CS 247 – Scientific Visualization
Lecture 14: Volume Rendering, Pt. 1

Markus Hadwiger, KAUST

Reading Assignment #8 (until Apr 5)

Read (required):

- Real-Time Volume Graphics, Chapter 7 (GPU-Based Ray Casting)
- Real-Time Volume Graphics, Chapter 4.5 – 4.8

Programming Assignments Schedule (tentative)

Assignment 0:	Lab sign-up: join discord, setup github account + get repo	until	**Feb 1**
	Basic OpenGL example		
Assignment 1:	Volume slice viewer	until	**Feb 15**
Assignment 2:	Iso-contours (marching squares)	until	**Mar 1**
Assignment 3:	Iso-surface rendering (marching cubes)	until	**Mar 15**
Assignment 4:	Volume ray-casting, part 1	until	**Apr 12**
	Volume ray-casting, part 2	until	**Apr 19**
Assignment 5:	Flow vis, part 1 (hedgehog plots, streamlines, pathlines)	until	**May 3**
Assignment 6:	Flow vis, part 2 (LIC with color coding)	until	**May 13**

The Gradient as Normal Vector

Gradient of the scalar field gives direction+magnitude of fastest change

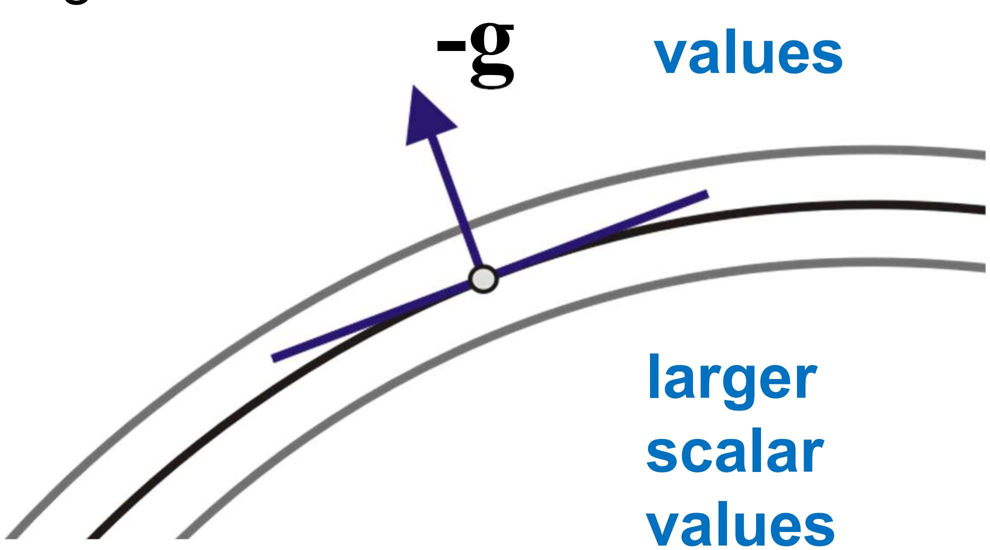
$$\mathbf{g} = \nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)^{\mathbf{T}}$$

(only correct in Cartesian coordinates: see later)

Local approximation to isosurface at any point:
tangent plane = plane orthogonal to gradient

Normal of this isosurface:
normalized gradient vector
(negation is common convention)

$$\mathbf{n} = -\mathbf{g}/|\mathbf{g}|$$



Gradient and Directional Derivative

Gradient $\nabla f(x,y,z)$ of scalar function $f(x,y,z)$:

$$\nabla f(x,y,z) = \left(\frac{\partial f(x,y,z)}{\partial x}, \frac{\partial f(x,y,z)}{\partial y}, \frac{\partial f(x,y,z)}{\partial z} \right)^T$$

(only correct in Cartesian coordinates: see later)

Directional derivative in direction $\mathbf{u}$:

$$D_{\mathbf{u}} f(x,y,z) = \nabla f(x,y,z) \cdot \mathbf{u}$$

And therefore also:

$$D_{\mathbf{u}} f(x,y,z) = ||\nabla f|| \; ||\mathbf{u}|| \; \cos\theta$$

Gradient and Directional Derivative

Gradient $\nabla f(x,y,z)$ of scalar function $f(x,y,z)$:

$$\nabla f(x,y,z) = \left(\frac{\partial f(x,y,z)}{\partial x}, \frac{\partial f(x,y,z)}{\partial y}, \frac{\partial f(x,y,z)}{\partial z} \right)^T$$

(only correct in Cartesian coordinates: see later)

(Cartesian vector components; basis vectors not shown)

But: always need **basis vectors**! With Cartesian basis:

$$\nabla f(x,y,z) = \frac{\partial f(x,y,z)}{\partial x}\,\mathbf{i} + \frac{\partial f(x,y,z)}{\partial y}\,\mathbf{j} + \frac{\partial f(x,y,z)}{\partial z}\,\mathbf{k}$$

Gradients as Differential Forms (1-Forms)

The Gradient as a Differential Form

The gradient as a *differential* (differential 1-form) is the “primary” concept (also “total differential” or “total derivative”)

$$df = \frac{\partial f}{\partial x}\, dx + \frac{\partial f}{\partial y}\, dy + \frac{\partial f}{\partial z}\, dz$$

A differential 1-form is a scalar-valued linear function that takes a (direction) vector as input, and gives a scalar as output

Each of the 1-forms df, dx, dy, dz takes direction vector as input, gives scalar output

In the expression of the gradient df above, all 1-forms on the right-hand side get the same vector as input

df is simply a linear combination of the coordinate differentials dx, dy, dz

The Gradient as a Differential Form

The gradient as a *differential* (differential 1-form) is the "primary" concept (also "total differential" or "total derivative")

$$df = \frac{\partial f}{\partial x}\, dx + \frac{\partial f}{\partial y}\, dy + \frac{\partial f}{\partial z}\, dz$$

The directional derivative and the gradient vector

$$D_{\mathbf{u}} f = df(\mathbf{u})$$
$$df(\mathbf{u}) = \nabla f \cdot \mathbf{u}$$

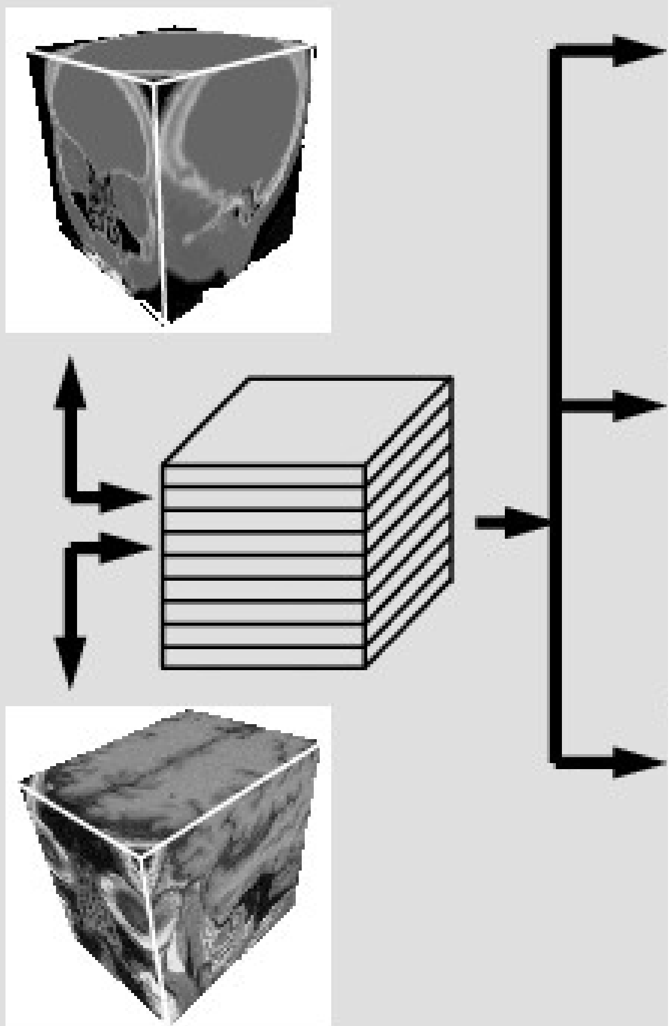
The gradient vector is then *defined*, such that:

$$\nabla f \cdot \mathbf{u} := df(\mathbf{u})$$

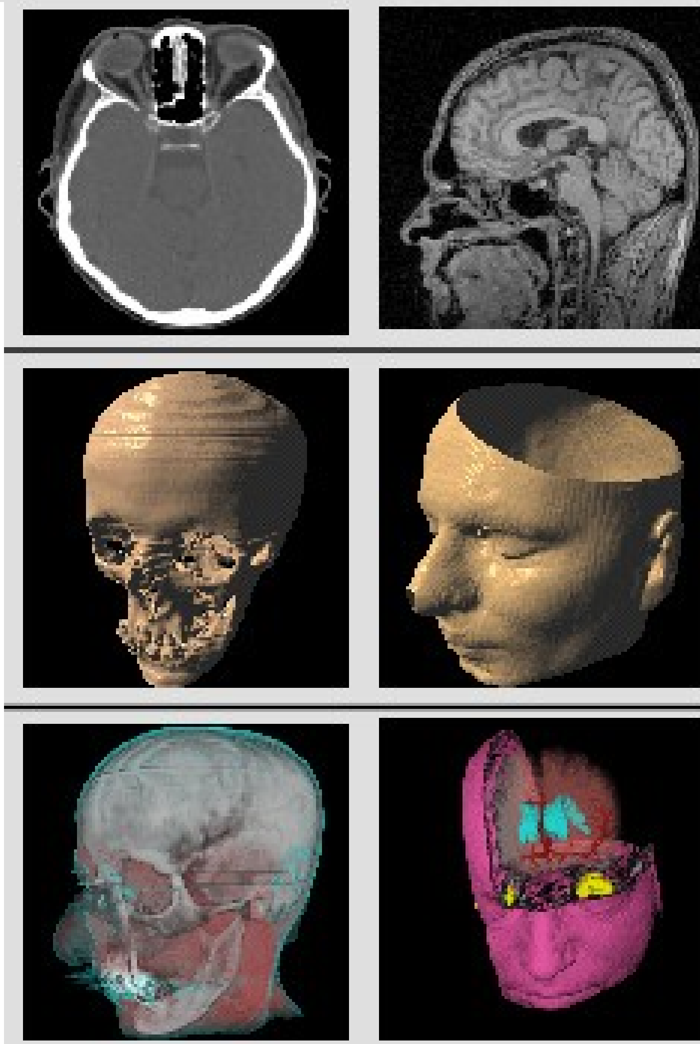
Volume Rendering

Theory

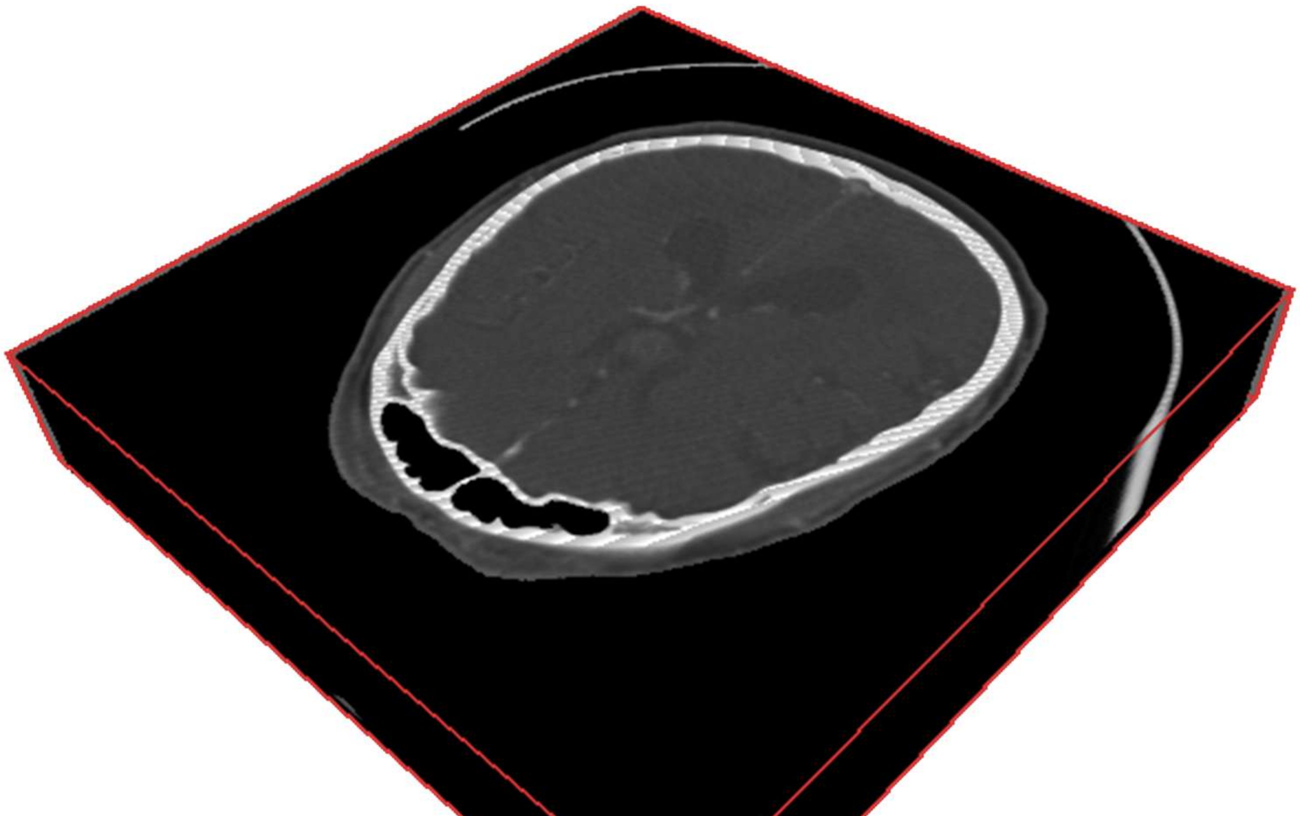
Volume Visualization



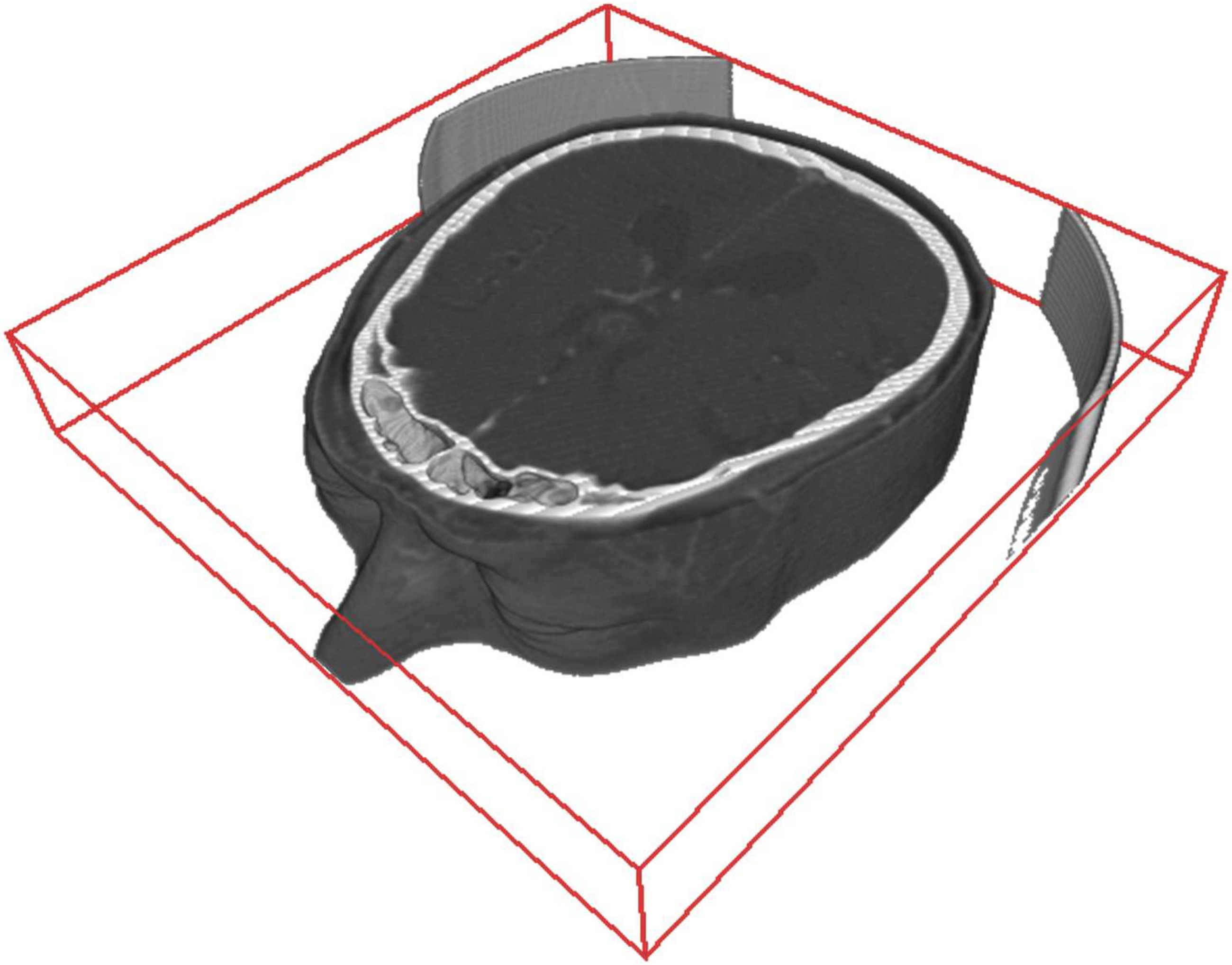
- 2D visualization
  slice images
  (or multi-planar
  reformatting MPR)
- *Indirect*
  3D visualization
  isosurfaces
  (or surface-shaded
  display: SSD)
- *Direct*
  3D visualization
  (direct volume
  rendering: DVR)



