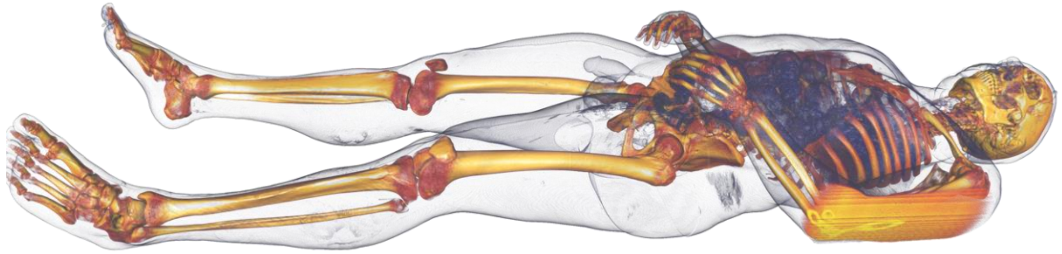
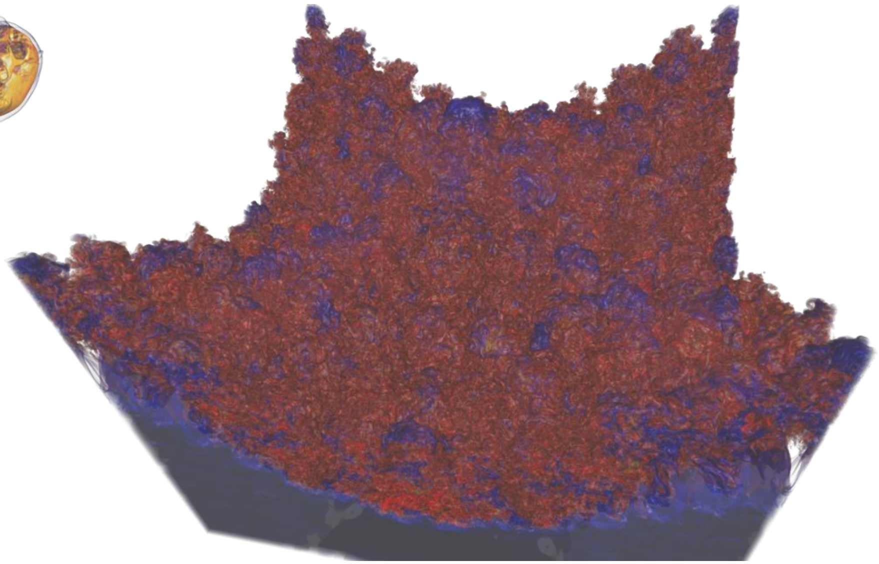
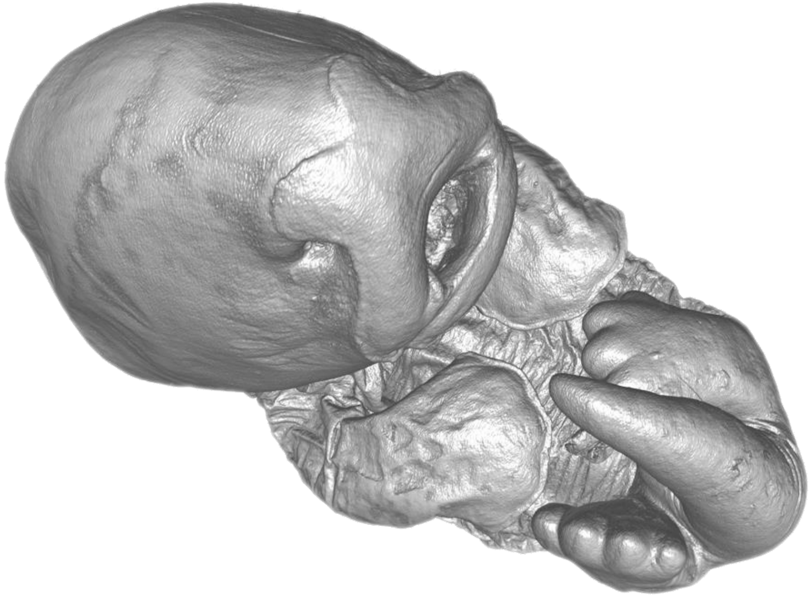
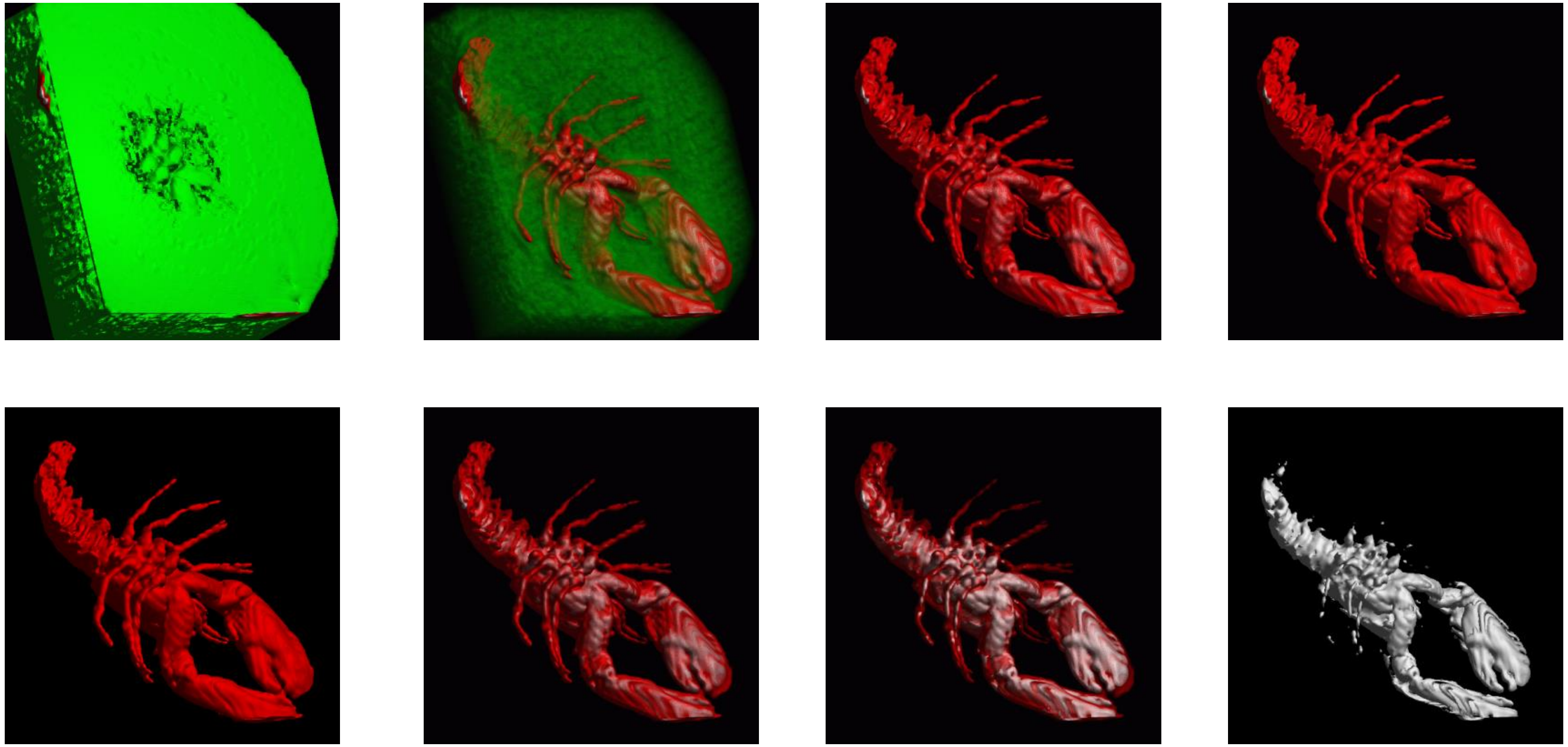
Visualization, DD2257

Prof. Dr. Tino Weinkauf

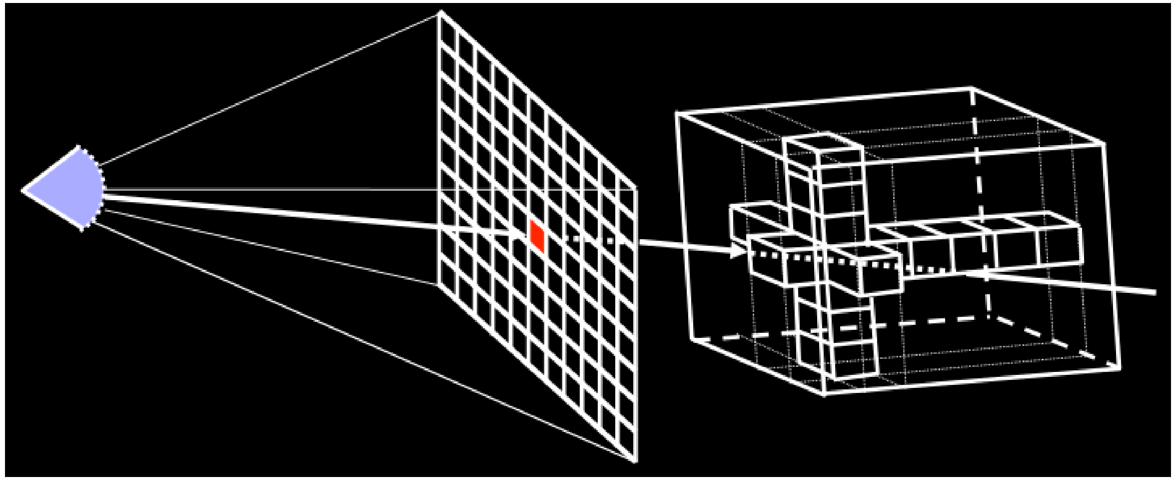
Direct Volume Rendering

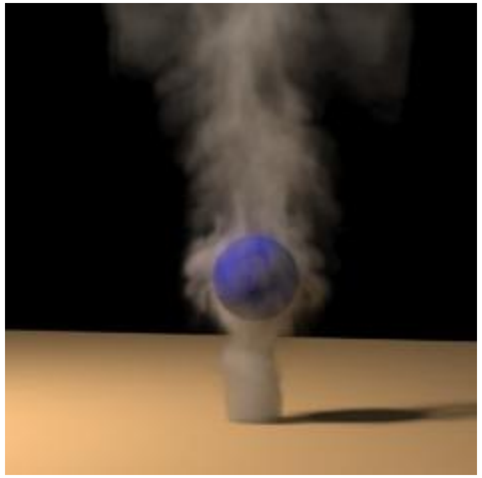




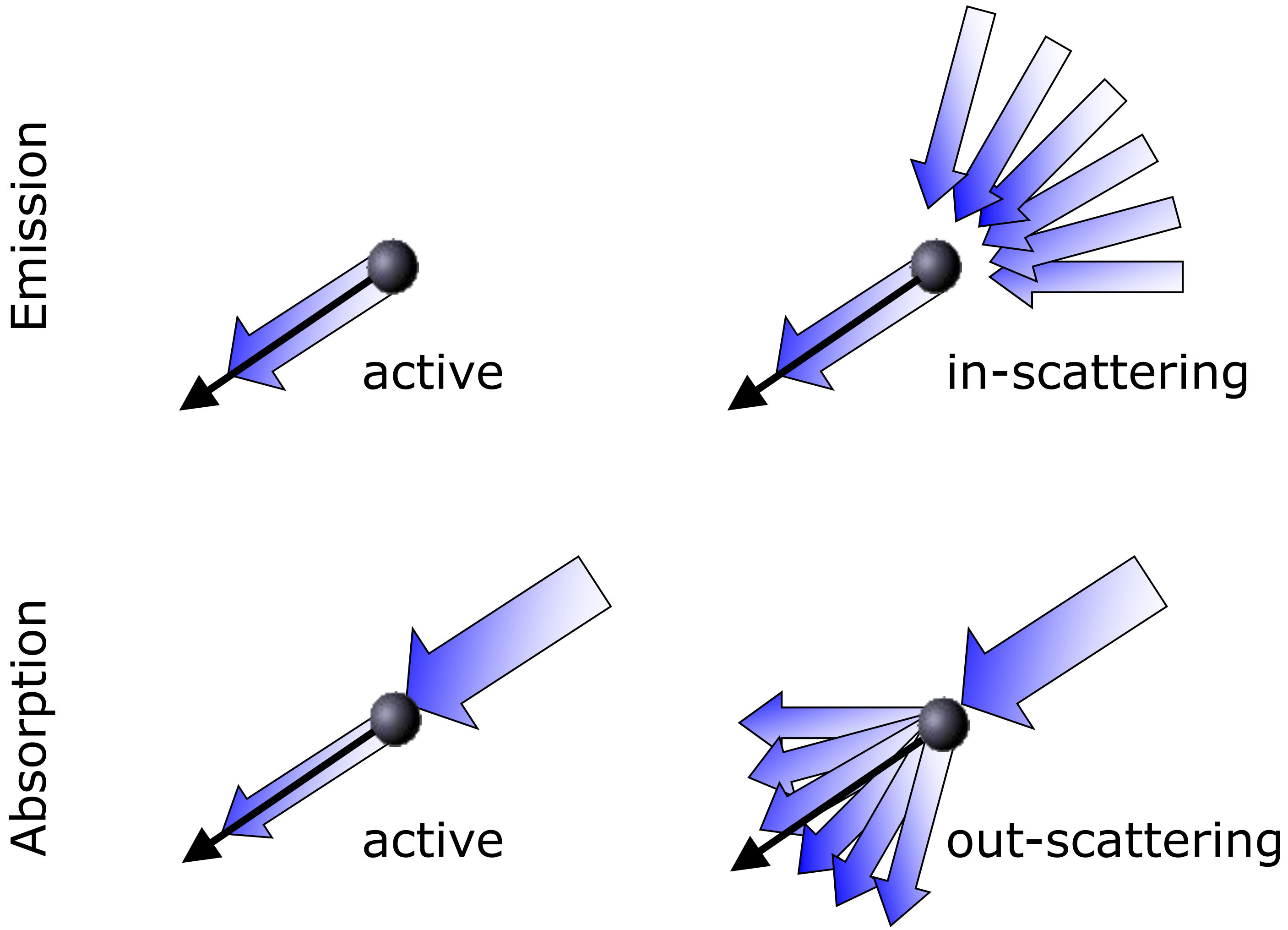



Same volume with different transfer functions that now include an alpha channel (=opacity) and are rendered volumetrically

Optical model: Each point in the volume is considered to *emit and absorb light*, according to the color and opacity specified by the transfer function.
Those contributions are *integrated along viewing rays* to produce the final image.






Emission
active
in-scattering
Absorption
active
out-scattering

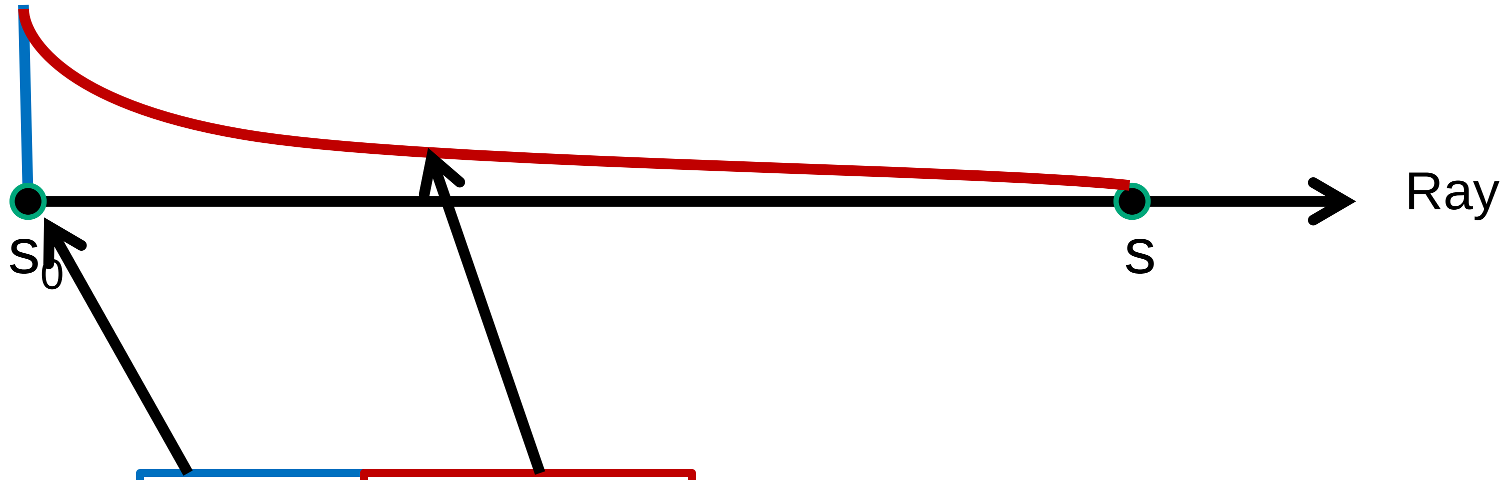
Light (Particle) interaction with density volume

- Emission only, Absorption only
- **Emission + Absorption**
- Scattering + shading/shadowing
- Multiple scattering

very important

$$I(s) =$$

Light intensity at s



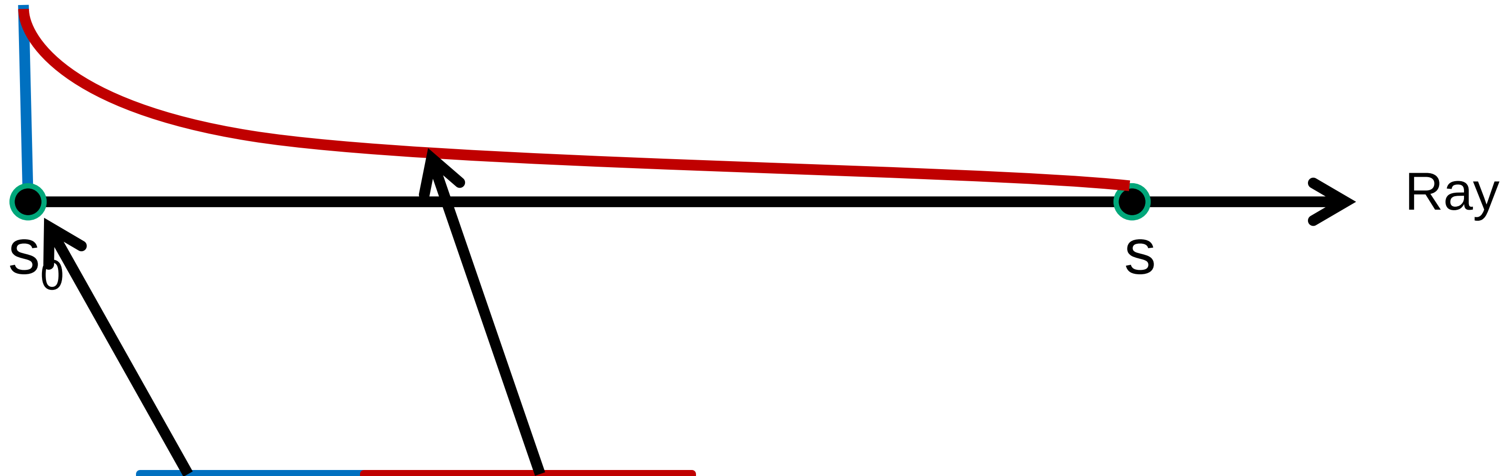
$$I(s) = I(s_0)\mathrm{e}^{-\tau(s_0,s)}$$

