Module #12:
Direct Volume Rendering

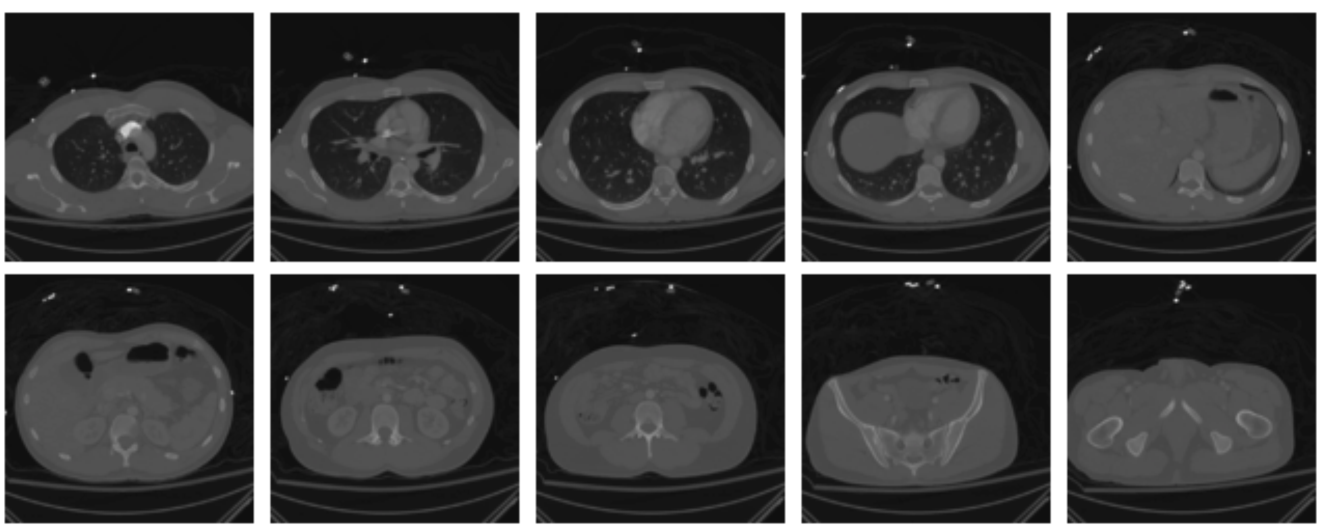
Objective

- Discuss the principles behind *direct volume rendering*
- Describe how to use transfer functions to change the properties of the visualization
- Explain the ray casting process
- Provide different examples of volume data

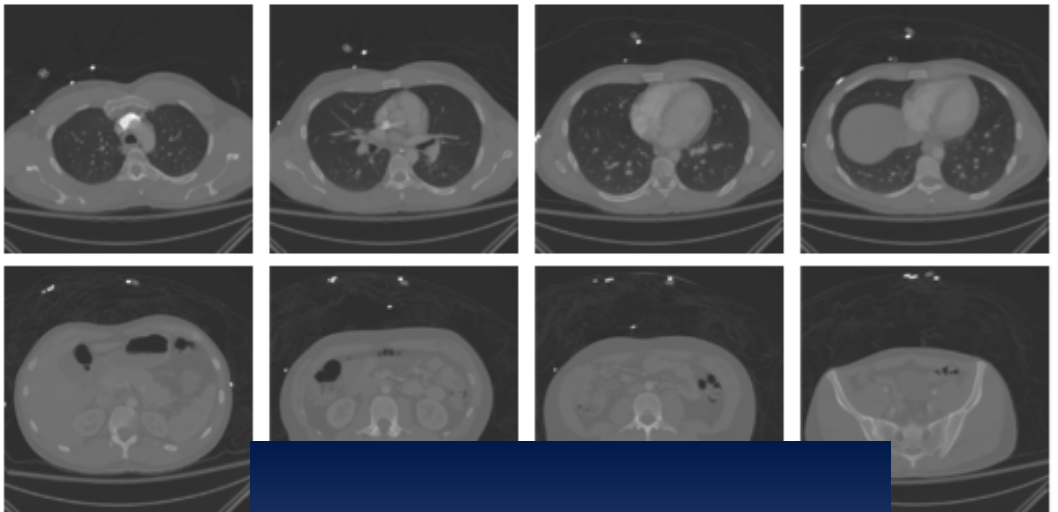
Why 3D Visualization?

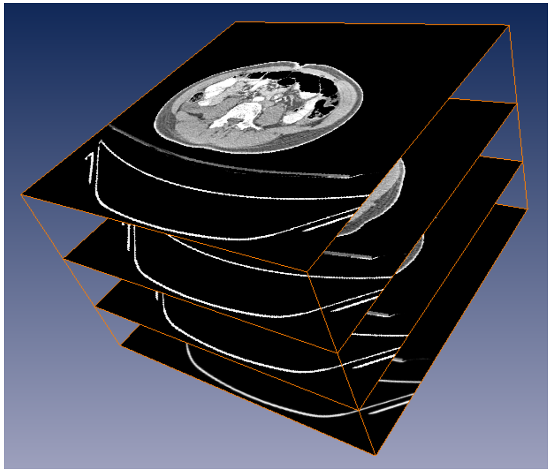
- Advances in 3D acquisition devices have created
  - Challenging task of analyzing large set of images
- Can we use volume visualization as a technique to quickly explore large datasets?

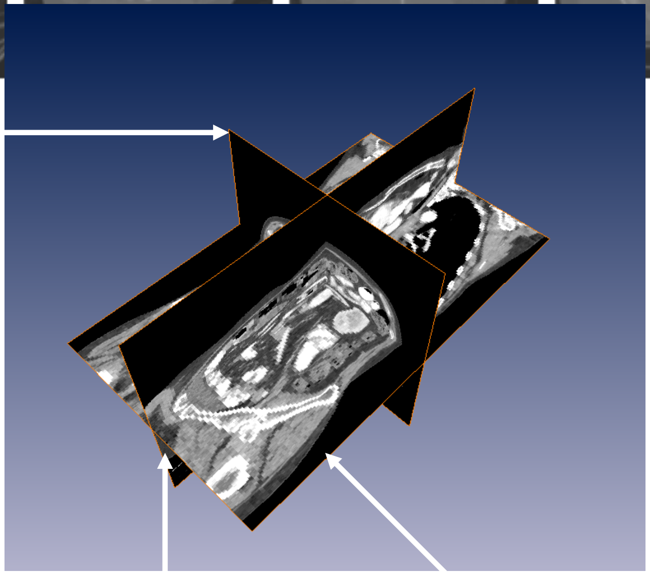
Siemens, 2007

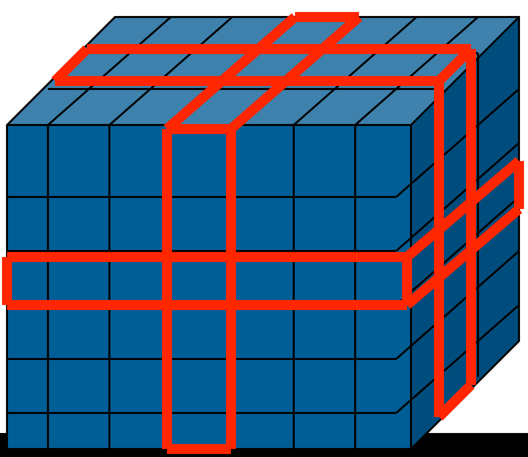


What's a volume?



