Computer Graphics Seminar

Interactive Relighting of Dynamic Refractive Objects

Michael Pfeuti

Universität Bern

UNIVERSITÄT BERN

Contents

UNIVERSITÄT BERN

Contents

UNIVERSITÄT BERN

Basis

ACM SIGGRAPH 2008 paper

Interactive Relighting Of Dynamic Refractive Objects

by

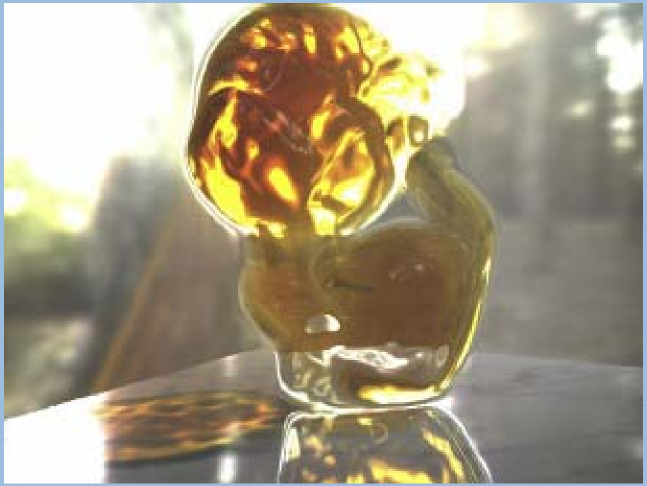
Xin Sun, Kun Zhou, Eric Stollnitz, Jiaoying Shi, and Baining Guo

UNIVERSITÄT BERN

Goal

- Paper:
  - Rendering of transparent objects with varying refraction indices at an *interactive* frame rate
  - Including effects:
    - Volumetric Caustics
    - Single Scattering
    - Dynamically changeable parameters (Camera, Light, Geometry)
  - Achieved by exploiting GPU parallelism through a special render pipeline
- **My Goal**: Reimplementation of this render pipeline and extensions

Goal (cont.)



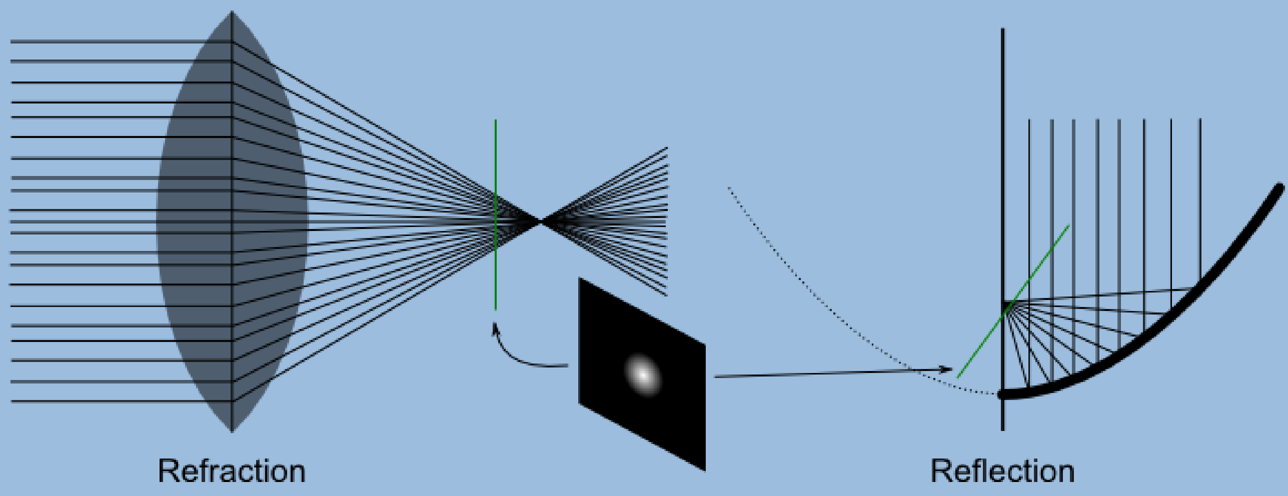
[Images from paper “Interactive Relighting of Dynamic Refractive Objects”] b UNIVERSITÄT BERN

Contents

Caustics

- **Definition**: A caustic is a light effect which occurs when:
  - light *passes through* curved transparent object
  - light is *reflected* of a curved specular surface

- Example: magnifying glass, parabolic mirror



Caustics (cont.)

[wikipedia.com]

[http://users.physik.tu-muenchen.de/cucke/ftp/lectures/kaustik3.pdf] b UNIVERSITÄT BERN

Scattering

> **Definition**:

