o preto é representado pelo valor 0 para todas as intensidades e o branco, pelo valor 1 para todas as intensidades. Uma imagem de 8 bits significa que cada pixel possui 8 bits (1 byte) para representação de seu índice na paleta de cores.

Em seguida, para dar continuidade ao procedimento, é chamada uma nova interface chamada 3DEditorBrowser, onde será considerada as imagens previamente alteradas pelas técnicas acima descritas.

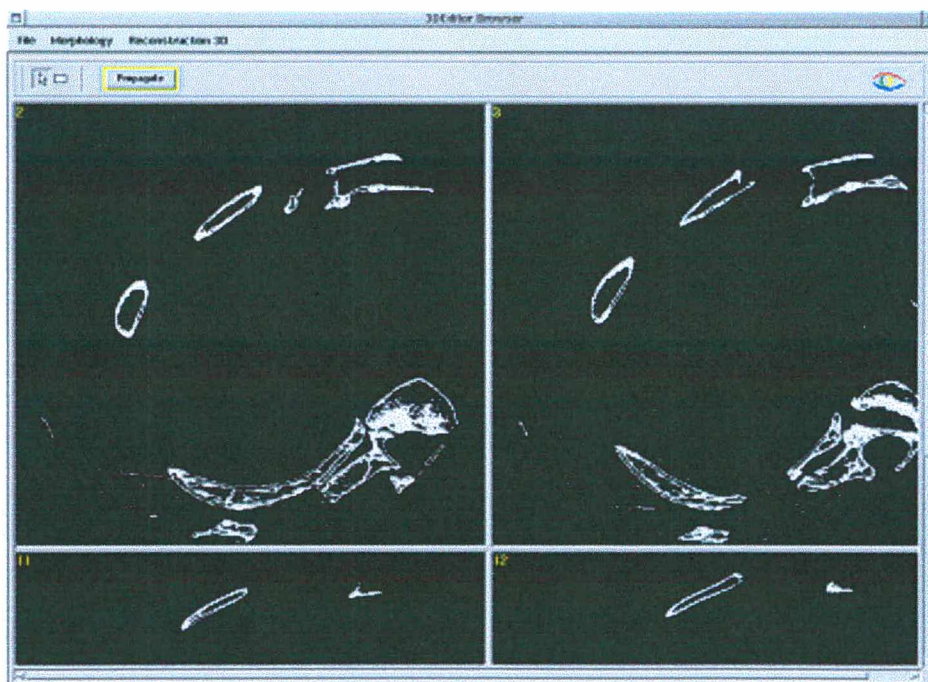


Figura 33: Interface 3DEditorBrowser (imagens alteradas)

Para eliminar os ruídos resultantes visualizadas na figura 38, é realizada uma operação de morfologia chamada Fechamento Morfológico (Closing) seguido de uma operação de Abertura Morfológica (Opening). O principal objetivo dessas duas aplicações de morfologia é o de eliminar segmentos que não correspondem com estruturas anatômica em questão.

Para executar os algoritmos de morfologia matemática Opening (*vopen*) e Closing (*vclose*), já existente na ferramenta *Khoros*, foram criadas rotinas para executar externamente esses métodos, onde o software exporta as imagens a serem processadas no formato VIFF 8 bits, formato padrão das aplicações *Khoros*. Para os algoritmos responsáveis por tais técnicas (*vopen* e *vclose*) é passado um arquivo de entrada, que por sua vez, retornará um arquivo no mesmo formato com a imagem processada.



Figura 34: Interface 3Deditor browser (imagens alteradas por morfologia)

Estas imagens resultantes do processamento visualizadas na figura 34, serão exibidas nesta nova interface, onde o usuário poderá escolher a estrutura ou patologia em questão a ser reconstruída, e executar os métodos disponíveis para reconstrução 3D e exportação na linguagem VRML. Desta forma, é possível o usuário retornar na janela anterior, caso o resultado não tenha sido satisfatório e fazer alterações nos parâmetros de *window*.

5 METODOLOGIA DE RECONSTRUÇÃO 3D

Neste capítulo será discutido a metodologia utilizada e pesquisada para o desenvolvimento do trabalho proposto.

Normalmente podemos encontrar vários métodos disponíveis que podem auxiliar a reconstrução 3D de imagens. Porém, nem sempre estes métodos são adequados e eficazes para o propósito requerido.

Neste trabalho foi utilizada uma técnica de reconstrução criada pelo projeto Cyclops. É importante ressaltar que a principal fase de uma reconstrução tridimensional esta na triangularização.

A triangularização é uma maneira de se criar faces para uma série de imagens, fazendo com que ela tenha uma representação em 3D. A forma triangular é a forma mais utilizada nesta área de reconstrução, pois é possível considerar a única forma que, quando utilizada unicamente, sua junção permite recriar praticamente todas as formas necessárias em uma reconstrução.

Para transformar uma imagem 2D em 3D, o primeiro passo é calcular o polígono que corresponde a borda da imagem. Neste caso, o software calcula quais são os vértices do polígono. O ponto relativo a cada vértice do polígono é igual a posição do pixel na imagem. Convém lembrar que todos os pixels são pontos em duas dimensões.

Após a obtenção do polígono em duas dimensões, é executado um método para gerar os polígonos em três dimensões.