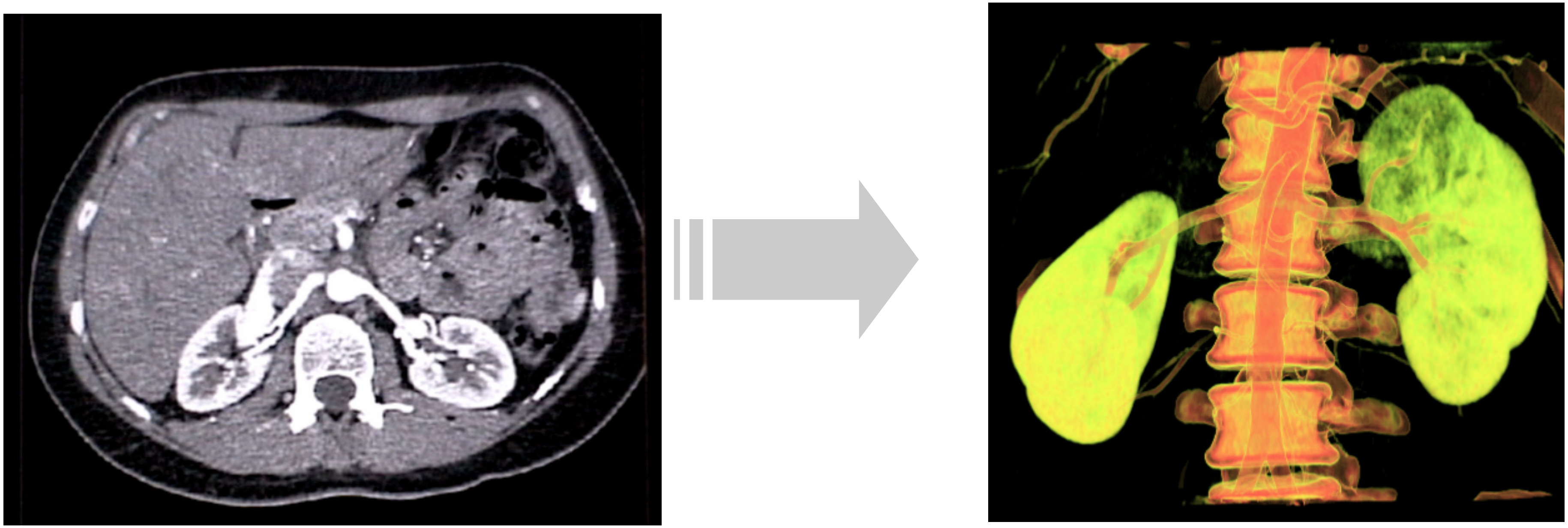






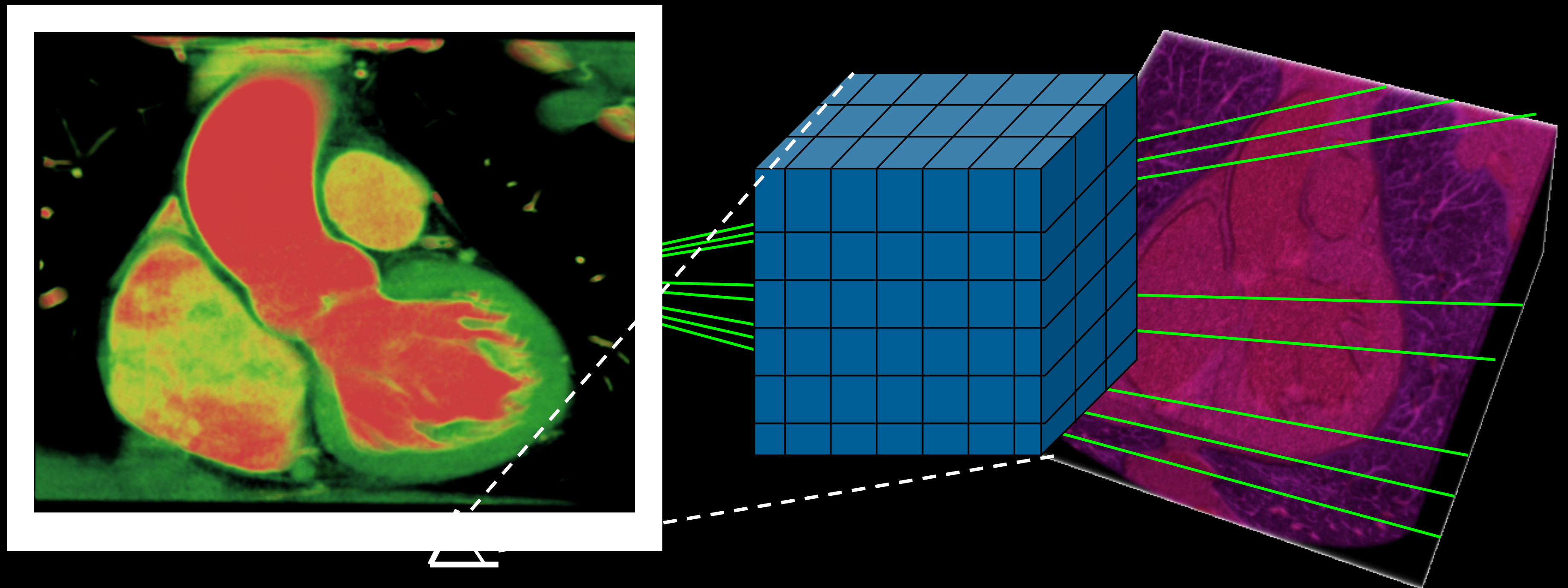
Direct Volume Rendering

- Direct Volume Rendering
  - Technique used to display a 2D projection/image of a 3D image
  - Effective method to render different materials



Raycast Function

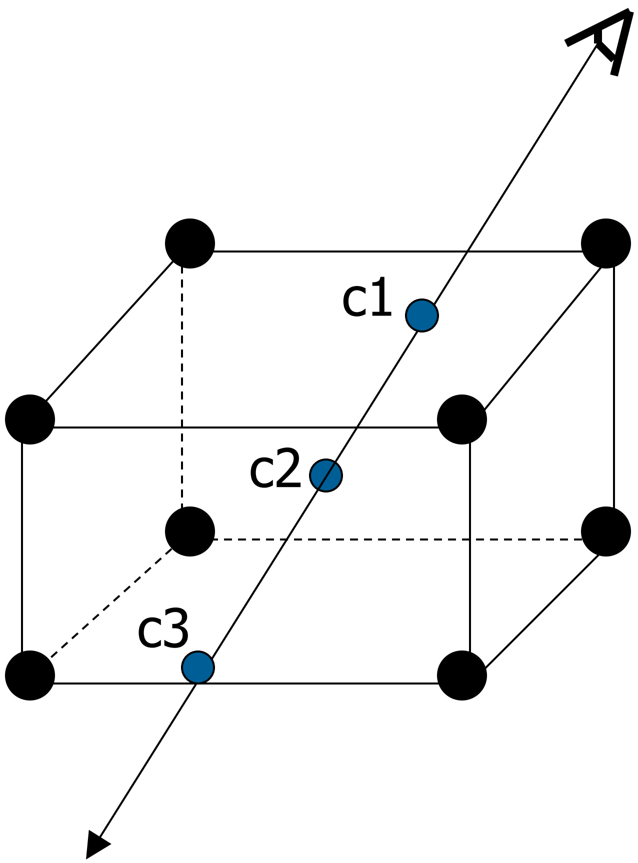
- Integration process to simulate the light transport within a volumetric medium



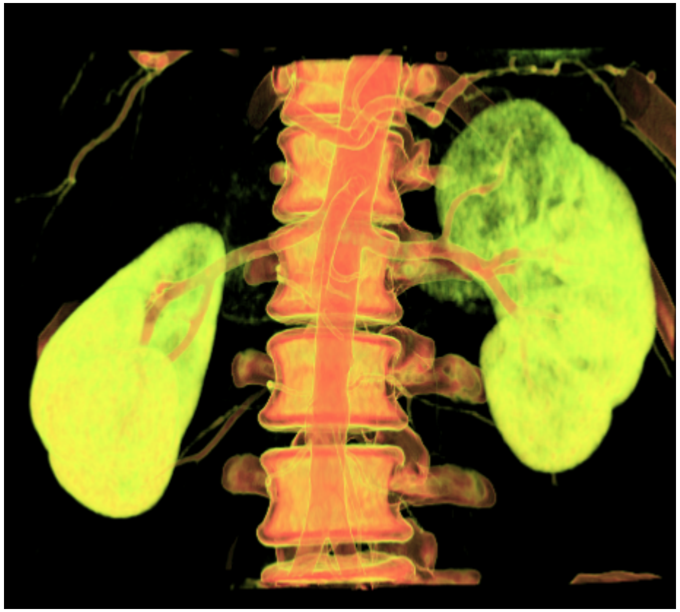
Ray Casting

- Stepping through the volume: a ray is cast into the volume, sampling the volume at certain intervals
  - The sampling intervals are usually equi-distant, but don't have to be
- At each sampling location, a sample is interpolated / reconstructed from the grid voxels
  - Popular filters are: nearest neighbor (box), trilinear (tent), Gaussian, cubic spline
- Classification:
  - map from density to color, opacity
  - Composite colors and opacities along the ray path