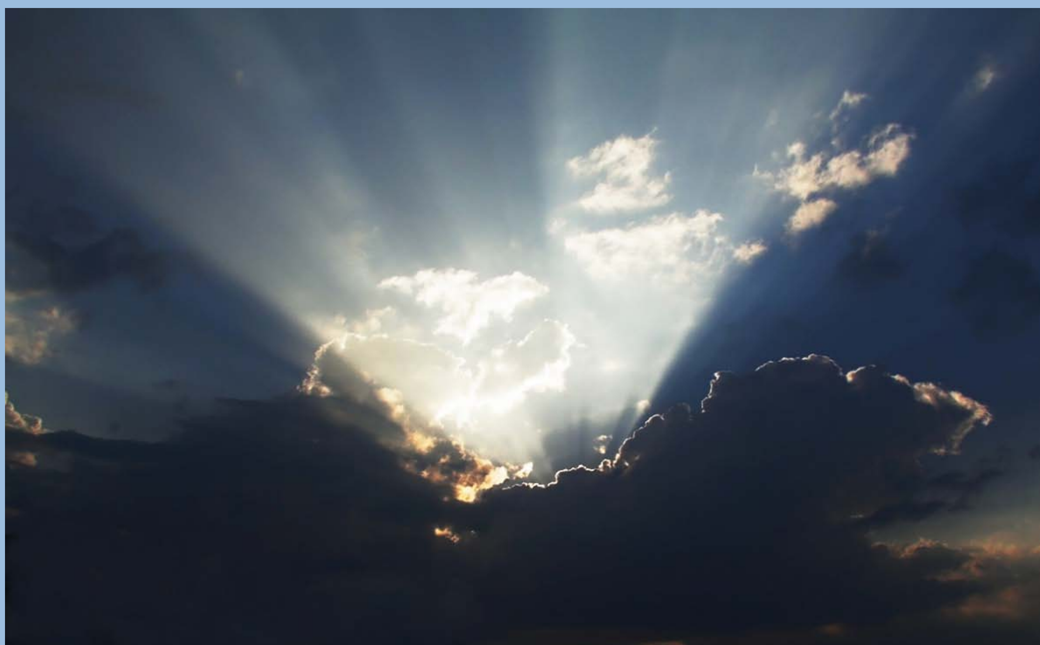
Scattering is a general physical process where some forms of radiation, such as light, sound, or moving particles, are forced to deviate from a straight trajectory by one or more localized *non-uniformities* in the medium through which they pass.

[wikipedia.com]

> Non-uniformities are for example:

— Small particles, bubbles, droplets, density fluctuations in fluids, defects in crystalline solids, surface roughness, cells in organisms, textile fibers in clothing, . . .

Scattering (cont.)

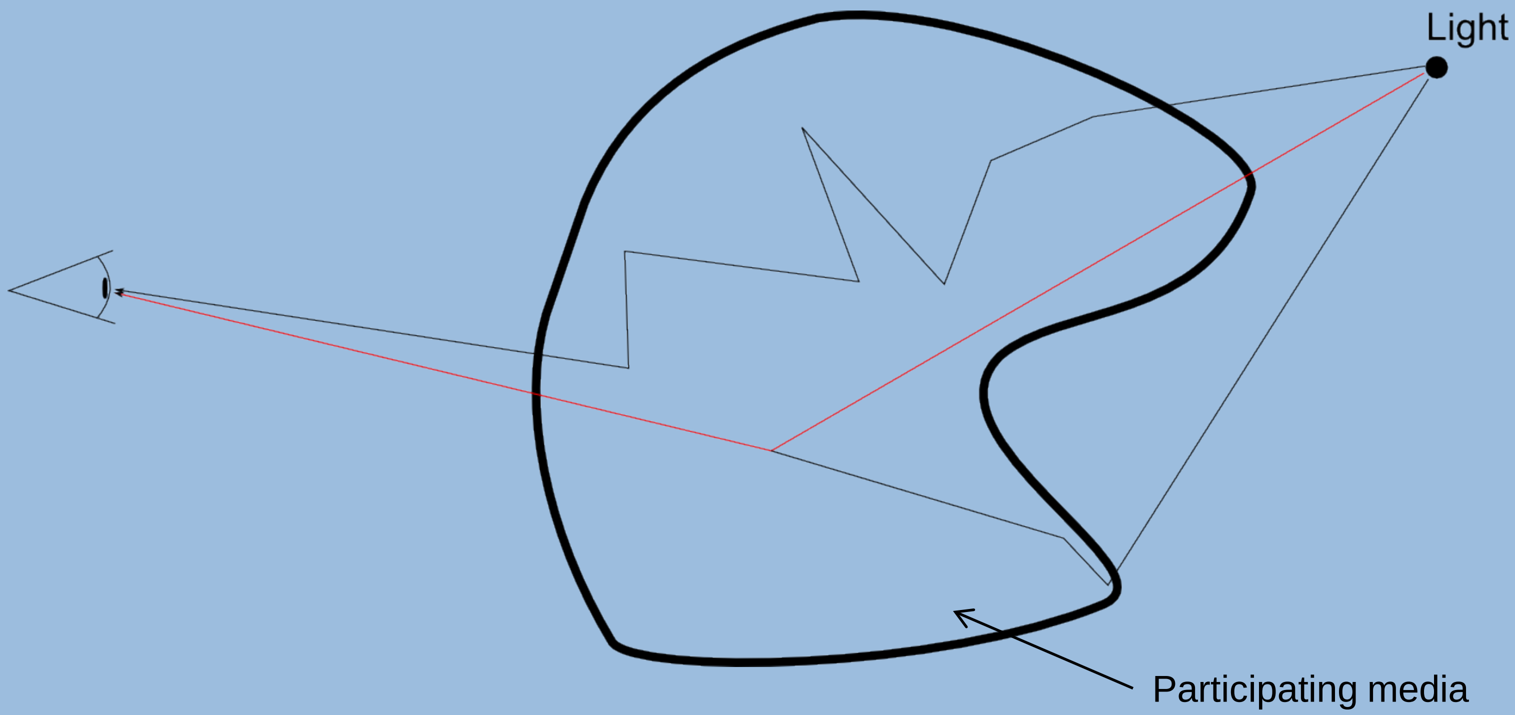


[wikipedia.com]

[wikipedia.com]

Scattering (cont.)