Os métodos responsáveis pela execução destes processos são: `createPolygons` e `getEdgePoints`, além do método de inicialização da classe `Polygon3D` já implementada no âmbito do Projeto Cyclops.

Outra possibilidade é a escolha do nível de "suavidade" do polígono, ou seja, do número de pontos da imagem original que formarão o polígono. Um polígono com mais pontos produzirá uma reconstrução com uma qualidade sensivelmente maior. O ponto negativo é que quanto mais detalhado for um polígono maior será o tempo necessário para

processamento, maior será o arquivo VRML gerado e também mais lenta será a manipulação desse arquivo no browser.

Com a série de polígonos dá-se início a fase de triangularização. Esta fase é de suma importância para a reconstrução, pois esses triângulos receberão faces no momento que a reconstrução for mostrada ao usuário, e, se uma dessas faces não estiver correta, a imagem como um todo, ficará prejudicada ocorrendo uma distorção do que seria uma cópia fiel de um paciente e/ou seus achados.

Para a demonstração de uma triangularização, considere o exemplo abaixo:

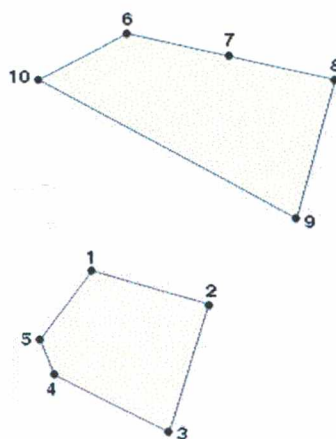


Figura 35: Duas imagens a serem reconstruídas em 3D

Num primeiro momento calcula-se um ponto intermediário (o ponto do meio) entre dois pontos de uma mesma imagem em 2D (Ex: 1 e 2 = 1,5). Baseado neste ponto, procura-se o ponto mais próximo dele na imagem subsequente levando sempre em conta suas distâncias em 3D.

Pode-se observar a Figura 36 que faz uma representação das conexões feitas e na Figura 37 que é a reconstrução do exemplo após a primeira fase. A “tabela 02” permite uma melhor compreensão do resultado desta primeira parte da triangularização.

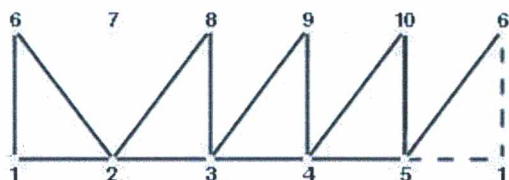


Figura 36: Representação das conexões após a primeira fase

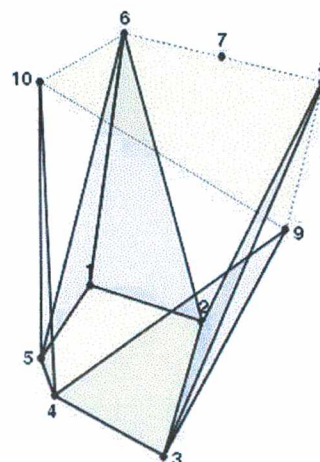


Figura 37: Imagem das conexões após a primeira Fase

Pode-se também observar na linha “E” da “Tabela 02” que é feita a conexão do último com o primeiro ponto (baseado na imagem em 2D).

Importante: Cada um dos valores mencionados no exemplo (1, 2, 3,...10) correspondem a uma coordenada em um espaço 3D (x, y, z).

Triangulações	ponto 1 (p1)	Ponto 2 (p2)	ponto 3 (p3)
A	1	2	6
B	2	3	8
C	3	4	9
D	4	5	10
E	5	1	6

Tabela 02 : Resultado da primeira fase da Triangulação

Nesta primeira fase faz-se a triangularização utilizando todos os pontos de uma das imagens em questão, a outra será feita na próxima fase. É importante salientar que o

primeiro e o último ponto do polígono de uma imagem possuem a mesma coordenada, fazendo com que o polígono das imagens esteja sempre fechado.

Na segunda fase, é utilizado os pontos de p2 resultantes da primeira fase para se completar as conexões. Utiliza-se um laço que percorra por todos os pontos da segunda imagem, fazendo a seguinte conexão:

$$p1 = n + 1$$

$$p2 = n$$

$p3 = p2$ de n na Tabela 02 ou seja,

$$p1 = 7, p2 = 6 \text{ e } p3 = 2;$$

$$p1 = 8, p2 = 7 \text{ e } p3 = 2; \dots \text{etc.}$$

Desta forma teremos todos os pontos conectados por triângulos que nos dará o resultado representado na “Tabela 03”.

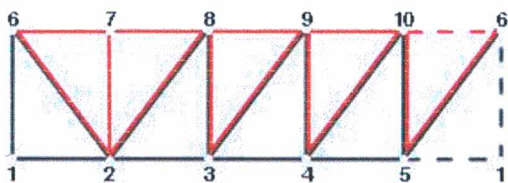


Figura 38: Representação das conexões após a segunda fase

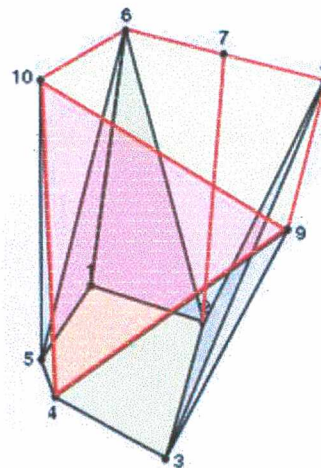


Figura 39: Imagem das conexões após a segunda fase

Uma representação das conexões pode ser observada na Figura 21, e a respectiva reconstrução das imagens em 3D pode ser visualizada na Figura 22.

