# Computer Graphics I

## Lecture 15: Volume rendering I

**Xiaopei LIU**

School of Information Science and Technology
ShanghaiTech University

# Rendering over surfaces

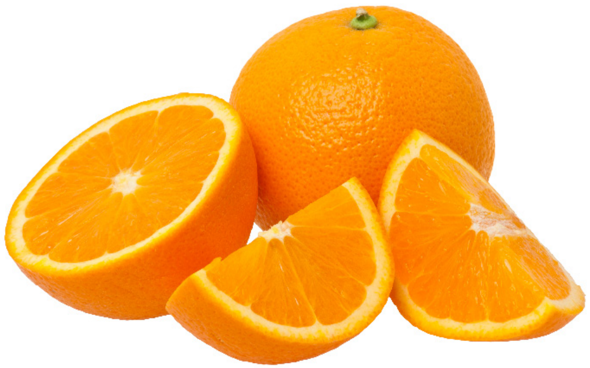
- **Optical rays do not go into surfaces**
  – Reflection/refraction happens at interfaces
  – No inter-reflection among media particles

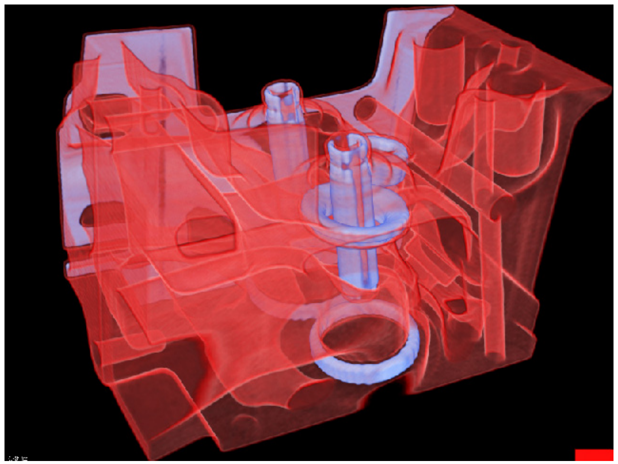
# Volumetric effects

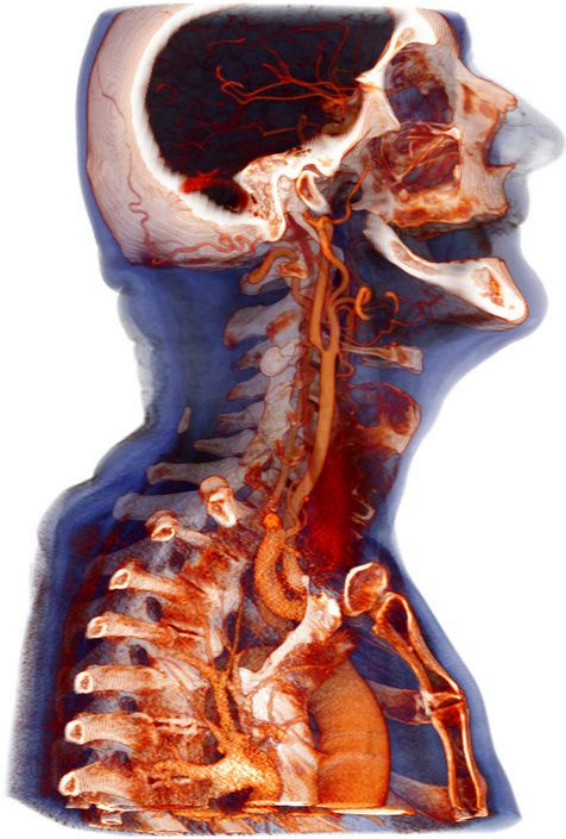
- **What lacks for surface rendering?**
  - Volumetric emission, absorption, scattering, etc.

# Volumetric modeling

- **Solid materials with different transmittance**







# Where can it be used?

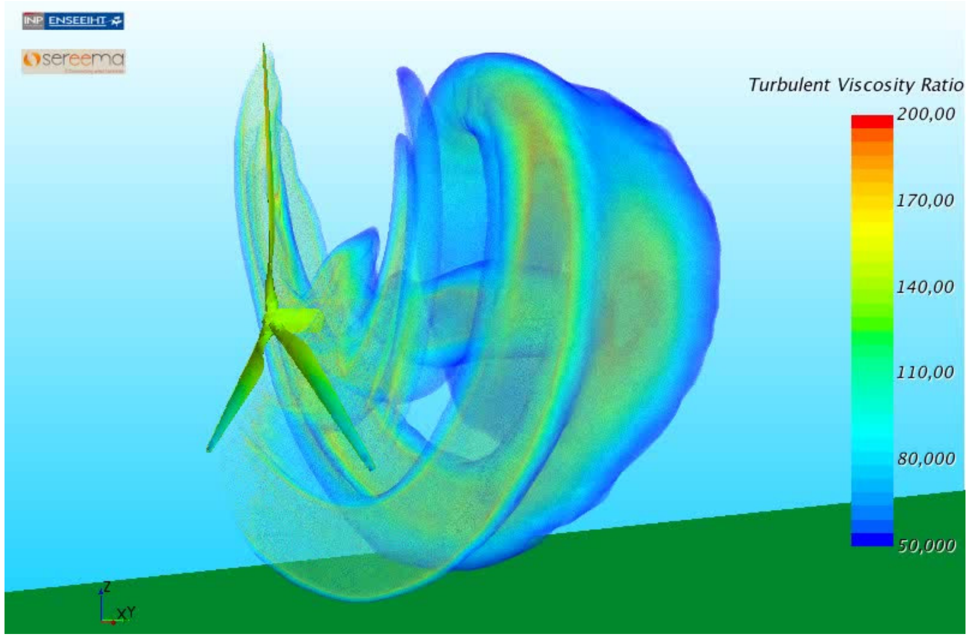
- **Realistic rendering of translucent objects**

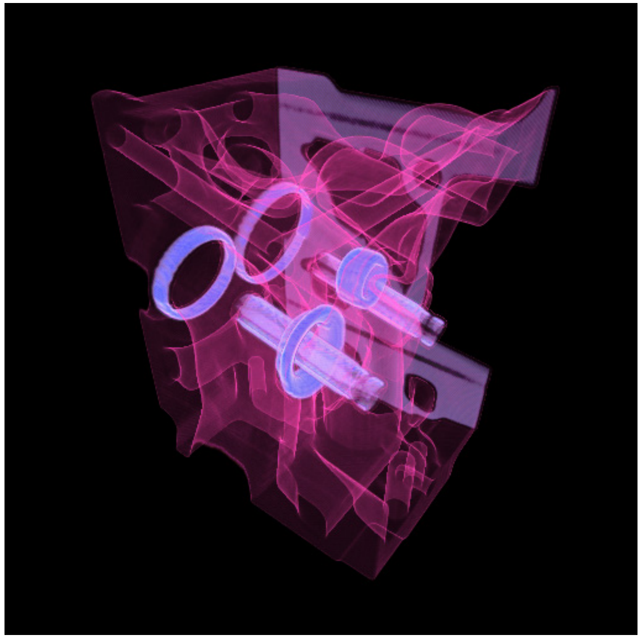
# Where can it be used?

- **Movie industry**


# Where can it be used?

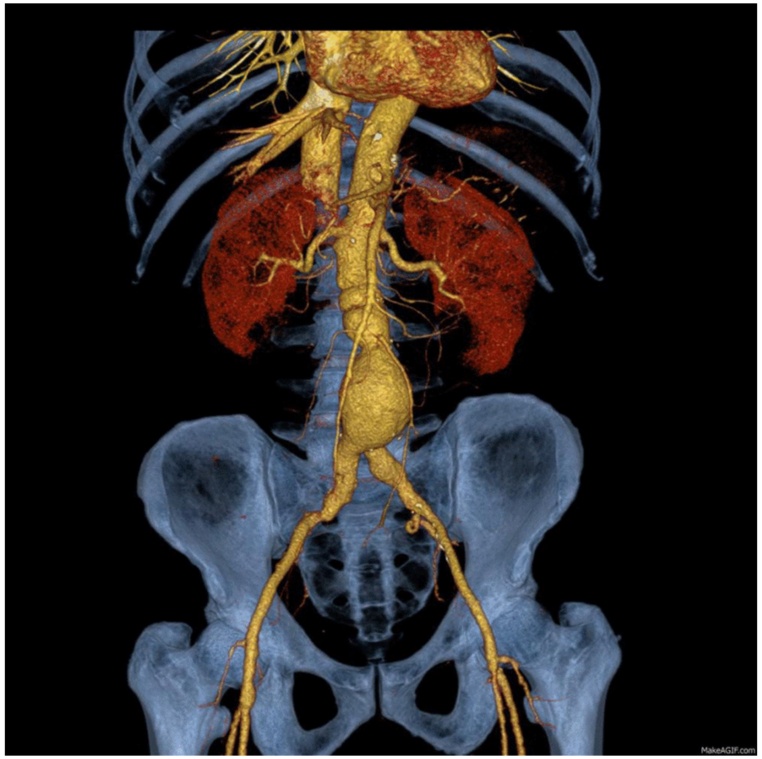
- **Computer aided design (CAD)**

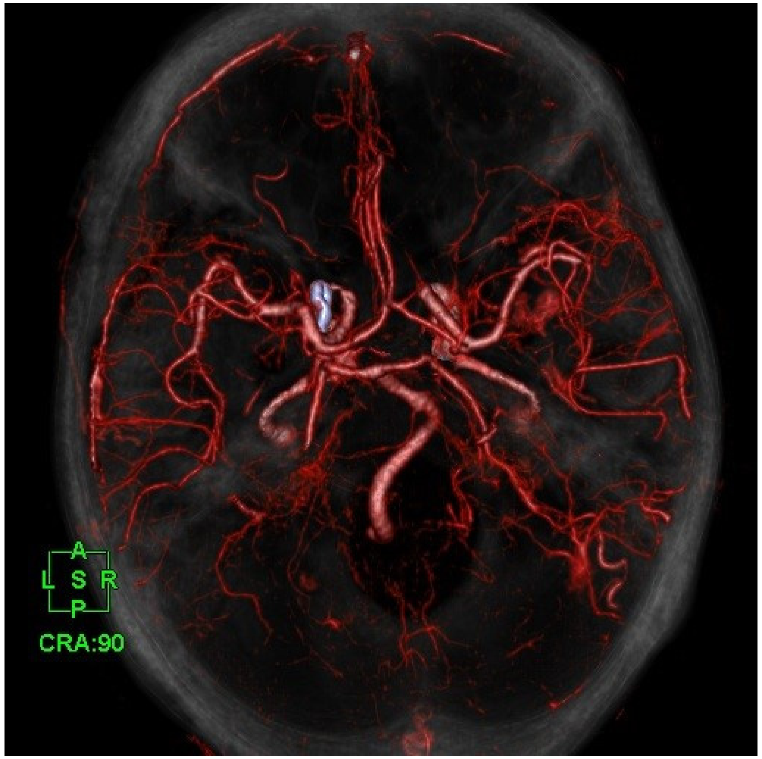


# Where can it be used?

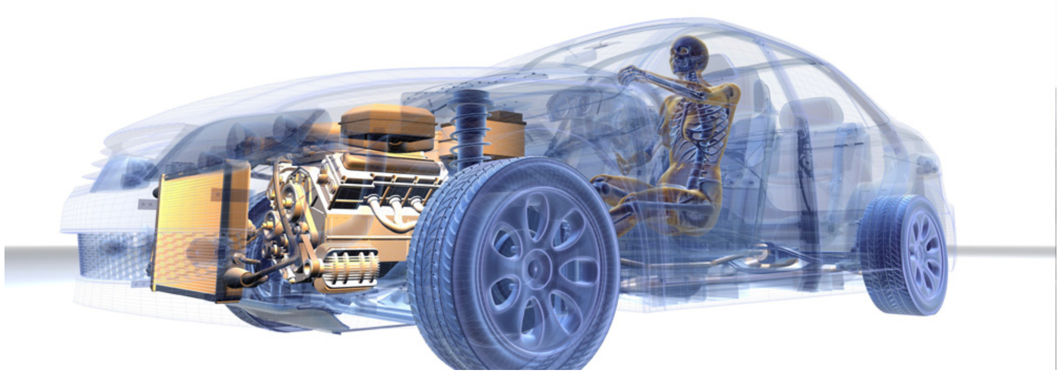
- **Scientific visualization**
  - Medical science
    - CT/MRI scan