> Distinction between two cases:
  - *Single Scattering*: Light scatters only on one particle
  - *Multiple Scattering*: Light scatters on two or more particles



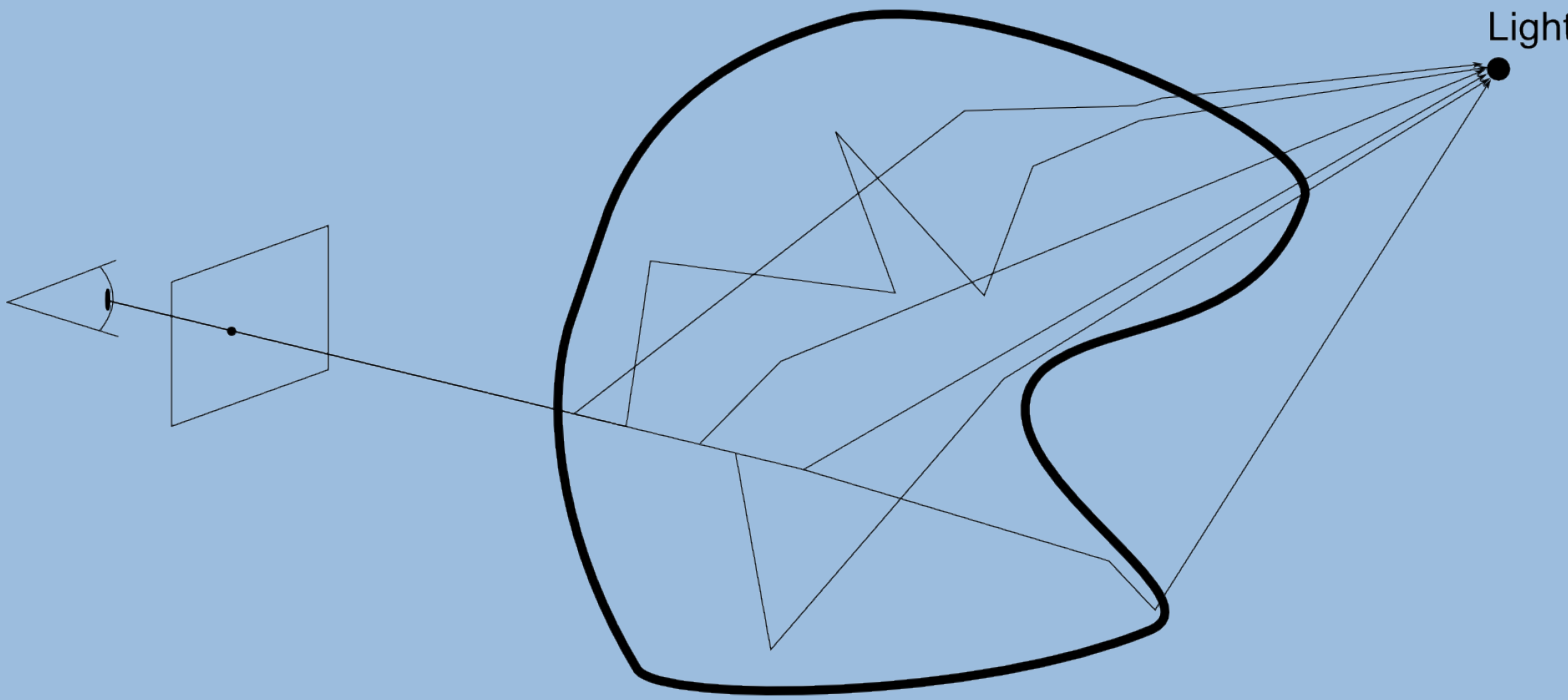
UNIVERSITÄT BERN

Contents

Rendering Pipeline (Overall Picture)

