

Visualização de Volumes Aplicada à Área Médica

ALEXANDRE XAVIER FALCÃO²
ROBERTO DE ALENCAR LOTUFO²
ROBERTO J. GONÇALVES¹

²UNICAMP—Universidade Estadual de Campinas
FEE—Faculdade de Engenharia Elétrica
DCA—Depto. de Eng. de Computação e Automação Ind.
Grupo de Computação de Imagens
Caixa Postal 6101
13081 - Campinas - SP
xavier@dca.fee.unicamp.br
lotufo@dca.fee.unicamp.br

¹University of Pennsylvania
Department of Radiology
Medical Image Processing Group
roberto@icarus.mipg.upenn.edu

Abstract. This work presents a summary of the basic techniques used for visualization of 3D biomedical data and shows the results of the implementation of voxel based techniques which extends the visualization tools available in the Khoros system.

1 Introdução

A visualização de volumes aplicada à área médica tem como principal objetivo facilitar o entendimento de estruturas complexas tridimensionais internas ao corpo humano. Pesquisadores ao longo das últimas duas décadas têm se empenhado em desenvolver técnicas com este propósito. Graças ao aumento da velocidade de processamento e capacidade de memória dos computadores, e ao surgimento de novas modalidades de aquisição de dados 3D; as tomografias por raios-x (CT), por emissão de pósitron (PET) e por emissão de fóton (SPECT), e o método de aquisição de imagens por ressonância magnética (MRI); um rápido crescimento no desenvolvimento de *hardwares* e *softwares* para implementação destas técnicas têm sido possível. Além da visualização, a manipulação e a análise de dados 3D também são ferramentas importantes em aplicações clínicas, fornecendo informações quantitativas, por exemplo, medidas de distância, área e volume.

Os dados volumétricos são normalmente adquiridos em forma de imagens de fatias paralelas e uniformemente espaçadas representando cortes transversais ao eixo longitudinal do paciente. O empilhamento destas imagens, mantendo o espaçamento original entre elas, pode ser idealizado de forma que cada pixel represente o volume de um pequeno cuboide (paralelepípedo) no espaço, denominado voxel

(pixel 3D). Cada voxel tem normalmente associado um número inteiro (densidade do voxel) proporcional ao tom de cinza do pixel na imagem correspondente. Cada um desses valores representa a integração de uma propriedade física que está sendo mensurada no interior do volume associado ao voxel. Este conjunto de voxels representando os dados 3D é conhecido por espaço cena, espaço voxel, ou simplesmente cena 3D (figura 1). A cena 3D não apresenta nenhuma manifestação visível e, portanto, para visualizar objetos de interesse (órgãos ou parte de órgãos) na cena é necessária a criação de uma representação intermediária. Os principais métodos de visualização podem ser classificados de acordo com o tipo de representação intermediária (pontos, linhas, superfícies, cubos, gels, etc.) [9]. Estes métodos são divididos em *surface-based techniques* (técnicas baseadas em superfície), *binary voxel techniques* (técnicas baseadas em voxel binário) e *semi-transparent volume rendering techniques* (técnicas de *rendering* de volume semitransparente). As técnicas baseadas em superfície procuram estimar o formato da superfície de interesse utilizando um conjunto de primitivas geométricas, que podem ser polígonos (normalmente triângulos) ou superfícies curvas. As técnicas baseadas em voxel binário representam o objeto por um conjunto de cubos opacos (voxels-1), ou sua superfície pelos polígonos que definem estes cubos. As

técnicas de *rendering* de volume semitransparente associam uma cor e uma opacidade parcial a cada voxel da cena 3D dando à imagem final a aparência de um gel colorido e semitransparente.

Infelizmente nenhuma das técnicas desenvolvidas até hoje apresenta um resultado completamente satisfatório [4]. As técnicas baseadas em superfície levam a vantagem que as primitivas geométricas são compactas (tornando econômico o armazenamento e a transmissão dos dados) e possuem um alto grau de coerência espacial (tornando eficiente as operações de *rendering*). Entretanto, estas técnicas exigem uma classificação binária da cena 3D; a superfície passa ou não passa através do voxel em questão; podendo gerar superfícies inexistentes no objeto real (falsas positivas) e/ou esconder superfícies reais (falsas negativas) na imagem final. Além disto, a construção automática da superfície raramente tem sucesso, necessitando a interferência do usuário. As técnicas baseadas em voxel binário são relativamente fáceis de serem implementadas e as estruturas de dados utilizadas são simples de serem manipuladas, porém padecem dos mesmos problemas causados pela necessidade de classificação binária. As técnicas de *rendering* de volume semitransparente oferecem uma importante vantagem sobre as técnicas baseadas em superfície e em voxel binário substituindo a classificação binária por uma classificação *fuzzy* [12]. Entretanto, o ajuste de parâmetros durante o processo para obter uma boa imagem é não trivial. Além disto, estas técnicas normalmente necessitam ter toda a massa de dados na memória e o tempo para *rendering* cresce proporcionalmente ao número de voxels, o que faz com que sejam consideradas técnicas computacionalmente caras (imagens típicas usando estações de trabalho necessitam de minutos e as vezes horas para serem geradas).