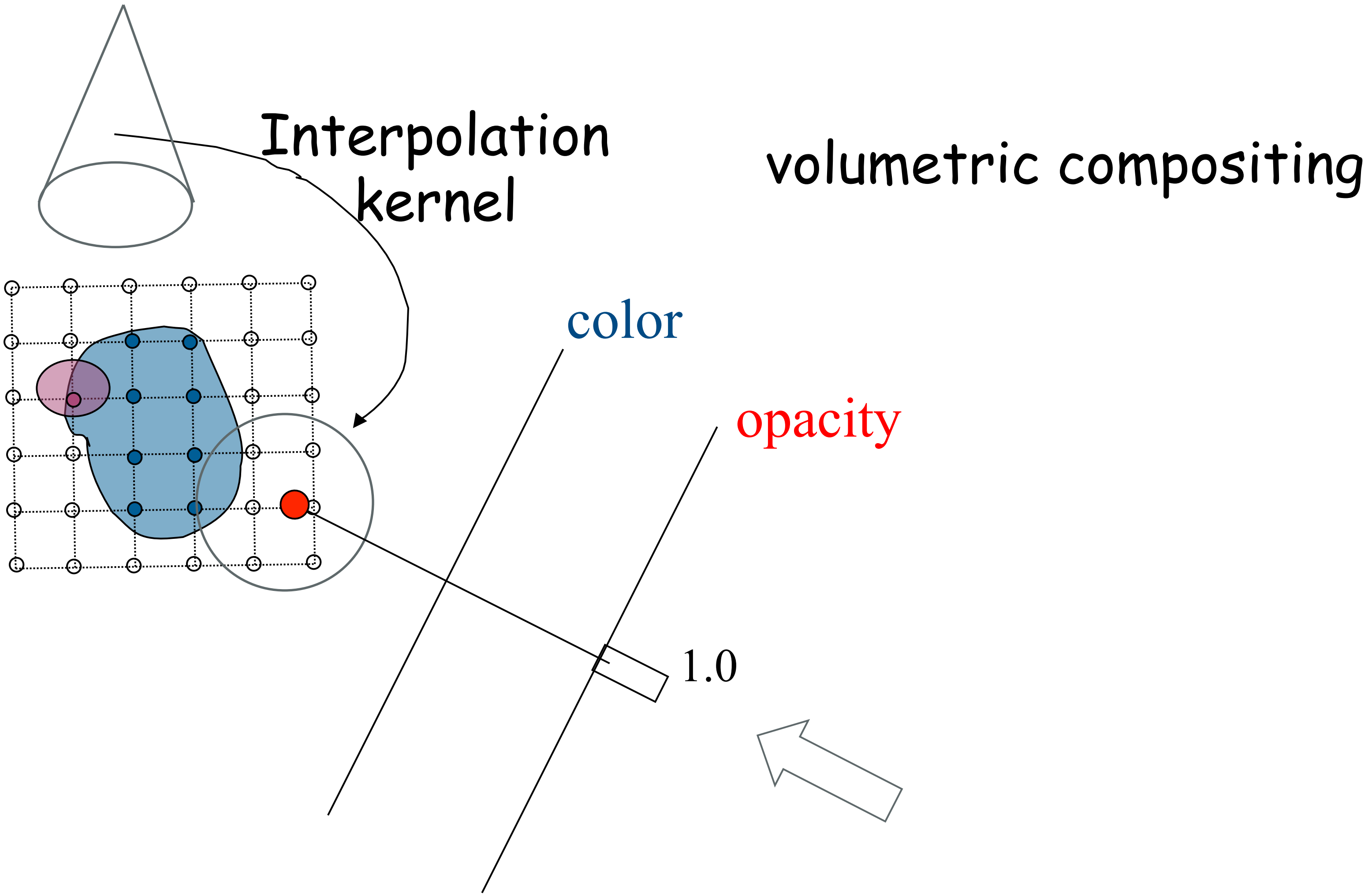
Basic Idea of Ray-casting Pipeline



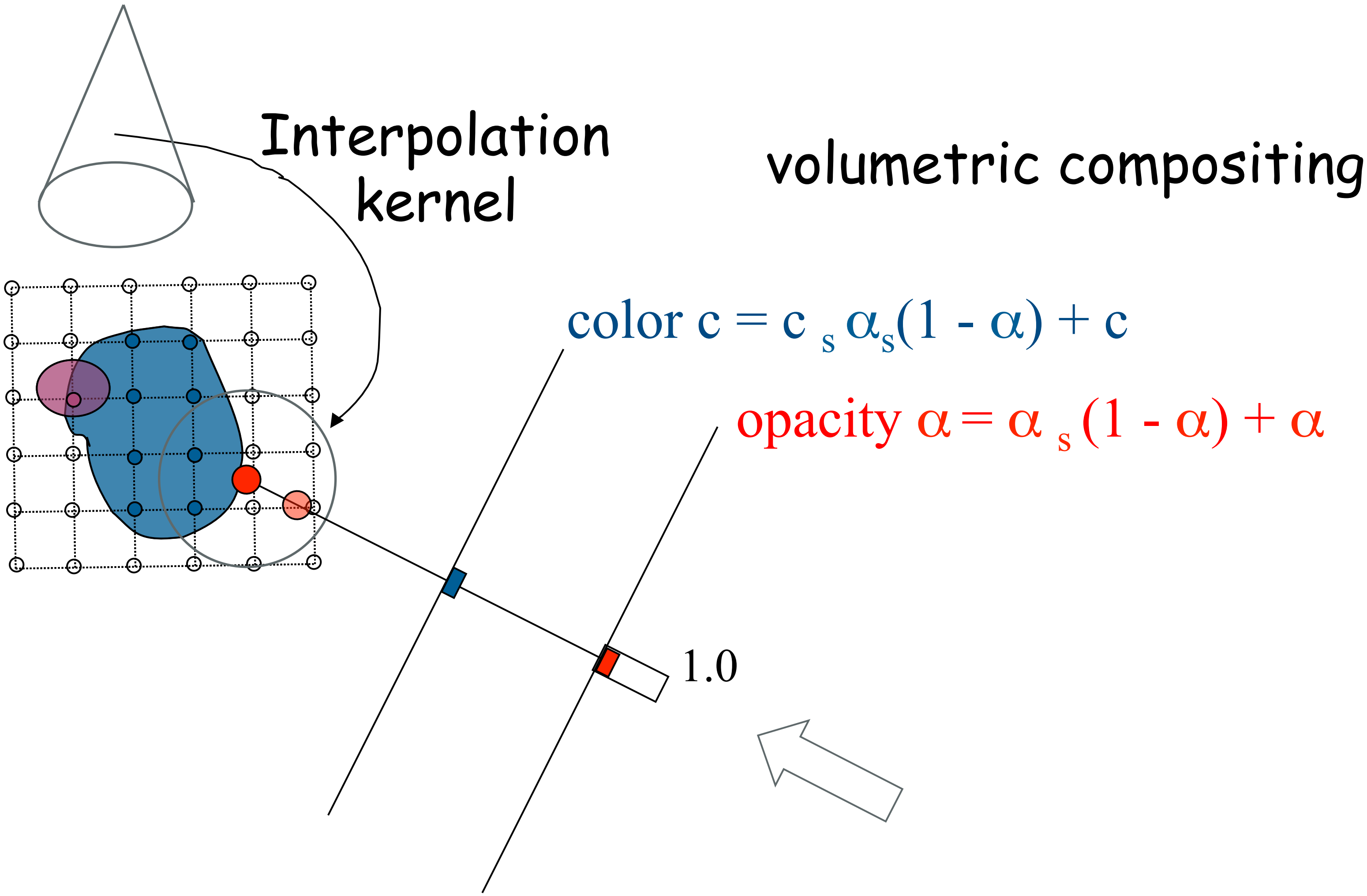
Direct Volume Rendering



Raycasting


Interpolation
kernel
volumetric compositing
color
opacity
1.0

Raycasting

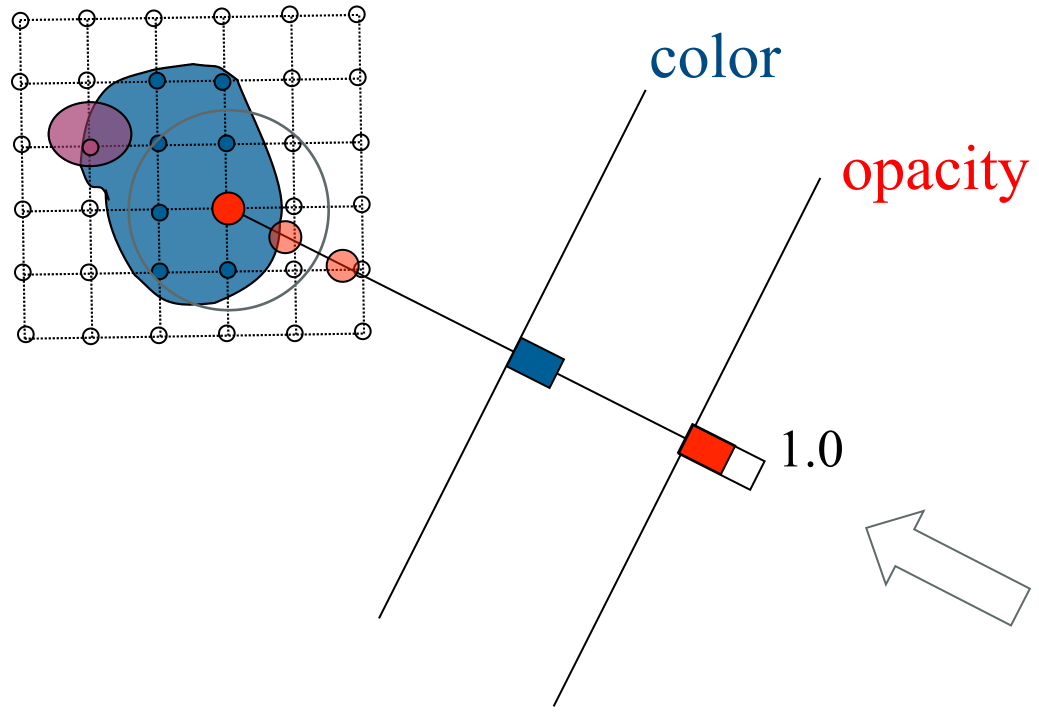


color $c = c_s \, \alpha_s (1 - \alpha) + c$

opacity $\alpha = \alpha_s (1 - \alpha) + \alpha$

