

Rendering Volumétrico Direto – A versão resumida

Haim Levkowitz

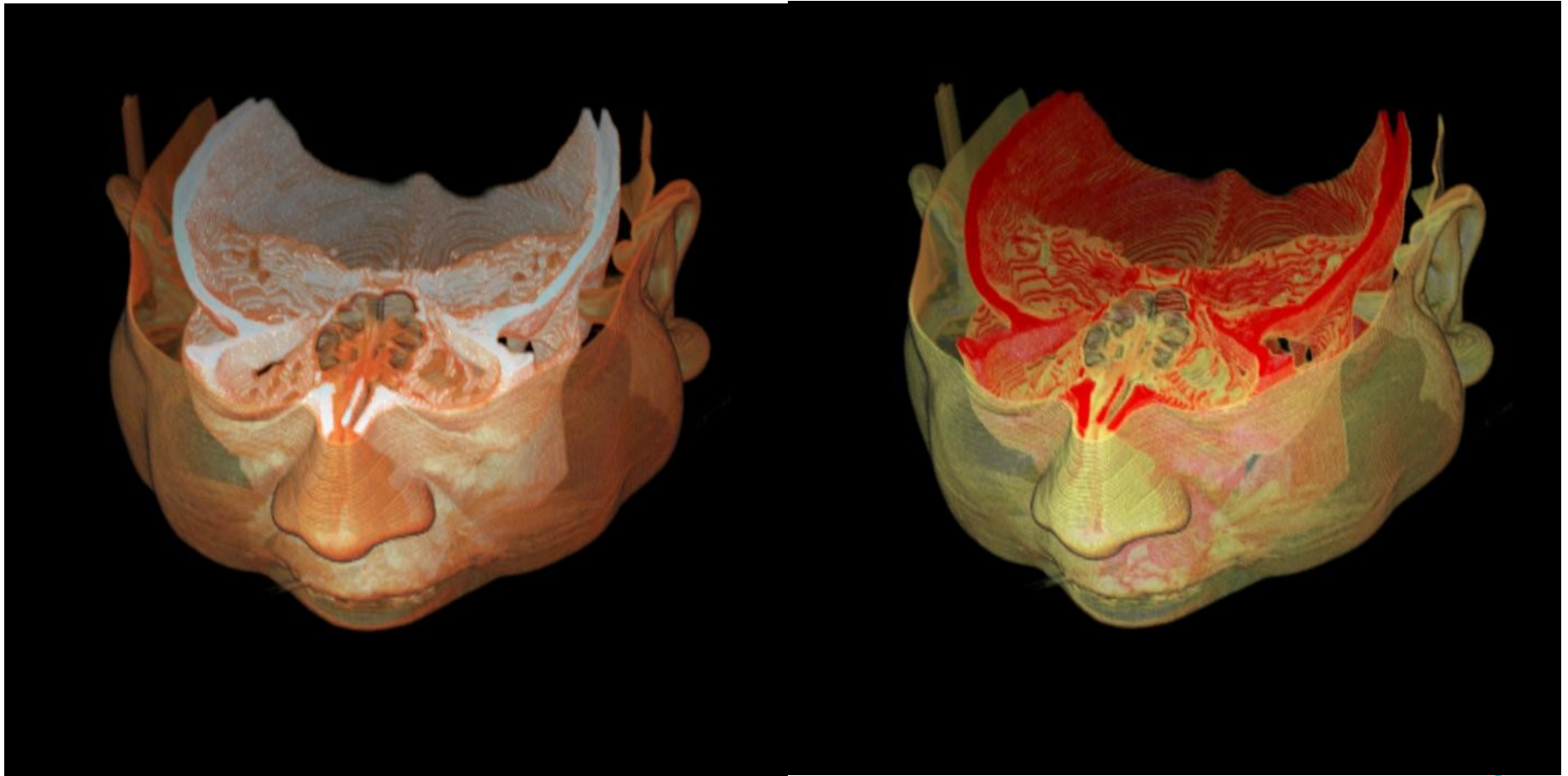
Maria Cristina F. de Oliveira

Rosane Minghim

Overview

- Rendering de superfícies vs. volumes
- Rendering Volumétrico Direto
 - Ray casting
 - Abordagens ordem da imagem vs. ordem dos objetos

Volume Rendering



<http://www.ablesw.com/3d-doctor/volume.html>

Volume Rendering

- Capacidade de exibir conteúdo interno ao volume
- Exibição direta
- dados volumétricos \Rightarrow pixels na tela
- transparência no rendering para 'enxergar' através do volume