Os artigos em geral costumam dar uma visão superficial do assunto e abordar soluções de problemas específicos. Devido a área de visualização de volumes ser relativamente nova, principalmente no Brasil, é importante um trabalho que forneça uma visão mais detalhada sobre este assunto. O objetivo deste trabalho é, portanto, apresentar um resumo das operações básicas necessárias no processo de visualização incluindo as equações utilizadas. O artigo também apresenta os resultados de extensão do ambiente Khoros [6] de visualização de imagens para implementar algumas técnicas baseadas em voxel. O Khoros é um pacote aberto de visualização de dados e processamento de sinais desenvolvido na Universidade do Novo México, EUA. Este programa possui como parte central um ambiente de programação visual que permite a interconexão de módulos de pro-

cessamento e de controle. Assim é possível construir estruturas complexas de processamento a partir de uma coleção de blocos interligados.

2 Terminologia

São definidos três espaços de trabalho nos quais as técnicas de visualização volumétrica são implementadas.

2.1 Espaço Voxel

Os dados gerados por um tomógrafo geralmente consistem de uma sequência de fatias 2D espaçadas de uma distância fixa "d". Considerando as fatias paralelas ao plano xy representando imagens de cortes transversais ao longo do eixo z, o espaço voxel pode ser idealizado como o cubóide que envolve toda a região rastreada pelo tomógrafo (figura 1). O cubóide é subdividido em pequenos outros cubóides denominados voxels. Cada voxel representa um volume $p \times p \times d$ formado ao se tomar uma distância $d/2$ em torno do pixel correspondente nas direções dos semi-eixos z+ e z-.

2.2 Espaço Objeto

O espaço objeto é definido como um espaço voxel binário, onde a cada voxel é associado um valor 1, quando o voxel pertence ao objeto, ou um valor 0 no caso contrário.

2.3 Espaço Imagem

O espaço imagem é representado por uma área retangular, no plano uv (plano de visão), na qual o objeto é projetado (figura 2). Esta área é subdividida em pequenos quadrados $p \times p$ denominados pixels. Exemplos típicos de dados neste espaço associam ao valor dos pixels a distância do plano de visão à superfície do objeto (d-buffer), ou a intensidade de luz da imagem final.

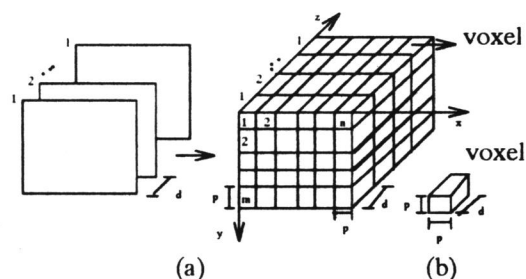


Figura 1: Espaço Voxel

3 Operações Básicas

As operações básicas necessárias no processo de visualização são técnicas de processamento de imagens e computação gráfica. Apesar de serem apresentadas seguindo uma ordem de processamento, é possível obter diferentes resultados favoráveis apenas trocando a ordem e/ou omitindo algumas operações abaixo. Maiores detalhes sobre estas operações podem ser vistos em Udupa [7]. É evidente que outras operações como *antialiasing*, por exemplo, podem ser acrescentadas ao processo de visualização conforme a necessidade e imaginação do programador.

3.1 Volume de Interesse (VOI)

O volume de interesse é formado pelo menor paralelepípedo que contém o objeto de interesse na cena 3D. Esta operação gera uma nova cena 3D com um subconjunto de voxels da cena original. A principal proposta da operação VOI é minimizar o espaço de memória requerido para o processamento posterior da cena. Uma outra razão para esta operação pode ser excluir parte de um órgão para que a cena 3D resultante revele o interior deste órgão.

3.2 Filtragem

A proposta mais ampla das operações de filtragem é a suavização, realce e preservação de partes selecionadas na cena 3D. A filtragem gera uma nova cena apenas alterando (ou não) a densidade de cada voxel em função da densidade de voxels numa vizinhança escolhida.

3.3 Interpolação

O espaçamento entre as fatias normalmente é diferente da distância entre os pixels nas imagens podendo causar distorções ao visualizar o objeto sob diferentes ângulos. Portanto, a interpolação visa gerar uma nova cena 3D cujos os voxels são cubos e não cubóides. Novas fatias podem ser interpoladas e a resolução das fatias também pode ser alterada interpolando novos pixels. O tipo mais simples, a interpolação linear, admite uma variação linear da densidade dos voxels na direção z :

$$f(x, y, z_i) = [f(x, y, z+1) - f(x, y, z)][z_i - z] + f(x, y, z)$$

onde:

$f(x, y, z)$ é a densidade do voxel (x, y, z) e $z \leq z_i \leq z+1$.

3.4 Segmentação

A segmentação (classificação binária) consiste em identificar o objeto de interesse na cena gerando uma

cena 3D binária cujos os voxels-1 pertencem ao objeto e os voxels-0 não pertencem. Esta operação pode ser vista como uma transformação do espaço voxel no espaço objeto. A forma mais simples de segmentação é a limiarização. Na limiarização define-se dois valores limiar de densidade, o inferior e o superior, entre os quais o voxel é considerado pertencer ao objeto de interesse.