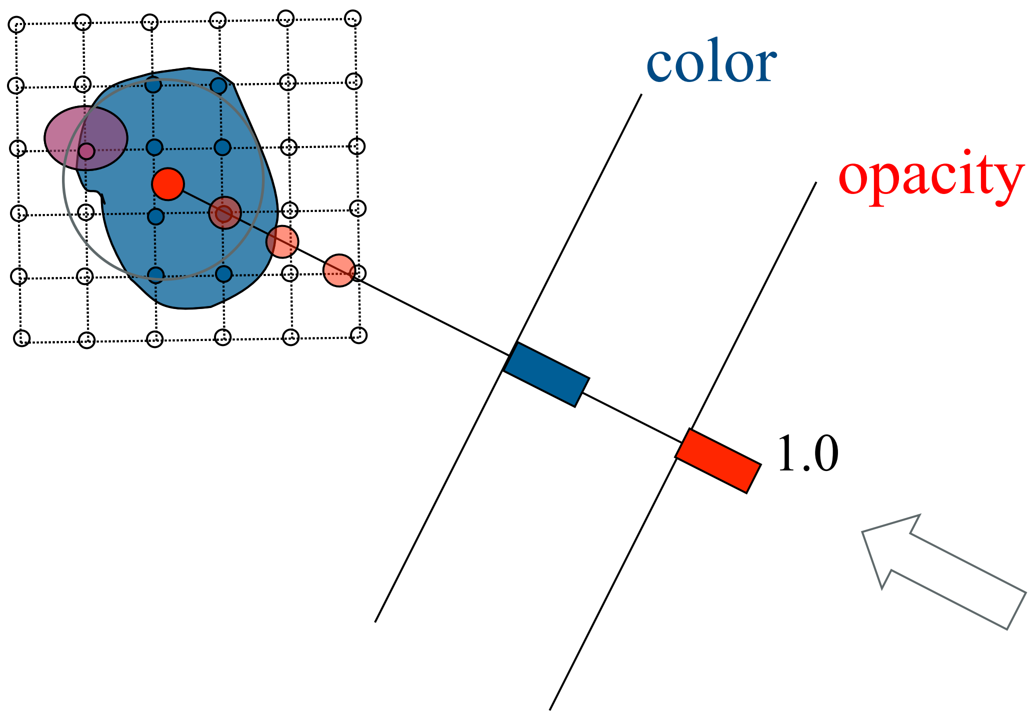
Raycasting

volumetric compositing



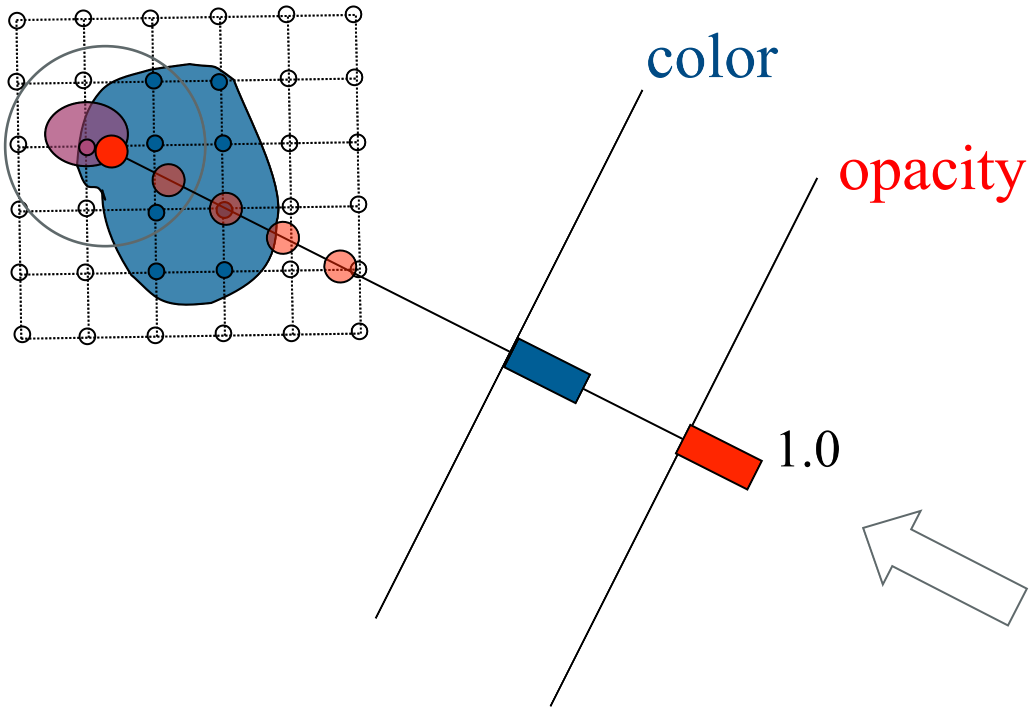
Raycasting

volumetric compositing



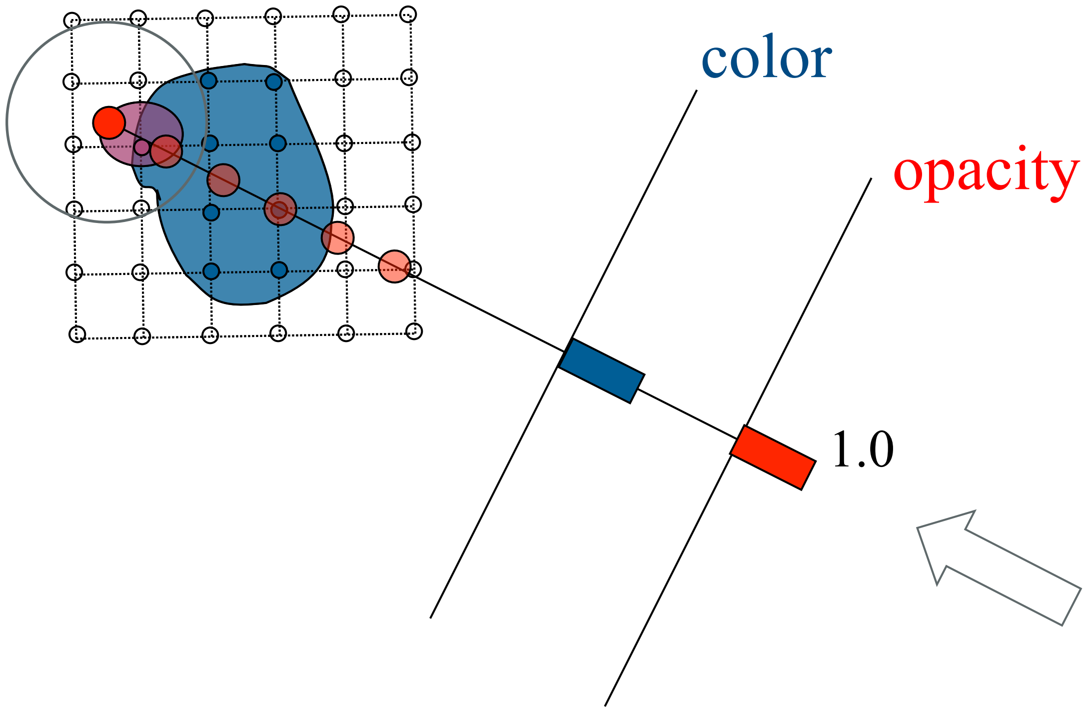
Raycasting

volumetric compositing



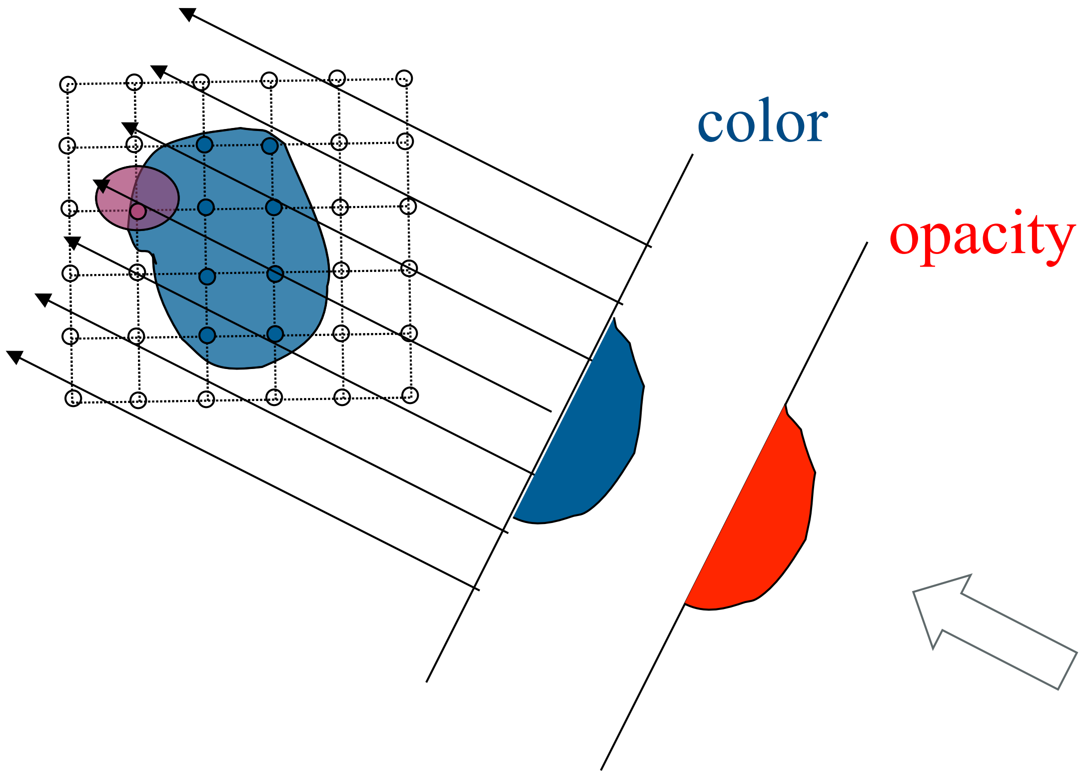
Raycasting

volumetric compositing

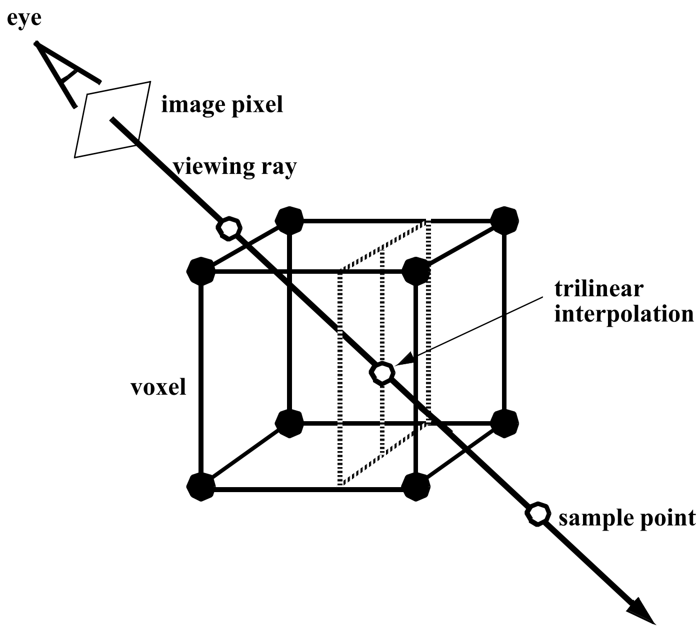