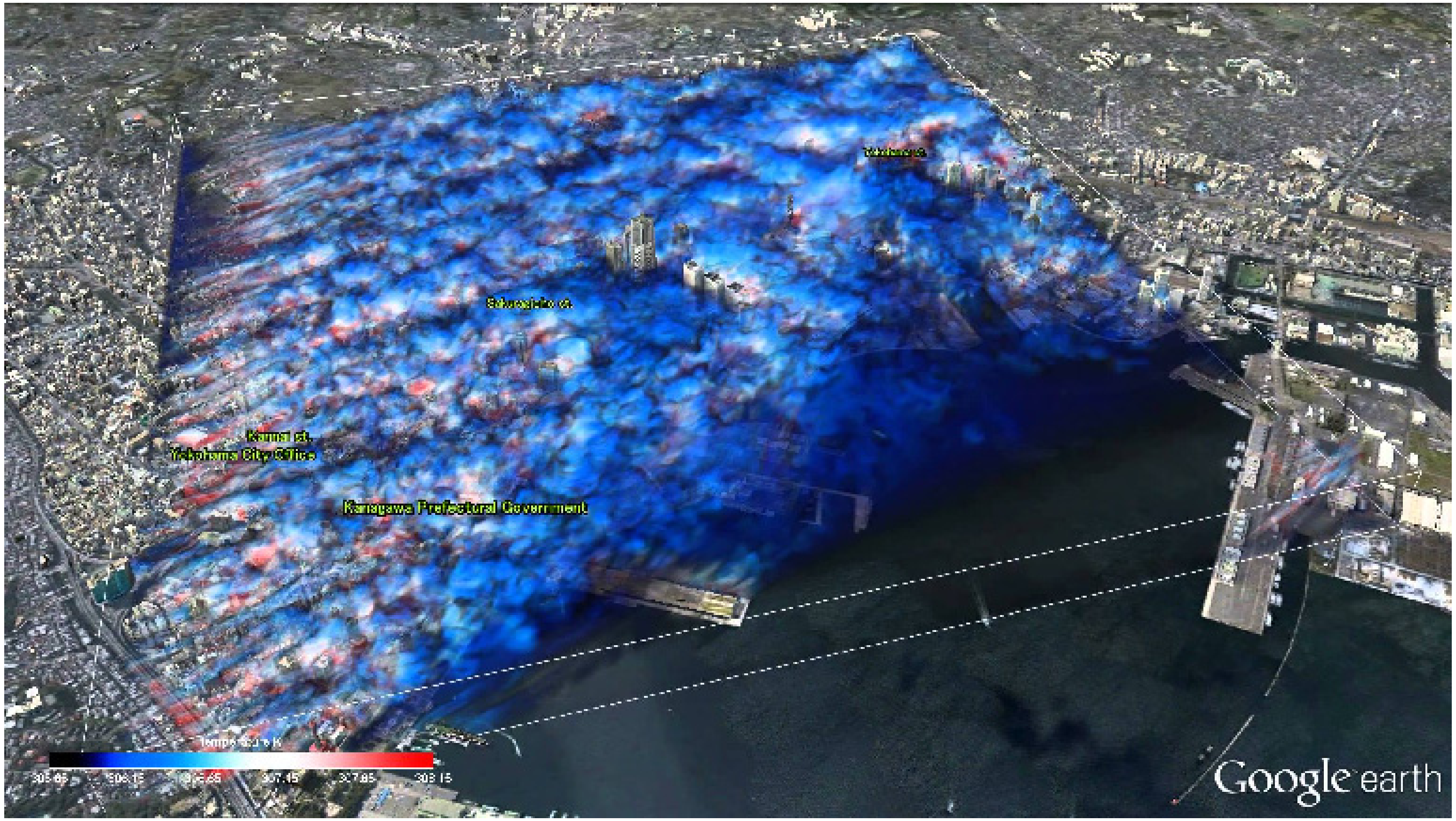
# Where can it be used?

- **Scientific visualization**
  - Mechanical engineering



# Where can it be used?

- **Scientific visualization**
  - Environment engineering



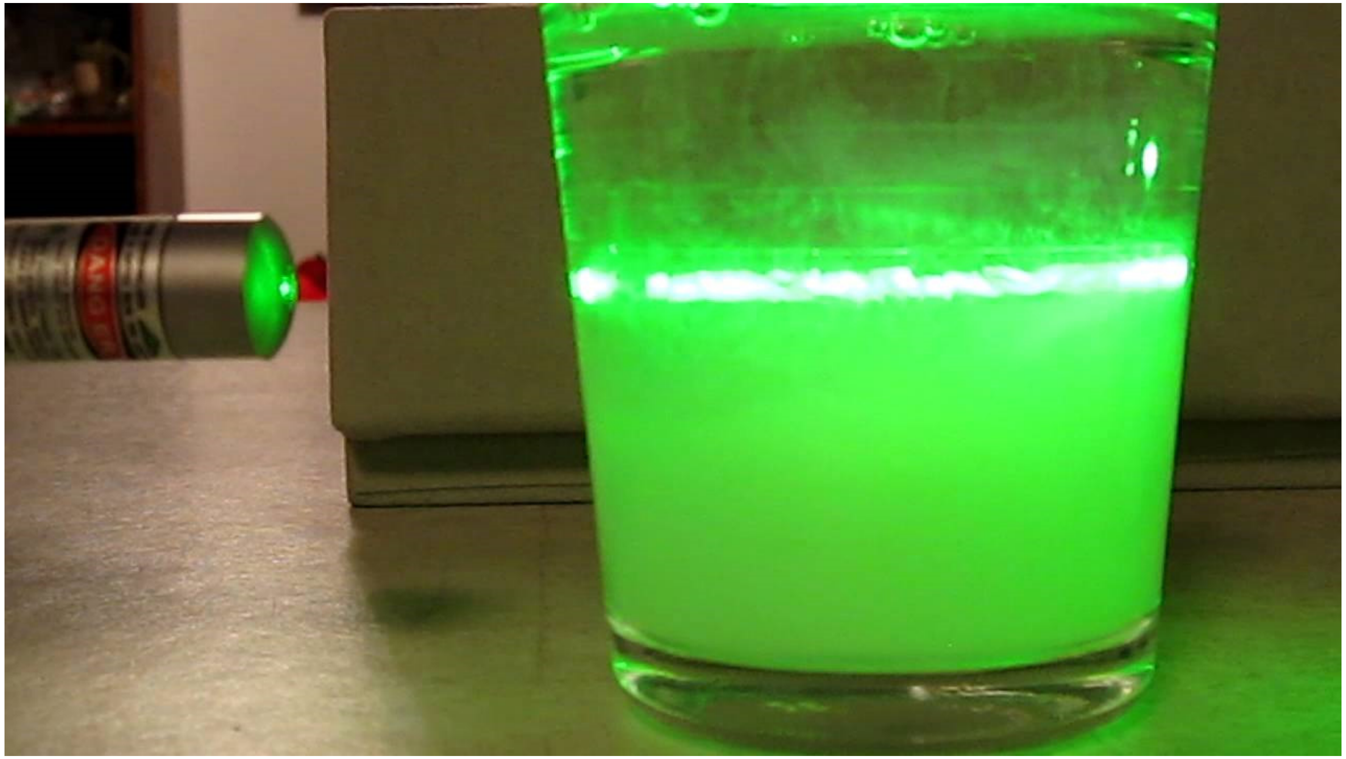
# Where can it be used?

- **Scientific visualization**
  - Astronomical observation

# 1. Light scattering

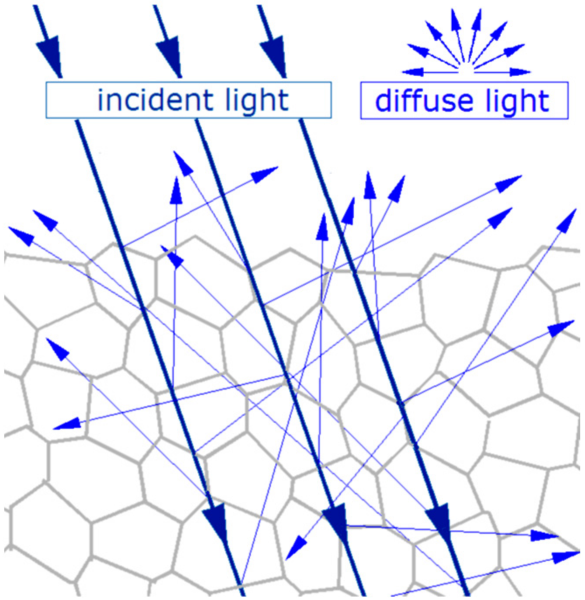
# Participating media

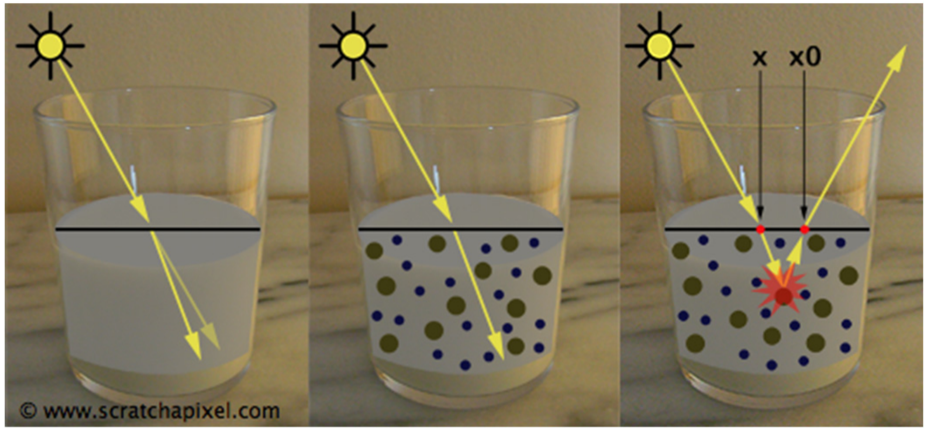
- **What is participating media?**
  - Light can go through the media
  - The media may interact with light throughout the way



# Light scattering

- **A form of scattering**
  - Light in the form of propagating energy is scattered
  - Can be thought of as the deflection of a ray from a straight path
  - By irregularities in the propagation medium (particles)



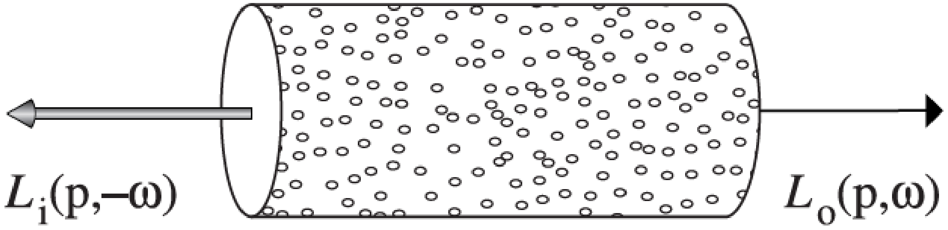



# Volume scattering processes

- **Three main processes**
  - **Absorption**
    - The reduction in radiance due to conversion of light to other form of energy, e.g., heat
  - **Emission**
    - Energy that is added to the environment from luminous particles
  - **Scattering**
    - How light in one direction is scattered to other directions due to collision with particles

# Volume scattering processes

- **Three main processes**
  - **Absorption**
    - Described by absorption cross section $\sigma_a$
    - Probability density that light is absorbed per unit distance
    - Vary with position and direction

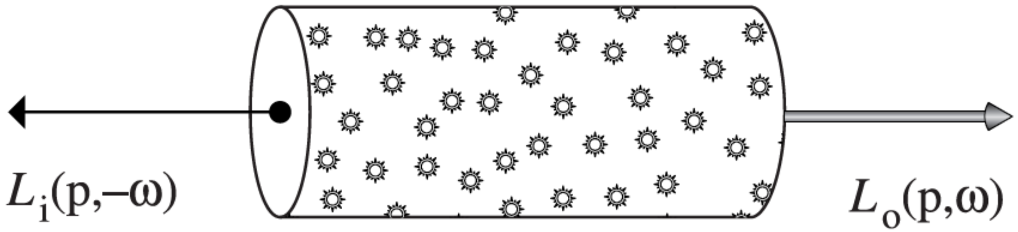



$$L_o(p, \omega) - L_i(p, -\omega) = dL_o(p, \omega) = -\sigma_a(p, \omega)\, L_i(p, -\omega)\, dt$$

$$e^{-\int_0^d \sigma_a(p+t\omega, \omega)\, dt}$$

# Volume scattering processes

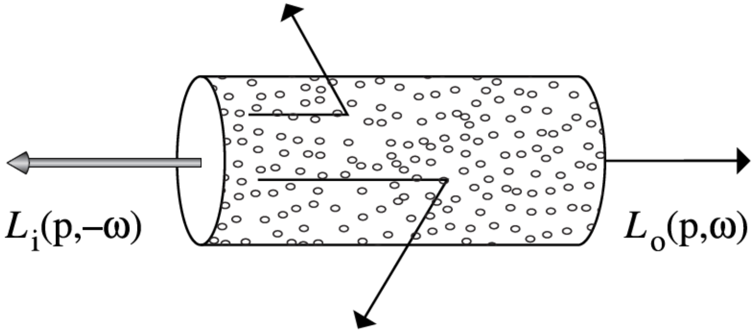
- **Three main processes**
  - **Emission**
    - Energy radiation




$$dL_o(p,\omega) = L_{ve}(p,\omega)\,dt$$

# Volume scattering processes

- **Three main processes**
  - **Out-scattering and extinction**
    - A beam of light collide with particles and scatter in different directions
    - Reduce radiance exiting differential region
    - Probability of out-scattering: $\sigma_s$




$$dL_o(p, \omega) = -\sigma_s(p, \omega)\, L_i(p, -\omega)\, dt$$

# Volume scattering processes

- **Three main processes**
  - **Out-scattering and extinction**
    - Extinction/attenuation: total reduction of radiation

$$\sigma_{\mathrm{t}}(\mathrm{p}, \omega) = \sigma_{\mathrm{a}}(\mathrm{p}, \omega) + \sigma_{\mathrm{s}}(\mathrm{p}, \omega)$$

    - Differential equation describing all attenuation:

$$\frac{\mathrm{d}L_{\mathrm{o}}(\mathrm{p}, \omega)}{\mathrm{d}t} = -\sigma_{\mathrm{t}}(\mathrm{p}, \omega)\, L_{\mathrm{i}}(\mathrm{p}, -\omega)$$

