Karlsruher Institut für Technologie

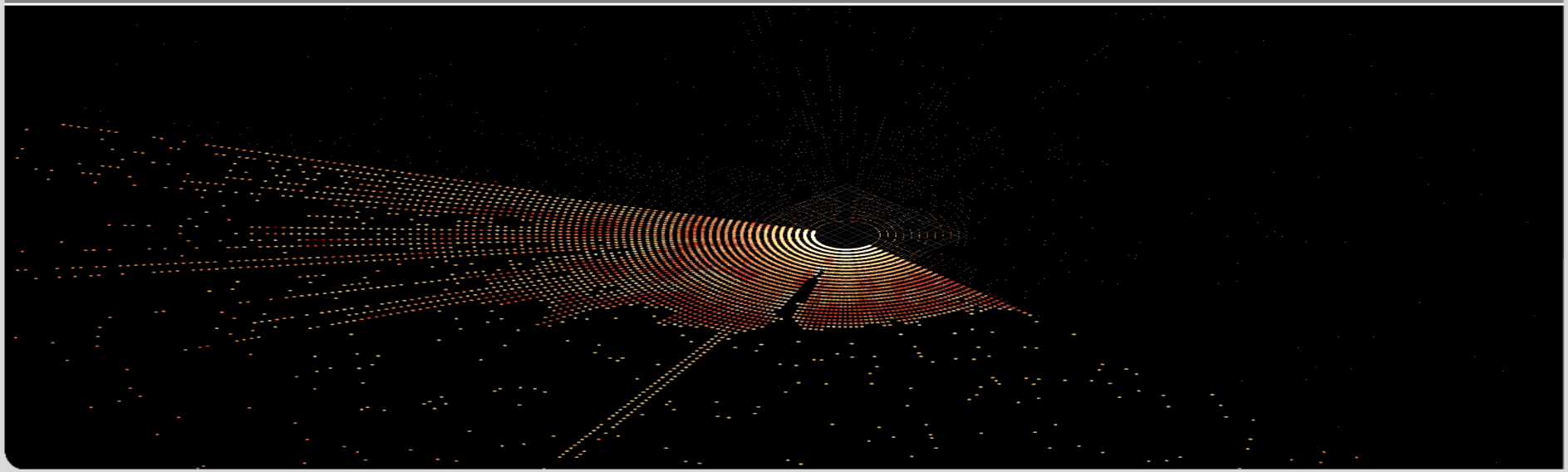
Workshop on Parallel Computing for Data Acquisition and Online Monitoring
KIT, Karlsruhe –7th March 2016

3D Web Visualization

Volume Ray Casting

Nicholas Tan Jerome

KIT, Institute for Data Processing and Electronics (IPE)

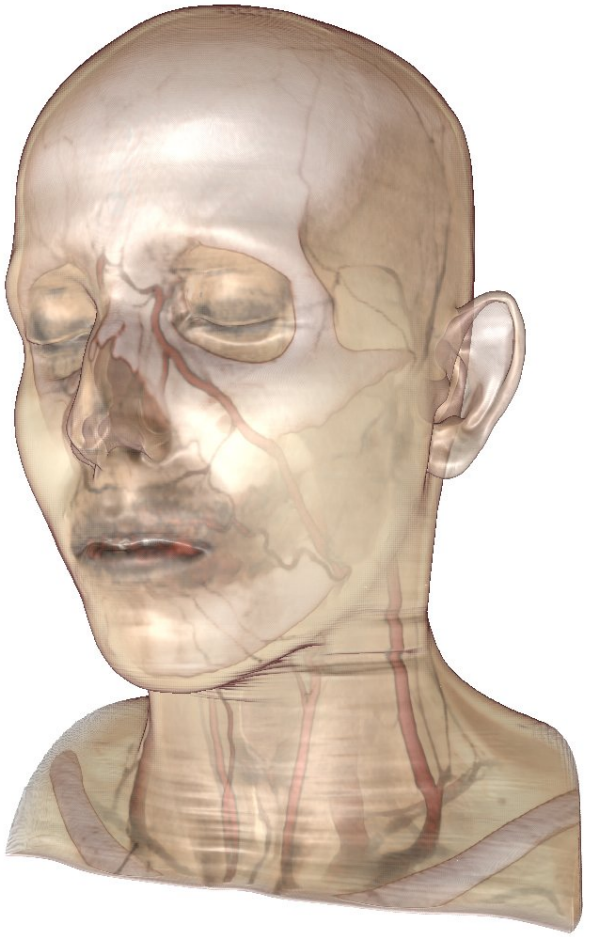


Outline

- Introduction
- Overview Volume Visualization
- Volume Ray Casting
- GPU Implementation
- Demo

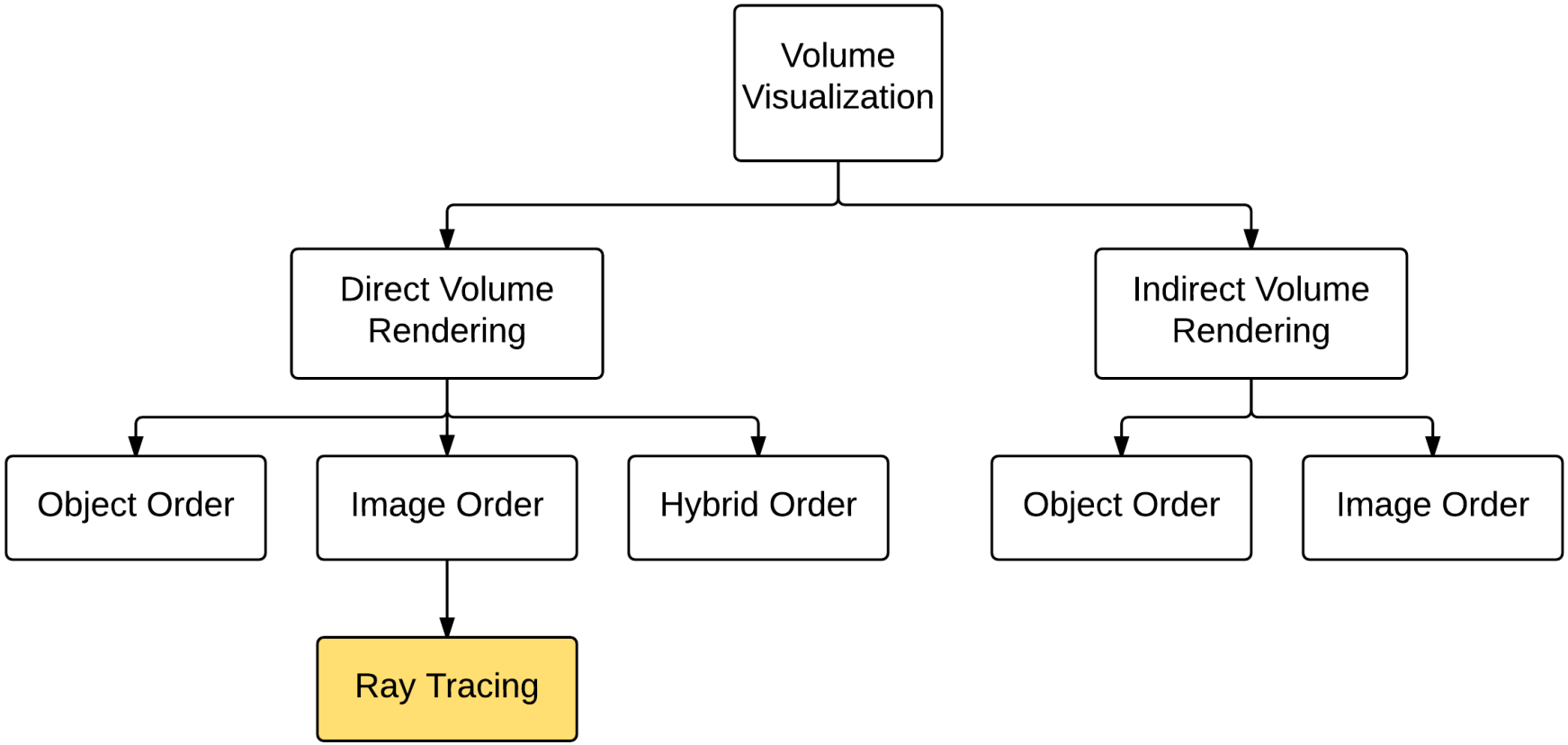
Introduction

- In traditional rendering, 3D objects are created using surface representation.
- Volume Rendering visualize the entire 3D data as a 2D image.
- Ray casting is a common technique in volume rendering.