3.5 Mascaramento

A operação de mascaramento pode ser utilizada quando não é possível fazer a segmentação automática da cena com sucesso. Desenhando-se uma curva fechada em torno do objeto de interesse nas fatias que provocam erros na segmentação é possível especificar uma subregião para cada fatia onde a segmentação automática funcione. Uma máscara, representada por uma imagem binária gerada a partir da curva traçada associando valor 1 para os pixels no interior da curva e 0 para os pixels no exterior, pode ser utilizada para gerar uma nova imagem da fatia que mantém o valor de todos os pixels da fatia original que têm densidade 1 na máscara e atribui valor 0 aos demais.

3.6 Modelagem

A modelagem pode ser aplicada a uma cena 3D binária ou em tons de cinza, envolvendo neste caso uma segmentação implícita, com o objetivo de gerar uma representação geométrica do objeto de interesse. A representação geométrica mais simples é a própria cena 3D binária que pode ser vista como um modelo do volume do objeto (modelo de volume binário). O modelo de volume pode ser armazenado usando um outro tipo de estrutura de dados que facilite o processamento posterior como, por exemplo, a estrutura *octree*. A representação geométrica também pode ser um modelo de superfície visto como uma casca oca. Um exemplo simples deste modelo é uma estrutura de dados que contém apenas a informação do centro geométrico das faces dos voxels-1 da superfície do objeto, quando a cena 3D é vista das seis direções principais [7]. Um modelo de superfície mais complexo, por exemplo, extrai o contorno do objeto em cada fatia e depois une os contornos empilhados formando uma malha de triângulos [5] [1]. Os dois primeiros modelos podem ser incluídos nas técnicas baseadas em voxel binário e este último nas técnicas baseadas em superfície.

3.7 Transformações Geométricas

Esta etapa exige a definição e posicionamento relativo entre os elementos do cenário. Uma simpli-

figuração é normalmente adotada considerando o observador e uma única fonte de luz na mesma posição. A figura 2 ilustra a situação inicialmente adotada, neste trabalho, onde o observador e a fonte de luz estão situados no semi-eixo $y-$, o plano de visão uv tem sua origem no centro geométrico do espaço imagem, sobre o semi-eixo $y-$ entre o observador e o volume de dados. O sistema de coordenadas (x, y, z) , que segue a regra da mão direita, tem sua origem no centro geométrico do volume de dados. Dois graus de liberdade são definidos pelos ângulos α e β para que o observador, fonte de luz e plano de visão possam girar em torno do volume de dados para visualização de qualquer de suas faces (uma outra opção pode ser girar apenas o volume de dados). Os ângulos de rotação α e β equivalem, respectivamente, aos movimentos de rotação (spin) e inclinação (tilt), do objeto em torno de seus eixos longitudinal (eixo z) e lateral (eixo x). Uma translação do plano de visão $\vec{T} = (T_x, T_y, T_z)$ também é possível para efetuar cortes no volume de dados. A matriz de rotação e translação utilizada é dada por

$$M = \begin{bmatrix} \cos\alpha & -\sin\alpha\cos\beta & \sin\alpha\sin\beta & 0 \\ \sin\alpha & \cos\alpha\cos\beta & -\cos\alpha\sin\beta & 0 \\ 0 & \sin\beta & \cos\beta & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}$$

As dimensões do espaço imagem Δu e Δv são calculadas em função dos ângulos de rotação e das dimensões do volume de dados Δx , Δy e Δz . A posição inicial da origem do plano de visão é $(0, y_0, 0)$ tal que para quaisquer ângulos de rotação não haja cortes desproporcionais no volume de dados.

3.8 Rendering de Volume Binário

Definido o posicionamento relativo entre o observador, fonte de luz e volume de dados, a formação da imagem requer duas etapas: projeção com remoção de superfícies escondidas e tonalização.

3.8.1 Projeção

A projeção procura identificar no volume de dados binário os voxels-1 pertencentes à superfície do objeto de interesse que estão sendo visualizados pelo observador. O método de projeção utilizado é conhecido por *ray casting* (lançamento de raios de procura) com projeção ortogonal¹ sobre o plano de visão. Neste método, para cada pixel do espaço imagem é lançado um raio em direção ao volume de dados

¹Um outro método, conhecido por *voxel projection* [7], rastreia o volume de dados projetando os voxels no plano de visão. Neste caso um algoritmo de remoção de superfícies escondidas é necessário [10]