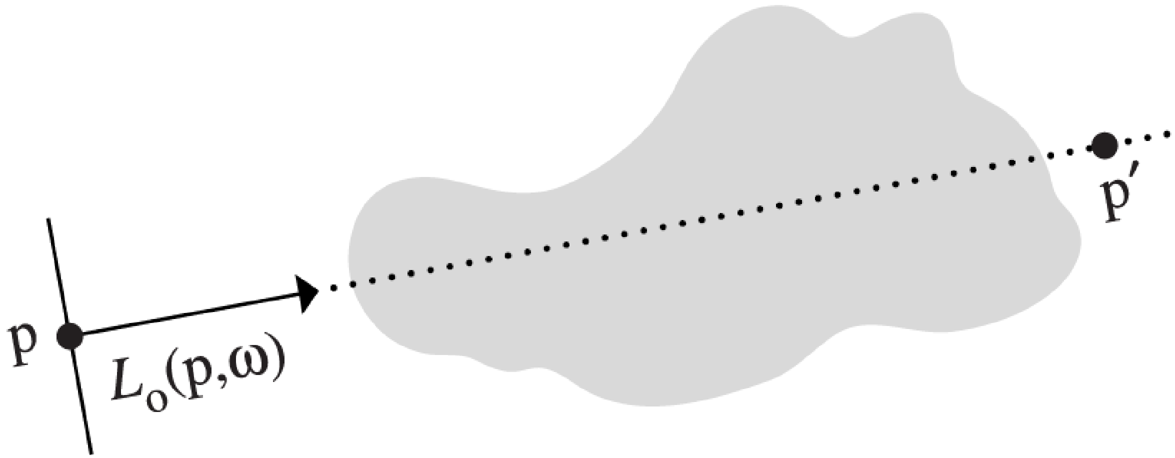
    - Beam transmittance

$$\frac{\mathrm{d}L_{\mathrm{o}}(\mathrm{p}, \omega)}{\mathrm{d}t} = -\sigma_{\mathrm{t}}(\mathrm{p}, \omega)\, L_{\mathrm{i}}(\mathrm{p}, -\omega) \Rightarrow T_r(\mathrm{p} \to \mathrm{p}') = \mathrm{e}^{-\int_0^d \sigma_{\mathrm{t}}(\mathrm{p}+t\omega, \omega)\, \mathrm{d}t}$$

# Volume scattering processes

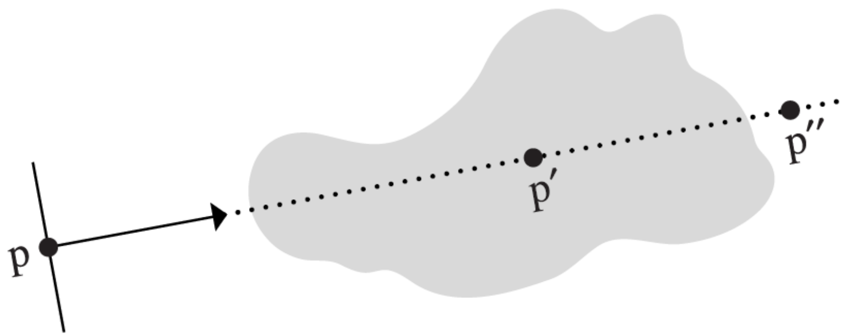
- **Three main processes**
  - **Out-scattering and extinction**
    - Beam transmittance : [0,1]
    - If exitant radiance from a point is $L_o(p,\omega)$, after accounting for extinction:

$$T_r(\mathrm{p} \rightarrow \mathrm{p}')L_o(\mathrm{p}, \omega)$$



# Volume scattering processes

- **Three main processes**
  - **Out-scattering and extinction**
    - Properties of beam transmittance:
      - Self transmittance $T_r(\mathrm{p} \to \mathrm{p}) = 1$
      - Vaccum transmittance $T_r(\mathrm{p} \to \mathrm{p}') = 1$
      - Multiplicative $T_r(\mathrm{p} \to \mathrm{p}'') = T_r(\mathrm{p} \to \mathrm{p}')T_r(\mathrm{p}' \to \mathrm{p}'')$




Incrementally compute
transmittance at many points along a ray

# Volume scattering processes

- **Three main processes**
  - **Out-scattering and extinction**
    - Optical thickness

$$\tau(\mathrm{p} \to \mathrm{p}') = \int_0^d \sigma_{\mathrm{t}}(\mathrm{p} + t\omega, -\omega)\, \mathrm{d}t$$

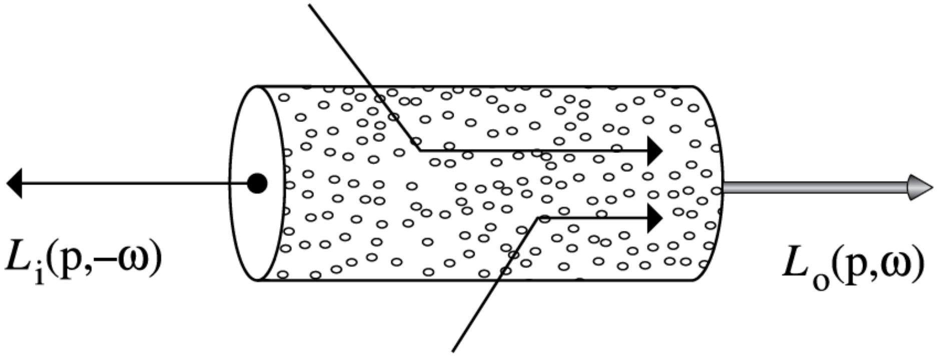
    - Homogeneous media
      - $\sigma_t$ is a constant

$$T_r(\mathrm{p} \to \mathrm{p}') = \mathrm{e}^{-\sigma_{\mathrm{t}} d}$$

# Volume scattering processes

- **Three main processes**
  - **In-scattering**
    - Account for increased radiance due to scattering from other directions
    - Phase function: description of angular distribution of scattered radiation at a point

$$\int_{\mathcal{S}^2} p(\omega \to \omega')\,\mathrm{d}\omega' = 1$$



# Volume scattering processes

- **Three main processes**
  - **In-scattering**
    - Total added radiance per unit distance due to in-scattering is given by the source term S

$$dL_o(p, \omega) = S(p, \omega)\, dt$$

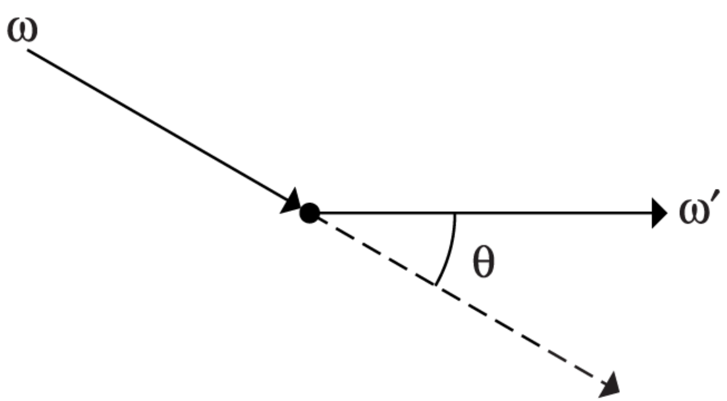
    - The source term accounts for

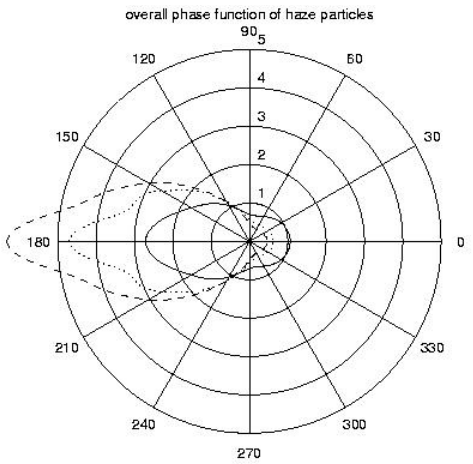
$$S(p, \omega) = L_{ve}(p, \omega) + \sigma_s(p, \omega) \int_{S^2} p(p, -\omega' \to \omega)\, L_i(p, \omega')\, d\omega'$$

# 2. Phase function

# Phase function

- **Definition**
  - Describe the angular distribution of scattered radiation at a point
  - The distribution is defined based on solid angle
- **Isotropic media**
  - 1D phase function
    - Depend only on the angle θ between two directions



# Phase function

- **Isotropic media**
  - Phase functions are usually written as

$$p(\cos\theta)$$

  - Reciprocal property

$$\cos(-\theta) = \cos(\theta)$$

  - Isotropic phase function

$$p_{\text{isotropic}}(\omega \to \omega') = \frac{1}{4\pi}$$

# Phase function

- **Isotropic media, anisotropic phase functions**
  - Different phase function models
    - Rayleigh model
      - Describe scattering from very small particles
      - Particle radius smaller than wavelength of light
      - E.g. air scattering at blue sky and red sunset
    - Mie scattering model
      - More general theory
      - Derived from Maxwell's equations
      - Describe scattering from a wider range of particle sizes
      - E.g. scattering due to water droplets and fog

# Phase function

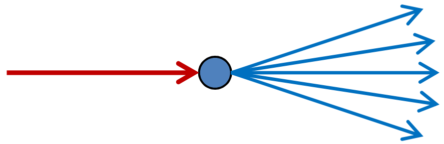
- **Isotropic media, anisotropic phase functions**
  - Model widely used phase functions in graphics
    - Henyey-Greenstein phase function

$$p_{\mathrm{HG}}(\cos\theta) = \frac{1}{4\pi}\frac{1-g^2}{(1+g^2-2g(\cos\theta))^{3/2}}$$

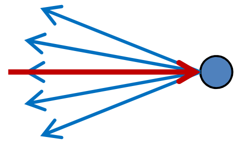
    - A single parameter g:
      - g in (-1,1)
      - Negative value: back scatter; light mostly scattered back
      - Positive value: forward scatter;

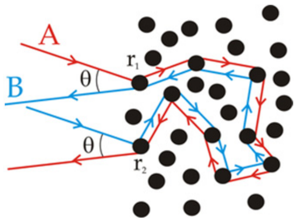
# Phase function

- **Isotropic media, anisotropic phase functions**
  - Forward scattering
    - Light mostly scattered in forward direction



  - Backward scattering
    - Light mostly scattered in backward direction





# Phase function

- **Isotropic media, anisotropic phase function**
  - g has a precise meaning
    - The average value of the product of the phase function being approximated and the cosine of angle between two directions

$$g = \int_{S^2} p(\omega \to \omega')(\omega \cdot \omega')\, \mathrm{d}\omega' = 2\pi \int_0^{\pi} p(\cos\theta)\ \cos\theta\ \sin\theta\ \mathrm{d}\theta$$

    - An isotropic phase function: g=0
  - More complex phase function
    - Weighted sum of modeled phase functions

$$p(\omega \to \omega') = \sum_{i=1}^{n} w_i\ p_i(\omega \to \omega')$$

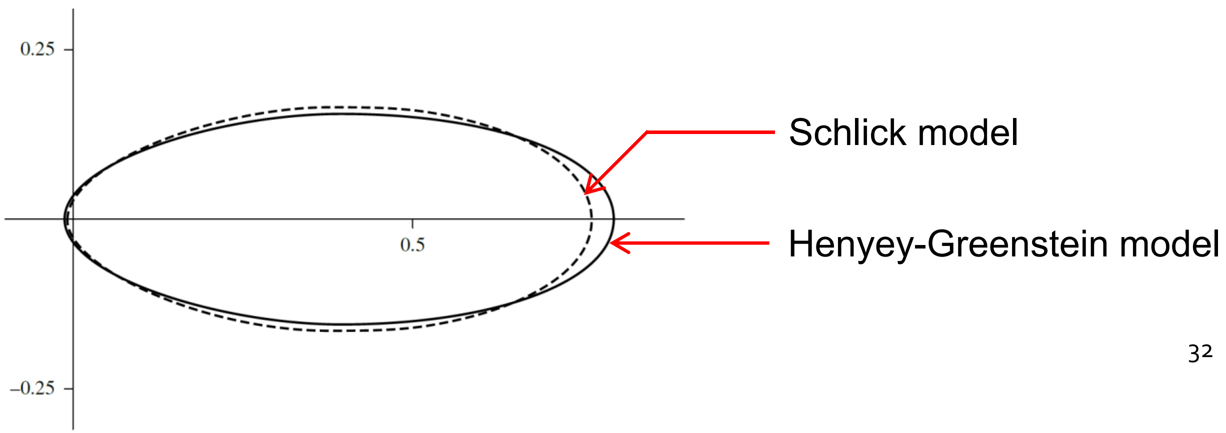
# Phase function

- **Isotropic media, anisotropic phase functions**
  - Efficient approximation to Henyey-Greenstein phase function (Blasi, Le Saëc and Schlick 1993)

$$p_{\text{Schlick}}(\cos\theta) = \frac{1}{4\pi}\frac{1-k^2}{(1-k\cos\theta)^2}$$

  - k and g relations

$$k = 1.55g - .55g^3$$



# 3. Sub-surface scattering

# BSSRDF

- **Subsurface light transport**
  - Light enters the material at one point and may exit quite far away

subsurface只有一层比较浅的layer
(一种特殊的volumn rendering)

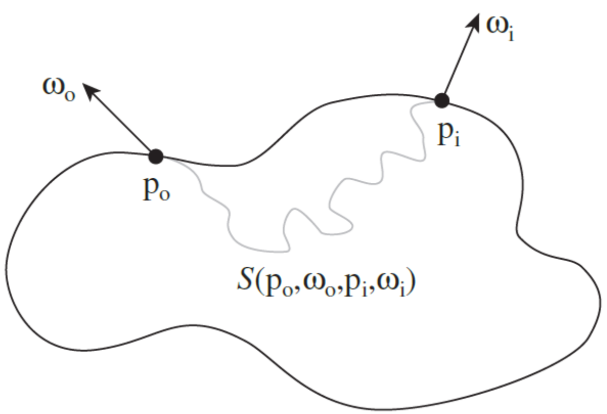
- **Bidirectional scattering-surface reflectance distribution function (BSSRDF)**
  - A formalism for describing scattering processes
  - Describe the ratio of exitant and incident differential radiance at different positions and solid angles

$$S(\mathrm{p_o}, \omega_\mathrm{o}, \mathrm{p_i}, \omega_\mathrm{i}) = \frac{\mathrm{d}L_\mathrm{o}(\mathrm{p_o}, \omega_\mathrm{o})}{\mathrm{d}E(\mathrm{p_i}, \omega_\mathrm{i})}$$

