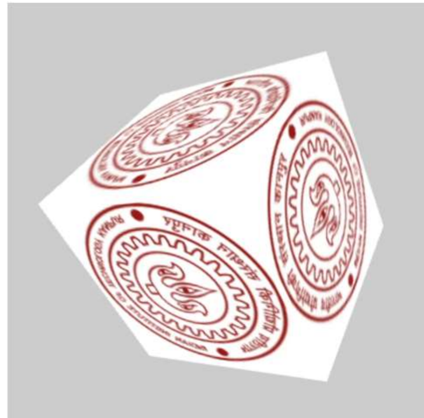


Introduction to Computer Graphics (CS360A)

Instructor: Soumya Dutta

Department of Computer Science and Engineering

Indian Institute of Technology Kanpur (IITK)

email: soumyad@cse.iitk.ac.in

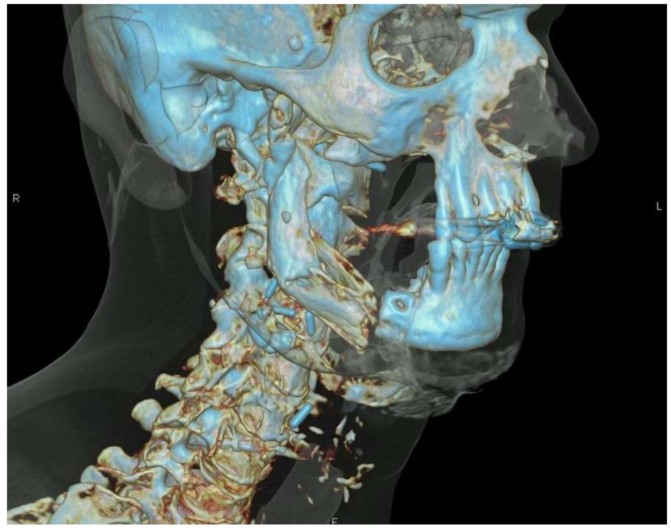
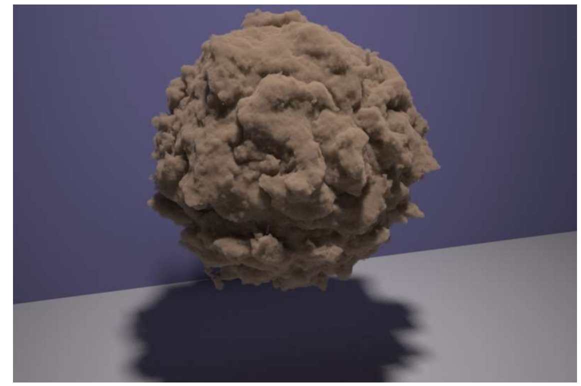
Acknowledgements

IITK CS360A

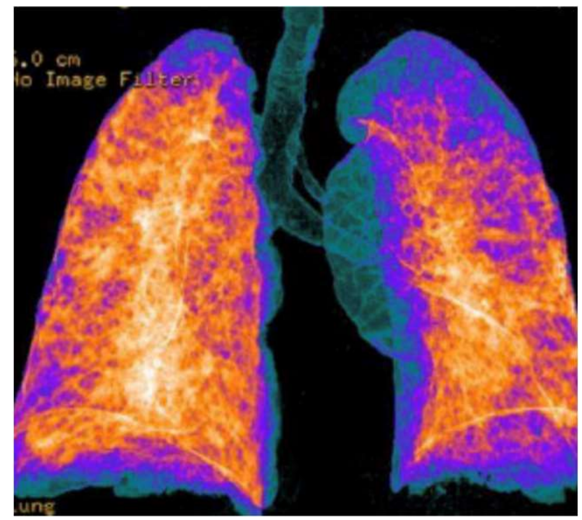
- A subset of the slides that I will present throughout the course are adapted/inspired by excellent courses on Computer Graphics offered by Prof. Han-Wei Shen, Prof. Wojciech Matusik, Prof. Frédo Durand, Prof. Abe Davis**,** Klaus Mueller, and Prof. Cem Yuksel
- Engel, Hadwiger, Salama; Real time volume graphics tutorial, EuroGraphics 2006