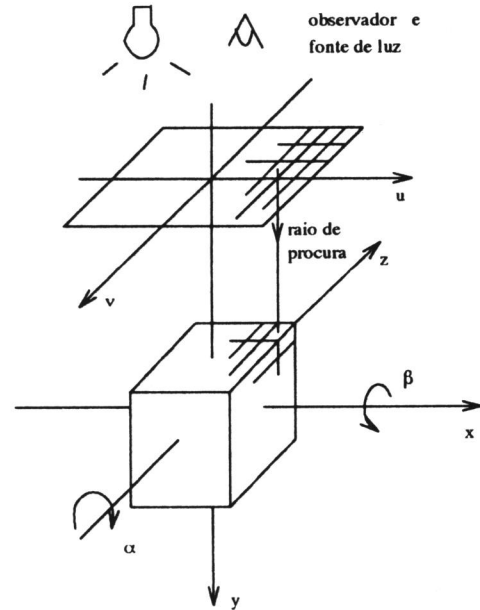


Figura 2: Sistema de coordenadas para projeção

(figura 2). Os raios são paralelos entre si e perpendiculares ao espaço imagem. Quando um raio penetra no volume de dados, ele caminha até encontrar o primeiro voxel pertencente ao objeto de interesse. A projeção do objeto de interesse sobre o espaço imagem é efetuada associando a cada pixel, sua distância ao primeiro voxel-1 encontrado pelo raio (formação do d-buffer)

A equação paramétrica inicial de um raio lançado do pixel $(u = x_0, y_0, v = z_0)$ na direção $\vec{D} = (0, 1, 0)$ é:

$$\begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix} = \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix} + t \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

sobre esta equação é aplicada a transformação M em coordenadas homogêneas:

$$\begin{bmatrix} x(t)' & y(t)' & z(t)' & 1 \end{bmatrix} = M \begin{bmatrix} x(t) & y(t) & z(t) & 1 \end{bmatrix}$$

com efeito chega-se a equação recursiva utilizada pelo algoritmo para caminhar sobre o raio de procura:

$$X(t_i) =$$

$$X(t_{i-1}) + \Delta t \begin{bmatrix} \sin\alpha & \cos\alpha\cos\beta & -\cos\alpha\sin\beta & 1 \end{bmatrix}$$

onde:

$$X(t_i) = \begin{bmatrix} x(t_i) & y(t_i) & z(t_i) & 1 \end{bmatrix}$$

t_i é o valor do parâmetro t no instante i ($t_0 = 0$)

$$\Delta t = t_i - t_{i-1}$$

3.8.2 Tonalização

A operação de tonalização visa a formação da imagem final através da distribuição de luminosidade sobre o plano de visão. A imagem é gerada associando

a cada pixel da grade um tom de cinza. O valor desse tom de cinza deve ser tal que simule a intensidade de luz na superfície do objeto em questão. A forma mais simples é a tonalização por distância (*depth shading*), onde os tons de cinza $I_{dist}(u, v)$ associados aos pixels (u, v) variam linearmente com as distâncias da superfície ao plano de visão:

$$I_{dist}(u, v) = \frac{I_{max} - I_{min}}{d_{min} - d_{max}} [d(u, v) - d_{max}] + I_{min}$$

onde:

I_{max} é a máxima intensidade desejada na imagem final

I_{min} é a mínima intensidade desejada na imagem final

d_{max} é a maior distância da superfície do objeto ao plano de visão

d_{min} é a menor distância da superfície do objeto ao plano de visão

$d(u, v)$ é a distância do pixel (u, v) à superfície do objeto.

Essa forma de tonalização dá pouca informação dos detalhes da superfície e, consequentemente, os defeitos da mesma também não aparecem. Uma ilustração deste tipo de tonalização pode ser vista na figura 3.8.2.

Uma forma mais complexa, que permite mostrar detalhes da superfície do objeto baseia-se na equação de Phong [2] modificada:

$$I(u, v) = I_{max} K_a + I_{dist}(u, v) (K_d \cos \theta + K_s \cos^n 2\theta)$$

onde:

$I(u, v)$ é o nível de cinza associado ao pixel (u, v)

K_a é o coeficiente de reflexão para a luz ambiente

K_d é o coeficiente de reflexão difusa

K_s é o coeficiente de reflexão especular

θ é o menor ângulo entre o vetor normal à superfície e o raio de incidência.

Esta equação necessita da estimativa do vetor normal em cada ponto visível da superfície. Esta estimativa pode ser feita de várias formas diferentes alterando a tonalização da imagem resultante e sugerindo uma classificação para os tipos de tonalização: tonalização no espaço voxel, no espaço objeto e no espaço imagem. A tonalização no espaço voxel utiliza a cena 3D em tons de cinza para estimar o vetor normal (figura 3.8.2), a tonalização no espaço objeto estima a normal usando a cena 3D binária [3] (figura 3.8.2), e a tonalização no espaço imagem usa o *d-zbuffer* para estimar o vetor normal (figura 3.8.2).

A estimativa do vetor normal $\vec{N} = (N_x, N_y, N_z)$ é considerada uma aproximação do cálculo do vetor



Figura 3: Região ocular - tonalização por distância



Figura 4: Região ocular - tonalização no espaço voxel

gradiente. No espaço voxel [7] e espaço objeto [3] foi utilizada a seguinte aproximação:

$$\begin{aligned} N_x &= \frac{f(x+1, y, z) - f(x-1, y, z)}{2} \\ N_y &= \frac{f(x, y+1, z) - f(x, y-1, z)}{2} \\ N_z &= \frac{f(x, y, z+1) - f(x, y, z-1)}{2} \end{aligned}$$

Sendo que no espaço voxel, $f(x, y, z)$ é o valor do voxel em nível de cinza e no espaço objeto, o valor do voxel assume valores 0 (caso não pertença ao objeto) e 1 (caso contrário).