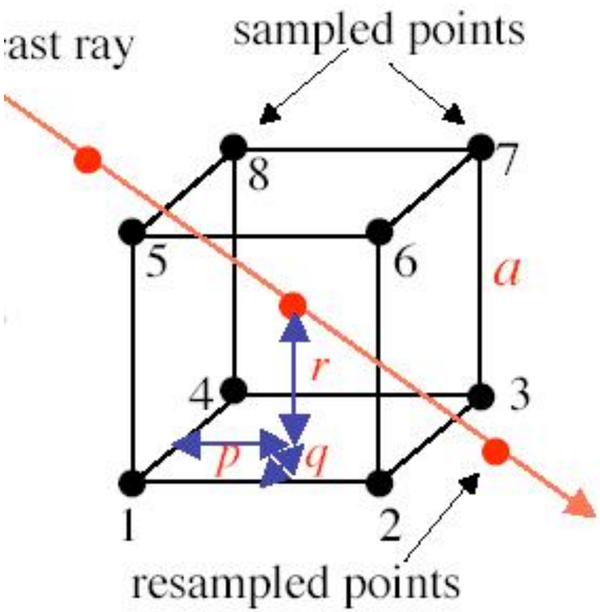
Raycasting

volumetric compositing



Trilinear - Interpolation





$$
\begin{aligned}
f_v = {} & f_1(1-p/a)(1-q/a)(1-r/a) + f_2(p/a)(1-q/a)(1-r/a) \\
& + f_3(p/a)(q/a)(1-r/a) + f_4(1-p/a)(q/a)(1-r/a) \\
& + f_5(1-p/a)(1-q/a)(r/a) + f_6(p/a)(1-q/a)(r/a) \\
& + f_7(p/a)(q/a)(r/a) + f_8(1-p/a)(q/a)(r/a)
\end{aligned}
$$

