Volumetric Rendering

3D Computer Graphics

Barnabás “Barney” Börcsök
bborcsok@iit.bme.hu

Computer Graphics Group (IIT)
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics

Spring Semester, 2022/2023

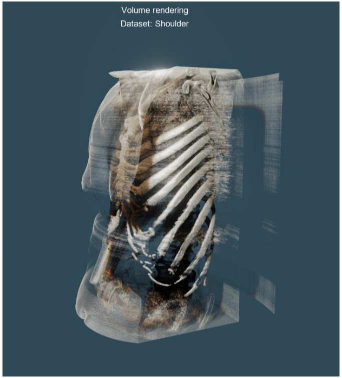
Motivation

Motivation

Motivation

Te Ka, a character from Disney's Moana (2016). Illustration from [1].

Motivation



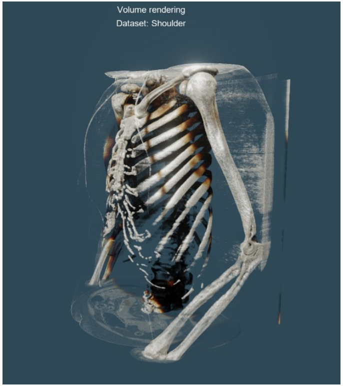
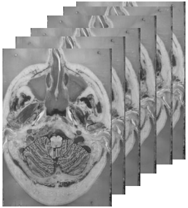
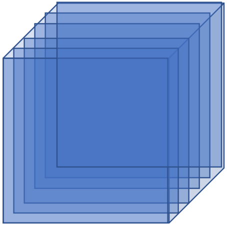


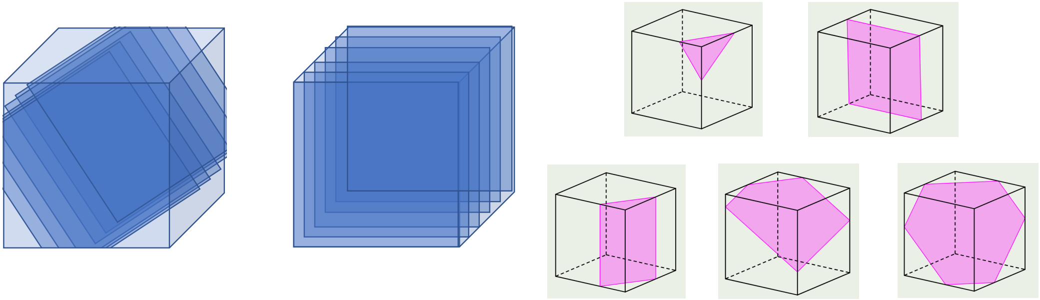
Illustration from [2].

Volume Rendering with Slices





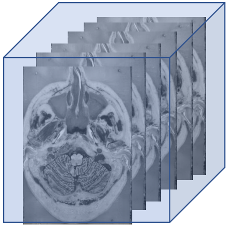
Volume Rendering with Slices



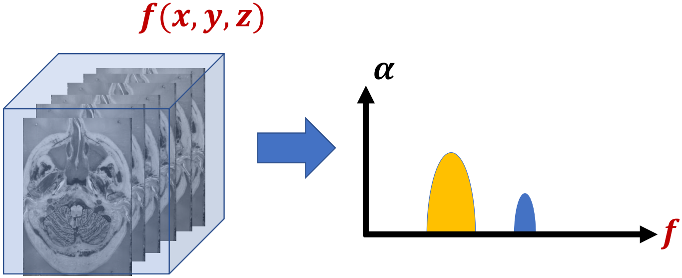
Interactive box-plane intersections: `https://www.geogebra.org/m/aY75dEkf`