O cálculo utilizado no espaço imagem [7] é dado por:

$$\begin{aligned} \vec{N} &= (N_u, 1, N_v) \\ N_u &= \frac{d(u+1, v) - d(u-1, v)}{2} \\ N_v &= \frac{d(u, v+1) - d(u, v-1)}{2} \end{aligned}$$

A utilização do vetor normal representa uma tentativa de estimar a forma que a superfície de interesse corta os voxels-1 para recuperar os pequenos e mal definidos aspectos da estrutura real perdidos com a classificação binária. Dependendo do tipo de tecido de interesse e do método de aquisição de dados

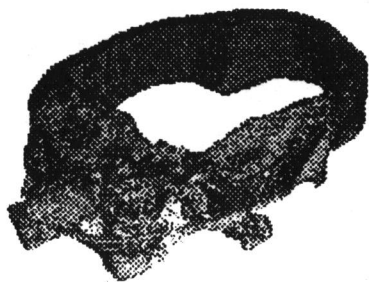


Figura 5: Região ocular - tonalização no espaço objeto

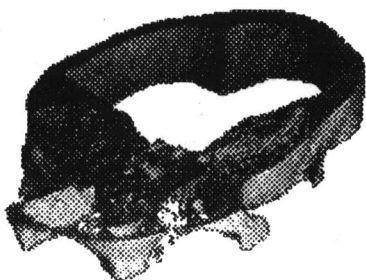


Figura 6: Região ocular - tonalização no espaço imagem

utilizado, o vetor normal pode afetar consideravelmente os resultados visuais. Portanto, a visualização de volumes e' mais utilizada como um suporte para a manipulação e análise de parâmetros do que como uma ferramenta suficiente em diagnósticos. Em imagens de CT, por exemplo, normalmente a tonalização no espaço voxel apresenta melhores resultados, mas para tecidos menos densos como a pele, foi notado que a tonalização no espaço imagem e' mais adequada (figura 7) ².

3.9 Combinação de Imagens

As imagens geradas no *rendering* de volume binário, ou em qualquer outro método, podem ser combinadas usando o conceito de opacidade [7]. A opacidade é representada por um valor real entre 0% e 100% proporcional a evidência desejada na imagem final, para o tecido ao qual está sendo associada. Um resultado desta técnica é a combinação de duas imagens tonalizadas no espaço imagem, uma com pele (face) a 70% e outra com osso (região ocular) a 30%

²É interessante observar que os riscos horizontais nesta imagem são provocados por movimento do paciente no momento da aquisição dos dados



Figura 7: Face - tonalização no espaço imagem

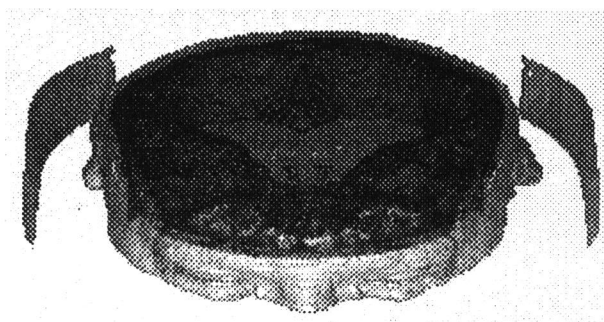


Figura 8: Combinação pele 70% e osso 30%

(figura 8).

$$I(u, v) = \frac{\sum_{i=0}^n \alpha_i I_i(u, v)}{\sum_{i=0}^n \alpha_i}$$

onde:

$I(u, v)$ é o nível de cinza associado ao pixel (u, v)

n é o número de imagens

α_i é a opacidade associada a i -ésima imagem

$I_i(u, v)$ é o nível de cinza associado ao pixel (u, v) da i -ésima imagem.

3.10 Classificação

A classificação é uma operação de pré-processamento para o *rendering* de volume semitransparente aplicada apenas à cena 3D em tons de cinza caracterizando uma forma diferente de segmentação (segmentação *fuzzy*) [12] onde não há objetos de interesse, mas regiões de tecidos (ou nuvens) de interesse. Na classificação a informação de densidade dos voxels é utilizada na escolha de um critério de associação de cor e opacidade a todos os voxels da cena, formando uma representação hipotética de uma cena colorida e semitransparente. Esta operação é utilizada apenas nas técnicas de *rendering* de volume semitransparente e representa uma transformação do espaço voxel no espaço voxel.