# Subsurface scattering

- **Accurately rendering translucent surfaces with subsurface scattering**
  - Integrating over area: points on surfaces
  - Integrating over incident directions
  - Evaluating the integral

$$L_o(p_o, \omega_o) = \int_A \int_{\mathcal{H}^2(\mathbf{n})} S(p_o, \omega_o, p_i, \omega_i)\, L_i(p_i, \omega_i) |\cos\theta_i| \, d\omega_i \, dA$$



# Subsurface scattering

- **How to evaluate such integral?**
  - **Monte-Carlo integration**
  - **Not efficient for high albedo material**
    - Essentially all of the light is scattered at each interaction
    - Almost none is absorbed
    - Hundreds or thousands of scatter must be considered
  - **How to model?**
    - Diffusion approximation
    - Solution found by composing closed-form solutions for point light sources
    - Assumptions
      - Homogeneous scattering
      - Implicitly assume the medium is semi-infinite

# 4. Equation of transfer

# Equation of transfer

- **Fundamental equation**
  - Govern the behavior of light in a medium that absorbs, emits and scatters radiation
  - Light transport equation is only a special case
  - Integral-differential form of the equation of transfer

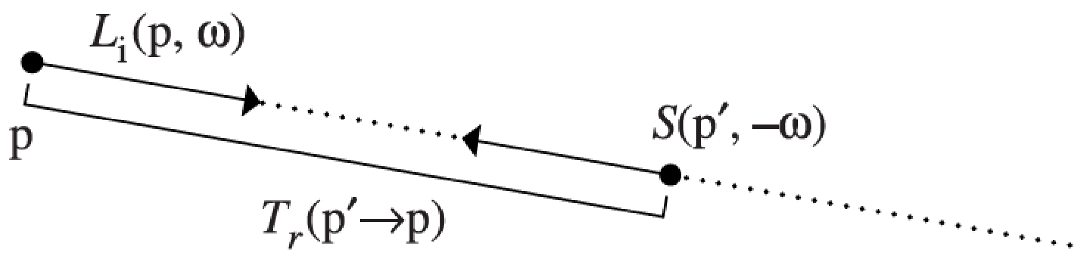
$$\frac{\partial}{\partial t} L_o(p, \omega) = -\sigma_t(p, \omega) L_i(p, -\omega) + S(p, \omega)$$

S光源(直接/间接)

# Equation of transfer

- **Fundamental equation**
  - No surfaces in the scene with an infinite ray length

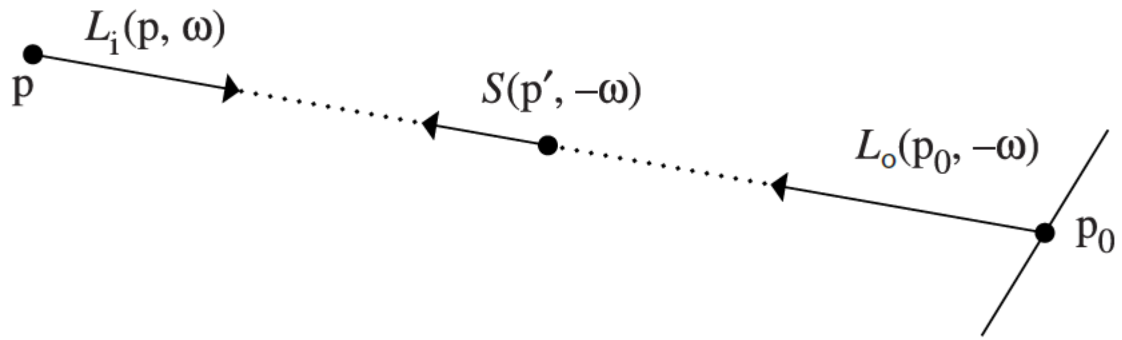
$$L_{\mathrm{i}}(\mathrm{p}, \omega) = \int_0^{\infty} T_r(\mathrm{p}' \to \mathrm{p}) S(\mathrm{p}', -\omega)\, \mathrm{d}t$$



# Equation of transfer

- **Fundamental equation**
  - More generally, with reflecting and/or emitting surfaces
    - Rays have finite length

$$L_i(\mathrm{p}, \omega) = T_r(\mathrm{p}_0 \to \mathrm{p}) L_o(\mathrm{p}_0, -\omega) + \int_0^t T_r(\mathrm{p}' \to \mathrm{p}) S(\mathrm{p}', -\omega) \mathrm{d}t'$$



# Emission-only integrator

- **Simplest volume integrator**
  - Only account for emission and attenuation

$$L_{\mathrm{i}}(\mathrm{p}, \omega) = T_r(\mathrm{p}_0 \to \mathrm{p}) L_{\mathrm{o}}(\mathrm{p}_0, -\omega) + \int_0^t T_r(\mathrm{p}' \to \mathrm{p}) L_{\mathrm{ve}}(\mathrm{p}', -\omega)\, \mathrm{d}t$$

  - An estimation of integral

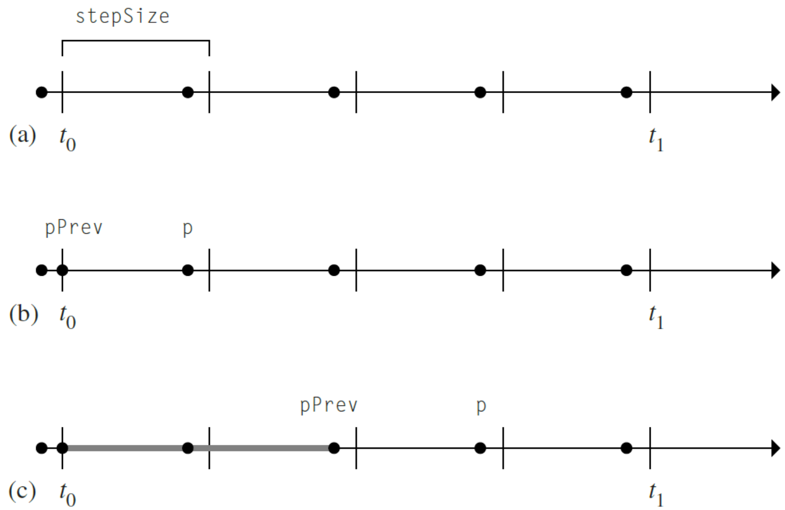
$$\int_{t_0}^{t_1} T_r(\mathrm{p}' \to \mathrm{p}) L_{\mathrm{ve}}(\mathrm{p}', -\omega)\, \mathrm{d}t'$$

$$\Downarrow$$

$$\frac{1}{N} \sum_i \frac{T_r(\mathrm{p}_i \to \mathrm{p}) L_{\mathrm{ve}}(\mathrm{p}_i, -\omega)}{p(\mathrm{p}_i)} = \frac{t_1 - t_0}{N} \sum_i T_r(\mathrm{p}_i \to \mathrm{p}) L_{\mathrm{ve}}(\mathrm{p}_i, -\omega)$$

# Emission-only integrator

- **Equal step-size sampling of the ray**
  – Using a fixed time-step, from near to far




  – Incrementally compute $T_r$

$$T_r(\mathrm{p}_i \to \mathrm{p}) = T_r(\mathrm{p}_i \to \mathrm{p}_{\mathrm{i}-1}) T_r(\mathrm{p}_{i-1} \to \mathrm{p})$$

# Single v.s. multiple scattering

- **Single scattering**
  - Light scattered once before reaching a position
  - Similar like a direct lighting

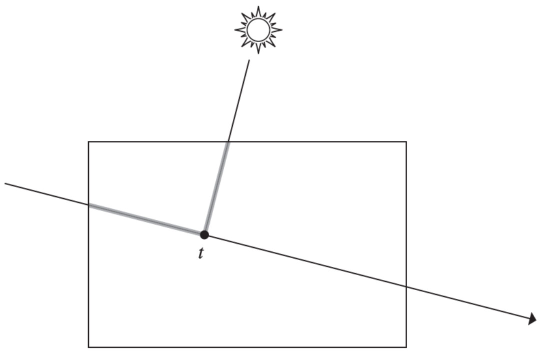
- **Multiple scattering**
  - Light scattered multiple times before reaching a position
  - Similar like a direct lighting with inter-reflection

# Single scattering integrator

- **Ignore multiple scattering**
  - Incident volume light only direct light source

$$\int_0^t T_r(\mathrm{p}' \to \mathrm{p}) \left( L_{\mathrm{ve}}(\mathrm{p}', -\omega) + \sigma_{\mathrm{s}}(\mathrm{p}', \omega) \int_{\mathcal{S}^2} p(\mathrm{p}', -\omega' \to -\omega) \boxed{L_{\mathrm{d}}(\mathrm{p}', \omega')} \mathrm{d}\omega' \right) \mathrm{d}t'$$

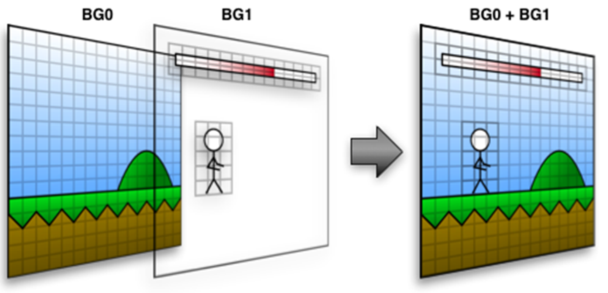
  - For scattering term
    - Usually ignore light source attenuation
    - Important sampling with Monte-Carlo integration

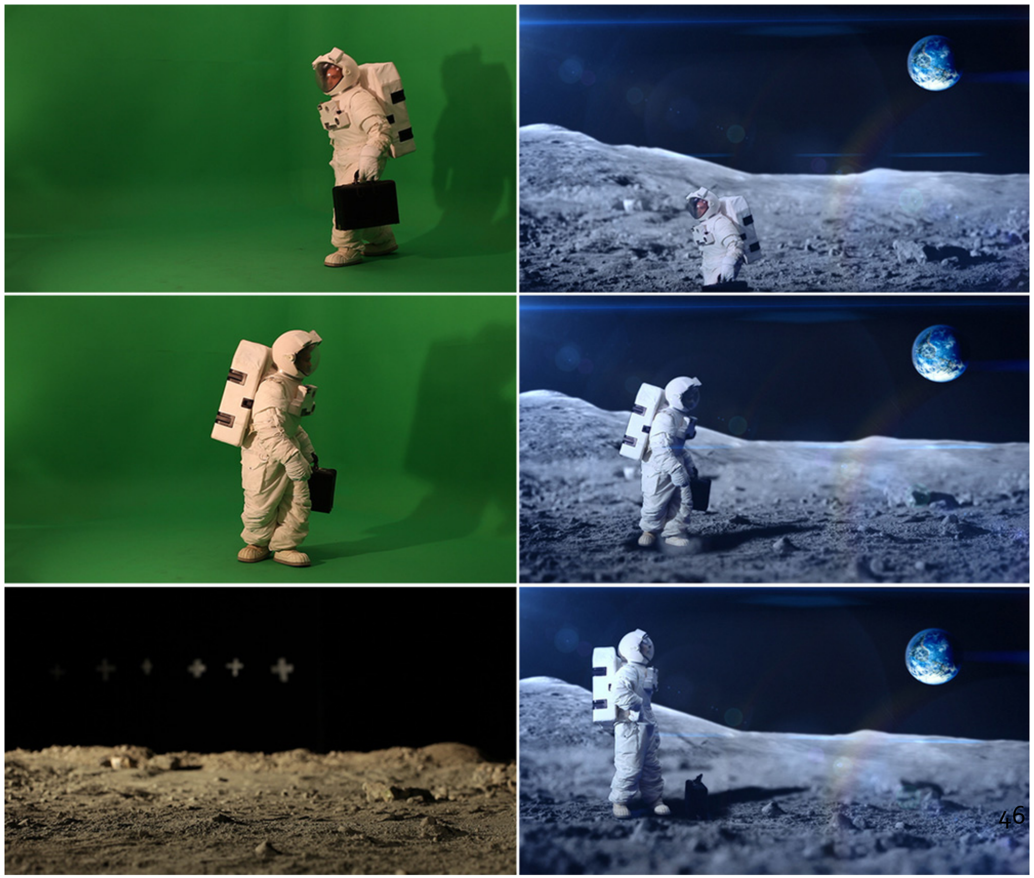


# 5. Volume rendering

# How to render volume data?

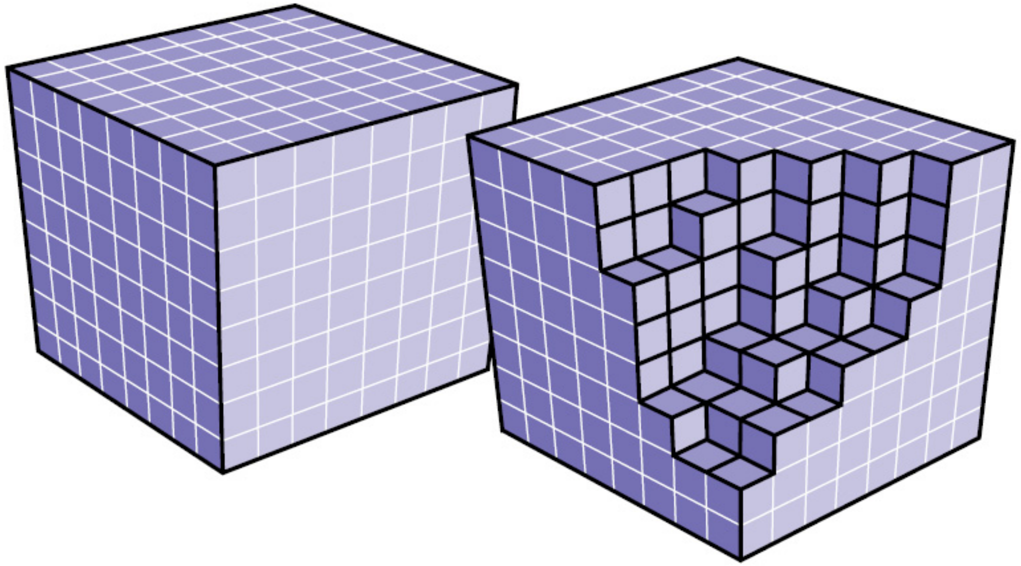
- **Layering**
  - Image composition



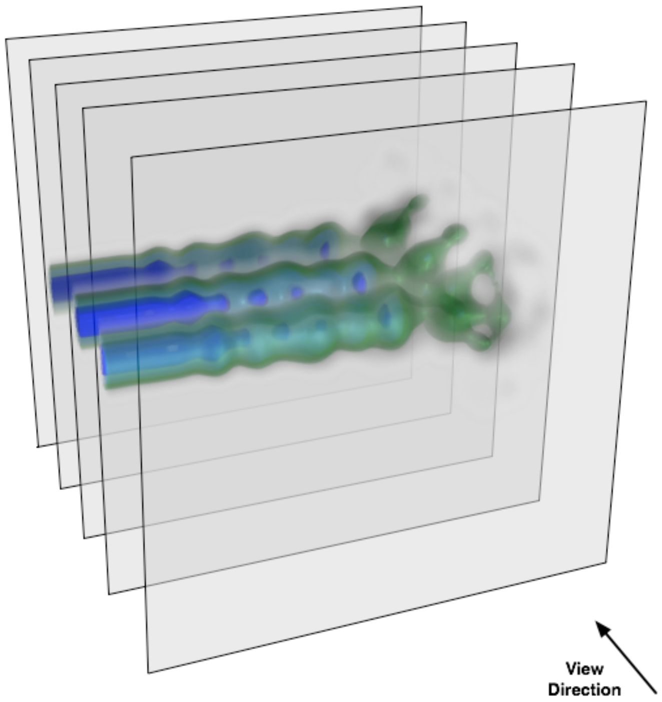
# How to render volume data?