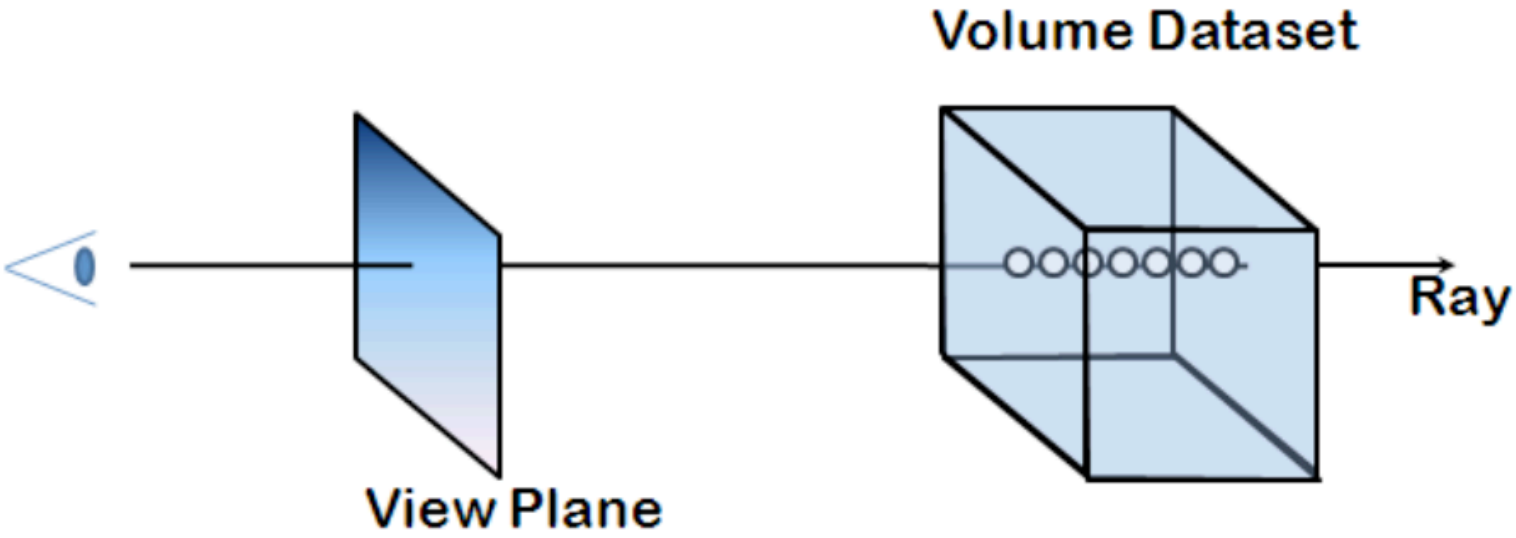


Source: Image from Course Note Advanced Illumination Techniques for GPU-Based Volume Raycasting

Overview

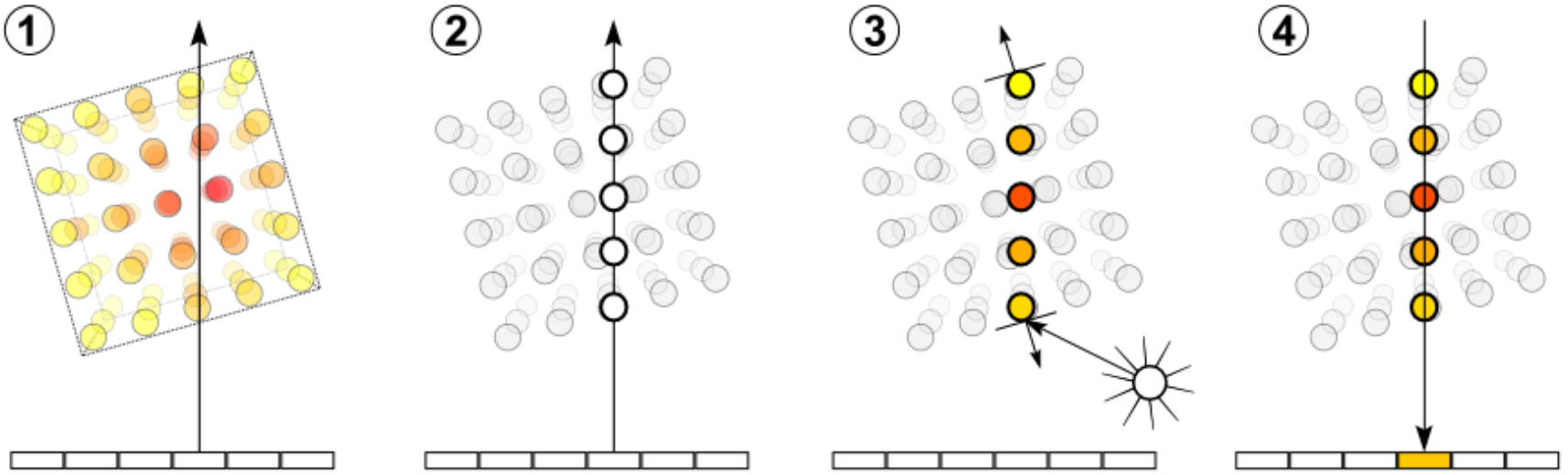

Volume Visualization
Direct Volume Rendering
Indirect Volume Rendering
Object Order
Image Order
Hybrid Order
Object Order
Image Order
Ray Tracing