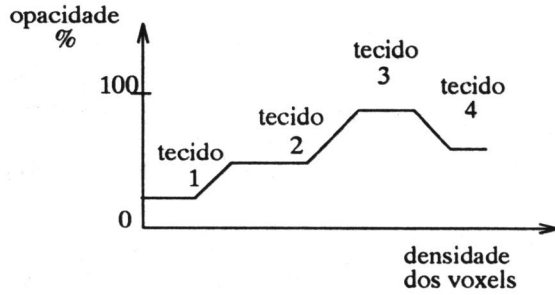


Figura 9: Distribuição linear de opacidade

Uma forma genérica de classificação é supor uma variação linear de opacidade com relação a intervalos disjuntos de densidade de voxel para identificar os tipos de tecidos contidos em cada voxel (figura 9). A um voxel associa-se o valor de opacidade referente ao intervalo que contém a densidade do voxel. O mesmo procedimento pode ser feito para associar cor ao voxel. A cor é normalmente expressa em função das primárias vermelho, verde e azul $C=(R,G,B)$.

$$\alpha_{v,linear} = \frac{(\alpha_{n+1} - \alpha_n)}{f_{n+1} - f_n}(f_v - f_n) + \alpha_n$$

$$C_v = \frac{(C_{n+1} - C_n)}{f_{n+1} - f_n}(f_v - f_n) + C_n$$

onde:

$\alpha_{v,linear}$ é a opacidade associada ao voxel $v = (x, y, z)$

$C_v = (R_v, G_v, B_v)$ é a cor associada ao voxel v

f_v é a densidade do voxel v

f_n é a densidade no início do intervalo n

α_n é a opacidade associada à densidade f_n

$C_n = (R_n, G_n, B_n)$ é a cor associada à densidade f_n .

O vetor gradiente calculado no espaço voxel (ver 3.8.2) representa as variações de densidade neste espaço. Nas regiões de interface entre tecidos ele costuma possuir módulo maior que nas regiões internas aos tecidos. Portanto, um método mais complexo utiliza esta propriedade para dar maior ênfase (maior opacidade) às regiões de interface e menor ênfase às regiões internas [8](figura 10).

$$\alpha_v = |\vec{g}_v| \alpha_{v,linear}$$

$$C_v = C_f \left(K_a + \frac{1}{K_1 d_v + K_2} (K_d \cos \theta + K_s (\cos 2\theta)^n) \right)$$

onde:

α_v é a opacidade associada ao voxel $v = (x, y, z)$

$C_v = (R_v, G_v, B_v)$ é a cor associada ao voxel v

$|\vec{g}_v|$ é o módulo do vetor gradiente calculado no voxel v

$C_f = (R_f, G_f, B_f)$ é a cor da fonte de luz

$K_a = (k_{ar}, k_{ag}, k_{ab})$ e' coeficiente de reflexão para a luz ambiente

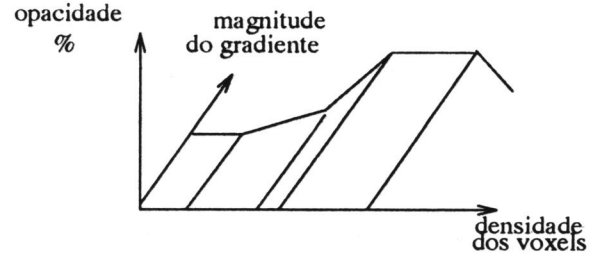


Figura 10: Distribuição linear de opacidade usando a magnitude do vetor gradiente

$K_d = (k_{dr}, k_{dg}, k_{db})$ é o coeficiente de reflexão difusa
 $K_s = (k_{sr}, k_{sg}, k_{sb})$ é o coeficiente de reflexão especular

θ é o menor ângulo entre o vetor normal à superfície e o raio de incidência

K_1 e K_2 são constantes de linearização

$d_v = d(u, v)$ é o valor do *d-buffer* para o voxel v .

3.11 Rendering de Volume Semitransparente

Uma característica singular nas técnicas de *rendering* de volume semitransparente é que a cor associada a cada pixel na imagem final é o resultado da combinação da cor da luz refletida e sucessivamente transmitida através de cada voxel atravessado pelo o raio de visão na cena 3D. Esta técnica é conhecida como *volumetric compositing* (composição volumétrica) [10]. Neste caso o algoritmo de lançamento de raios é conhecido por *ray tracing*. A composição volumétrica pode ser representada pela seguinte equação recursiva.

$$C_{out}\alpha_{out} = C_{in}\alpha_{in} + C_n\alpha_n(1 - \alpha_{in})$$

$$\alpha_{out} = \alpha_{in} + \alpha_n(1 - \alpha_{in})$$

onde para o n -ésimo voxel atravessado pelo raio:

$C = C_{out}\alpha_{out}$ e' a cor associada ao pixel (u,v) no n -ésimo estágio do algoritmo

$C_{in}\alpha_{in}$ é a cor no $(n-1)$ -ésimo estágio do algoritmo

C_n e α_n são respectivamente cor e opacidade associadas ao n -ésimo voxel

α_{in} é a opacidade acumulada no $(n-1)$ -ésimo estágio

α_{out} é a opacidade acumulada no n -ésimo estágio.

Imagens típicas resultantes da aplicação desta técnica usando os dois métodos de classificação apresentados na subseção 3.10 podem ser vistas nas figuras 9 e 10.