Transfer Function

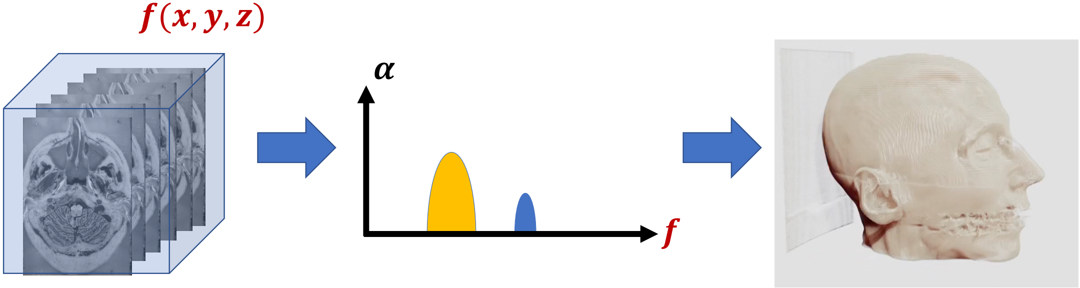
$f(x, y, z)$



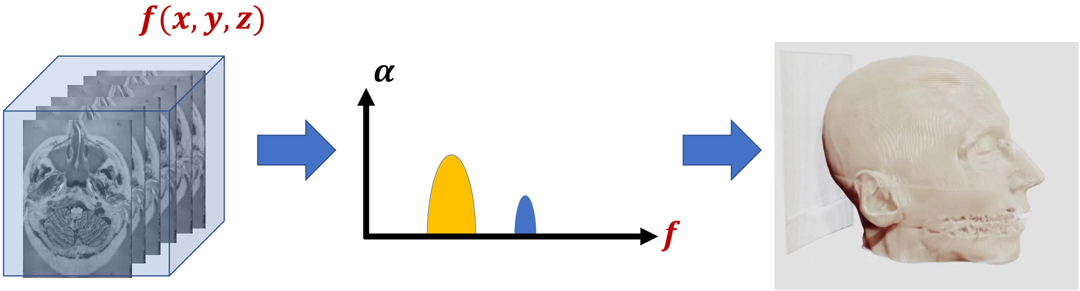
Transfer Function



Transfer Function

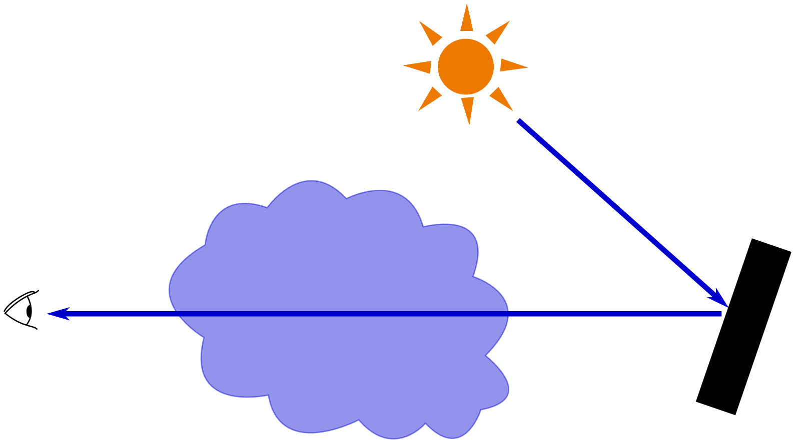


Transfer Function

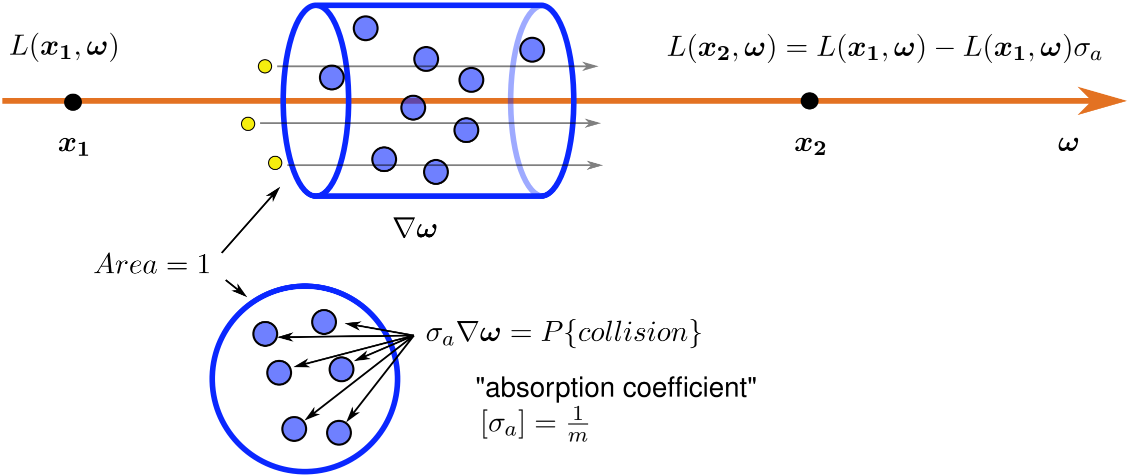


Let’s see this in action! (Video Demo)

