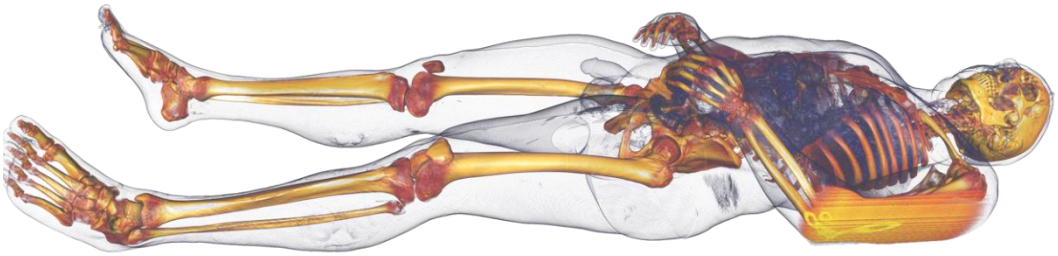
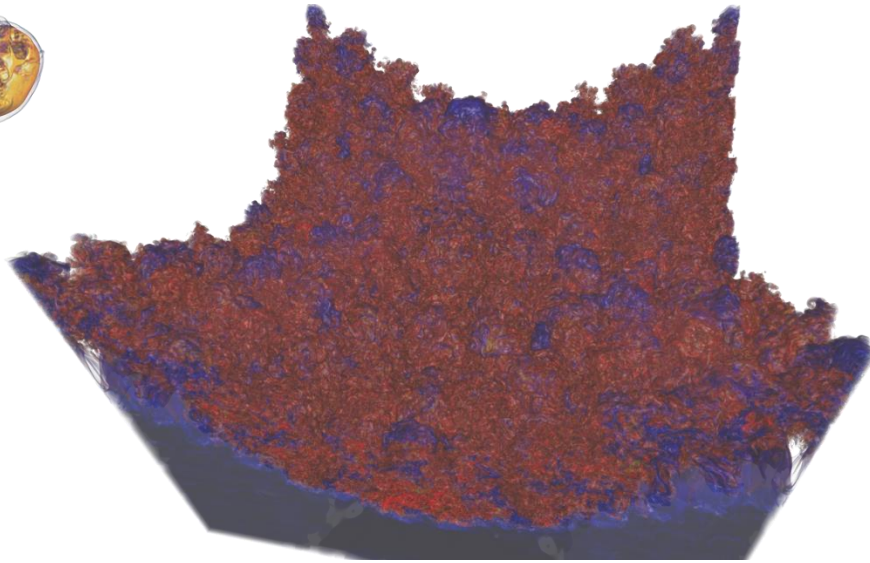
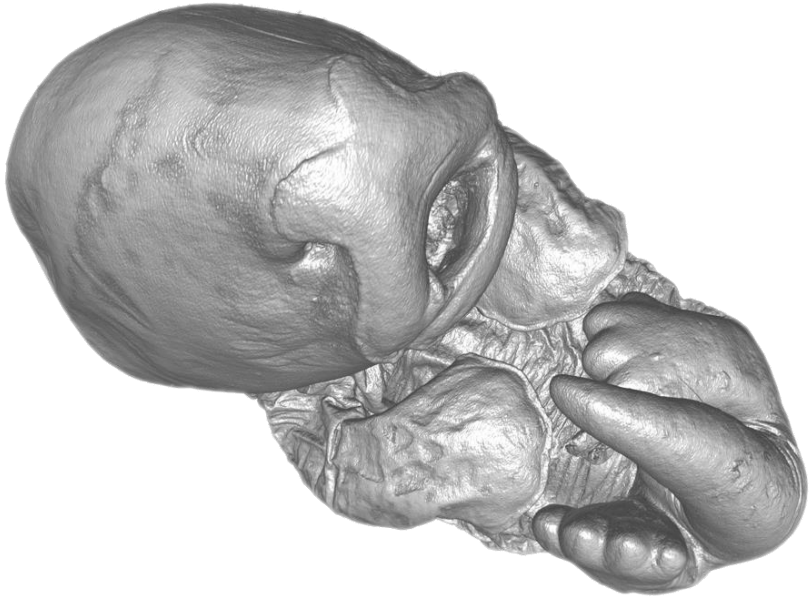
Introduction to Visualization and Computer Graphics
DH2320, Fall 2015

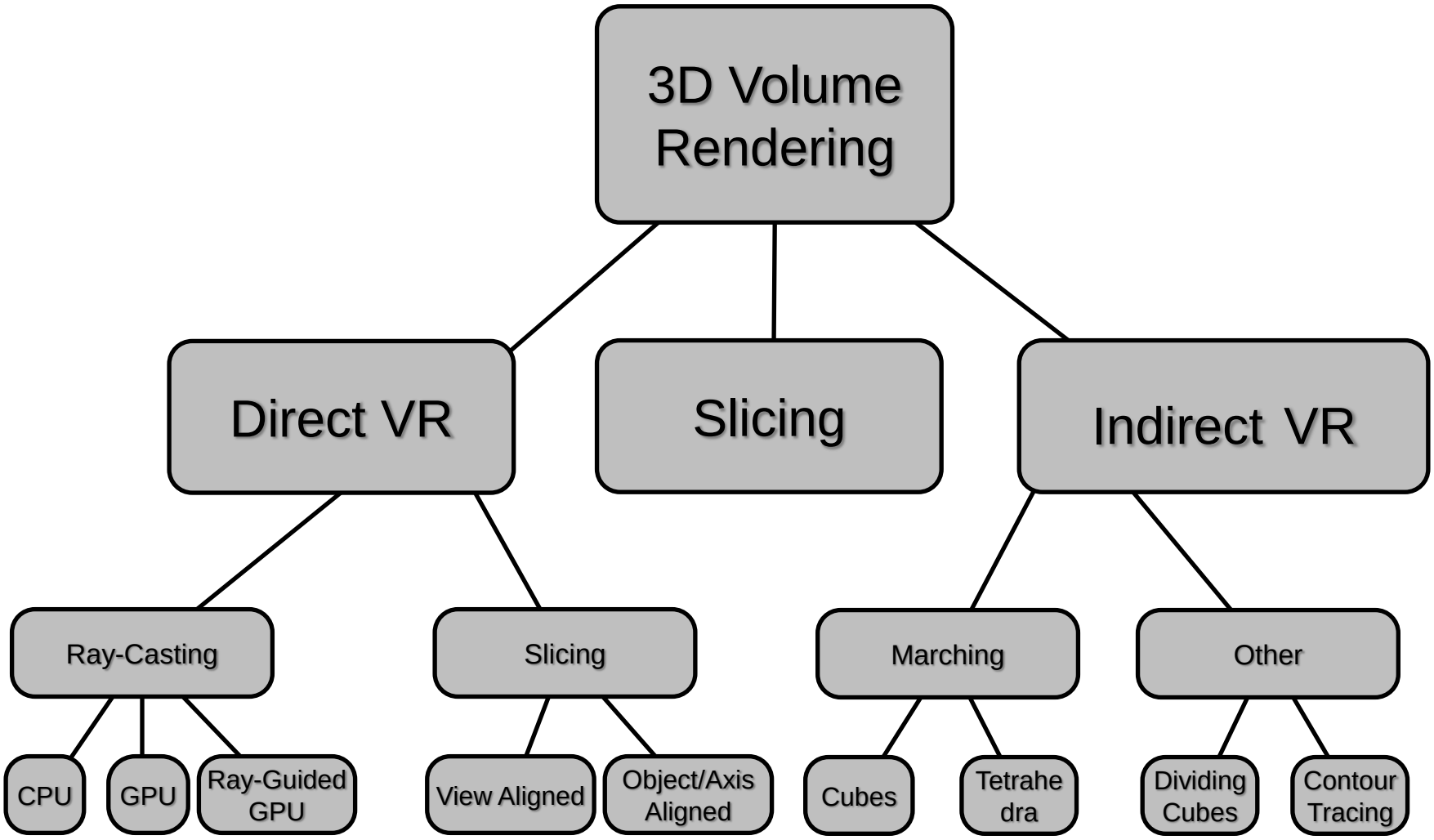
Prof. Dr. Tino Weinkauf

Direct Volume Rendering





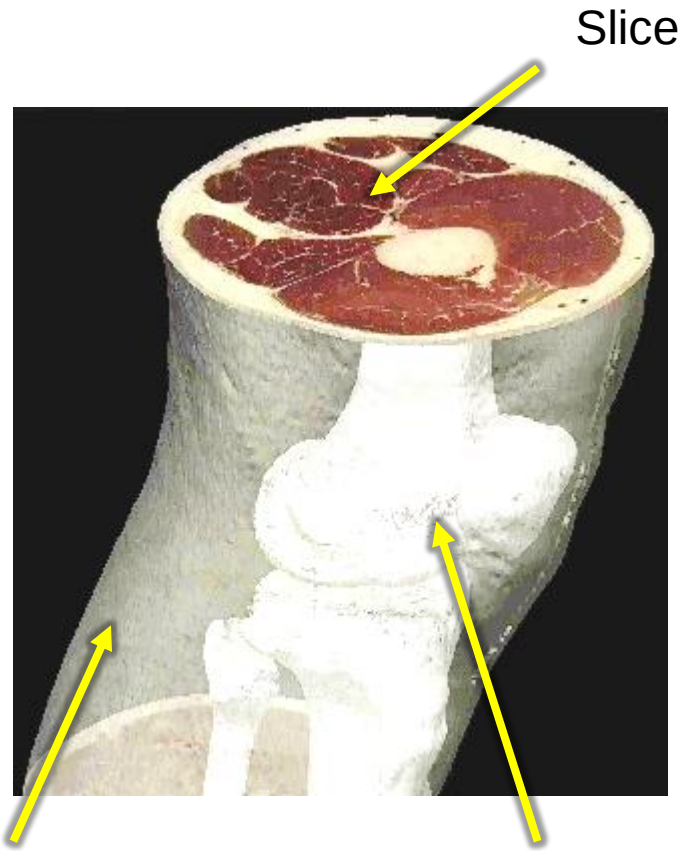



3D Volume Rendering
Direct VR
Slicing
Indirect VR
Ray-Casting
Slicing
Marching
Other
CPU
GPU
Ray-Guided GPU
View Aligned
Object/Axis Aligned
Cubes
Tetrahedra
Dividing Cubes
Contour Tracing

- **Slicing**
  - Visualize 2D cuts through the volume
- **Direct volume rendering**
  - Consider the data as a light-emitting medium with specific emission and absorption properties. The visual impression when looking at it is simulated according to the laws of physics
- **Indirect volume rendering techniques**
  - Convert/reduce volume data to an intermediate representation (e.g., surface representation), which can be rendered with traditional techniques

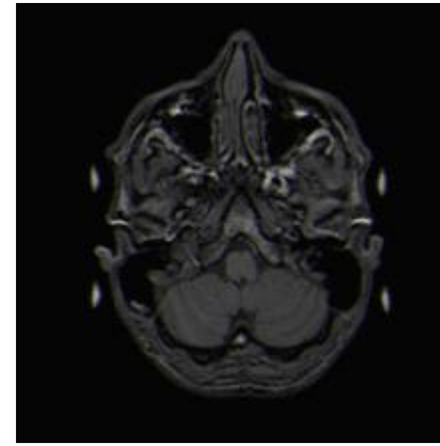
All three strategies can be combined:

- **Slicing:**
  Map data to colors, display on a slice plane
- **Isosurfacing:**
  (Semi-)opaque surfaces, material boundaries
- **Transparency effects:**
  Volume material attenuates reflected or emitted light



Map data on a series of 2D planes through the volume

- In medicine, this is still the most frequently used visualization
- If data was acquired as a series of 2D slices, natural to show those
- Other axis-aligned views
- Rotate ("re-angulate") to an object/patient-aligned view
  - Involves interpolation



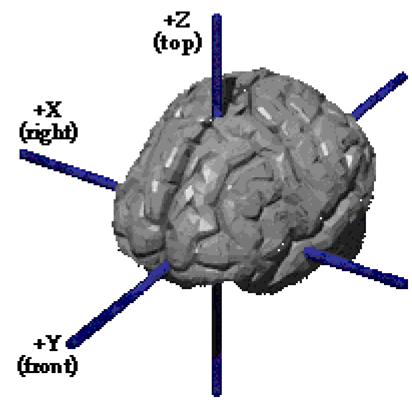


Image Sources: Wikipedia, Graham Wideman

A **transfer function** maps data values to colors that can be used to visually display the data:

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

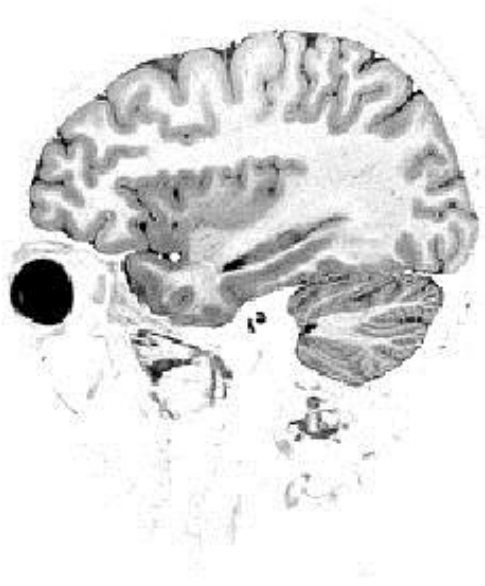
In most implementations, the color type C is a (red, green, blue) triple of either floating-point [0,1] or unsigned char [0,255] values.

When to apply?

- **Pre-classification** applies T to the sampled data values and interpolates colors
- **Post-classification** interpolates the data and applies T to the resulting value

One of the simplest transfer functions maps a range of data values **(“window”)** to a linear ramp of grayscales. Values outside that range are mapped to black or white, respectively.