Volume vs surface

Volume rendering

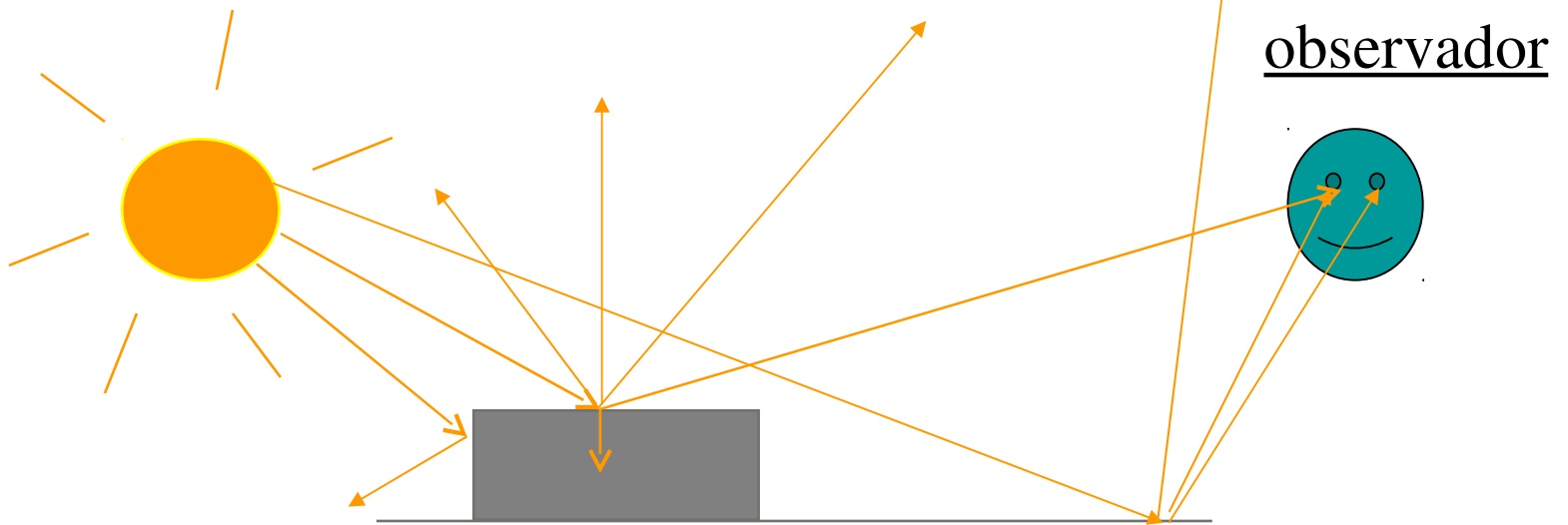
- Problema complexo em termos de
- Rendering
 - Conteúdo 3D em imagens 2D
- Custo computacional
 - Memória e processamento
 - Dados volumétricos são GRANDES
 - Superfícies: 1 milhão de polígonos é bastante...
 - 1 milhão de voxels é um conj de dados pequeno (100 x 100 x 100)
 - Visualização requer **interação**

Volume rendering

- Rendering de objetos semi-transparentes
- Superfícies: rendering assume que superfícies refletem ou absorvem luz, mas não que a luz a transmitida através do objeto
- Transparência: propriedade de transmitir luz
 - Alpha, ou opacidade, em CG
- Opacidade: propriedade de não transmitir luz

Processo Físico de Geração de uma Imagem

Fonte de luz



cena

Transparency and alpha values

- Para volumes, precisamos considerar como a luz se comporta **dentro** de um material
 - i.e., objetos transmitem luz
- Alpha
 - Specification of opacity (or transparency)
 - Alpha = 1 → object opaque
 - Alpha = 0 → object completely transparent
 - $0 < \text{Alpha} < 1$ → object translucent

The formula ($A = \alpha$)

$$R = A_s R_s + (1 - A_s) R_b$$

$$G = A_s G_s + (1 - A_s) G_b$$

$$B = A_s B_s + (1 - A_s) B_b$$

$$A = A_s + (1 - A_s) A_b$$

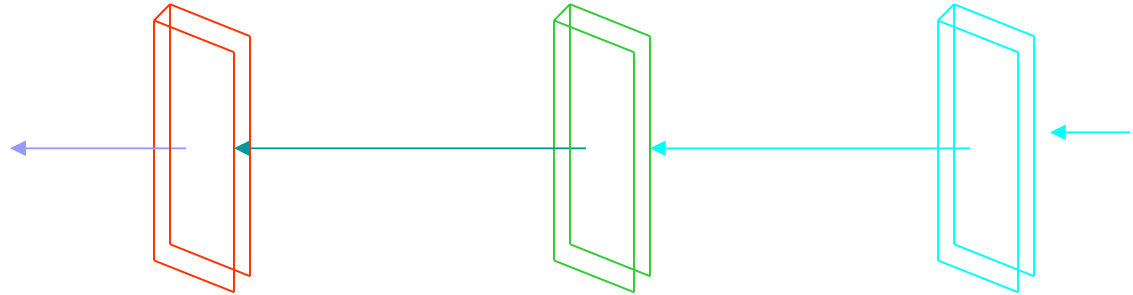
where (R_s, G_s, B_s, A_s) : front surface

(R_b, G_b, B_b, A_b) : surface behind

(R, G, B, A) : resulting color

Example

Polygon Color RGBA (0.8, 0, 0, 0.5) (0, 0.8, 0, 0.5) (0, 0, 0.8, 0.5)



Resulting Color (0.4, 0.2, 0.1, 0.875) (0, 0.4, 0.2, 0.75) (0, 0, 0.4, 0.5)

Volume Rendering

- Motivation
 - visualize inherently volumetric data
 - MRI, CT, confocal microscope, ultrasound
 - Weather analysis
- Volume
 - Regular array of points in 3D space
 - Often defined as series of 2D images (slices) arranged along z-axis

Volume Rendering

- Directly view volume data
- No intermediate surface primitives
 - No geometry
- Contrast with Surface Rendering