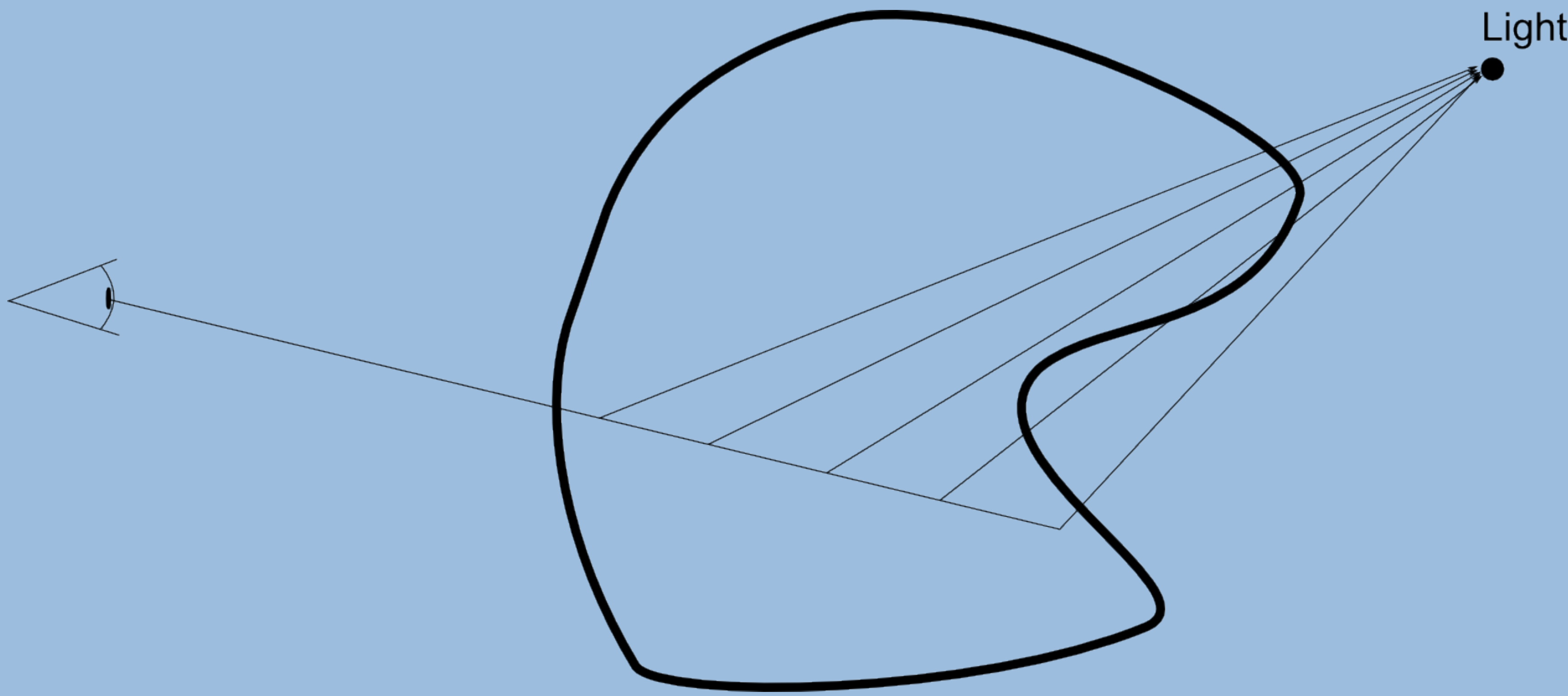
UNIVERSITÄT BERN

> Trace multiple paths for one primary ray.



Rendering Pipeline (Overall Picture)

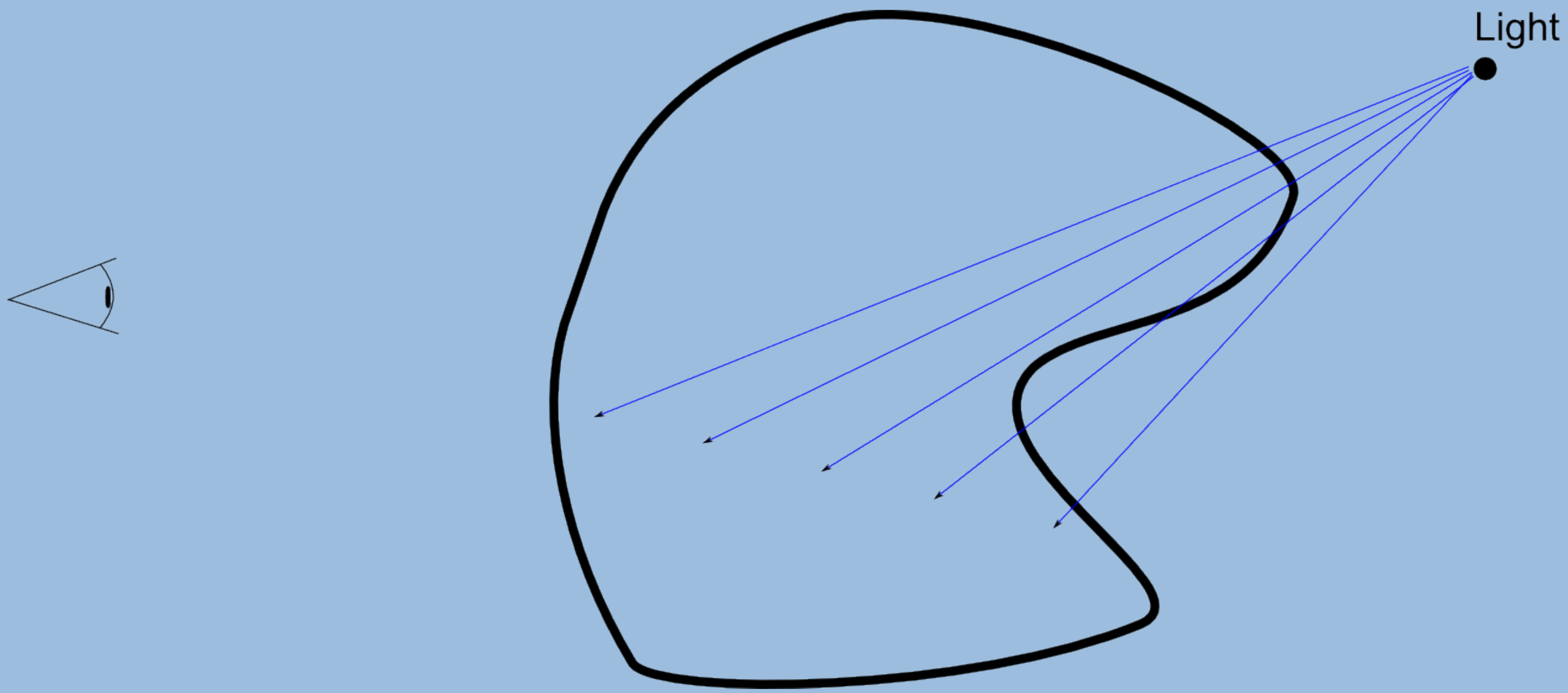
> Ray Marching through the volume.



Rendering Pipeline (Overall Picture)