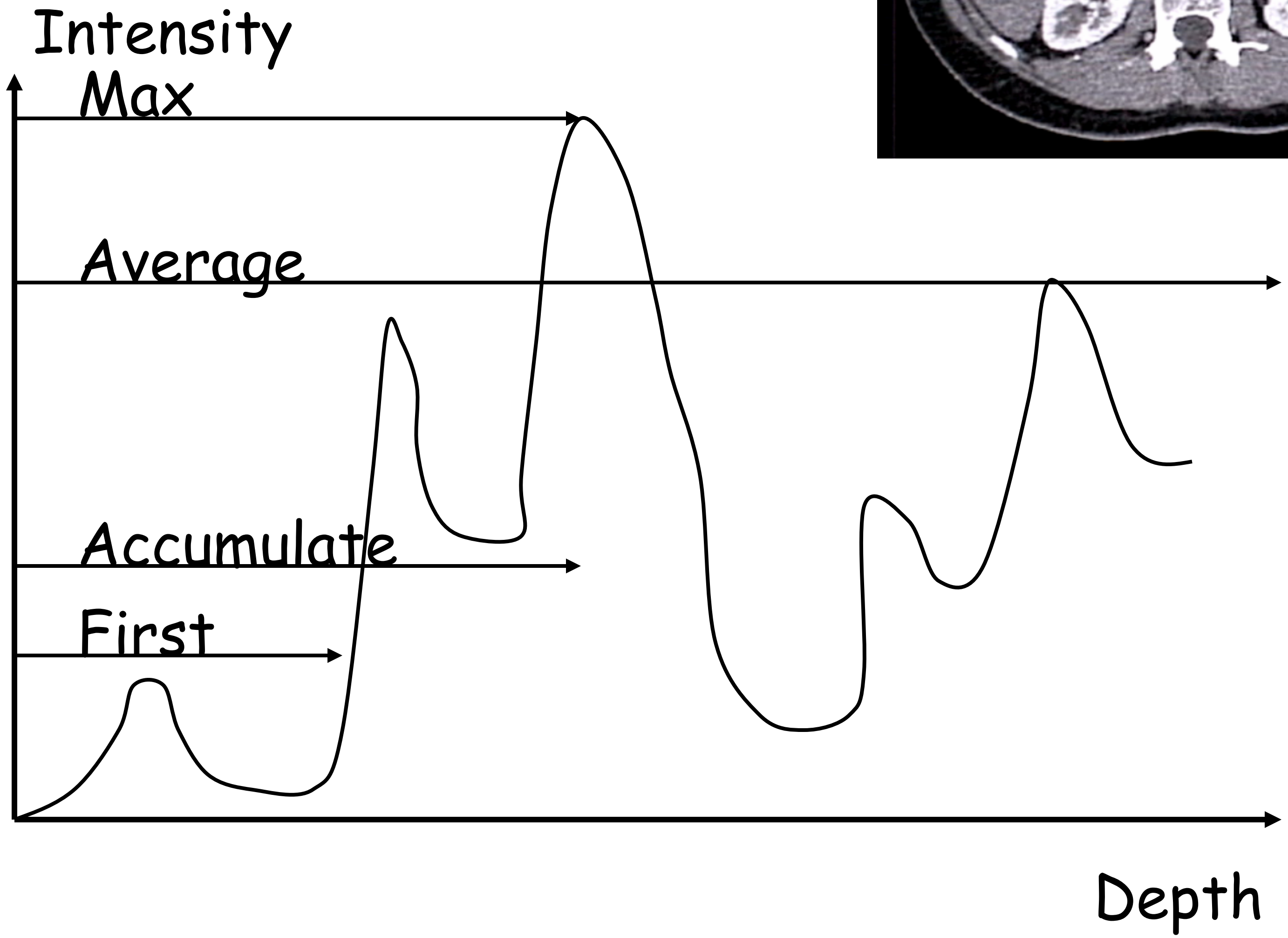
Classification/Transfer Function

- Maps raw voxel value into presentable entities:
  - color, intensity, opacity, etc.
  - Raw-data → material (R, G, B, a, Ka, Kd, Ks, .)

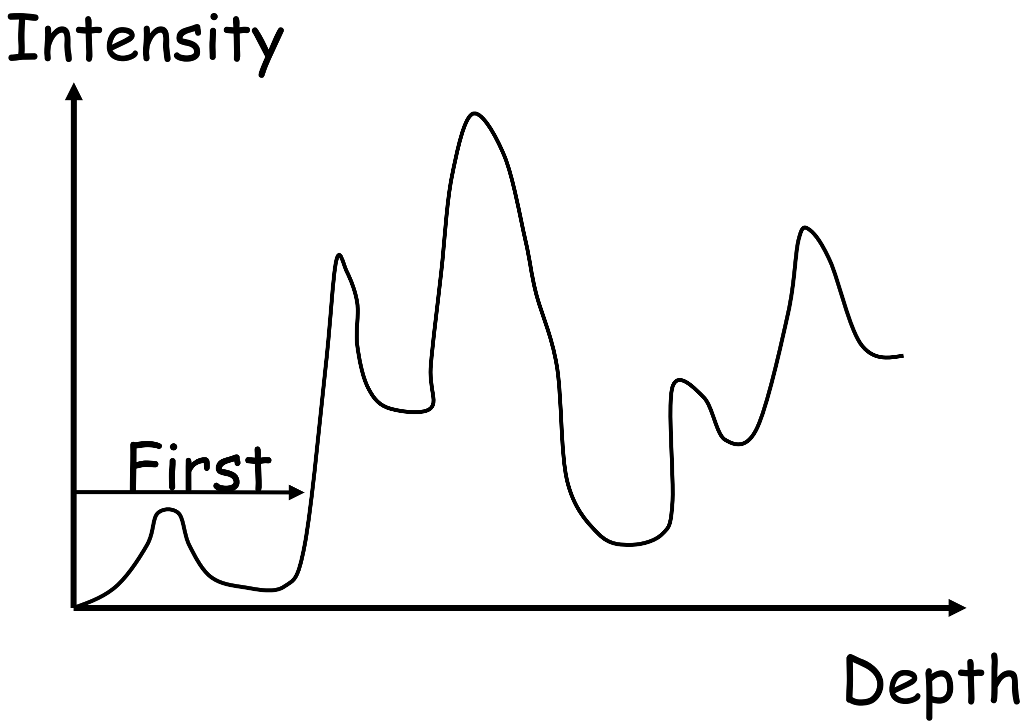
- Region of interest:
  - high opacity (more opaque)
  - no interest: translucent or transparent