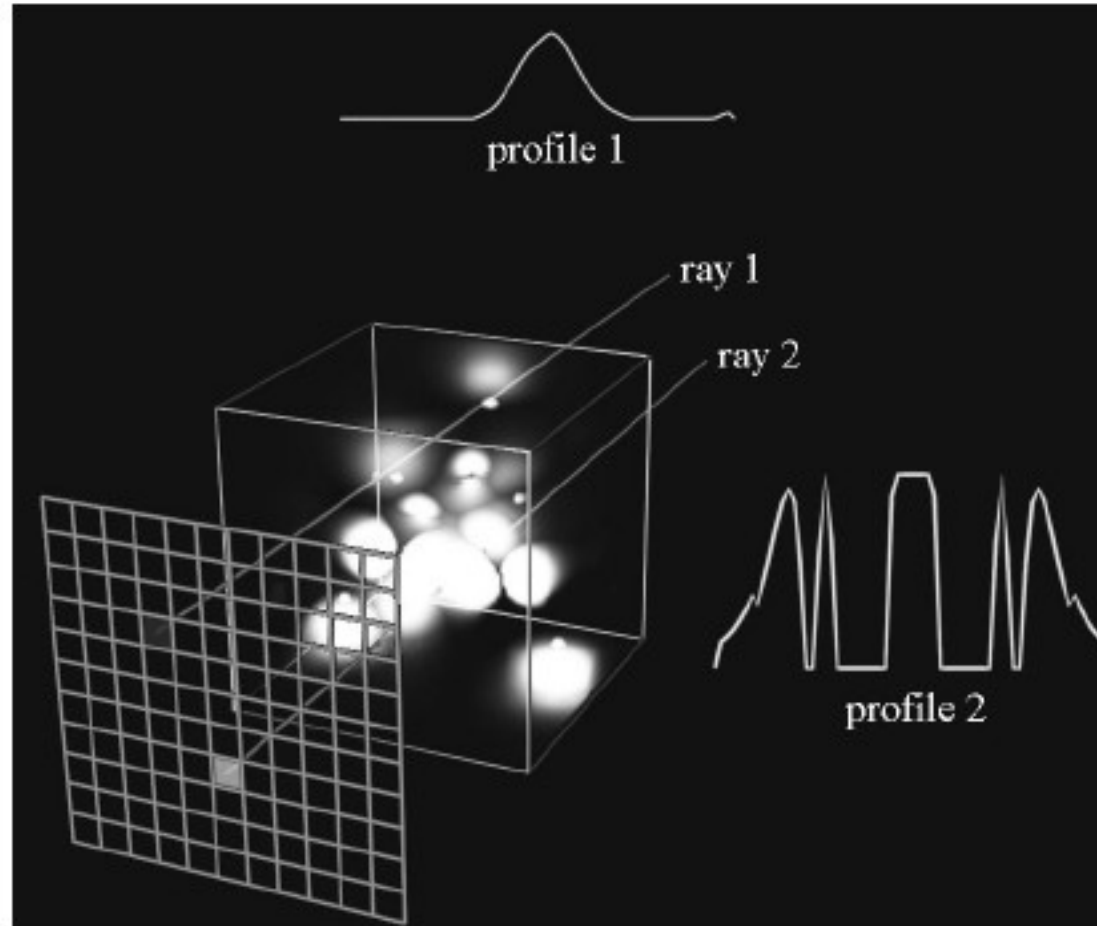
- *Determine* voxels that contribute to each pixel in image
- *Integrate* contributions of those voxels to obtain color for that pixel
- How integration performed is key difference between volume rendering techniques

Image-order Volume Rendering

- “Ray casting”
- Basic idea
 - Determine value of each pixel in image
 - Send ray through pixel into scene according to current camera parameters
 - Evaluate data along ray using some specified function
 - → compute pixel value

The ray-casting process

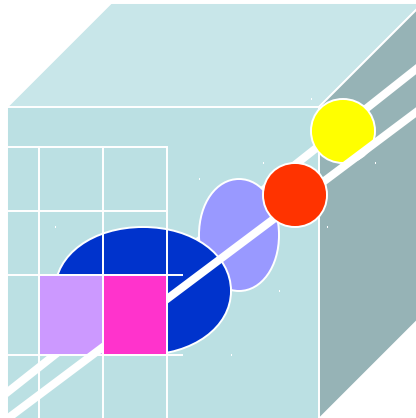
- Para cada raio, amostra os valores escalares no volume
- Coleta perfil de valores escalares
- Como converte esse perfil em uma intensidade para exibição?



d

The ray-casting process

Scalar value



maximum value

average

Distance from
view plane

d

Ray functions

- Pergunta: que função usar para mapear os escalares em intensidades?
- (ignoramos cor, por enquanto)
- Modelo de iluminação para dados volumétricos
 - Assume que voxels emitem luz proporcionalmente ao seu valor escalar

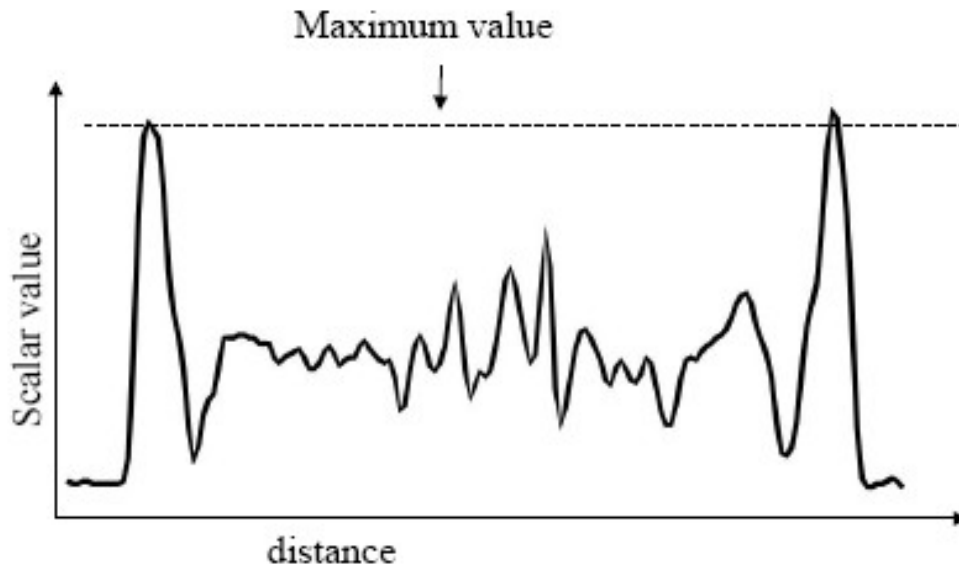
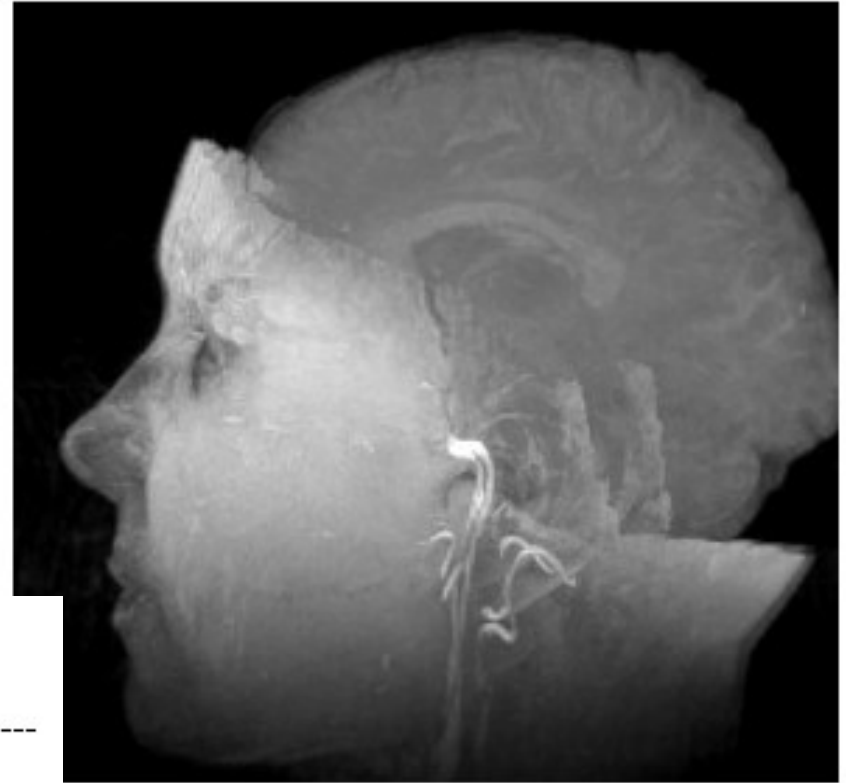
Ray functions