Emission at s_0

Attenuation along s_0-s

Optical depth τ
Absorption κ

$$\tau(s_1, s_2) = \int_{s_1}^{s_2} \kappa(s)\, ds$$



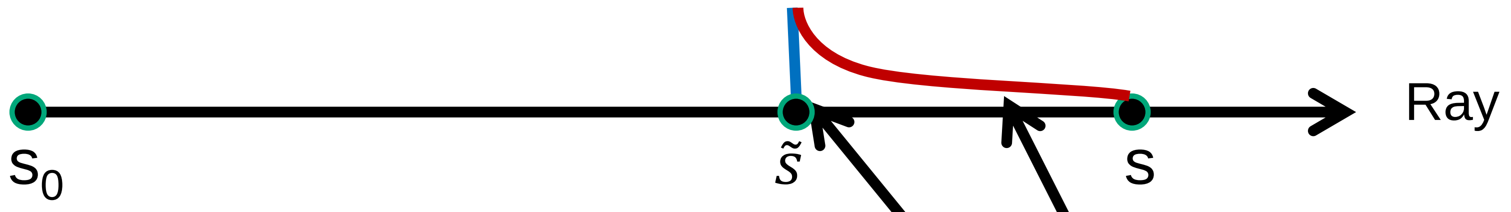
$$I(s) = I(s_0)e^{-\tau(s_0,s)}$$

Initial intensity at s_0

Attenuation along s_0-s

Optical depth τ
Absorption κ

$$\tau(s_1, s_2) = \int_{s_1}^{s_2} \kappa(s)\, ds$$



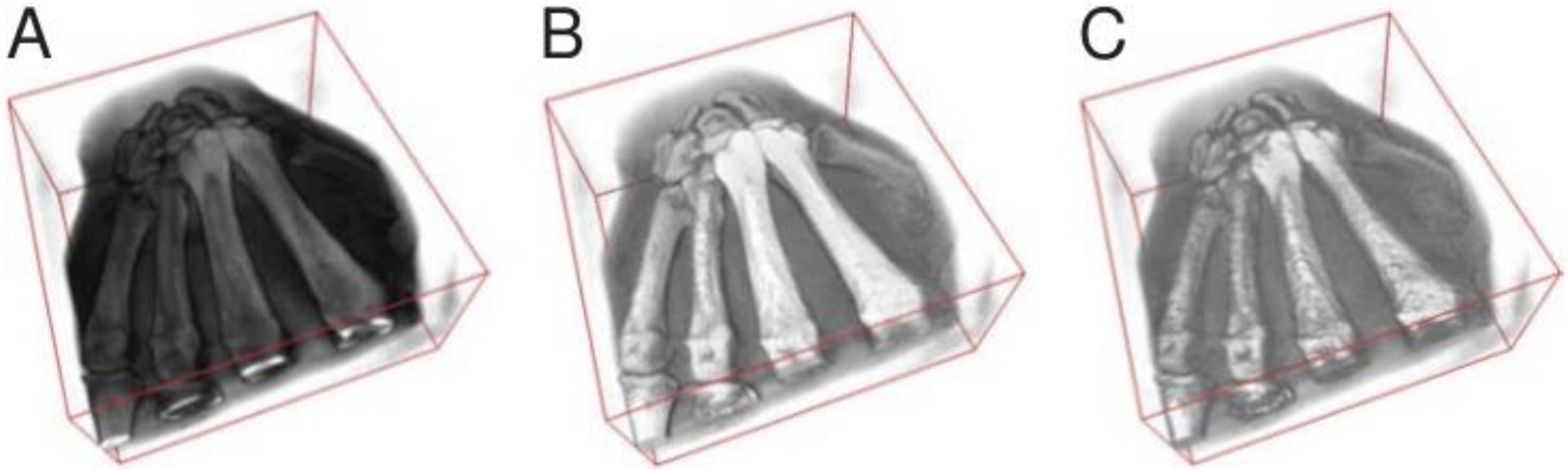
$$I(s) = I(s_0)\mathrm{e}^{-\tau(s_0,s)} + \int_{s_0}^{s} q(\tilde{s})\, \mathrm{e}^{-\tau(\tilde{s},s)}\, d\tilde{s}$$

Emission at $\tilde{s}$

Attenuation along $\tilde{s}$ -s

$$I(s) = I(s_0)e^{-\tau(s_0,s)} + \int_{s_0}^{s} q(\tilde{s})e^{-\tau(\tilde{s},s)}d\tilde{s}$$

Integrate contributions of all points along the ray

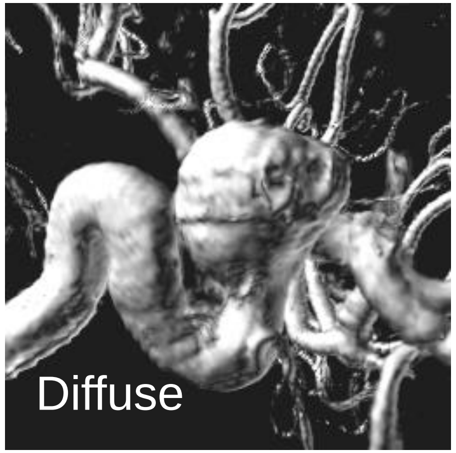


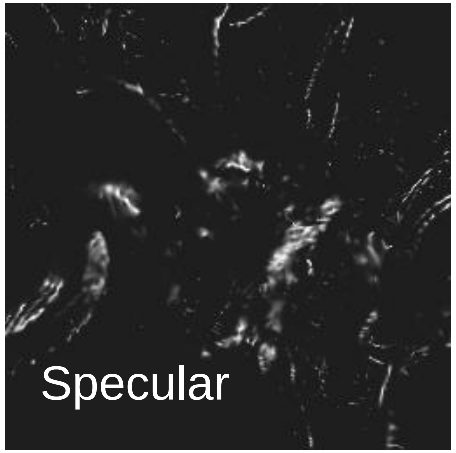
A: Pure emission + absorption, no illumination

B: Same with diffuse lighting

C: Same with specular lighting

Image Source: Markus Hadwiger and Christof Rezk Salama





$K_a = 0.1$

$K_d = 0.5$

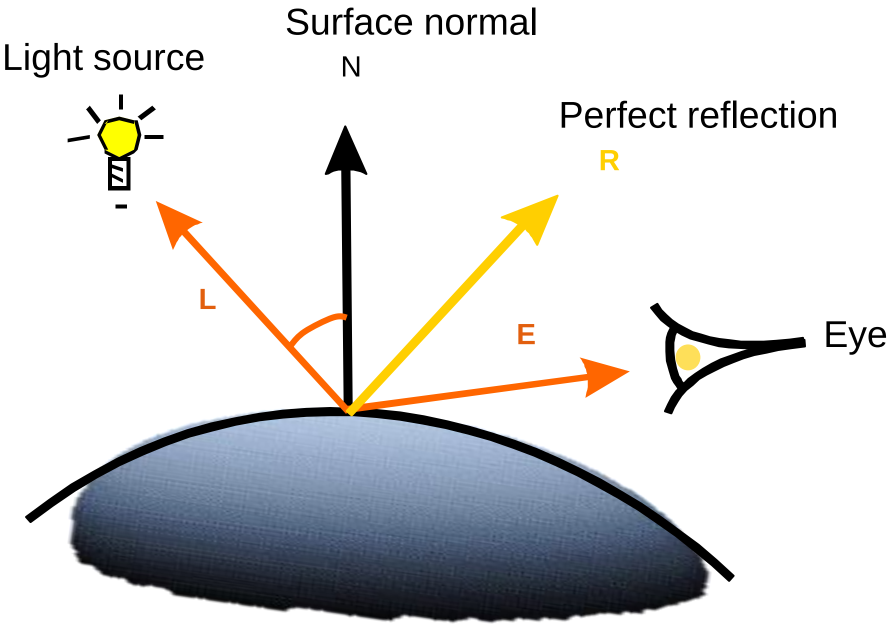
$K_s = 0.4$

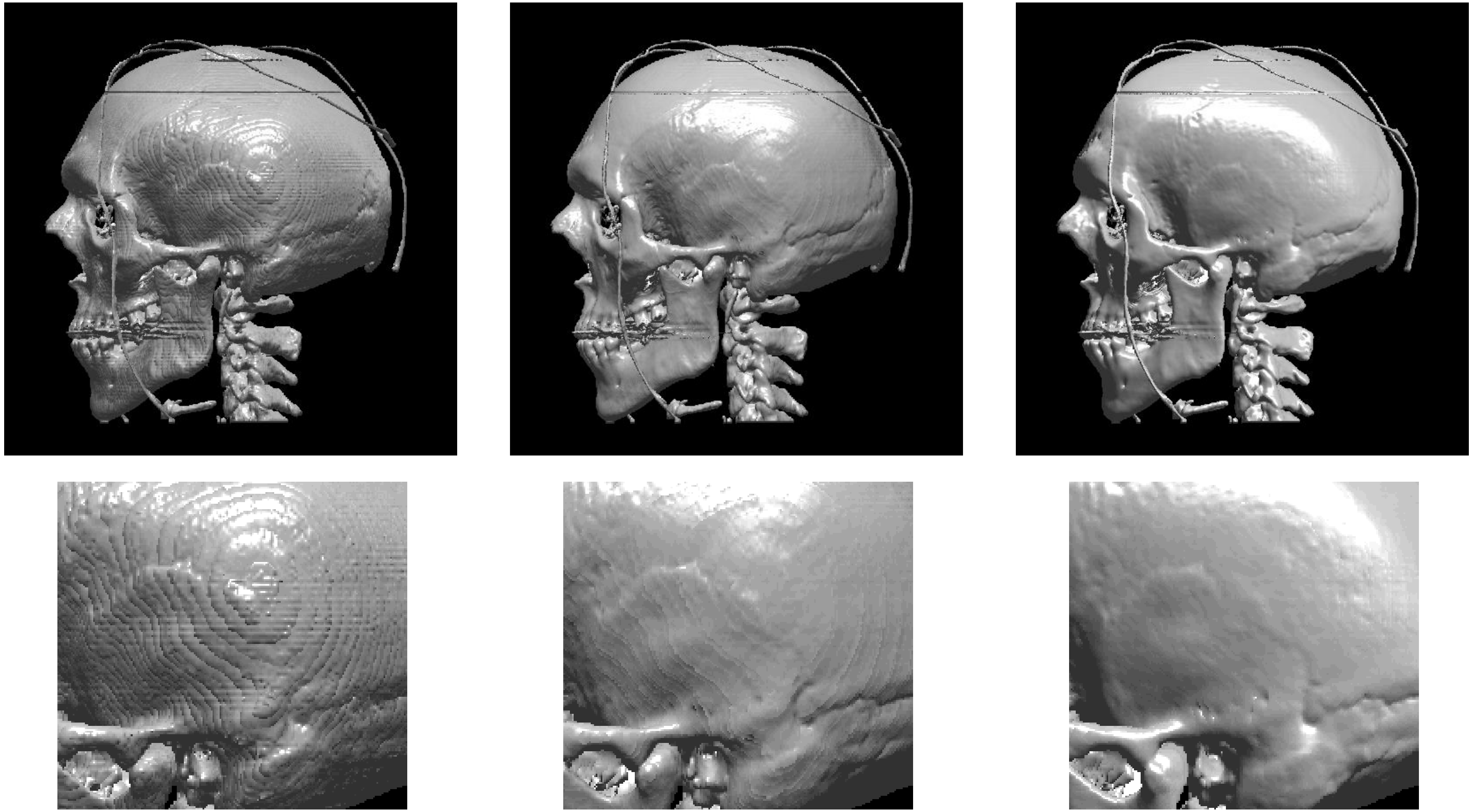
What is the normal vector in a scalar field f(**x**)?

Gradient ∇f(**x**) is perpendicular to isosurface!

Numerical computation of the gradient:

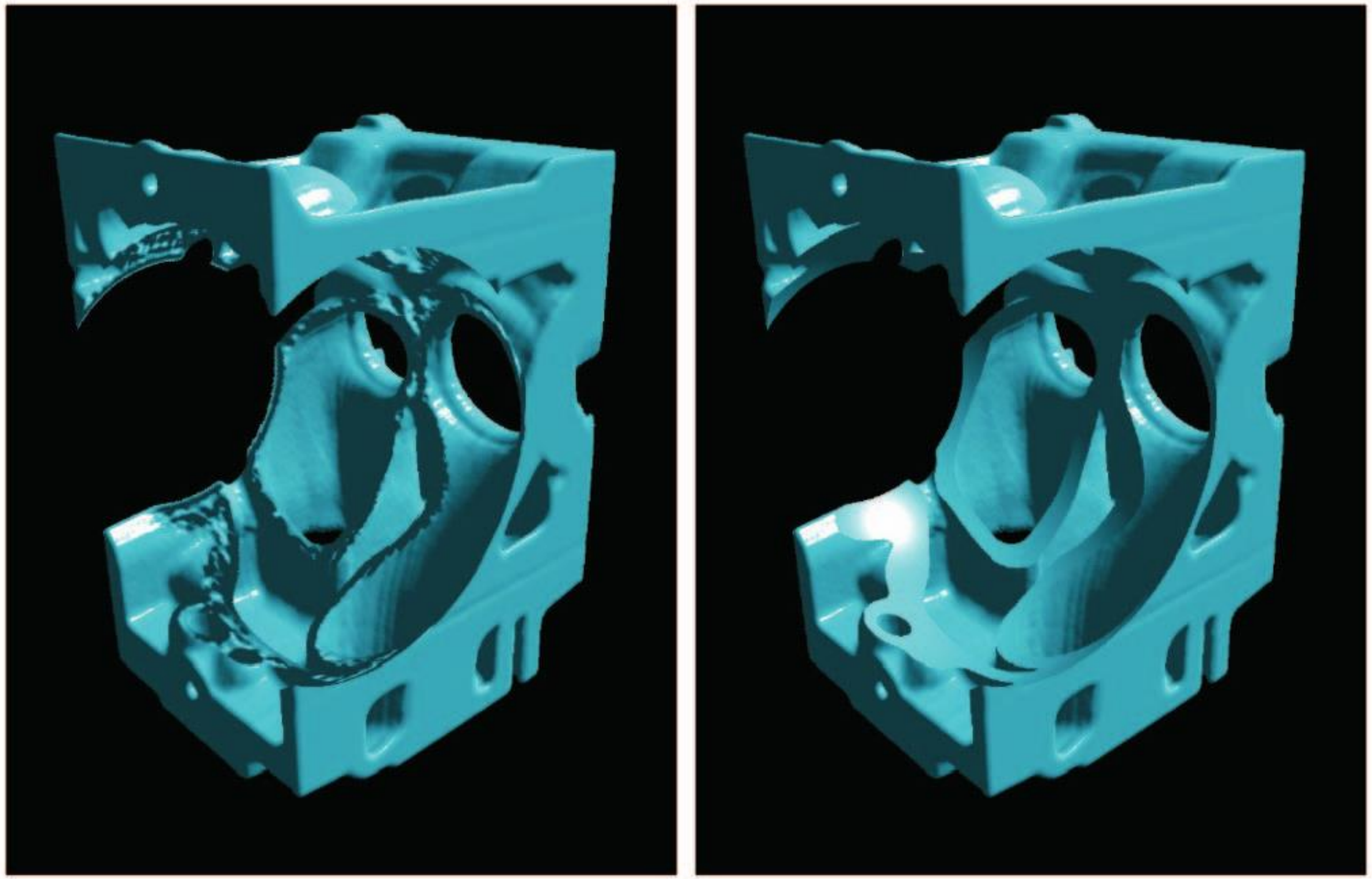
- forward/backward differences
- central differences
- Sobel Operator





Forward/Backward differences   Central differences   Sobel operator

- Parts of the volume can be made transparent by specifying clipping geometry.
- At its boundary, the surface normal of the clipping geometry, rather than the gradient of the volume, should be used for illumination:



(a) (b)

Image Source:
Weiskopf et al., TVCG 2003

very important

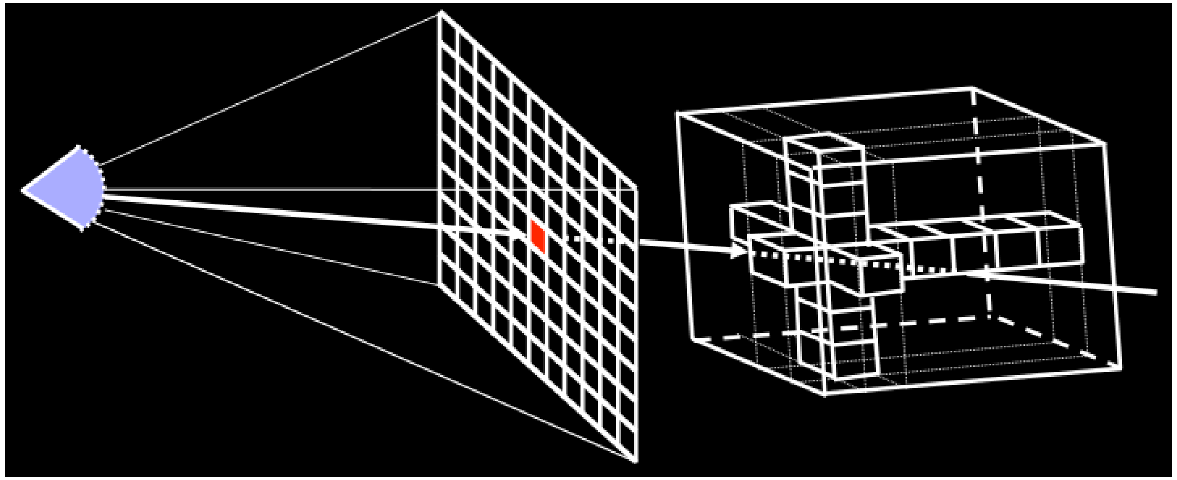
$$I(s) = I(s_0)e^{-\tau(s_0,s)} + \int_{s_0}^{s} q(\tilde{s})e^{-\tau(\tilde{s},s)}d\tilde{s}$$

There is no closed form solution of the integral in general

➔ Approximate the integral with a discrete sum:

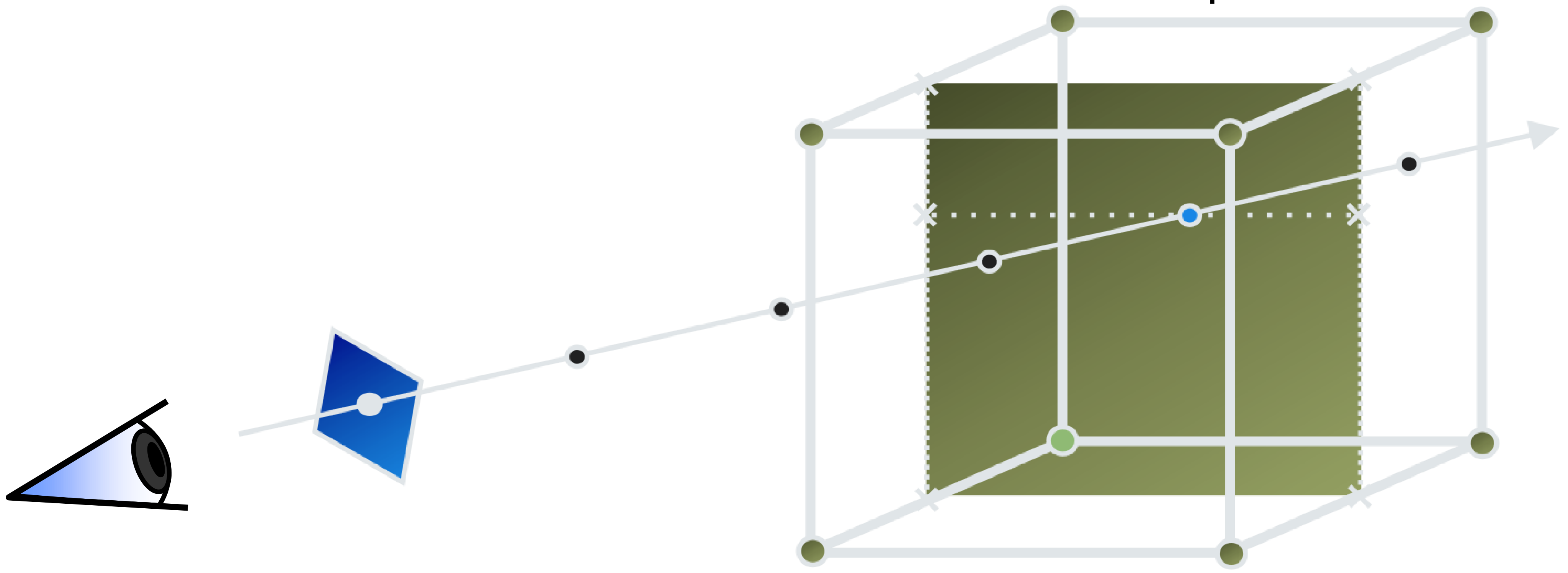
- Discretization: split ray into segments having constant opacity and emission
- Sampling intervals are usually equidistant, but don’t have to be (e.g. importance sampling)
- At each sampling location, a sample is reconstructed from the voxel grid by interpolation

- Similar to ray tracing in surface-based computer graphics
- In volume rendering,
  - we only deal with primary rays; hence: *ray-casting*
  - we composit in each step, rather than searching a ray/surface intersection



- Shot a ray through every pixel on the screen
- Collect color and opacity information along the rays

- Numerical approximation of the volume rendering integral
- Resample volume at equi-spaced intervals along the ray
- Tri-linear interpolation

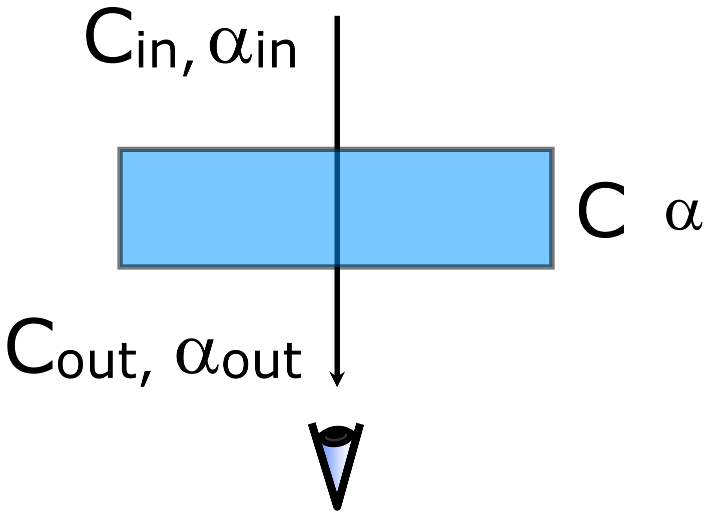


- Opacity and (emissive) color in each cell according to transfer function
- Additional color due to external lighting
- No shadowing, no secondary effects

$$C_{out} = (1 - \alpha)\, C_{in} + \alpha C$$

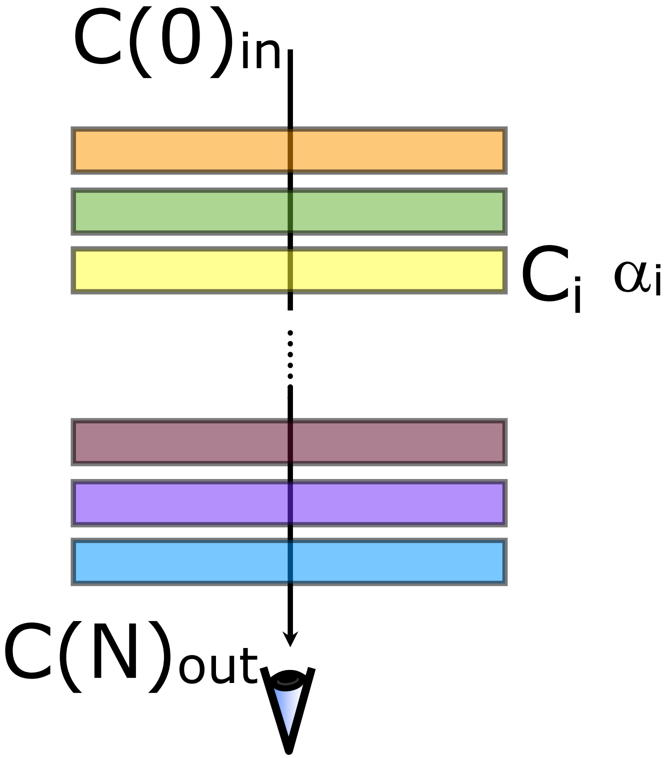
$$\alpha_{out} = (1 - \alpha)\, \alpha_{in} + \alpha$$

Note that the α-computation is not necessary for computing the RGB color. It is therefore often omitted.



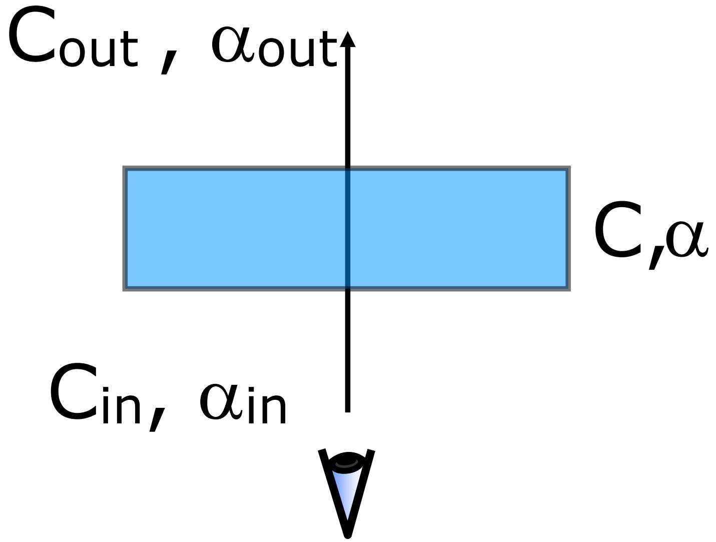
$$C(i)_{in} = C(i\text{-}1)_{out}$$

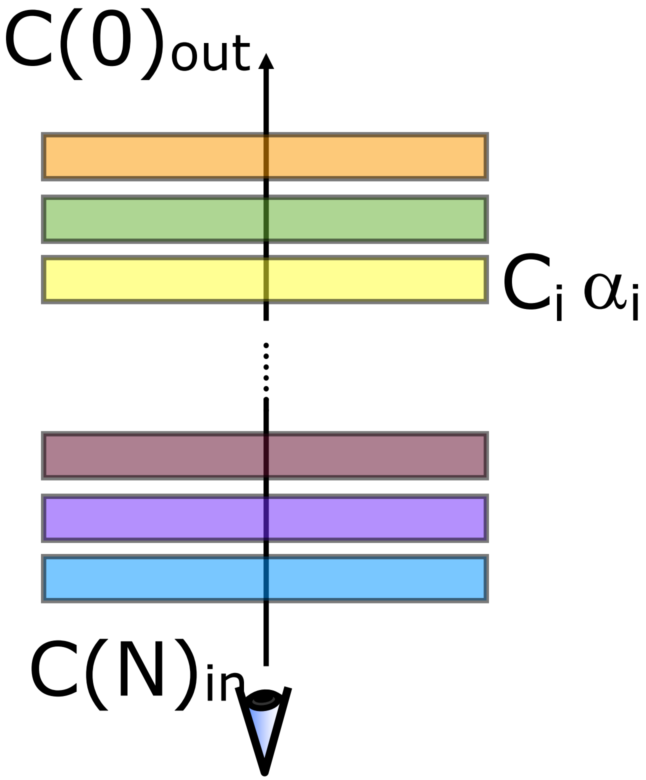
Iterative process. The output color of the previous step becomes the input color of the next step.

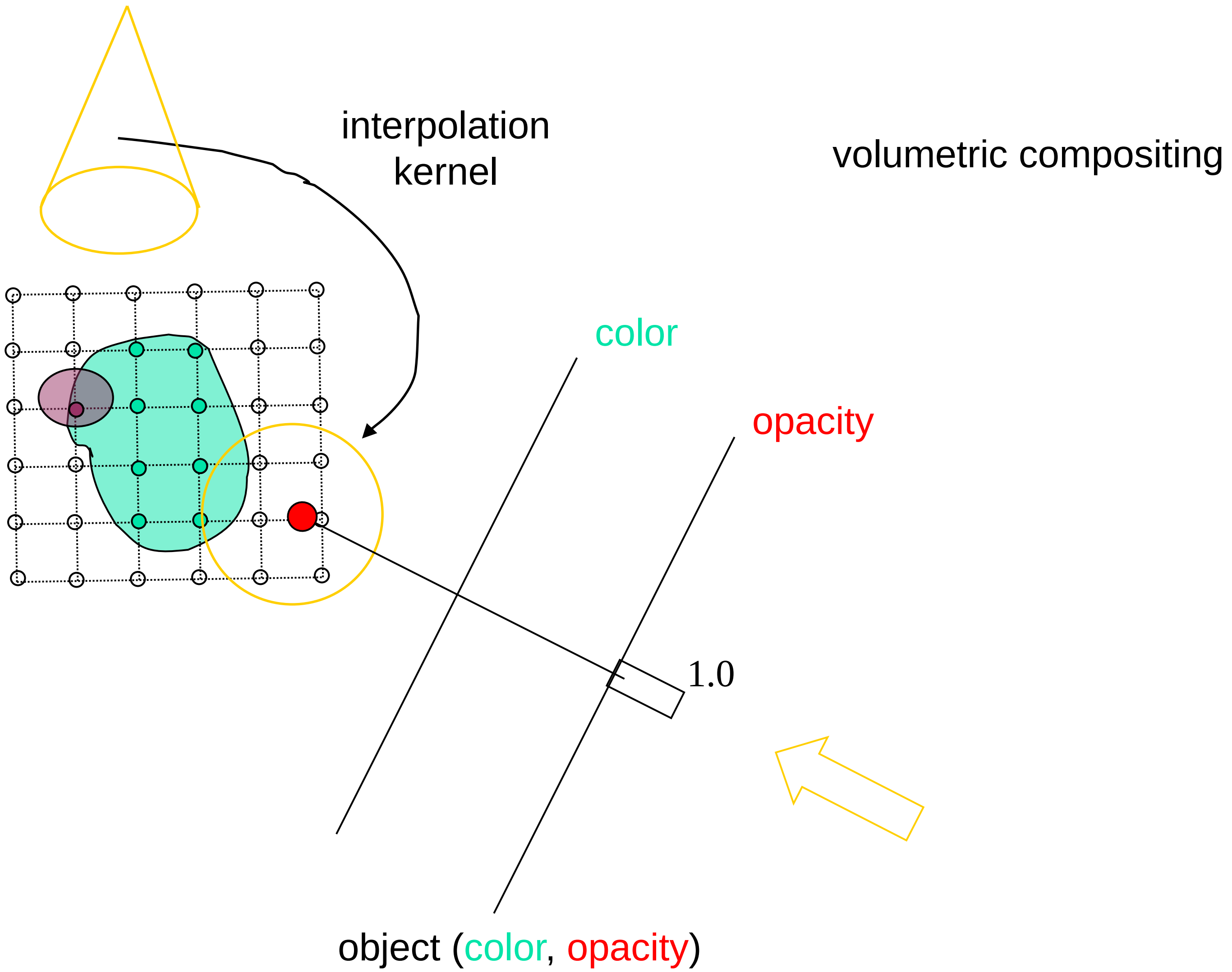


$$C_{out} = C_{in} + (1 - \alpha_{in})\, \alpha C$$

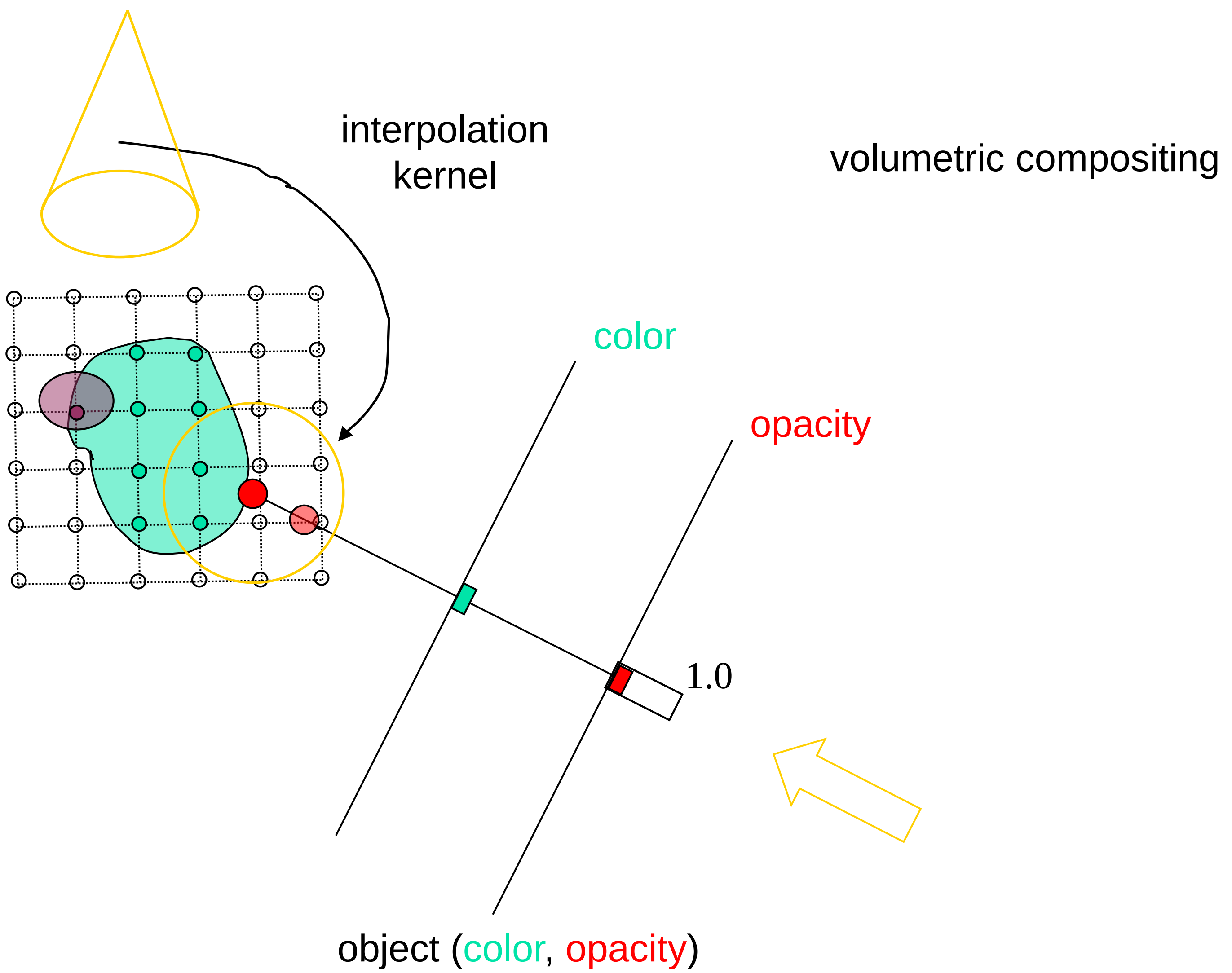
$$\alpha_{out} = \alpha_{in} + (1 - \alpha_{in})\, \alpha$$



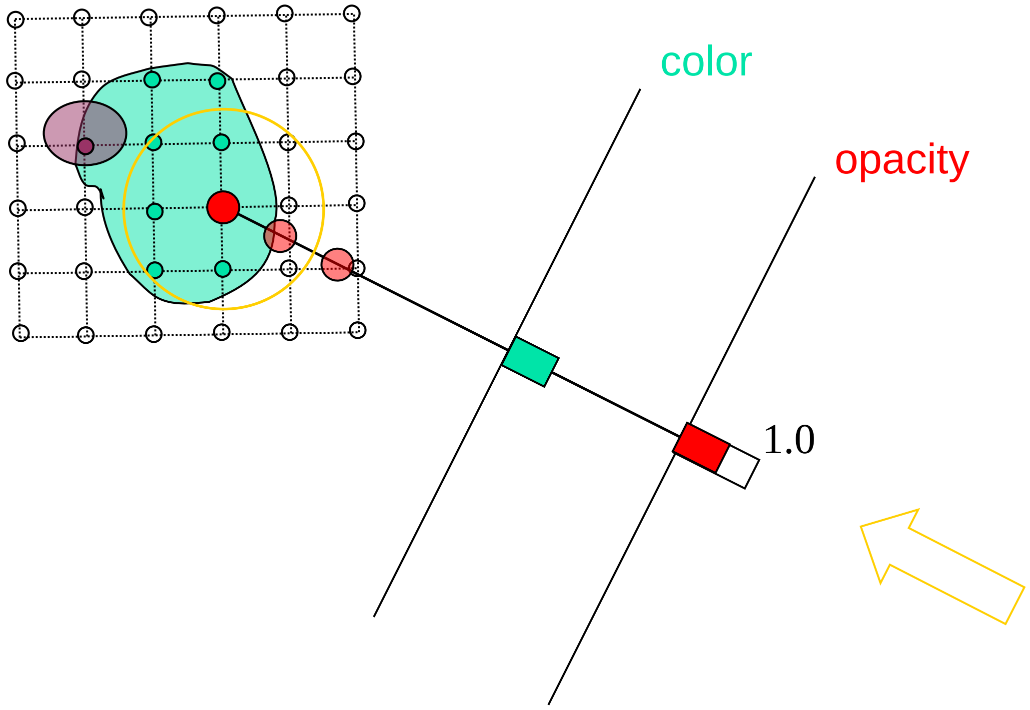




interpolation kernel
volumetric compositing
color
opacity
1.0
object (color, opacity)