UNIVERSITÄT BERN

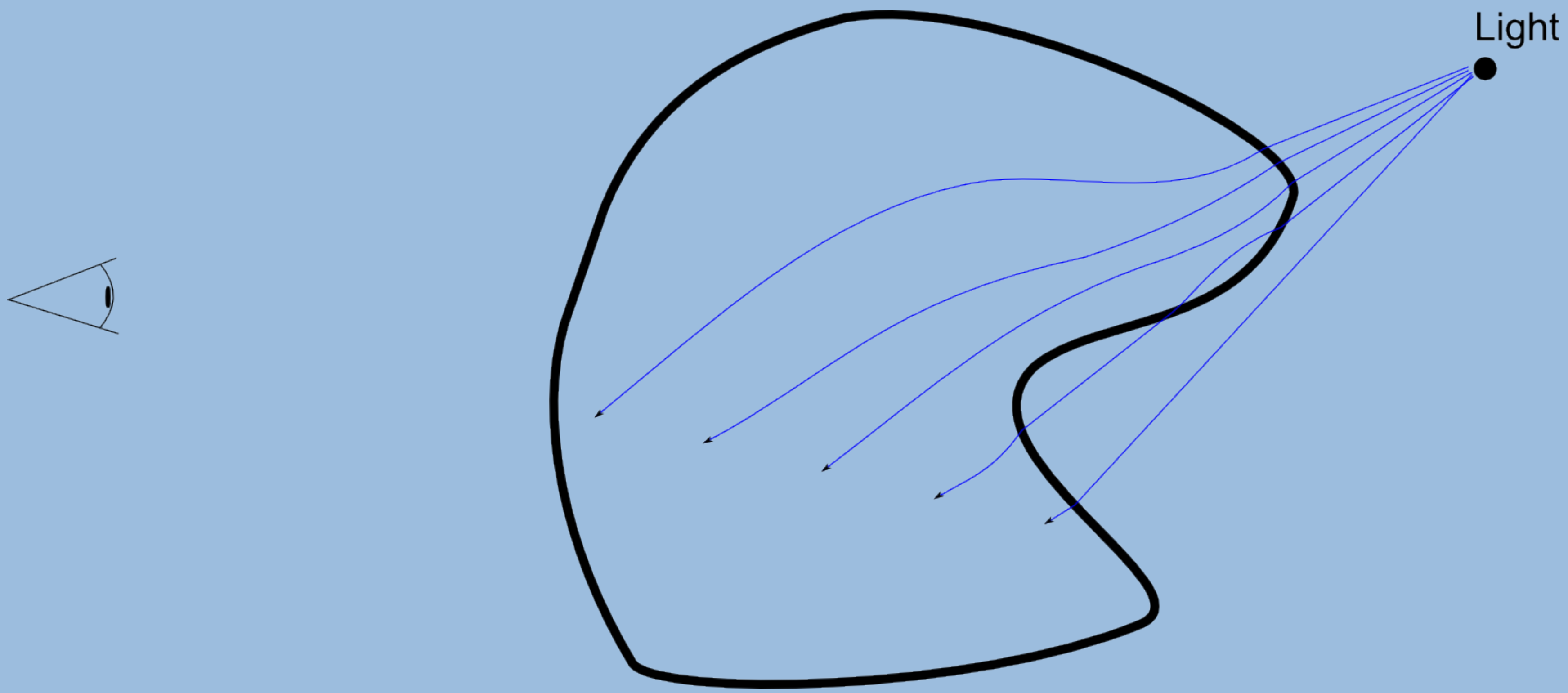
> Photon Tracing



UNIVERSITÄT BERN

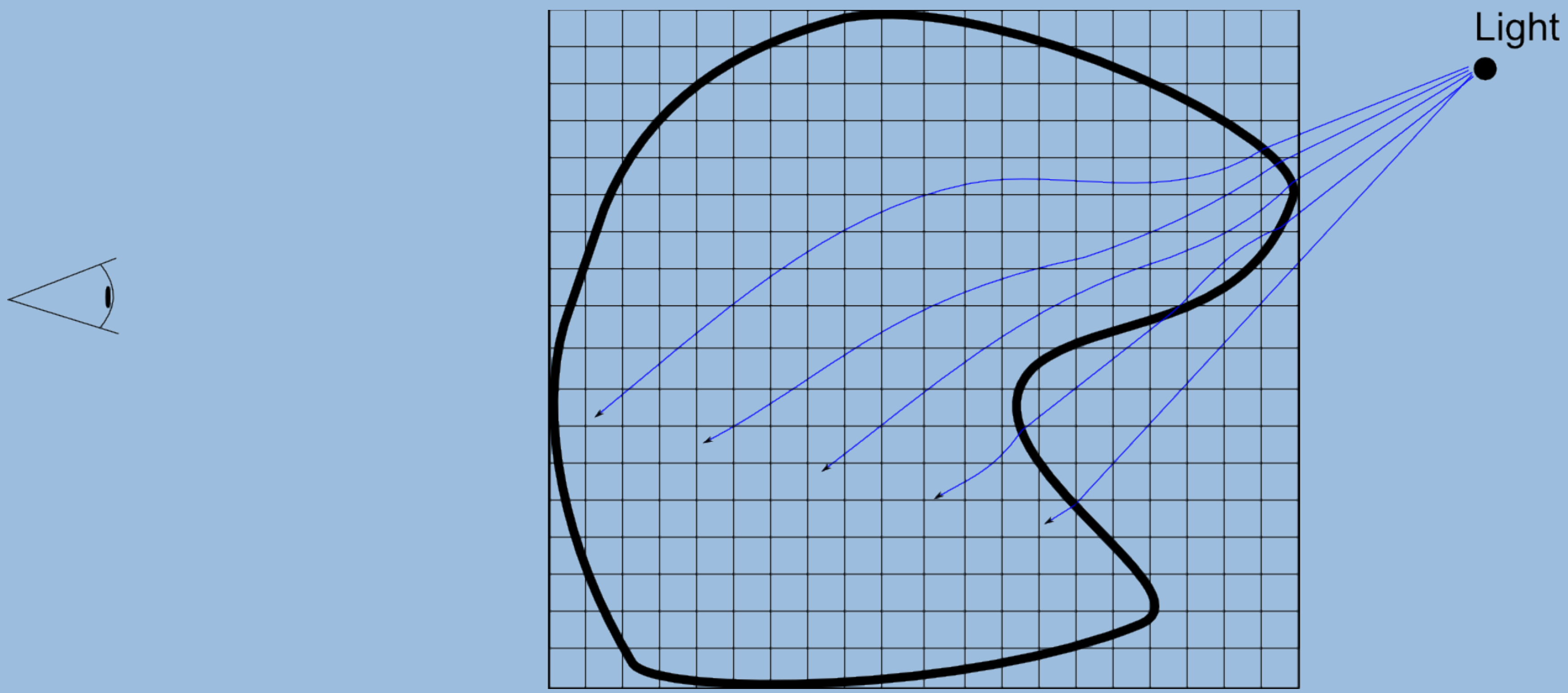
Rendering Pipeline (Overall Picture)