- Que função usar?
- (1) Maximum value
- (2) Average value
- (3) Threshold distance
- (4) Alpha composite technique

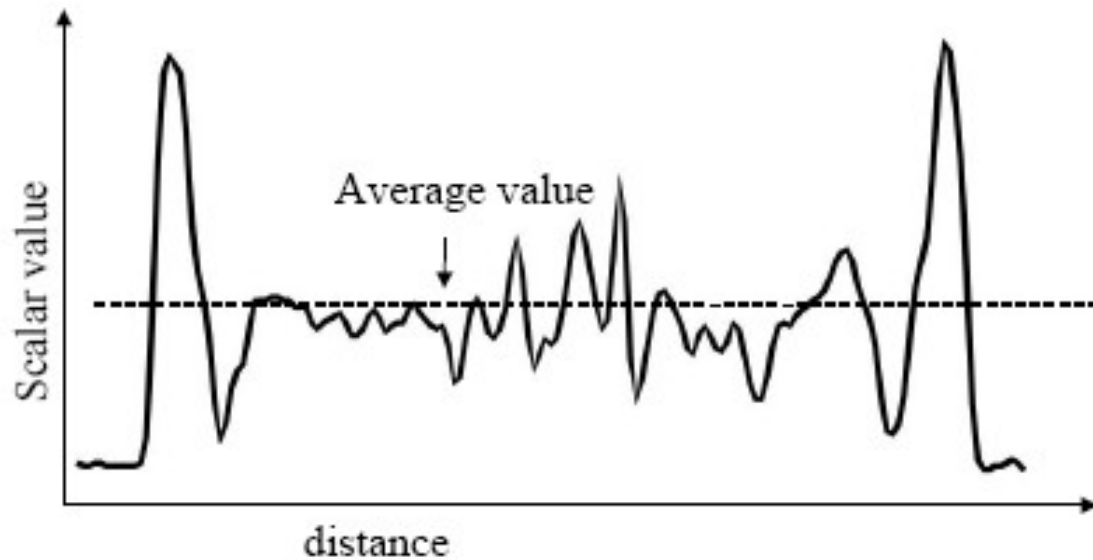
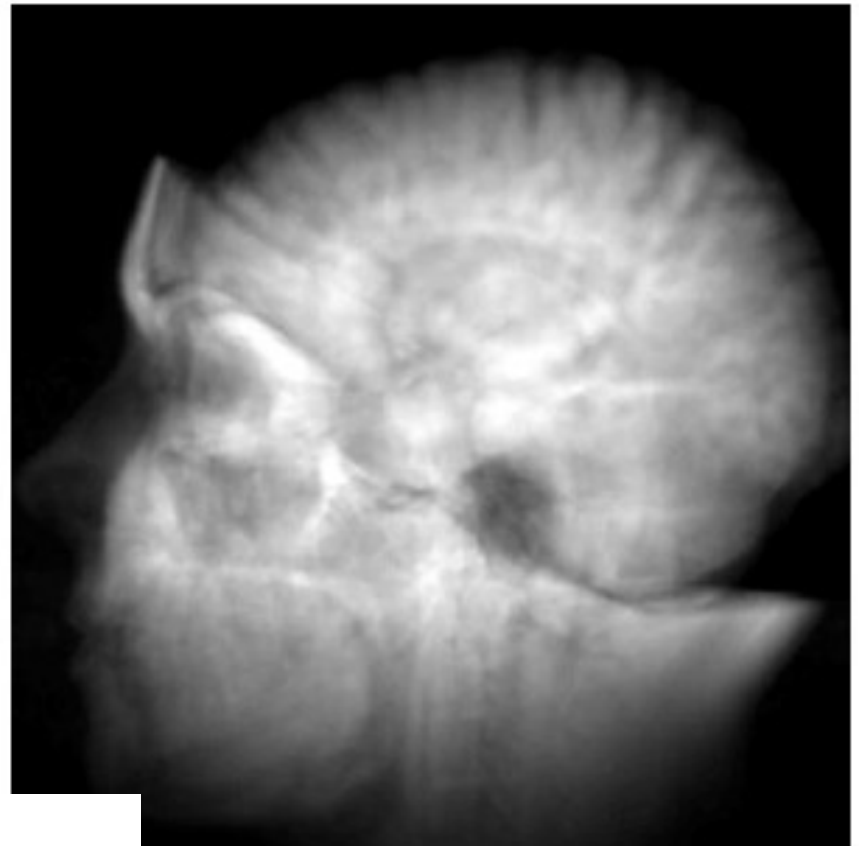
Maximum Value

Sem blending/transparência

Limitação: percepção de profundidade...