3D Scientific Visualization

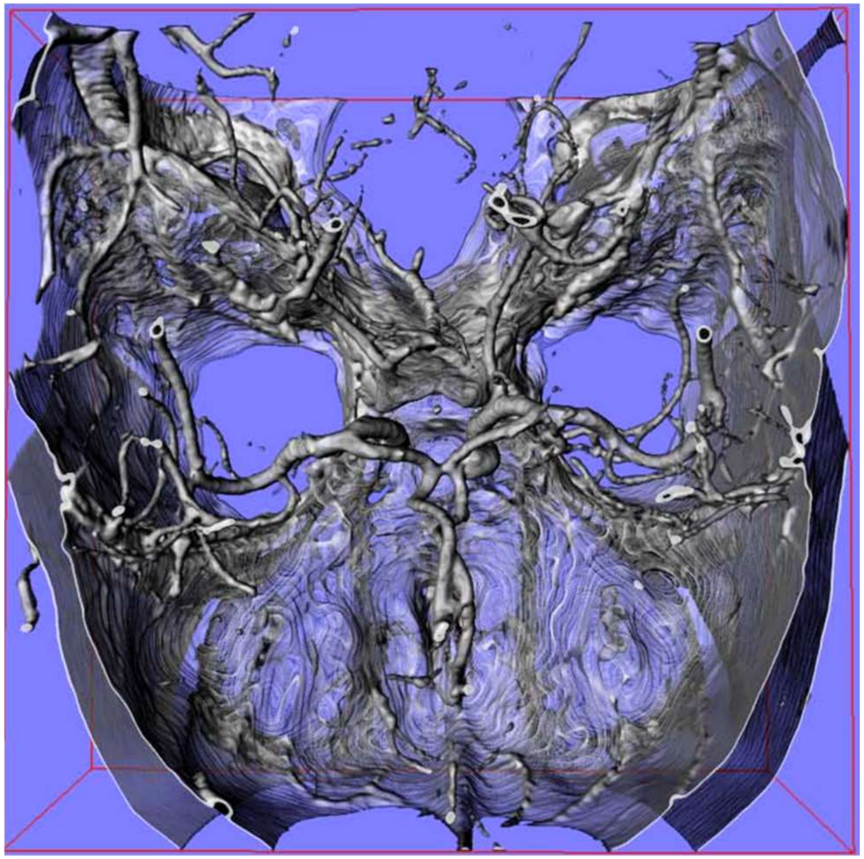


Volume Rendering

Applications: Medical Science

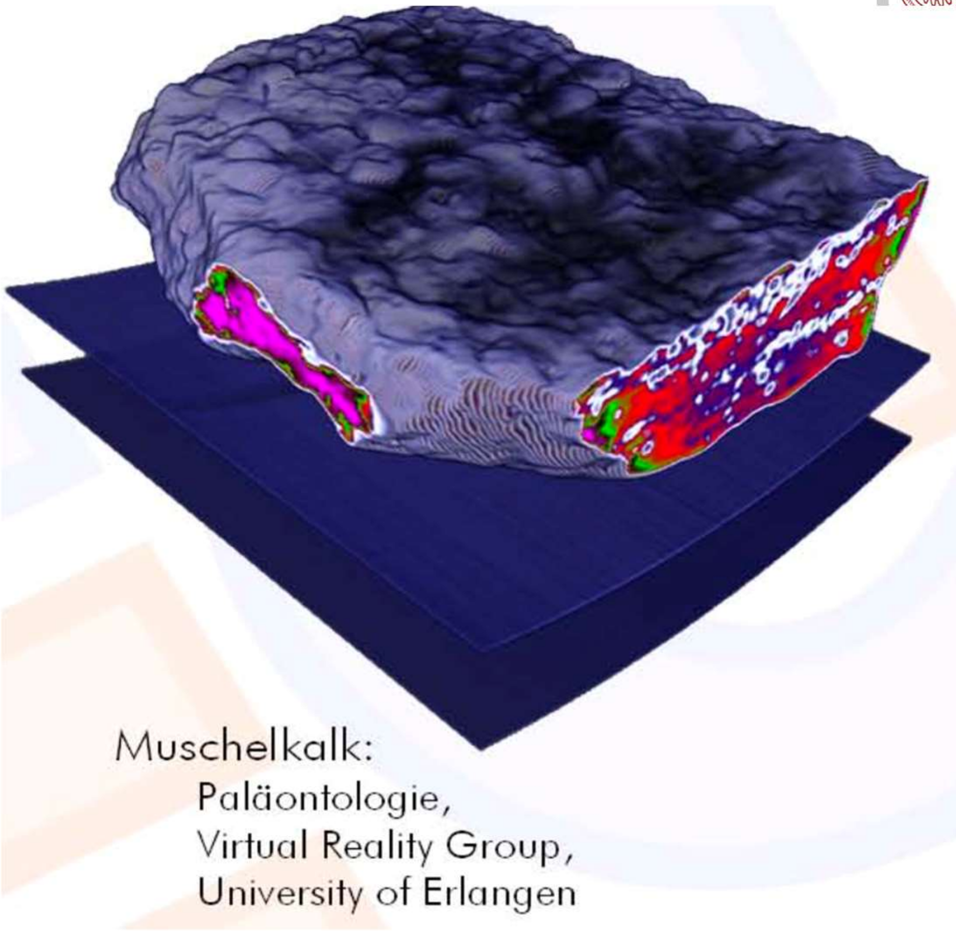


CT Angiography

Applications: Medical Science

Applications: Geology

Deformed plastic model

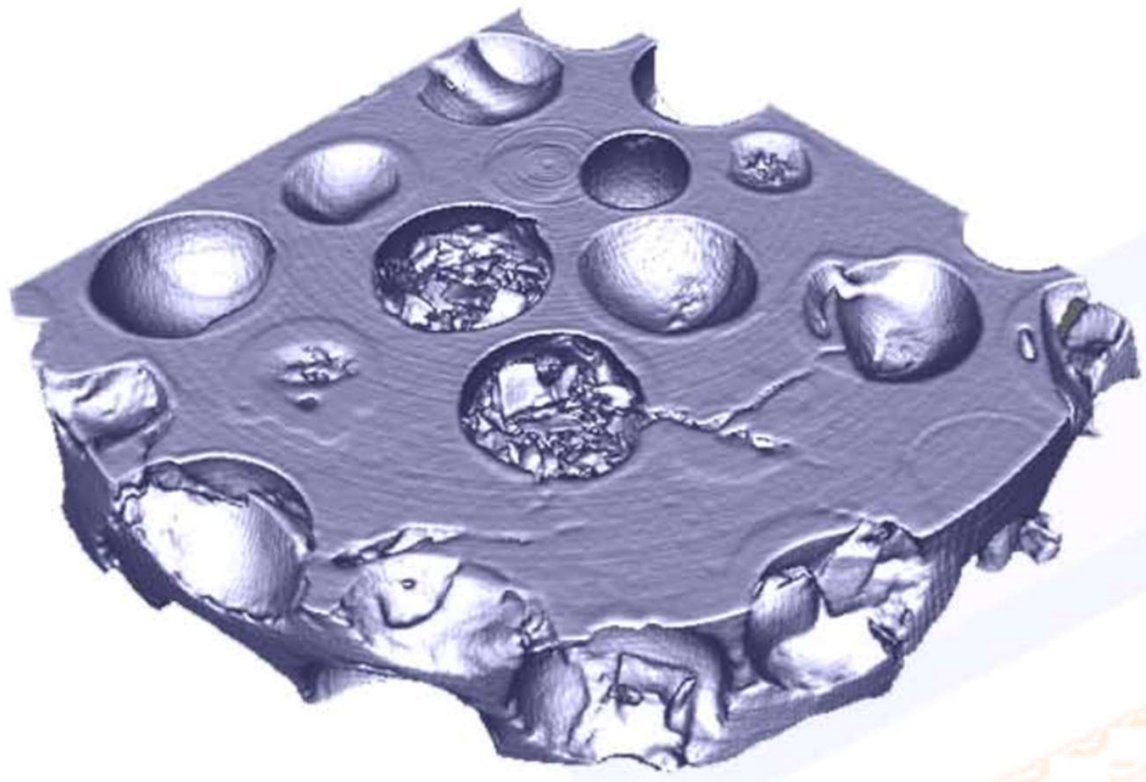


Applications: Archeology

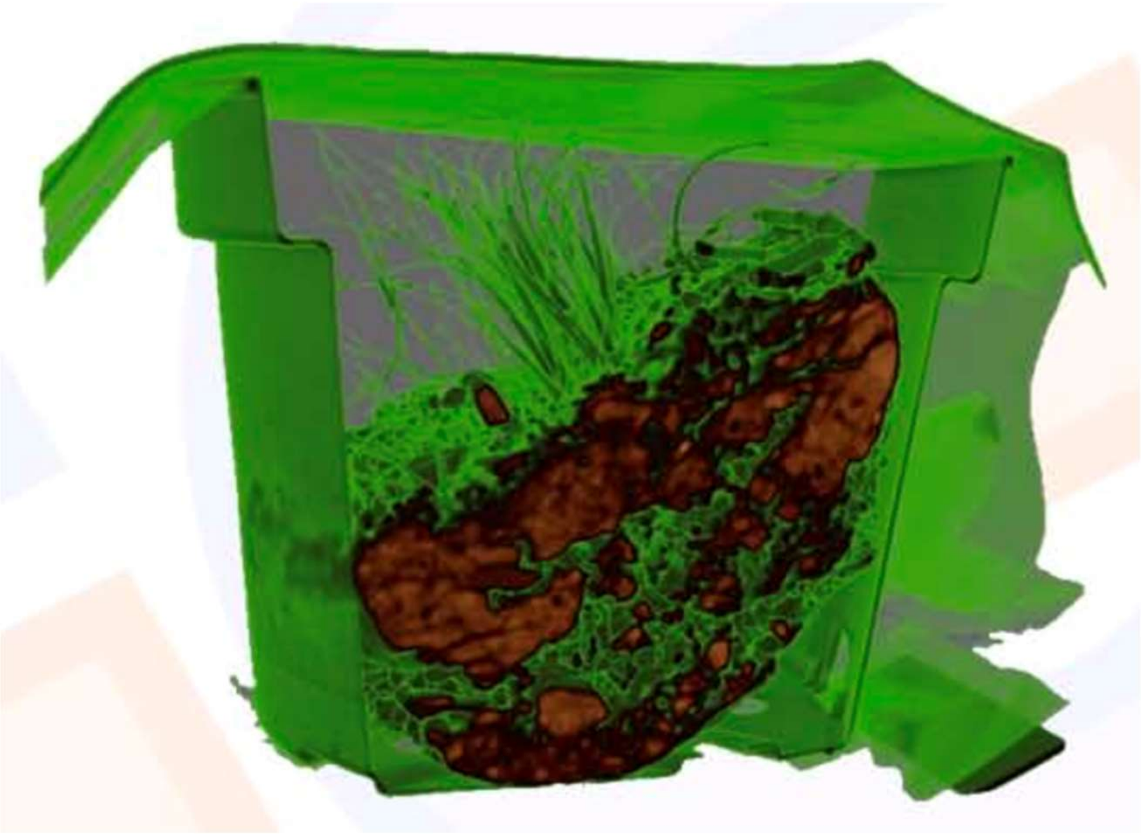
Historical Statute

Applications: Materials Science

Quality control



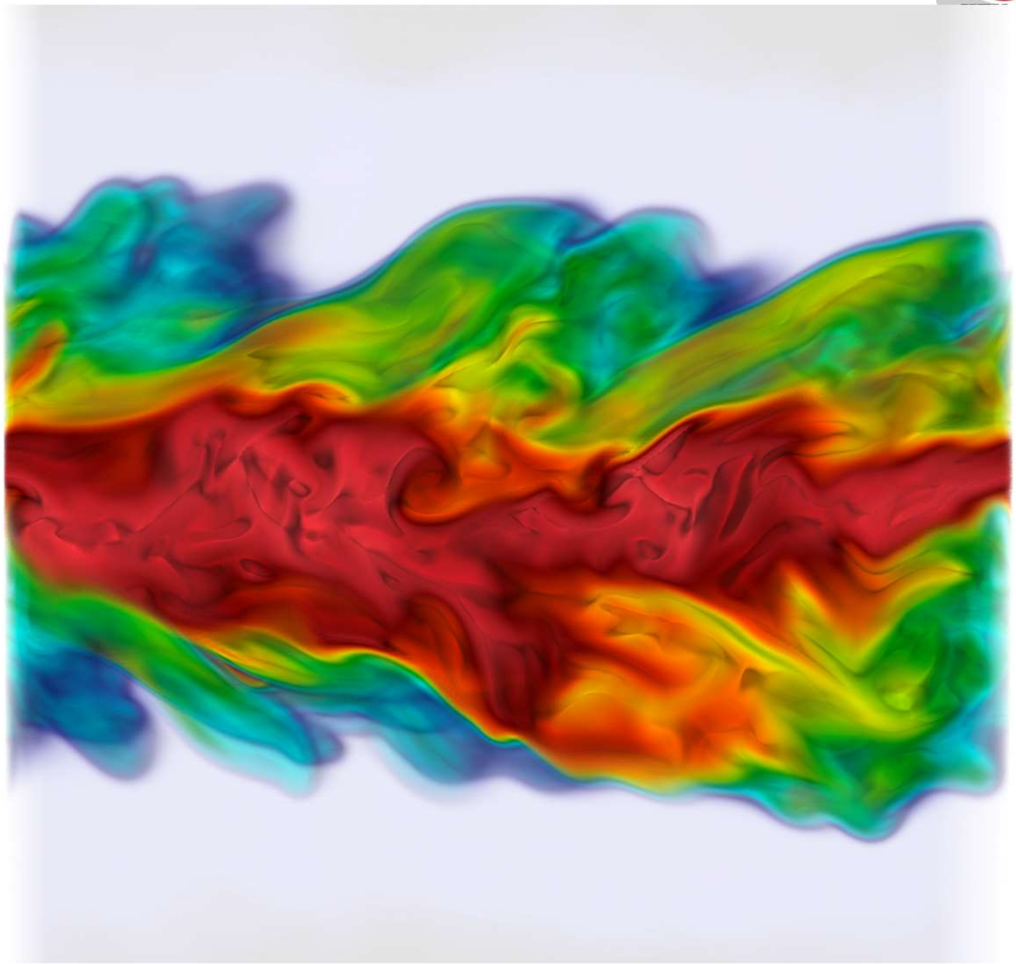
Applications: Biology



Biological soil samples

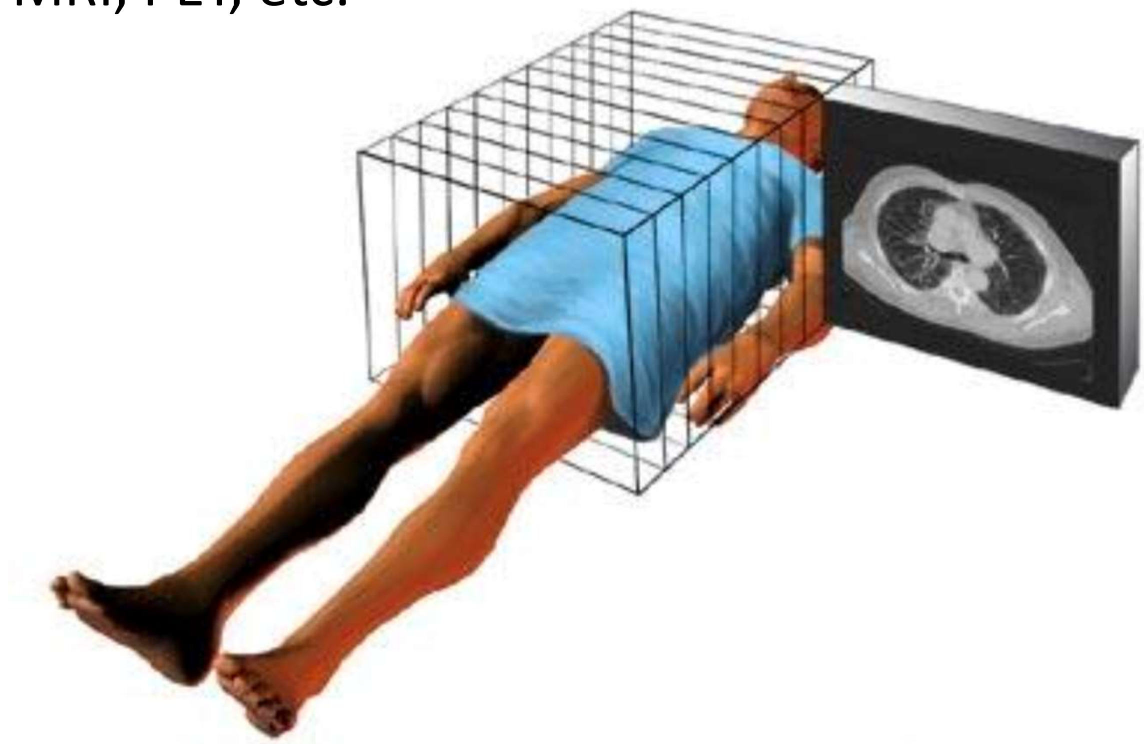
Applications: Computational Sciences

Study Combustion process



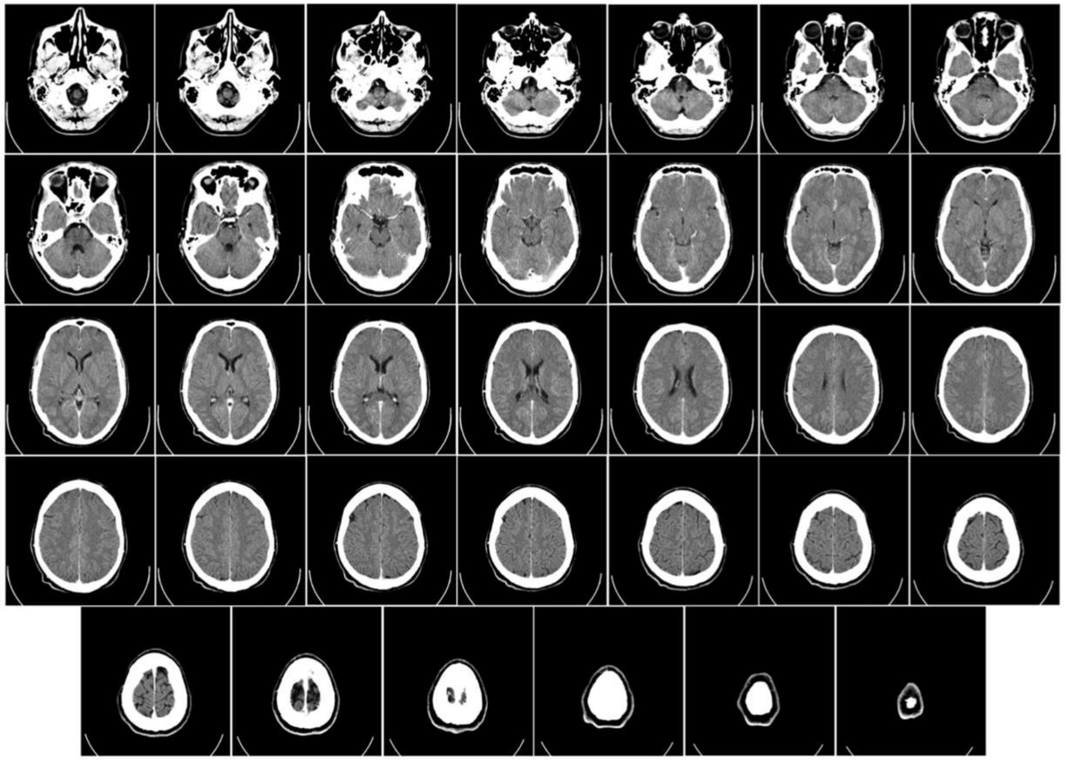
How is Volume Data Generated?

- Often obtained by scanning
  - X-ray, CT, MRI, PET, etc.