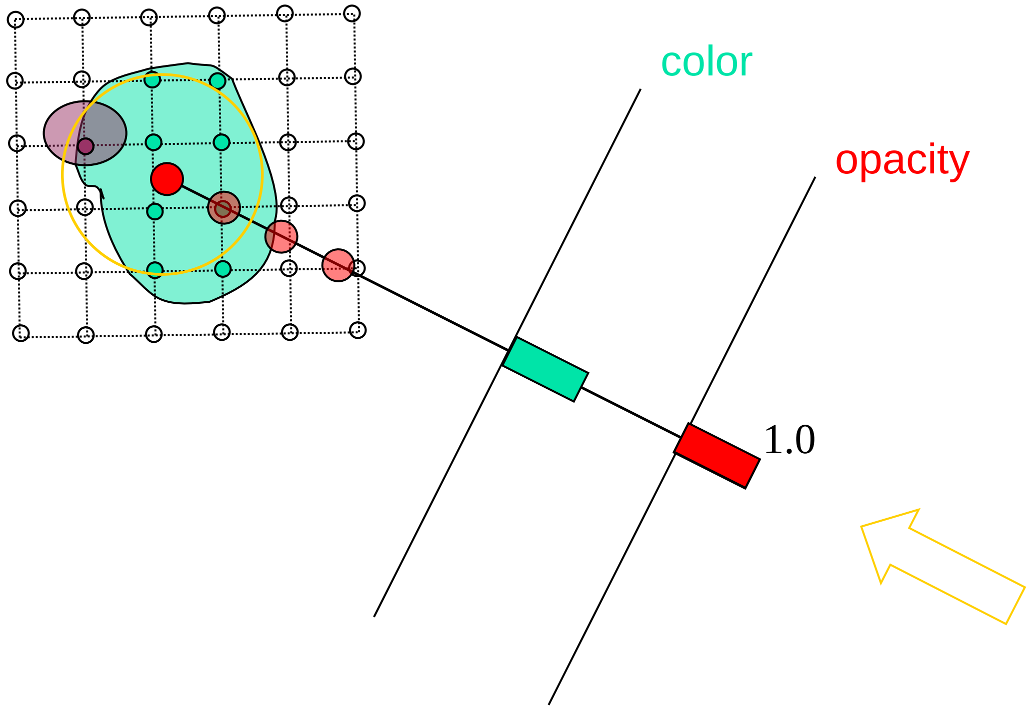

interpolation
kernel
volumetric compositing
color
opacity
1.0
object (color, opacity)

volumetric compositing



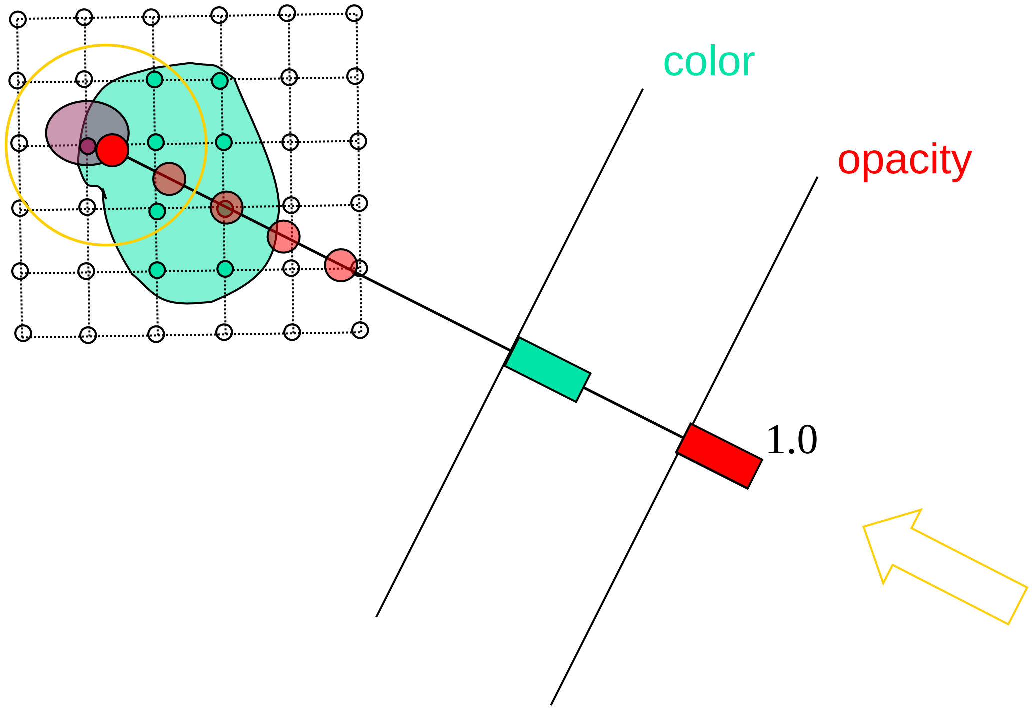
object (color, opacity)

volumetric compositing



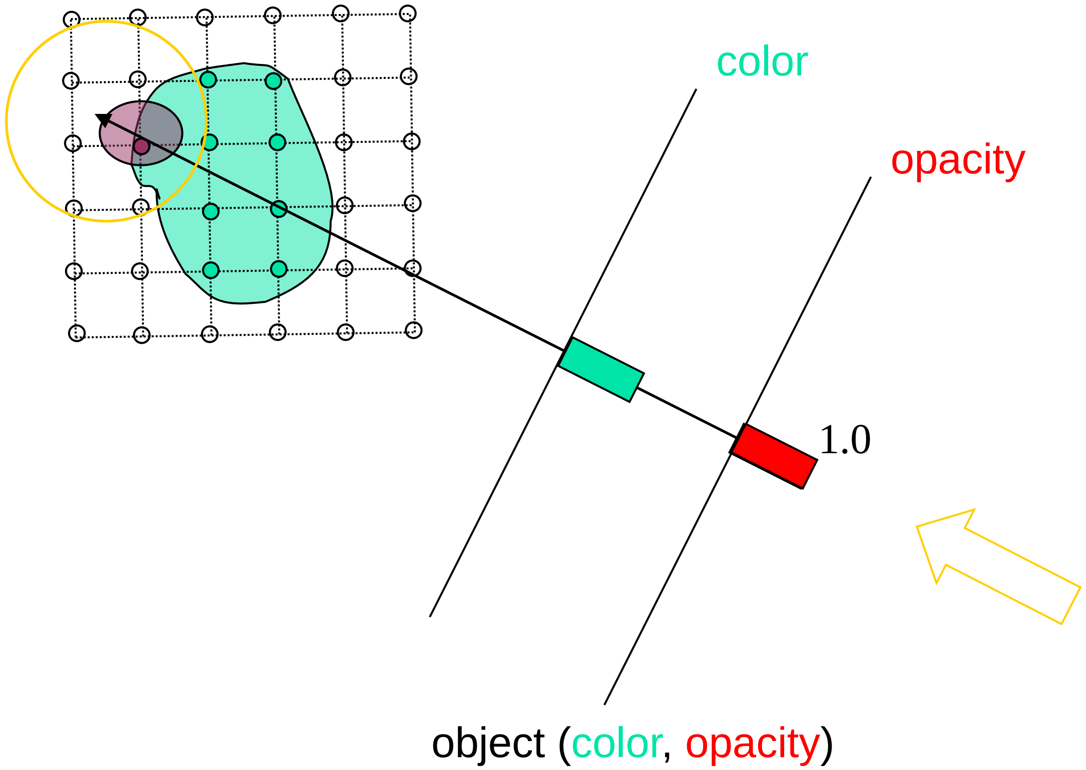
object (color, opacity)

volumetric compositing

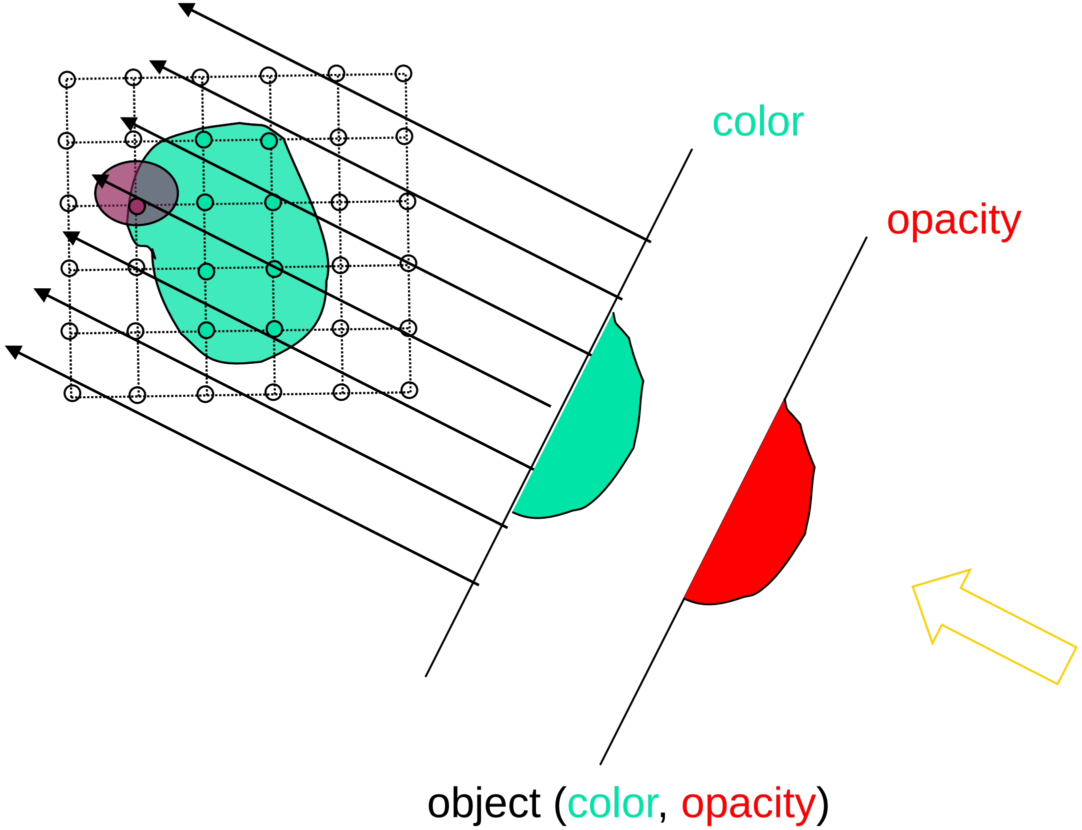


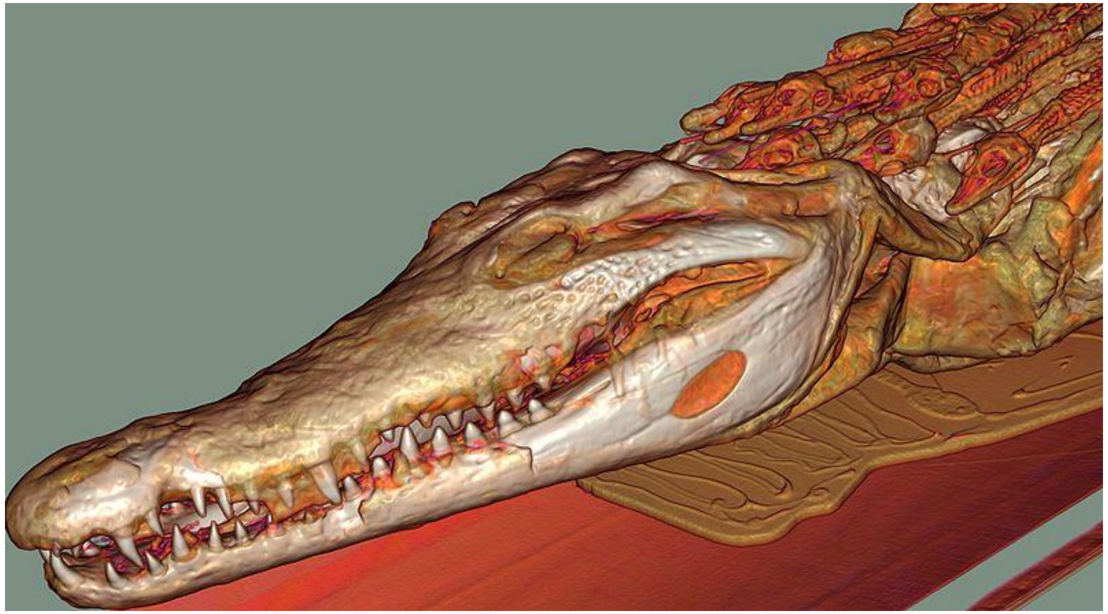
object (color, opacity)