- Often use look-up tables (LUT) to store the transfer function

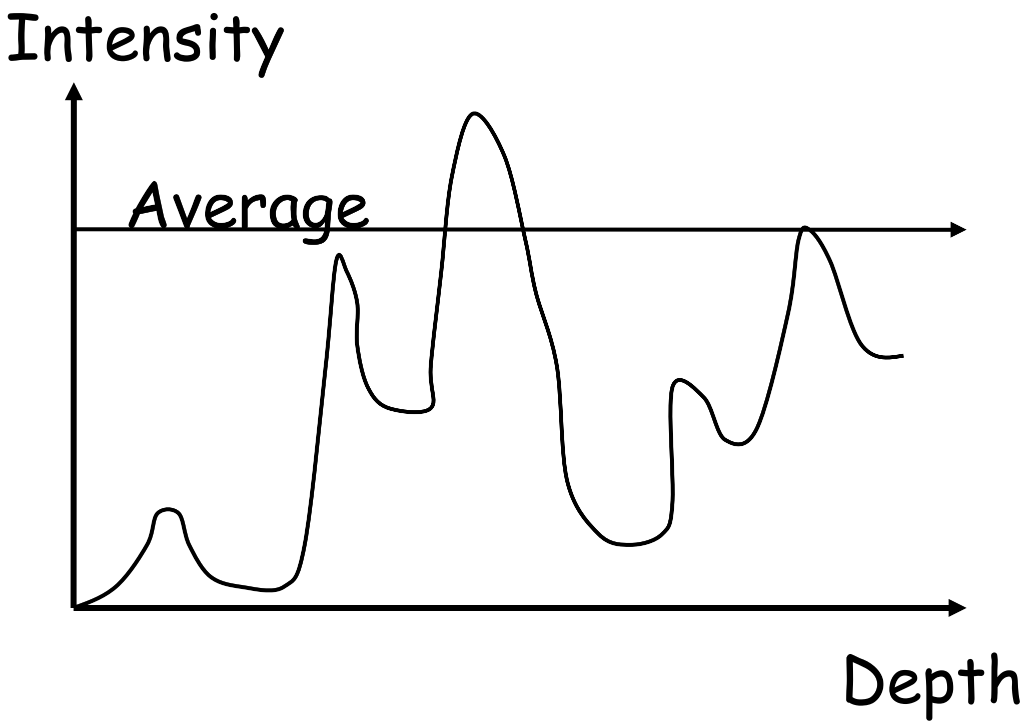
Ray Traversal Schemes

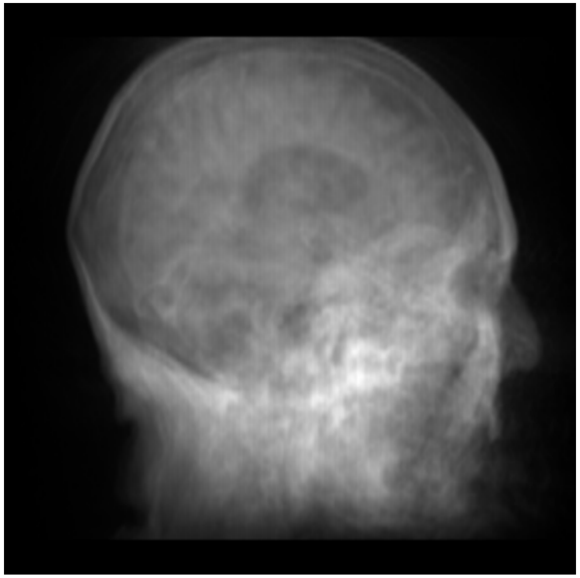


Ray Traversal - First



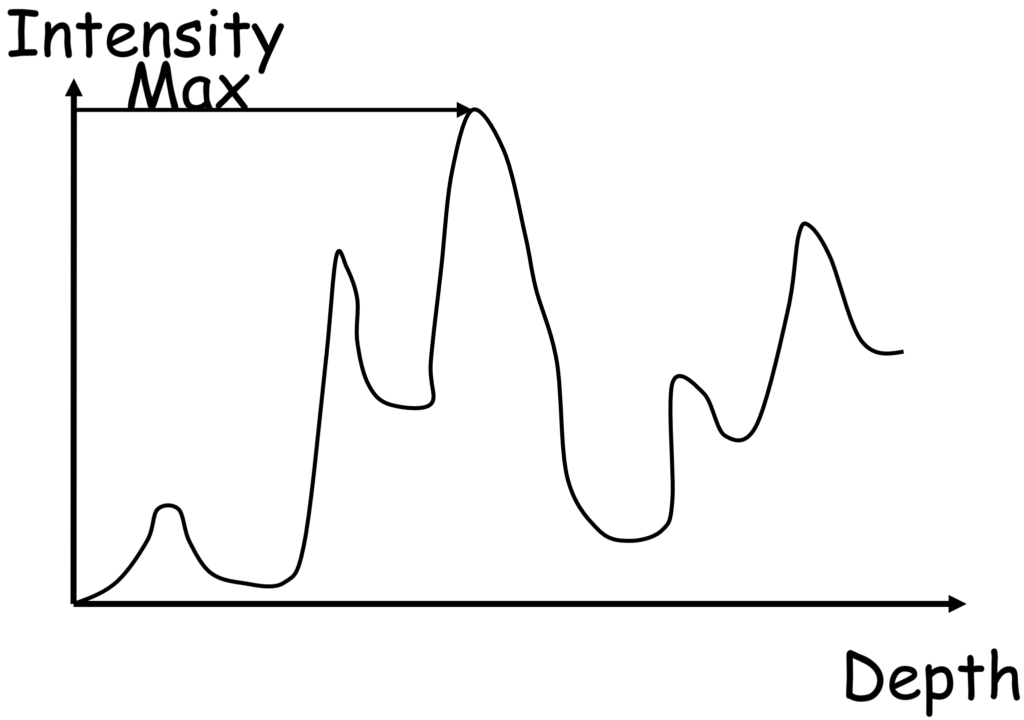
Ray Traversal - Average

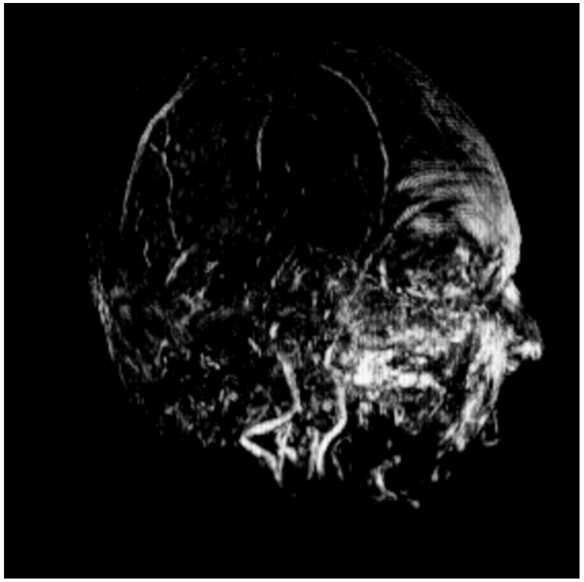




- **Average**: produces basically an X-ray picture

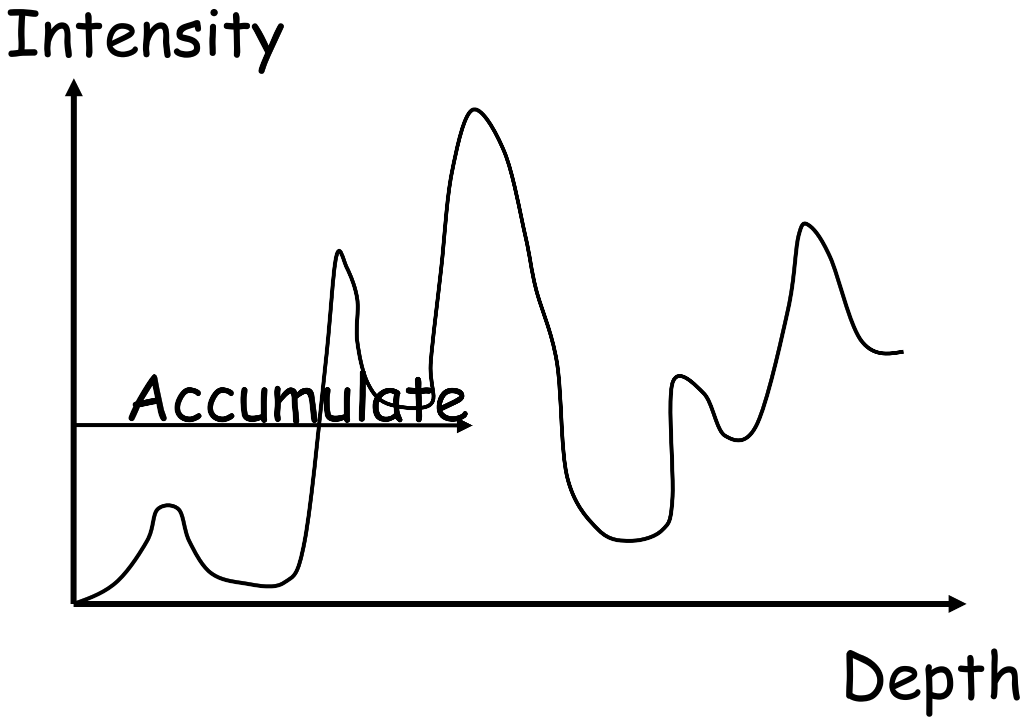
Ray Traversal - MIP

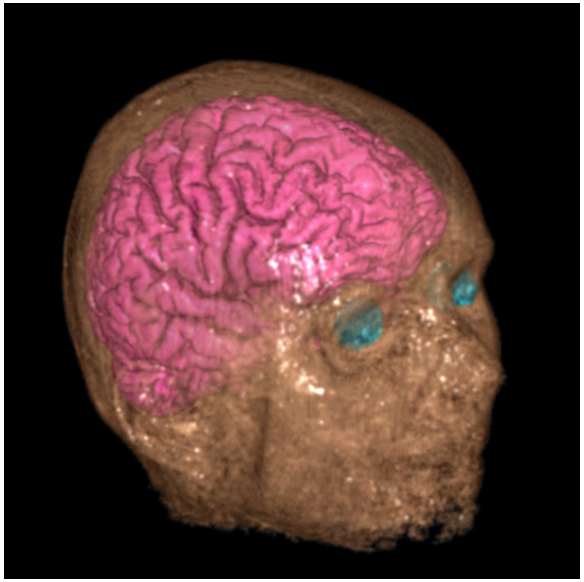