Average



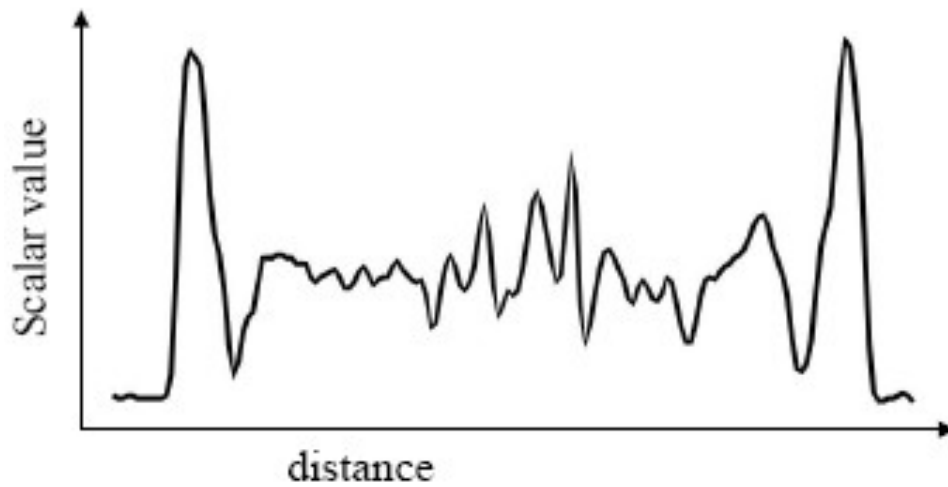
Threshold distance

- At or above threshold that ray first encounters
- Intensidade determinada pela profundidade do voxel/célula

Alpha composite technique

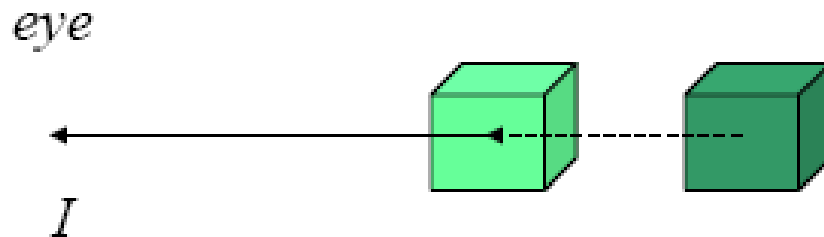
- Treat values along ray as samples of opacity accumulated per unit distance
- Valor acumulado ao longo do raio
- Opção 1: valor escalar amostrado usado como o valor alfa do voxel
 - \Rightarrow mais brilhante (intenso) = mais visível
 - Captura informação de oclusão
 - Percepção de profundidade relativa
 - Voxels mais a frente favorecidos

Composite



Composite

- Opção 2: especificação de transparência independentemente do valor escalar
- Valor escalar = 'brilho' célula



- Luz no olho (I) = $A E$
 - 'brilho' da célula + luz transmitida
- A (opacidade) * E (luz emitida pela célula)

Composite

- Se célula é totalmente transparente, seu 'brilho' é invisível
 - Mas o brilho das células atrás dela é visível
- Se célula é totalmente opaca, seu 'brilho' é visível
 - Mas o das células atrás dela não é
- Back-to-front ray casting
 - Precisa armazenar apenas o valor corrente de I

Composite

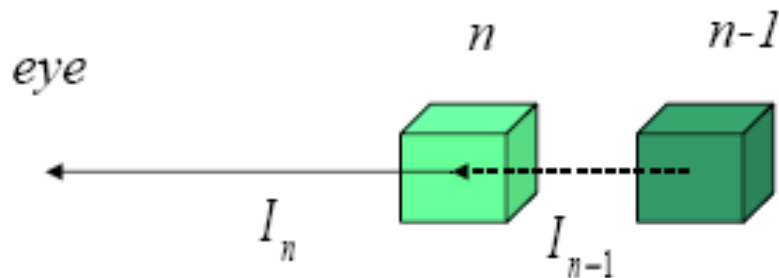
□ Back-to-front ray casting

□ só precisa armazenar o valor corrente de I

$$I_n = A_n E_n + (1 - A_n) I_{n-1}$$

Subscript n refers to cell n .

A refers to object *opacity*.



- Start with furthest away cell and blend towards the camera.

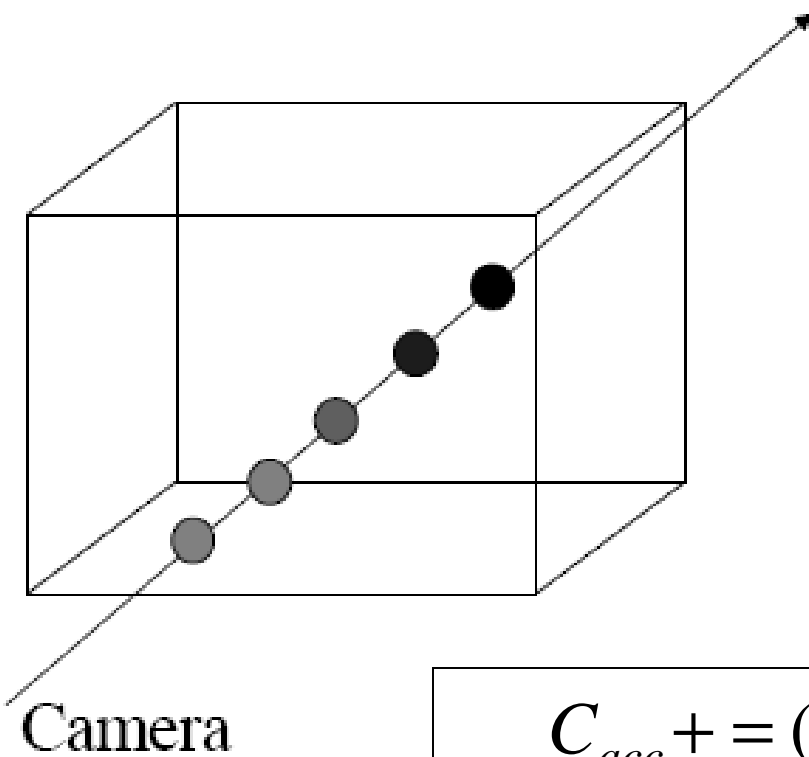
- I_n corresponds to current contents of the frame buffer.