- **Volume data**
  - Usually a uniform grid storing volume samples



# How to render volume data?

- **Layer(slice)-based rendering**
  - Alpha blending slice by slice
  - Slices are placed orthogonal to view direction



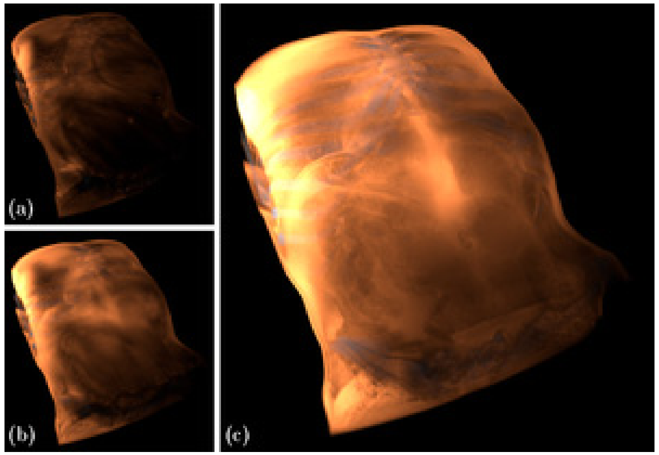
# How to render volume data?

- **Point sprite**
  - Point samples to represent a volume function
  - Finite size point (usually textured) with transparency

Point sprite texture

# Physically-based volume graphics

- **Rendering based on volumetric light effects**
  - Realistic
  - Artificial (artistic, visualization)



# Volume-rendering integral

- **Volume rendering equation**
  - Radiance addition and reduction along a ray
    - Absorption-emission model (no scattering)

$$\underbrace{\frac{\mathrm{d}I(s)}{\mathrm{d}s} = \underbrace{-\kappa(s)I(s)}_{\text{Absorption}} + q(s)}_{} \longleftarrow \text{Emission}$$

  - Integrating along the direction of light

$$I(D) = I_0\, e^{-\int\limits_{s_0}^{D} \kappa(t)\,\mathrm{d}t} + \int\limits_{s_0}^{D} q(s)\, e^{-\int\limits_{s}^{D} \kappa(t)\,\mathrm{d}t}\,\mathrm{d}s$$

# Volume-rendering integral

- **Volume rendering equation**
  - Optical depth

$$I(D) = I_0 \, e^{-\int_{s_0}^{D} \kappa(t)\,\mathrm{d}t} + \int_{s_0}^{D} q(s)\, e^{-\int_{s}^{D} \kappa(t)\,\mathrm{d}t}\,\mathrm{d}s$$

$$\tau(s_1, s_2) = \int_{s_1}^{s_2} \kappa(t)\,\mathrm{d}t$$

    - A measure for how long light may travel before it is absorbed
    - Small values indicate transparent media, vice versa

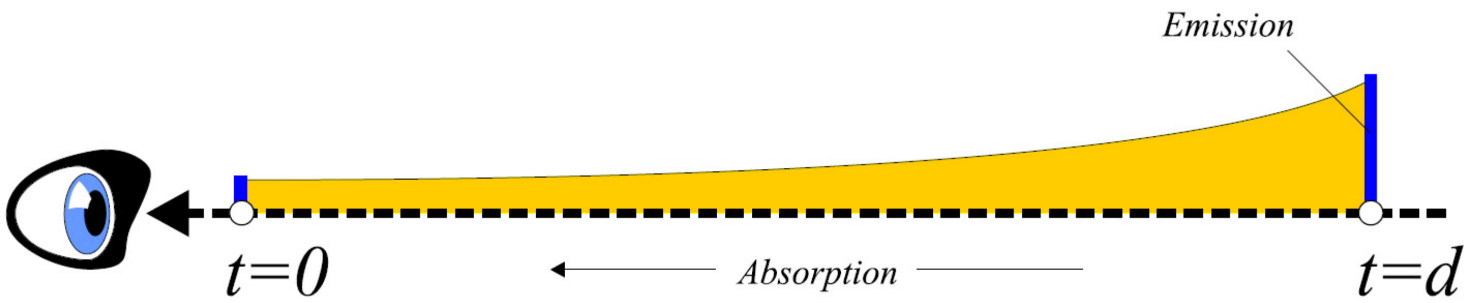
  - Transparency (between $s_1$ and $s_2$)

$$T(s_1, s_2) = e^{-\tau(s_1, s_2)} = e^{-\int_{s_1}^{s_2} \kappa(t)\,\mathrm{d}t}$$

# Volume-rendering integral

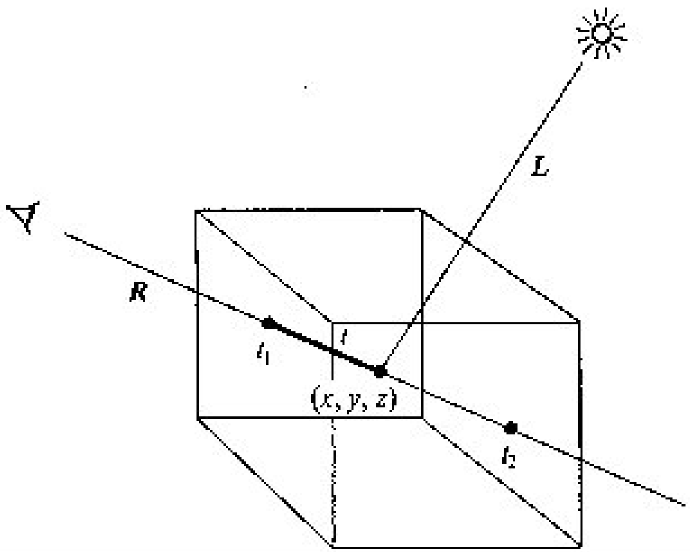
- **Volume rendering equation**
  - Different form

$$I(D) = I_0\, T(s_0, D) + \int_{s_0}^{D} q(s) T(s, D)\, \mathrm{d}s$$



# Local illumination for volume rendering

- **Volumetric lighting**
  - Simple setting
    - Assume no impeding
    - The external illumination directly reaches a point in the volume from an outside light source




A ray R cast into a scalar function of three spatial variables

# Local illumination for volume rendering

- **Volumetric lighting**
  - Local illumination
    - Surface normal estimation
      - Gradient of volumetric data
      - Normal of an implicit isosurface
    - Extended source term

$$q_{\text{extended}}(\mathbf{x}, \omega) = q_{\text{emission}}(\mathbf{x}, \omega) + q_{\text{illum}}(\mathbf{x}, \omega)$$

$$I(D) = I_0\, T(s_0, D) + \int_{s_0}^{D} q(s) T(s, D)\, \mathrm{d}s$$

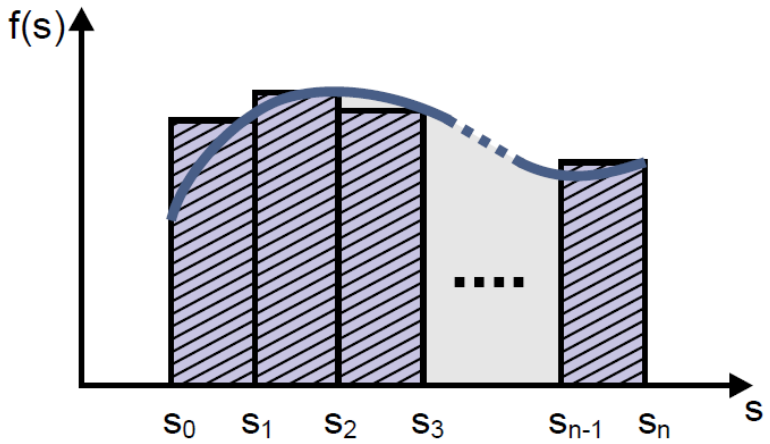
# 6. Numerical scheme for volume rendering

# Discretization

- **Main goal of volume rendering**
  - Compute the volume-rendering integral

$$I(D) = I_0 T(s_0, D) + \int_{s_0}^{D} q(s) T(s, D)\, \mathrm{d}s$$

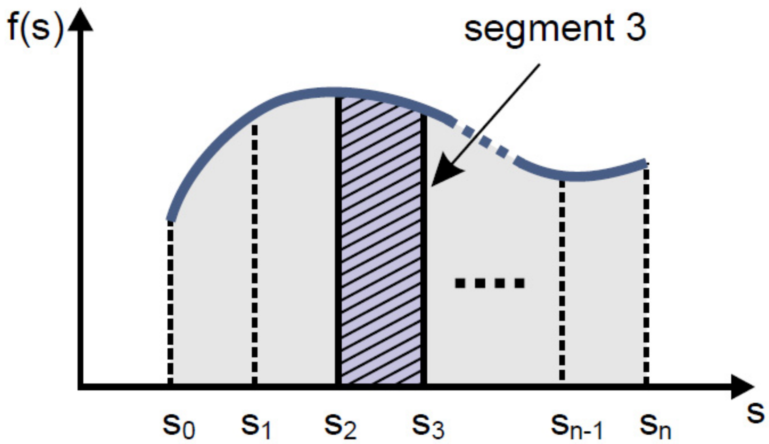
  - Usually, the integral cannot be evaluated analytically
  - Numerical integral



# Discretization

- **Splitting**
  - Split the integration domain into n subsequent intervals: $s_0 < s_1 < \cdots < s_{n-1} < s_n$
  - Consider light transport within the i-th interval $[s_{i-1}, s_i]$

$$I(s_i) = I(s_{i-1})T(s_{i-1}, s_i) + \int_{s_{i-1}}^{s_i} q(s)T(s, s_i)\,\mathrm{d}s$$



# Discretization

- **Iterative evaluation**
  - Quantity definition $$I(s_i) = I(s_{i-1})T(s_{i-1}, s_i) + \int_{s_{i-1}}^{s_i} q(s)T(s, s_i)\,\mathrm{d}s$$

$$T_i = T(s_{i-1}, s_i)\,, \qquad c_i = \int_{s_{i-1}}^{s_i} q(s)T(s, s_i)\,\mathrm{d}s$$

$$\Downarrow$$

$$I(D) = I(s_n) = I(s_{n-1})T_n + c_n = \left(I(s_{n-2})T_{n-1} + c_{n-1}\right)T_n + c_n = \ldots$$

$$\Downarrow$$

$$I(D) = \sum_{i=0}^{n} c_i \prod_{j=i+1}^{n} T_j\,, \quad \text{with } c_0 = I(s_0)$$

# Compositing schemes

- **Definition from a Riemann sum**

透明度
不透明度

  - Transparency $T_i \approx e^{-\kappa(s_i)\Delta x}$
  - Opacity $\alpha_i = 1 - T_i$
  - Color $c_i \approx q(s_i)\Delta x$

- **Front-to-back composition**
  - Viewing rays are traversed from eye into the volume

$$\hat{C}_i = \hat{C}_{i+1} + \hat{T}_{i+1} C_i$$
$$\hat{T}_i = \hat{T}_{i+1}(1 - \alpha_i)$$

Alpha blending $\longrightarrow$

$$C_{\text{dst}} \leftarrow C_{\text{dst}} + (1 - \alpha_{\text{dst}}) C_{\text{src}}$$
$$\alpha_{\text{dst}} \leftarrow \alpha_{\text{dst}} + (1 - \alpha_{\text{dst}}) \alpha_{\text{src}}$$