Triangulações	Ponto 1 (p1)	Ponto 2 (p2)	ponto 3 (p3)
F	7	6	2
G	8	7	2
H	9	8	3
I	10	9	4
J	6	10	5

Tabela 03 : Resultado da segunda fase da Triangulação

Assim, pode-se reconstruir imagens em 3D a partir de imagens 2D.

Quanto às imagens que possuem algum tipo de “bifurcação”, o processo para a reconstrução tridimensional torna-se um pouco mais complexo, pois desta maneira um corte pode possuir mais de um objeto a ser reconstruído.

Portanto, para tal finalidade foi criado métodos diferenciados para reconstruir em 3D essas imagens, onde a triangularização desses cortes seria da mesma forma acima explanada, porém passando os dados de forma diferente do método comum.

A ilustração abaixo mostra um teste realizado simulando uma imagem que possui mais de um objeto a ser reconstruído em um mesmo corte com as técnicas já existentes no 3DEditor para reconstrução. Como é possível notar as ligações internas se cruzam, não permitindo que seja visualizado internamente.

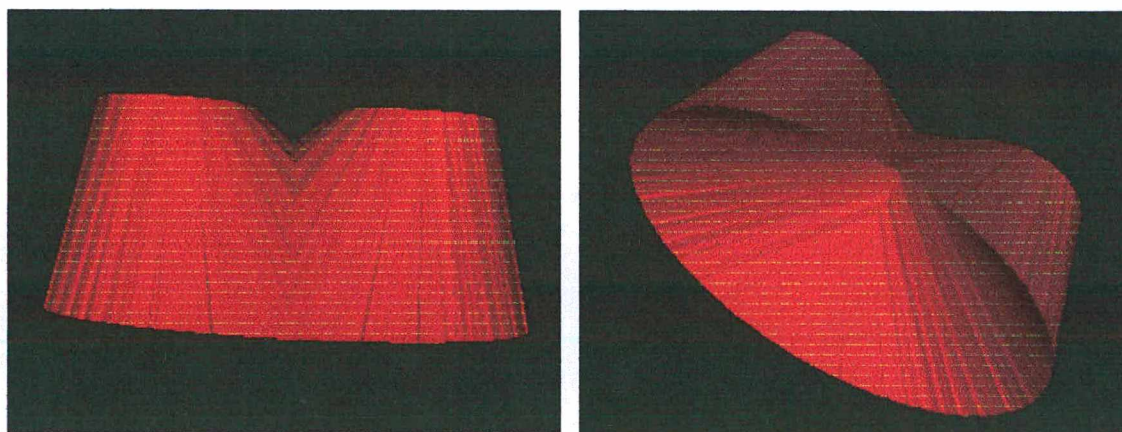


Figura 40 : Reconstrução com Bifurcação (método antigo)

Para que fosse resolvido esse impasse, foi dado início a uma rigorosa pesquisa e a uma série de testes, onde foi possível a implementação de uma técnica de reconstrução 3D para imagens com “bifurcações”.

É importante ressaltar que o modelo elaborado para solução da “bifurcação” deste trabalho, foi baseado em simulações de imagens da aorta⁴.

```
createBranching: polygonObjectBase and: polygonObjectCollection and: aNameVrml
```

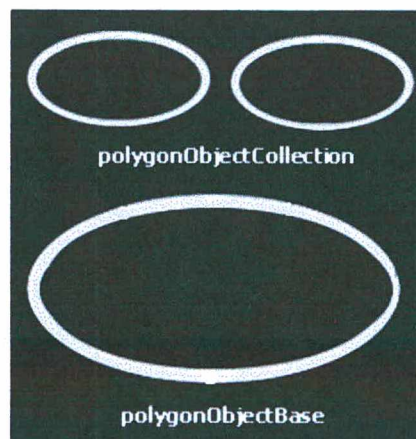


Figura 41: Ilustração do exemplo da Bifurcação

O primeiro passo foi criar um “ponto de sela” entre os dois cortes para que fosse feita a bifurcação.

```
middlePoint3D := Bifurca createMiddlePointNew: polygonObjectBase and:
polygonObjectCollection.
```

⁴ A Aorta é a maior artéria do corpo humano, e possui várias bifurcações.

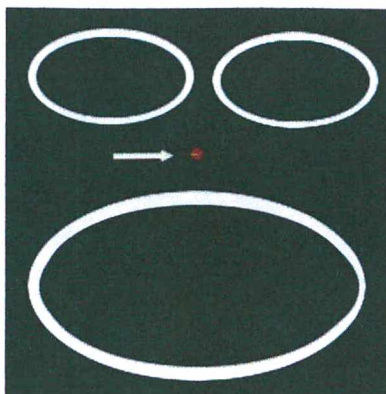


Figura 42 : Ilustração do Ponto de Sela

```
middlePoint := (((((polygonObjectCollection first findLimits at: 3) x) -
  ((polygonObjectCollection last findLimits at: 4) x) / 2) +
  (polygonObjectCollection last findLimits at: 4) x)) asFloat)@
```

Nesta demonstração do código o algoritmo encontra as extremidades do objeto localizado no corte (slice) superior (polygonObjectCollection), acima ilustrado na figura 41, neste caso o objeto em questão se trata da imagem direita. Este ponto será a primeira coordenada do middlePoint3D.

```
(((polygonObjectCollection first findLimits at: 3) y) -
  ((polygonObjectCollection last findLimits at: 4) y) / 2) +
  ((polygonObjectCollection last findLimits at: 4) y)) asFloat).
```

Em seguida, para encontrar a segunda coordenada do ponto é procurado as extremidades do objeto da esquerda.