Propagation of light in a medium

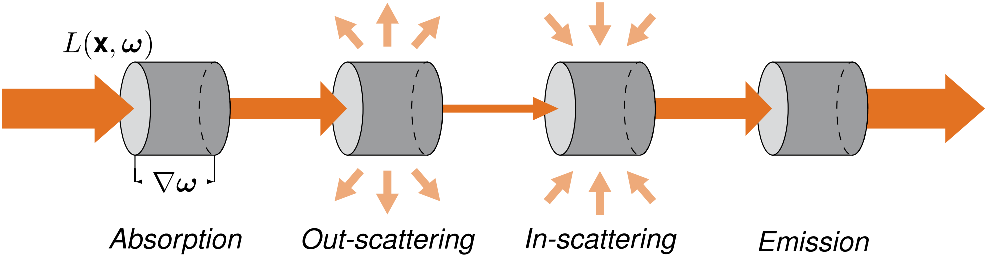


Change of radiance in a differential volume

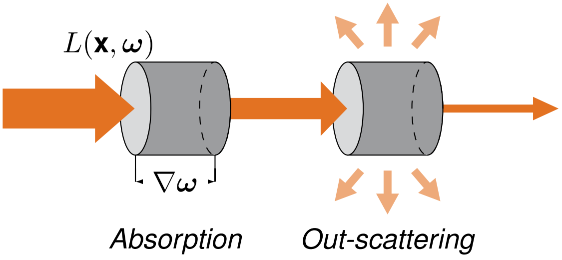


Possible interactions

between the volume and the light traveling through the medium



Summing up the losses



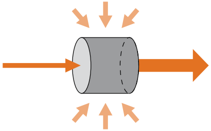
σ_a : Absorption coefficient
σ_s : Scattering coefficient
$\sigma_a + \sigma_s = \sigma_t$: Extinction coefficient

We lose $\sigma_t(\boldsymbol{x})L(\boldsymbol{x}, \boldsymbol{\omega})$ radiance due to *absorption* and *out-scattering*.

$\sigma_t \implies$ Homogeneous

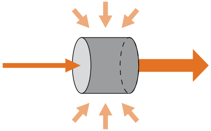
$\sigma_t(\boldsymbol{x}) \implies$ Heterogeneous

In-scattered radiance



$$L_s(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{S^2} f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}') L_i(\boldsymbol{x}, \boldsymbol{\omega}') d\boldsymbol{\omega}'$$

In-scattered radiance



$$L_s(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{S^2} f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}') L_i(\boldsymbol{x}, \boldsymbol{\omega}') d\boldsymbol{\omega}'$$

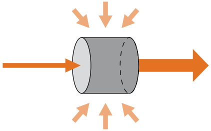
Phase function

$f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}')$

$\approx BSDF$

(in surface rendering)

In-scattered radiance



$$L_s(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{S^2} f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}') L_i(\boldsymbol{x}, \boldsymbol{\omega}') d\boldsymbol{\omega}'$$

Phase function

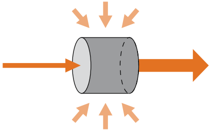
$f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}')$

$\approx BSDF$

(in surface rendering)

- scattering at point x, given incident (ω) and outgoing (ω') directions

In-scattered radiance



$$L_s(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{S^2} f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}') L_i(\boldsymbol{x}, \boldsymbol{\omega}') d\boldsymbol{\omega}'$$

Phase function

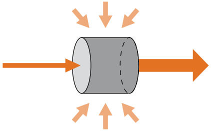
$f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}')$

$\approx BSDF$

(in surface rendering)

- scattering at point x, given incident (ω) and outgoing (ω') directions
- $\int_{S^2} f_p = 1$

In-scattered radiance



$$L_s(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{S^2} f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}') L_i(\boldsymbol{x}, \boldsymbol{\omega}') d\boldsymbol{\omega}'$$

Phase function

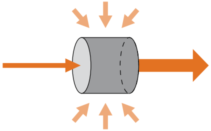
$f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}')$

$\approx BSDF$

(in surface rendering)

- scattering at point $\boldsymbol{x}$, given incident ($\boldsymbol{\omega}$) and outgoing ($\boldsymbol{\omega}'$) directions
- $\int_{S^2} f_p = 1$
- $f_p(\theta)\big|_{\theta = \measuredangle(\boldsymbol{\omega}, \boldsymbol{\omega}')}$

In-scattered radiance



$$L_s(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{S^2} f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}') L_i(\boldsymbol{x}, \boldsymbol{\omega}') d\boldsymbol{\omega}'$$