Volume Ray Casting



Volume Ray Casting Steps

- Volume ray casting algorithm comprises of four steps

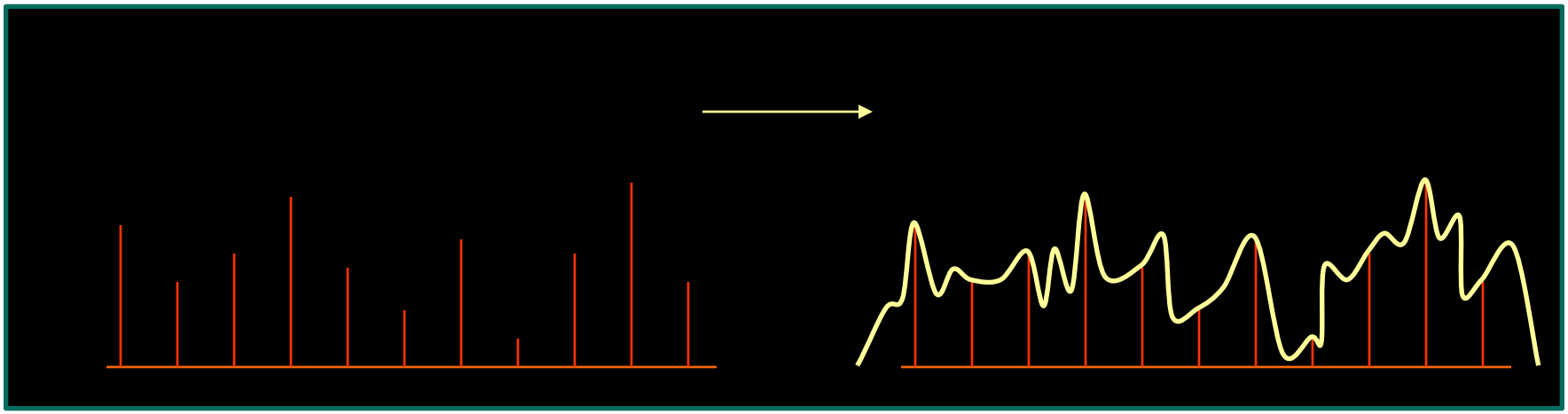


Ray casting Sampling Shading Compositing

2. Sampling

Interpolation

- Estimate values between samples
- Reconstruct continuous function

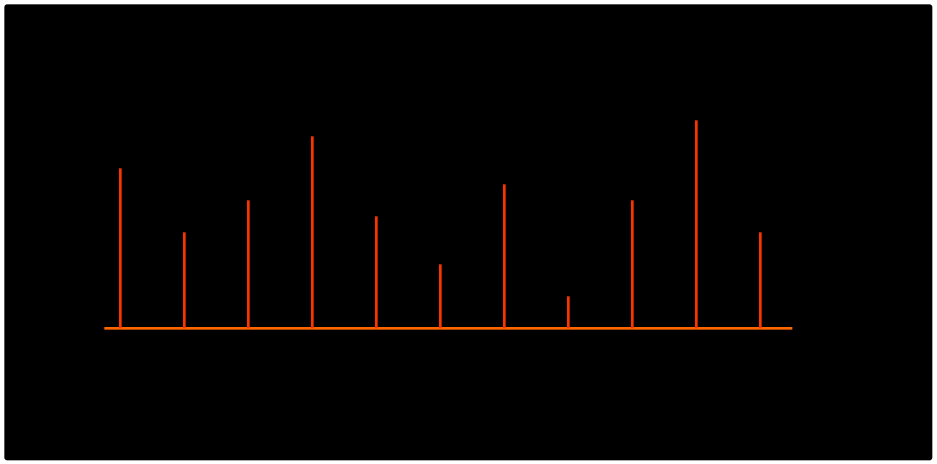


Sampled Function

Real Function

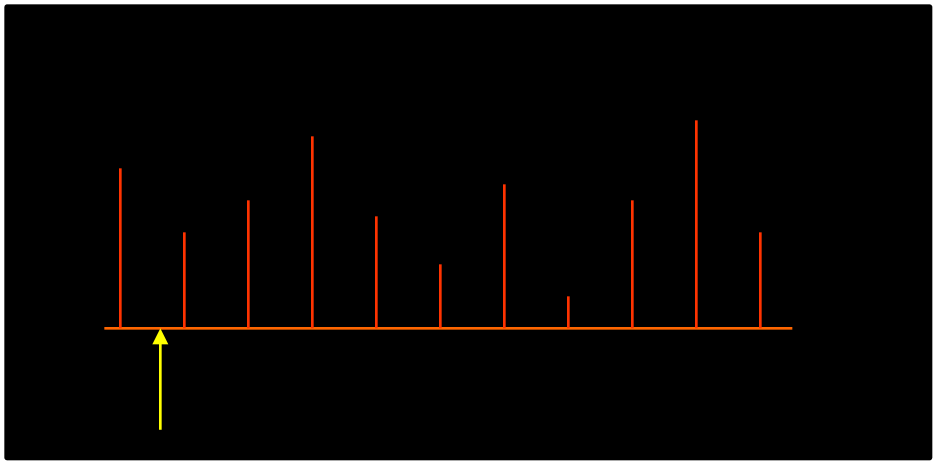
Interpolation

Laura Bush



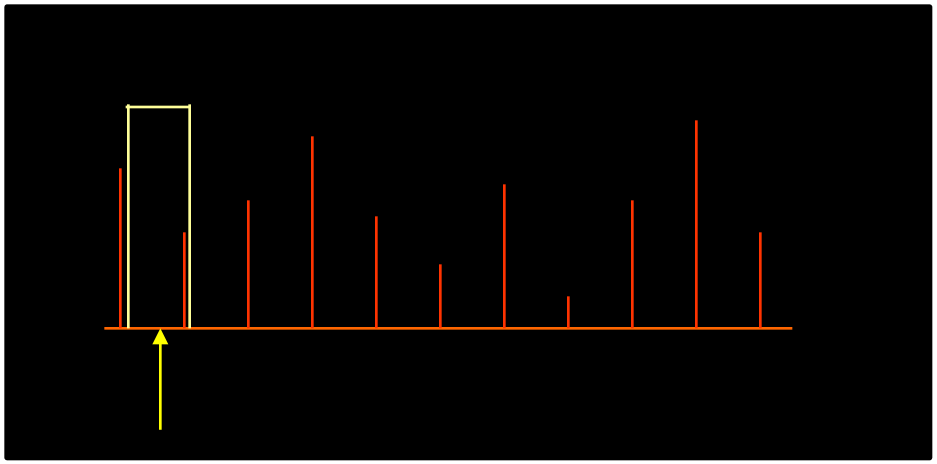
Sampled Function

Interpolation



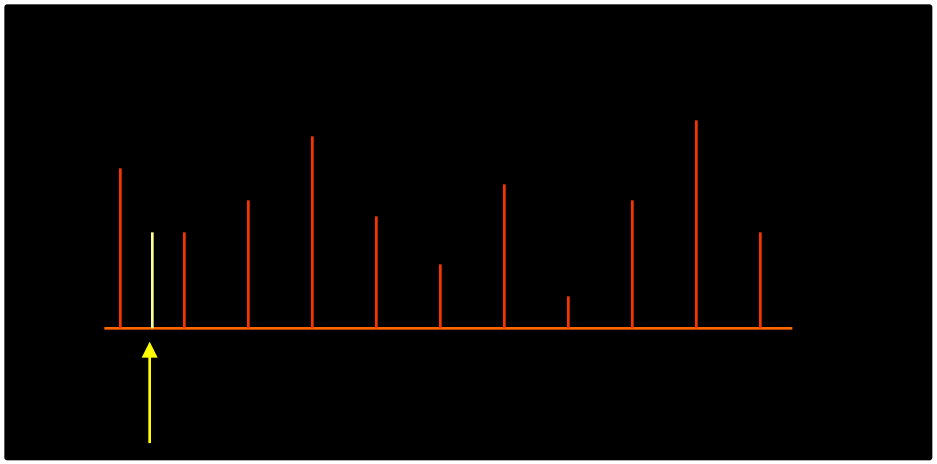
What is the value here

Interpolation



What is the value here

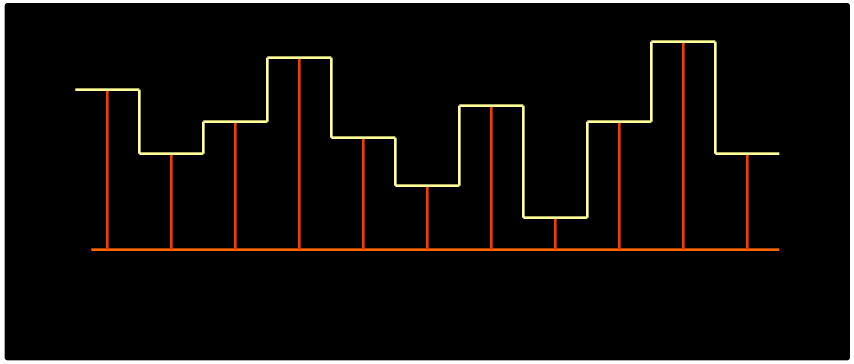
Interpolation