- E_n Light emitted from cell n

Podem ser feitos cálculos irrelevantes se houver voxels muito opacos na frente...

Composite

□ Front-to back ray casting

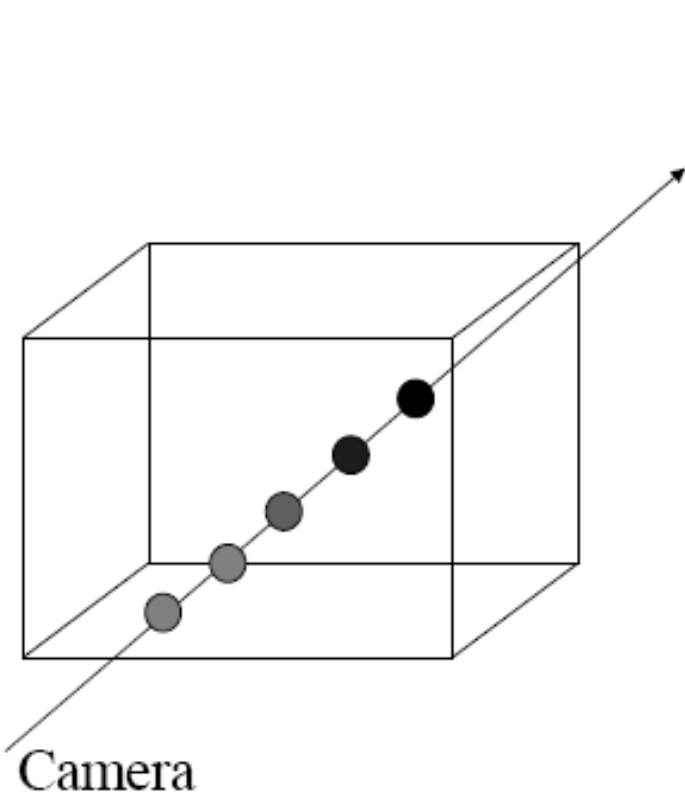


- buffer para armazenar o alfa corrente permite acumular começando das células mais a frente
- término prematuro qdo opacidade se aproxima de 1.0