Phase function

$f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}')$

$\approx BSDF$

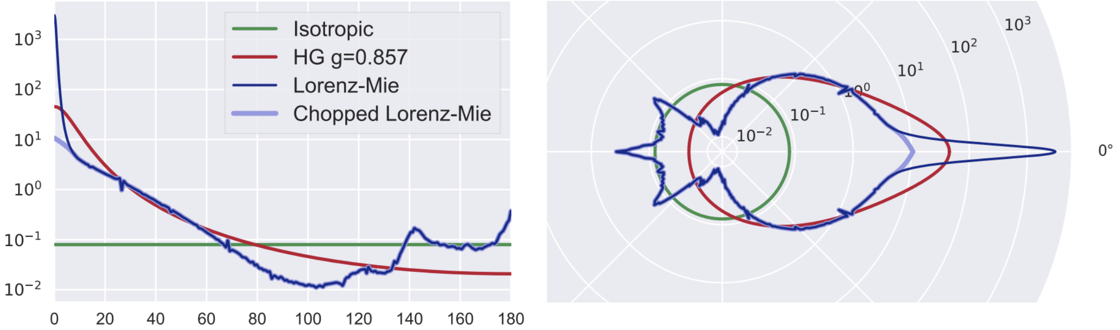
(in surface rendering)

- scattering at point $\boldsymbol{x}$, given incident ($\boldsymbol{\omega}$) and outgoing ($\boldsymbol{\omega}'$) directions
- $\int_{S^2} f_p = 1$
- $f_p(\theta)\big|_{\theta = \measuredangle(\boldsymbol{\omega}, \boldsymbol{\omega}')}$
- $f_p(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\omega}') = 1/(4\pi)$, if the medium is *isotropic* (otherwise, *anisotropic*)

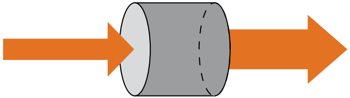
Phase function examples

- isotropic: $f_p = \frac{1}{4\pi}$
- Henyey-Greenstein: $f_p(\theta) = \frac{1}{4\pi} \frac{1-g^2}{(1+g^2-2g\cos(\theta))^{3/2}}$

Illustration from [5]:



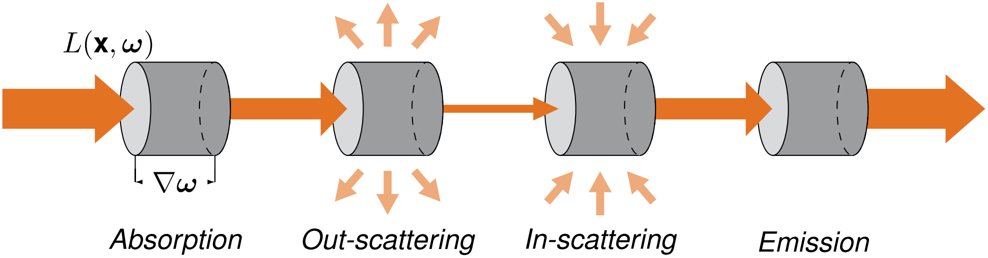
Emission



$$L_e(\boldsymbol{x}, \boldsymbol{\omega})$$

$$\sigma_a(\boldsymbol{x}) L_e(\boldsymbol{x}, \boldsymbol{\omega})$$

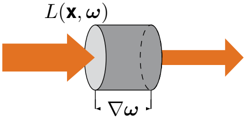
Assembling all the parts



- Loses $\sigma_a L(x, \omega)$ due to absorption
- Loses $\sigma_s L(x, \omega)$ due to out-scattering
- Gains $\sigma_s L_i(x, \omega)$ due to in-scattering
- Gains $\sigma_a L_e(x, \omega)$ due to emission

RTE – Radiative Transfer Equation

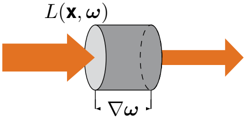
The change in radiance L traveling along direction $\boldsymbol{\omega}$ through a differential volume element at point $\boldsymbol{x}$.



$$(\boldsymbol{\omega}\nabla)L(\boldsymbol{x},\boldsymbol{\omega}) = \underbrace{-\sigma_t(\boldsymbol{x})L(\boldsymbol{x},\boldsymbol{\omega})}_{Extinction} + \underbrace{\sigma_s(\boldsymbol{x})L_s(\boldsymbol{x},\boldsymbol{\omega})}_{In-scattering} + \underbrace{\sigma_a(\boldsymbol{x})L_e(\boldsymbol{x},\boldsymbol{\omega})}_{Emission}$$