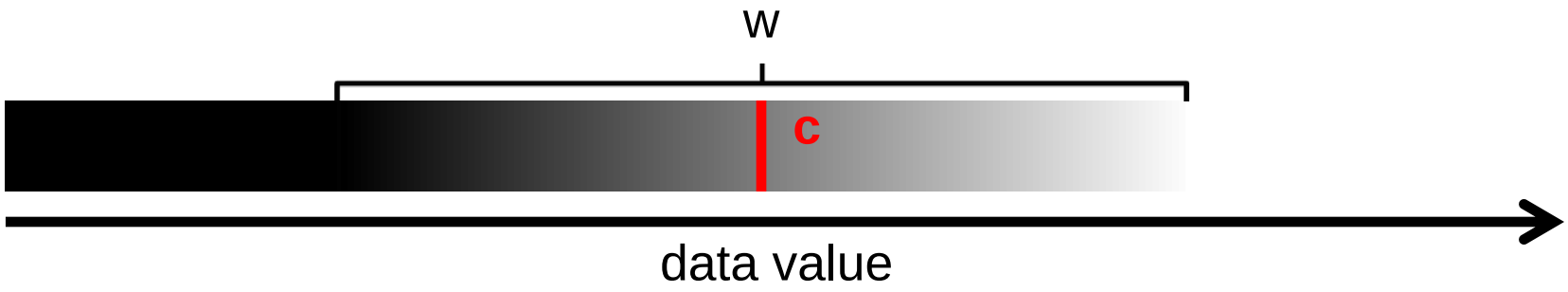
Full Data Range



Brain Window

Tissue Window

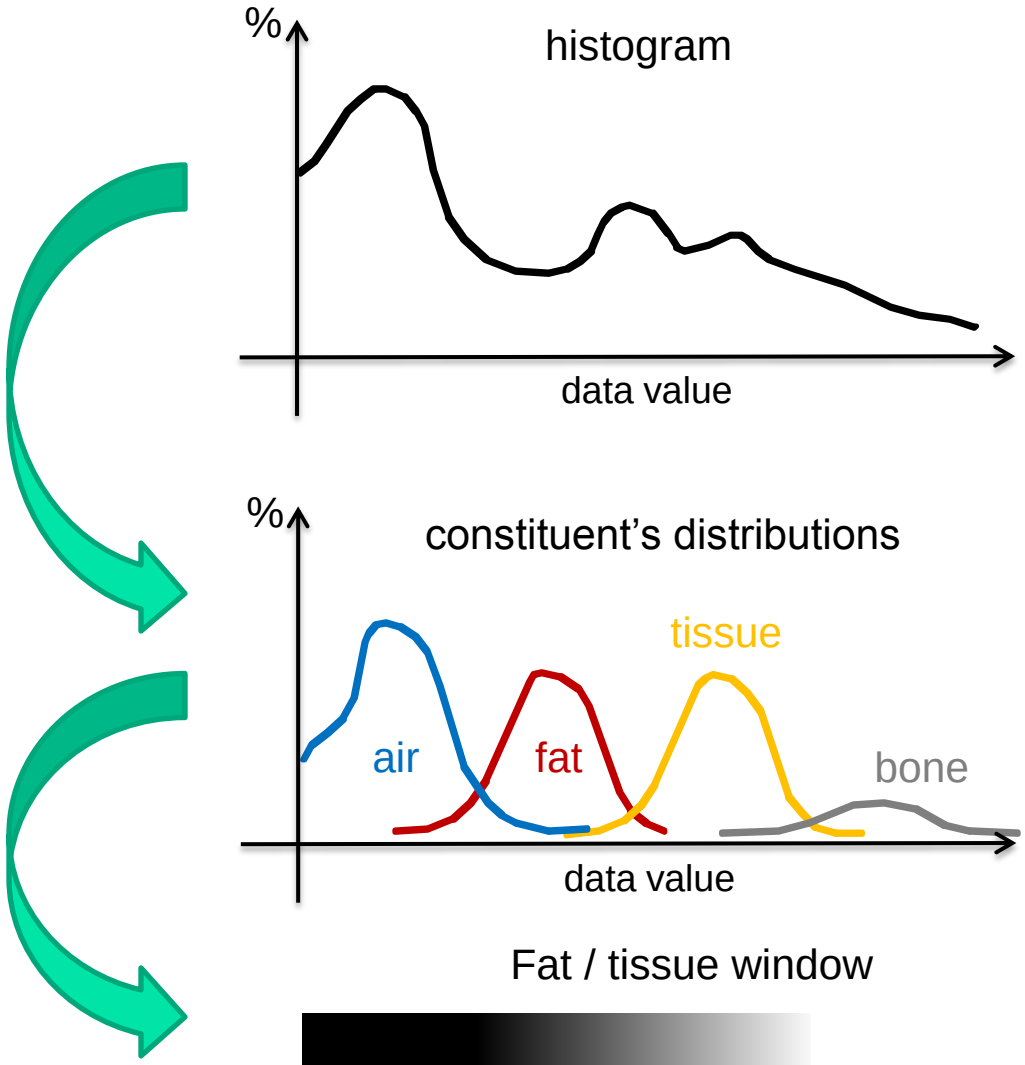
Windows are commonly defined by their center c and width w≥1.



For data value x and color output range C∈[0,1], the following pseudo-code applies:

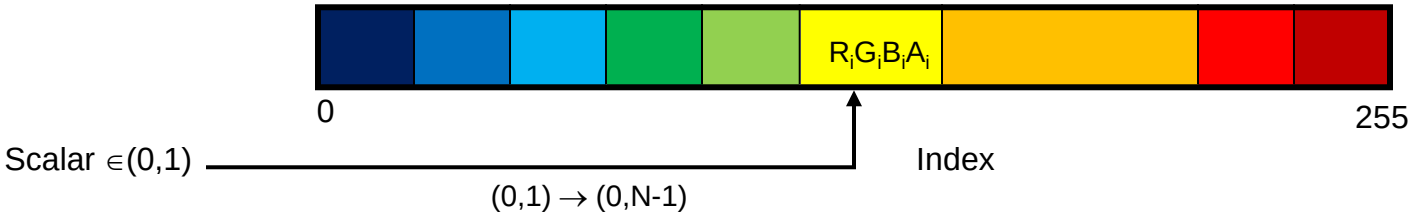
```
if (x <= c - 0.5 - (w-1)/2) then C := 0
else if (x > c - 0.5 + (w-1)/2) then C := 1
else C := (x - (c - 0.5)) / (w-1) + 0.5
```

Histogram of all data values in a volume dataset:

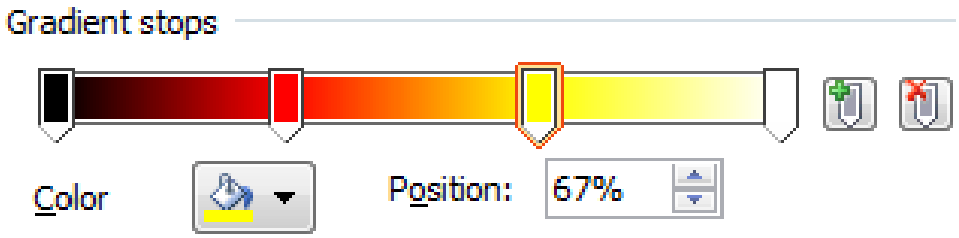


Transfer functions can also be used to map scalar data to color

- Can be implemented as a color lookup table:



- Often specified by defining the color for a discrete set of data values (“gradient stops”) and interpolating in between (e.g., in RGB):



All these examples show exactly the same data:

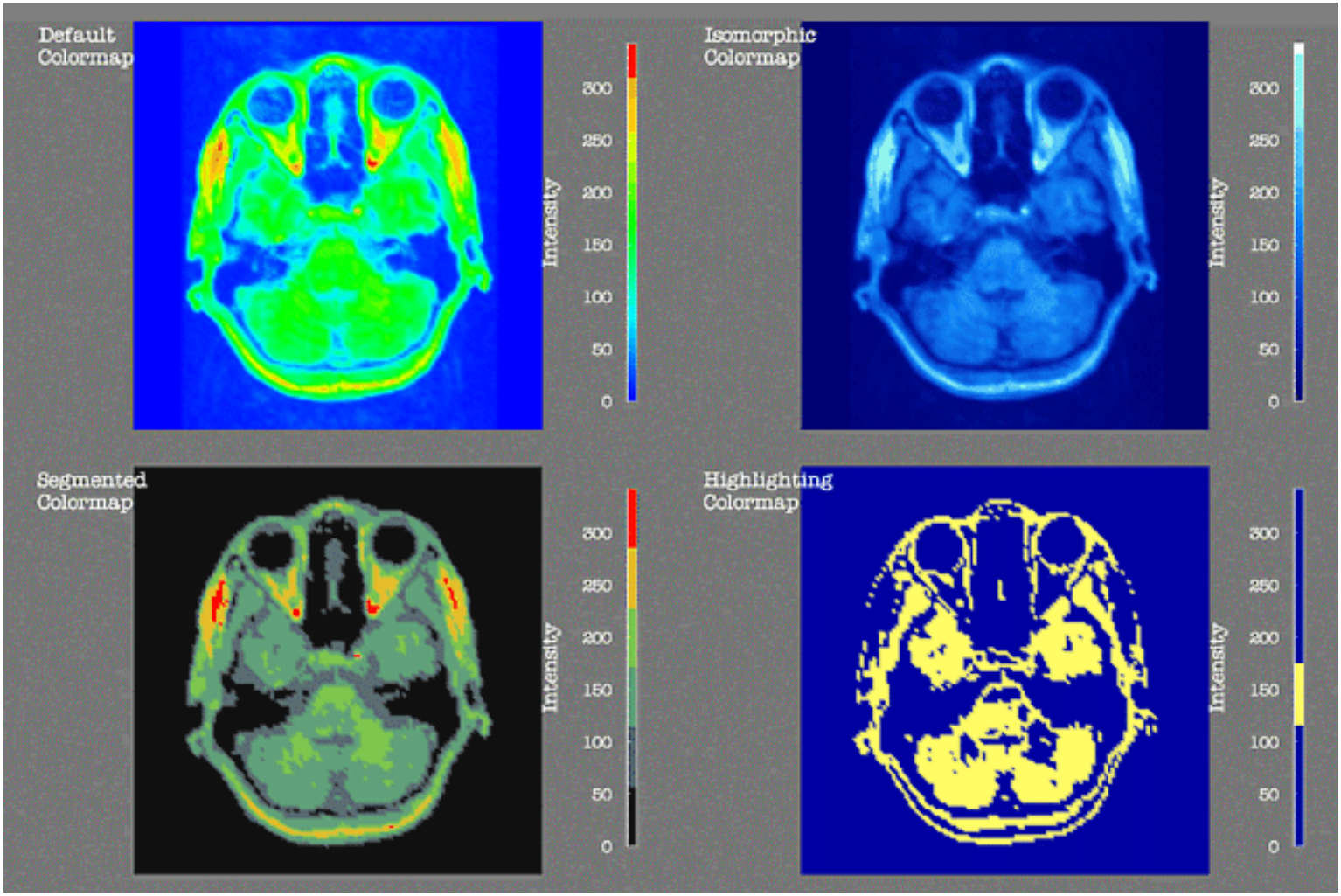


Image Source:
Rogowitz et al.

In this case, unnecessary surgery was performed based on a poorly adjusted color map:

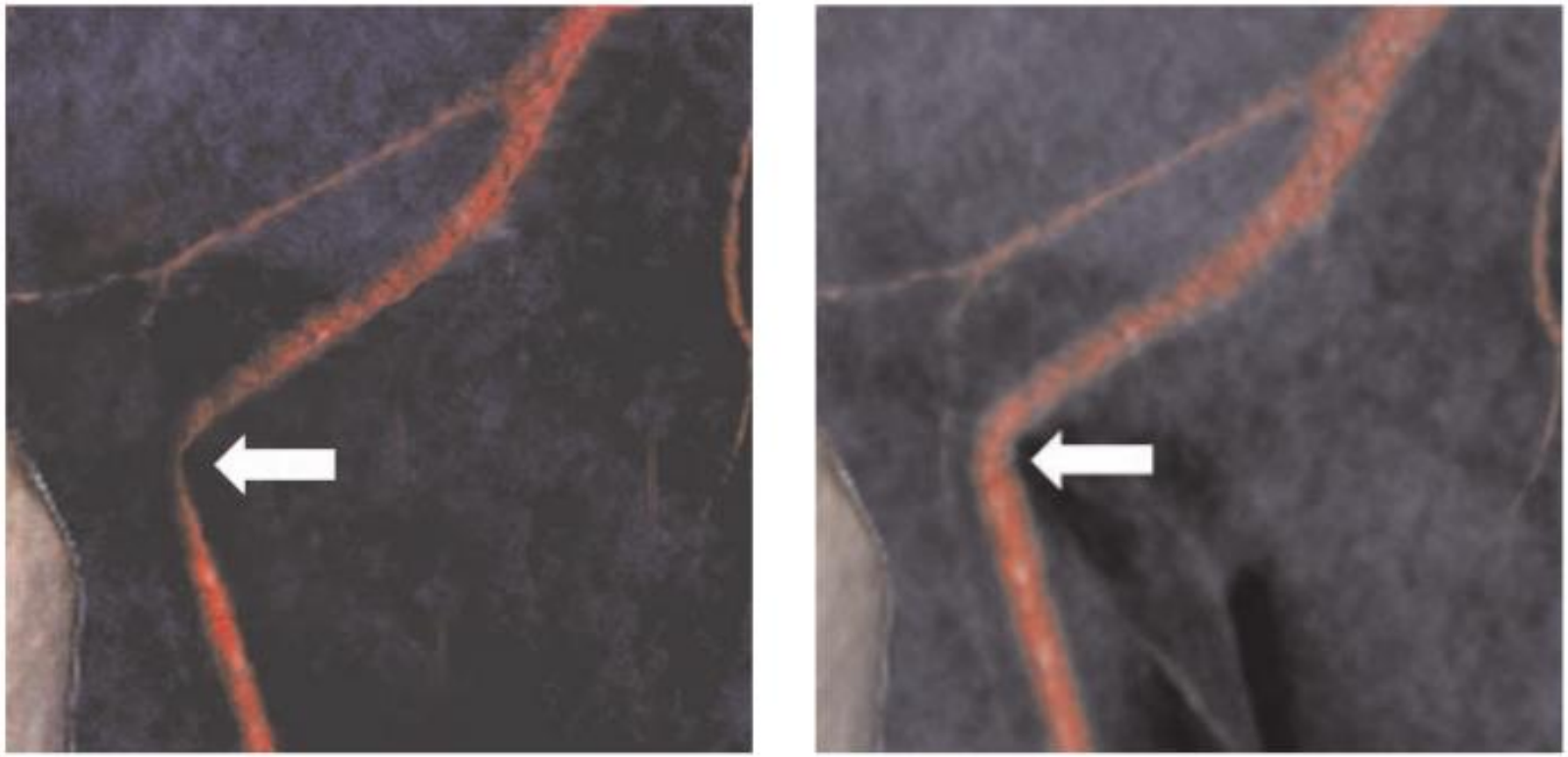
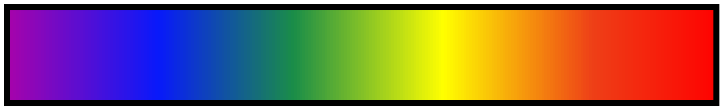


Image Source: Lundström et al.

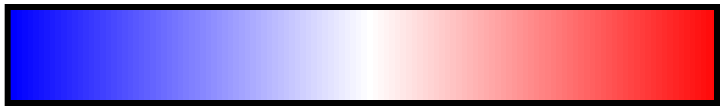
- Gray scale color table
  - Intuitive ordering

- Rainbow color table
  - Based on HSV color space

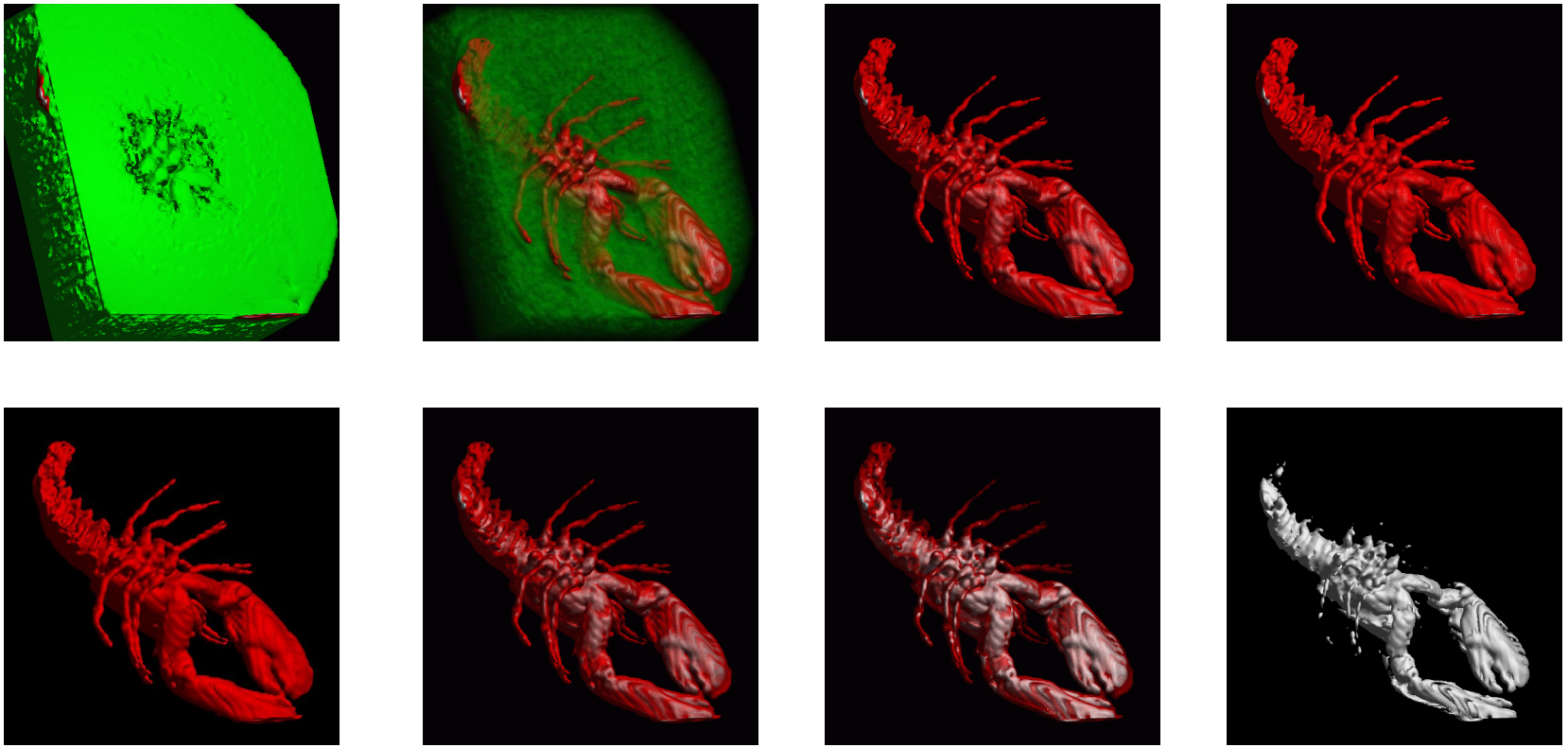


- “Black body radiation”