$$C_{acc} + = (1 - \alpha_{acc}) \times (\alpha_{sample} C_{sample})$$

$$\alpha_{acc} + = (1 - \alpha_{acc}) \times \alpha_{sample}$$

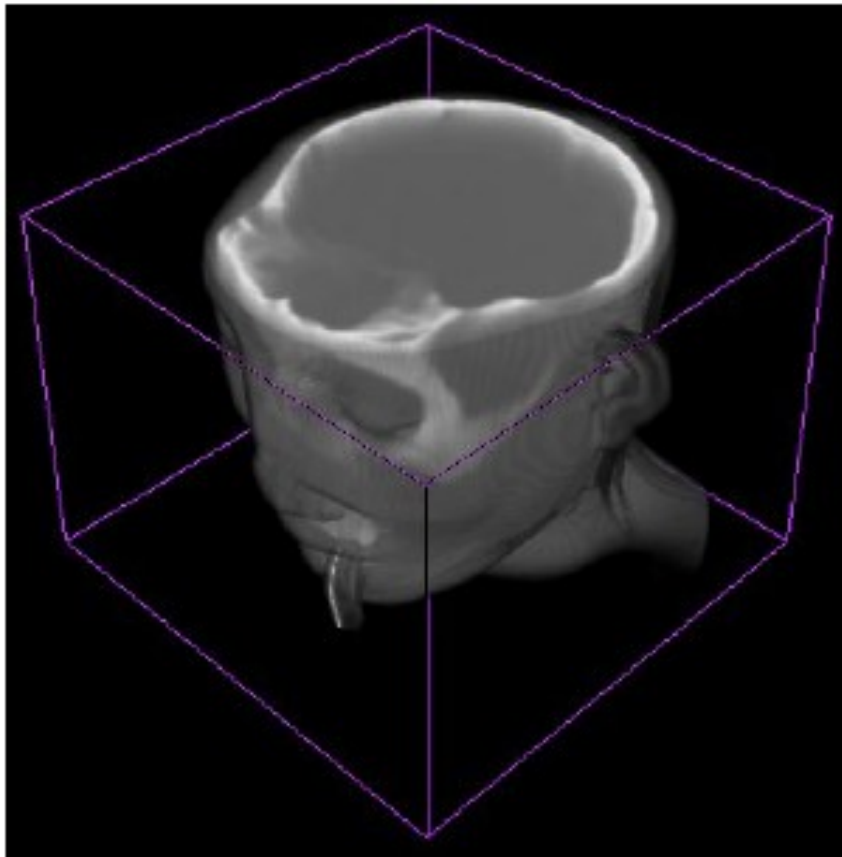
Composite front-to-back



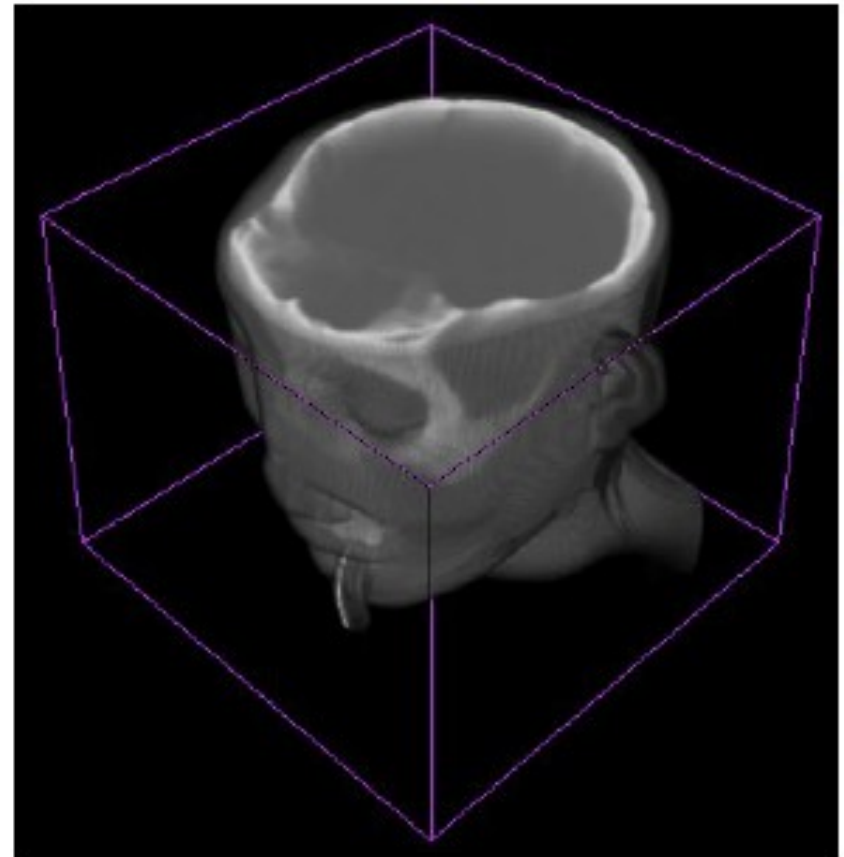
Current Alpha	Accumulated Alpha
0.2	0.2
0.4	$0.4 + (1 - 0.4) * 0.2 = 0.52$
0.5	$0.5 + (1 - 0.5) * 0.52 = 0.76$
0.4	$0.4 + (1 - 0.4) * 0.76 = 0.86$
0.6	$0.6 + (1 - 0.6) * 0.86 = 0.94$

Pode interromper processo qdo
opacidade (alfa) ~ 1.0

Efeito do tamanho do passo

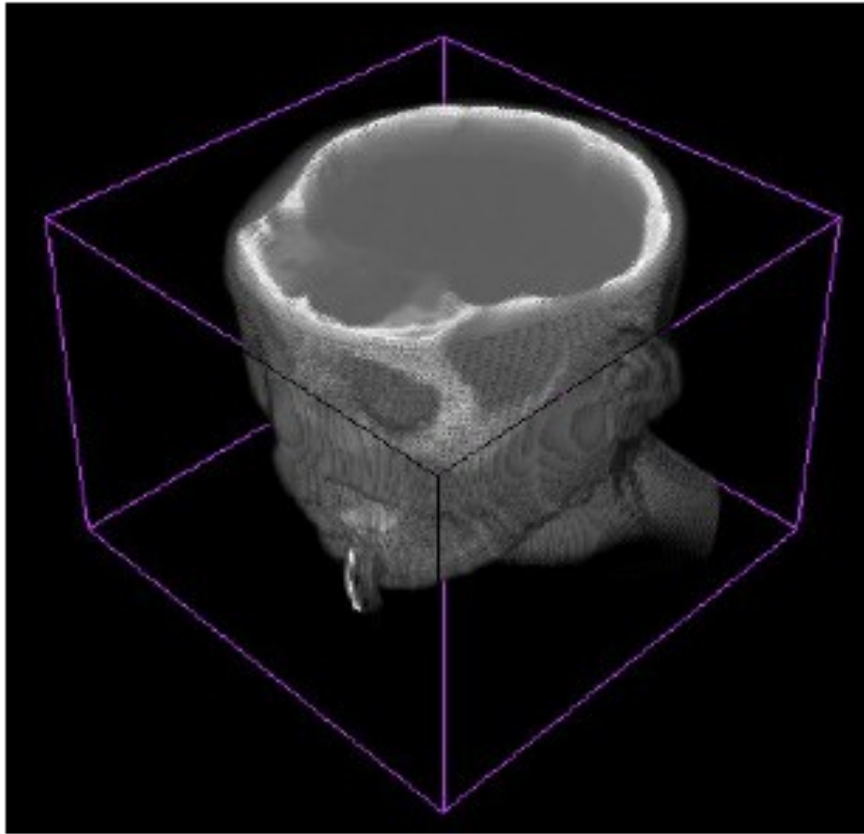


Step = 0.2

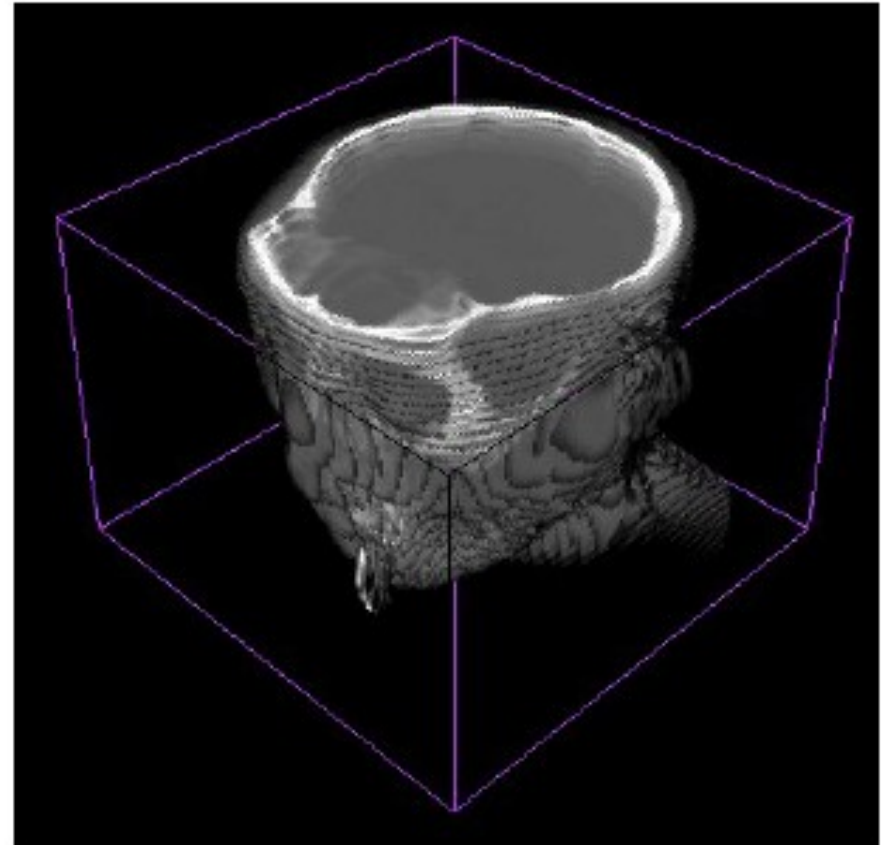


Step = 1.2
(mild artefacts)