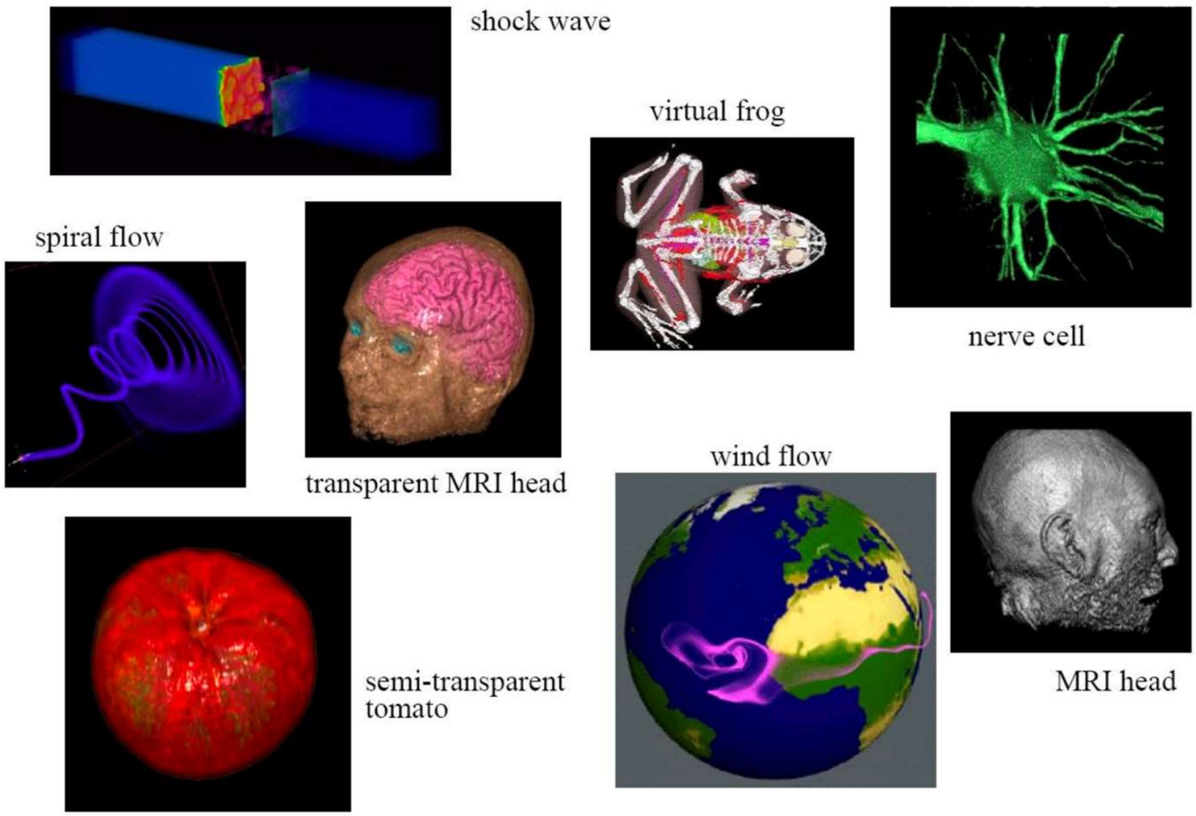
How is Volume Data Generated?

- Often obtained by scanning
  - X-ray, CT, MRI, PET, etc.



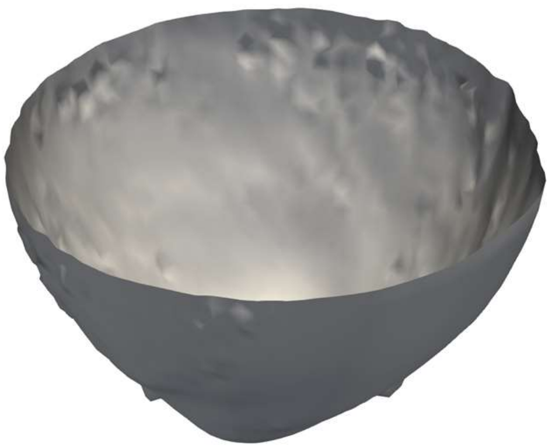
How is Volume Data Generated?

- Numerical Simulations in another large source of Volume data

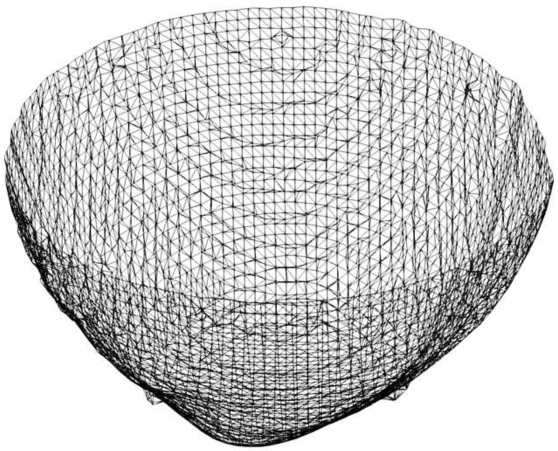


Indirect Visualization of Volume Data

- Isosurface based rendering for 3D data
  - Example of indirect technique for volume data exploration
  - Using geometric representations
    - Points, meshes, surfaces, etc.



Isosurface

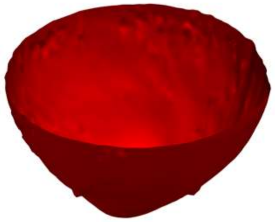


Geometric mesh of the isosurface

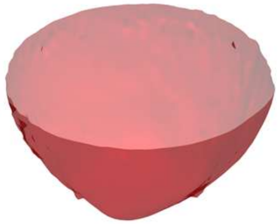
Volume Rendering

IITK CS360A

- Surface visualization is a way of showing data as opaque objects
  - Though you can apply transparency on surfaces
- Many applications demand techniques that allows <u>“see-through”</u> capability
  - Make parts of the data (semi)transparent so the data behind can be seen
- <u>Volume Rendering</u> technique is the answer!
  - Direct mapping of underlying 3D data into an image space
  - Assumes data as a <u>translucent gel</u> that allows light to go through



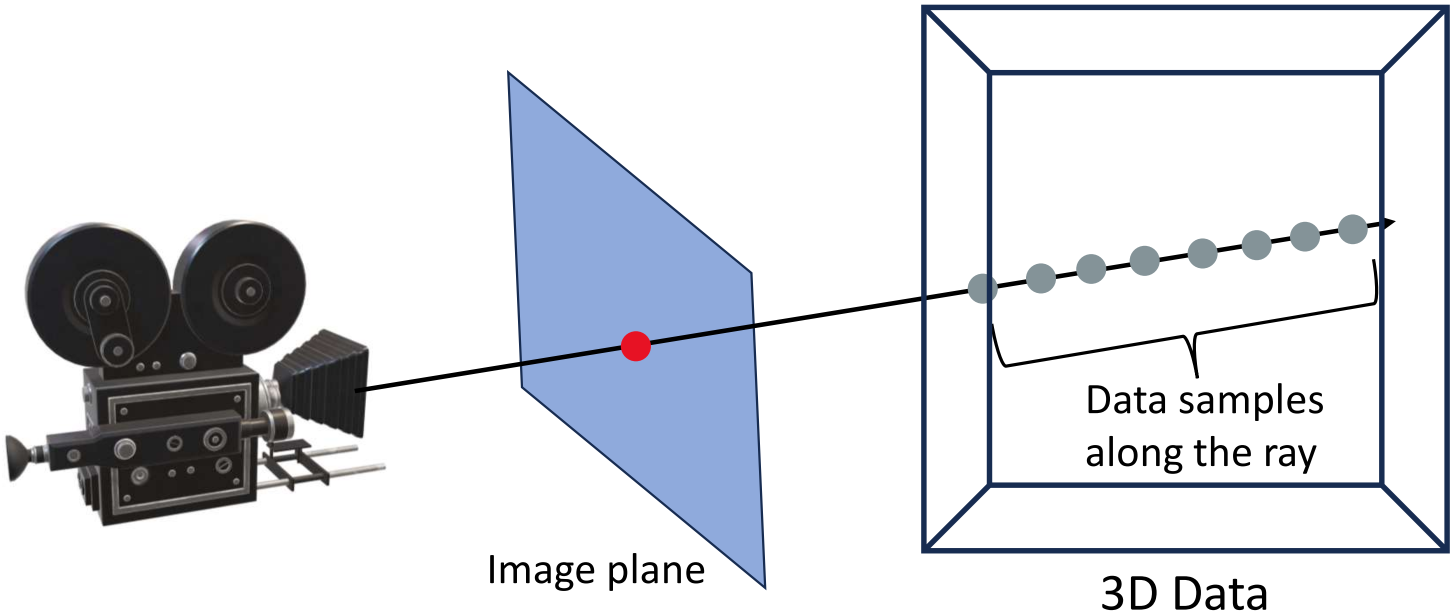
Opaque



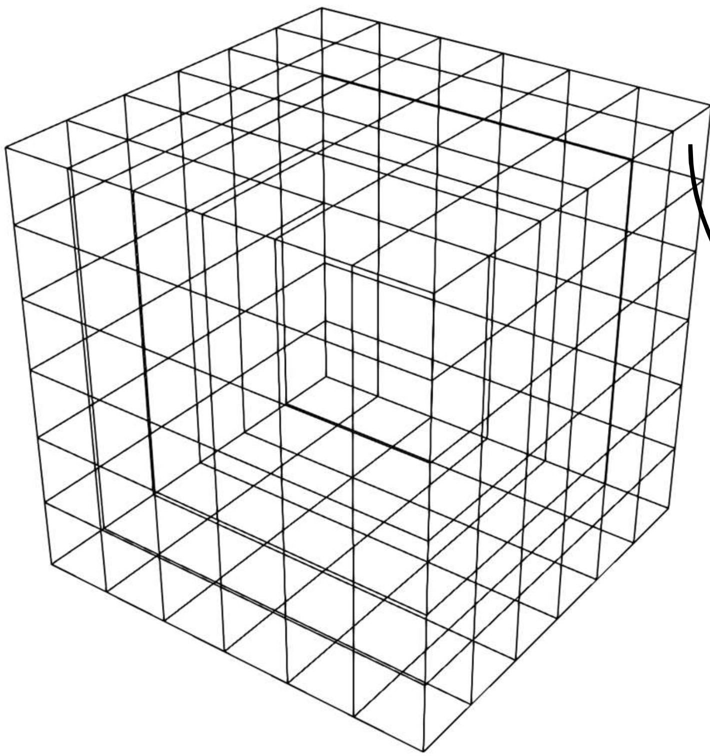
Semi-transparent

Volume Rendering Key Idea

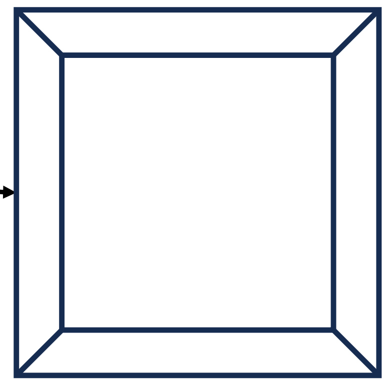
- Data is considered as a translucent gel
- Rays are cast into the volume data through each pixel to observe data values
- Rays accumulate color and opacity values along the ray for final pixel color