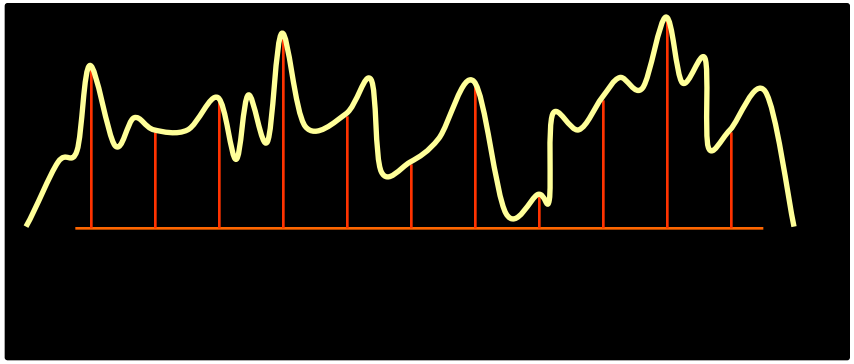
What is the value here

Interpolation

Box == Nearest Neighbor

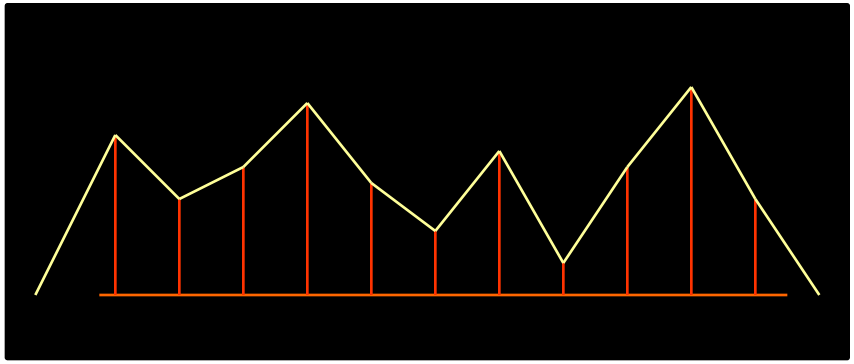


Reconstructed function

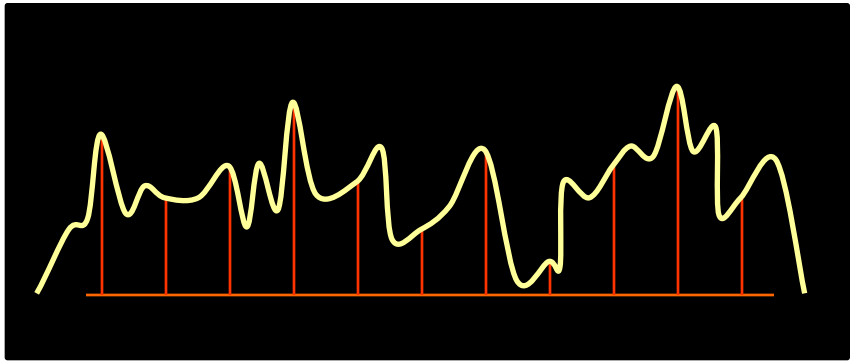


Interpolation

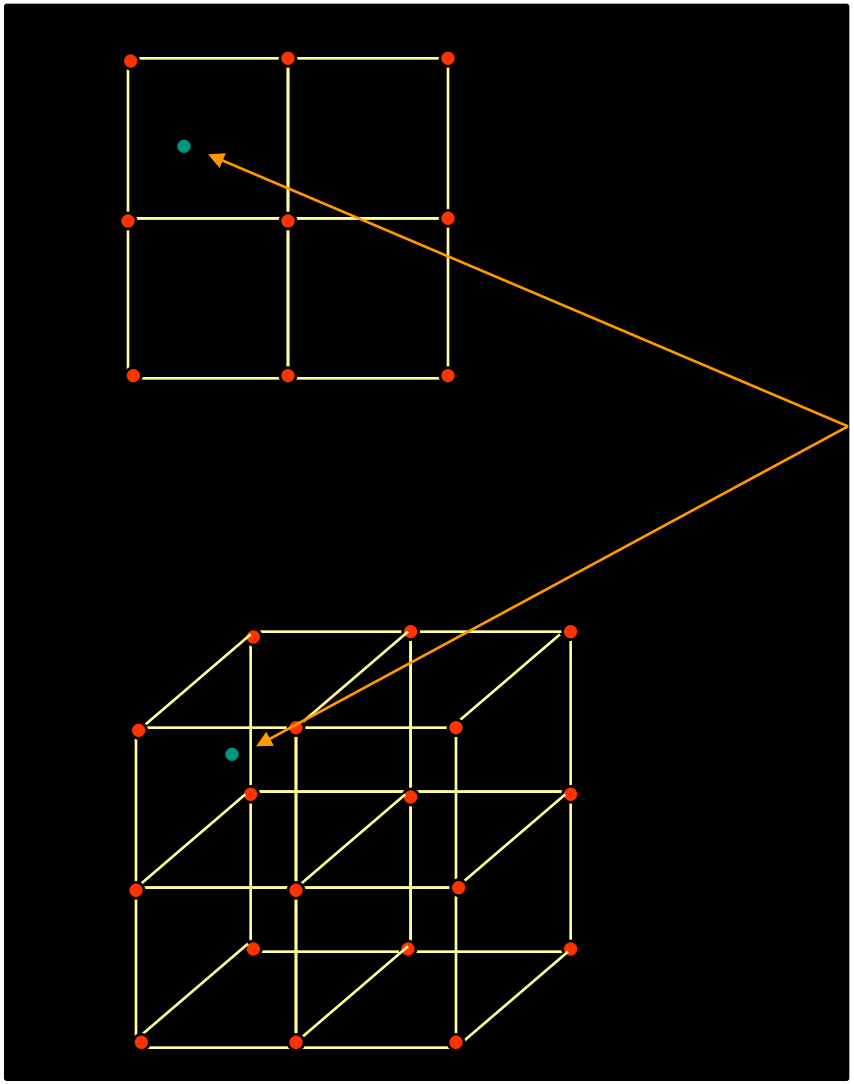
Linear == connect the dots



Reconstructed function



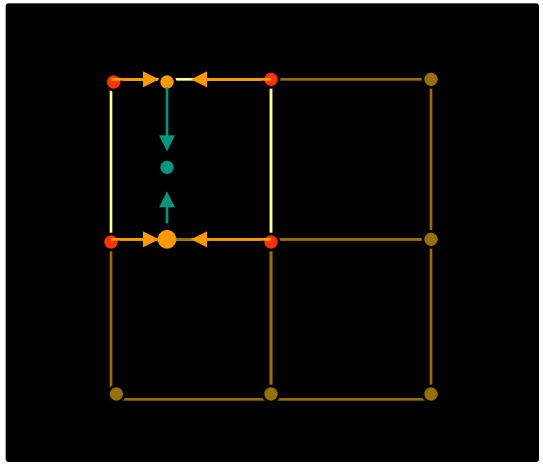
Interpolation



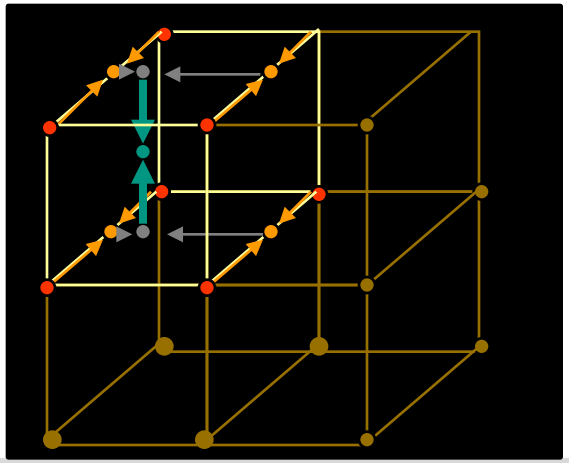
What are the values here?

Interpolation

- Bilinear
  - 3 linear interpolations along two axes
  - 1st along x
  - 2nd along y



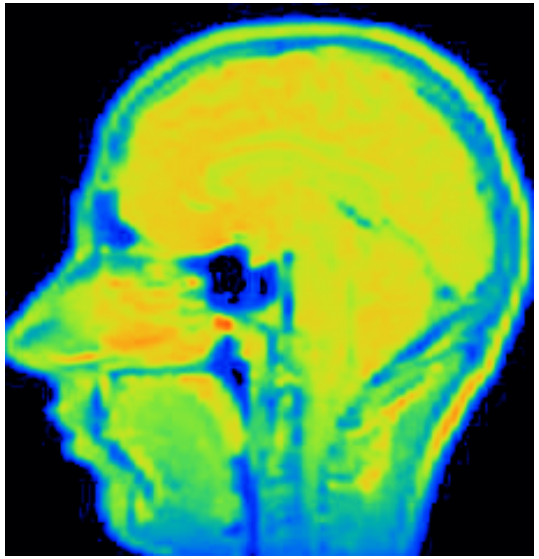
- Trilinear
  - 7 linear interpolations along three axes
  - 1st along x
  - 2nd along y
  - 3rd along z