# Compositing schemes

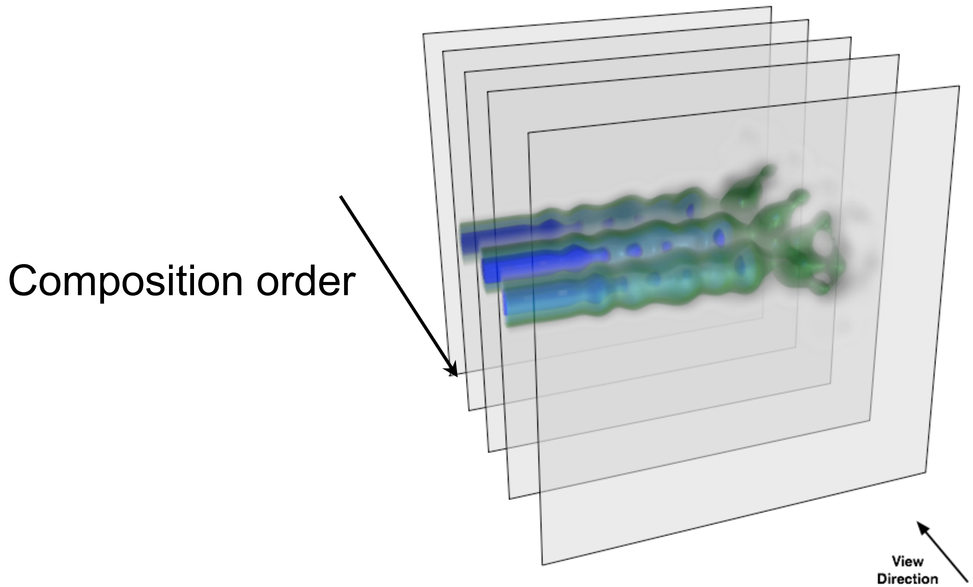
- **Back-to-from composition**

$$\hat{C}_i = \hat{C}_{i-1}(1-\alpha_i) + C_i$$
$$\hat{T}_i = \hat{T}_{i-1}(1-\alpha_i)$$

Alpha blending

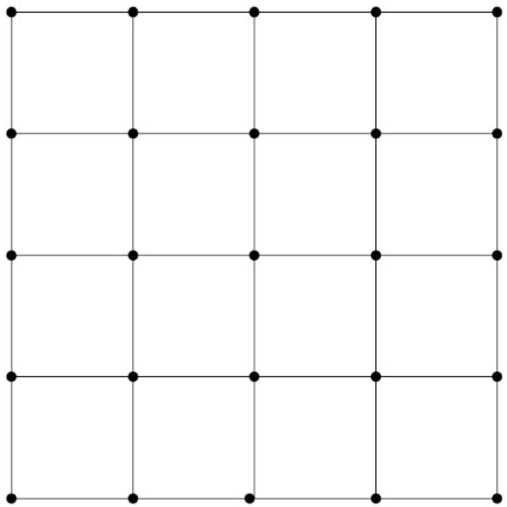
$$C_{\text{dst}} \leftarrow (1-\alpha_{\text{src}})C_{\text{dst}} + C_{\text{src}}$$

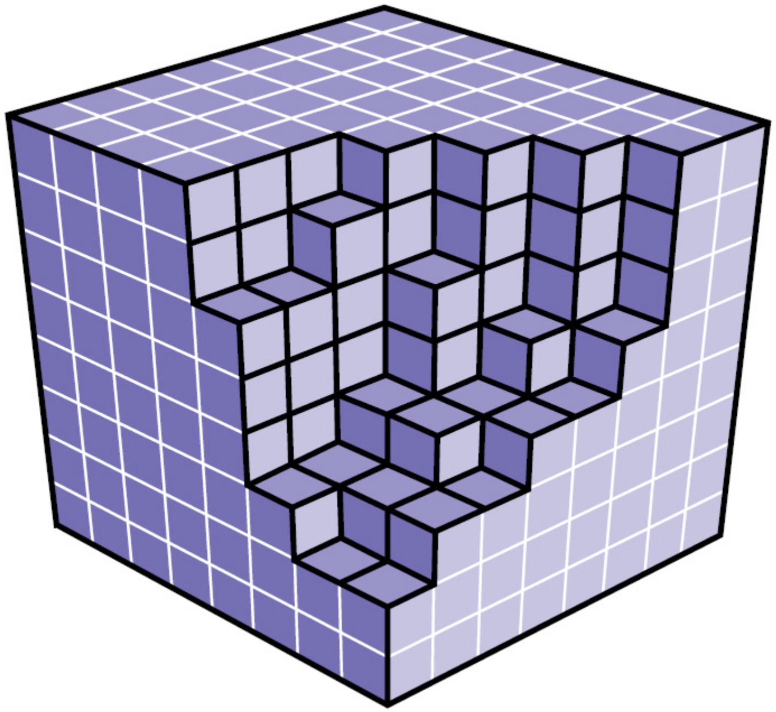
- **Slice/point-sprite-based rendering**
  - Back-to-front composition order



# Classification of grids

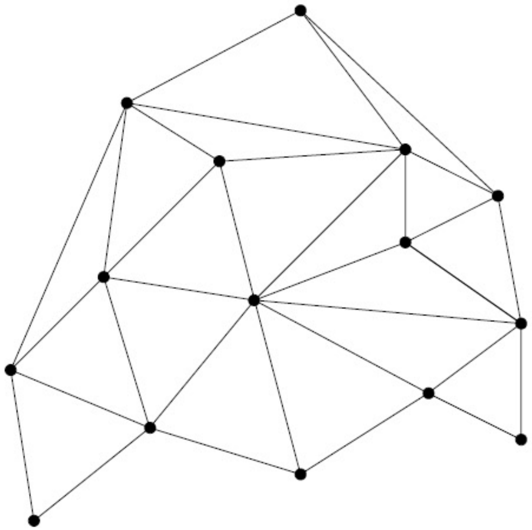
- **Uniform grid**
  - From pixel to voxel

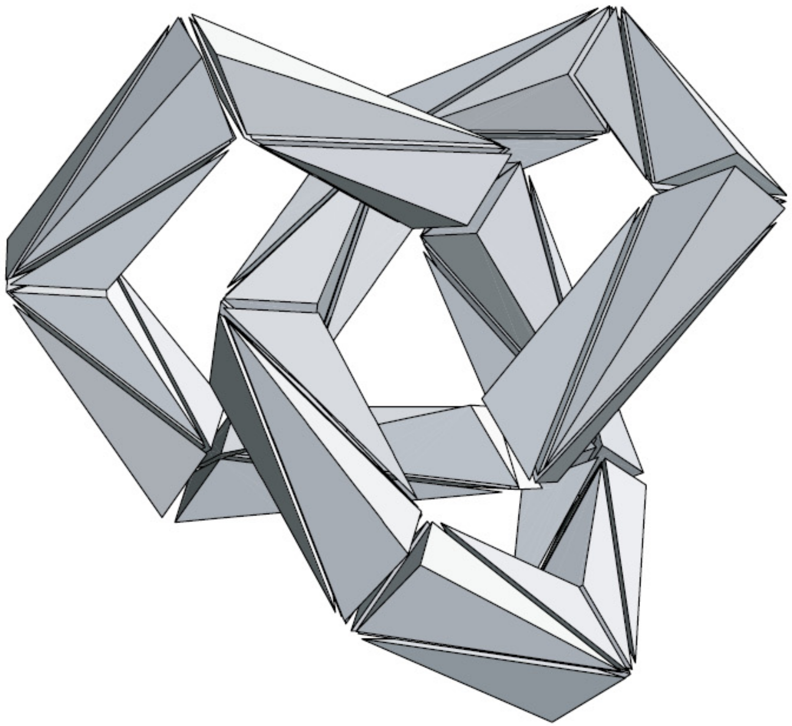




# Classification of grids

- **Simplical grids**
  - Irregular shapes and arrangement



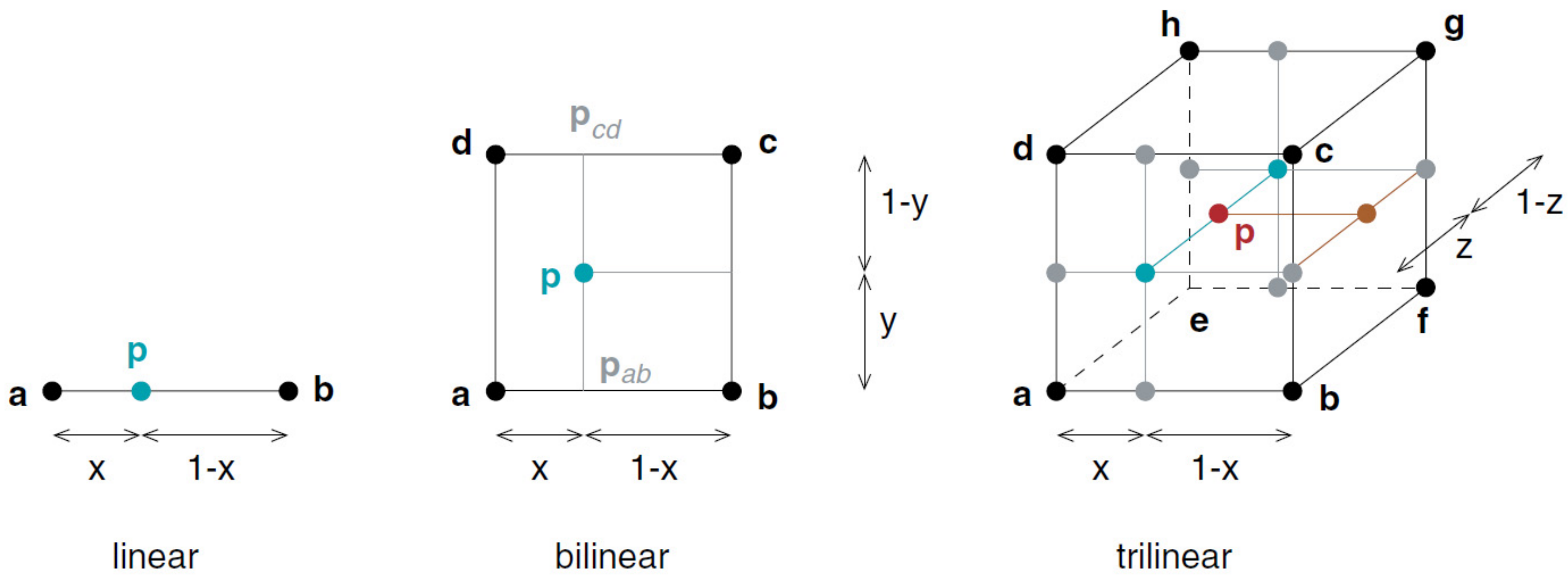


# Data sources and volume acquisition

- **Computerized tomography (CT)**
  - X-ray scan
- **Magnetic resonance imaging (MRI)**
  - Rely on nuclear magnetic resonance
  - Identify different materials in a 3D spatial context
- **Volumetric simulation results**
  - Computational fluid dynamics (CFD) data
  - Astronomical simulations
- **Graphical modeling**

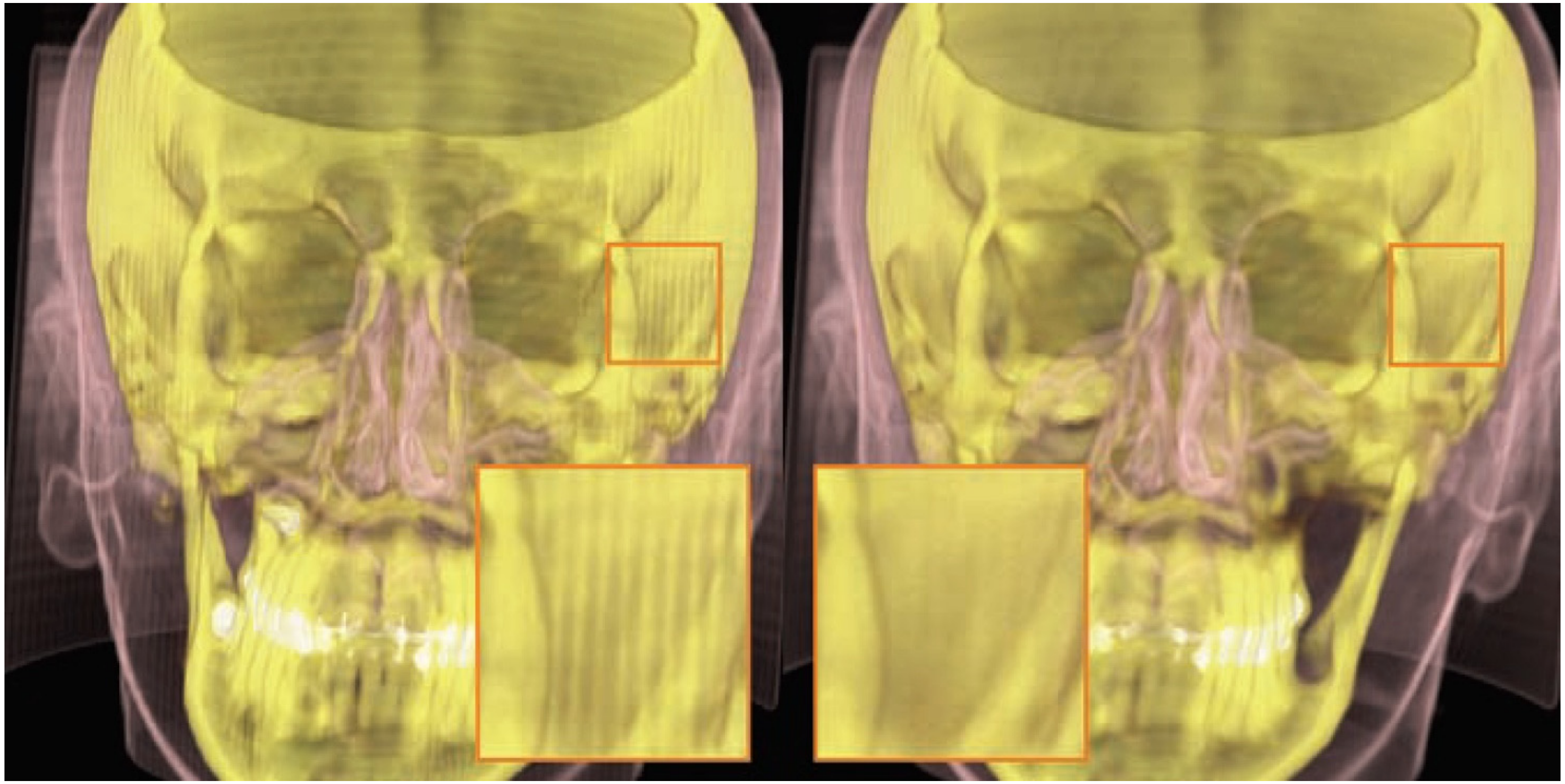
# Reconstruction

- **Reconstruction needed when sampling the volume data**
  – Linear interpolation
  – Cubic-spline interpolation