> Photon tracing with dynamic refraction indices



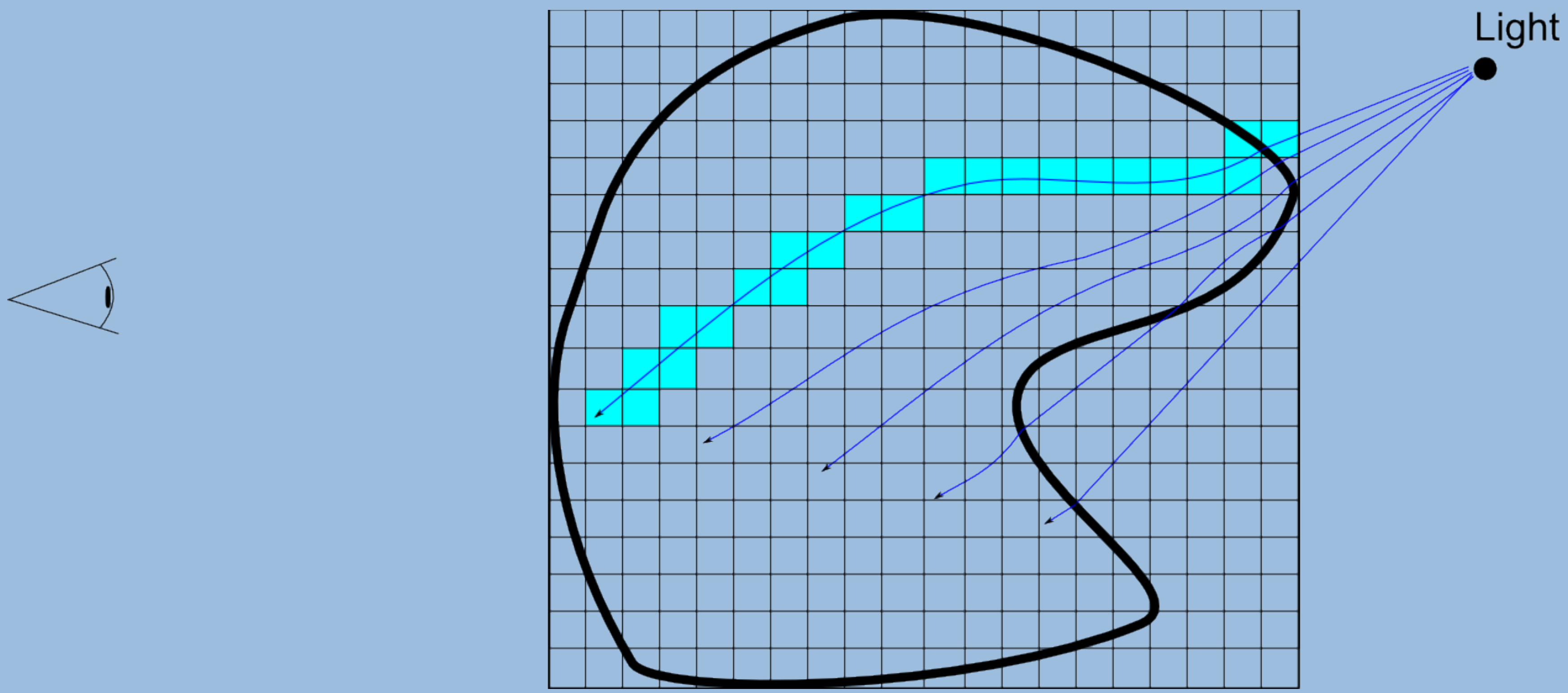
Rendering Pipeline (Overall Picture)

> Use raster to store refraction indices.



Rendering Pipeline (Overall Picture)