3.12 Técnicas de Reprojeção

Uma outra forma de exploração dos dados 3D é a simples reprojeção, com pouco ou nenhum artifício, da densidade dos voxels sob diferentes pontos de vista.

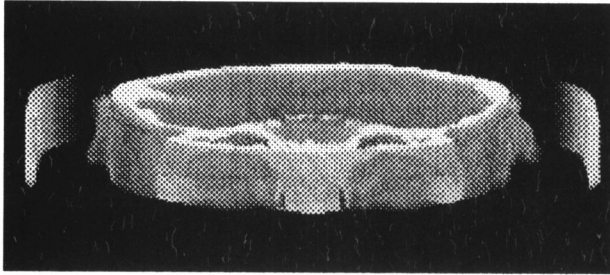


Figura 11: Rendering de Volume semitransparente usando a distribuição linear de opacidade apenas

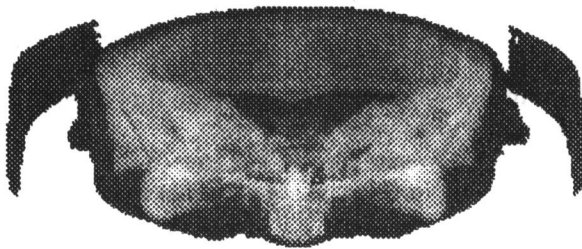


Figura 12: Rendering de Volume semitransparente usando a distribuição linear de opacidade e a magnitude do vetor gradiente

Diferente das técnicas anteriores, estas técnicas apenas combinam ou selecionam densidades de voxels sobre um mesmo raio de procura para calcular o nível de cinza do respectivo pixel na imagem final.

3.12.1 Reprojeção Aditiva

A reprojeção aditiva associa a cada pixel (u,v) do plano de visão uma média aritmética das densidades dos voxels visitados pelo raio de procura. Em imagens de CT o resultado desta técnica é uma espécie de radiografia 3D (figura 13).

$$I(u, v) = \frac{\sum_{v=0}^n f_v}{n}$$

onde:

$I(u, v)$ é o nível de cinza associado ao pixel (u, v)
 f_v é a densidade do n -ésimo voxel atravessado pelo raio
 n é o número de voxels atravessados pelo raio.

3.12.2 Reprojeção Radiográfica

No caso de existir um maior interesse em algum tipo de tecido na cena, a reprojeção aditiva leva a desvantagem nos casos em que a maior parte dos voxels pertencem a outros tecidos. Um exemplo é quando desejamos visualizar as partes pertencentes a osso



Figura 13: Vista superior da região ocular - reprojeção aditiva

nos dados 3D. Dependendo do ponto de vista, voxels pertencentes a tecidos de baixa densidade podem atenuar bastante o nível de cinza da imagem final dificultando a visualização do tecido ósseo. Uma forma de evitar este problema é associar a cada pixel (u,v) do plano de visão uma média ponderada das densidades dos voxels visitados pelo raio, onde voxels com maior nível de cinza são considerados mais opacos. Os pesos podem ser escolhidos utilizando uma distribuição de opacidades como a da figura 9. Um resultado desta técnica pode ser visto na figura 14.

$$I(u, v) = \frac{\sum_{v=0}^n \alpha_v f_v}{\sum_{v=0}^n \alpha_v}$$

onde:

$I(u, v)$ é o nível de cinza associado ao pixel (u, v)
 f_v é a densidade do n -ésimo voxel atravessado pelo raio
 α_v é a opacidade do n -ésimo voxel atravessado pelo raio
 n é o número de voxels atravessados pelo raio.

3.12.3 Máxima Intensidade de Projeção

A máxima intensidade de projeção tem como objetivo identificar as regiões de maior densidade do conjunto de dados 3D. Esta técnica associa a cada pixel (u,v) do plano de visão o nível de cinza correspondente a maior densidade de voxel encontrada pelo respectivo raio de procura. Um resultado desta técnica pode ser visto na figura 15, onde regiões mais claras representam maior densidade.

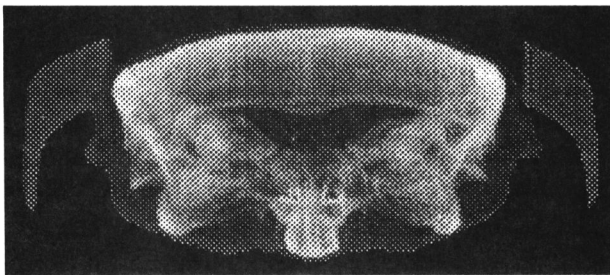


Figura 14: Reprojeção radiográfica

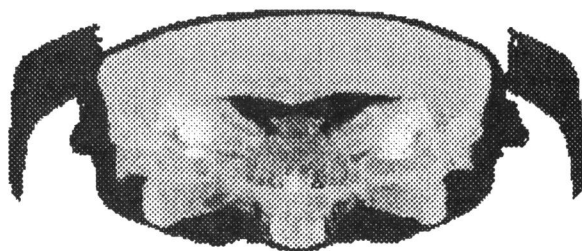


Figura 15: Projeção dos voxels de máxima intensidade

4 Implementação

Os dados 3D foram obtidos de um tomógrafo GE9800. No Khoros foram desenvolvidos cinco programas:

soft2viff, **vinter**, **dbuffer**, **shading** e **raytrace**. O programa **soft2viff** converte os dados 3D para o formato VIFF do Khoros. O programa **vinter** possui várias opções de operações de interpolação e a operação de segmentação por limiar. Este programa foi utilizado para fazer uma interpolação linear na direção perpendicular ao plano das fatias. Os programas **dbuffer** e **shading** implementam o *rendering* de volume binário. O programa **dbuffer** carrega na memória principal todo o volume de dados e executa o método de projeção *ray casting* gerando um novo arquivo com o *d-buffer* e, quando desejado, com os ângulos θ entre os vetores normais e o raio de incidência. Neste programa também está embutida uma operação de segmentação por limiar. O programa **shading** gera o arquivo final com a imagem tonalizada. O programa **raytrace** gera diretamente a imagem final, e implementa as técnicas de reprojeção e de *rendering* de volume semitransparente.

Os parâmetros de entrada e saída de cada programa são atualizados interativamente no ambiente Khoros. Outros recursos do Khoros podem ser acrescentados, como por exemplo, operações de filtragem nas normais ou na imagem tonalizada. Uma in-

teração direta entre o usuário e os arquivos intermediários também é possível.

5 Conclusões

Este trabalho mostra uma visão geral das principais técnicas de visualização de volumes baseadas em voxel. Além disso descreve as rotinas de visualização de volumes que foram incorporadas ao pacote Khoros.

A experiência quanto ao uso do Khoros confirmou como sendo um ambiente apropriado para a pesquisa e ensino. Sua flexibilidade permite explorar diversas combinações de operações primitivas e escolher a melhor técnica de visualização para cada tipo de dado.

Uma sugestão com relação a ordem das operações iniciais pode ser dada para melhorar a qualidade das imagens resultantes: filtragem mediana sobre os dados 3D originais, segmentação por limiar e interpolação combinada [11].

6 Agradecimentos

Os dados referentes à região ocular foram obtidos no Departamento de Radiologia da Faculdade de Ciências Médicas da UNICAMP. A elaboração da rotina de conversão de dados 3D para o formato VIFF e da rotina que permite várias opções de interpolação (inclusive a interpolação combinada [11]) destes dados foi feita pelo Prof. Dr. Roberto de A. Lotufo.

Referências

- [1] Ekoule A.B., Peyrin F.C. e Odet C.L. "A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours". *ACM Transactions on Graphics*, 10(2):182-199, Apr 1991.
- [2] Phong B.T. "Illumination for Computer Generated Pictures". *Communication of the ACM: Graphics and Image Processing*, 18(6):311-317, Jun 1975.
- [3] Jense G.J. e Huijsmans D.P. "Interactive Voxel-Based Graphics For 3D Reconstruction of Biological Structures". *Computer & Graphics*, 13(2):145-150, Fev 1989.
- [4] Fuchs H., Levoy M. e Pizer S.M. "Interactive Visualization of 3D Medical Data". *IEEE Computer*, 22(8):46-50, Aug 1989.
- [5] Fuchs H., Kedem Z.M. e Uselton S.P. "Optimal Surface Reconstruction from Planar Contours". *Communications of the ACM*, 20(10):693-702, Oct 1977.

- [6] Rasure J. e Williams. "An Integrated Visual Language and Software Development Environment". *Visual Languages and Computing*, 2:217-246, 1991.
- [7] Udupa J.K. "Computer Aspects of 3D Imaging in Medicine: A Tutorial". Technical Report MIPG 153, Medical Image Processing Group, Dept. of Radiology, Univ. of Pennsylvania, Philadelphia, 1989.
- [8] Levoy M. "Display of Surfaces from Volume Data". *IEEE Computer Graphics and Applications*, 8(3):29-37, May 1988.
- [9] Levoy M., Hanrahan P., Hoehne K.H., Kaufman A. e Lorensen W. "Course Notes: Volume Visualization Algorithms and Architectures". In *SIGGRAPH 1990, 17th International Conference On Computer Graphics and Interactive Techniques*, Dallas, Aug 6th-10th 1990.
- [10] Stytz M.R., Frieder G. e O.Frieder. "Three-Dimensional Medical Imaging: Algorithms and Computer Systems". *ACM Computing Surveys*, 23(4):421-499, Dec 1991.
- [11] Lotufo R.A., Herman G.T. e Udupa J.K. "Combining Gray-level into Shape-based Interpolation". In *Visualization in Biomedical Computing 1992 (VCB'92)*, pages 13-16, Chapel Hill, North Carolina, USA, Oct 1992.
- [12] Tiede U., Hoehne K.H., Bomans M., Pommert A., Riemer M. e Wiebecke G. "Surface Rendering - Investigation of Medical 3D-Rendering Algorithms". *IEEE Computer Graphics and Applications*, pages 41-53, Mar 1990.