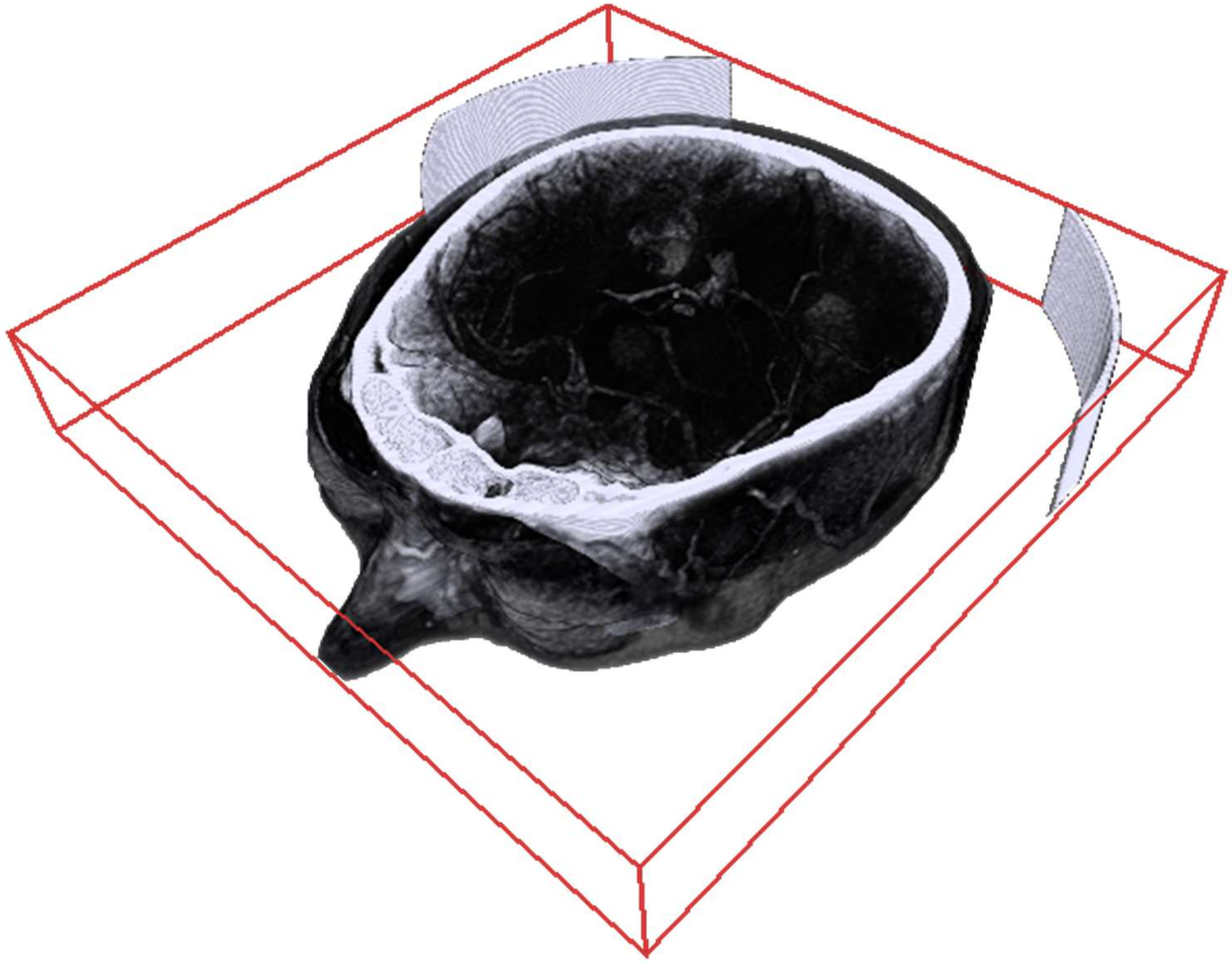
Direct Volume Rendering



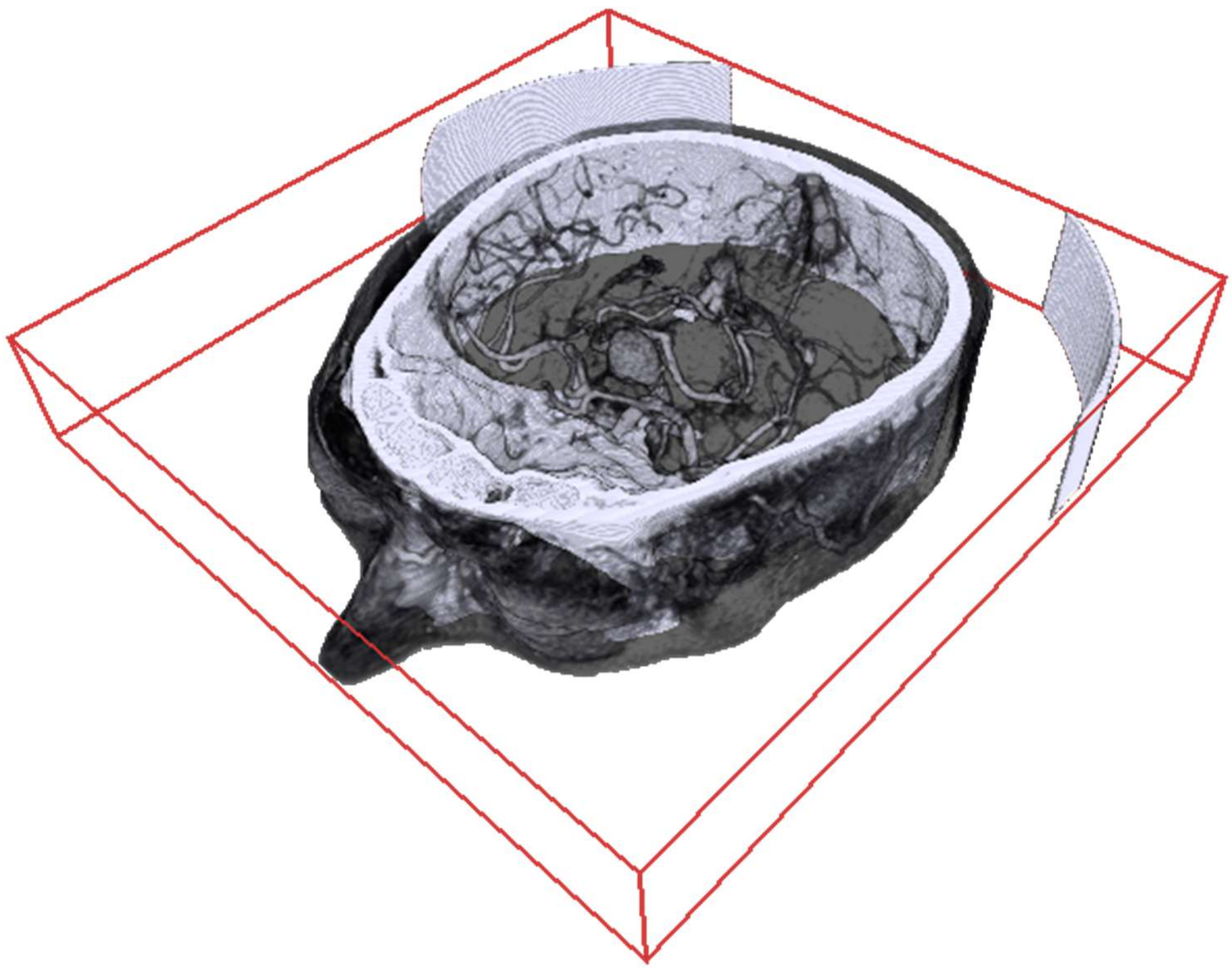
Direct Volume Rendering



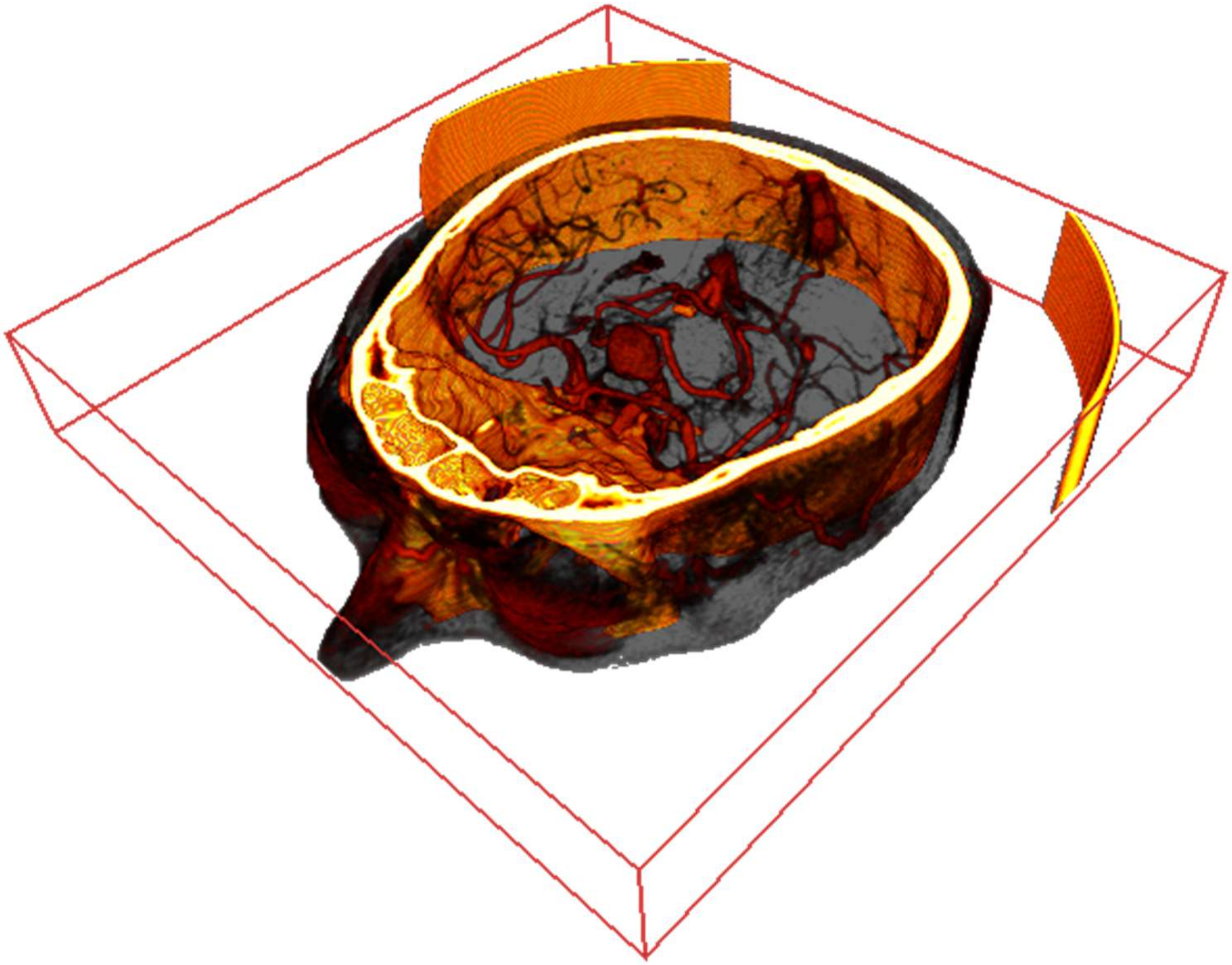
Direct Volume Rendering