Sampling via Trilinear Interpolation

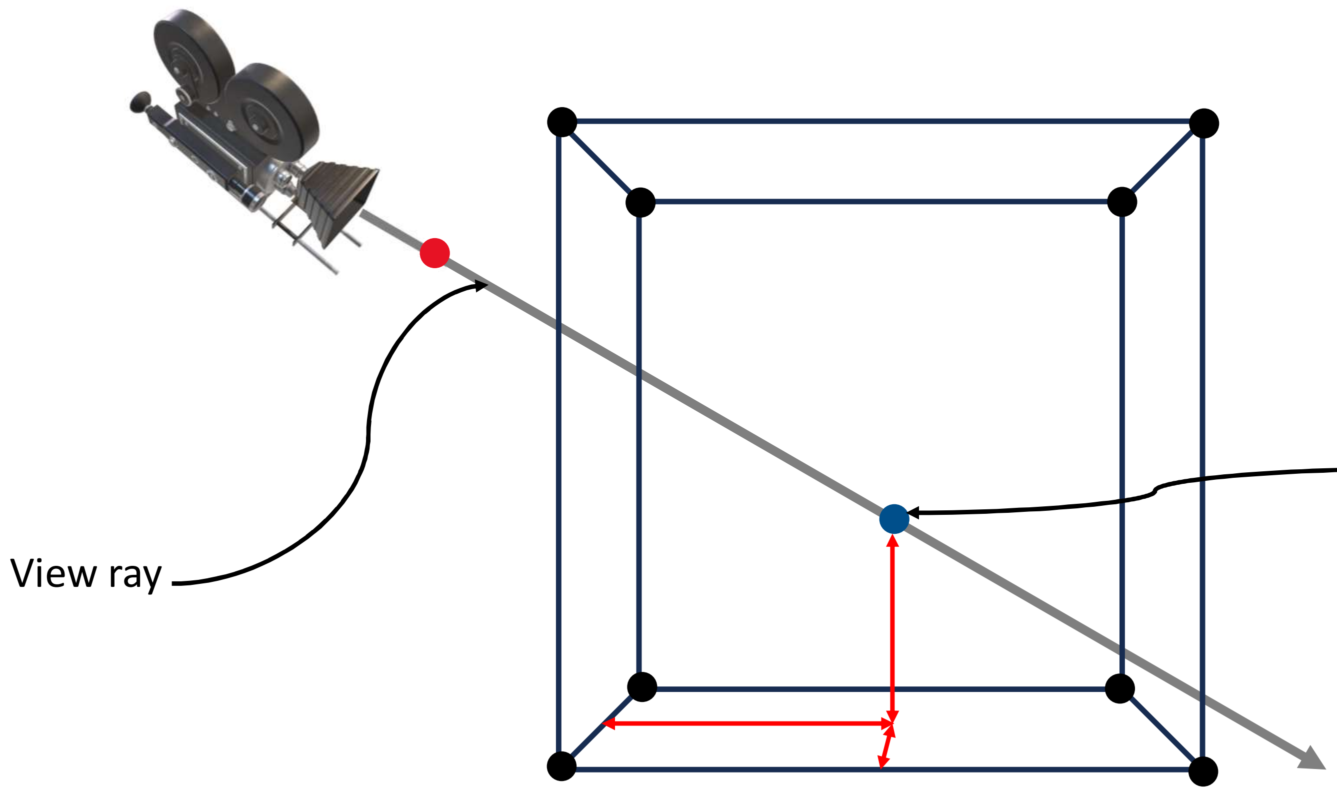


Grid of the data set



A single cube cell of the data

Sampling via Trilinear Interpolation



A single cube cell of the data

Different Volume Rendering Algorithms

- Image-order techniques
  - Ray casting approaches
- Object-order techniques
  - Splatting
  - Texture mapping

Different Volume Rendering Algorithms

- **Image-order techniques**
  - **Ray casting approaches**
- Object-order techniques
  - Splatting
  - Texture mapping

Image-Order Volume Rendering Techniques

- Typically known as Ray casting methods
- Given an image plane, for each pixel, we compute the color by casting a ray through each pixel to the data
- We evaluate the data along the ray using some pre-specified functions for computing the final pixel color

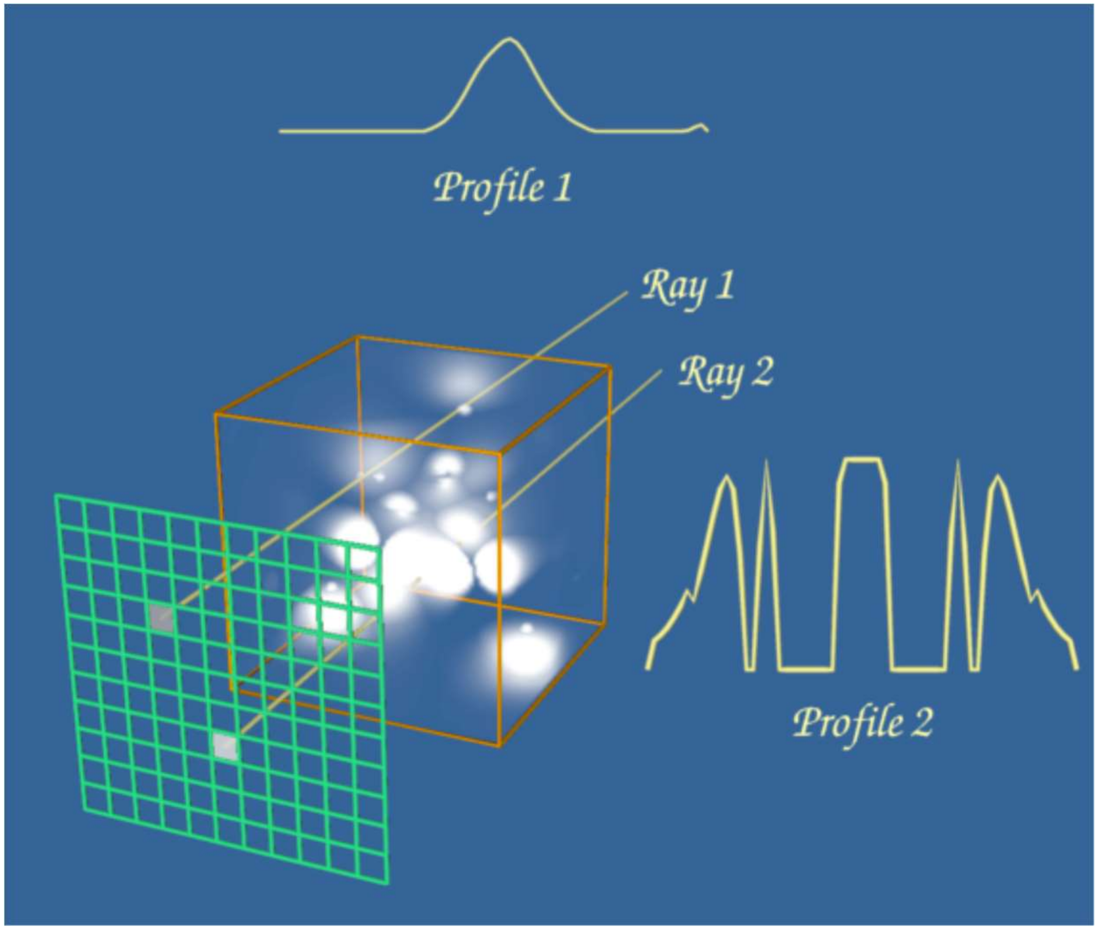
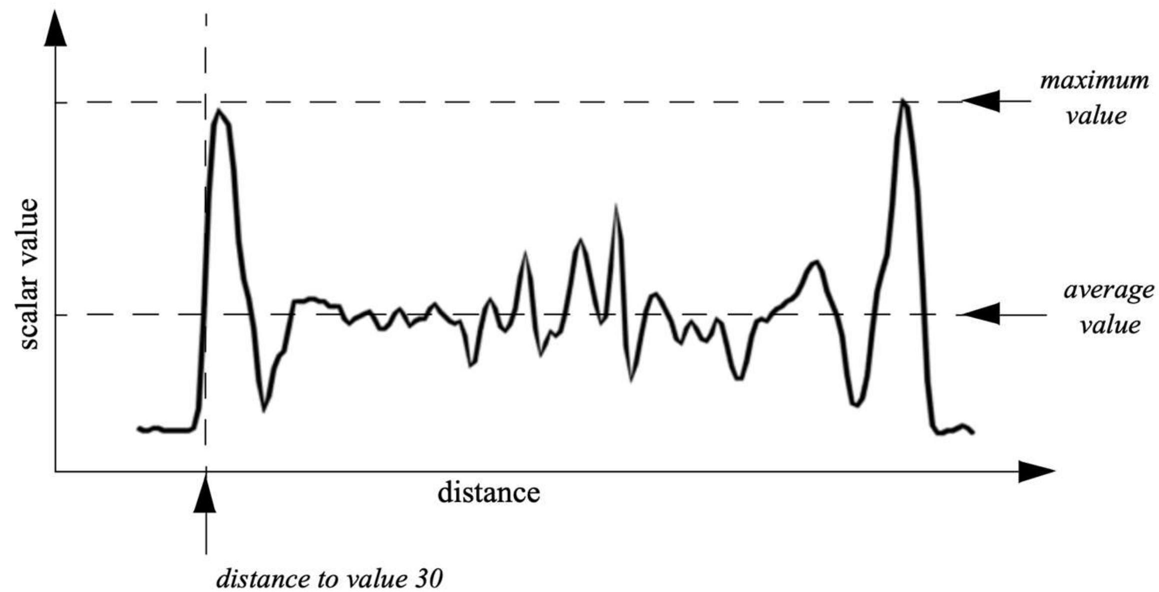


Image-Order Volume Rendering Techniques

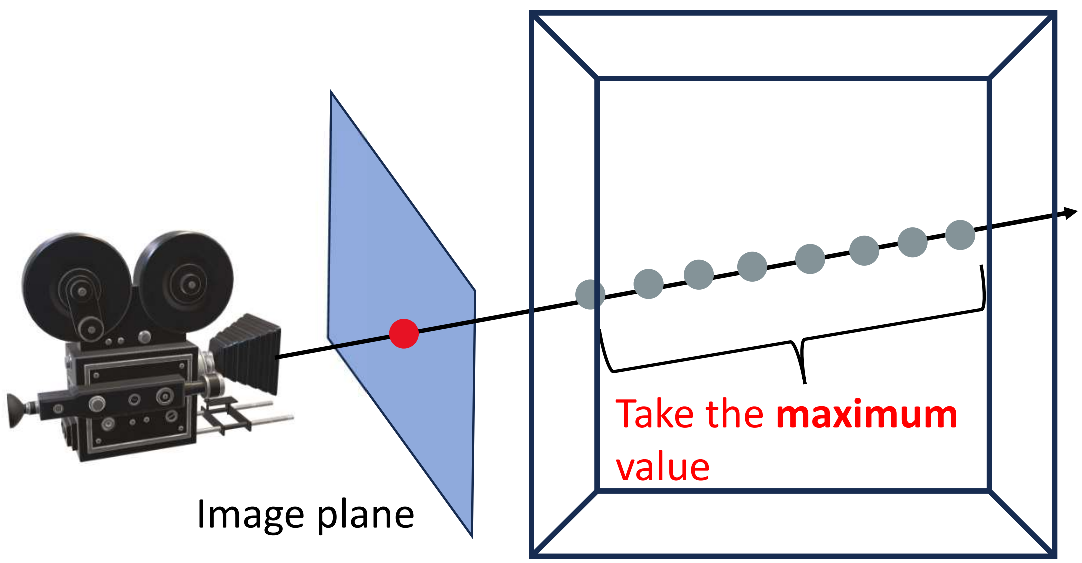
- How do we accumulate the data values along the ray to produce a final pixel color?