Quanto a terceira coordenada do “ponto de sela” que se refere a altura em que o ponto deve se situar na imagem, o código abaixo mostra a técnica utilizada, onde é utilizado um método da classe *Point3D* para transformar um ponto 2D em 3D, passando as informações necessárias.

```
middlePoint3D := Point3D Point: middlePoint z: (((polygonObjectCollection
  last vertices first z) - (polygonObjectBase vertices first z) / 2)
  + (polygonObjectBase vertices first z) asFloat).
```

```
^middlePoint3D.
```

Neste caso, é passado ao método os valores das coordenadas z do objeto (*polygonObjectCollection*) e os valores das coordenadas z do objeto (*polygonObjectBase*). Retornando para o método de Bifurcação “Branching” o valor do ponto 3D.

Em seguida, para que seja feita a ligação dos pontos o código verifica as menores distâncias de um corte em relação ao outro. Por exemplo:

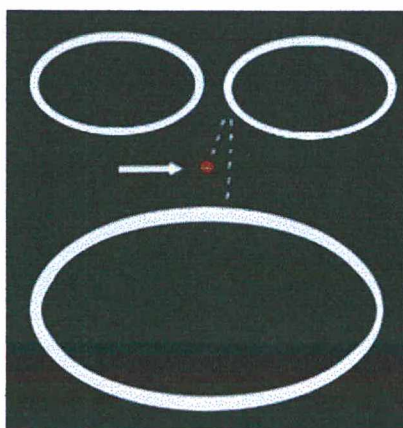


Figura 43 : Ilustração do Processo de Reconstrução da Bifurcação (1)

```
(polygonObjectCollection) vertices keysAndValuesDo: [:key :point1 |
    sorted1 := SortedCollection sortBlock: [: x :y | x first < y first ].
    sorted2 := OrderedCollection new.
```

Para que seja encontrada as menores distancias de um determinado ponto em relação a outro corte, primeiramente é percorrido a lista que contem os vértices do polígono, ou seja, os valores dos pontos que representam o objeto, e comparados com a lista que contem os vértices do outro objeto em questão.

```
(polygonObjectBase) vertices do: [: point2 | |temp |
    temp := OrderedCollection new.
    temp add: (point1 distance: point2);
```

```

    add: point2.
sorted1 add: temp.
1.

```

Após ter obtido as menores distâncias e armazená-las em uma lista, o algoritmo guarda os pontos restantes, ou seja, os pontos que não estiverem na lista de menores distancias serão armazenados em outra lista chamada *remainingPoints*.

Como esse exemplo trata o caso de um objeto ter o tamanho bem maior em relação aos outros do outro corte, é normal que sobre alguns pontos sem as respectivas distancias. Neste caso, a solução mais simples é fazer a conexão desses pontos com o “ponto se sela”, evitando “buracos” na textura do objeto a ser reconstruído.

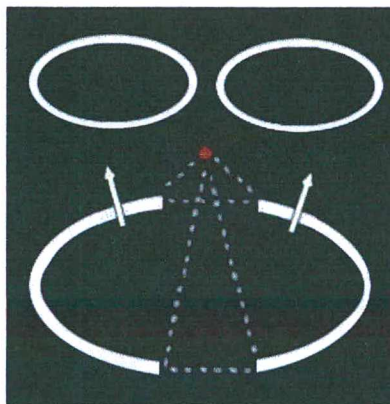


Figura 44 : Ilustração do Processo de Reconstrução da Bifurcação (2)

A partir do momento que as listas estiverem prontas o passo seguinte é a criação dos triângulos.

```

triangleList := OrderedCollection new.

```

Para tanto, é criada uma *orderedCollection* que armazenara o resultado dos índices dos pontos que formaram os triângulos.

```

triangleList addAll: (cyclopsWF triangulate: polygon1 and: polygon2
allPoints: pointList remainingPoints: allRemainingPoints cellPoint: aCellPoint).

```

Para fazer a triangularizacão dos pontos é criado um método na classe *CyclopsWireFrame*, responsável pelos métodos de triangularizacão já implementadas no Projeto Cyclops.

```
triangulateRemainingNearestConnection: triangleList allPoints: pointList
allPolygons: polygonList remainingPoints: allRemainingPoints cellPoint: CellPoint
```

Para este método é passado a lista de triângulos (*triangleList*), a lista de todos os pontos (*pointList*), a lista de polígonos (*polygonList*), os pontos restantes (*remainingPoints*) e o ponto de sela (*cellPoint*).

```
triangleList do: [ :aTriangle | | test |
    test := OrderedCollection with: (pointList at: (aTriangle point1)).
    (((pointList at: (aTriangle point2)) z) ~= (test last z)) ifTrue: [
        test add: (pointList at: (aTriangle point1)).
    ].
    (((pointList at: (aTriangle point3)) z) ~= (test first z)) &
    ((pointList at: (aTriangle point3)) z) ~= (test last z) ifTrue: [
        test add: (pointList at: (aTriangle point1)).
    ].
    (test size)=3 ifTrue: [
        extrem add: aTriangle.
    ].
].
```

Neste código é encontrada as extremidades que se ligaram ao “ponto de Sela”, e armazenadas em uma *orderedCollection* chamada *extrem*.