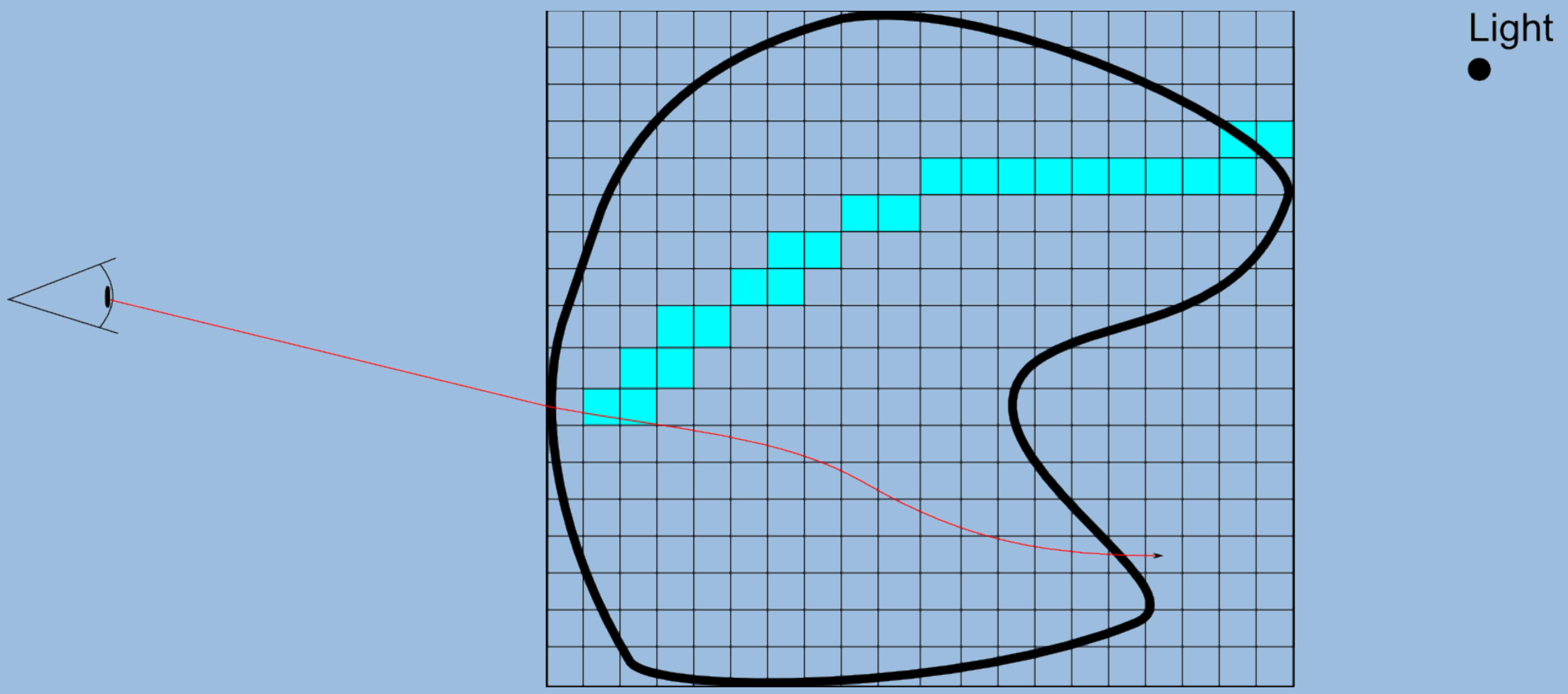
> Also, store radiance in raster.



Rendering Pipeline (Overall Picture)

UNIVERSITÄT BERN

> Compute pixel color by marching through raster.

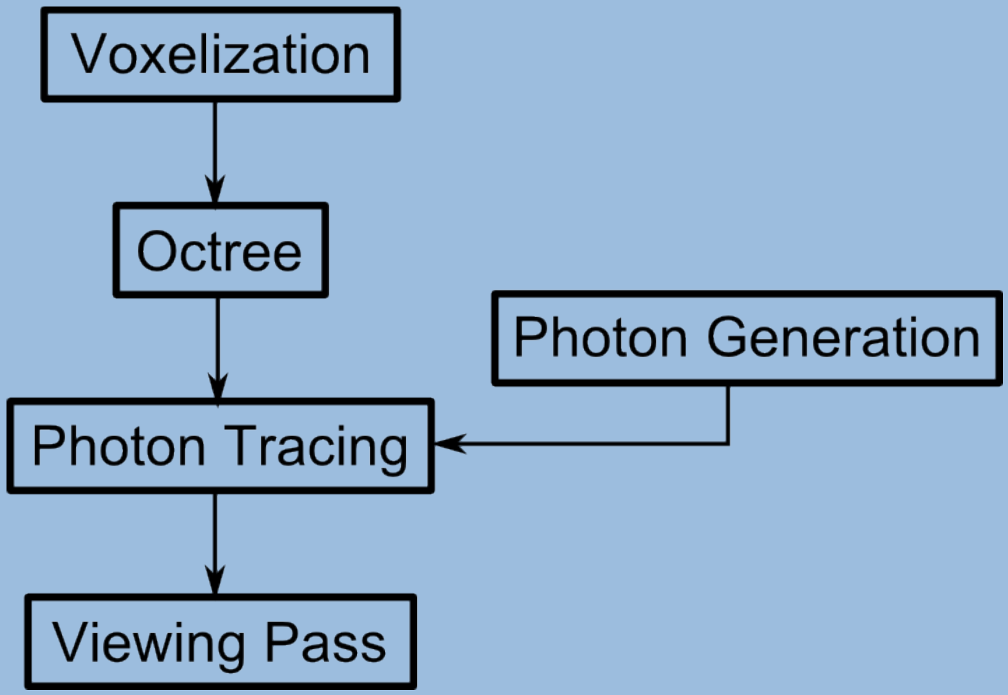


UNIVERSITÄT BERN

Contents

Rendering Pipeline Passes

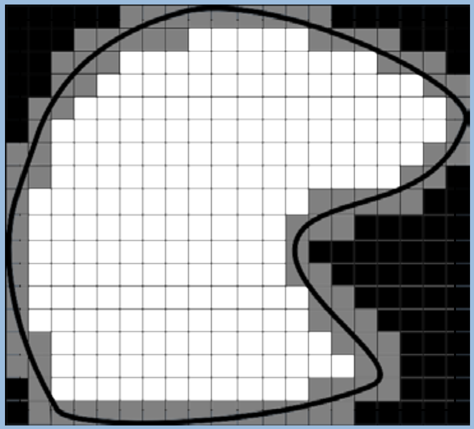
- Input: A watertight triangle mesh (OBJ Format)
- Output: Rendered image with caustics (within milliseconds)