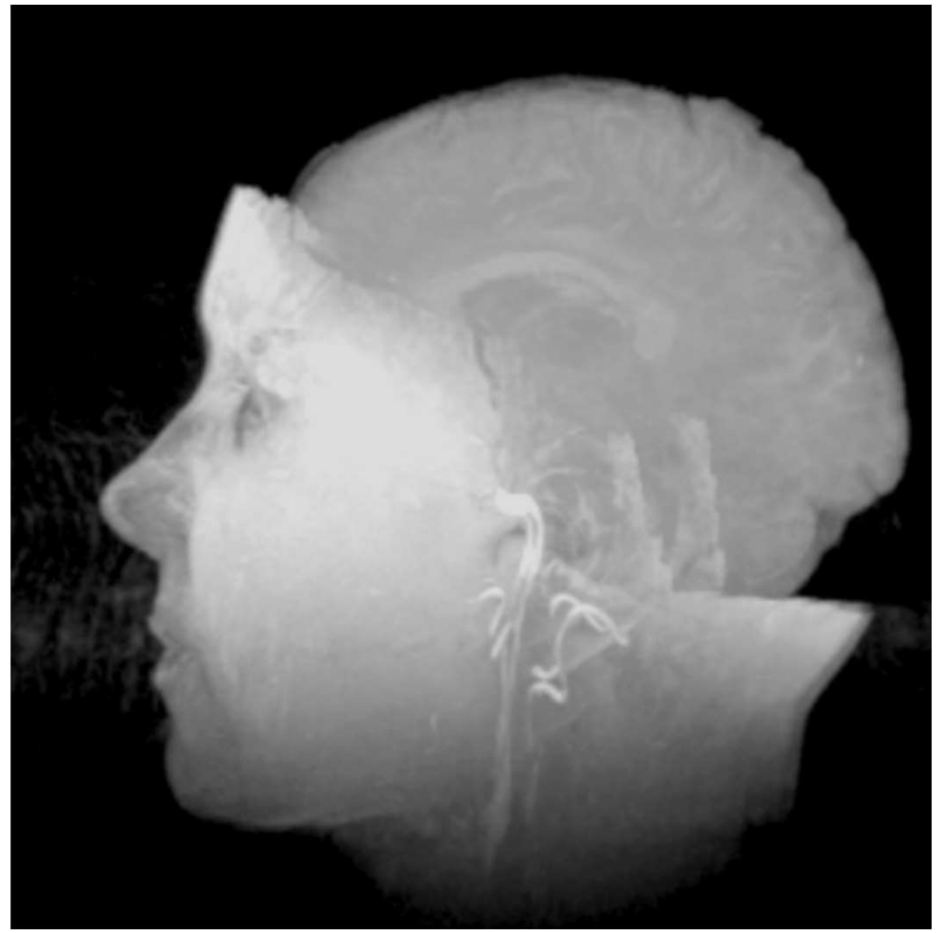
1. Maximum Intensity Projection (MIP)
2. Average Value
3. Distance to a value
4. Composite



1. Maximum Intensity Projection

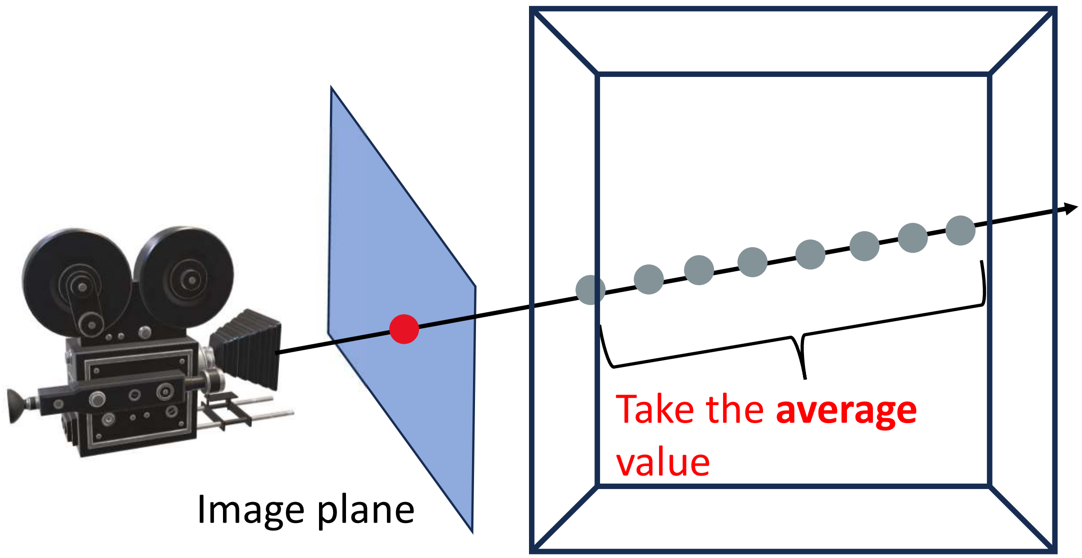
- Compute the **maximum** intensity (data) value along each casted ray and the map the value to a color using a color scale

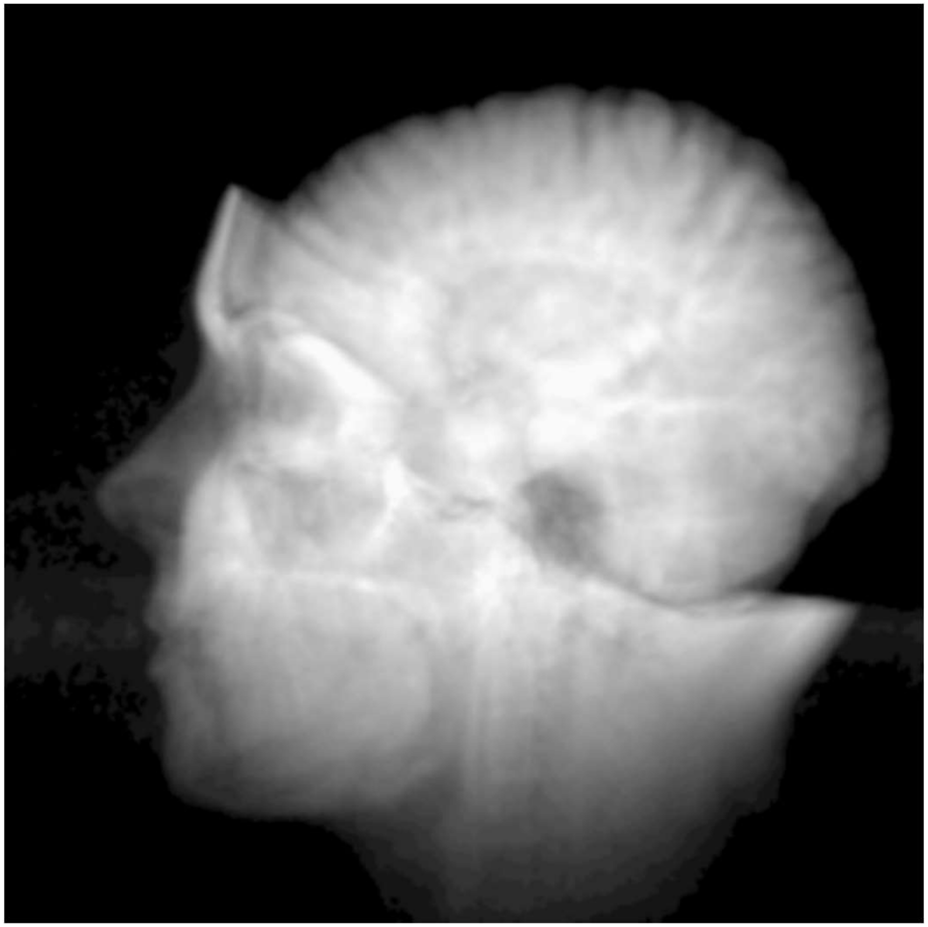




2. Average Intensity Projection

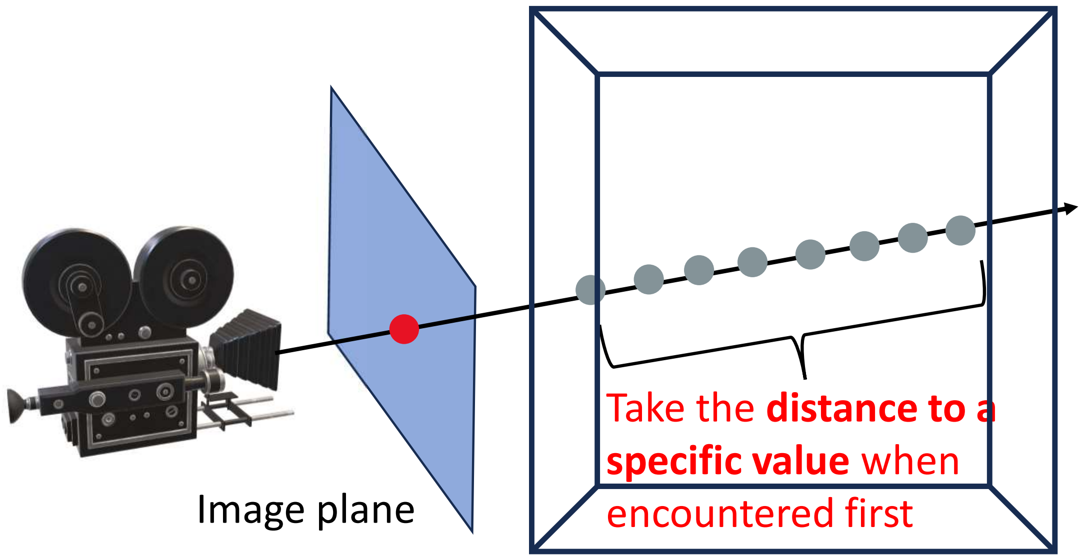
- Compute the **average** intensity (data) value along each casted ray and the map the value to a color using a color scale

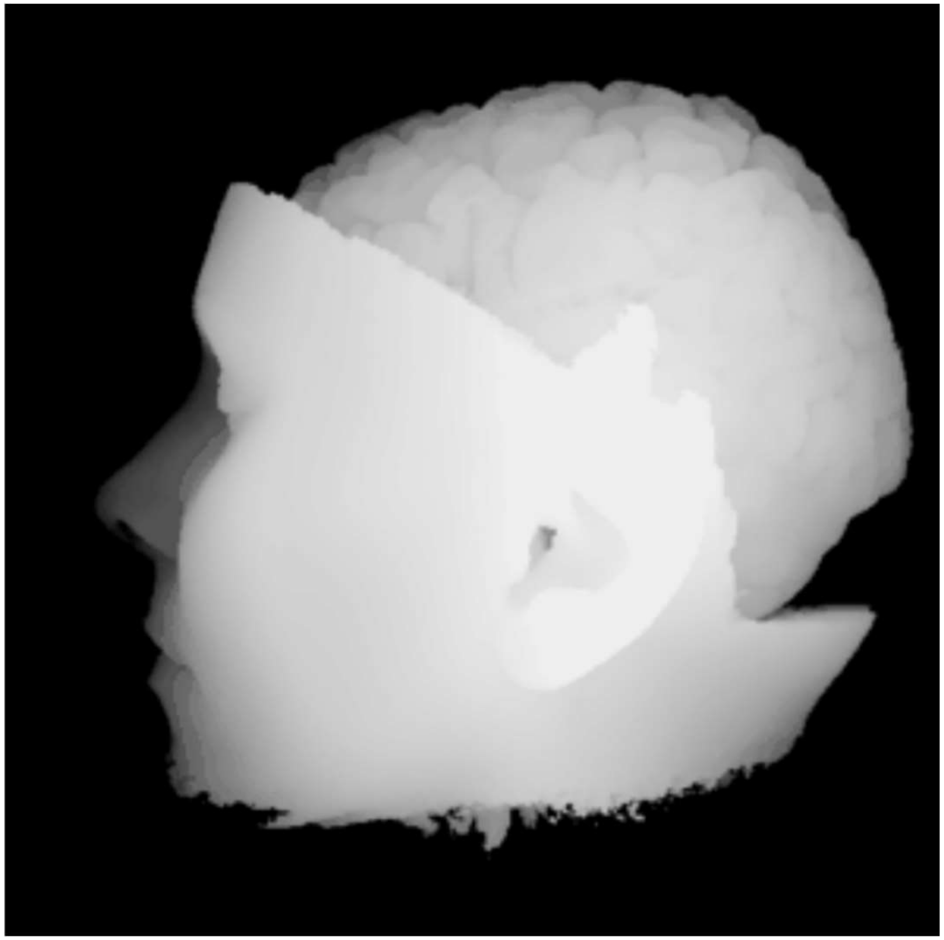




3. Distance to a Value Projection

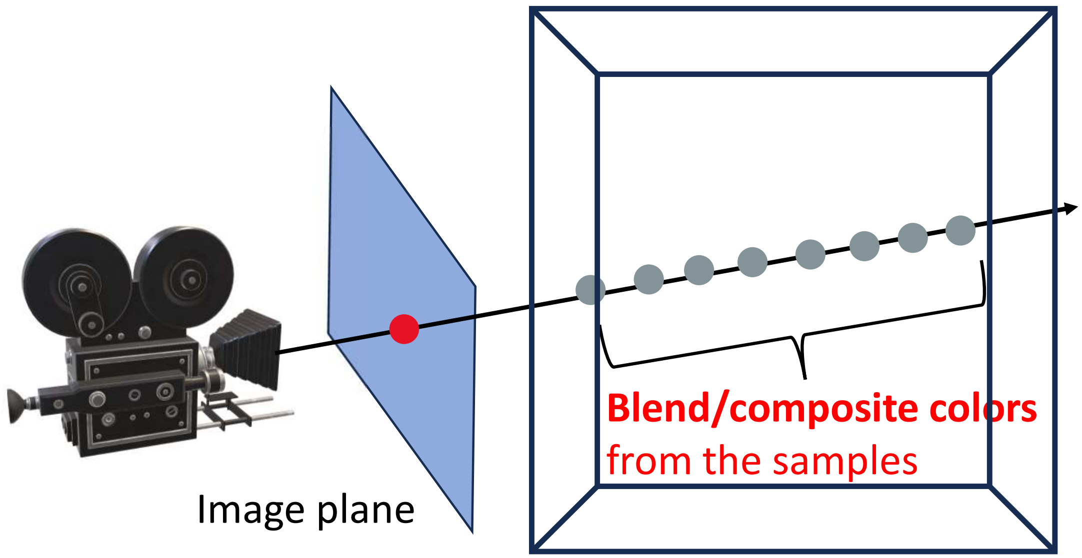
- Distance to value 30 is shown in the rendered image
  - Provides the notion of the depth as to where the data value 30 is encountered