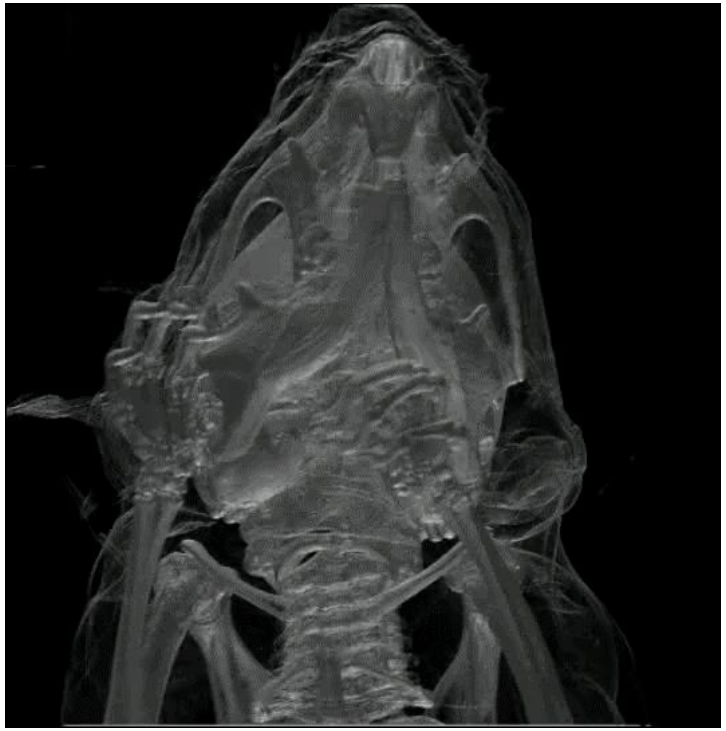
volumetric compositing



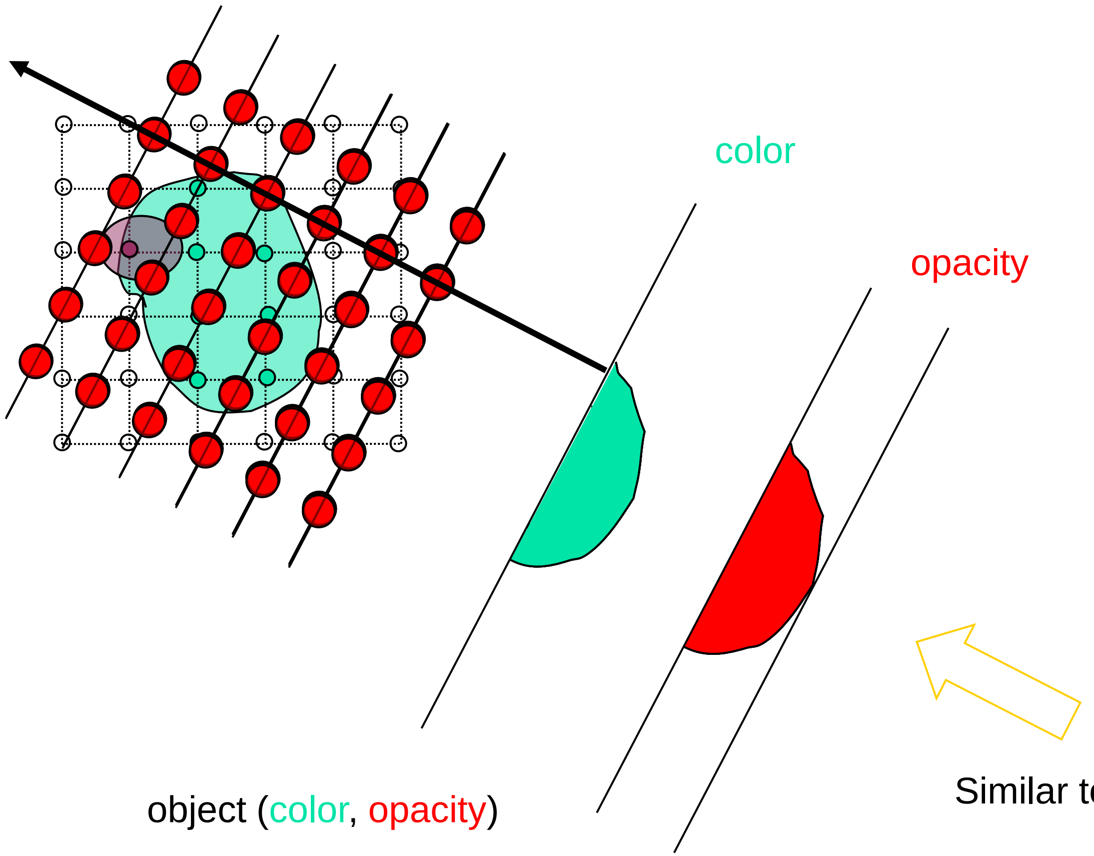
volumetric compositing



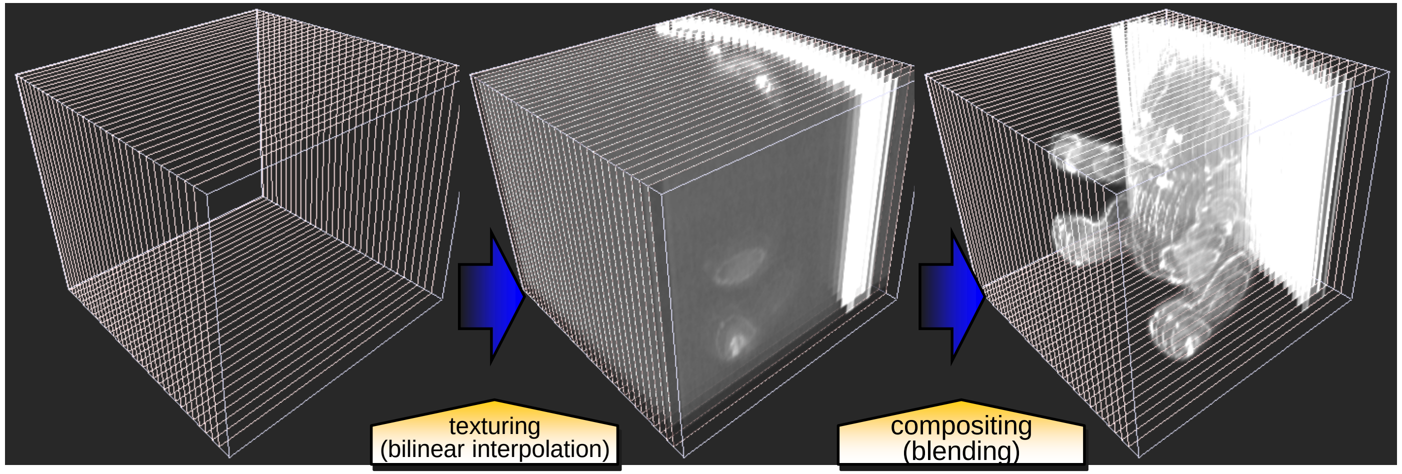




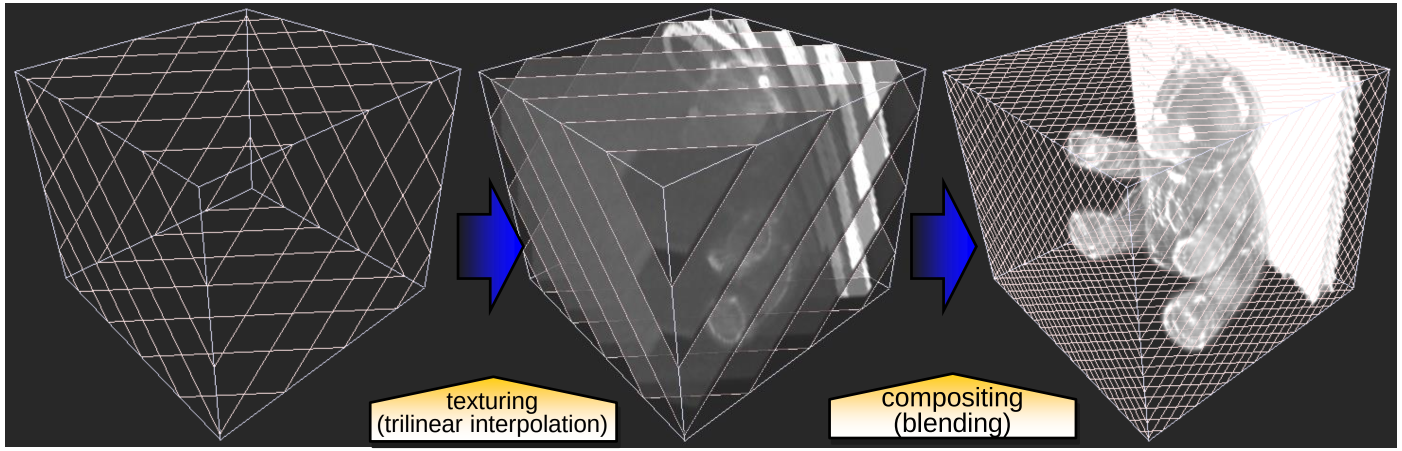
- Slice-based rendering



Similar to ray casting with simultaneous rays



2D textures
axis-aligned

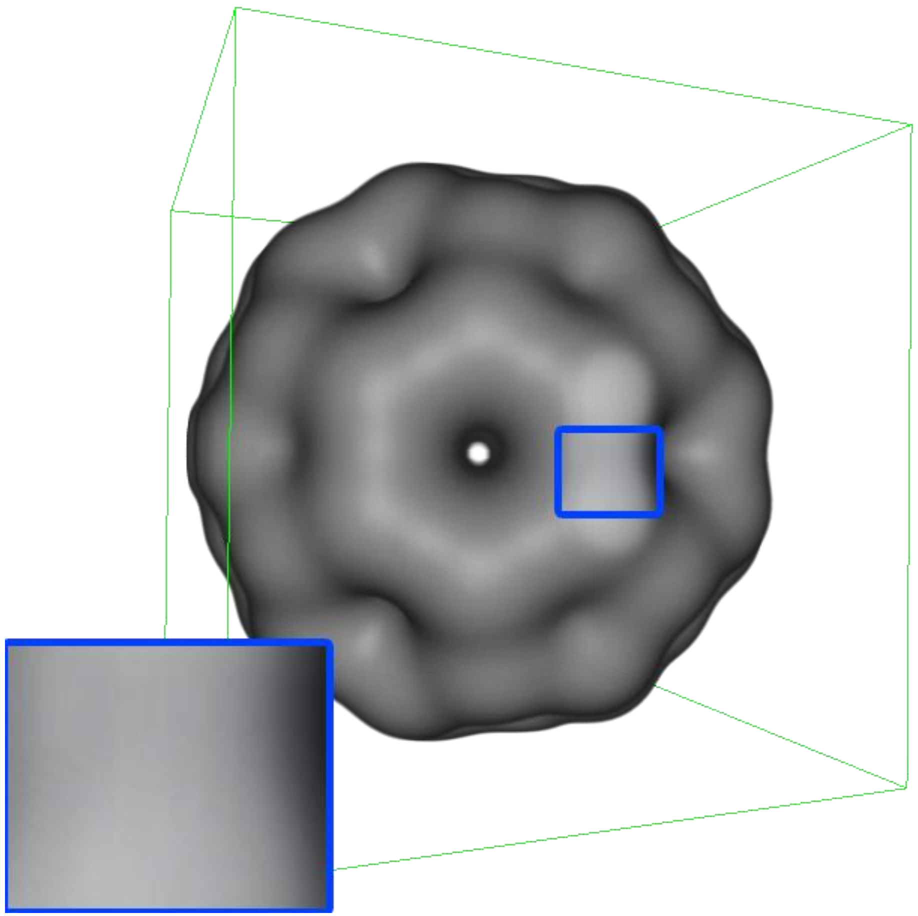


3D texture
view-aligned

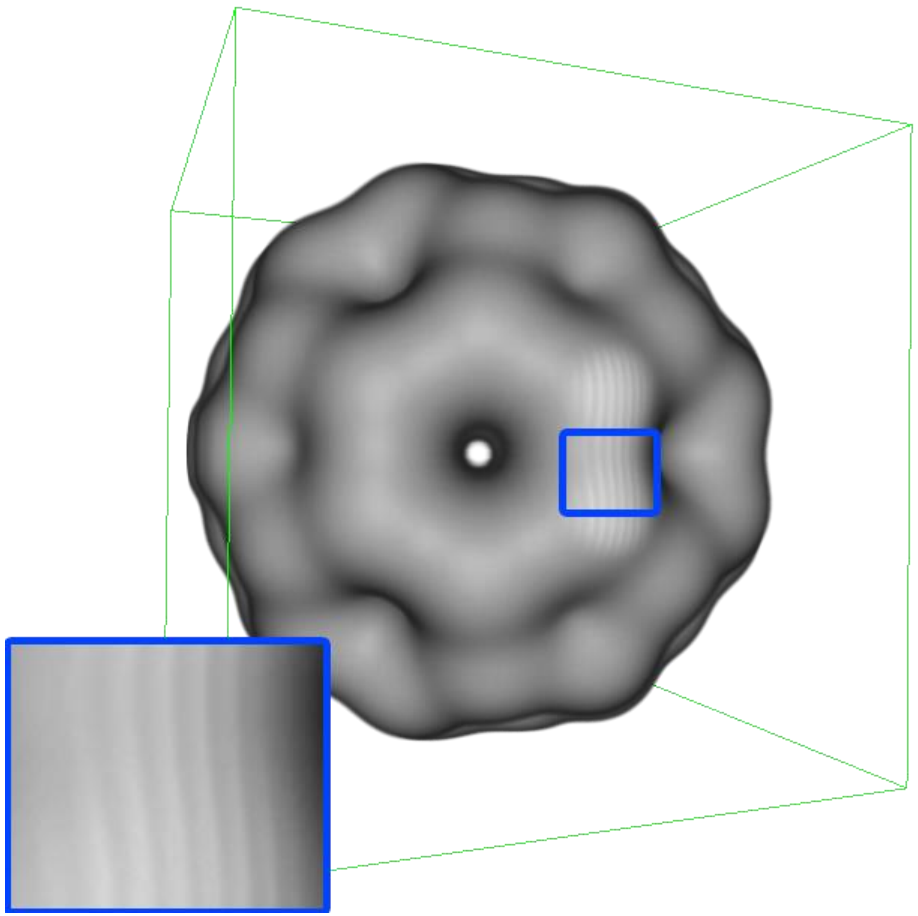
Image Source: Klaus Engel

Artifacts when stack is viewed close to 45 degrees

On modern (non-mobile) GPUs, 3D textures are the norm and are most commonly used.