RTE – Radiative Transfer Equation

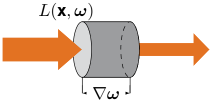
The change in radiance L traveling along direction $\boldsymbol{\omega}$ through a differential volume element at point $\boldsymbol{x}$.



$$(\boldsymbol{\omega}\nabla)L(\boldsymbol{x},\boldsymbol{\omega}) = \underbrace{-\sigma_t(\boldsymbol{x})L(\boldsymbol{x},\boldsymbol{\omega})}_{Extinction} + \underbrace{\sigma_s(\boldsymbol{x})L_s(\boldsymbol{x},\boldsymbol{\omega})}_{In-scattering} + \underbrace{\sigma_a(\boldsymbol{x})L_e(\boldsymbol{x},\boldsymbol{\omega})}_{Emission}$$

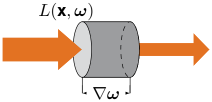
Let's integrate it!

Integrating the loss of radiance



$$L(\boldsymbol{x} + dx) = L(\boldsymbol{x}) - L(\boldsymbol{x})\sigma_t(\boldsymbol{x})dx\Big|_{dx=\nabla\boldsymbol{\omega},L(\boldsymbol{x})=L(\boldsymbol{x},\boldsymbol{\omega})}$$

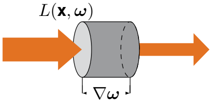
Integrating the loss of radiance



$$L(\boldsymbol{x} + dx) = L(\boldsymbol{x}) - L(\boldsymbol{x})\sigma_t(\boldsymbol{x})dx\Big|_{dx=\nabla\boldsymbol{\omega}, L(\boldsymbol{x})=L(\boldsymbol{x},\boldsymbol{\omega})}$$

$$\boxed{\frac{dL(\boldsymbol{x})}{dx} = -L(\boldsymbol{x})\sigma_t(\boldsymbol{x})}$$ ("exponential extinction")

Integrating the loss of radiance

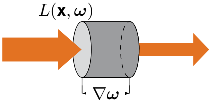


$$L(\boldsymbol{x}+dx) = L(\boldsymbol{x}) - L(\boldsymbol{x})\sigma_t(\boldsymbol{x})dx\Big|_{dx=\nabla\boldsymbol{\omega},L(\boldsymbol{x})=L(\boldsymbol{x},\boldsymbol{\omega})}$$

$$\boxed{\frac{dL(\boldsymbol{x})}{dx} = -L(\boldsymbol{x})\sigma_t(\boldsymbol{x})}$$ ("exponential extinction")

$$\int_{L(x)}^{L(x+S)} \frac{1}{L}dL = -\int_0^S \sigma_t(\boldsymbol{x})dx$$

Integrating the loss of radiance



$$L(\boldsymbol{x} + dx) = L(\boldsymbol{x}) - L(\boldsymbol{x})\sigma_t(\boldsymbol{x})dx \Big|_{dx=\nabla\boldsymbol{\omega}, L(\boldsymbol{x})=L(\boldsymbol{x},\boldsymbol{\omega})}$$

$$\boxed{\frac{dL(\boldsymbol{x})}{dx} = -L(\boldsymbol{x})\sigma_t(\boldsymbol{x})}$$ ("exponential extinction")

$$\int_{L(x)}^{L(x+S)} \frac{1}{L} dL = -\int_0^S \sigma_t(\boldsymbol{x})dx$$

$$ln(L(\boldsymbol{x} + S)) - ln(L(\boldsymbol{x})) = -\int_0^S \sigma_t(\boldsymbol{x})dx$$

Transmittance
The Beer-Lambert Law

$$\frac{dL(\boldsymbol{x})}{dx} = -L(\boldsymbol{x})\sigma_t(\boldsymbol{x}) \quad \text{("exponential extinction")}$$

$$\boxed{L(\boldsymbol{x} + S) = L(\boldsymbol{x})e^{-\int_0^S \sigma_t(\boldsymbol{x}+s)ds}}$$

Transmittance

The Beer-Lambert Law

$$\frac{dL(\boldsymbol{x})}{dx} = -L(\boldsymbol{x})\sigma_t(\boldsymbol{x}) \quad \text{("exponential extinction")}$$

$$\boxed{L(\boldsymbol{x}+S) = L(\boldsymbol{x})e^{-\int_0^S \sigma_t(\boldsymbol{x}+s)ds}}$$