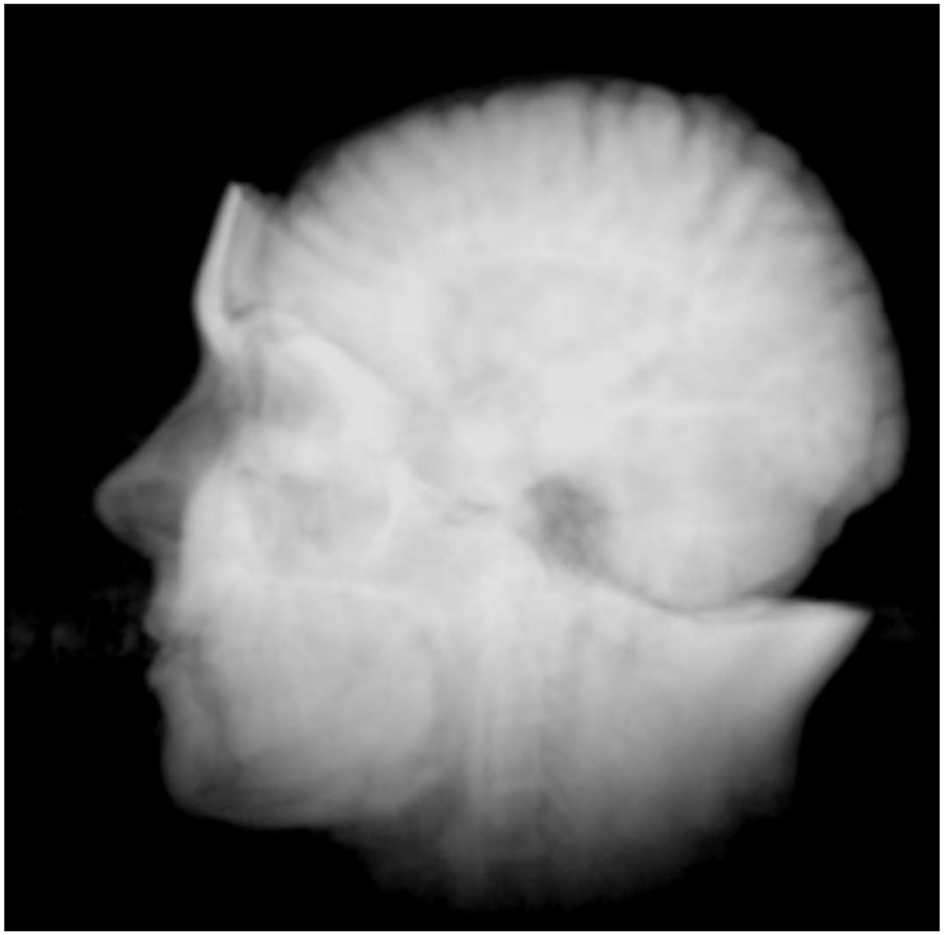




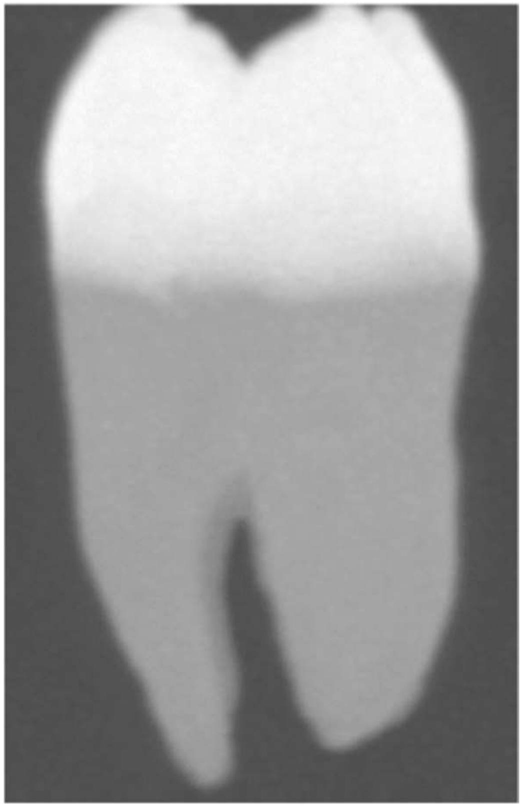
4. Composite Values Along the Ray

- Use an alpha (opacity) composting technique to blend the color values obtained from data samples along the ray

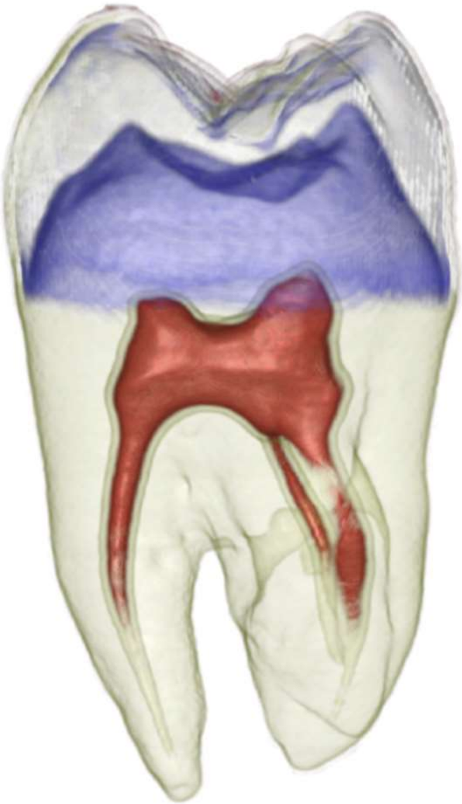




MIP vs Composite-based Volume Rendering



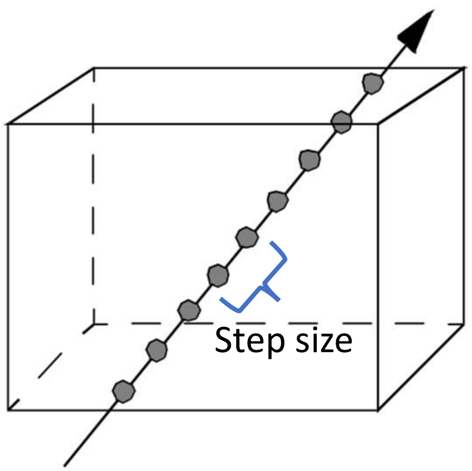
Maximum Intensity Projection

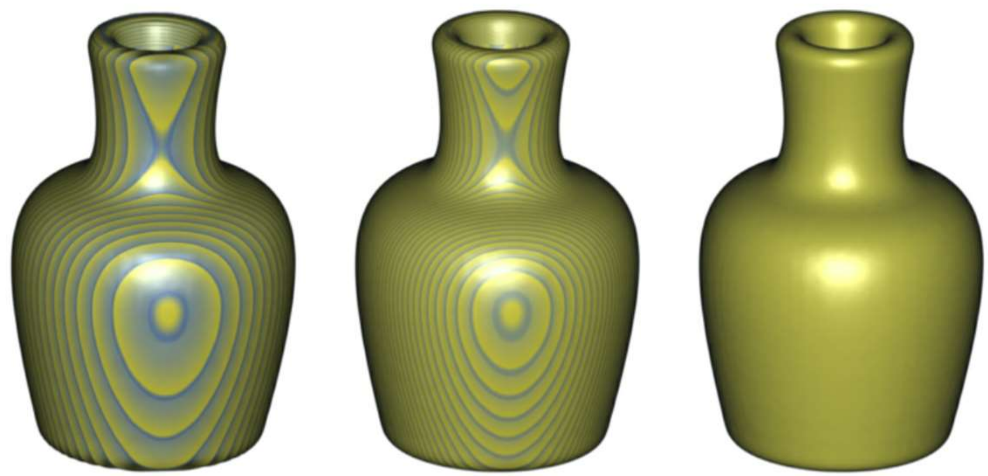


Direct Volume Rendering using Compositing

Effect of Step Size During Ray Traversal

- The quality of the image produced from the data depends on the step size when each ray is traversed through the data
  - Large hop/step size causes artifacts in the final image
  - Smaller step size makes image more accurate but also computationally more expensive

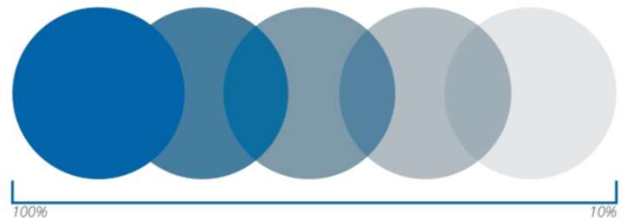




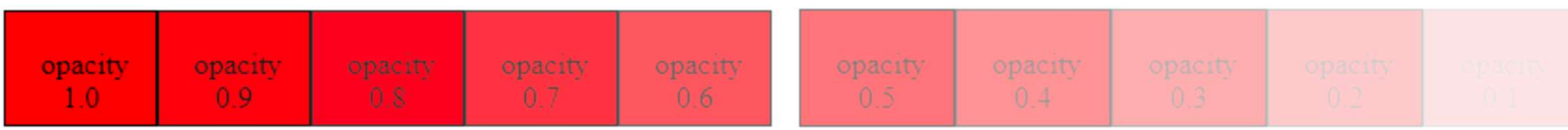
Step size = 2.0   Step size = 1.0   Step size = 0.1

Transparency and Opacity

- Other than RGB color values, there is one more channel – **opacity (A)**
  - Compute RGBA color components
  - Opacity (A) = 1 – transparency (T)
  - Range [0.0 ... 1.0]



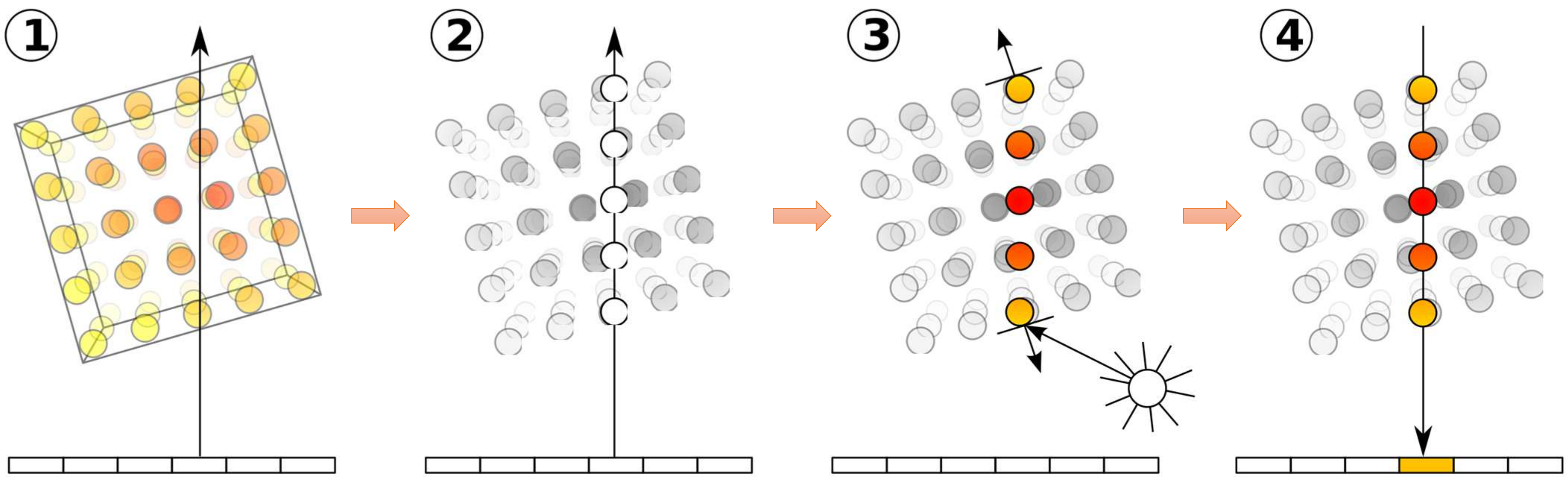
- Opacity (A) multiplied by RGB color creates a weighting effect