Direct Volume Rendering



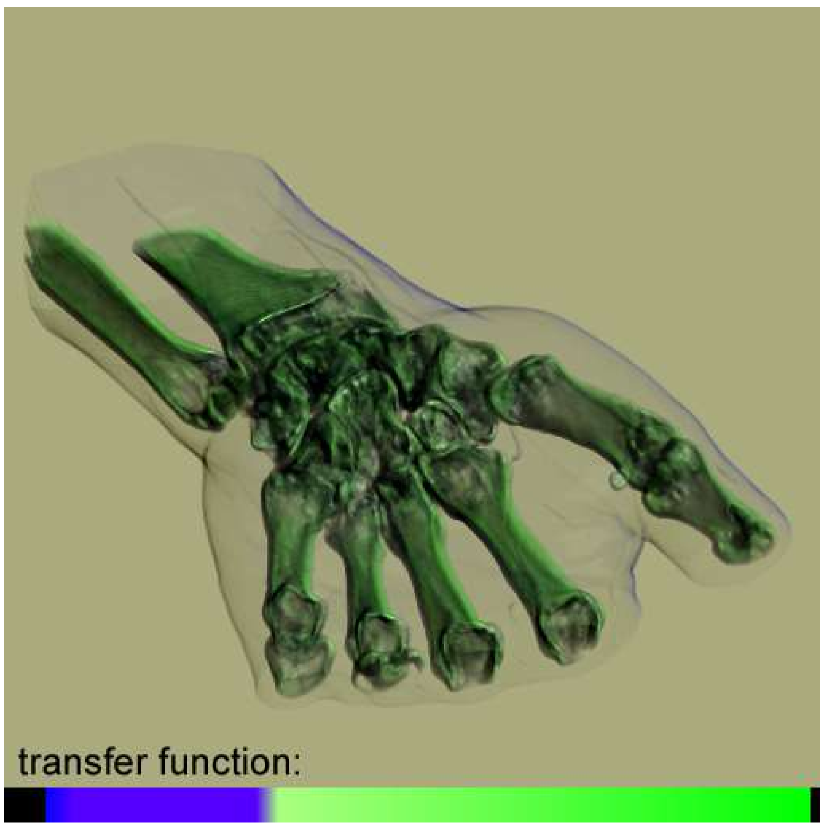
Direct Volume Rendering



Transparent Volumes vs. Isosurfaces

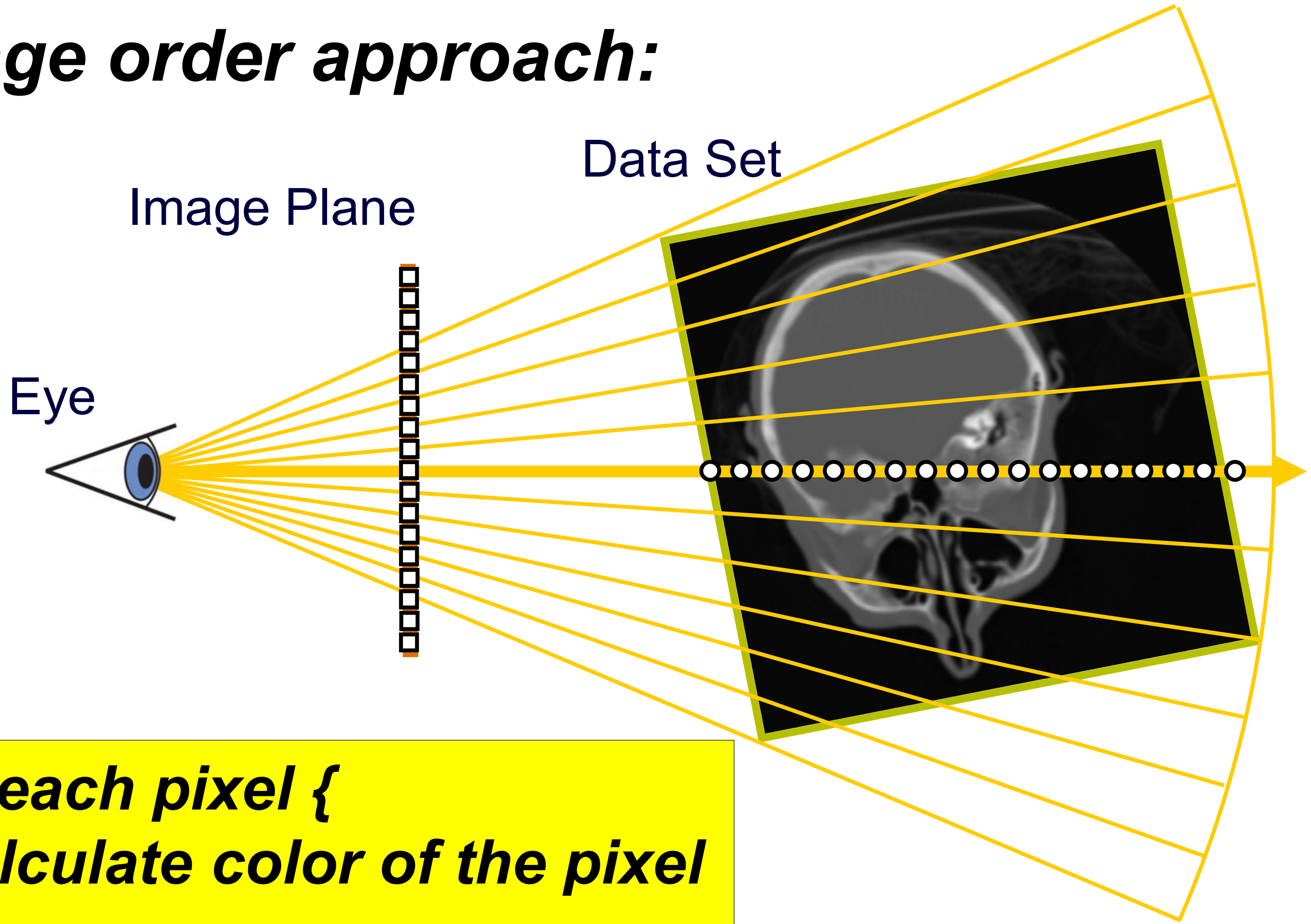
The *transfer function* assigns *optical properties* to data