- Cool-to-warm

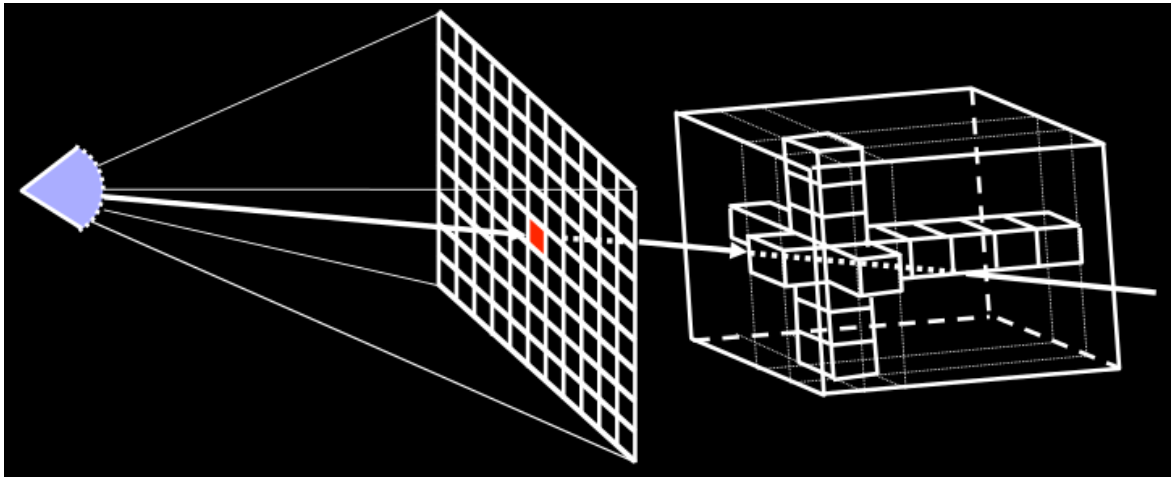


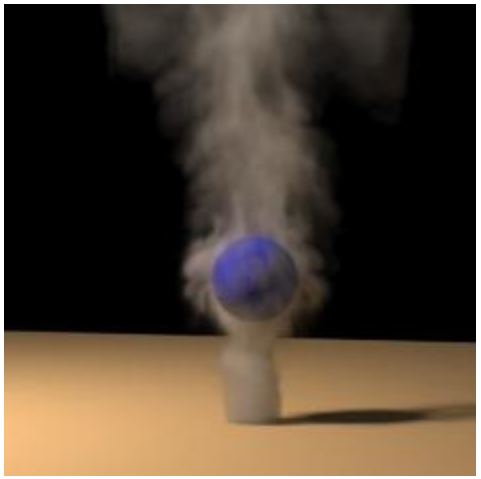
- Blue-to-yellow

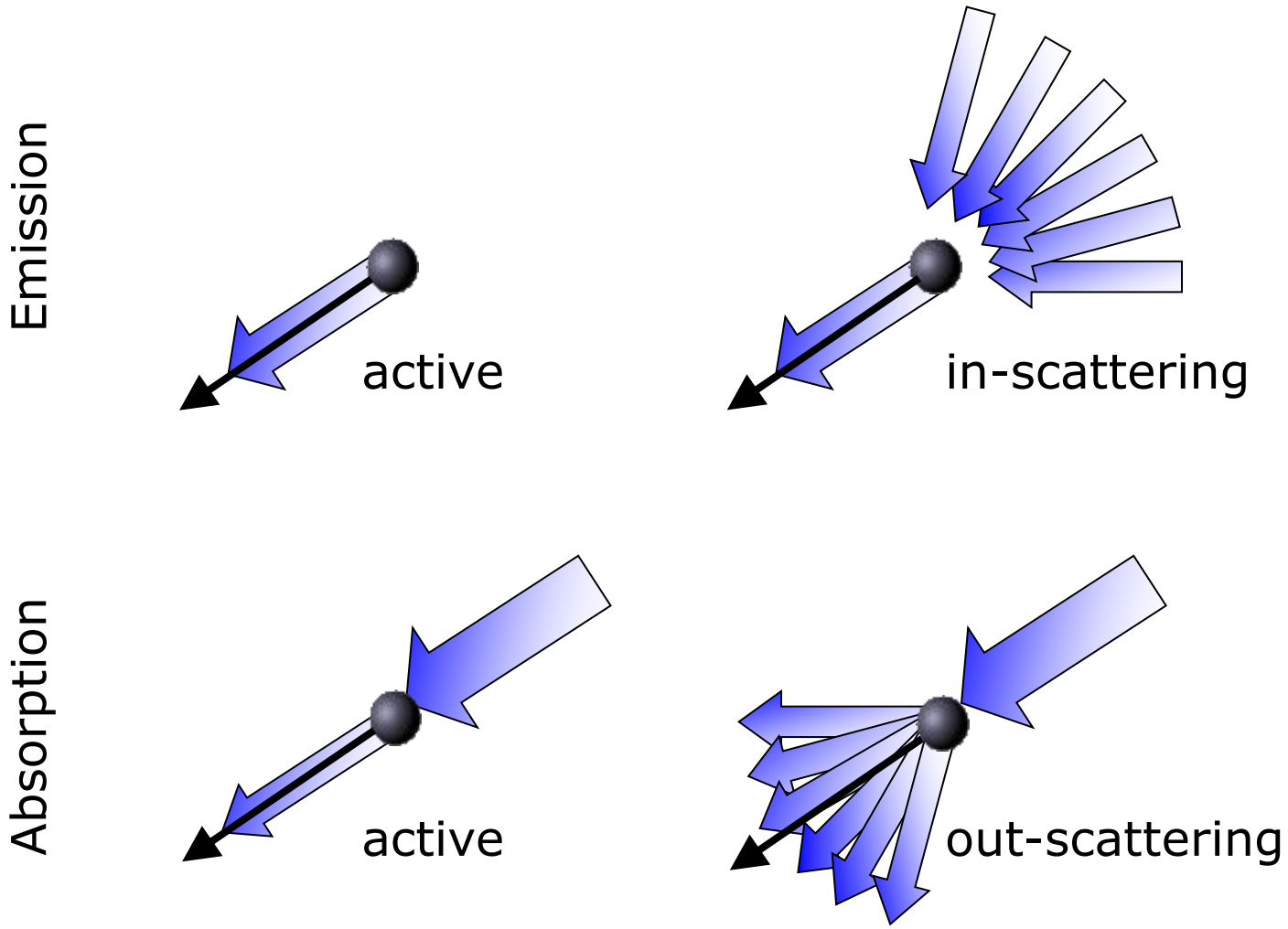


Same volume with different transfer functions that now include an alpha channel (=opacity) and are rendered volumetrically

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.

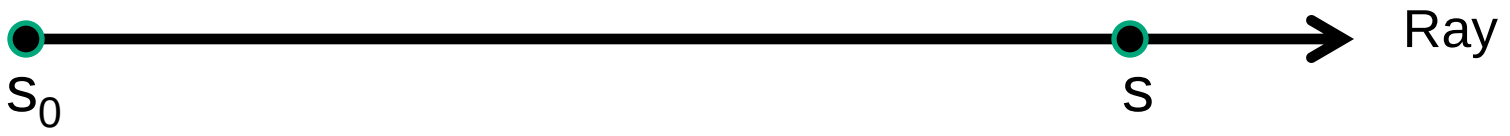




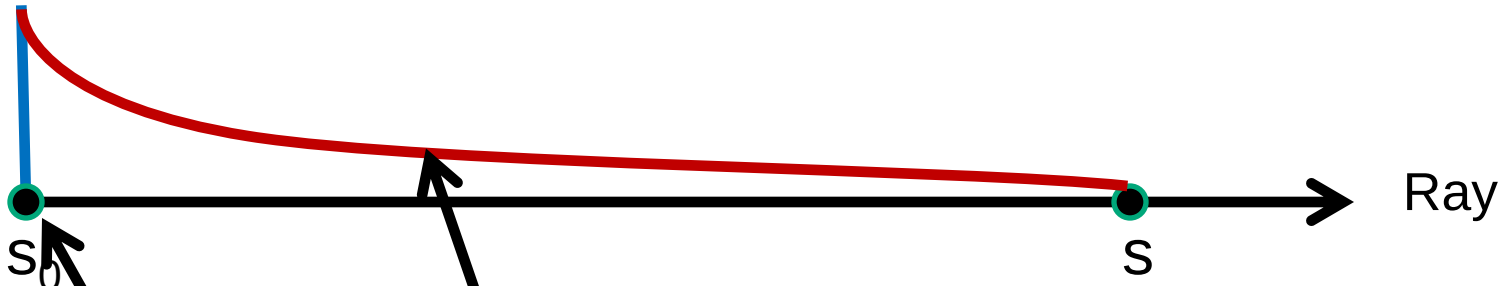

Emission
active
in-scattering
Absorption
active
out-scattering

Light (Particle) interaction with density volume

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



$$I(s) =$$ Light intensity at s

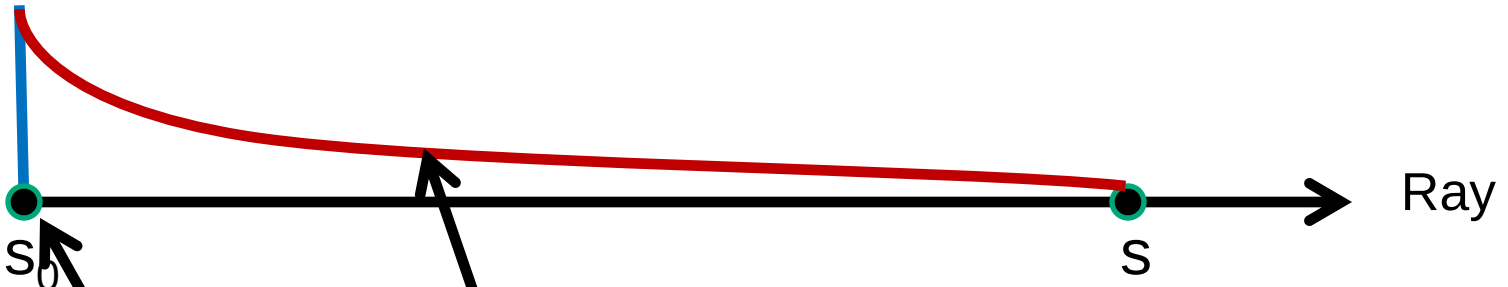


$$I(s) = I(s_0)\,\mathrm{e}^{-\tau(s_0,s)}$$

Emission at s_0 Attenuation along s_0-s

Optical depth τ
Absorption κ

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



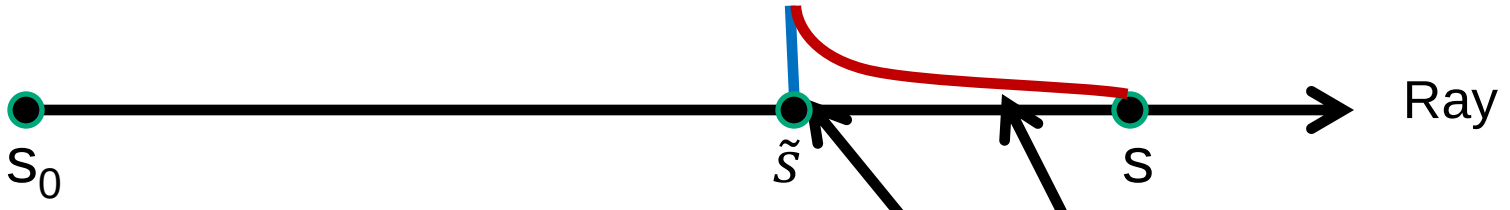
$$I(s) = I(s_0)\, e^{-\tau(s_0, s)}$$

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

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

Integrate contributions of all points along the ray

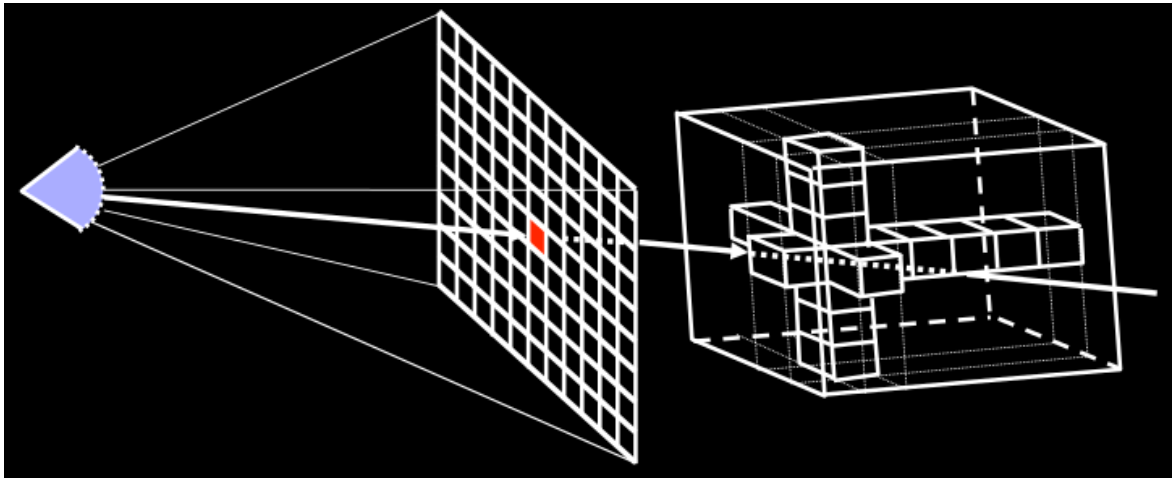
$$I(s) = I(s_0)\mathrm{e}^{-\tau(s_0,s)} + \int_{s_0}^{s} q(\tilde{s})\mathrm{e}^{-\tau(\tilde{s},s)}d\tilde{s}$$

There is no closed form solution of the integral in general

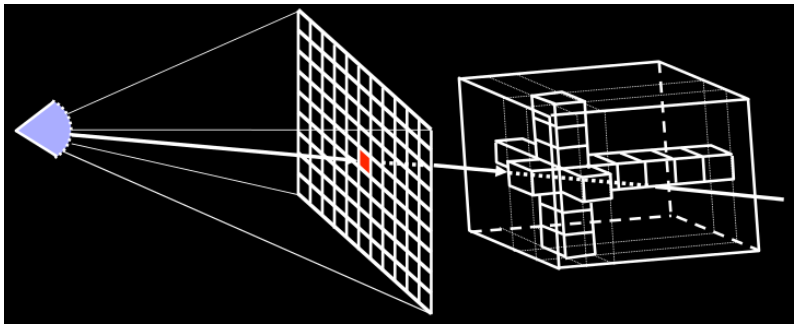
➔ Approximate the integral with a discrete sum:

- 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

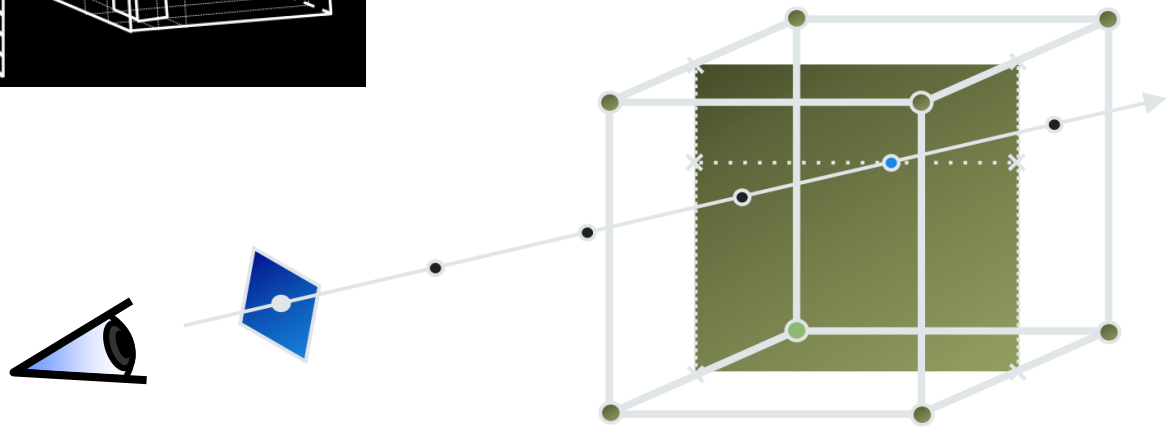
- Similar to ray tracing in surface-based computer graphics
- In volume rendering,
  - we only deal with primary rays; hence: *ray-casting*
  - we composit in each step, rather than searching a ray/surface intersection



- 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

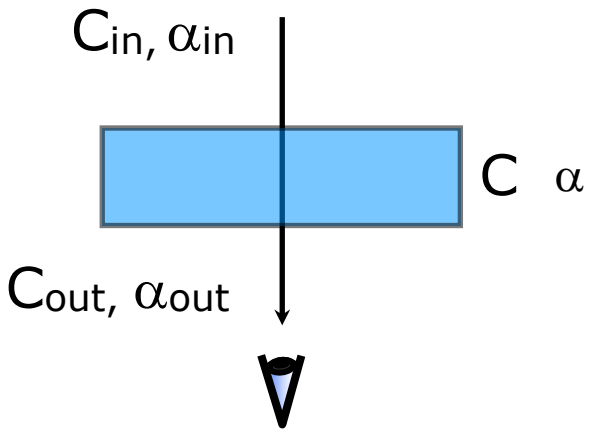


- How is color and opacity determined at each integration step?
- Opacity and (emissive) color in each cell according to classification
- Additional color due to external lighting
- No shadowing, no secondary effects

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

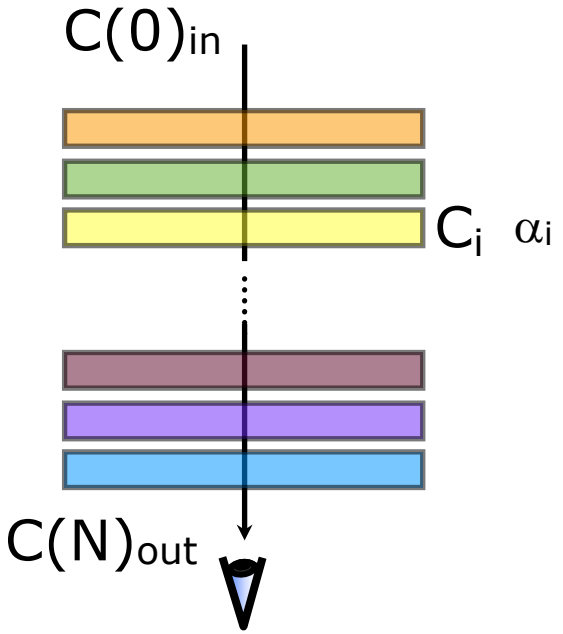
$$\alpha_{out} = (1 - \alpha)\, \alpha_{in} + \alpha$$

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

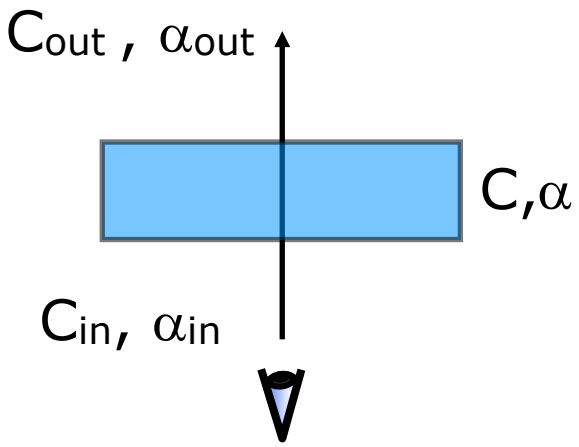


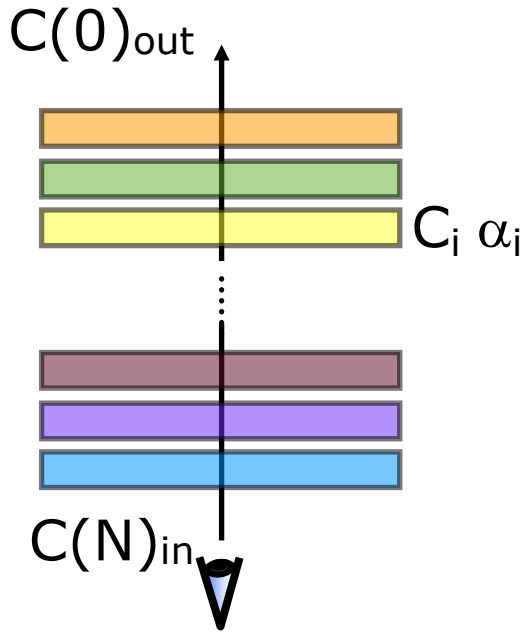
$$C(i)_{in} = C(i\text{-}1)_{out}$$

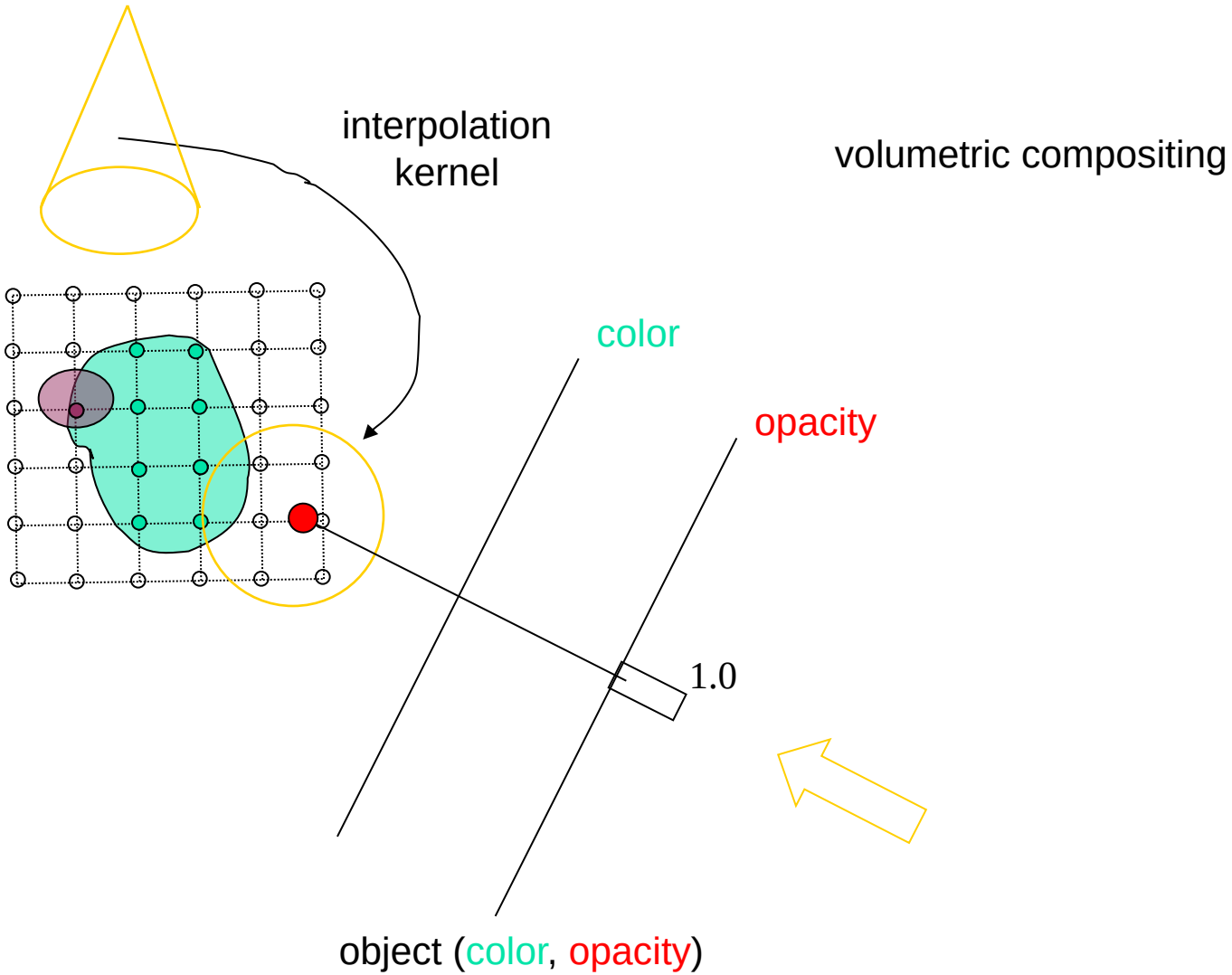
Iterative process. The output color of the previous step becomes the input color of the next step.

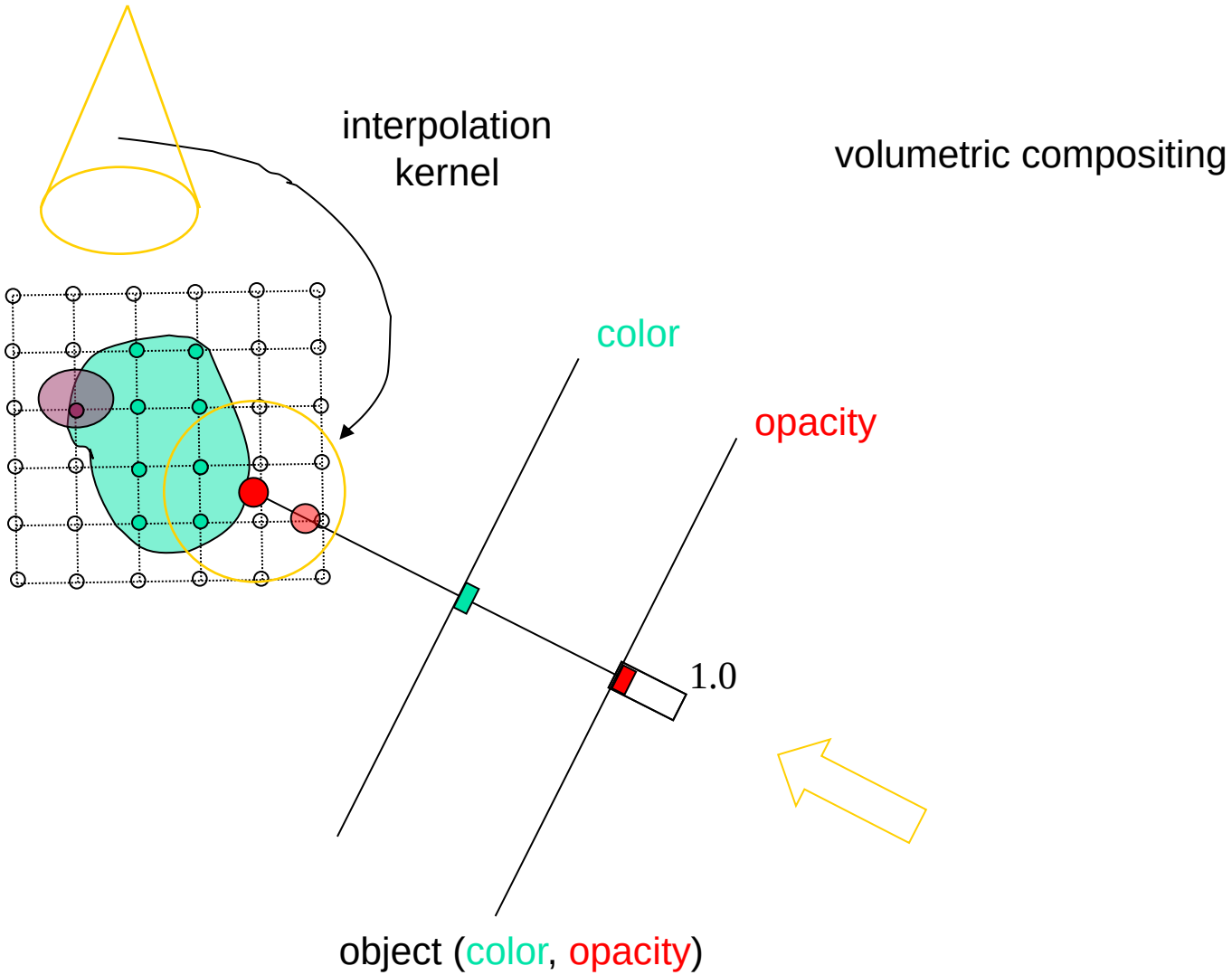


$$C_{out} = C_{in} + (1 - \alpha_{in})\, \alpha C$$
$$\alpha_{out} = \alpha_{in} + (1 - \alpha_{in})\, \alpha$$