Ray Casting and Compositing

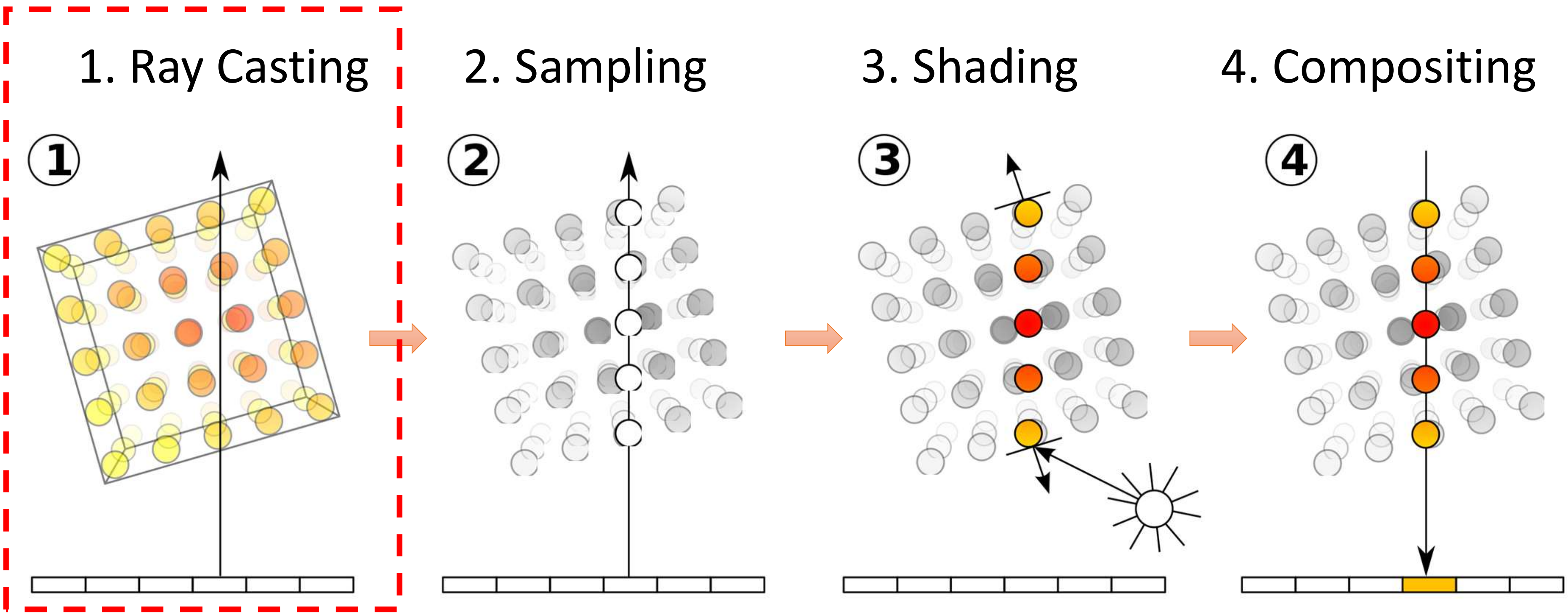
- Direct Volume Rendering

1. Ray Casting 2. Sampling 3. Shading 4. Compositing



Ray Casting and Compositing

- Direct Volume Rendering



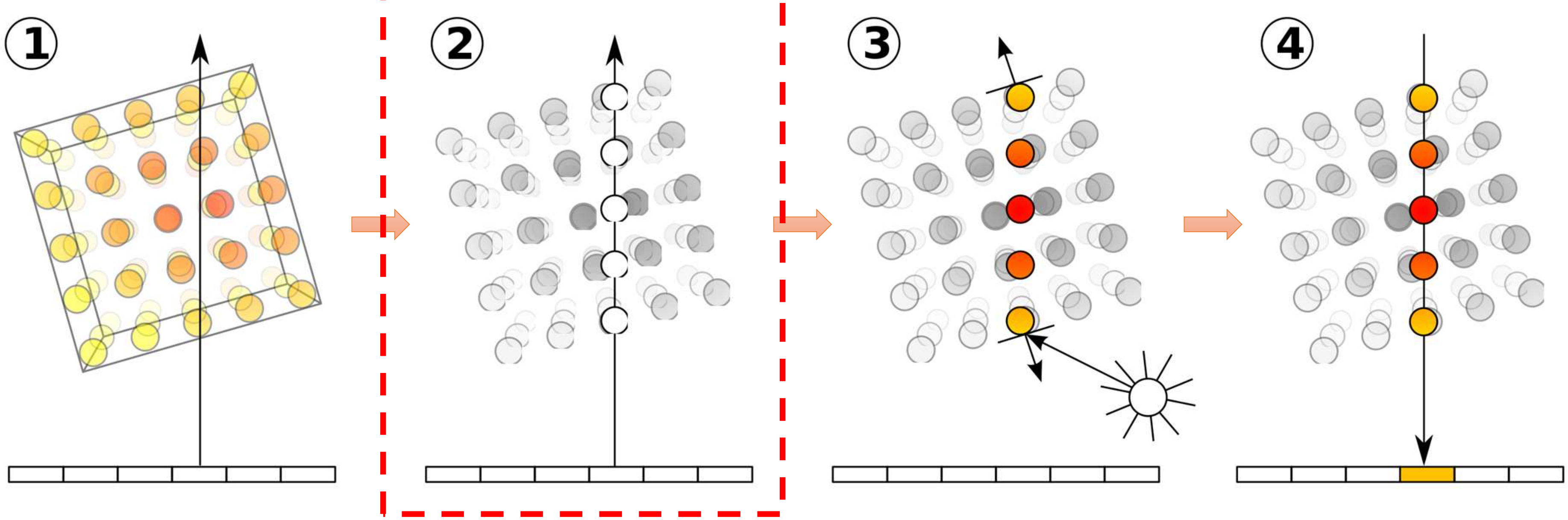
Wikipedia

Ray Casting and Compositing

IITK CS360A

- Direct Volume Rendering

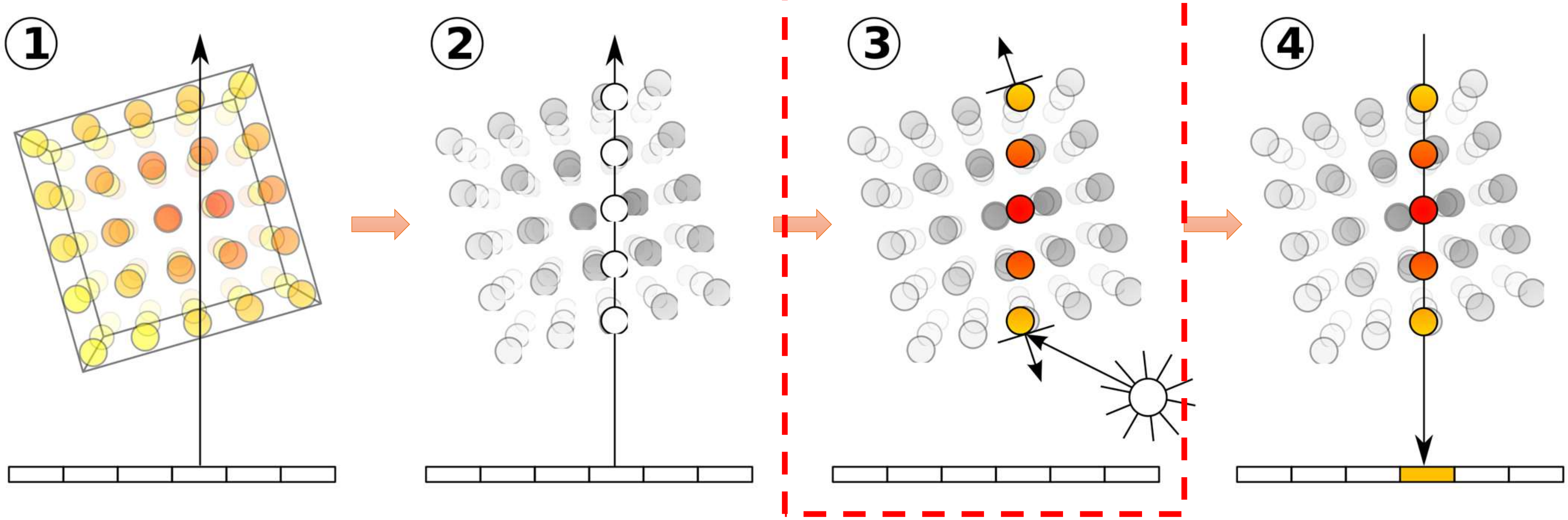
1. Ray Casting 2. Sampling 3. Shading 4. Compositing