Em seguida, os pontos restantes (*allRemainingPoints*) são conectados ao “ponto de Sela” (*middlePoint3D*).

```
1 to: (allRemainingPoints size) do: [ :count | | tempPoint tempTri |
    middlePoint := (allRemainingPoints at: count) middlePoint:
(allRemainingPoints at: count).
    tempPoint := self connectPoints: middlePoint with: aPointCollection.
```

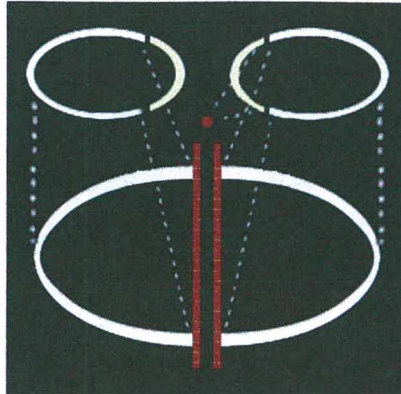



Figura 45 : Ilustração do Processo de
Reconstrução da Bifurcação (3)

A partir desta fase, a triangularização abaixo demonstrada no código é feita conforme a explicação do começo do capítulo.

```
tempTri := (Triangle3D
  point1: (pointList indexOf: (allRemainingPoints at: count))
  point2: (pointList indexOf: (allRemainingPoints at: count))+1
  point3: (pointList indexOf: tempPoint)).
```

A fim de evitar quebras na textura, a ligação das extremidades é refeita , podendo ser observado na figura 45.

```
adjacentes := OrderedCollection new.
extrem do: [ :aTriangle | | cellIndex |
  cellIndex := pointList indexOf: aCellPoint.
```

Na figura 46, é mostrado o resultado final deste método, mostrando a visualização de dois ângulos diferentes.

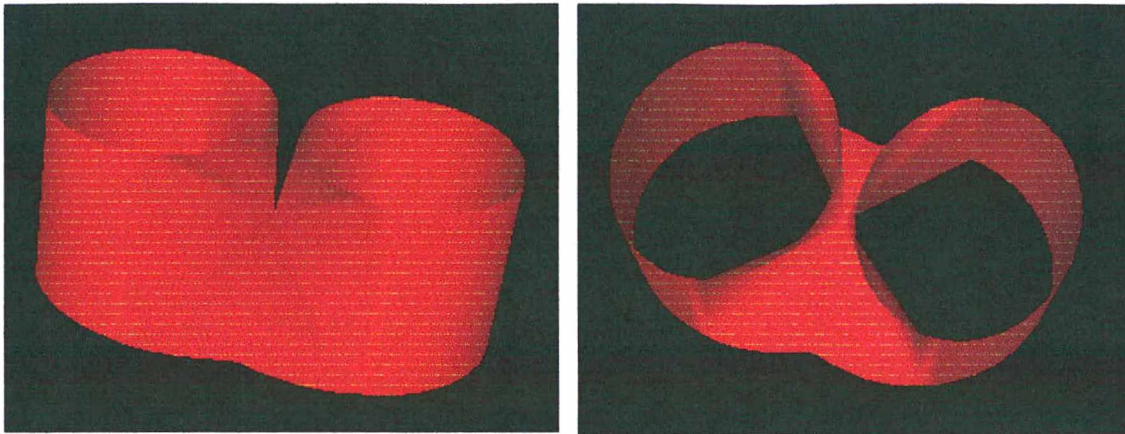


Figura 46 : Reconstrução Bifurcação

Após a fase de triangularização, é feita a exportação dos resultados destes métodos para a linguagem VRML, onde a mesma se encarrega de criar faces para representar o objeto.

6 TRABALHOS FUTUROS

Para que haja um aperfeiçoamento do trabalho aqui especificado, a próxima técnica de reconstrução tridimensional a ser incorporada nos testes será a de organizar o volume dos cortes (slices) das imagens médicas em uma hierarquia onde o elemento topo representaria o volume todo e os “filhos” deste elemento seriam referentes aos cortes da imagem.

Desta forma, a imagem seria dividida em quadrees⁵[18], onde poderíamos encontrar as bordas das imagens.

Neste caso, para que se possa extrair a borda da imagem, é necessário montar uma lista apenas com as “folhas brancas”. Esta colocação deve-se ao fato da estrutura de organização de quadrees, serem referidas como folhas. É importante ressaltar que estas imagens que serão submetidas a este processo possuem somente duas cores, preto e branco, podendo assumir assim que o fundo da imagem é preto, e as estruturas que serão extraídas a borda são brancas.