- Translucent volumes
- But also: isosurface rendering using step function as transfer function



Direct Volume Rendering: Image Order

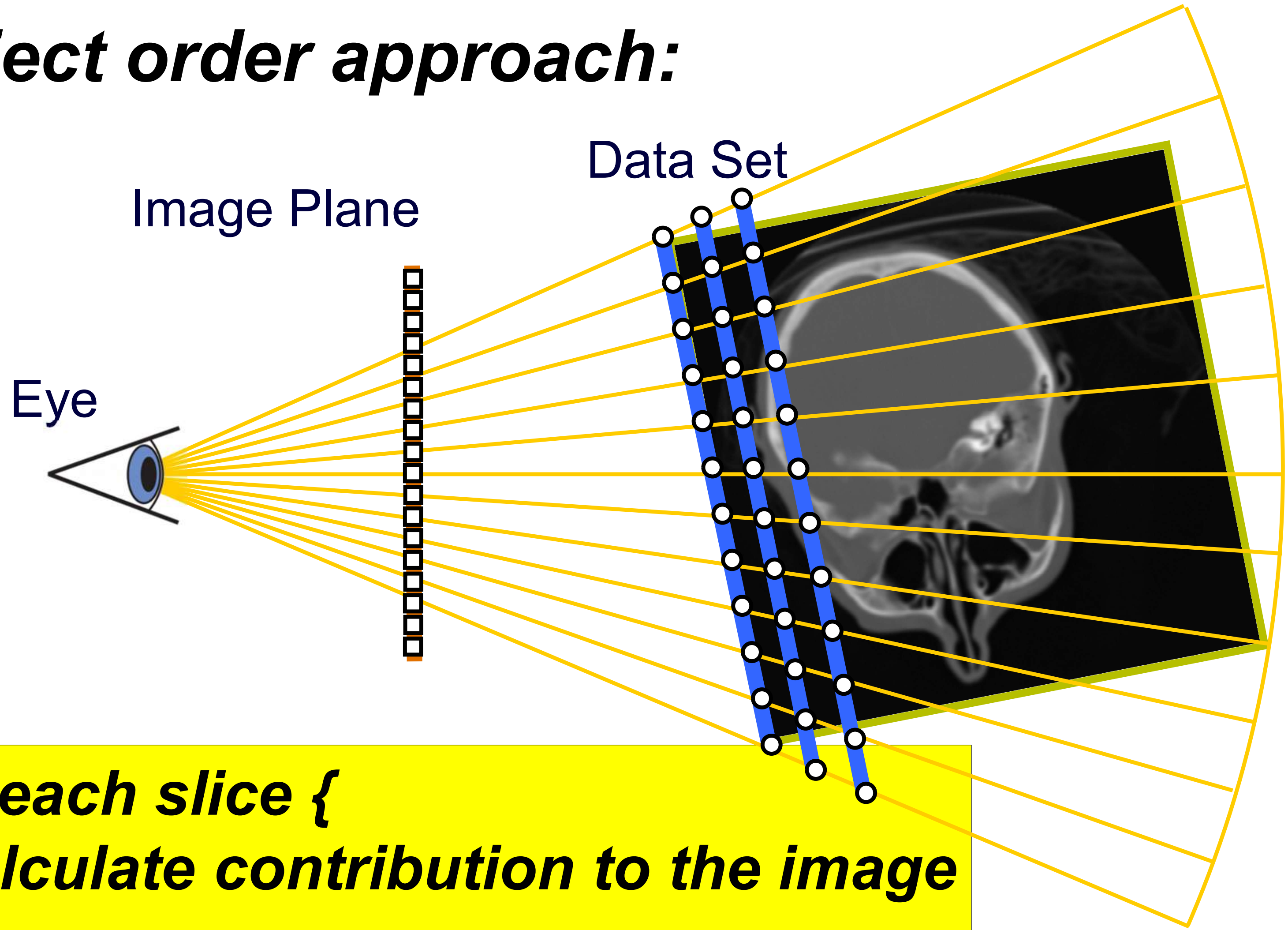
Image order approach:



For each pixel {
calculate color of the pixel
}

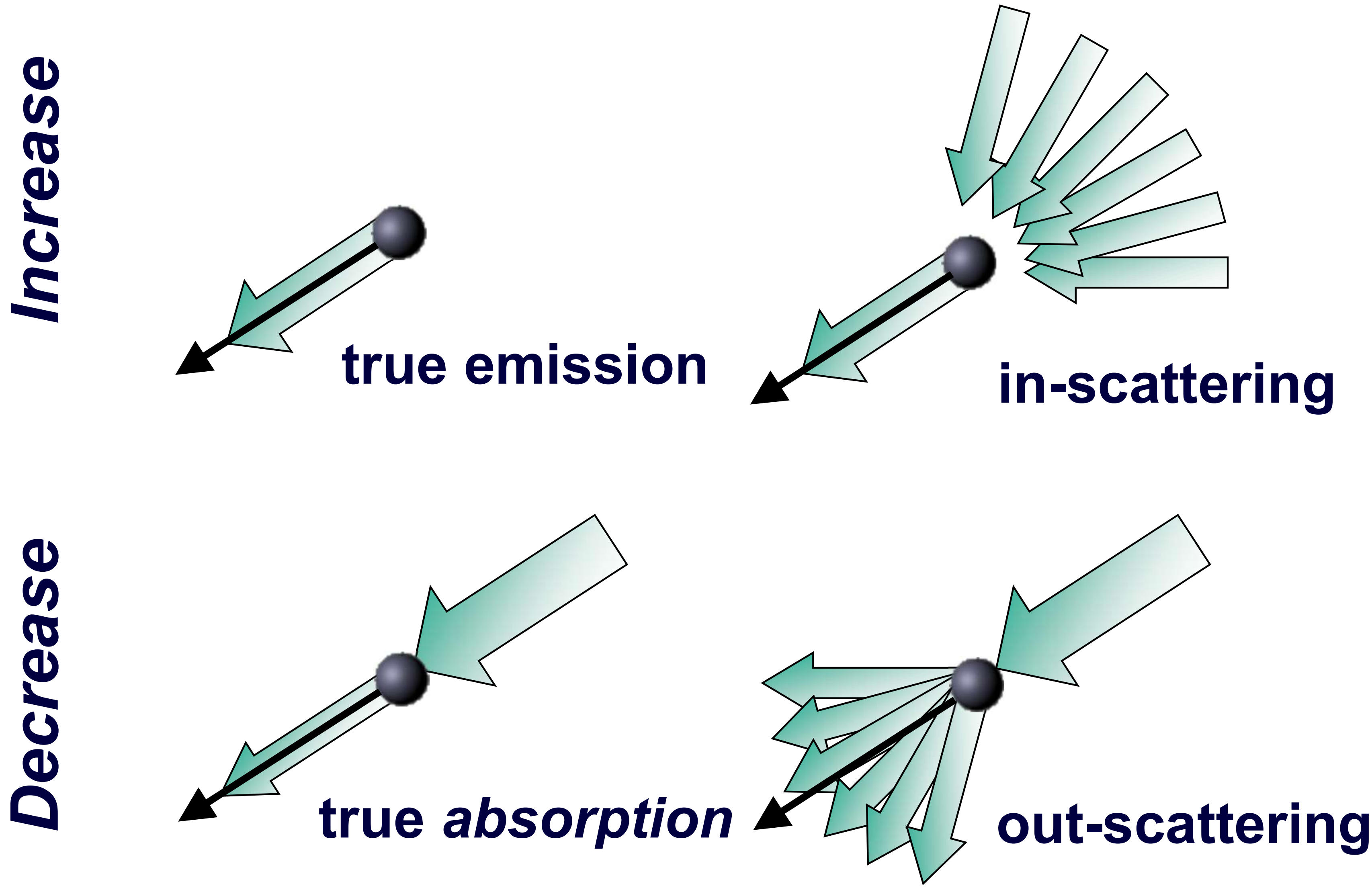
Direct Volume Rendering: Object Order

Object order approach:

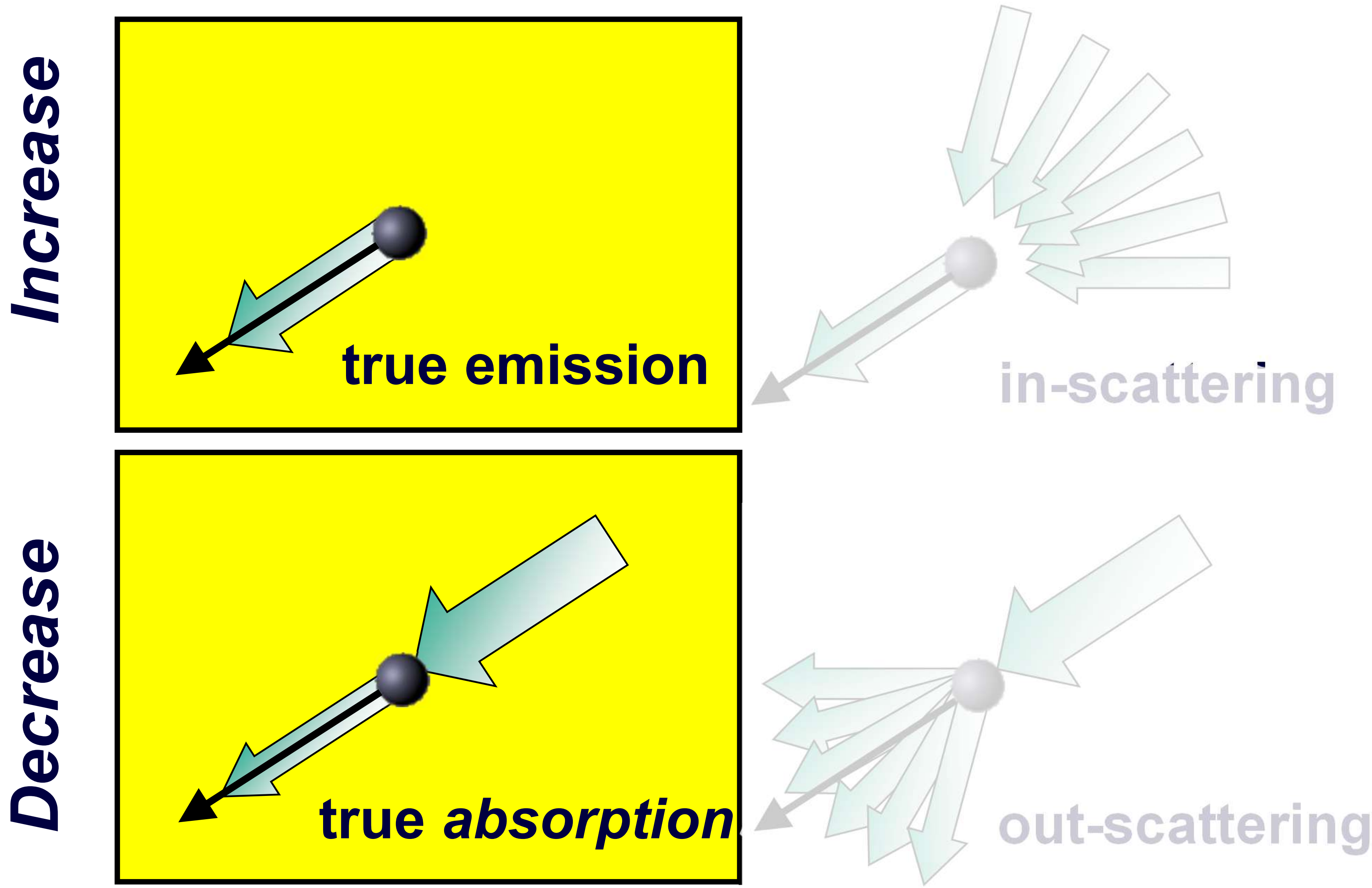


For each slice {
calculate contribution to the image
}

Physical Model of Radiative Transfer



Physical Model of Radiative Transfer



Optical Models: Physical Model gives ODE

Optical Models for Direct Volume Rendering, Nelson Max

Emission-Absorption optical model

$$\frac{dI}{ds}(s) = q(s) - \kappa(s)\, I(s)$$

Right-hand side: *Rates of change* (derivatives) of light intensity along ray

Absorption rate is proportional to light intensity: Solution is exponential

Volume Rendering Integral

Volume rendering integral
for *Emission Absorption* model

true emission **true *absorption***

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

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

Iterative/recursive numerical solutions:

Back-to-front compositing

$$C'_i \;=\; C_i + (1 - A_i)C'_{i-1}$$

here, all colors are *associated* colors!

Front-to-back compositing

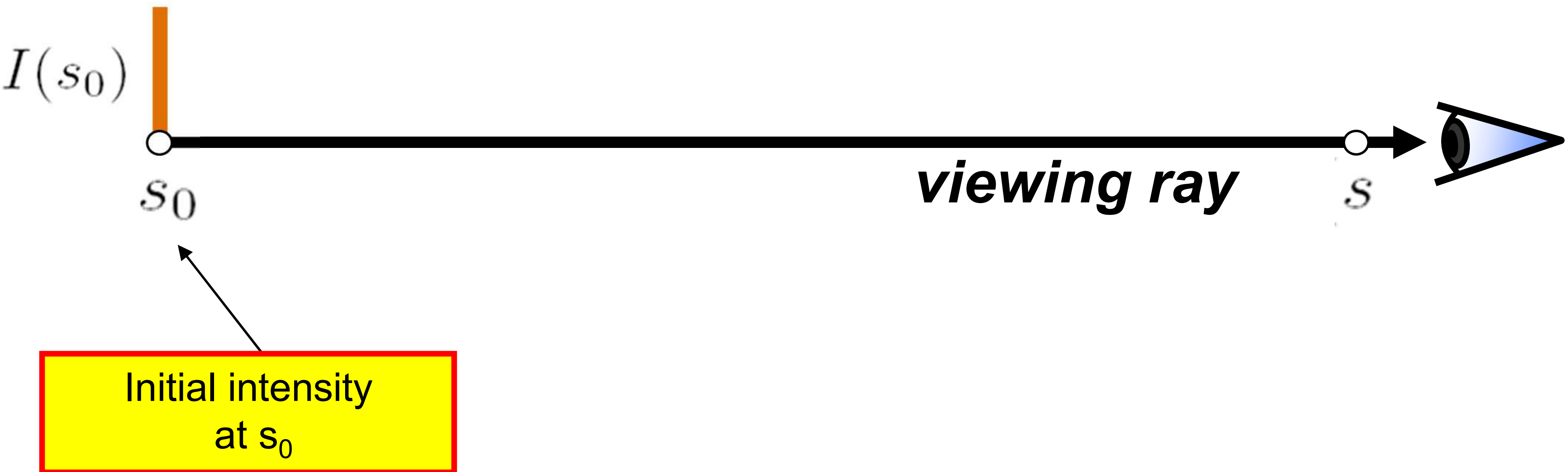
$$C'_i \;=\; C'_{i+1} + (1 - A'_{i+1})C_i$$

$$A'_i \;=\; A'_{i+1} + (1 - A'_{i+1})A_i$$

Volume Rendering Integral

How do we determine the radiant energy along the ray?

Physical model: emission and absorption, no scattering

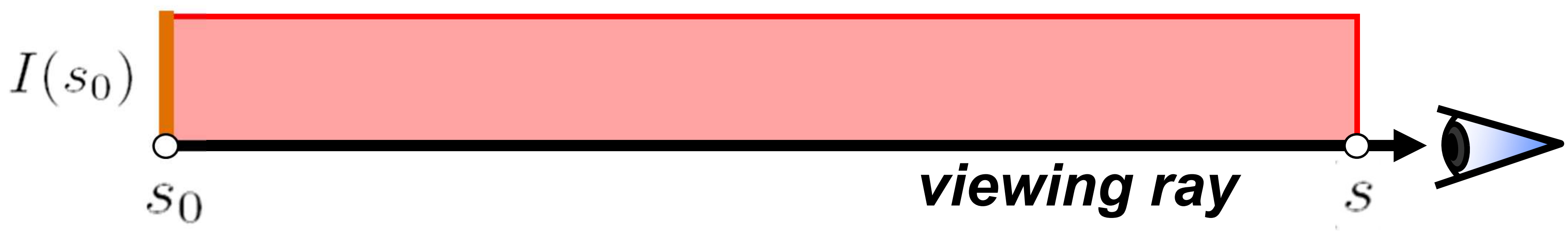


$$I(s) = I(s_0)$$

Volume Rendering Integral

How do we determine the radiant energy along the ray?