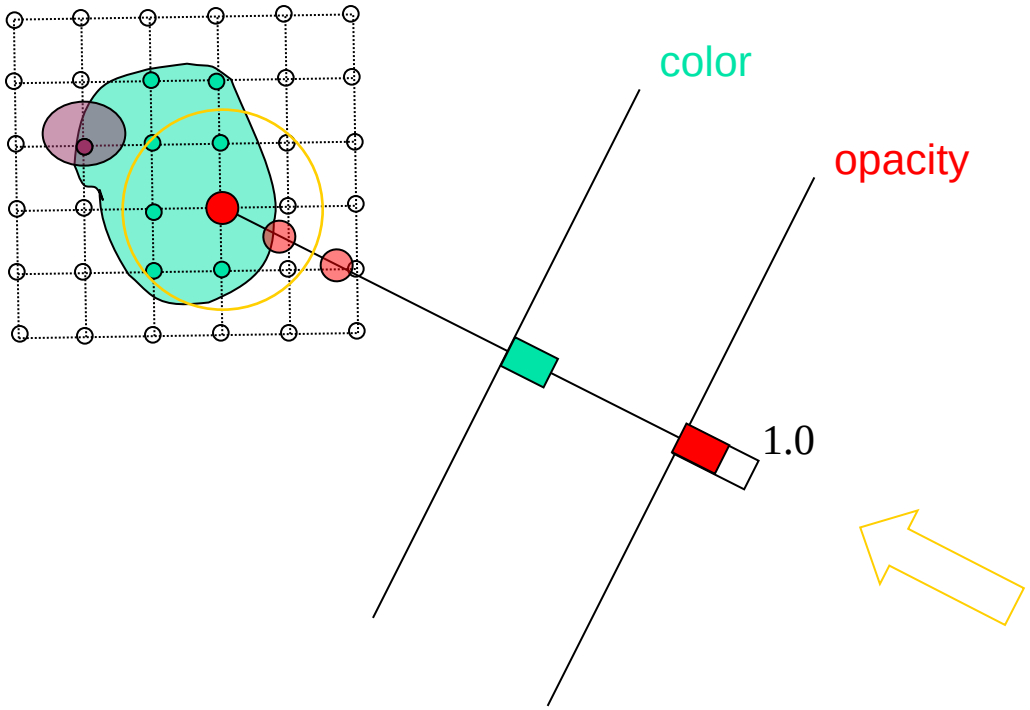




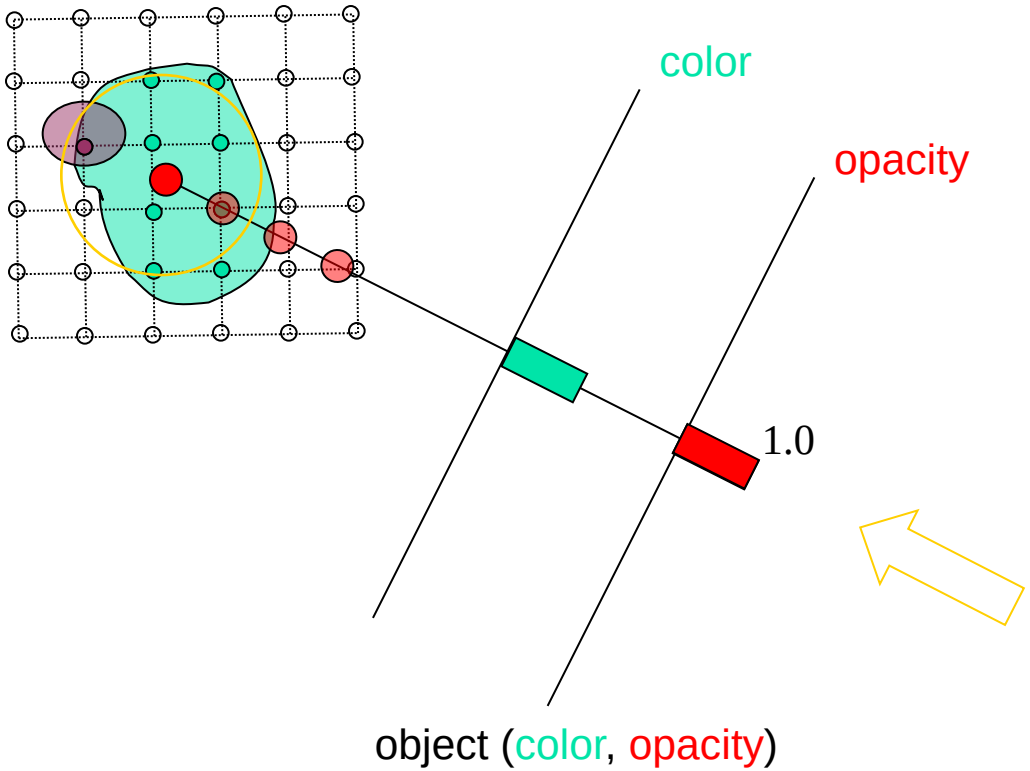

interpolation
kernel
volumetric compositing
color
opacity
1.0
object (color, opacity)