E para cada folha da lista criada, será seguida a seguinte especificação:

- ✓ Procurar as folhas vizinhas;
- ✓ Descobrir se em alguma direção a folha não tenha vizinhanças adjacentes na lista, para considera-la folha borda;

Se uma folha não tem nenhuma folha vizinha adjacente, e nenhuma folha vizinha em diagonal, provavelmente é um ruído e não deve ser desenhada.

Na figura abaixo as folhas pretas são a figura e as folhas brancas são o fundo. É possível observar nesta ilustração que as folhas pretas adjacentes à folha X estão marcadas com um ponto vermelho e a folha X tem vizinhanças adjacentes em todas as direções,

⁵ O termo Quadtree é usado para descrever a classe de estruturas de dados hierárquicas cuja propriedade comum é baseada no princípio da decomposição recursiva do espaço. Quadrees são usados para representação de imagens, codificando uma imagem como uma estrutura em forma de árvore.

exceto uma, já que a folha Y, é de outra cor, e portanto não consta na lista. A folha X é considerada borda então.

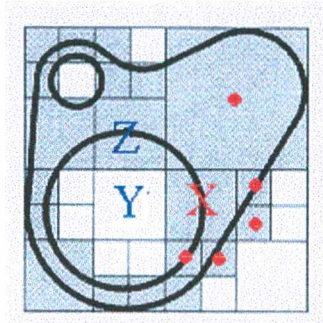


Figura 47 : Folhas Borda

Como estamos trabalhando com imagens onde o número de bits é sempre potência de dois, duas células adjacentes têm pontos em comum nas arestas, então, para descobrir se uma folha é adjacente à outra tem que se verificar se elas têm pontos em comum nas arestas.

Depois de encontrar as folhas bordas, o próximo passo é encontrar o ponto que fica no meio de cada um destes quadrantes, e então finalmente estes pontos serão utilizados para que possamos fazer a triangulação.

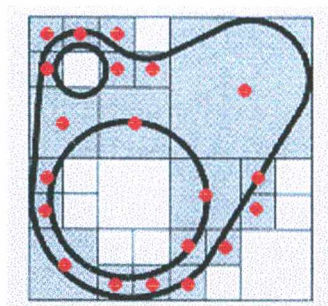


Figura 48 : Pontos centrais das folhas borda

Em seguida, o próximo passo será encontrar quais folhas são vizinhas no volume, e descobrir quais folhas de uma imagem são vizinhas de folhas nas outras imagens (slices) que formam o volume.

7 CONCLUSÃO

Neste trabalho foram discutidos vários métodos de processamento de imagens e reconstrução 3D de imagens médicas, onde foi possível apresentar técnicas e desenvolvê-las para proporcionar uma ferramenta de software que auxilie o diagnóstico médico.

Através dos testes executados podemos concluir que a ferramenta desenvolvida no projeto mostrou-se eficiente, com alto desempenho e funcionalidade. Porém, a necessidade no desenvolvimento de ferramentas que auxiliem equipes médicas nas diversas áreas, refletindo isto para os pacientes onde poderão ter seus exames de forma precisa e rápida, ocasionado melhores tratamentos, faz com que este trabalho continue evoluindo na busca por soluções cada vez mais diversificadas e eficientes nas aplicabilidades médicas.

A importância da pesquisa realizada é evidente, a introdução de técnicas de análise de imagens e reconstrução tridimensional de imagens médicas digitais, reunidas e disponibilizadas para fácil acesso médico, justificam o desenvolvimento da metodologia proposta no sistema *Cyclops*, contribuindo de forma significativa, melhorando aceitação do produto existente e multiplicando os benefícios alcançados pelo mesmo.

8 REFERÊNCIAS BIBLIOGRÁFICAS

- [01] Hopkins, Trevor; Horan, Bernard. **SMALLTALK : An Introduction to Application Development using VisualWorks**. Prentice Hall, 1995. 1ª Edition.
- [02] VisualWorks – ObjectShare – 1993 – 1998 – EUA.
- [03] Russ, John C. **The Image Processing HandBook**. Raleigh – North Carolina : CRC Press, 1995. 2ª Edition.
- [04] Gonzales, Rafael C.; Woods, Richard E. **Processamento de Imagens Digitais**. São Paulo – SP : Editora Edgard Blücher Ltda, 1992.
- [05] Marrin, Chris; Campbell, Bruce. **VRML 2 : in 21 days**. Indianópolis – Indiana : SAMS net, 1997. 1ª Edition.
- [06] Roehl, Bernie; Couch, Justin; Reed-Ballreich, Cindy; et al. **VRML 2.0 with Java**. Emeryville – Califórnia : ZIFF-Davis Press, 1997. 1ª Edition.
- [07] Haaga, John R.; Lanzieri, Charles F.; Sartoris, David J.; et al. **Tomografia Computadorizada e Ressonância Magnética do Corpo Humano**. Rio de Janeiro : 1996. 3ª Edição – Volume 1.
- [08] CYCLOPS, Documentação oficial. Disponível na Internet no endereço: <http://www.inf.ufsc.br/cyclops>.