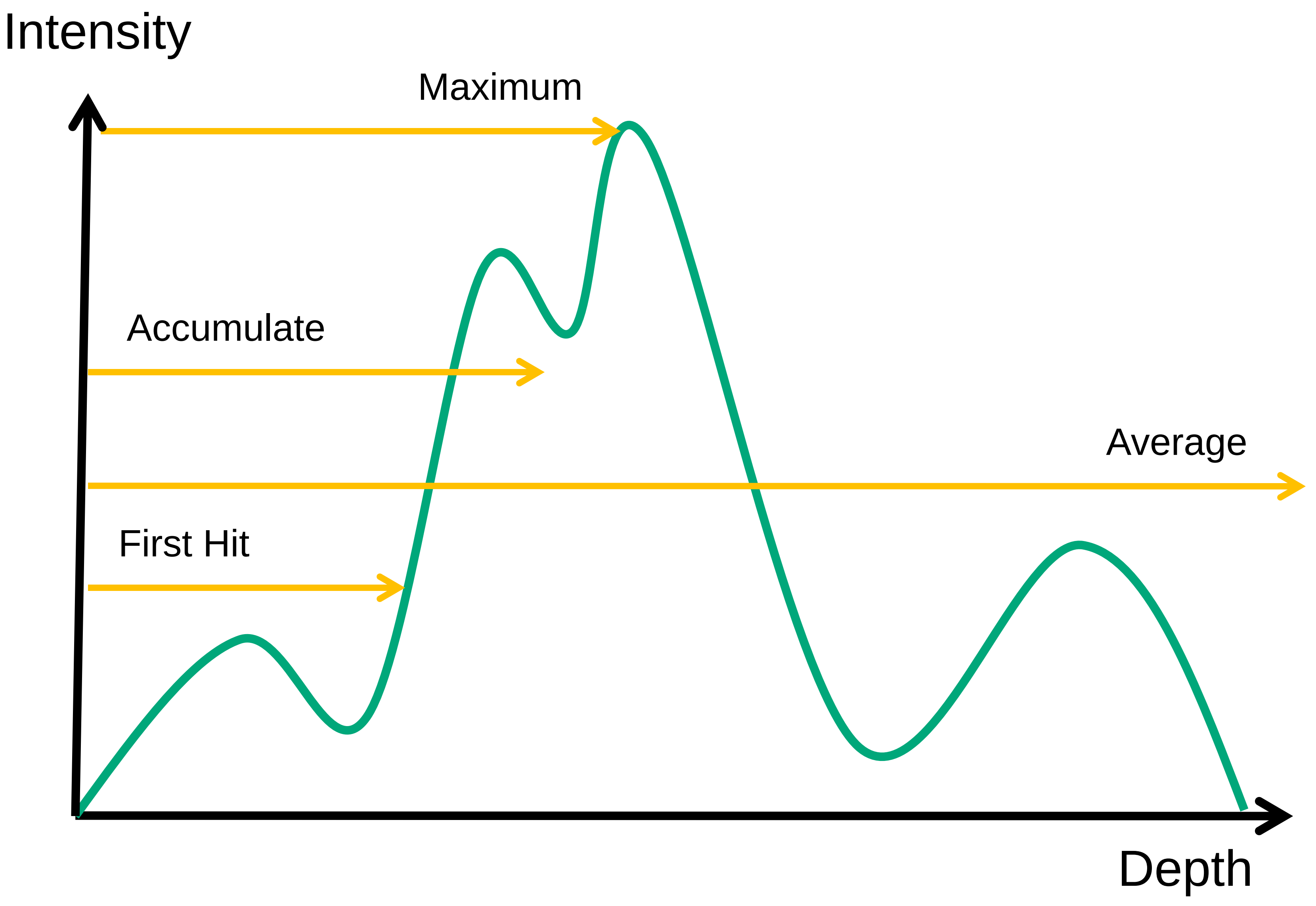


Ray Casting

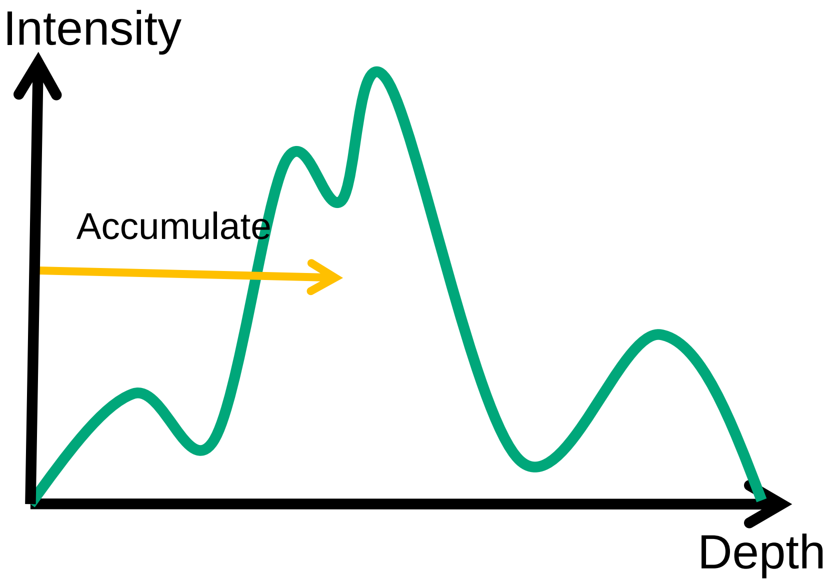


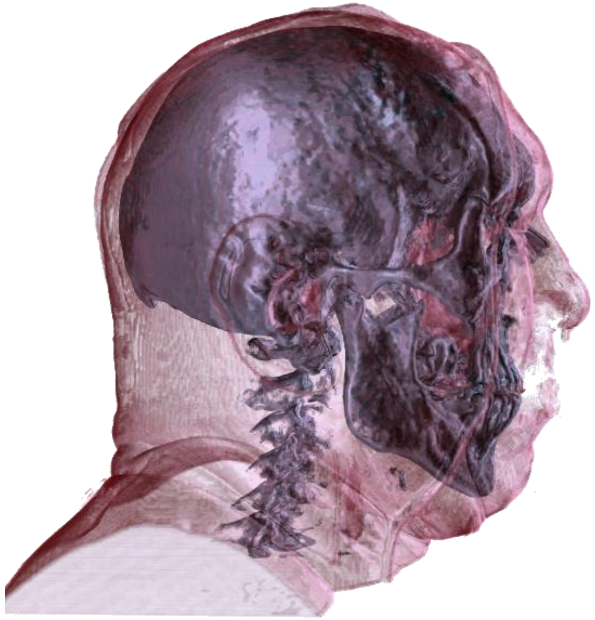
View-Aligned Slicing

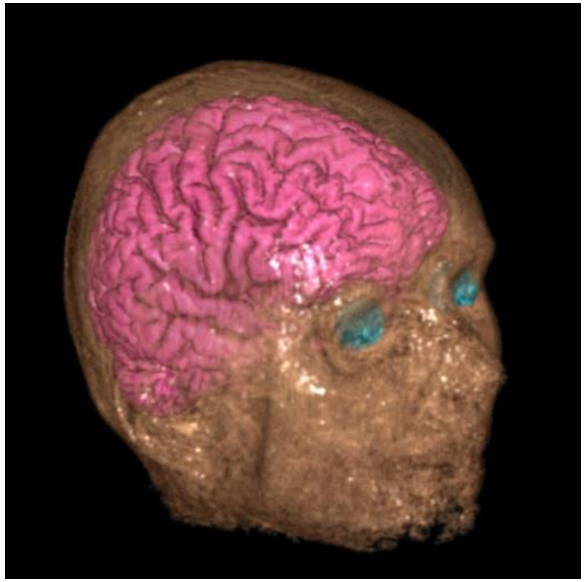
very important



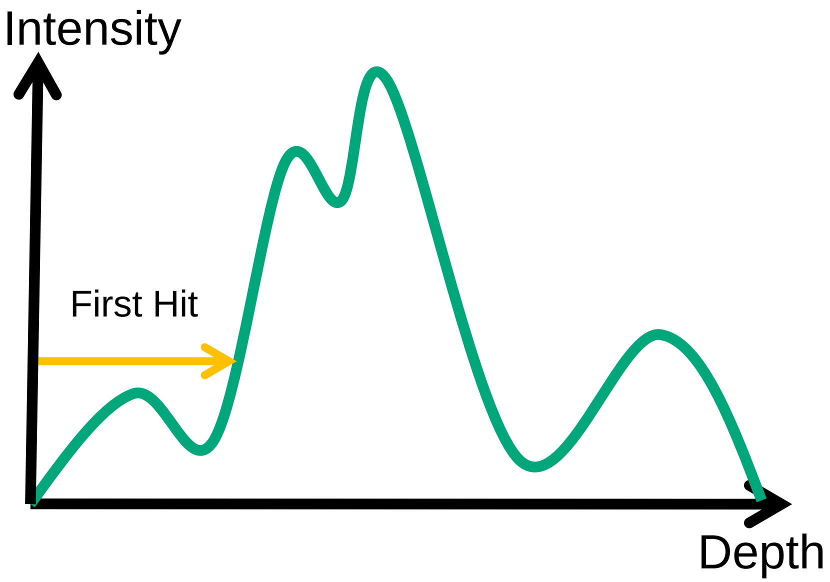
Accumulate: Standard Volume Rendering

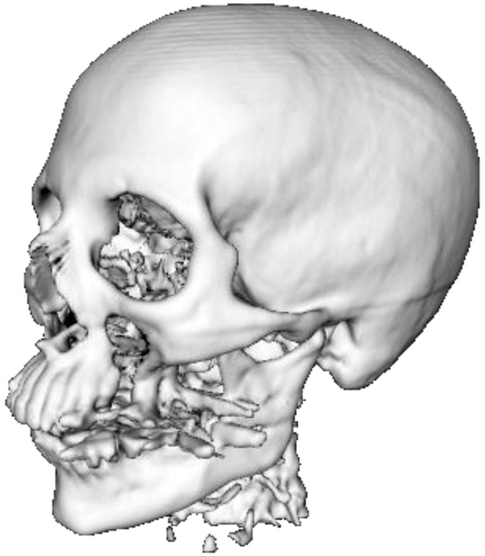




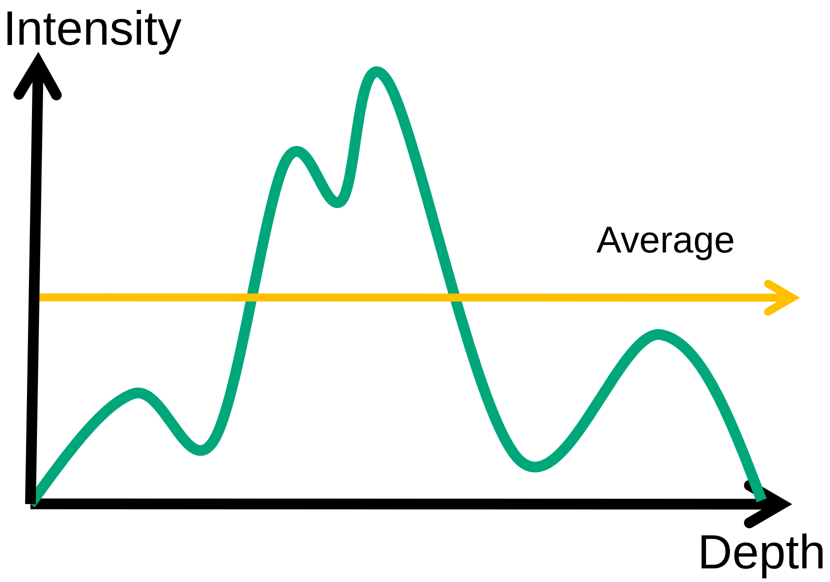


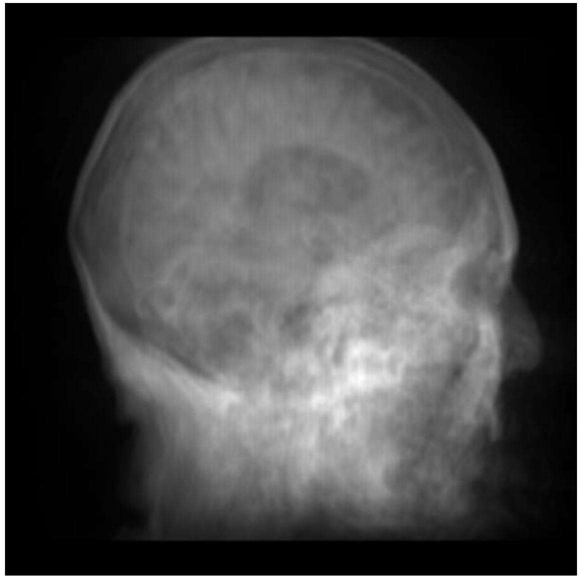
First Hit: Renders an Isosurface





Averaging: Produces X-ray(-like) images





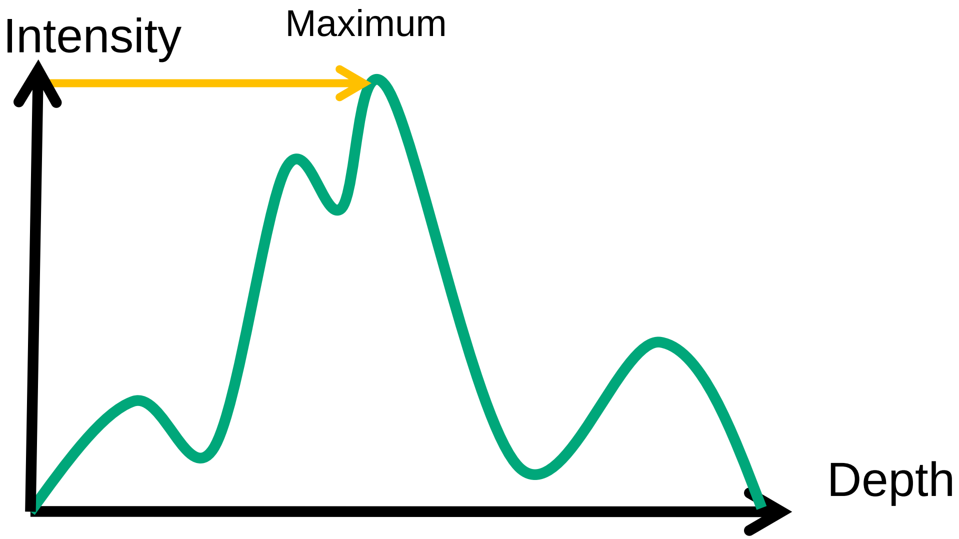
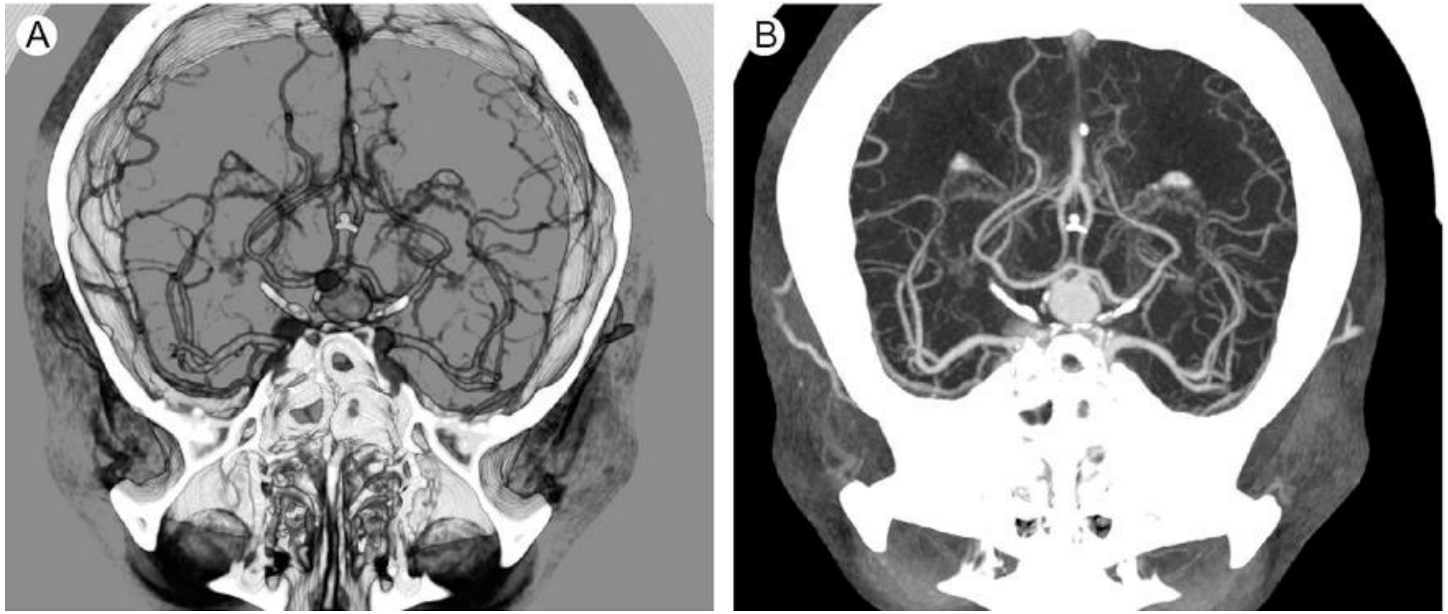
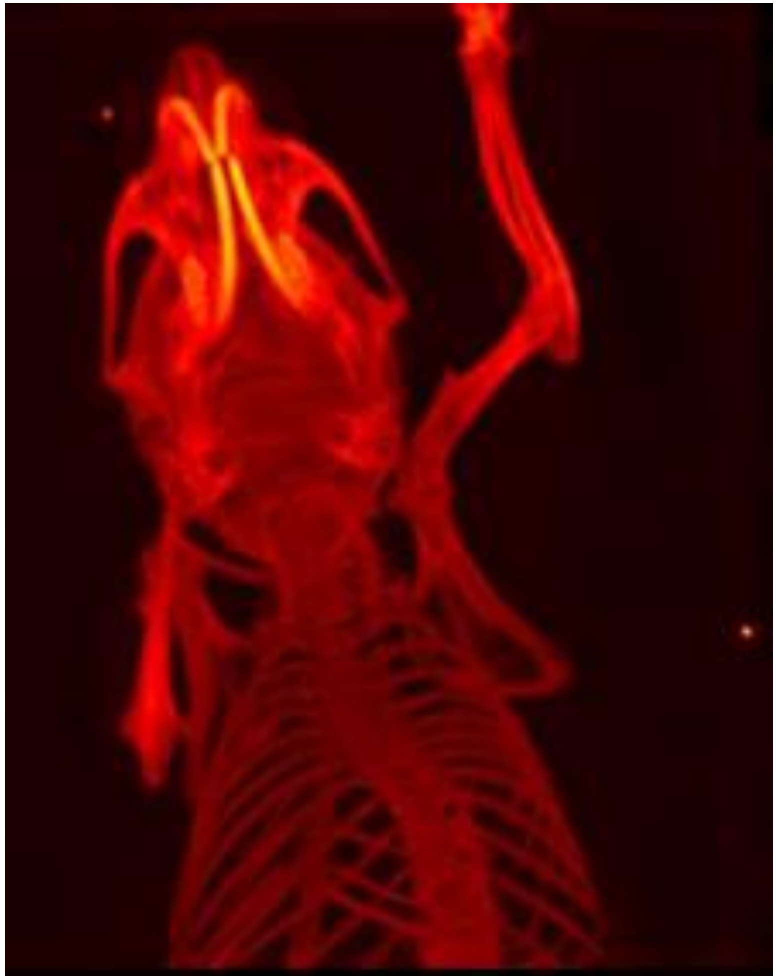


Image Source: Thomas Fogal

Often used to extract vessel structures in magnetic resonance angiograms



Maximum Intensity Projection (MIP) Example



Increasing stepsize in ray casting, or reducing the number of slices reduces the computational effort, but can lead to artifacts in the visualization.

Common trick: Increase stepsize to allow for a more fluent interaction (e.g., changing the viewpoint), produce a high-quality still image afterwards.

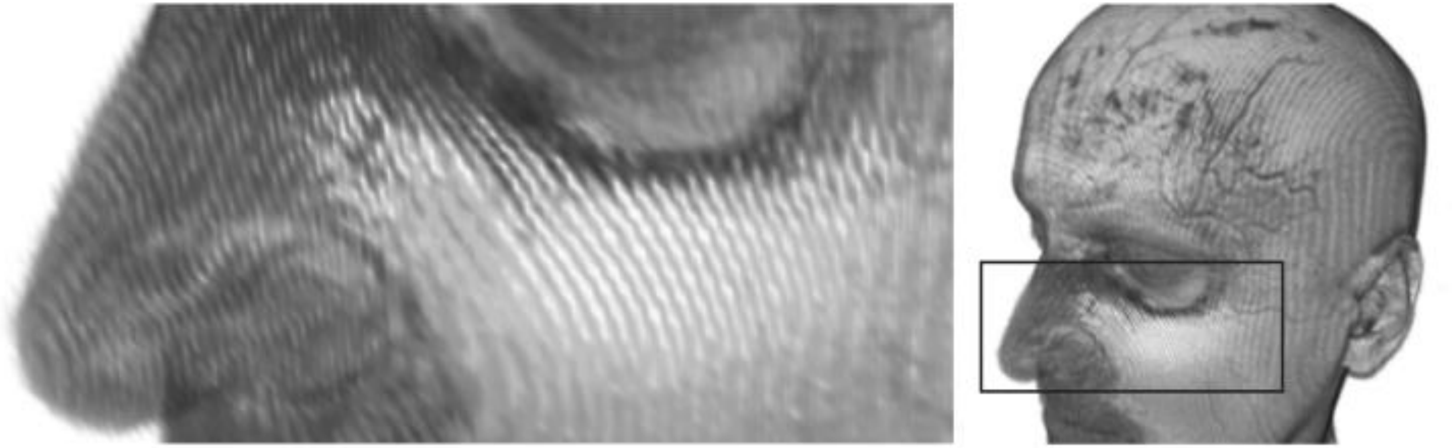
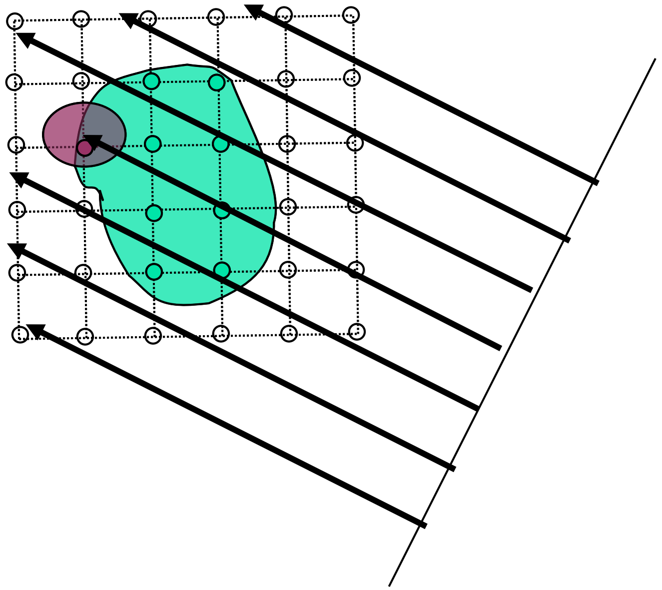


Image Source: Hadwiger and Rezk Salama, 2004

- **Problem:** ray casting can be time consuming
- **Idea:**
  - Neglect „irrelevant“ information to accelerate the rendering process
  - Exploit coherence

- Early-ray termination
  - Idea: colors from distant regions do not contribute if accumulated opacity is too high
  - Stop traversal if contribution of sample becomes irrelevant
  - Front-to-back compositing

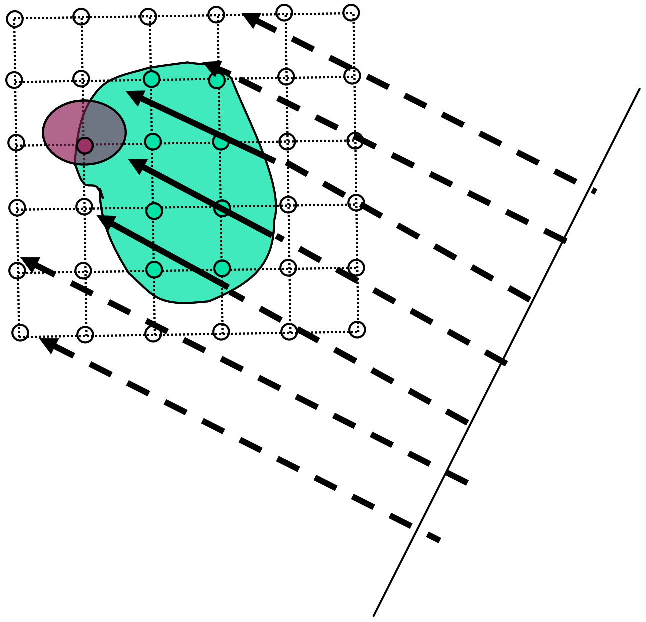


Space leaping

- Skip empty cells

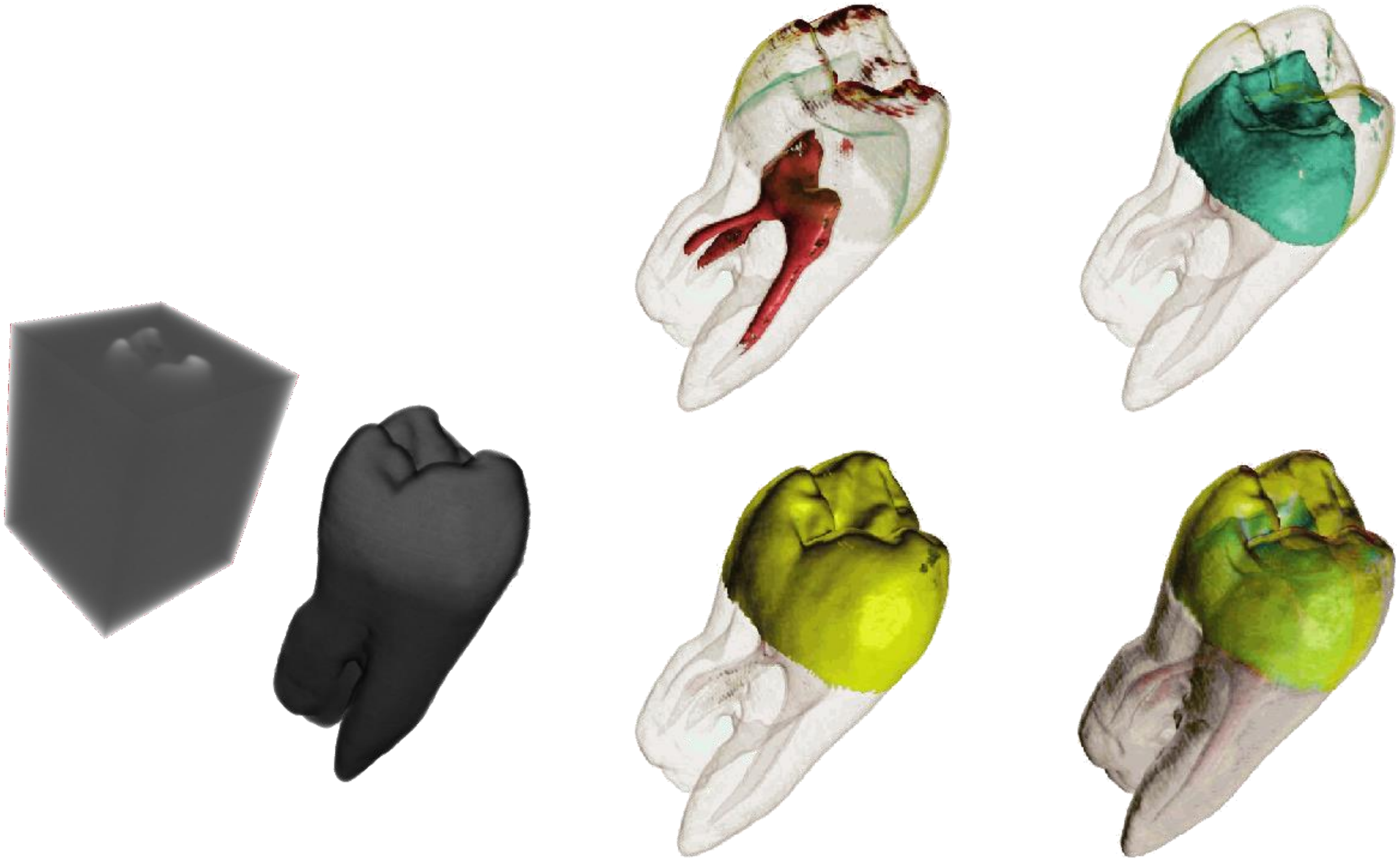
Or: Homogenety acceleration

- Approximate homogeneous regions with fewer sample points

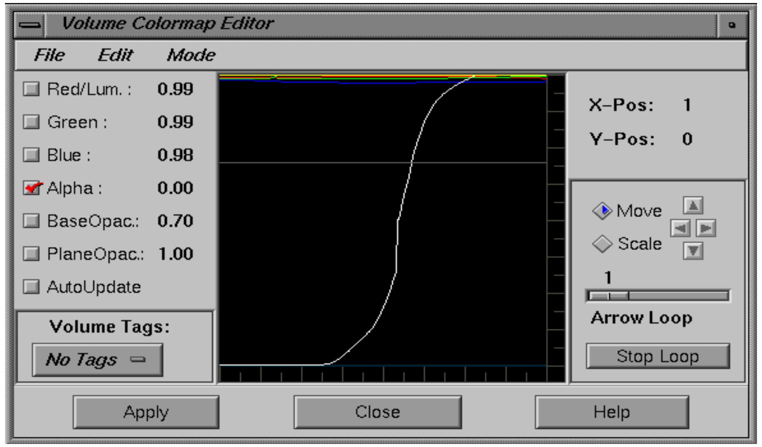


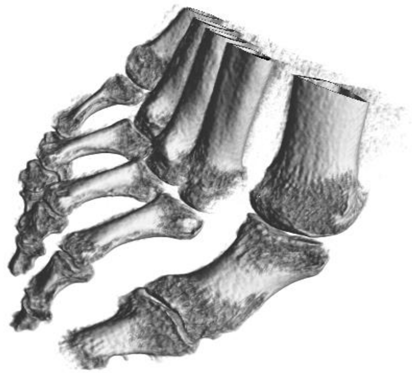
Approaches:

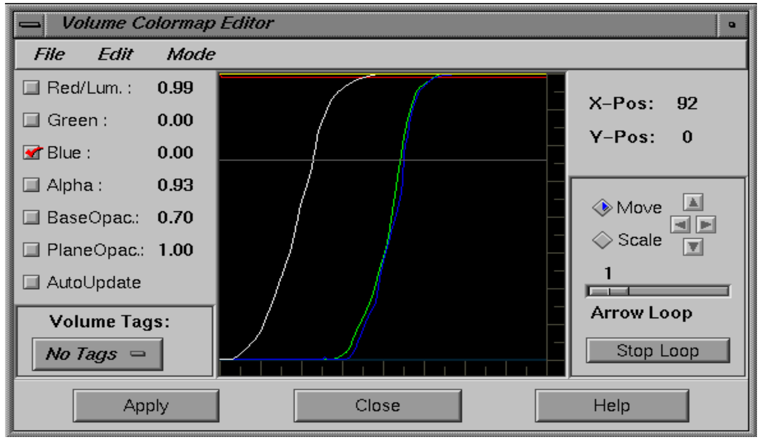
- Hierarchical spatial data structure
- Bounding boxes around objects
- Proximity clouds
- …



Basic Transfer Function Editing







We introduced a **transfer function** as a map of data values to colors:

$$T : \mathbb{R} \rightarrow C$$

More precisely, that transfer function should be seen as the *composition of two functions:*

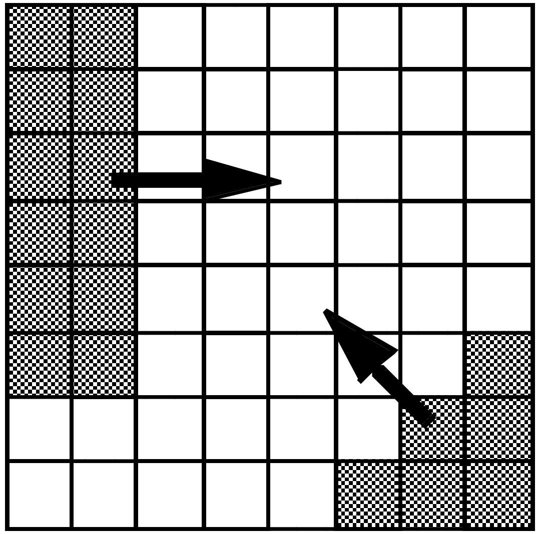
1. Map data value to probabilities of the presence of any of *n* materials at this point: $p : \mathbb{R} \rightarrow [0, 1]^n$

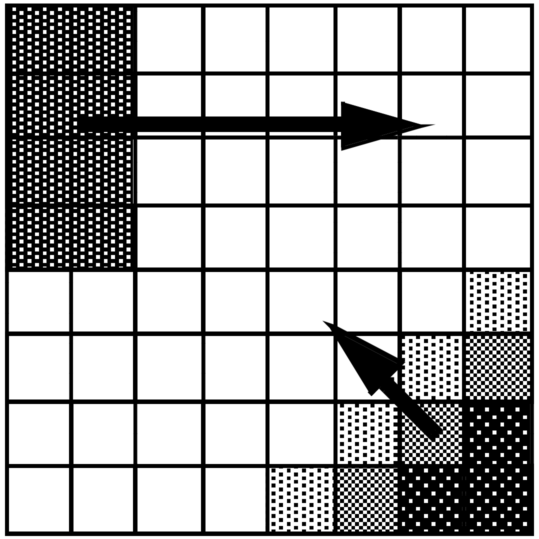
2. Map the probabilities to a color value by assigning a color C_i to each material:

$$c : [0, 1]^n \rightarrow C \qquad c = \sum_{i=1}^{n} p_i C_i$$

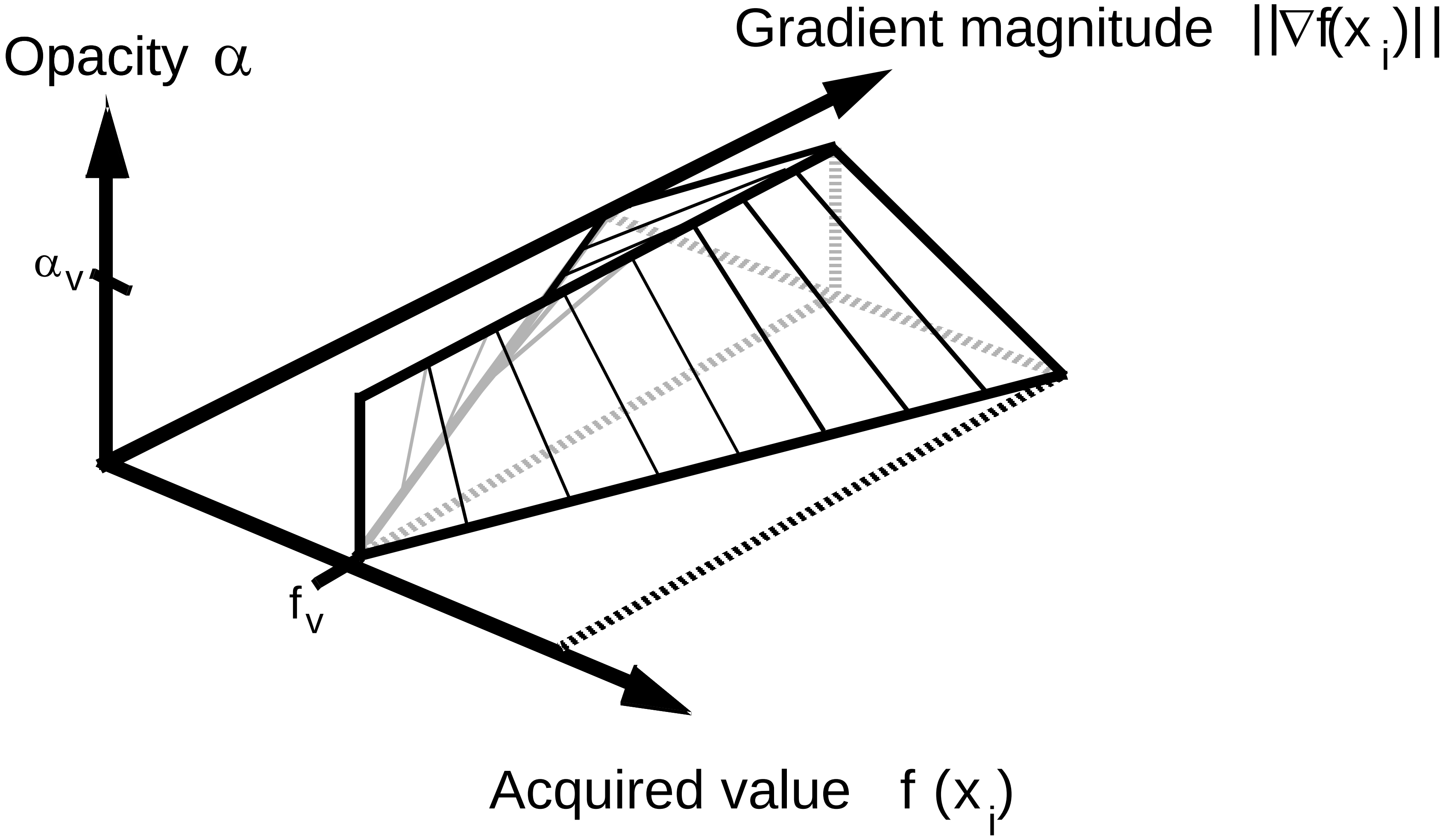
In many cases, rapid changes in data values are important features

- Boundaries between different objects indicated by a large gradient magnitude
- Isosurface “strength”
- May want to assign a high value of opacity in regions of change

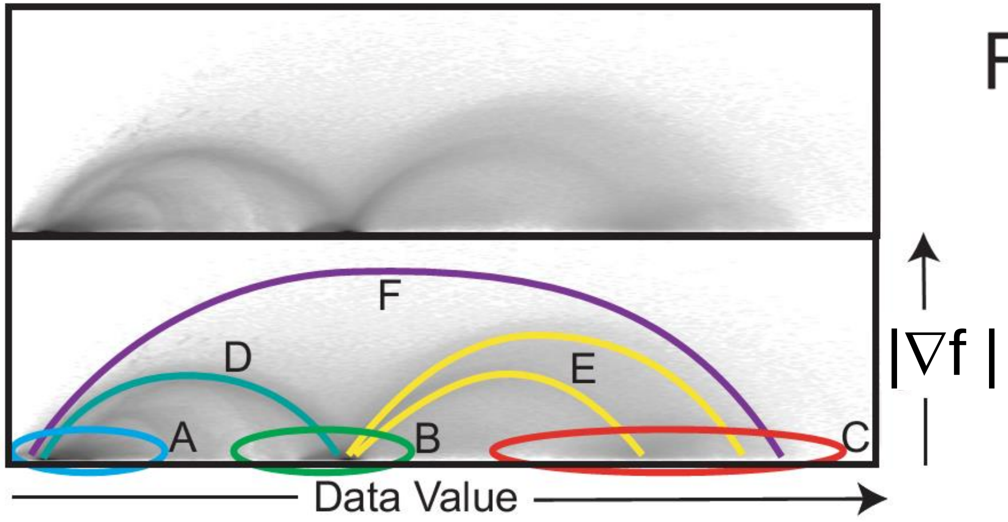




Levoy [1988] suggests to combine scalar value and gradient of the scalar field in a transfer function to render isosurfaces: 2d transfer function.

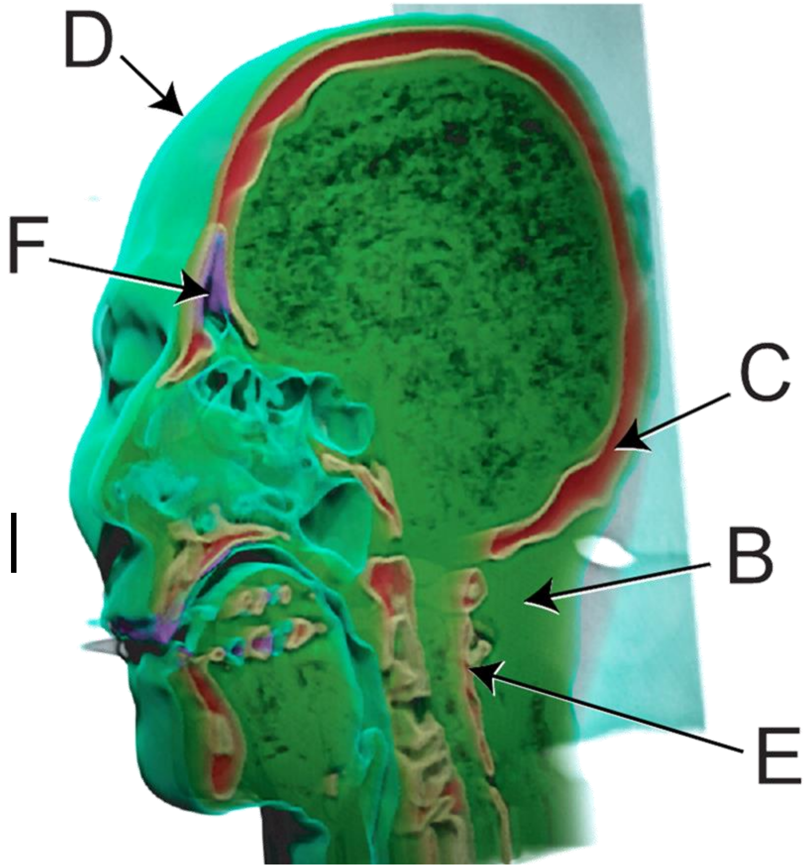


- 2D histogram (x: value, y:gradient magnitude)
  - Material boundaries correspond to arcs



Materials:
A: air
B: soft tissue
C: bone

Material boundaries:
D: air – soft tissue
E: soft tissue – bone
F: air – bone



Visualization using 2D transfer function

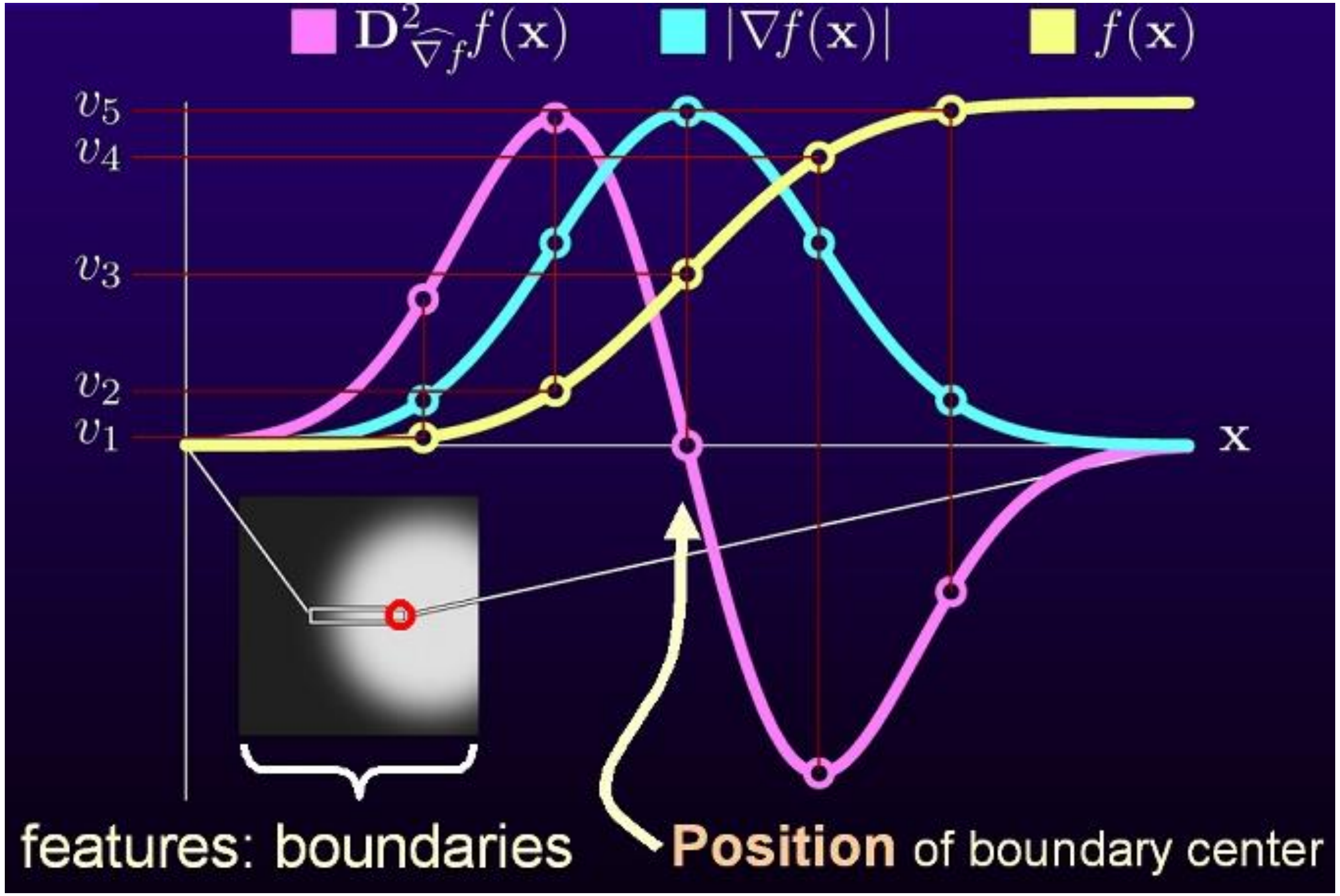
From Hansen, Johnson: The Visualization Handbook

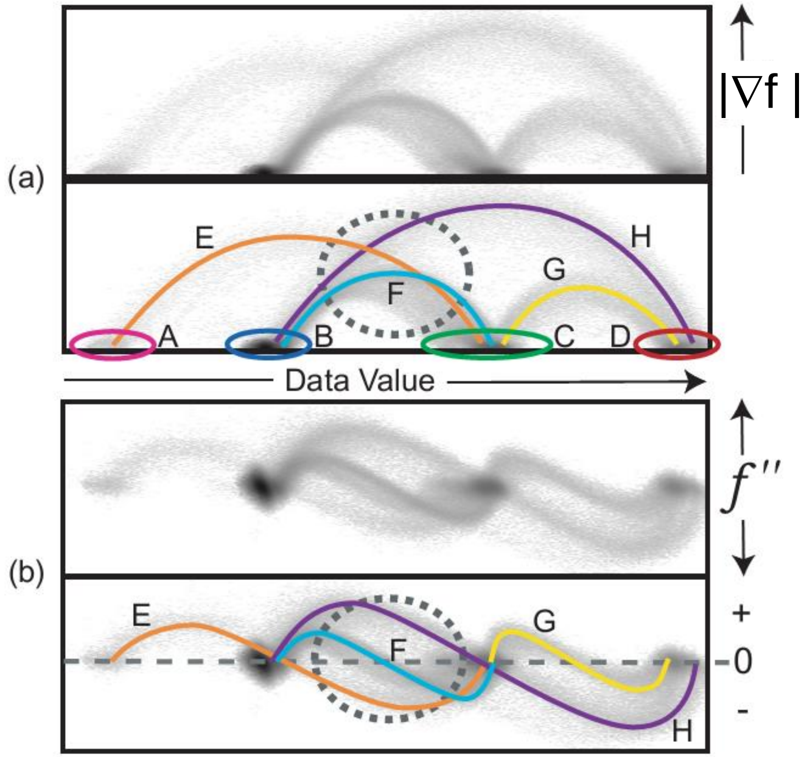
In many cases, scalar value alone is not enough to reliably identify boundary regions/surfaces