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

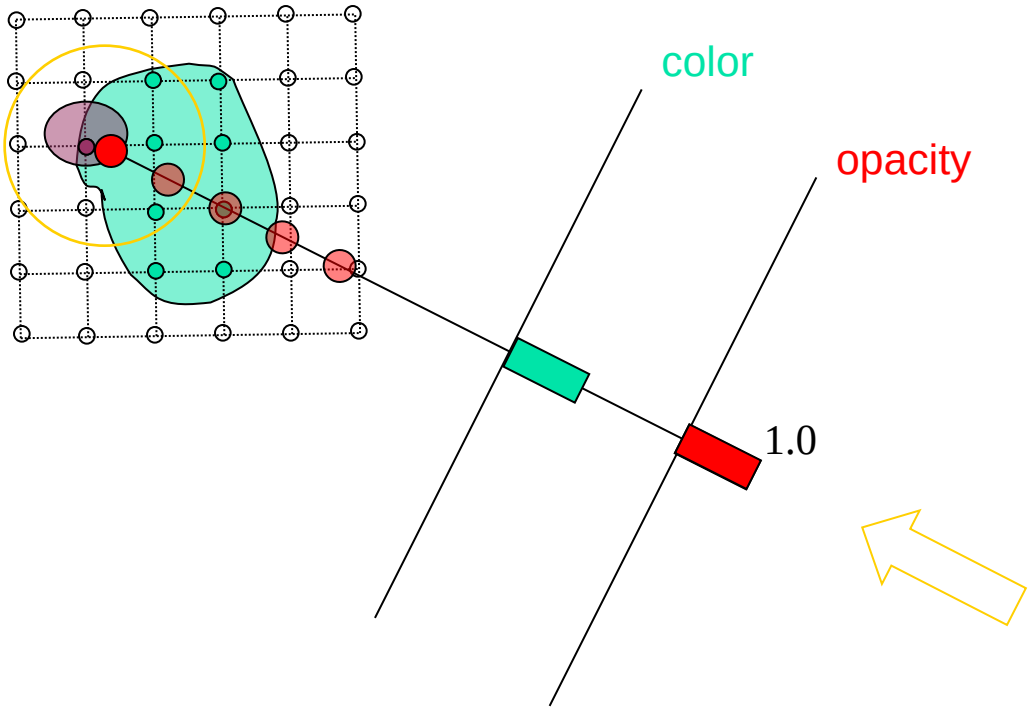
volumetric compositing



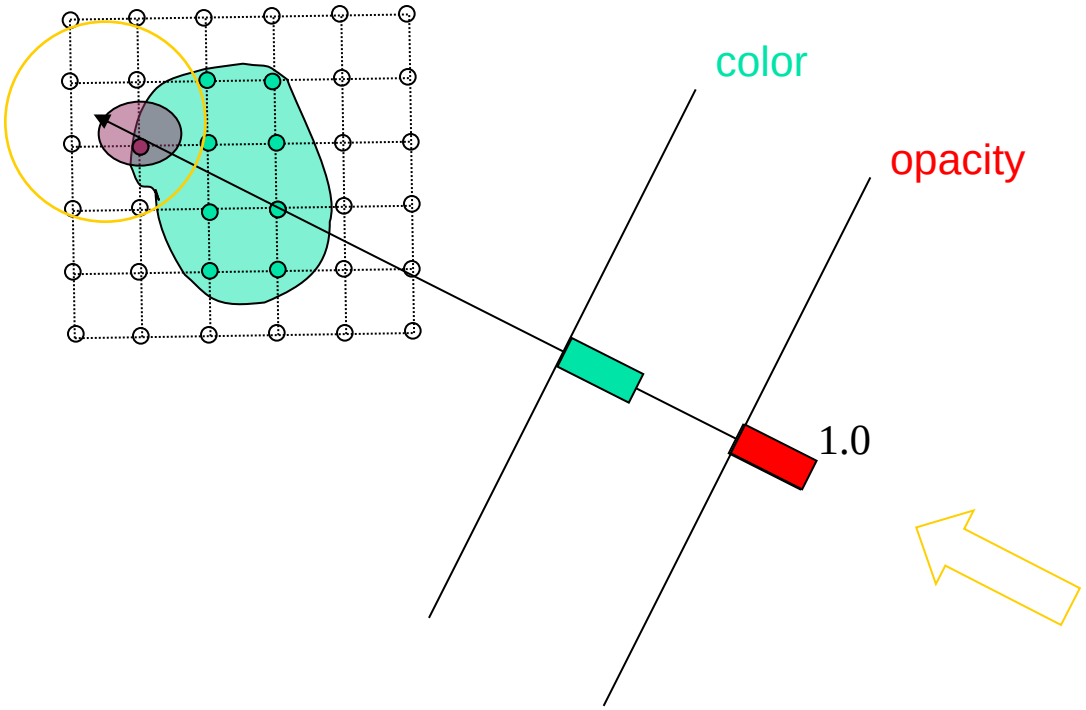
volumetric compositing



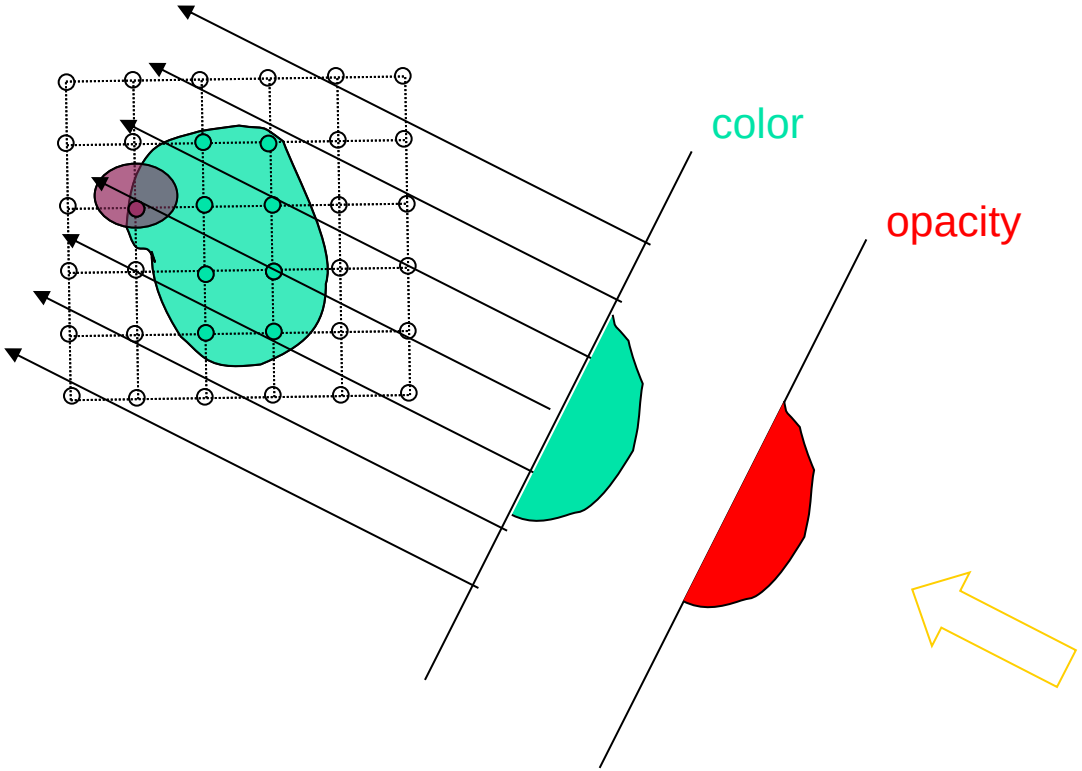
volumetric compositing



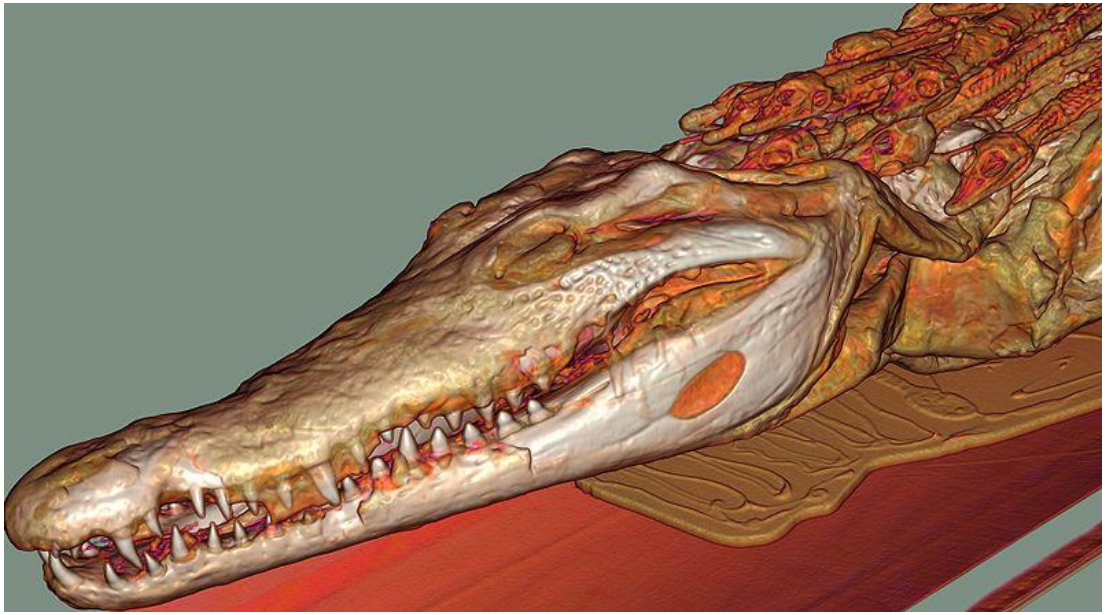
volumetric compositing

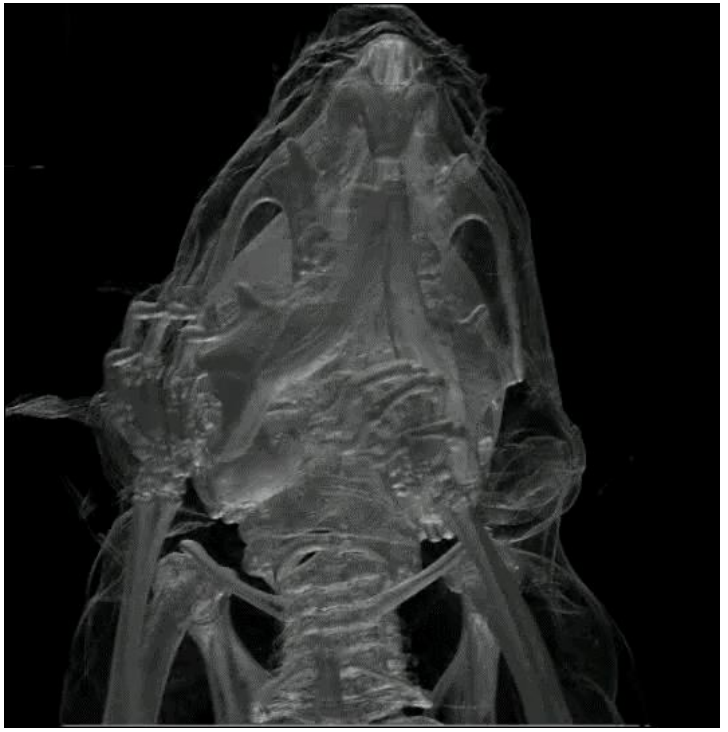


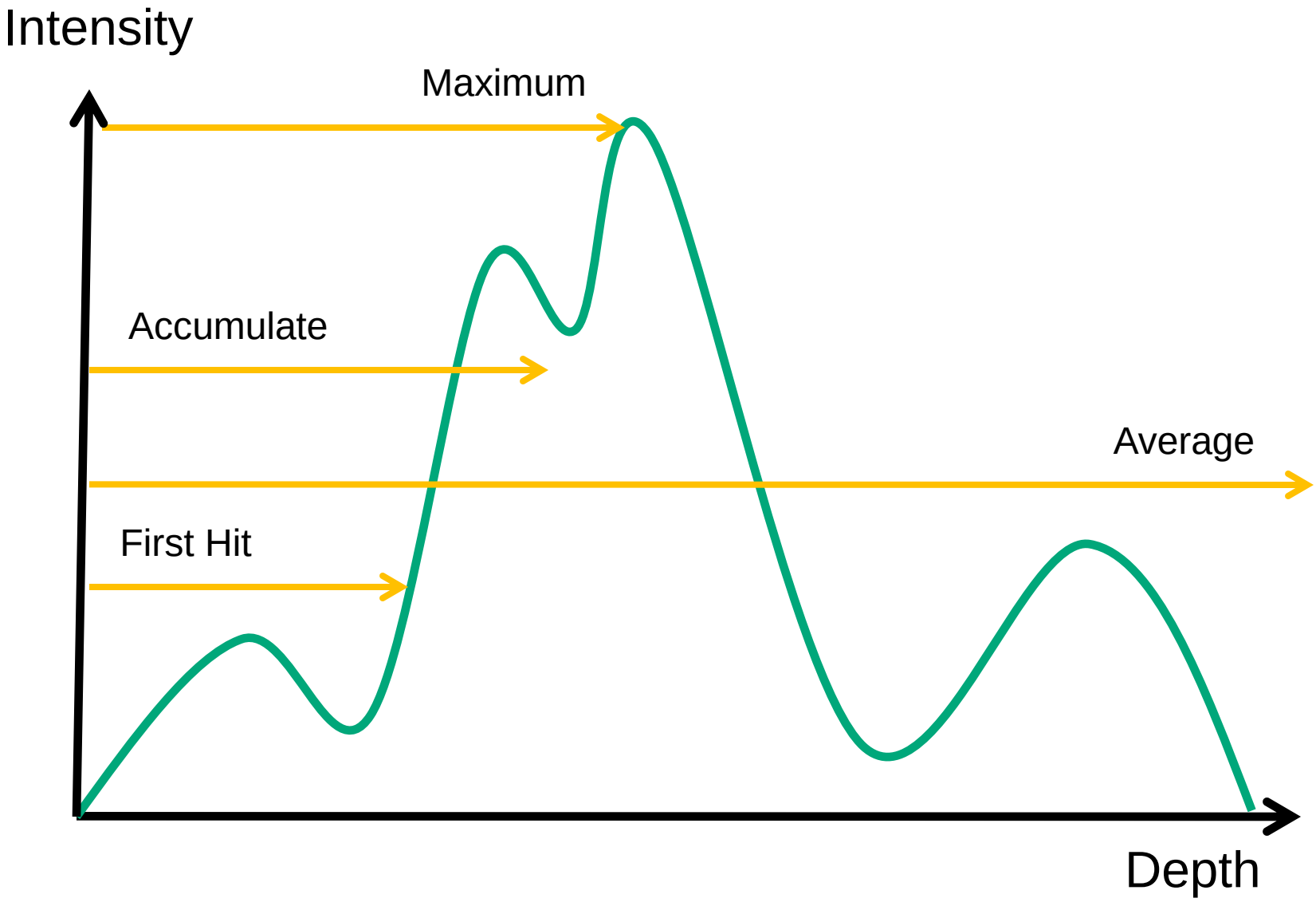
volumetric compositing

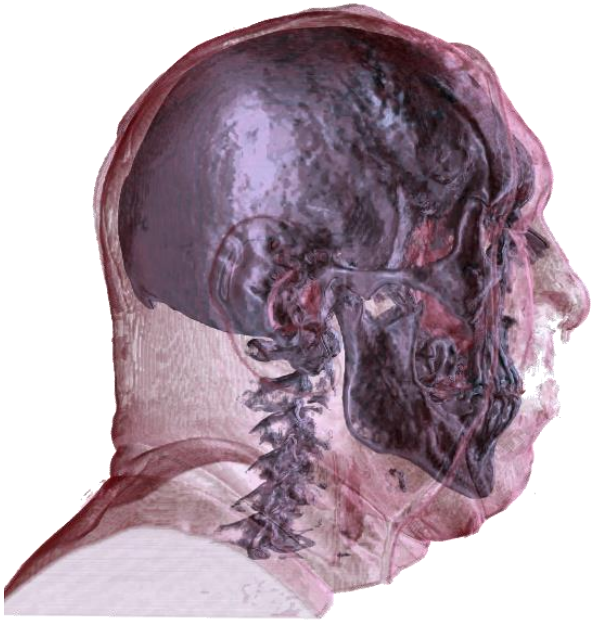


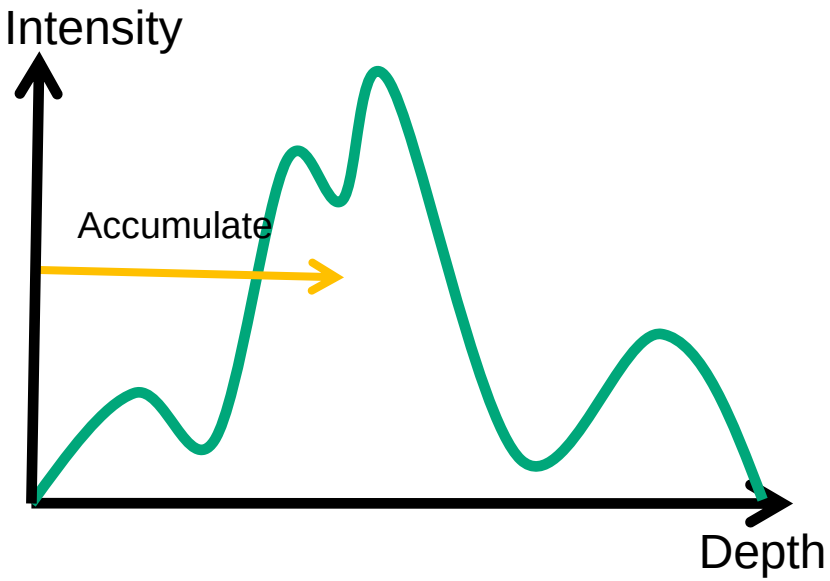
object (color, opacity)

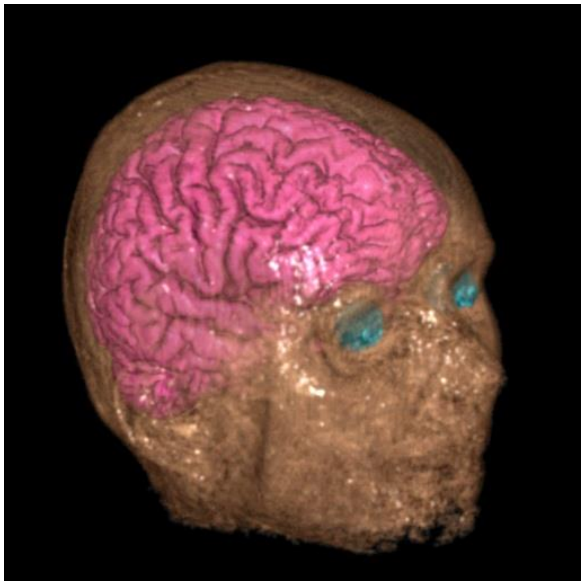


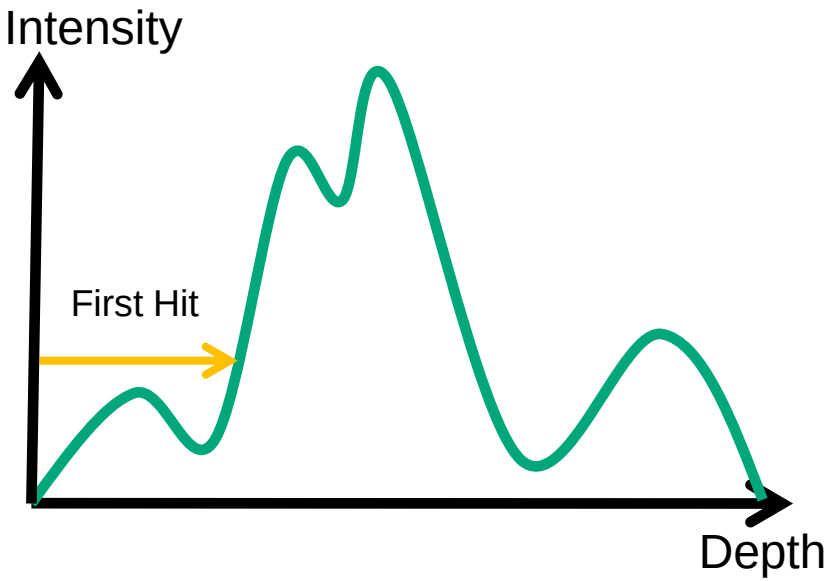


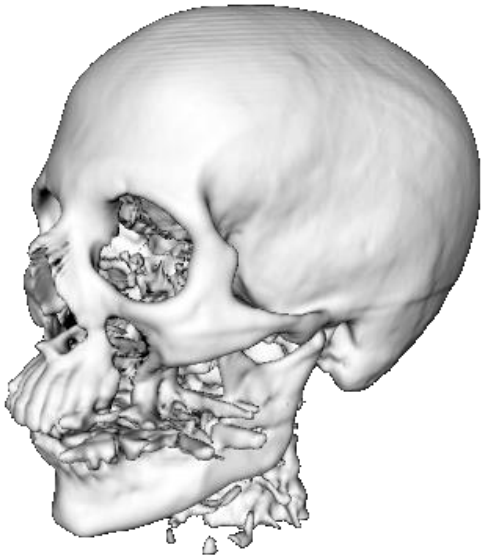

Intensity
Maximum
Accumulate
Average
First Hit
Depth

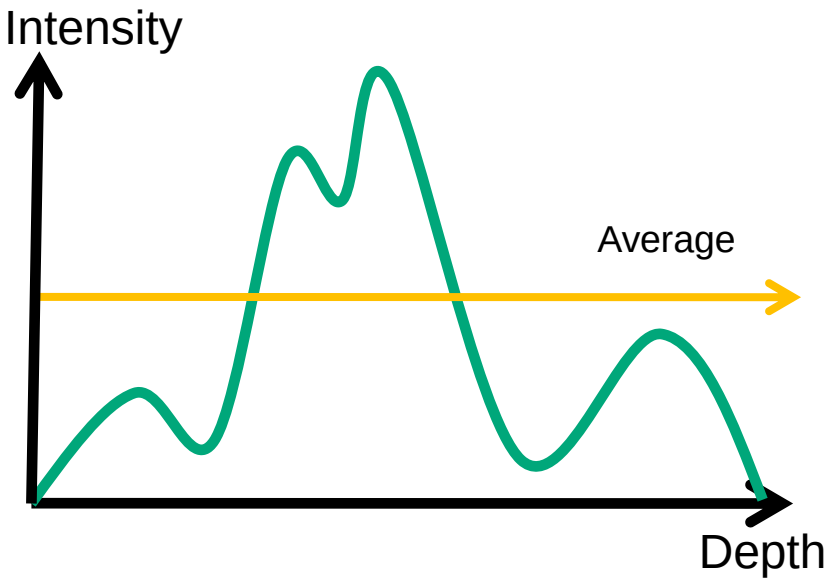


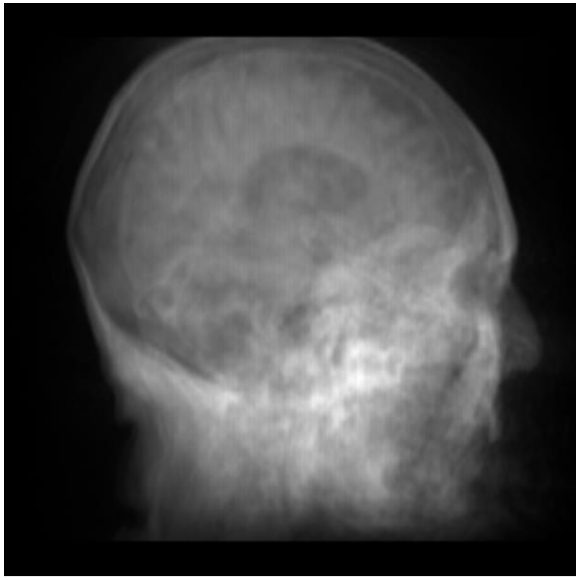

Intensity
Accumulate
Depth




Intensity
First Hit
Depth




Intensity
Average
Depth



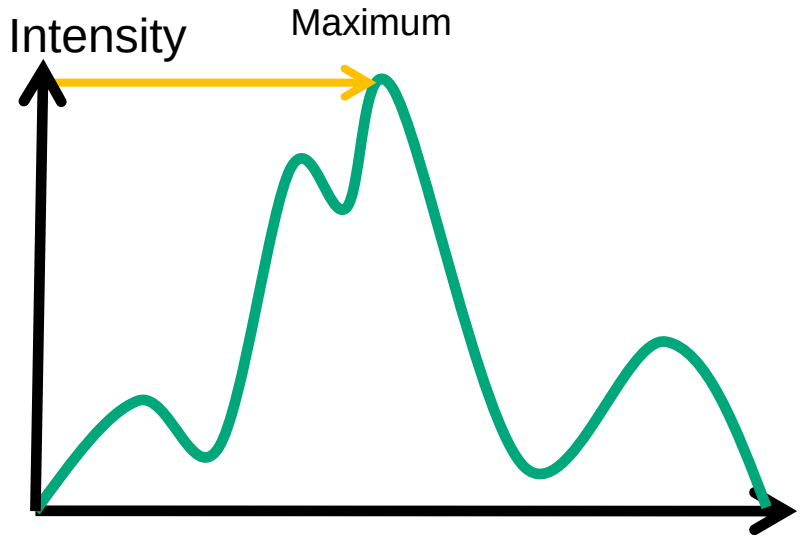
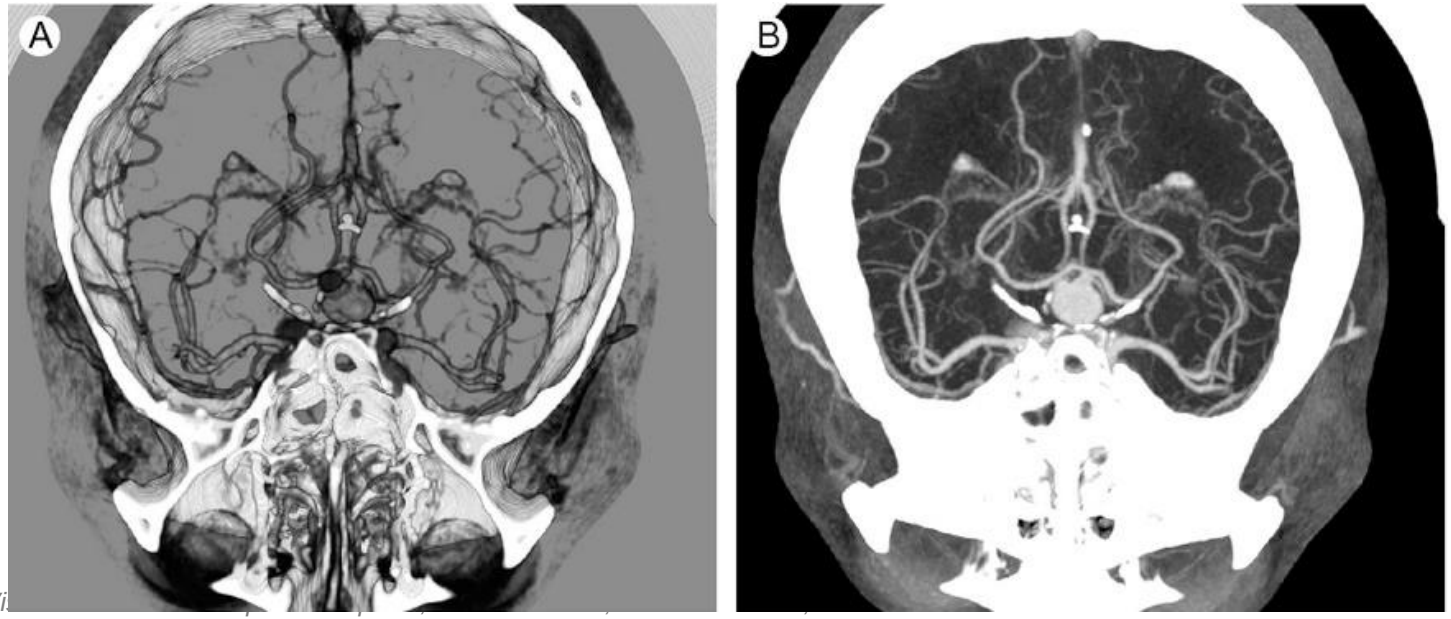
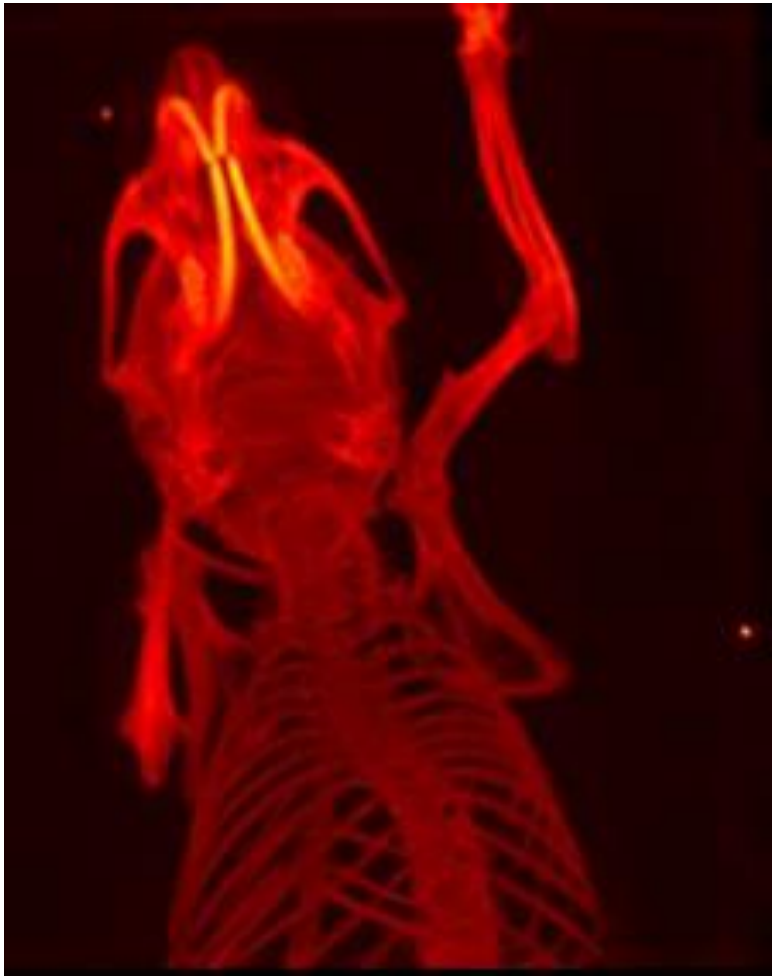


