

INF1608 – Análise Numérica

Projeto: Visualização Volumétrica

Prof. Waldemar Celes
Departamento de Informática, PUC-Rio

Descrição

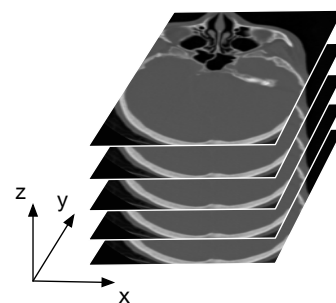
Dados volumétricos, como imagens 3D de tomografia computadorizada, são visualizados aplicando o modelo de iluminação que considera absorção e emissão. Numa visualização monocromática simplificada, baseada em traçado de raio, a intensidade de luz $I \in [0, 1]$ de cada pixel com este modelo é dada pela Integral de Renderização Volumétrica:

$$I = \int_0^L \tau(d(s)) e^{-\int_0^s \tau(d(t)) dt} ds$$

onde L representa o comprimento do raio traçado para cada pixel, $d(\cdot)$ é a densidade do volume e $\tau(\cdot)$ representa a opacidade em função da densidade ao longo do raio, dado pelo que é chamado de *função de transferência*. Para dados médicos, os dados de densidade são obtidos capturando imagens em fatias, formando uma grade regular de valores com dimensão n_x, n_y, n_z , onde n_x e n_y representam as dimensões de cada imagem e n_z o número de fatias, como ilustra a figura.

Por exemplo, o resultado de um *CT scan*, disponibilizado como exemplo, é representado por 99 imagens, cada uma na resolução de 256×256 , formando um volume com dimensão $256 \times 256 \times 99$. O valor da densidade no caso é representado em 8 bits, variando de 0 a 255. Para cálculo da integral, este valor é convertido para o intervalo $[0, 1]$.

O resultado de uma visualização volumétrica é uma imagem, como ilustrada na figura, que pode ser visualizada diretamente na tela do computador ou salva em formato de imagem. Para imagens em tons de cinza, um formato simples é o PGM:



```
P2
# exemplo.pgm
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

A primeira linha deve ser “P2”; em seguida, deve-se indicar a largura e altura da imagem (no exemplo, 24 e 7). A seguir, indica-se o maior valor a ser encontrado (no exemplo, 15). Depois, seguem os valores de cada linha da imagem.

Tarefa

O objetivo deste trabalho é gerar a imagem resultante de uma visualização volumétrica por traçado de raio do dado de *CT scan* disponibilizado.

Os dados do *CT scan* estão num arquivo binário com $256 \times 256 \times 99$ valores do tipo `unsigned char`, que devem ser carregados e armazenados num vetor em memória. Neste vetor, o acesso ao valor de índice (i, j, k) é dado pelo índice $(kn_y n_x + jn_x + i)$.

A imagem resultante deve ter resolução $n_x = 128 \times n_z = 99$ com os raios sendo traçados na direção y . Assim, o valor de cada pixel (i, k) da imagem, para $i \in [0, 127]$ e $k \in [0, 98]$, é obtido pelo valor médio da integral de dois raios traçados no volume: os raios com origem em $(2i, 0, k)$ e $(2i + 1, 0, k)$, com direção de traçado igual a $(0, 1, 0)$, até que se alcance o final dos raios $(2i, n_y - 1, k)$ e $(2i + 1, n_y - 1, k)$. Note então que o comprimento total do raio será de $n_y - 1 = 255$ unidades; no entanto, o intervalo de integração pode ser dado por um número ponto flutuante h . Isso significa que é necessário acessar o valor de densidade $d(\cdot)$ de índice (i, j_f, k) , onde j_f é um “índice” ponto flutuante. O valor associado a este índice deve ser a interpolação linear dos valores nos índices (i, j, k) e $(i, j + 1, k)$, onde $j = \text{floor}(j_f)$. Para o cálculo das integrais, considere o Método de Simpson com $h = 4.5$. Numa primeira etapa, no entanto, use um valor de h inteiro, evitando assim a necessidade de interpolação.

Considere uma função de transferência que especifica:

$$\tau(d(t)) = \begin{cases} 0.0, & \text{se } d(t) < 0.3 \\ 0.05(v - 0.3), & \text{caso contrário} \end{cases}$$

Gere um arquivo no formato PGM com a imagem resultante da visualização volumétrica. O valor final de cada integração, deve ser limitado ao intervalo $[0, 1]$ e então mapeando para um `unsigned char`.

Numa segunda etapa, implemente o método de integração Simpson adaptativo, aceitando erro local dado por um parâmetro *tol*.

Análise

Ao desenvolver seu trabalho e testá-lo, procure, baseado em experimentos computacionais, responder as seguintes perguntas:

- Seu programa gera corretamente a imagem final?
- Qual o desempenho de seu programa usando diferentes passos de integração? A qualidade da imagem é afetada?
- Qual o desempenho e a qualidade da imagem quando se usa método adaptativo de integração?
- Que alterações seriam necessárias no seu algoritmo para usar um raio de visualização qualquer, diferente de $(0, 1, 0)$?