Usually written as:

$$e^{-\int_0^y \sigma_t(\boldsymbol{x}-s\boldsymbol{\omega})ds} = T(\boldsymbol{x},\boldsymbol{y})$$

"transmittance coefficient" $T(\boldsymbol{x},\boldsymbol{y})$

net reduction factor between $\boldsymbol{x}$ and $\boldsymbol{y}$

due to absorption and out-scattering

Transmittance

The Beer-Lambert Law

$$\frac{dL(\boldsymbol{x})}{dx} = -L(\boldsymbol{x})\sigma_t(\boldsymbol{x}) \quad \text{("exponential extinction")}$$

$$\boxed{L(\boldsymbol{x}+S) = L(\boldsymbol{x})e^{-\int_0^S \sigma_t(\boldsymbol{x}+s)ds}}$$

Usually written as:

$e^{-\int_0^y \sigma_t(\boldsymbol{x}-s\boldsymbol{\omega})ds} = T(\boldsymbol{x},\boldsymbol{y})$

"transmittance coefficient" $T(\boldsymbol{x},\boldsymbol{y})$

net reduction factor between $\boldsymbol{x}$ and $\boldsymbol{y}$

due to absorption and out-scattering

$\int_0^y \sigma_t(\boldsymbol{x}-s\boldsymbol{\omega})ds = \tau(\boldsymbol{x},\boldsymbol{y})$

"optical thickness" τ

Transmittance

The Beer-Lambert Law

$$\frac{dL(\boldsymbol{x})}{dx} = -L(\boldsymbol{x})\sigma_t(\boldsymbol{x})$$ ("exponential extinction")

$$\boxed{L(\boldsymbol{x}+S) = L(\boldsymbol{x})e^{-\int_0^S \sigma_t(\boldsymbol{x}+s)ds}}$$

Usually written as:

$$e^{-\int_0^y \sigma_t(\boldsymbol{x}-s\boldsymbol{\omega})ds} = T(\boldsymbol{x},\boldsymbol{y})$$

"transmittance coefficient" $T(\boldsymbol{x},\boldsymbol{y})$

net reduction factor between $\boldsymbol{x}$ and $\boldsymbol{y}$

due to absorption and out-scattering

$$\int_0^y \sigma_t(\boldsymbol{x}-s\boldsymbol{\omega})ds = \tau(\boldsymbol{x},\boldsymbol{y})$$

"optical thickness" τ

$$T(t) = e^{-\tau(t)} = e^{-\int_0^t \sigma_t(\boldsymbol{x}-s\boldsymbol{\omega})ds}$$

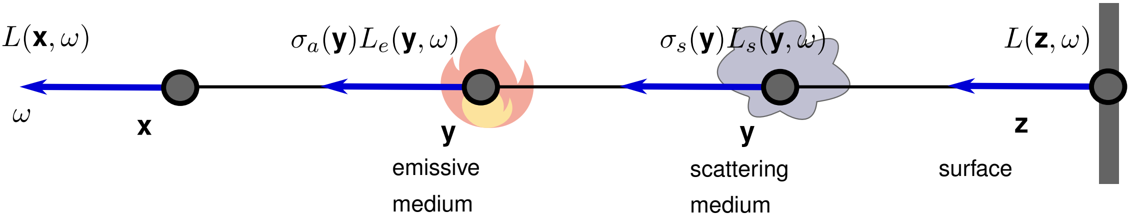
over distance t

RTE – Radiative Transfer Equation

The integral version

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = \int_0^\infty \underbrace{e^{-\int_0^y \sigma_t(\boldsymbol{x} - s\boldsymbol{\omega})ds}}_{\text{Transmittance } T(\boldsymbol{x},\boldsymbol{y})} \Big[\underbrace{\sigma_s(\boldsymbol{y}) L_s(\boldsymbol{y}, \boldsymbol{\omega})}_{\text{in-scatter}} + \underbrace{\sigma_a(\boldsymbol{y}) L_e(\boldsymbol{y}, \boldsymbol{\omega})}_{\text{emission}} \Big] d\boldsymbol{y}$$

VRE – Volume Rendering Equation



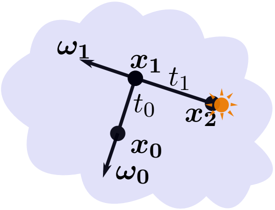
$$L(\boldsymbol{x}, \boldsymbol{\omega}) = \int_0^z T(\boldsymbol{x}, \boldsymbol{y})\big[\sigma_a(\boldsymbol{y})L_e(\boldsymbol{y}, \boldsymbol{\omega}) + \sigma_s(\boldsymbol{y})L_s(\boldsymbol{y}, \boldsymbol{\omega})\big]dy + T(\boldsymbol{x}, \mathbf{z})L(\mathbf{z}, \boldsymbol{\omega})$$