# Reconstruction

- **Aliasing for different interpolations**



# Components of the volume-rendering pipeline

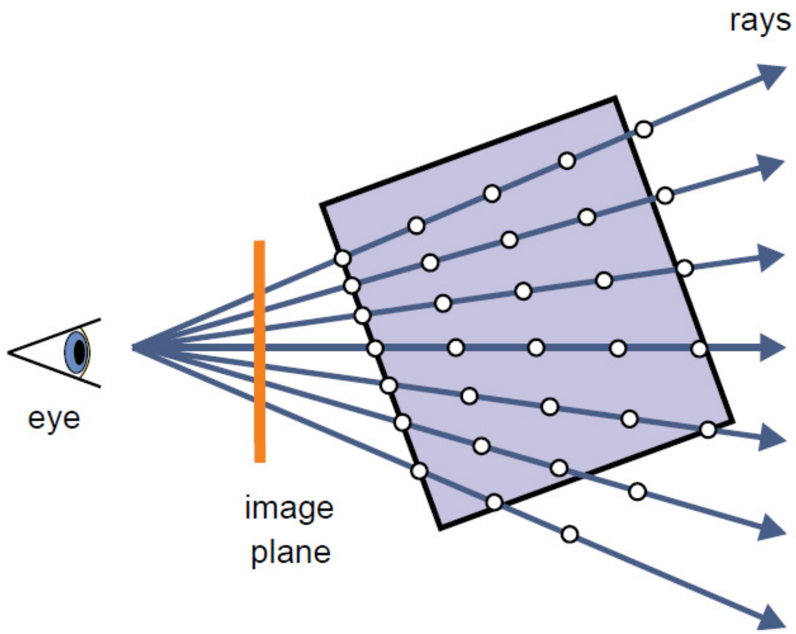
- **Data traversal**
  - Sampling positions are chosen throughout the volume
- **Interpolation**
  - Sampling positions are often different from grid points
  - A continuous 3D field needs to be reconstructed
- **Gradient computation**
  - The gradient of the scalar field is often used to compute local illumination

# Components of the volume-rendering pipeline

- **Classification**
  - Map data properties to optical properties for volume-rendering integral
  - Based on transfer functions
- **Shading and illumination**
  - Adding an illumination term to the emissive source term
- **Compositing**
  - The basis for the iterative computation
  - Front-to-back iteration is used when tracing from the eye point

# Overview of rendering methods

- **Ray casting**
  - The most popular image-order method for volume rendering
  - Directly evaluate volume-rendering integral along rays



# Overview of rendering methods

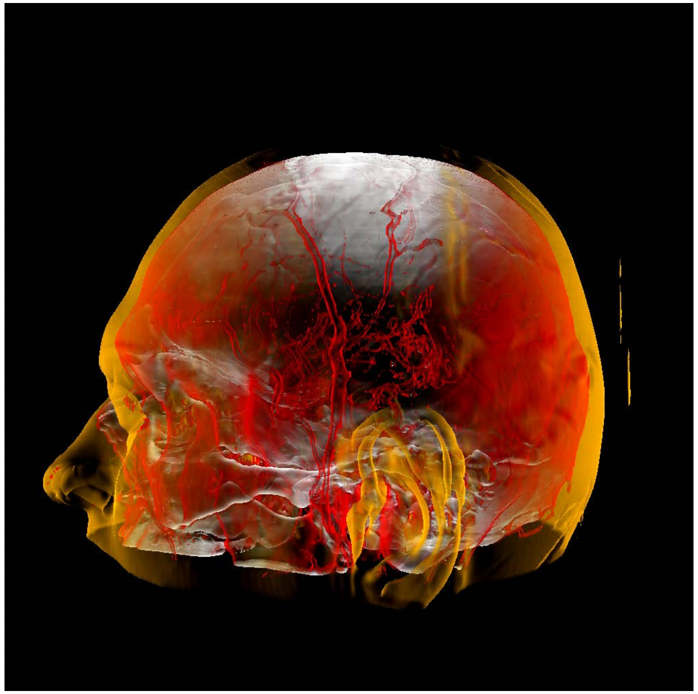
- **Texture slicing**
  - 2D slices located in 3D object space are used to sample the volume
  - Projected onto the image plane and combined according to the compositing scheme
  - Directly supported by graphics hardware
  - 3D texture
  - Restriction to uniform grids, aliasing

# Transfer function

- **Optical properties for volume data**
  - Emission, absorption, scattering
- **Given any scalar volume data**
  - Determine virtual optical properties
  - Reflect certain property when rendered
  - E.g., surface represented by volume data (volume modeling)
  - Transfer function: a mapping from arbitrary scalar volume data to optical properties

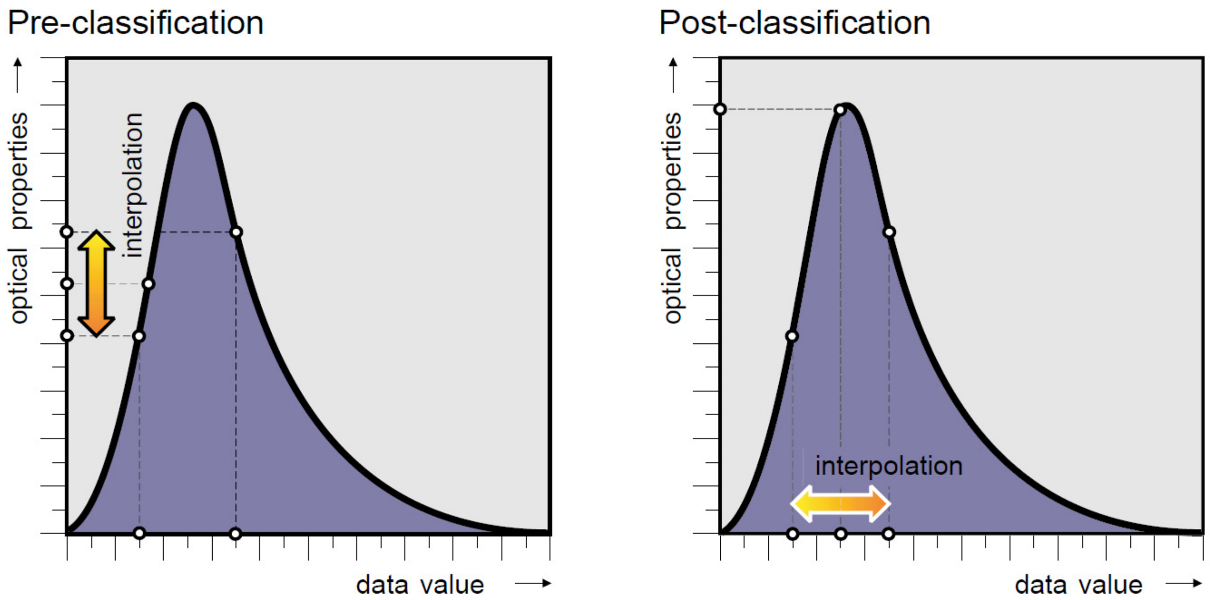
# Classification

- **Definition**
  - The process of finding an appropriate transfer function
  - Essentially a pattern-recognition problem
    - Different patterns found in raw data are assigned to specific categories, or *classes*



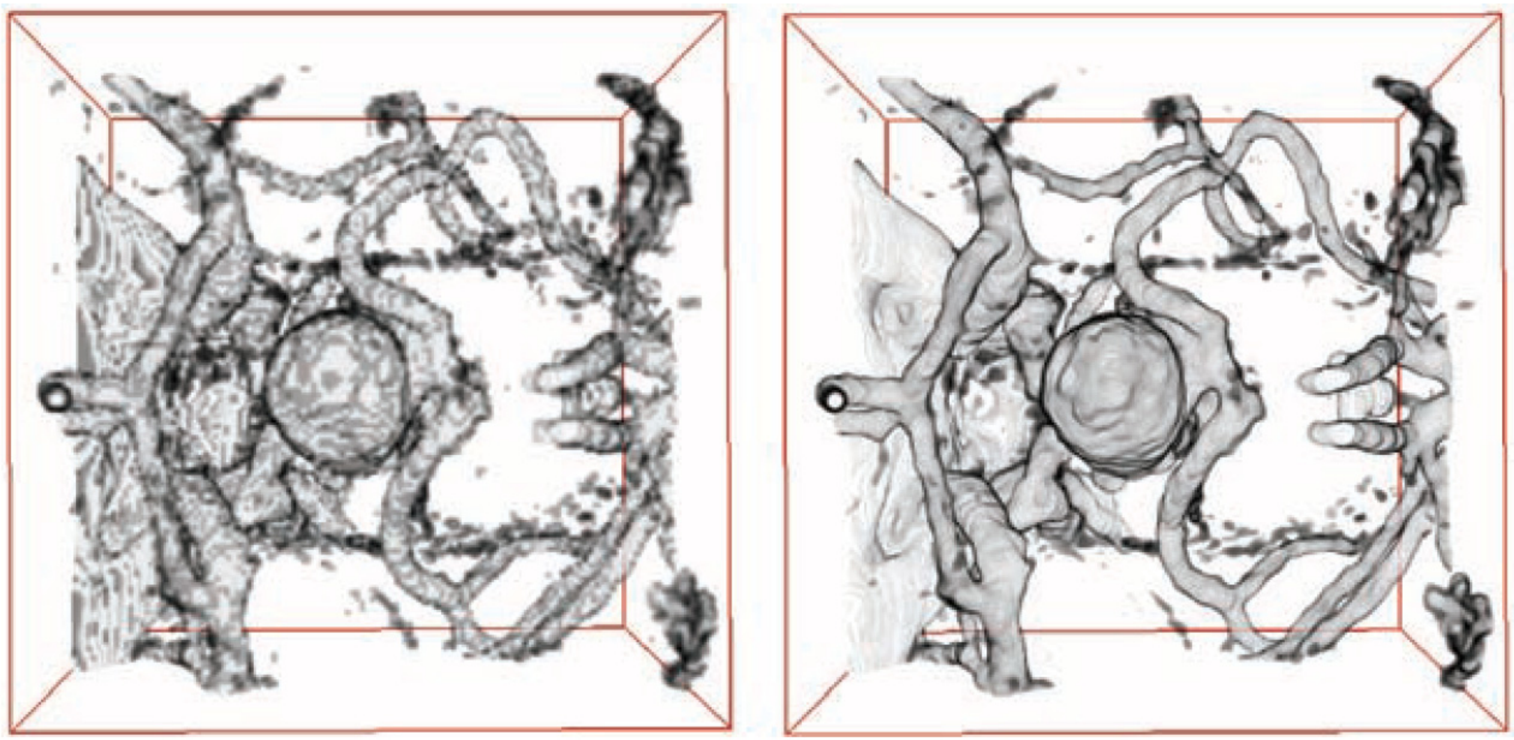
# Pre/post-interpolative transfer functions

- **Pre-interpolative transfer function**
  - Apply a transfer function before data interpolation
- **Post-interpolative transfer function**
  - Apply a transfer function after data interpolation



# Pre- v.s post-interpolative transfer functions

- **Comparison between two interpolative transfer functions**



# Pre- v.s post-interpolative transfer functions

- **Why post-interpolative transfer function produces much better results?**
  - Consider a continuous 1D signal obtained by casting a ray through the volume
  - Reconstruction from discrete values

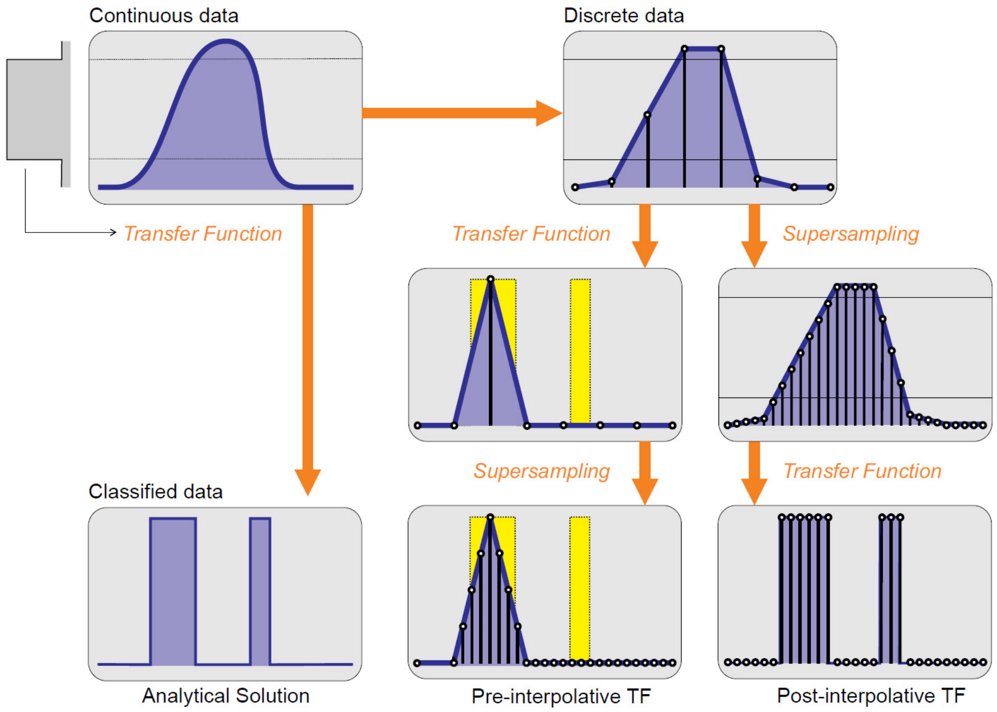
$$f(x) = \sum_k f(k \cdot \tau) \cdot \mathrm{sinc}\big(\frac{1}{\tau}(x - k\tau)\big)\,, \qquad k \in \mathbb{N}$$