3. Shading

Transfer function

- Assign optical property to data
  - Colour
  - Opacity

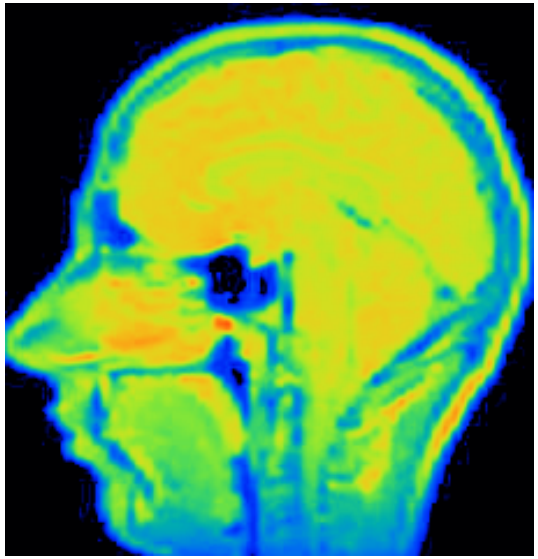


Transfer function

Transfer function

- Assign optical property to data
  - Colour
  - Opacity

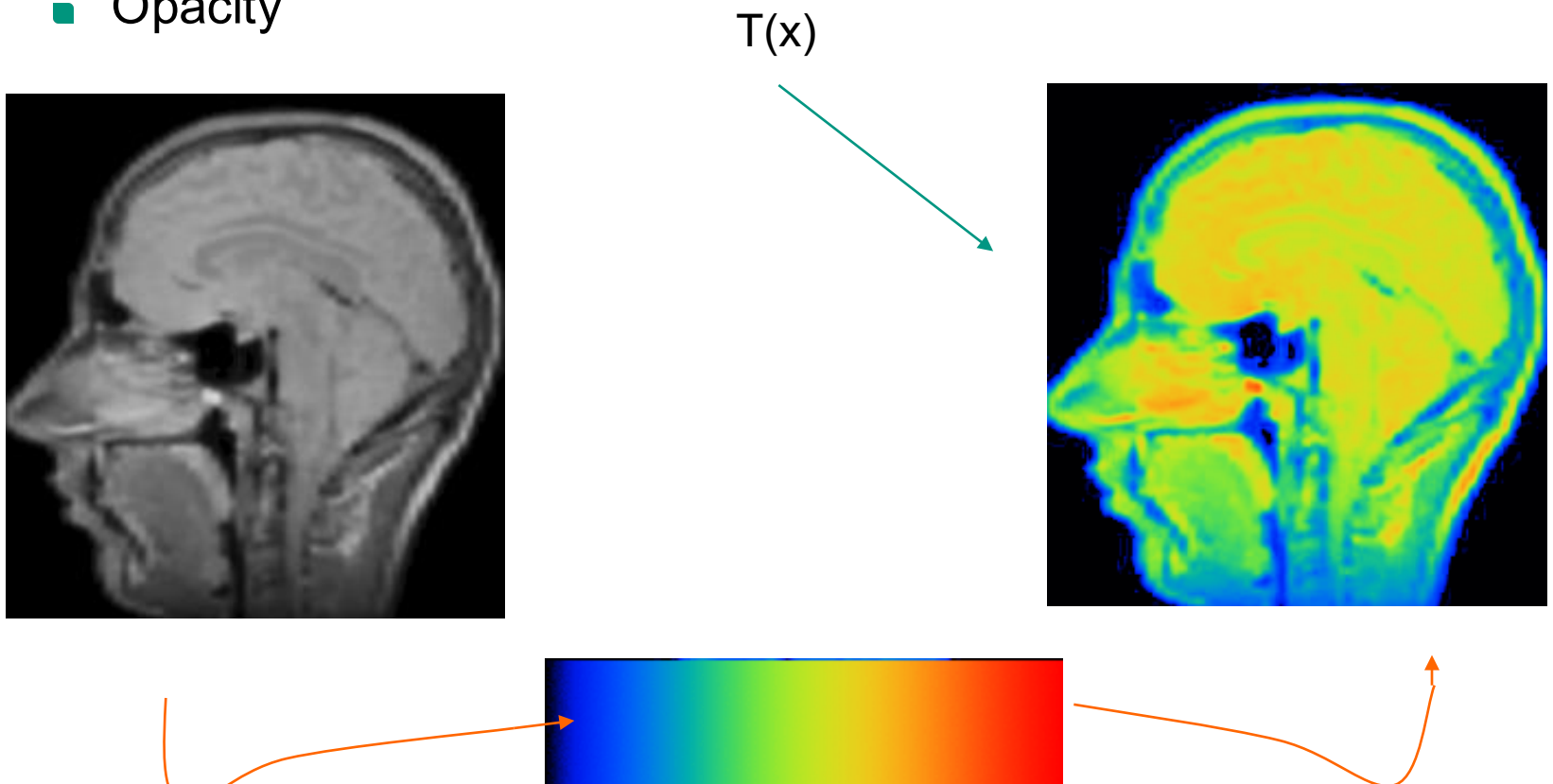
x



Transfer function

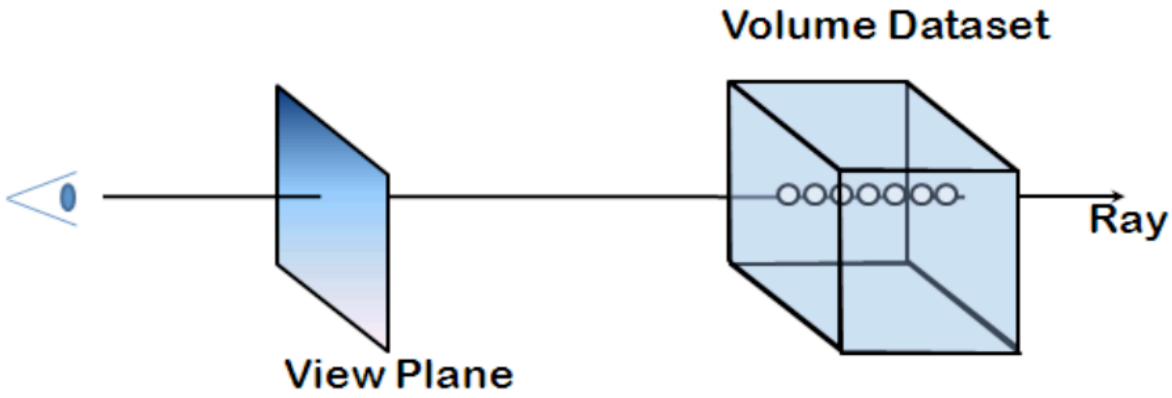
Transfer function

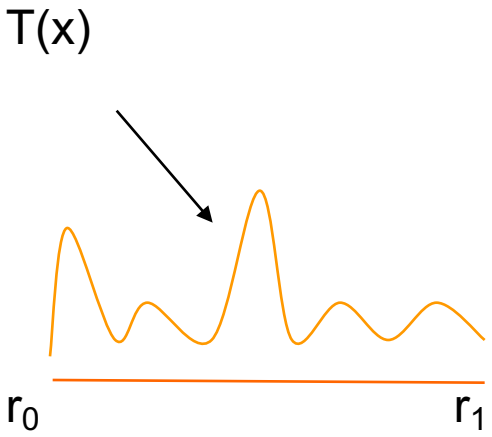
- Assign optical property to data
  - Colour
  - Opacity



4. Compositing

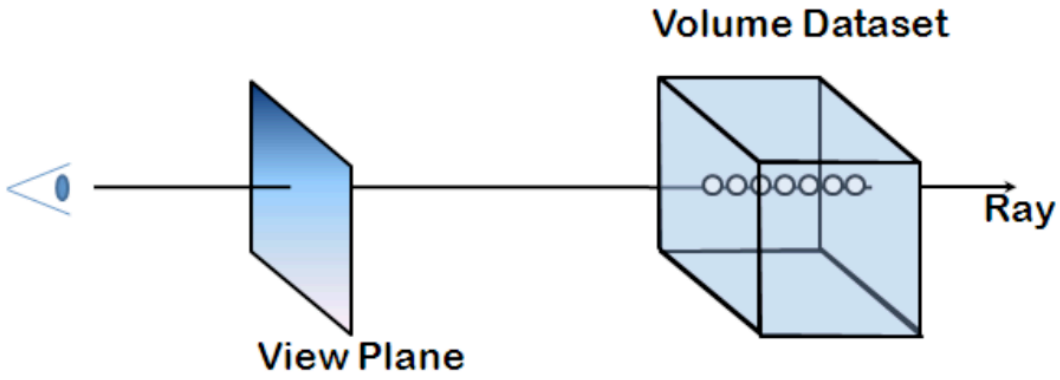
Color compositing

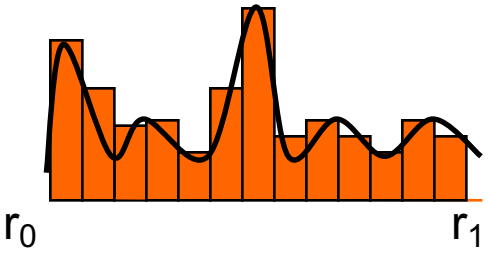




$$\int_{r0}^{r1} T(x)dx$$

Color compositing

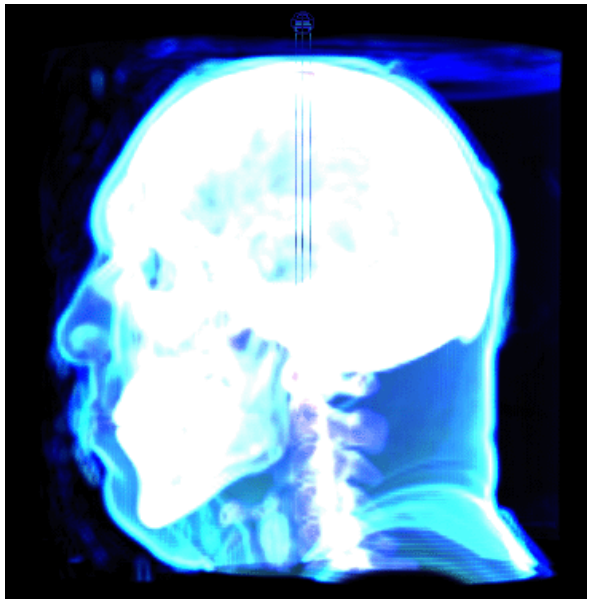




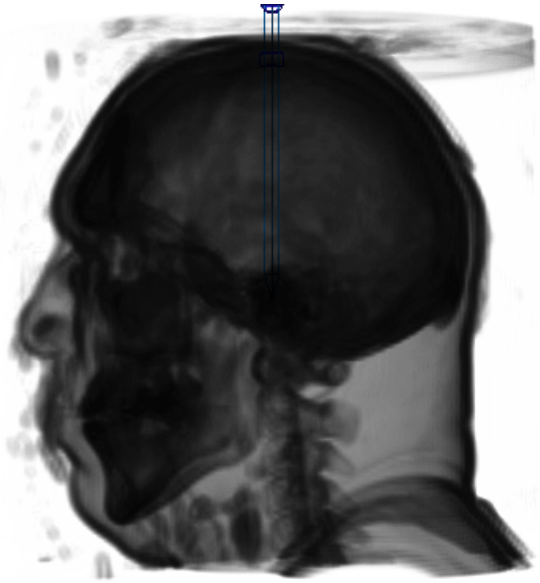
$$\int_{r0}^{r1} T(x)dx \rightarrow \sum_{i=0}^{n} T(x_i)\Delta x$$