Ray Casting and Compositing

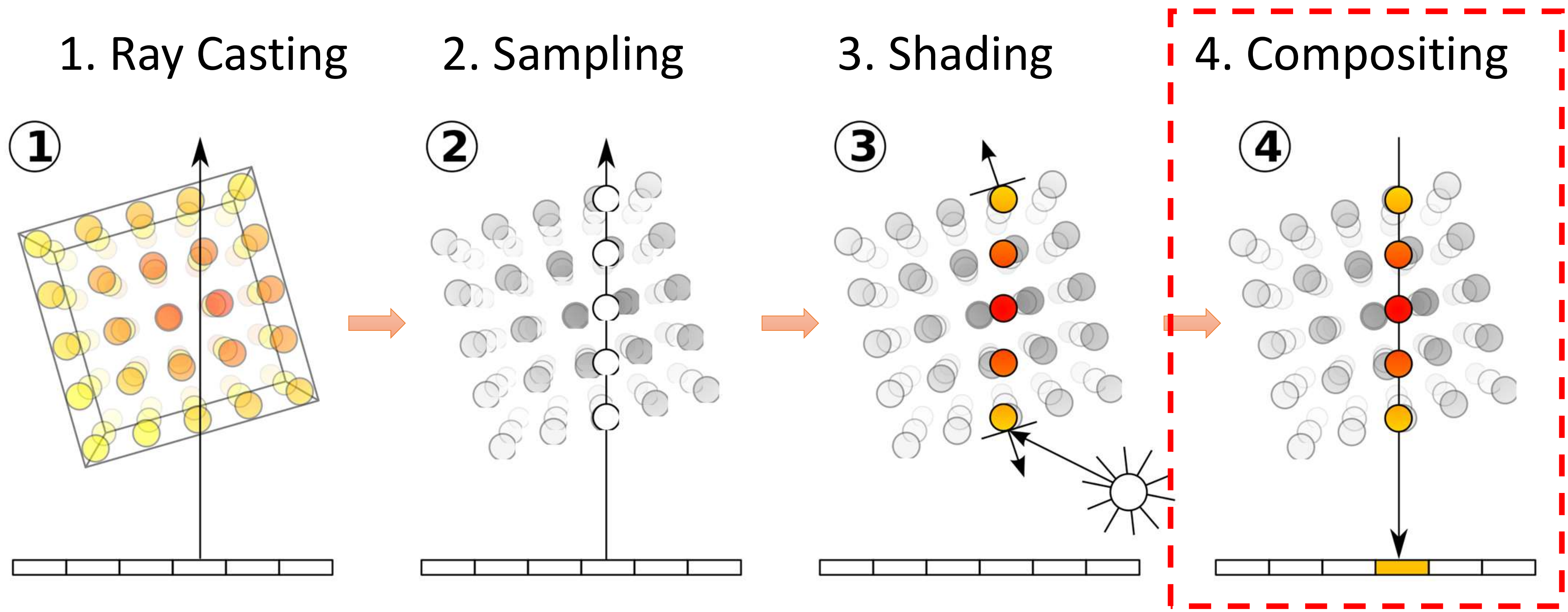
- Direct Volume Rendering

1. Ray Casting   2. Sampling   3. Shading   4. Compositing



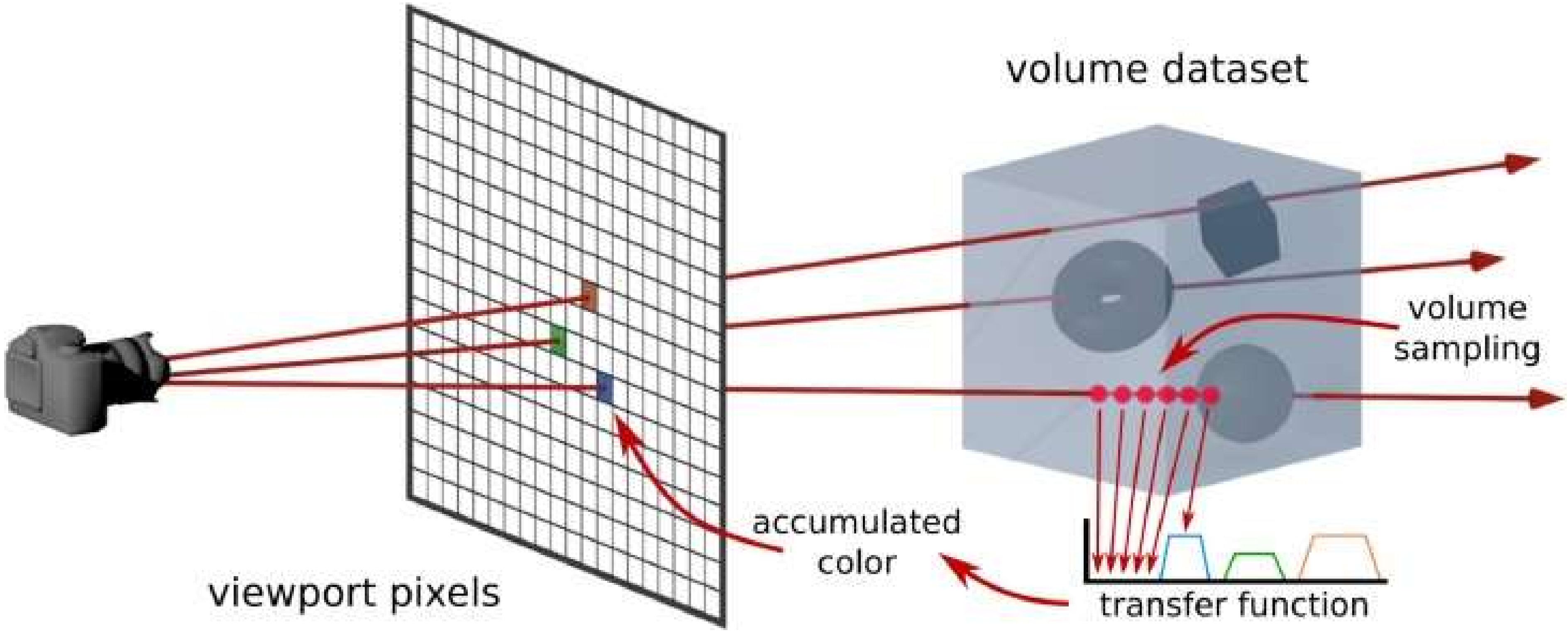
Ray Casting and Compositing

- Direct Volume Rendering



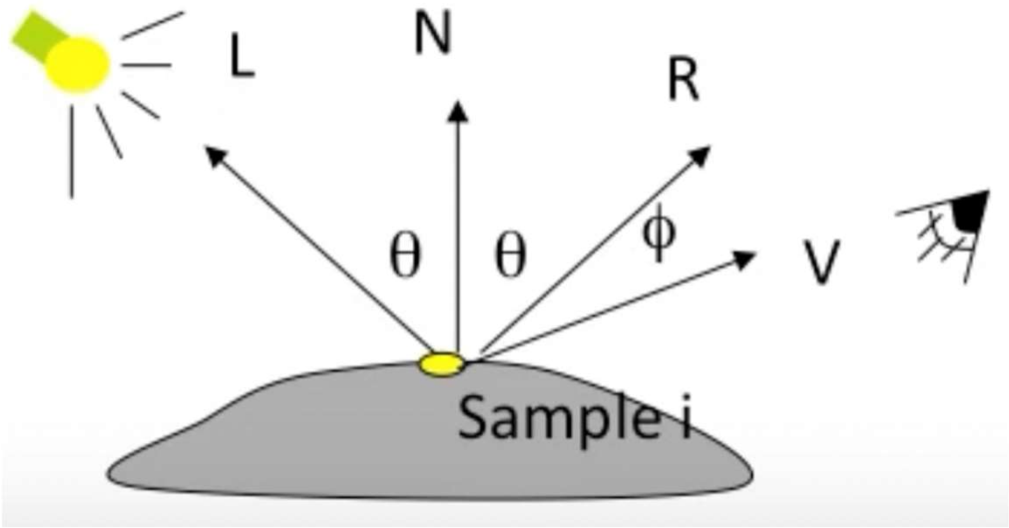
Ray Casting and Compositing

- Direct Volume Rendering



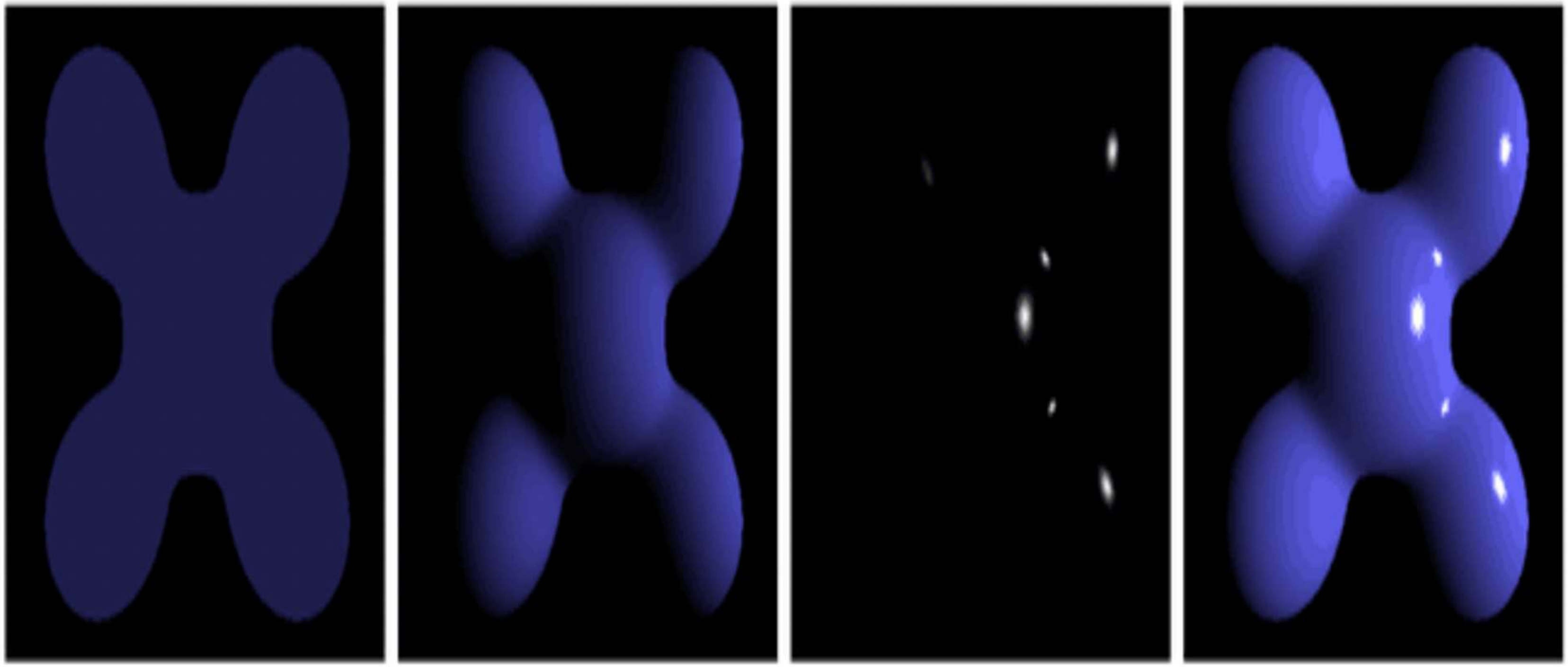
Shading: Phong Illumination Model

- Shading is the process of computing final color for each pixel considering its color, opacity, location of the viewer, distance and direction of the light, etc.
- Phong Illumination = ambient + diffuse + specular



Shading: Phong Illumination Model

- Illumination = ambient + diffuse + specular

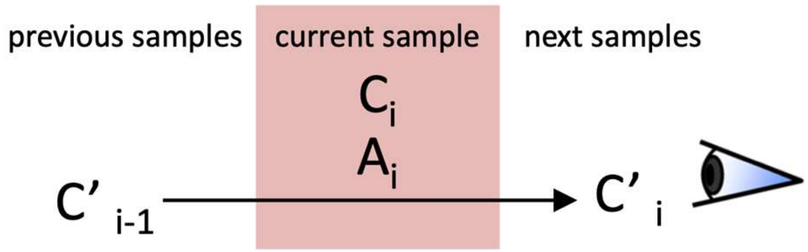


Ambient　　Diffuse　　Specular　　Ambient + Diffuse + Specular

https://svenbob.github.io/report%20of%20Project3

Opacity and Color Blending: Compositing

Back-to-front rendering



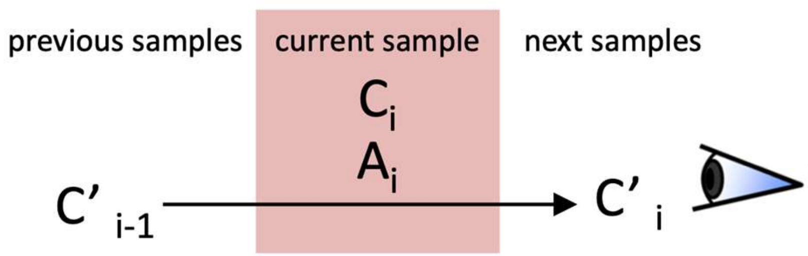
$$C_i' = C_i A_i + (1 - A_i) C_{i-1}'$$

A: Opacity = 1- Transparency = 1 - T

C: Color

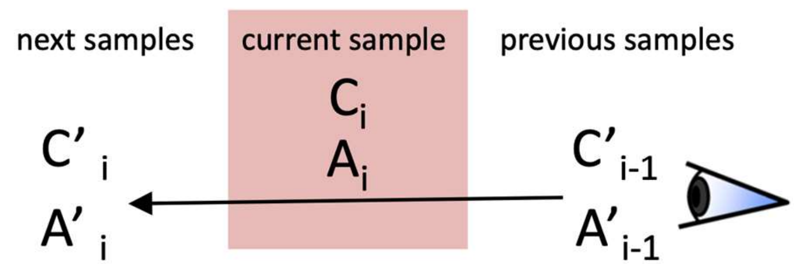
Opacity and Color Blending: Compositing

Back-to-front rendering



$$C_i' = C_i A_i + (1 - A_i) C_{i-1}'$$

Front-to-back rendering



$$C_i' = C_{i-1}' + (1 - A_{i-1}')C_i A_i$$

$$A_i' = A_{i-1}' + (1 - A_{i-1}')A_i$$

A: Opacity = 1- Transparency = 1 - T

C: Color