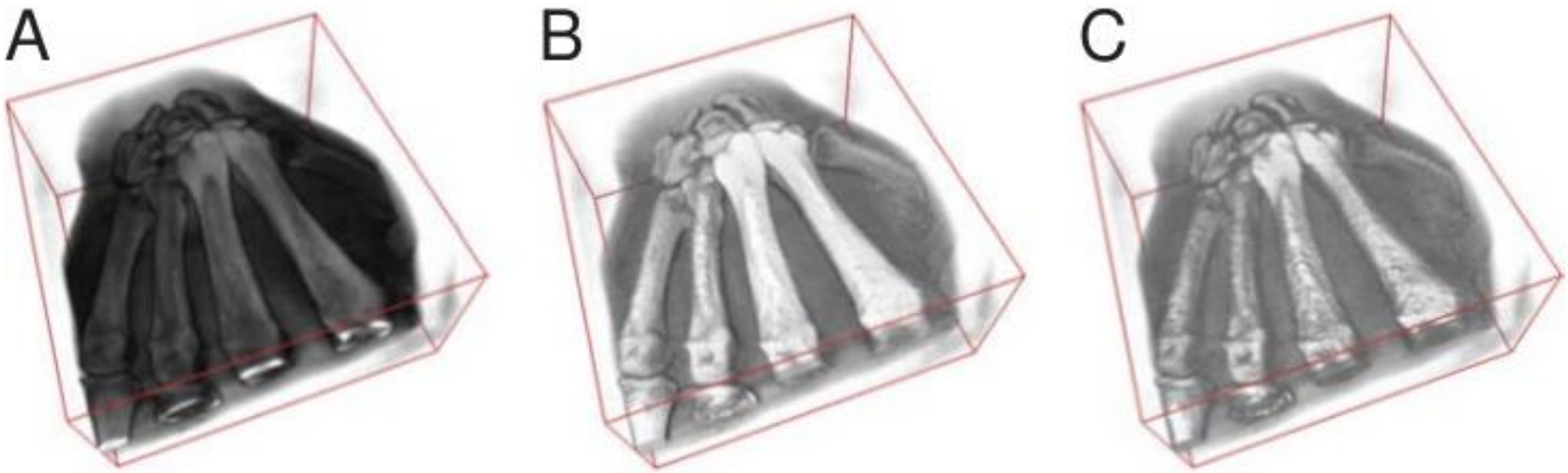
Image Source: Thomas Fogal

Often used to extract vessel structures in magnetic resonance angiograms



Maximum Intensity Projection (MIP) Example





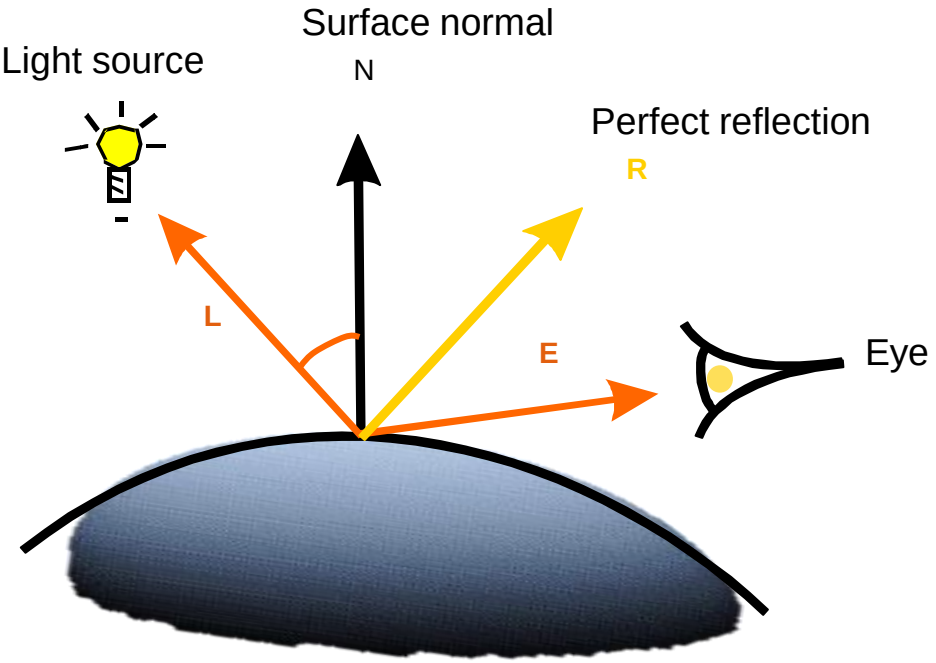
A: Pure emission + absorption, no illumination

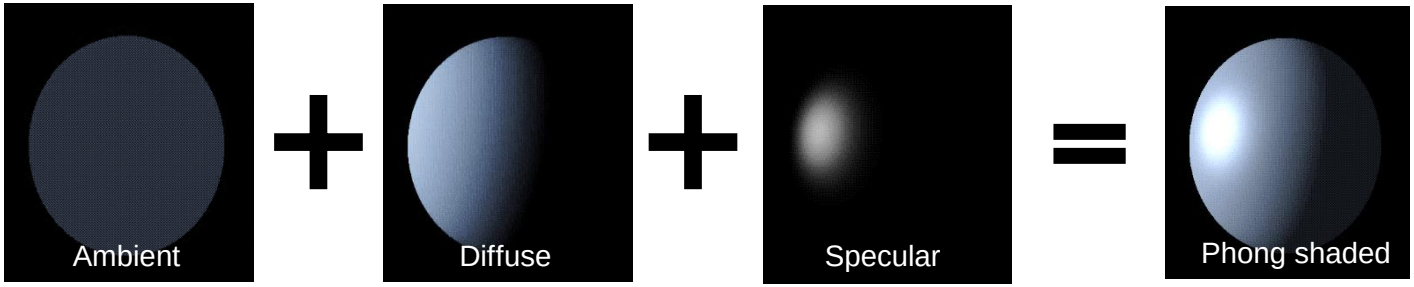
B: Same with diffuse lighting

C: Same with specular lighting

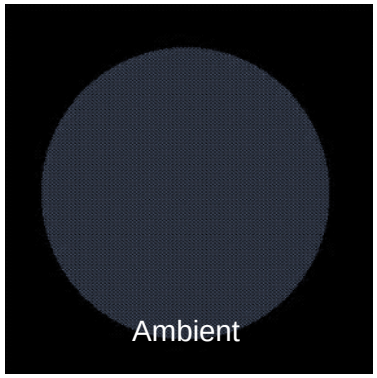
Image Source: Markus Hadwiger and Christof Rezk Salama

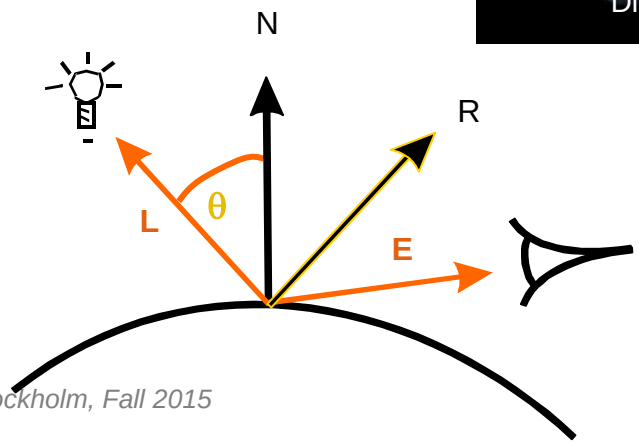
- Ambient light
- Diffuse light
- Specular light



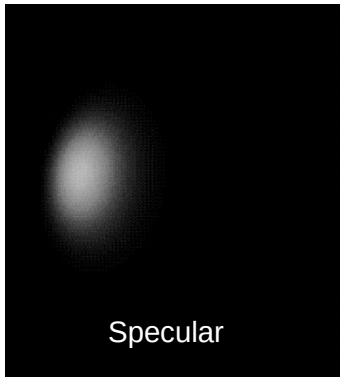


- Ambient light: $C = \mathbf{k_a C_a O_d}$
  - k_a is ambient contribution
  - C_a is color of ambient light
  - O_d is diffuse color of object
- Diffuse light: $C = k_a C_a O_d + \mathbf{k_d C_p O_d \cos(\theta)}$
  - k_d is diffuse contribution
  - C_p is color of point light
  - O_d is diffuse color of object
  - θ is angle of incoming light



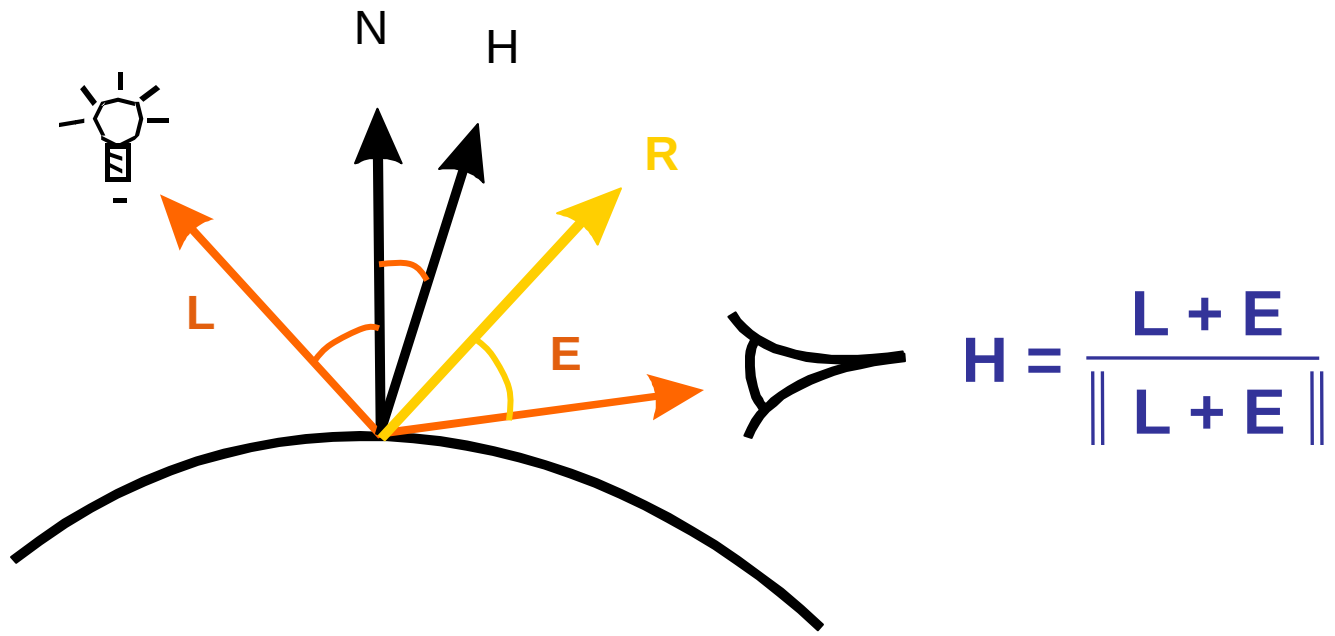


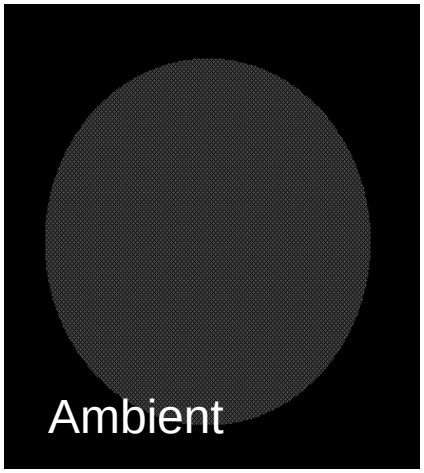
- Specular light: $C = k_a C_a O_d + k_d C_p O_d \cos(\theta) + \mathbf{k_s C_p O_s \cos^n(\sigma)}$
  - k_s is specular contribution
  - C_p is color of point light
  - O_s is specular color of object
  - n is exponent of highlight
  - σ is either
    - angle between reflected light and eye direction (Phong) or
    - angle between surface normal and halfway vector (Blinn-Phong)

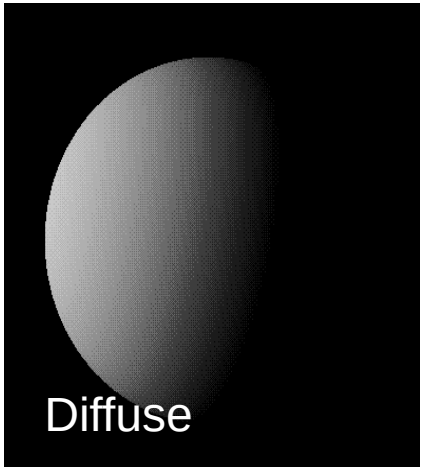


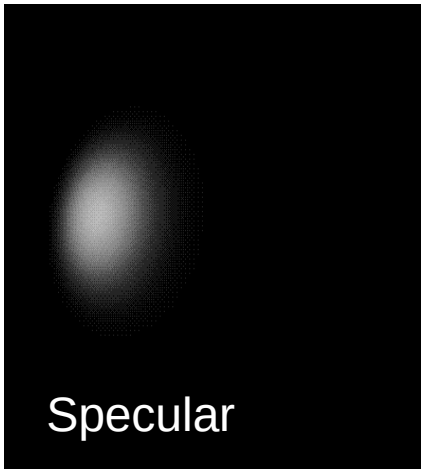
$\cos(\theta) = \mathbf{R} * \mathbf{E}$ (Phong)