- **Max:** Maximum Intensity Projection used for Magnetic Resonance Angiogram

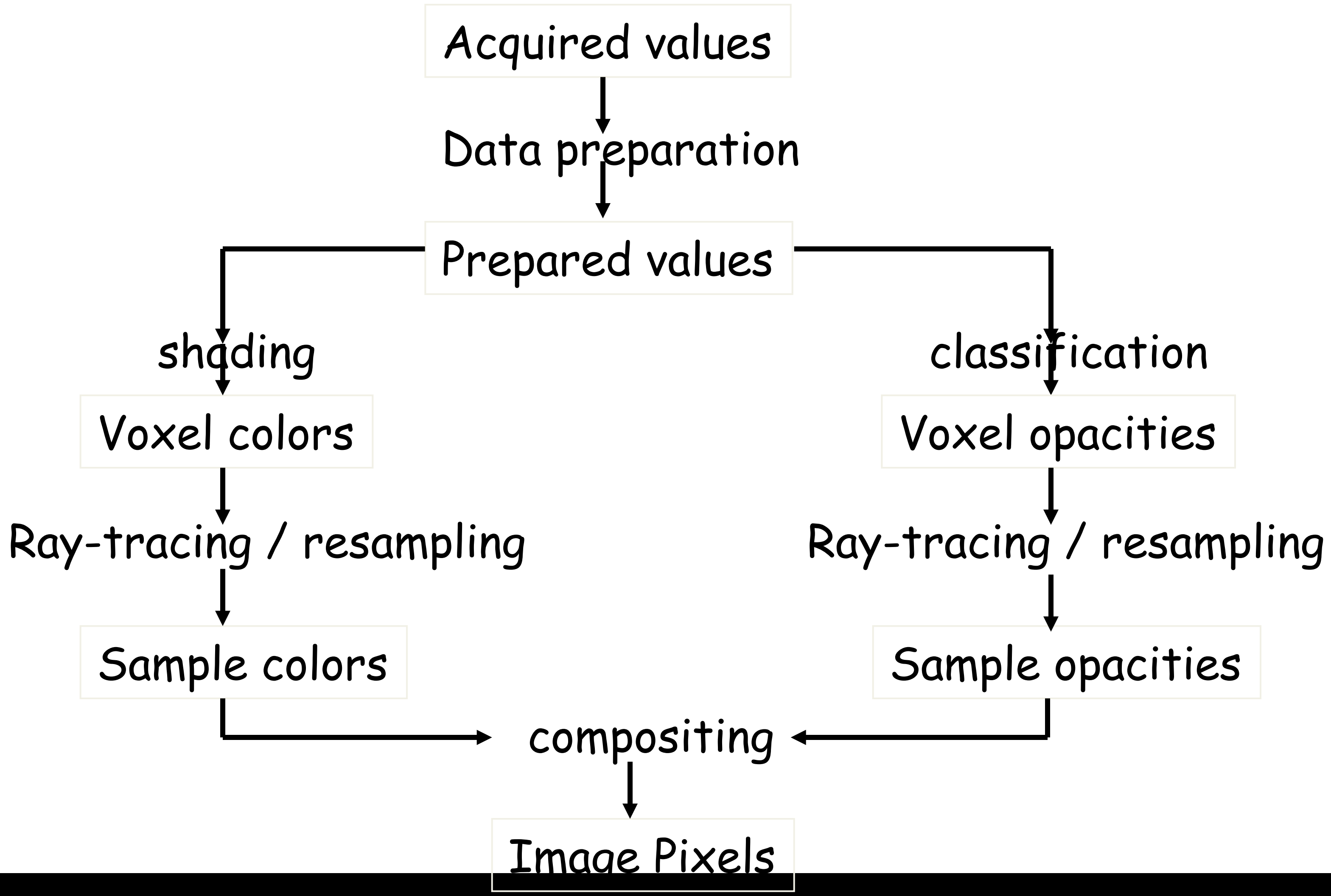
Ray Traversal - Accumulate





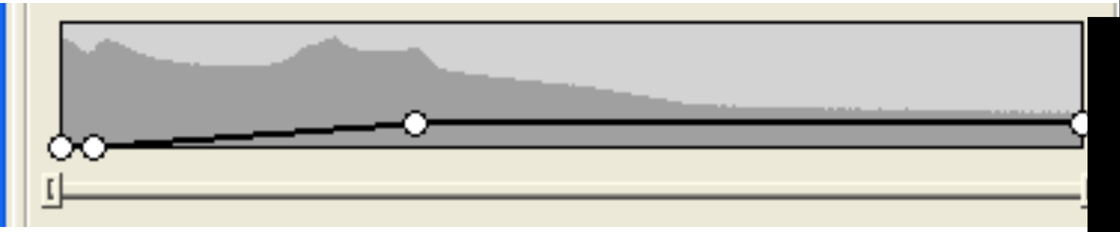
- Accumulate opacity while compositing colors: make transparent layers visible! (Levoy '88)

Volume Rendering Pipeline

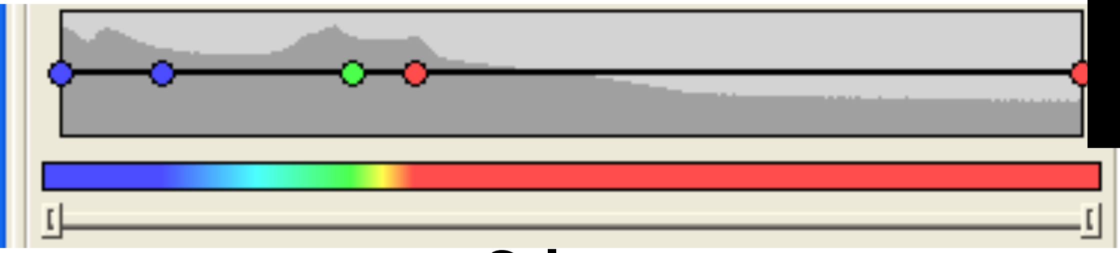


Transfer Functions

- How do we assign rendering properties?



Intensity Values



Color