Color compositing

Emission model

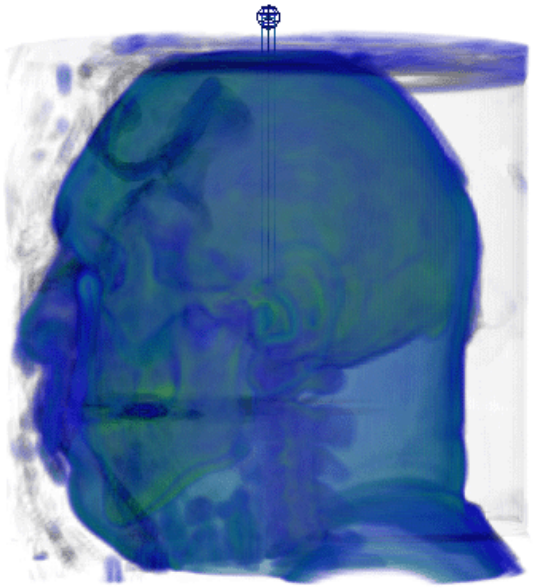


Absorption model



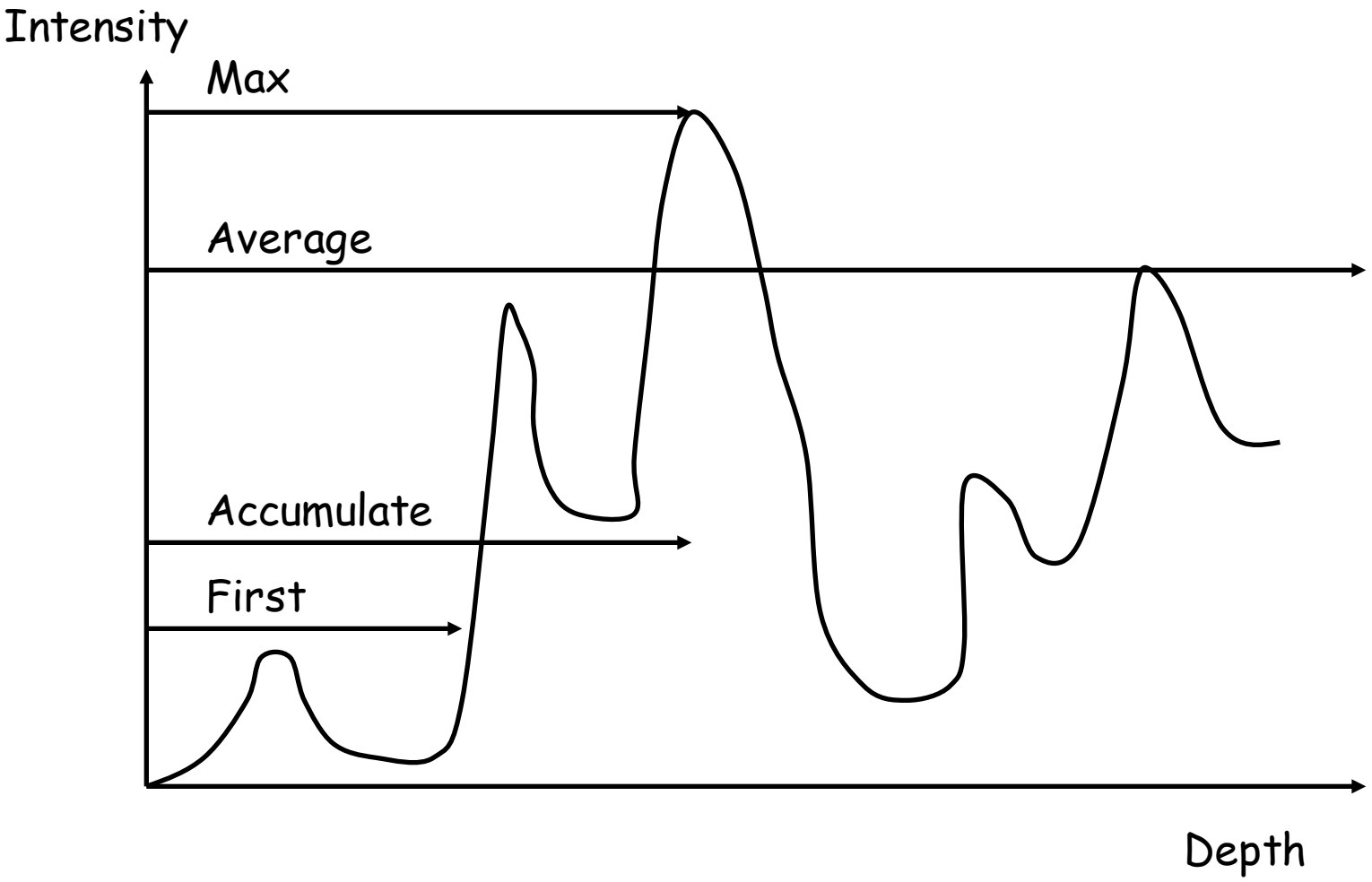
Color compositing

Emission and Absorption model

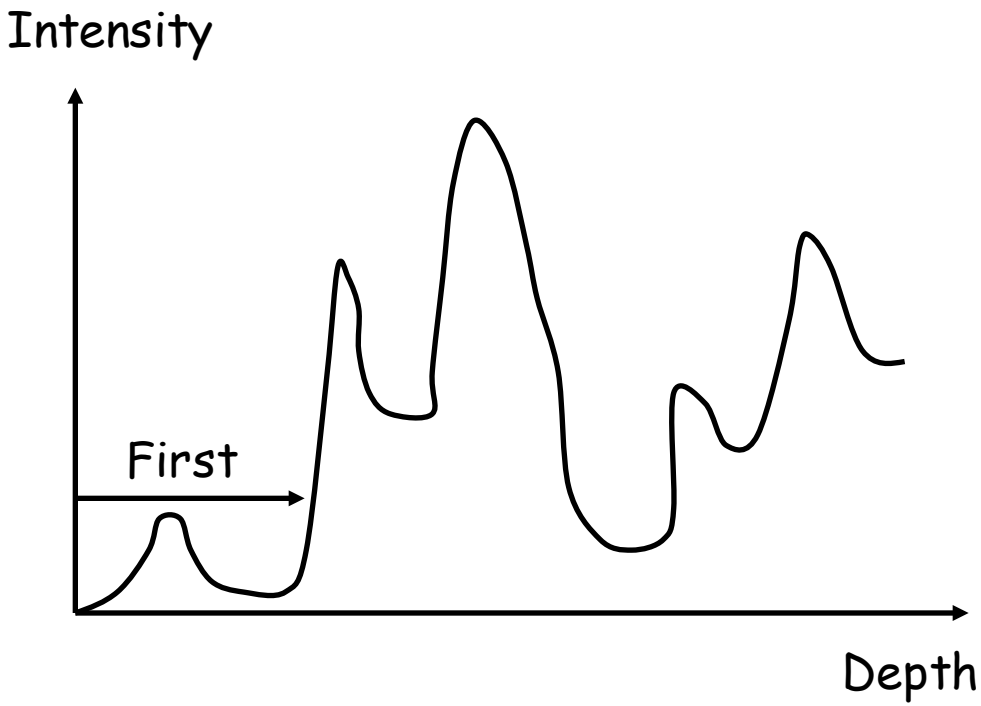


$$
\begin{aligned}
C_{\mathrm{Acc}} &= (1.0 - \alpha_{\mathrm{Acc}}) * C_{\mathrm{sample}} + C_{\mathrm{Acc}} \\
\alpha_{\mathrm{Acc}} &= (1.0 - \alpha_{\mathrm{Acc}}) * \alpha_{\mathrm{sample}} + \alpha_{\mathrm{Acc}}
\end{aligned}
$$

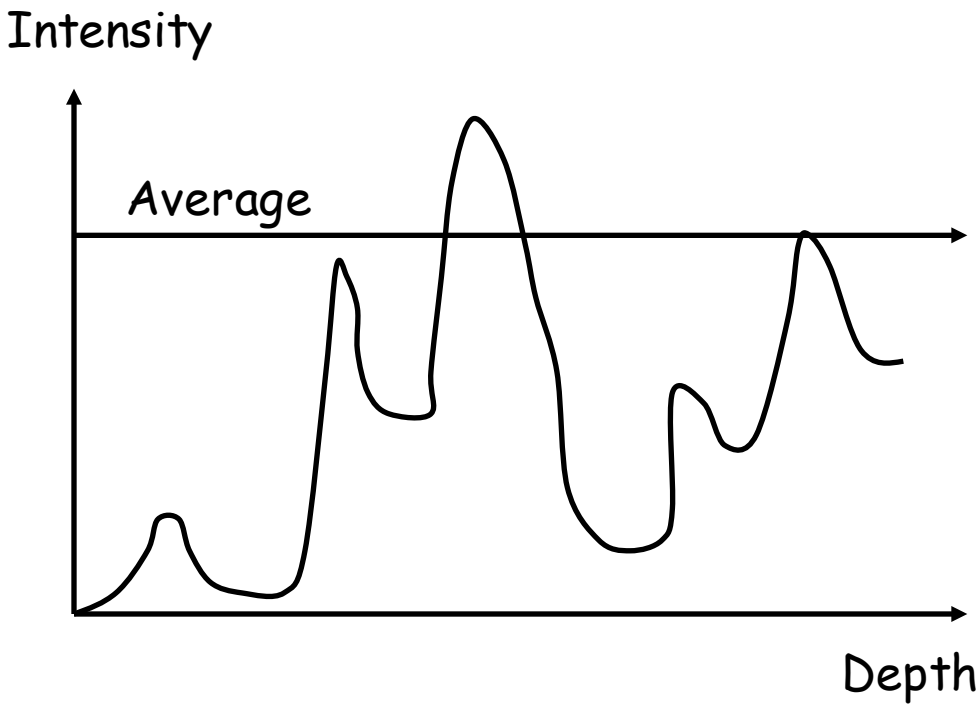
Ray Traversal Scheme

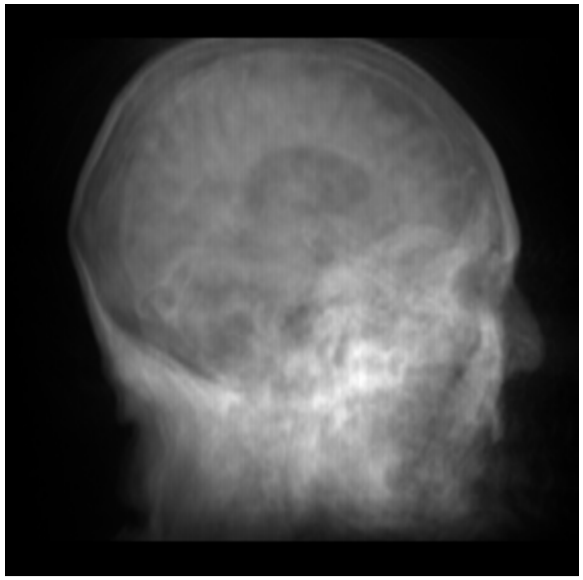


Ray Traversal (First)