Physical model: emission and absorption, no scattering



Initial intensity at s_0

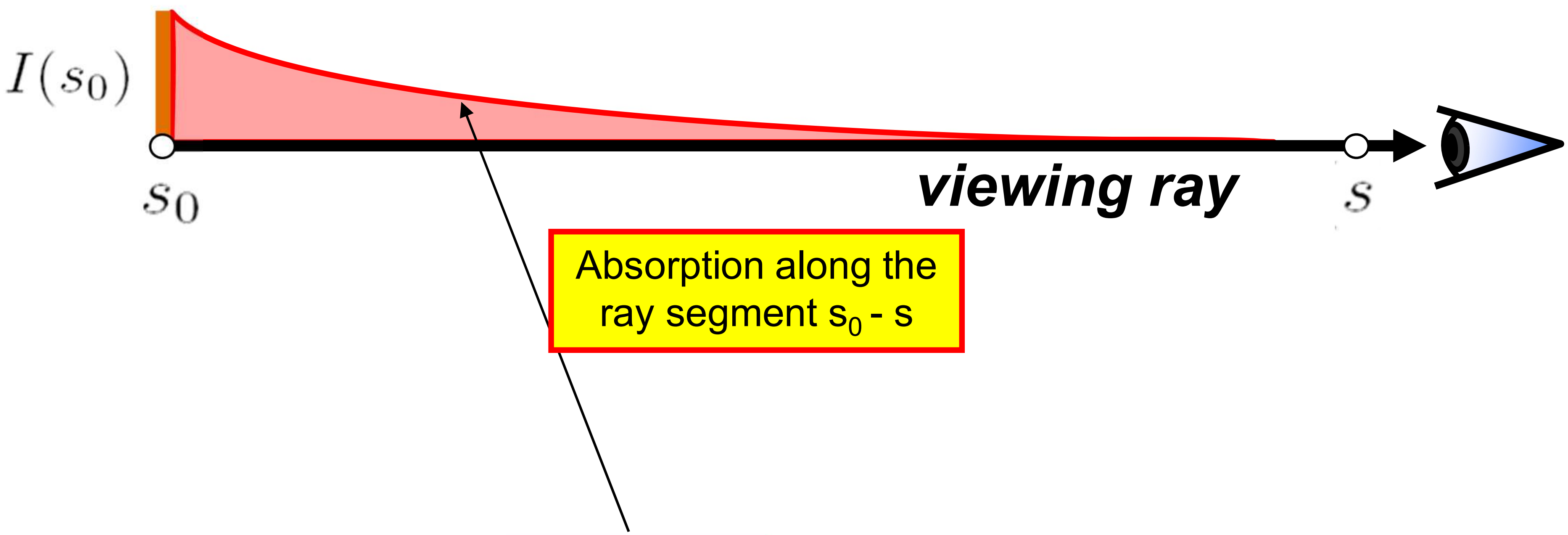
$$I(s) = I(s_0)$$

Without absorption all the initial radiant energy would reach the point ***s.***

Volume Rendering Integral

How do we determine the radiant energy along the ray?

Physical model: emission and absorption, no scattering

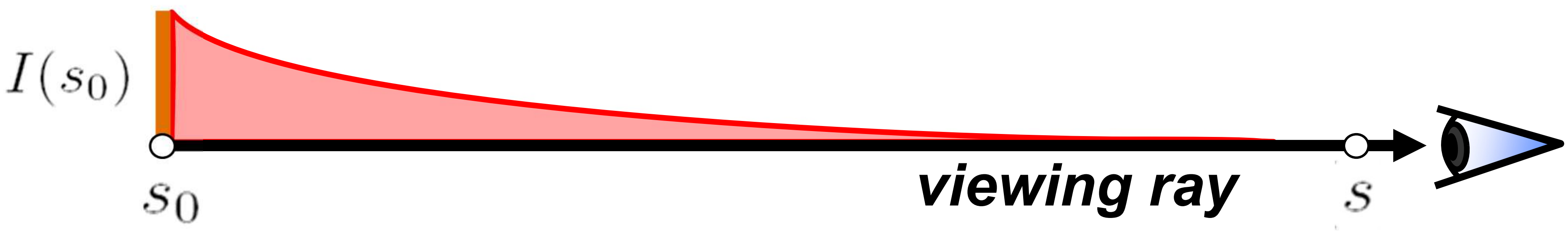


$$I(s) = I(s_0)\, e^{-\tau(s_0,s)}$$

Volume Rendering Integral

How do we determine the radiant energy along the ray?

Physical model: emission and absorption, no scattering



Optical depth τ
Absorption κ

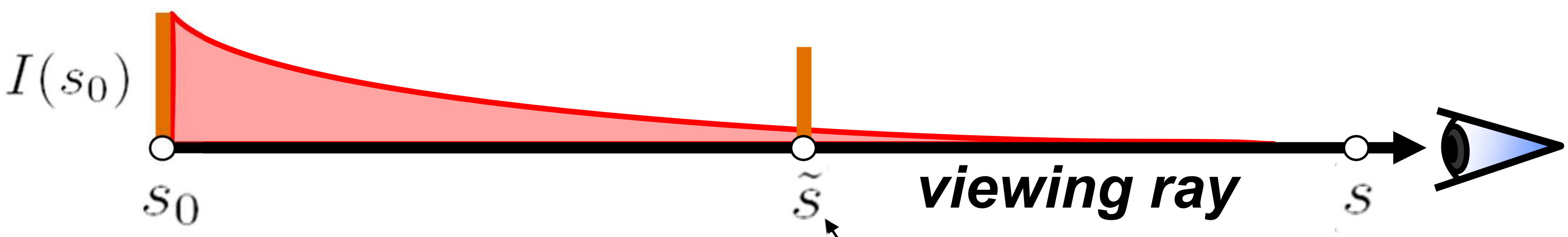
$$\tau(s_1, s_2) = \int_{s_1}^{s_2} \kappa(s)\, ds.$$

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

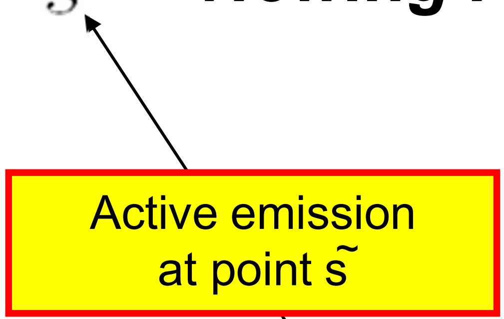
Volume Rendering Integral

How do we determine the radiant energy along the ray?

Physical model: emission and absorption, no scattering



One point $\tilde{s}$ along the viewing ray emits additional radiant energy.

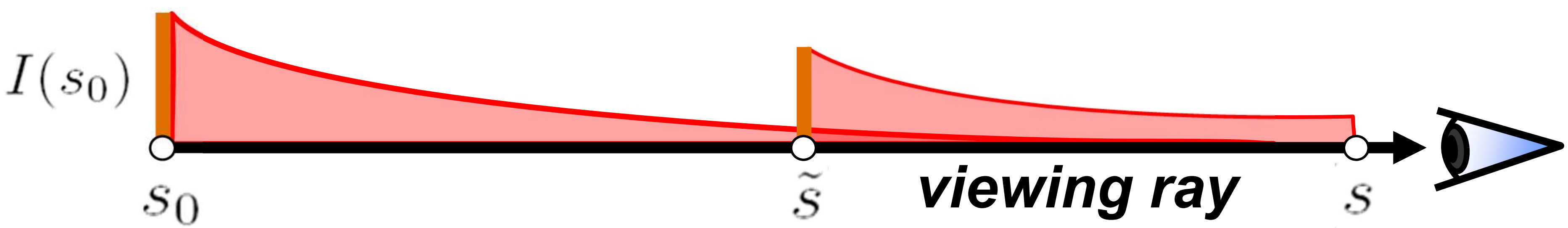


$$I(s) = I(s_0)\, e^{-\tau(s_0,s)} + q(\tilde{s})$$

Volume Rendering Integral

How do we determine the radiant energy along the ray?

Physical model: emission and absorption, no scattering



Every point $\tilde{s}$ along the viewing ray emits additional radiant energy

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

Thank you.

Thanks for material

- Helwig Hauser
- Eduard Gröller
- Daniel Weiskopf
- Torsten Möller
- Ronny Peikert
- Philipp Muigg
- Christof Rezk-Salama