  - Application of a transfer function T to the discrete samples

$$T\big(f(x)\big) \neq \sum_k T\big(f(k \cdot \tau)\big) \cdot \mathrm{sinc}\big(\frac{1}{\tau}(x - k\tau)\big)$$

# Pre- v.s post-interpolative transfer functions

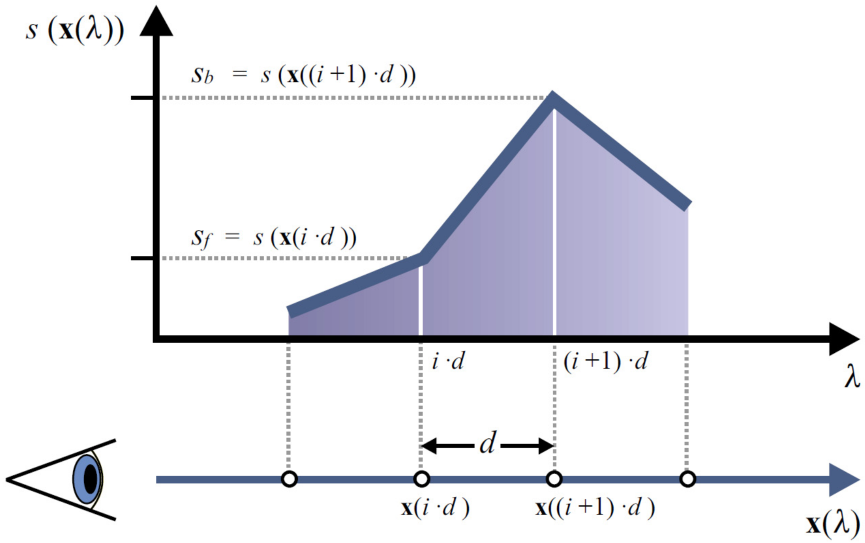
- **Transfer function introduces additional high-frequencies into the result signal**
  – Data grid only accounts for the frequency of the original signal



# Pre-integrated transfer functions

- **Look at the transfer function again**
  - May introduce additional high frequency components
  - Cause aliasing artifacts during ray integration

- **Sampling & interpolation along path**



# Pre-integrated transfer functions

- **Recall definition**

$$T(s_1, s_2) = e^{-\tau(s_1,s_2)} = e^{-\int_{s_1}^{s_2} \kappa(t)\,\mathrm{d}t} \qquad T_i = T(s_{i-1}, s_i)\,, \qquad c_i = \int_{s_{i-1}}^{s_i} q(s)T(s, s_i)\,\mathrm{d}s$$

- **Approximation**

$$\begin{aligned} \alpha_i &= 1 - \exp\left(-\int_{i\,d}^{(i+1)d} \kappa\Big(s\big(\mathbf{x}(\lambda)\big)\Big)\mathrm{d}\lambda\right) \\ &\approx 1 - \exp\left(-\int_0^1 \kappa\big((1-\omega)s_f + \omega s_b\big)d\,\mathrm{d}\omega\right) \end{aligned}$$

$$\begin{aligned} c_i \approx &\int_0^1 q\big((1-\omega)s_f + \omega s_b\big) \\ &\times \exp\Big(-\int_0^\omega \kappa\big((1-\omega')s_f + \omega' s_b\big)d\,\mathrm{d}\omega'\Big)d\,\mathrm{d}\omega \end{aligned}$$

$$\Rightarrow \quad I \approx \sum_{i=0}^{n} c_i \prod_{j=0}^{i-1} (1 - \alpha_j)$$

# Pre-integrated transfer functions

- **Efficiency improvement**
  - Integral property

$$F(x) = \int_a^b f(x)\,\mathrm{d}x \longleftrightarrow F(x) = G(b) - G(a) \quad G(s) = \int_0^s f(x)\,\mathrm{d}x$$

  - Change of variables

$$\begin{aligned} \alpha_i &\approx 1 - \exp\left(-\int_0^1 \kappa\big((1-\omega)s_f + \omega s_b\big)d\,\mathrm{d}\omega\right) \\ &= 1 - \exp\left(-\frac{d}{s_b - s_f}\int_{s_f}^{s_b} \kappa(s)\mathrm{d}s\right) \\ &= 1 - \exp\left(-\frac{d}{s_b - s_f}\big(T(s_b) - T(s_f)\big)\right) \end{aligned}$$

$$T(s) = \int_0^s \kappa(s')\,\mathrm{d}s'$$

# Pre-integrated transfer functions

- **Efficiency improvement**
  - Change of variables

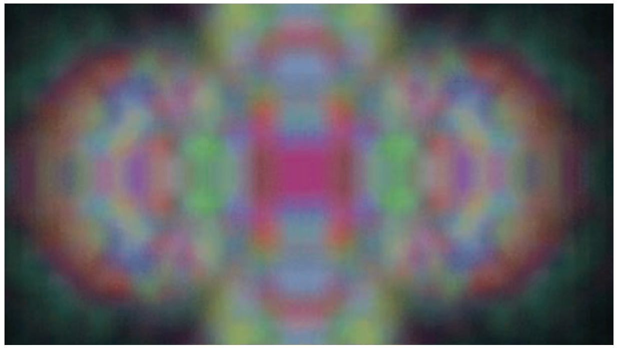
$$
\begin{aligned}
c_i &\approx \int_0^1 q\big((1-\omega)s_f + \omega s_b\big)d \,\mathrm{d}\omega \\
&= \frac{d}{s_b - s_f}\int_{s_f}^{s_b} q(s)\mathrm{d}s \\
&= \frac{d}{s_b - s_f}\big(K(s_b) - K(s_f)\big)
\end{aligned}
$$

$$K(s) = \int_0^s q(s')\,\mathrm{d}s'$$

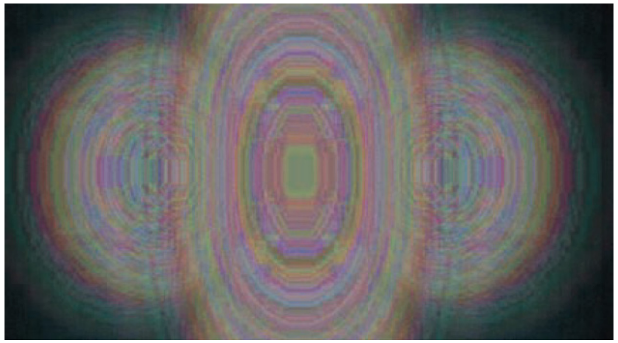
  - T and K are iteratively computed using ray marching

# Pre-integrated transfer functions

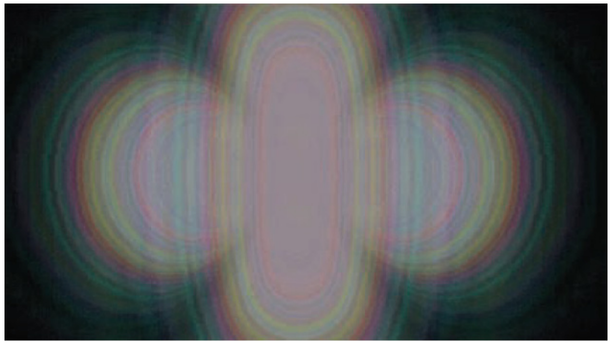
- **Comparison**
  – Pre-, post-, and pre-integrated classification for a random transfer function



Pre-interpolative transfer function



Post-interpolative transfer function



Pre-integrated transfer function

# Gradient-based local illumination

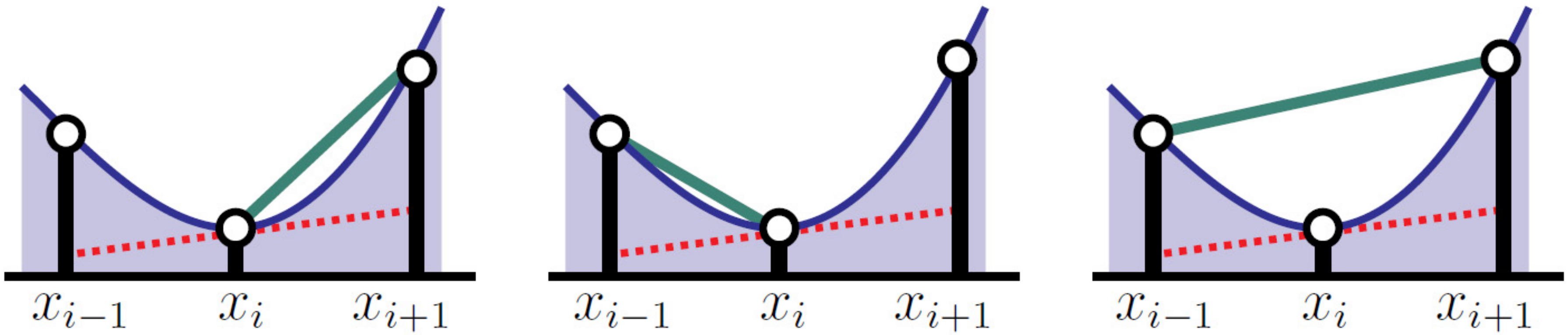
- **Assumption**
  - External light is reflected at iso-surfaces
  - Normal vector we use for shading

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}, \qquad \text{if } \|\nabla f(\mathbf{x})\| \neq 0$$

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x} \\ \frac{\partial f(\mathbf{x})}{\partial y} \\ \frac{\partial f(\mathbf{x})}{\partial z} \end{pmatrix}$$

# Gradient-based local illumination

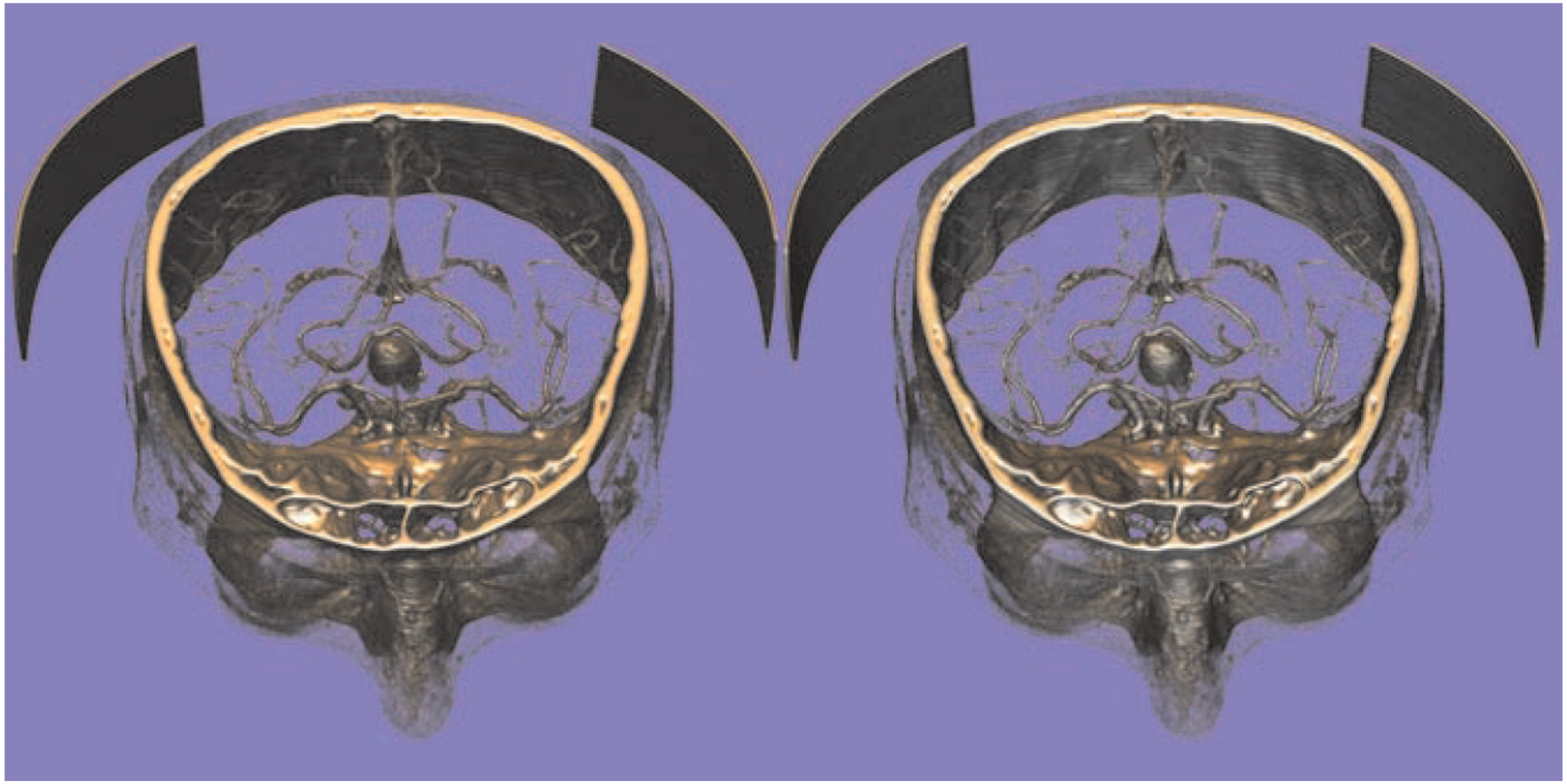
- **Gradient estimation**
  - Finite differences




$$\nabla f(x,y,z) \approx \frac{1}{2h} \begin{pmatrix} f(x+h,y,z) - f(x-h,y,z) \\ f(x,y+h,z) - f(x,y-h,z) \\ f(x,y,z+h) - f(x,y,z-h) \end{pmatrix}$$

# Gradient-based local illumination

- **Volumetric local illumination**
  - Blinn-Phong (left) and Cook-Torrance (right) models



# Next lecture: Volume rendering 2