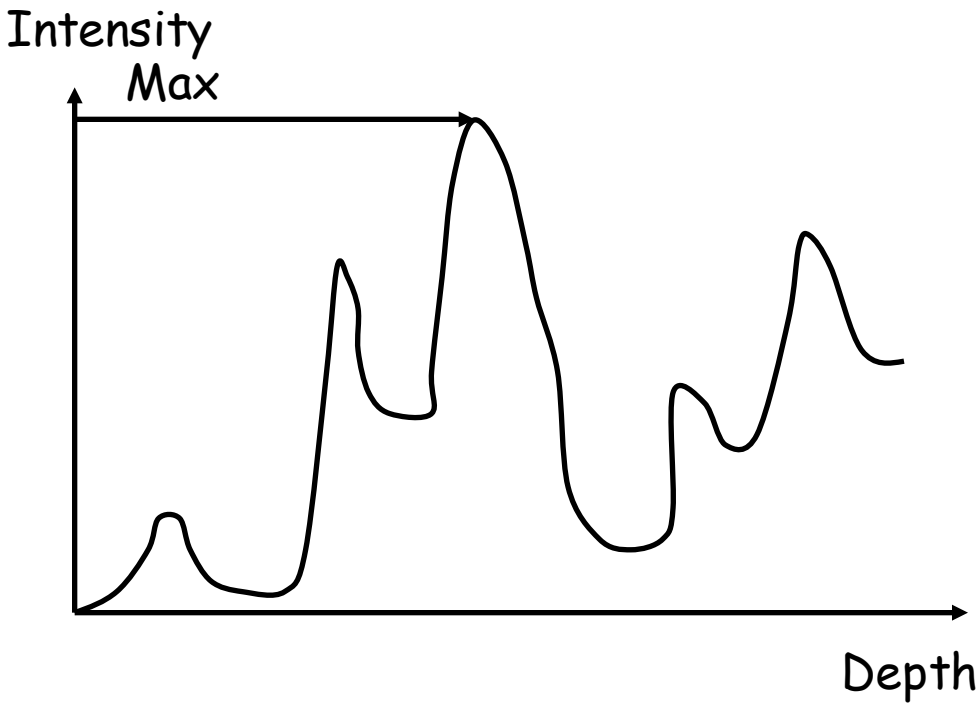


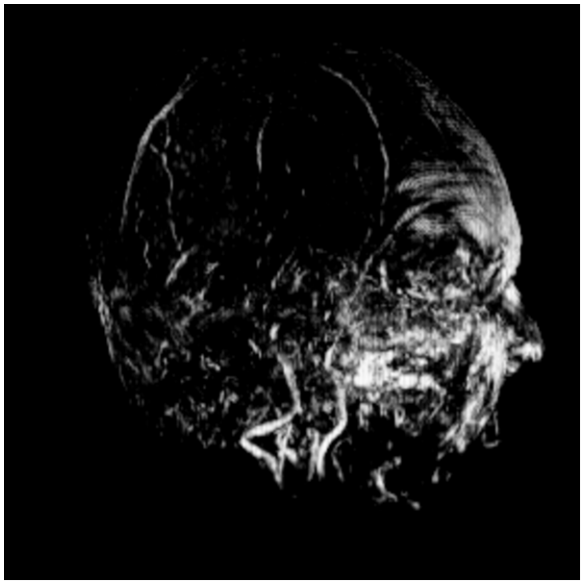
Ray Traversal (Average)



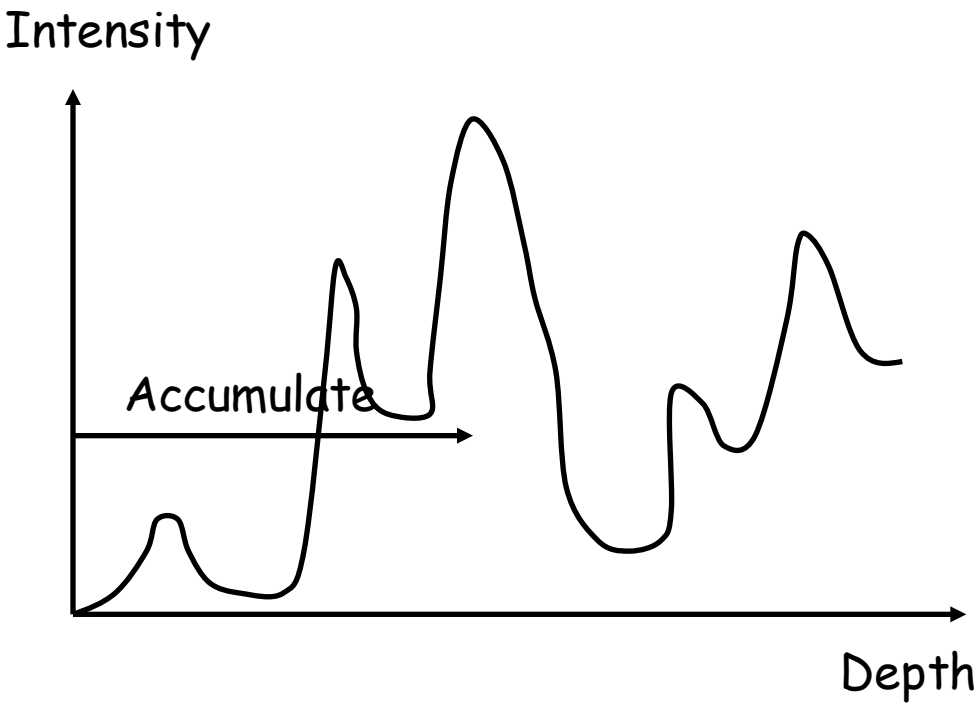


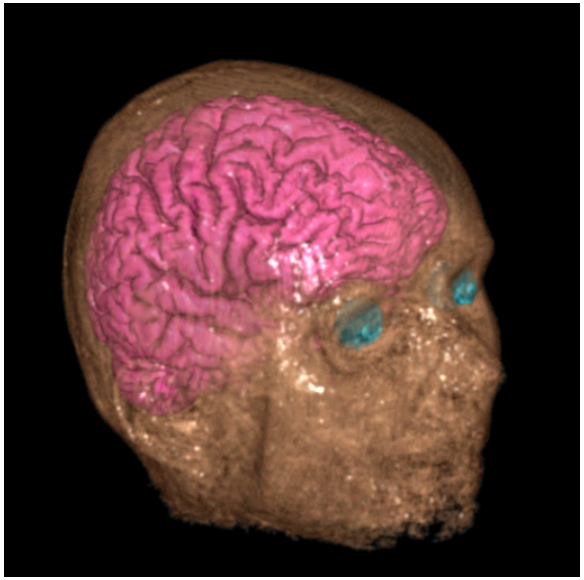
Ray Traversal (MIP)



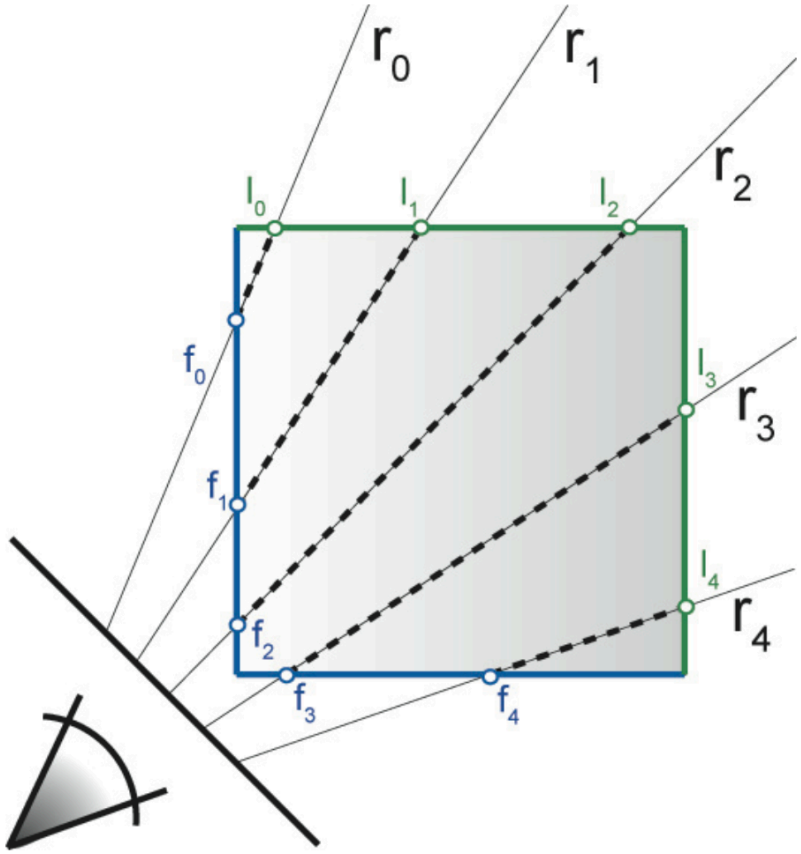


Ray Traversal (Accumulate)