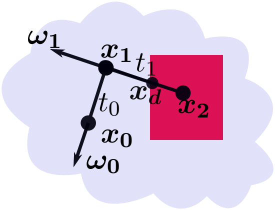
Tracking

In homogeneous volumes

- Simulate how a photon bounces around inside a volume
- Explicitly modeling absorption and scattering effects

$$T(t) = e^{-\int_0^t \sigma_t(\boldsymbol{x}-s\boldsymbol{\omega})ds} = e^{-\int_0^t \sigma_t ds} = \boxed{e^{-\sigma_t t} = T(t)}$$



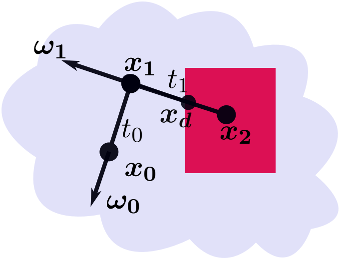


Tracking

In heterogeneous volumes

What happens if the volume is **not homogeneous**? $\implies \quad \sigma_t(\boldsymbol{x})$

$\implies$ apply closed-form tracking to homogeneous sub-parts? $\implies \quad \sigma_t$

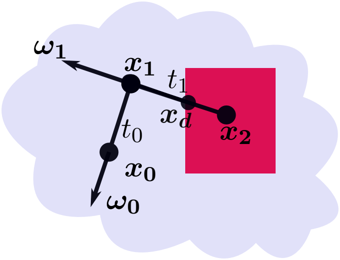


Tracking

In heterogeneous volumes

What happens if the volume is **not homogeneous**? $\implies \sigma_t(\boldsymbol{x})$

$\implies$ apply closed-form tracking to homogeneous sub-parts? $\implies \sigma_t$

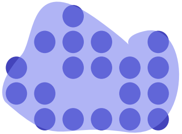


If only...

Delta tracking / Woodcock tracking

Introducing null-collisions

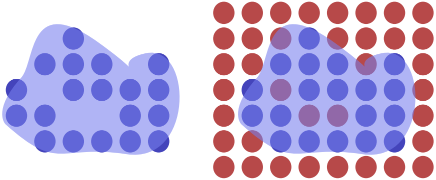
1. Problem: the volume is heterogeneous
2. Idea: **Increase the number of interactions** to make it homogeneous, but **reject** some of the interactions $\implies$ **null-collisions**



Delta tracking / Woodcock tracking

Introducing null-collisions

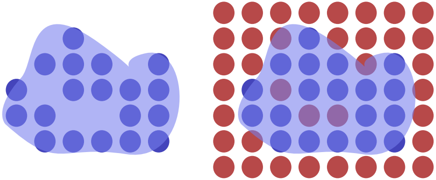
1. Problem: the volume is heterogeneous
2. Idea: **Increase the number of interactions** to make it homogeneous, but **reject** some of the interactions $\Longrightarrow$ **null-collisions**



Delta tracking / Woodcock tracking

Introducing null-collisions

1. Problem: the volume is heterogeneous
2. Idea: **Increase the number of interactions** to make it homogeneous, but **reject** some of the interactions $\Longrightarrow$ **null-collisions**



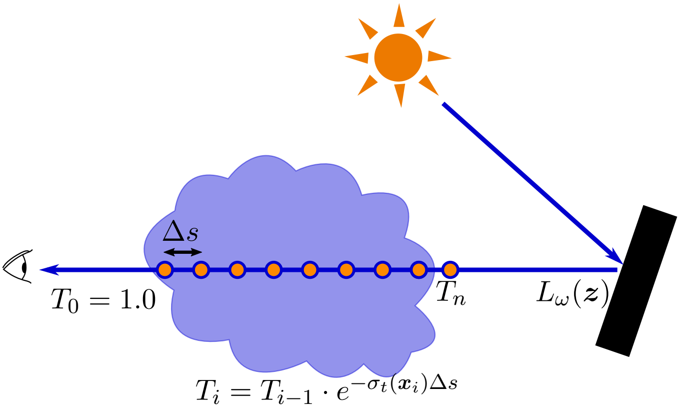
$$\bar{\sigma} = \sigma_s(\boldsymbol{x}) + \sigma_a(\boldsymbol{x}) + \sigma_n(\boldsymbol{x})$$

$$\sigma_n(\boldsymbol{x}) = \bar{\sigma} - \sigma_t(\boldsymbol{x})$$

$$T_{\bar{\sigma}}(\boldsymbol{x}, \boldsymbol{y}) = e^{-\int_0^y \sigma_s(\boldsymbol{s}) + \sigma_a(\boldsymbol{s}) + \sigma_n(\boldsymbol{s}) d\boldsymbol{s}}$$

Transmittance Estimation

Ray Marching [1]



[1] You will use ray marching during tomorrow's lab session for finding surfaces.