Efeito do tamanho do passo

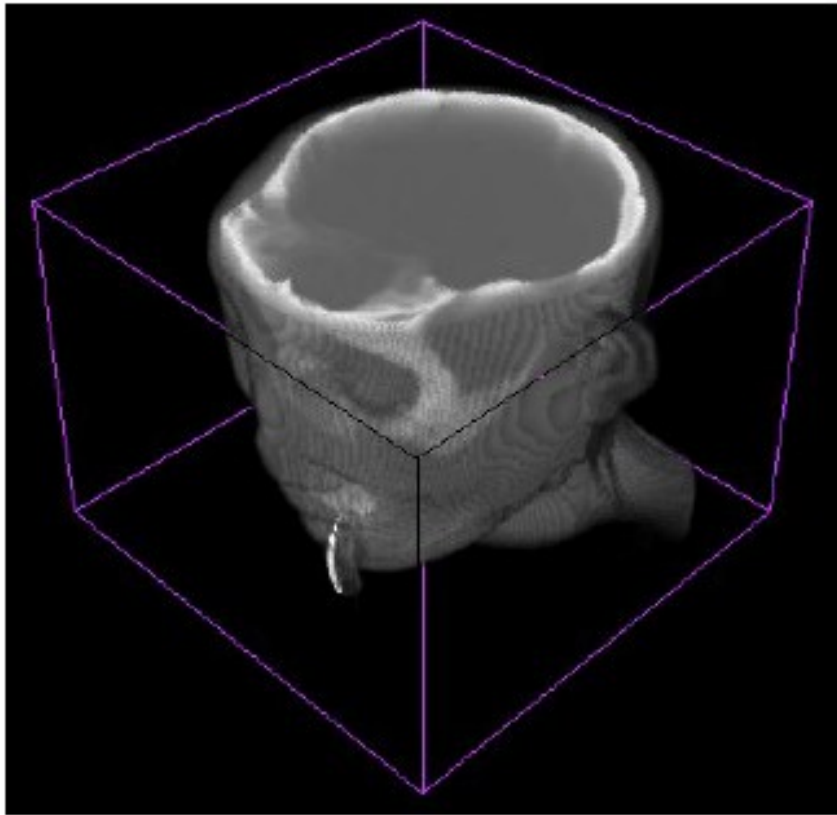


Step = 3.5
(visible artefacts)

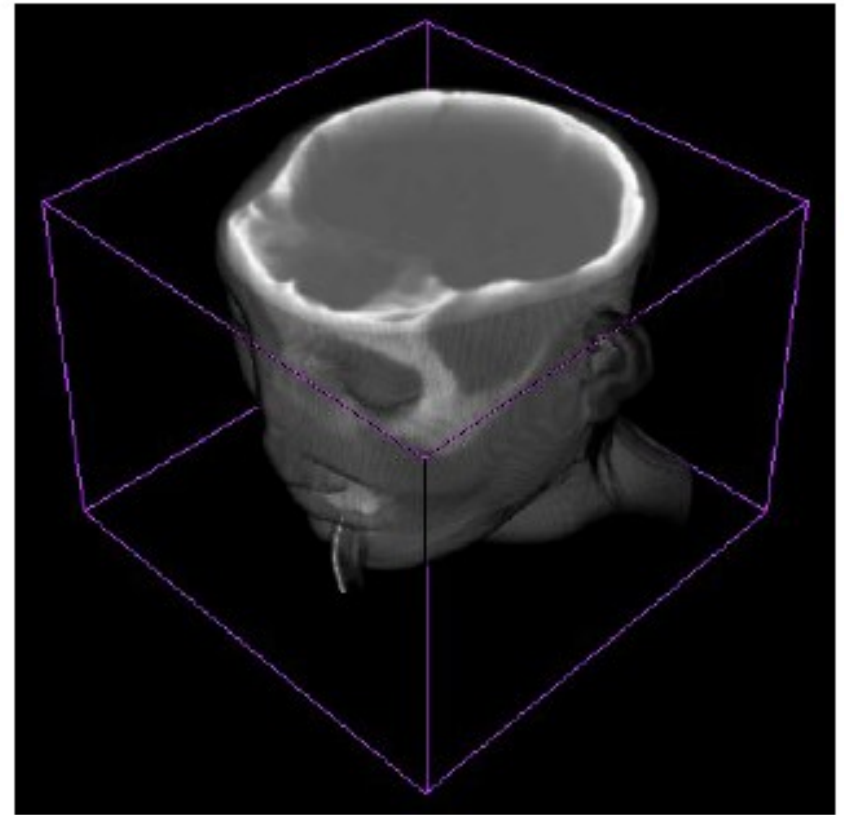


Step = 7.0
(highly visible artefacts
- distract from data itself)

Efeito da interpolação



Nearest neighbour
(visible artefacts in rendering)



Tri-linear interpolation (i.e. in X, Y & Z)
(smoother, no artefacts)

Step = 1.0

Acceleration methods

- Space leaping (group of techniques)
 - Attempt to avoid processing regions of volume that will not contribute to final Image
 - E.g., build octree data structure ...

Conclusions

- Obrigada!!!!