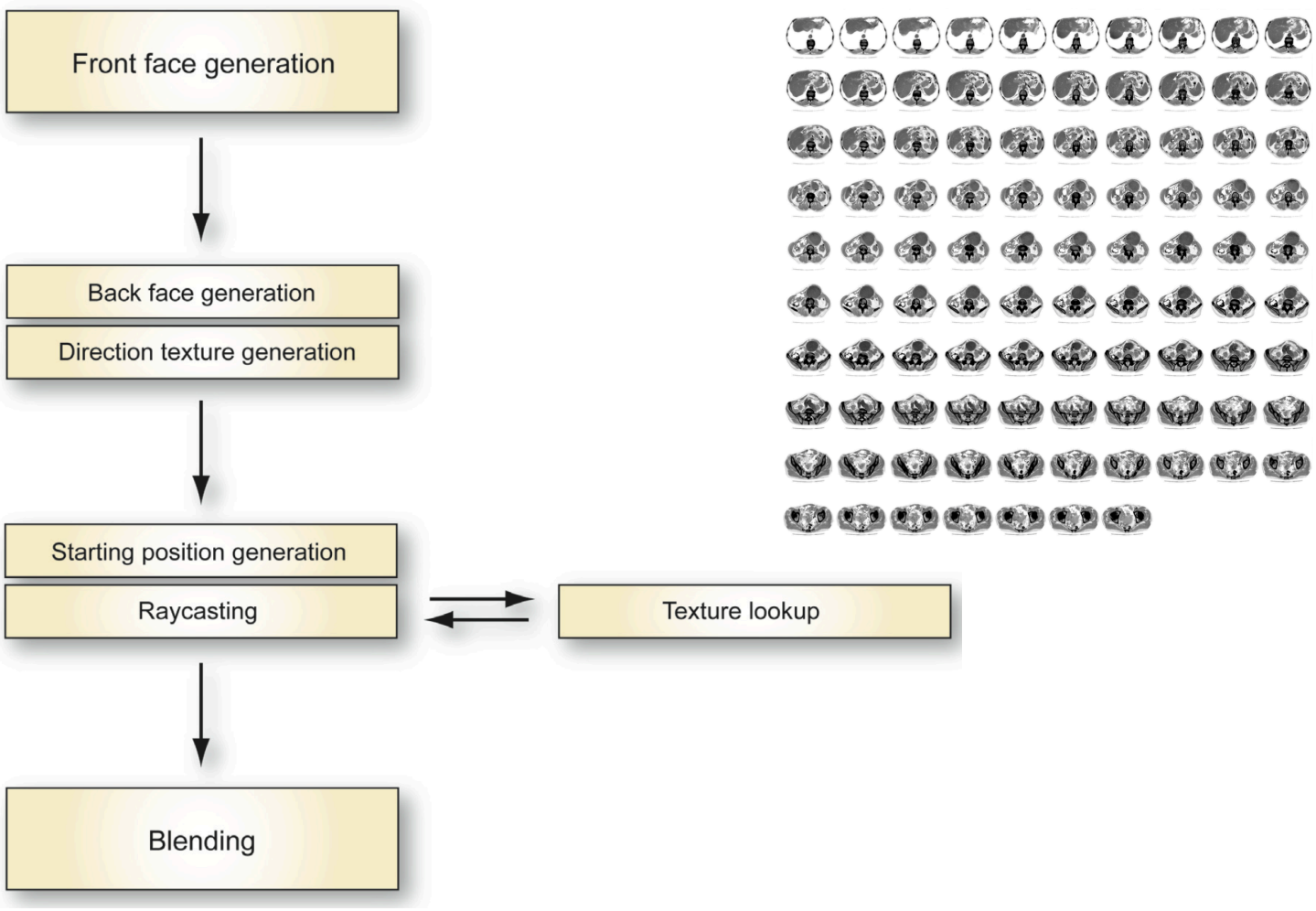




GPU Implementation

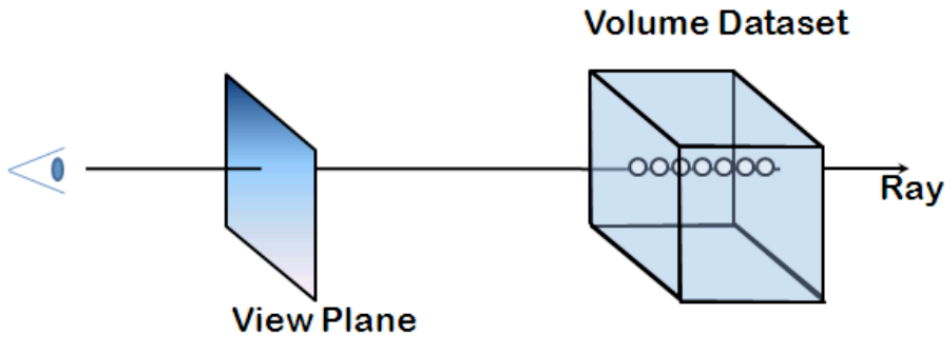


Rendering pipeline (GPU)



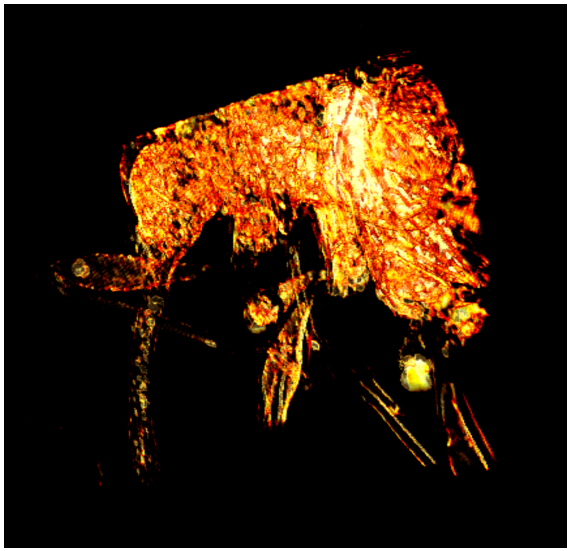
Accelerating Ray Casting

- Empty Space Skipping
- Early Ray Termination

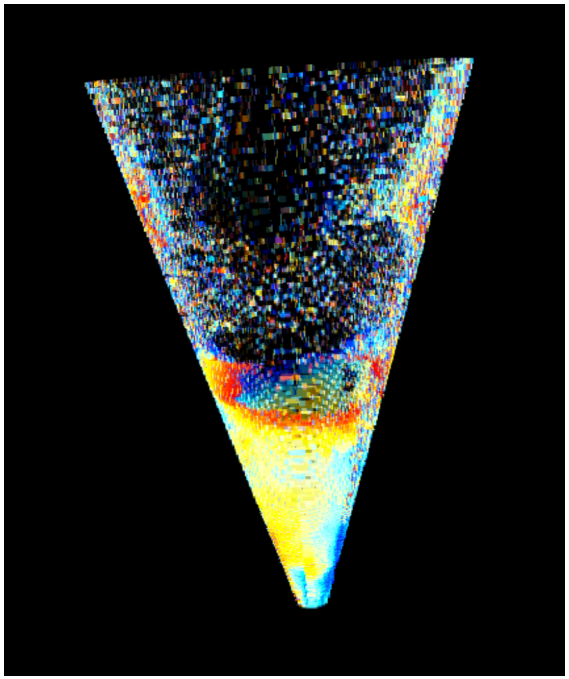


Demo

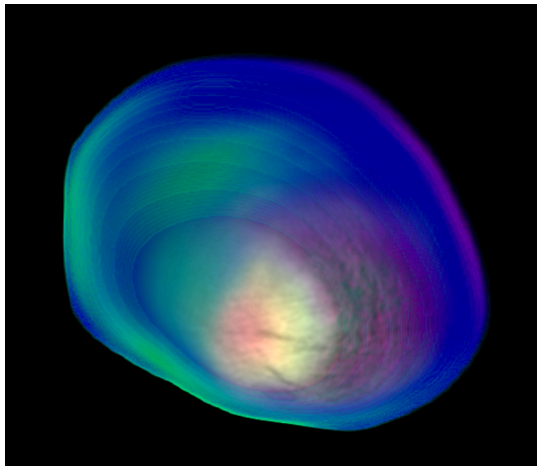
[ASTOR] Segmented Bee



[KITcube] Lidar Scan



[USCT] Breast



Karlsruher Institut für Technologie

Thanks for your attention.

References

- Nelson Max, **Optical Models for Direct Volume Rendering**, *IEEE Visualization and Computer Graphics* (v1 #2 June 1995)
- Mark Levoy, **Display of Surfaces from Volume Data**, *IEEE Computer Graphics and Applications* (v8 #3 May 1988)
- Kenneth R. Castleman, **Digital Image Processing**, *Prentice Hall* (c1996 ISBN:0-13-211467-4)
- Markus Hadwiger et al. , **Advanced Illumination Techniques for GPU-Based Volume Raycasting**

Repository

- Tomoraycaster
  - https://github.com/kit-ipe/tomo_raycaster2.git
- USCT Web Interface Framework
  - https://github.com/kit-ipe/usct-vis.git