$\cos(\sigma) = \mathbf{N} * \mathbf{H}$ (Blinn-Phong)





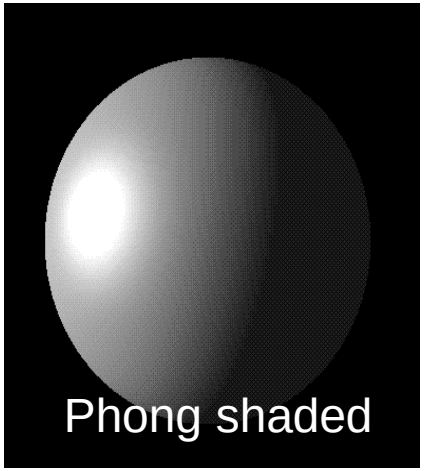


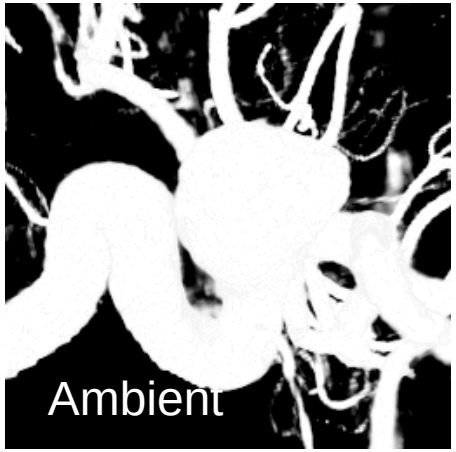


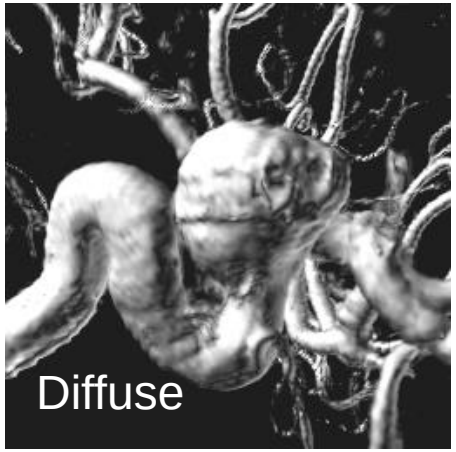
$K_a = 0.1$

$K_d = 0.5$

$K_s = 0.4$







$K_a = 0.1$

$K_d = 0.5$

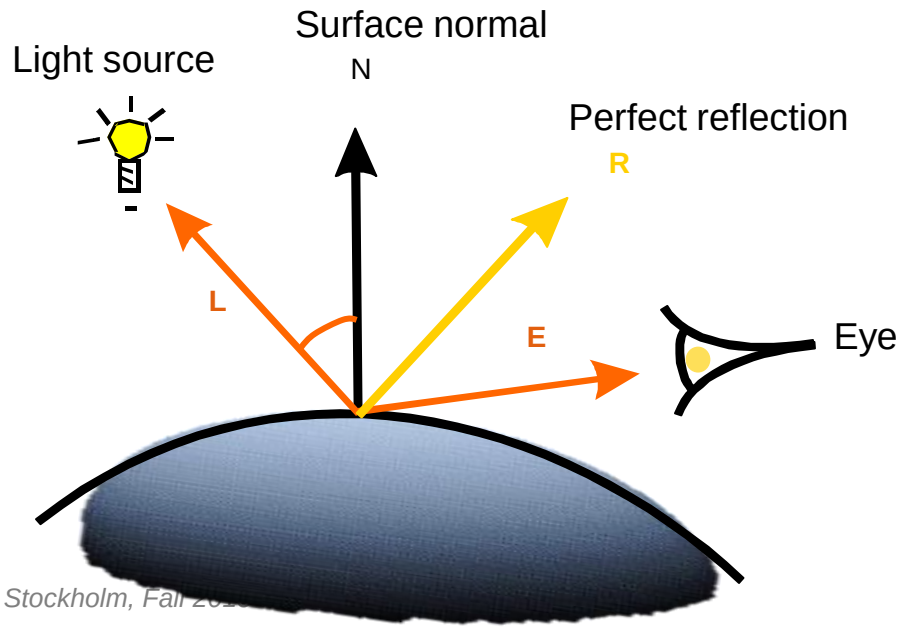
$K_s = 0.4$

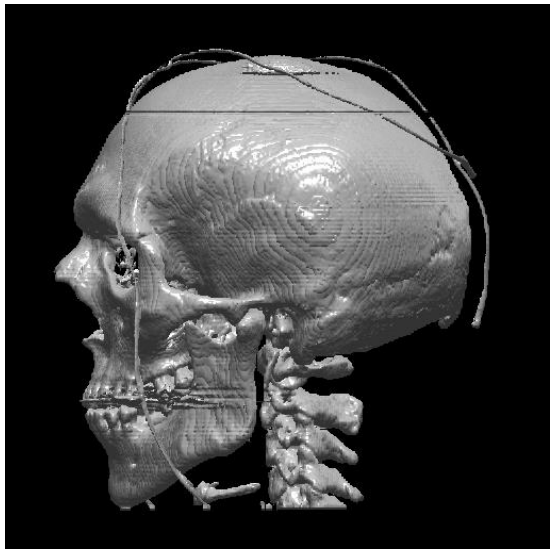
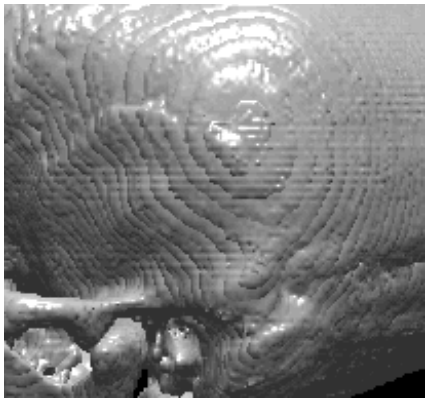
What is the normal vector in a scalar field f(**x**)?

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

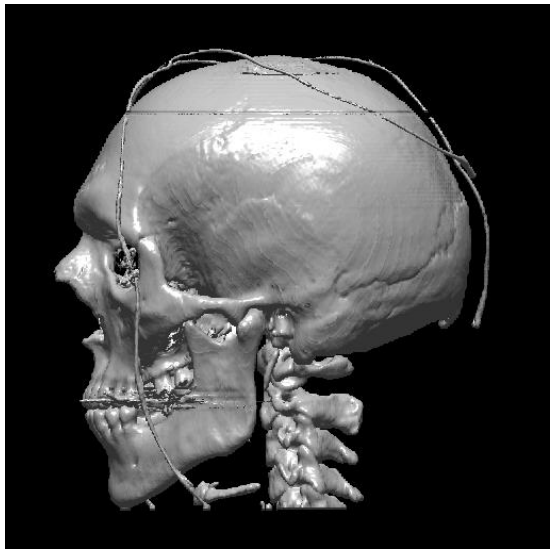
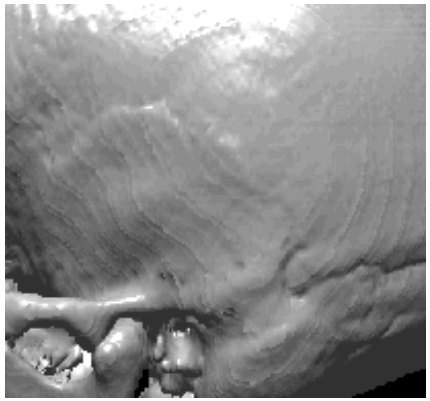
Numerical computation of the gradient:

- forward/backward differences
- central differences
- Sobel Operator

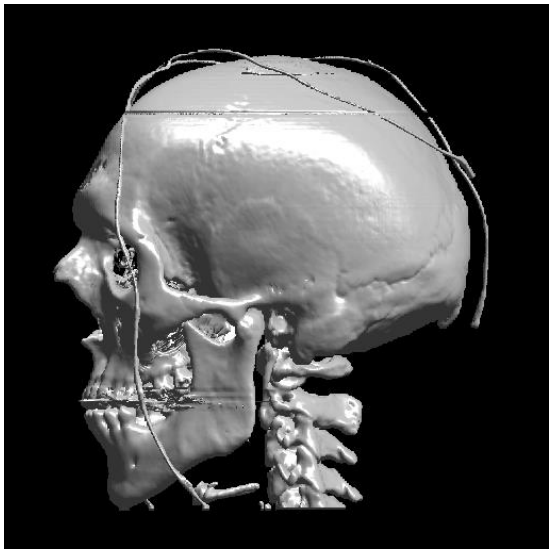
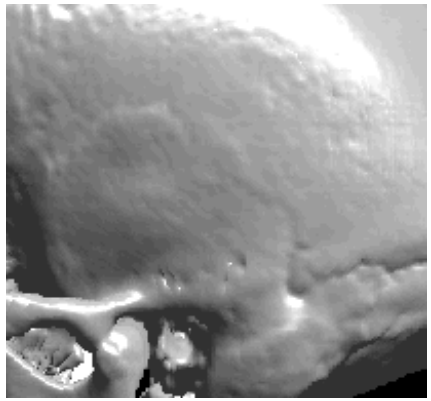


Forward/Backward differences

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:

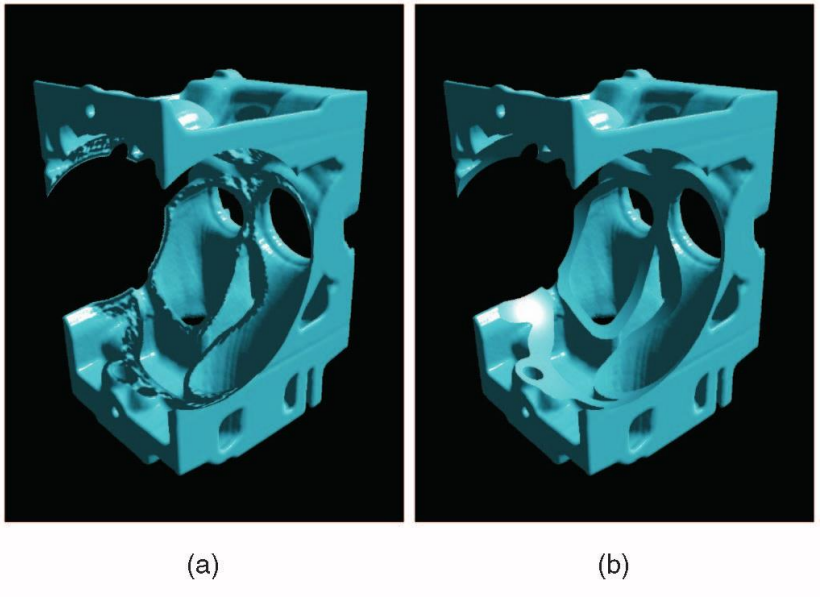


Image Source:
Weiskopf et al., TVCG 2003

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.

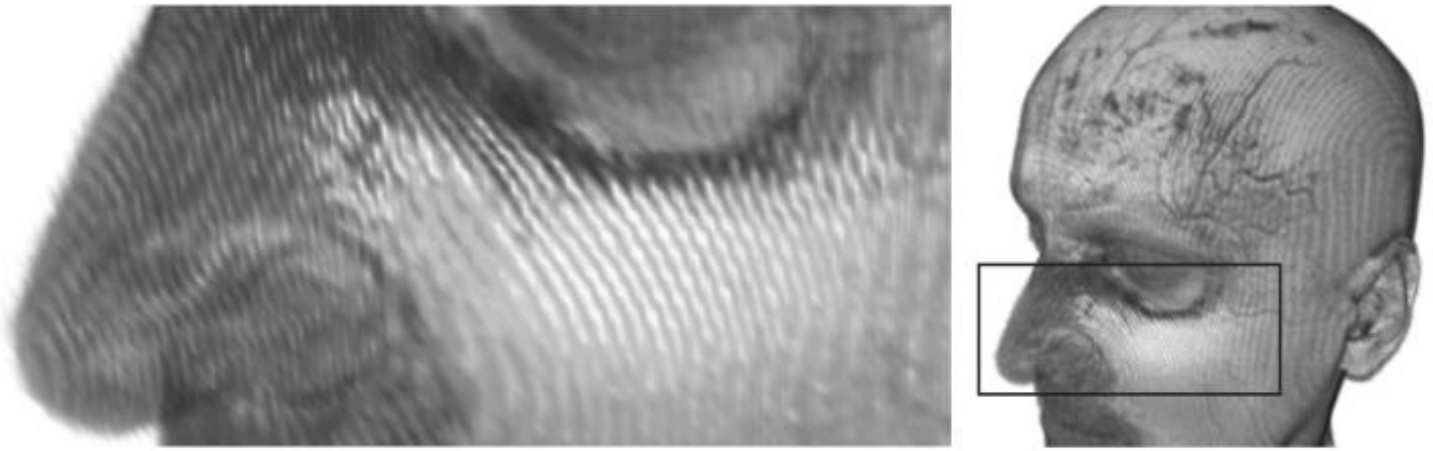
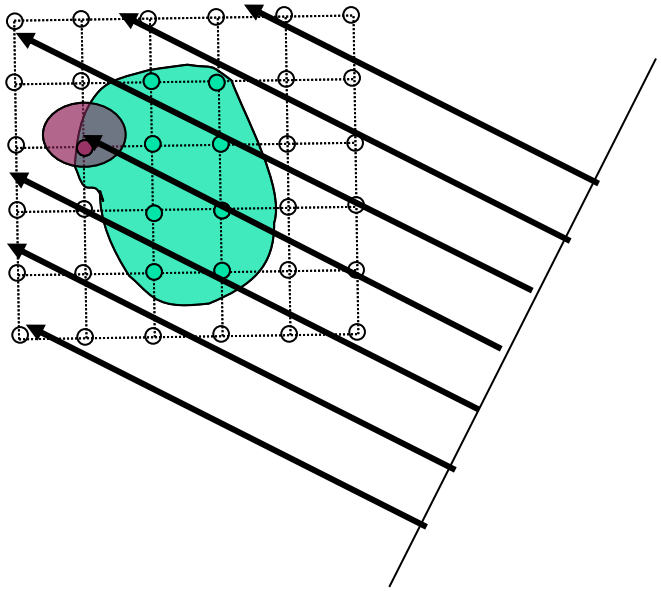


Image Source: Hadwiger and Rezk Salama, 2004

- **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

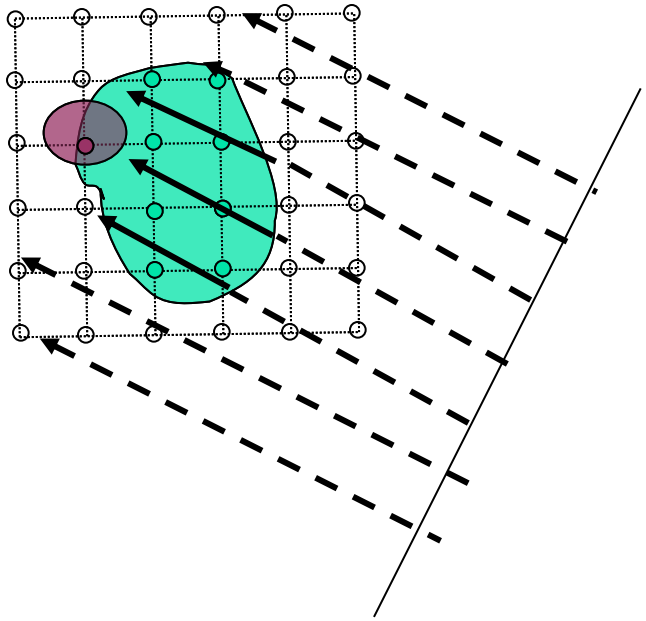


Space leaping

- Skip empty cells

Or: Homogenety acceleration

- Approximate homogeneous regions with fewer sample points



Approaches:

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

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