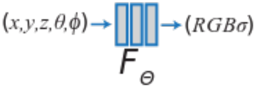
Acceleration Data Structures

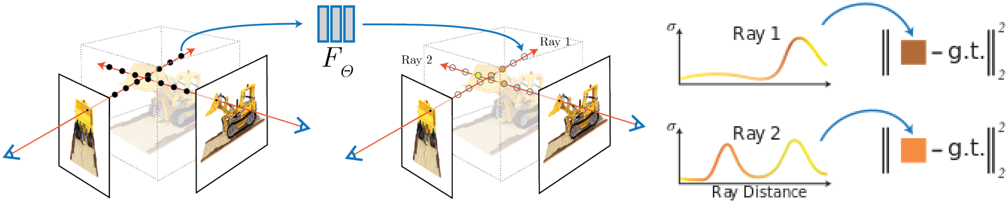
- Spatially-varying properties
- Data access usually dominates the render time
  $\Longrightarrow$ data structures are key for achieving good performance
- Volume data can quickly grow into hundreds of gigabytes for production
  - For example, peak storage needed for a single shot of the movie Soul was 80 TBs.[3]

`https://www.openvdb.org/`
`https://developer.nvidia.com/nanovdb`

Bonus: NeRF: Neural Radiance Fields

- See [4] (`https://www.matthewtancik.com/nerf`)





Bonus: Deep Learning for Rendering Clouds

- Vast cost of data access and tracking particles in high-albedo volumes (resulting in lots of scattering) – e.g. clouds
- Approximating the indirect in-scattered radiance with a Neural Network.
- They achieved $24\times$ speed-up! See [5].

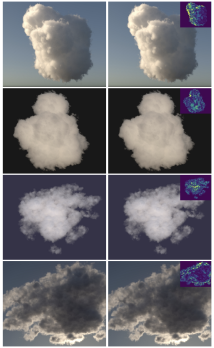
(a) 1 interaction
Render time = 0.017 h

(a) 4 interactions
Render time = 0.89 h

(a) 16 interactions
Render time = 6.5 h

(a) 64 interactions
Render time = 16 h

(a) All interactions
Render time = 34 h



(e) PT reference (f) RPNN (ours)

Volumetric Rendering

3D Computer Graphics

Barnabás “Barney” Börcsök
bborcsok@iit.bme.hu

Computer Graphics Group (IIT)
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics

Spring Semester, 2022/2023

Further Reading I

[1] Julian Fong, Magnus Wrenninge, Christopher Kulla, and Ralf Habel. Production Volume Rendering: SIGGRAPH 2017 Course. 2017. https://doi.org/10.1145/3084873.3084907.

[2] Zoltán Simon. Térfogatvizualizációs algoritmusok implementálása grafikus hardveren. Bsc thesis, 2022. https://diplomaterv.vik.bme.hu/hu/Theses/Terfogatvizualizacios-algoritmusok https://github.com/TheFlyingPiano99/HomebrewGraphicsEngine.

[3] Sasha Ouellet, Daniel Garcia, Stephen Gustafson, Matt Kuruc, Michael Lorenzen, George Nguyen, and Grace Gilbert. Rasterizing volumes and surfaces for crowds on soul. In *ACM SIGGRAPH 2020 Talks*, 2020. https://doi.org/10.1145/3388767.3407374.

[4] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for

Further Reading II

view synthesis. In *ECCV*, 2020. https://www.matthewtancik.com/nerf.

[5] Simon Kallweit, Thomas Müller, Brian Mcwilliams, Markus Gross, and Jan Novák. Deep scattering: Rendering atmospheric clouds with radiance-predicting neural networks. *ACM Trans. Graph.*, 2017. URL https://doi.org/10.1145/3130800.3130880. https://studios.disneyresearch.com/2017/11/20/deep-scattering-rendering-atmospheric-clouds-with-radiance-predicti