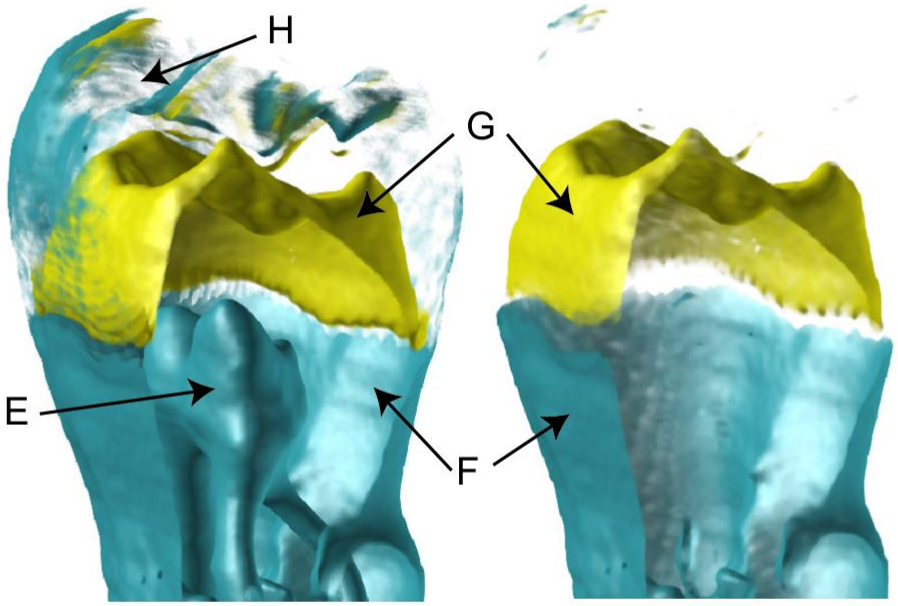
Approach by [Kindlmann & Durkin 98; Kniss, Kindlmann, Hansen 01]:

3D transfer functions, depending on

- Scalar value
- Magnitude of the gradient
- Second derivative along the gradient direction


$D^2_{\widehat{\nabla f}} f(\mathbf{x})$
$|\nabla f(\mathbf{x})|$
$f(\mathbf{x})$
v_5
v_4
v_3
v_2
v_1
x
features: boundaries
Position of boundary center





(c) 2D transfer function

(d) 3D transfer function

Material boundaries:

E: dentin – pulp

F: background – dentin

G: dentin – enamel

H: background – enamel

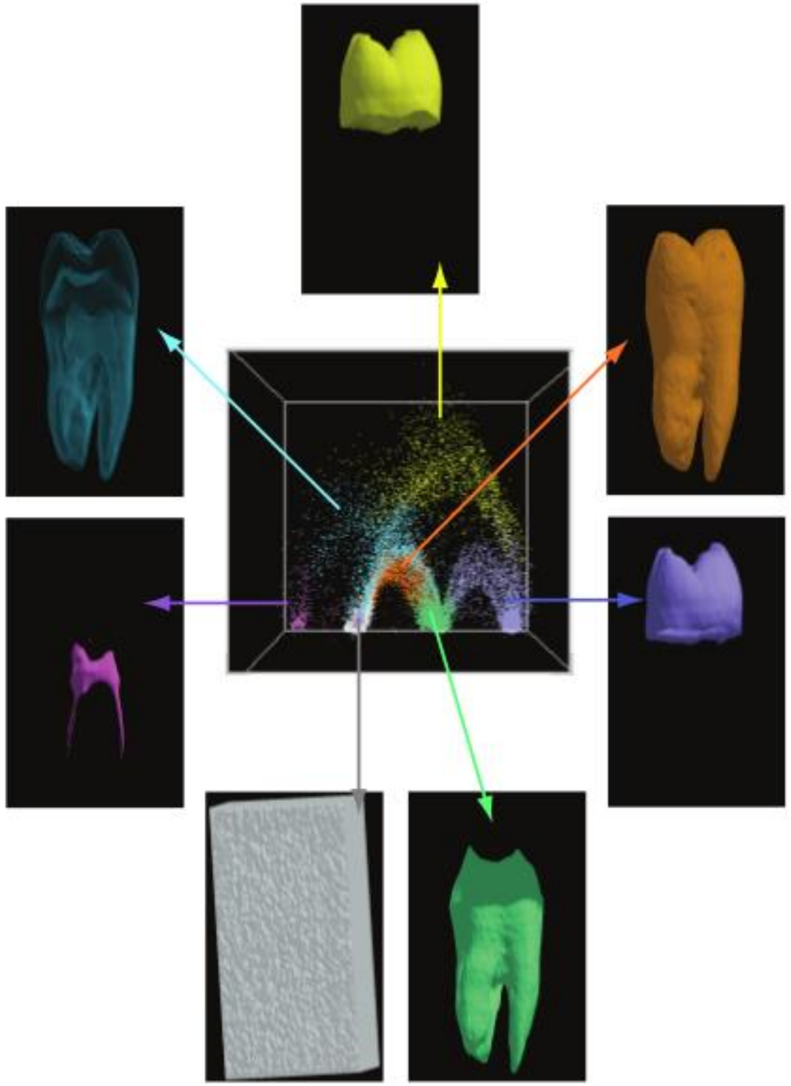
Only using a 3D transfer function, F and G can be separated clearly from other material boundaries like E and H.

From Hansen, Johnson: The Visualization Handbook

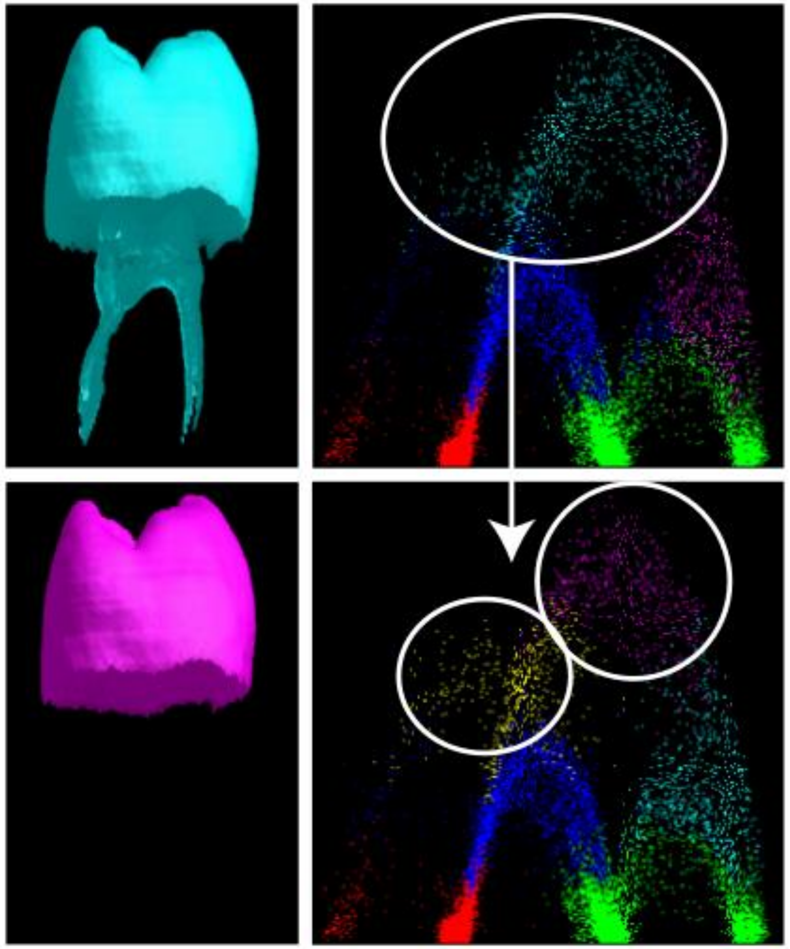
Problem: In high dimension, interactive specification of transfer function becomes more challenging

Approach by [Tzeng & Ma 04]:

- Cluster Voxels in Feature Space
- Assign Material Properties per Cluster
- Interactive Splitting and Merging of Clusters



Correspondence Cluster - Material

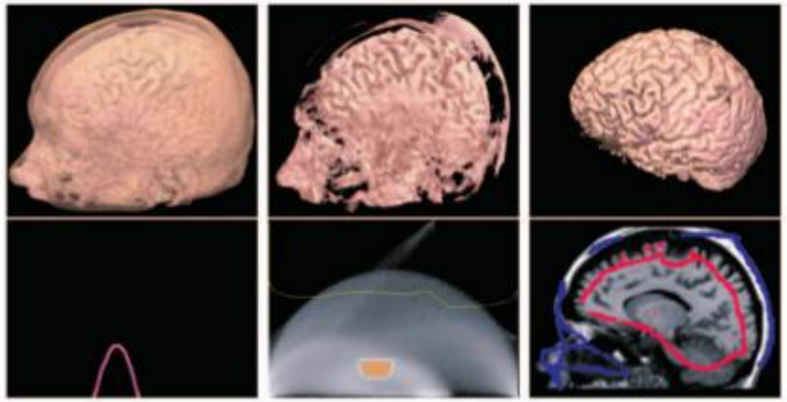


Interactive Cluster Splitting

Problem: In high dimension, interactive specification of transfer function becomes more challenging

Approach by [Tzeng et al. 05]:

- User scribbles on a Slice Visualization
  - Examples of foreground/background voxels
- Train a Classifier / Transfer Function
  - Neural Network
  - Support Vector Machine
- Generalizes to high dimension (e.g., 10D)

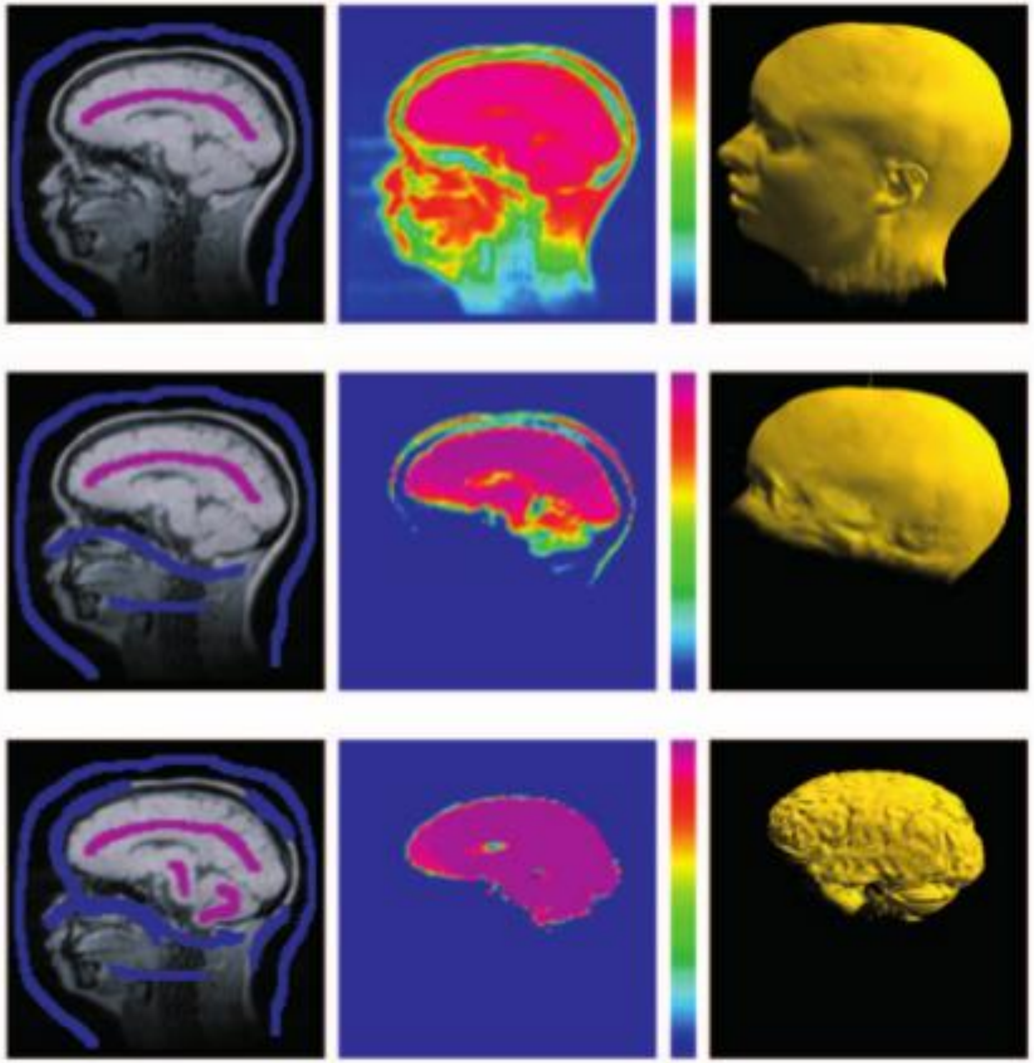


1D TF 2D TF 10D TF

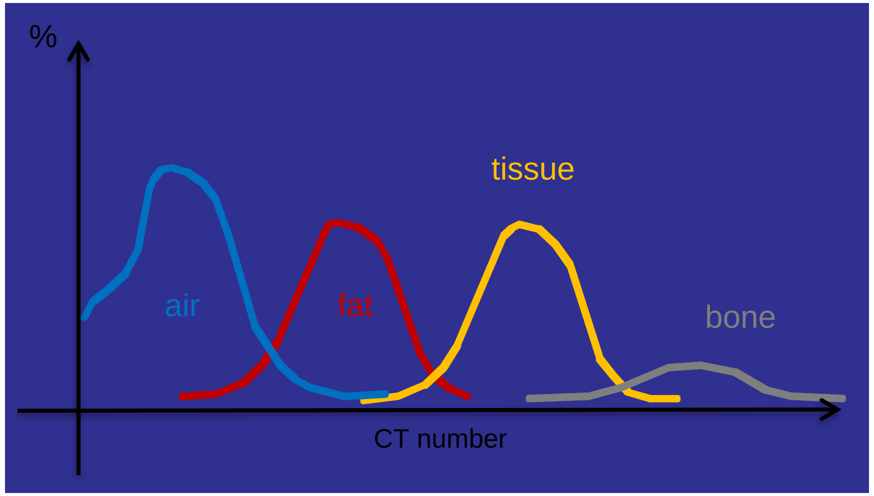
1st Sketch

2nd Sketch

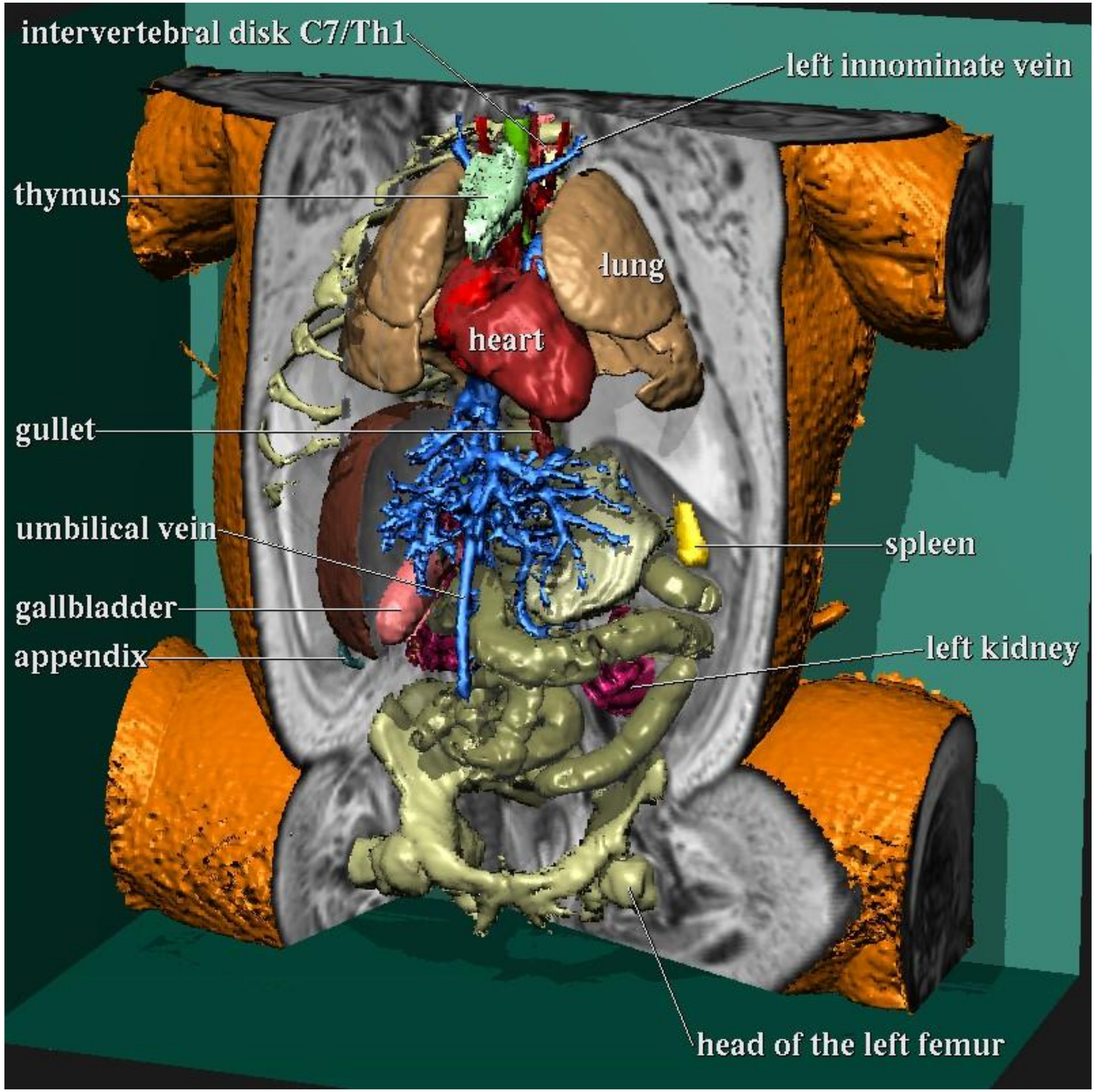
Final Sketch



- Problem: Some structures are impossible to distinguish even with high-dimensional transfer functions
  - Example CT: different organs have similar X-ray absorption



- Approach: In pre-segmentation, assignment of voxels to classes (i.e., the first step of the transfer function) is done by a specialized, (semi-) automated segmentation algorithm before visualizing the data
- Can better account for spatial continuity
- Can account for prior knowledge



Anatomic atlas

Summary

- **Optical Volume Rendering Model**
  - Volume Rendering Integral
  - Illumination
- **Direct Volume Rendering**
  - Ray Casting
  - Composition Schemes
  - Transfer Functions