- [09] Silva, Claudio A.; Seixas, Robreto de B.; **Detecção Automática de Contornos com Geração de Superfícies em Dados Volumétricos**: UERJ – Universidade do Estado do Rio de Janeiro.
- [10] Lamot, Marko; Zalik, Borut; **An Overview of Triangulation Algorithms for Simple Polygons**: Slovenia, 2000.
- [11] Peixoto, Adelailson; Gattass, Marcelo; **Reconstrução de Superfícies a partir de Seções Bidimensionais**: PUC-Rio, 2000.
- [12] Pao, Tsung; Piegl, Les A.; **Delaunay Triangulation in Three Dimensions**: University of South Florida, 1995.
- [13] Wangenheim, Aldo v.; **Introdução à Visão Computacional**: Universidade Federal de Santa Catarina - UFSC, 1999.
- [14] de Biasi, Herculano; **Desenvolvimento de uma Metodologia de Visão Computacional para o Auxílio no Planejamento Cirúrgico no Implante de Próteses Endoluminais**: Universidade Federal de Santa Catarina - UFSC, 2001.
- [15] Comunello, Eros; **Desenvolvimento de uma Metodologia Computacional para a Identificação, Contagem e Mensuração de Calcificações relacionadas com Neurocisticercose em Imagens de Tomografia Computadorizada**: Universidade Federal de Santa Catarina - UFSC, 1999.
- [16] Dellani, Paulo R.; **Desenvolvimento de um Servidor de Imagens Médicas Digitais no Padrão Dicom**: Universidade Federal de Santa Catarina - UFSC, 2001.

[17] Barros Jr, Euclides M.; **Sala de Laudo Virtual: “Um Ambiente de Teleradiologia para Diagnóstico Cooperativo via Internet**: Universidade Federal de Santa Catarina – UFSC, 2000.

[18] Miguel, Harley V; **Octrees e Qdtrees**. Universidade Federal de Santa Catarina – UFSC, 2001.

ANEXO 1

Modelagem da Implementação em UML



ANEXO 2

Manual do Usuário do 3DEditor

Neste apêndice é descrito como utilizar o sistema 3DEditor, criado para manipular imagens e disponibilizar reconstruções em 3D de estruturas anatômicas e/ou patologias.

Sua interface é mostrada na figura abaixo.

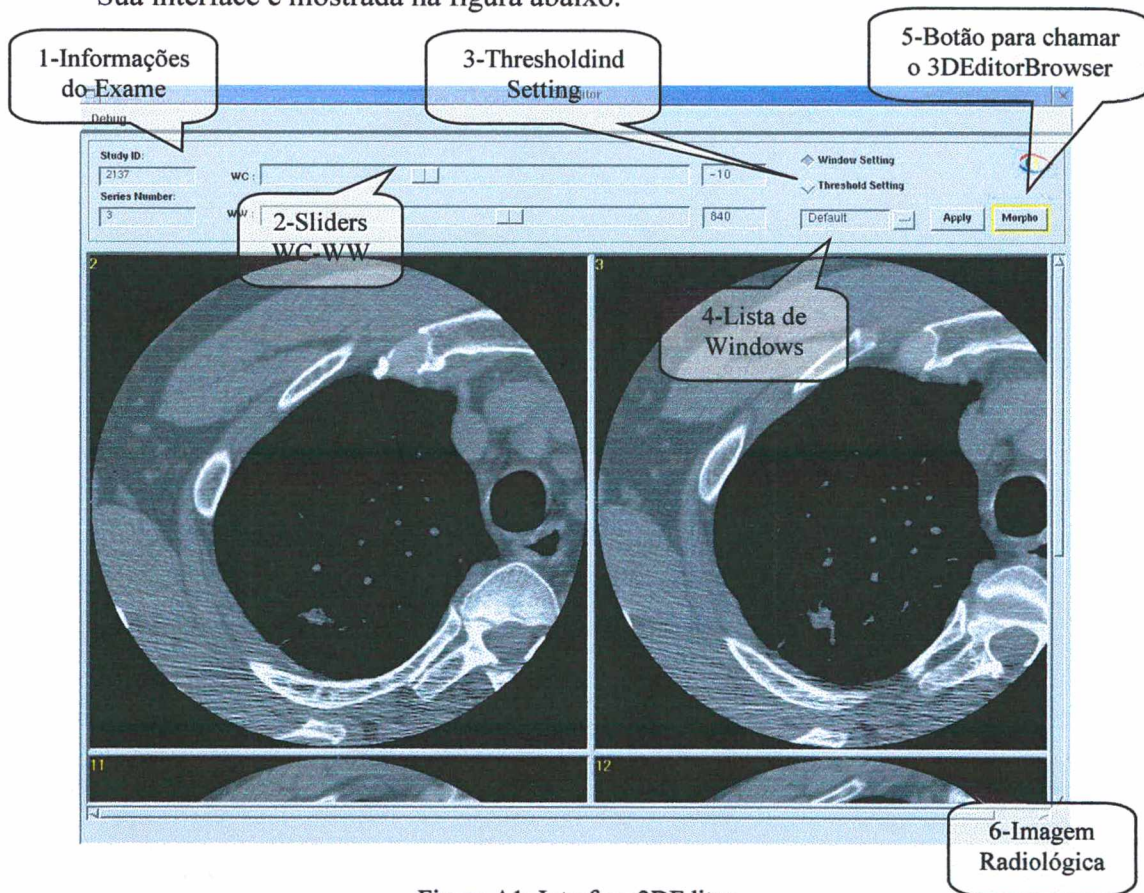


Figura A1: Interface 3DEditor

Descrição dos textos explicativos:

- 1- *Informações do Exame*: Mostra o número do estudo e o número de série do exame, estas informações são utilizadas pelas equipes radiológicas para a elaboração dos laudos sobre a imagem que esta sendo trabalhada.
- 2- *Sliders WC-WW*: Estes dois “sliders” são referentes aos valores de *WindowWidth*(Largura da Janela) e *WindowLevel*(Nível da Janela) da imagem, tais valores indicam a largura da paleta (quantidade de diferentes tons de cinza) e o seu centro na escala. Estes parâmetros não indicam valores na paleta e sim valores na escala Houndsfield. Esta escala define valores entre -1000 (corresponde ao preto) e +1000 (corresponde ao branco).
- 3- *Thresholding Setting*: Neste campo é possível selecionar a técnica baseada em segmentação por limiarização (Thresholding). A técnica de limiarização consiste em definir um valor limiar para a cor de um pixel. Pixels com valores de cor acima do limiar se tornam pretos e de valor abaixo se tornam brancos.
- 4- *Lista de Window*: Este campo possui uma lista de *windows*(valores de janelas pré-definidos para determinadas estruturas do corpo humano).
- 5- *Botão para chamar o 3DEditorBrowser*: Este botão chama outra interface do software para continuação do processo de reconstrução.
- 6- *Imagem Radiológica*: Espaço onde se localiza a imagem radiológica no sistema para manipulação.

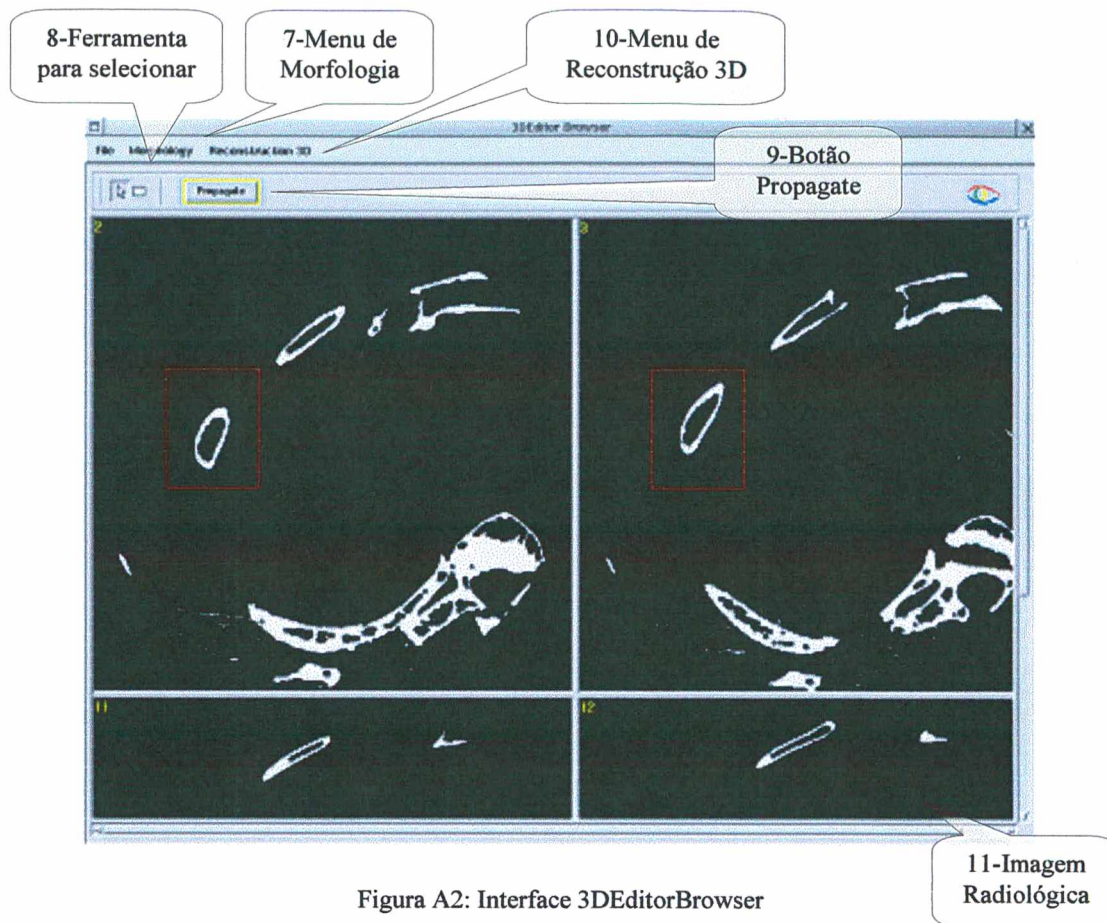


Figura A2: Interface 3DEditorBrowser

- 7- *Menu de Morfologia*: Menu para escolha de aplicação dos métodos de Opening e Closing.
- 8- *Ferramenta para Selecionar*: Esta ferramenta é para selecionar estruturas que serão reconstruídas em 3D.
- 9- *Botão Propagate*: Este botão propaga para todos os cortes da imagem a seleção feita para reconstrução 3D.

- 10- *Menu de Reconstrução*: Neste menu é possível efetuar a reconstrução 3D das estruturas selecionadas, inclusive imagens com bifurcação e exportação na linguagem VRML.