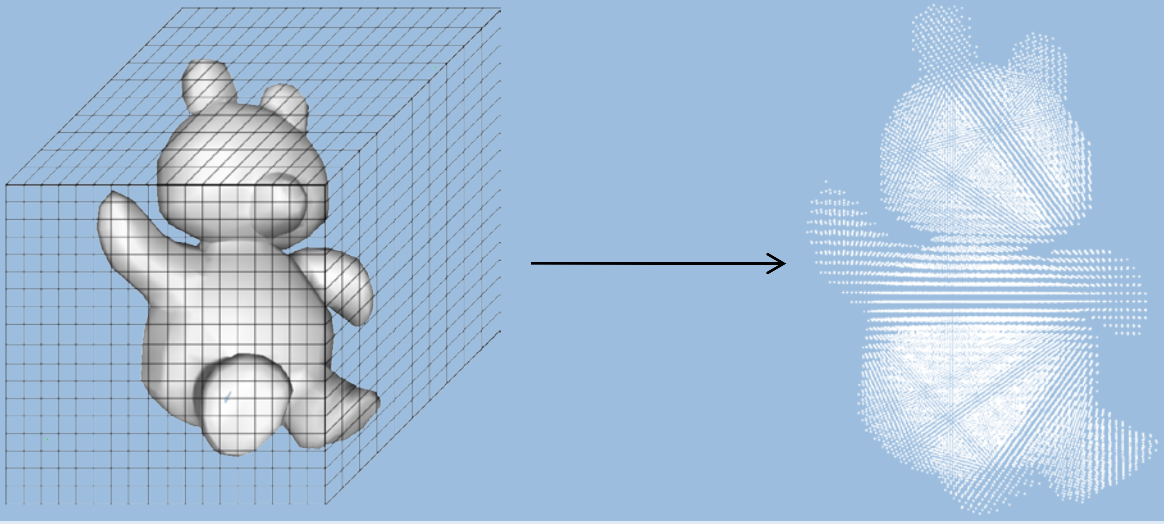
Contents

UNIVERSITÄT BERN

Voxelization



- 2D Graphics:
  - Smallest unit is a pixel.
  - Rasterization: Convert continuous geometry to pixel grid.
- 3D Graphics:
  - Smallest unit is a voxel (volume pixel, cube).
  - Voxelization: Convert continuous geometry to voxel grid.



u^b UNIVERSITÄT BERN

Voxelization (cont.)

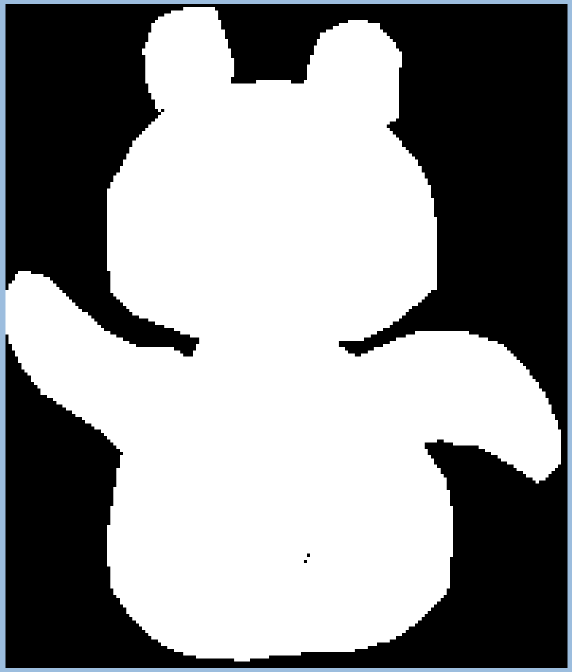
> Build voxel grid for refraction indices and radiance storage.

> Apply (Gauss) filtering for smoothing.

> Convert voxel grid to refraction indices.
  — White voxels = model refraction index
  — Black voxels = refraction index of surrounding material
  — Grey voxels = interpolated refraction index

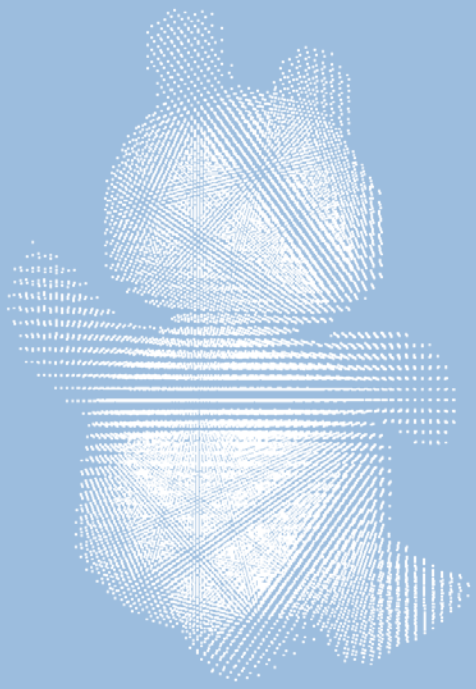
Voxelization (cont.)

Original Model

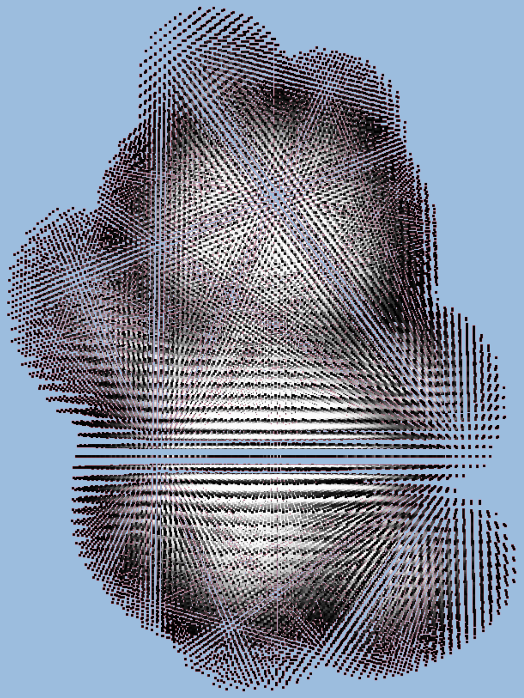


Single Slice

Voxelization (cont.)



Voxel Grid



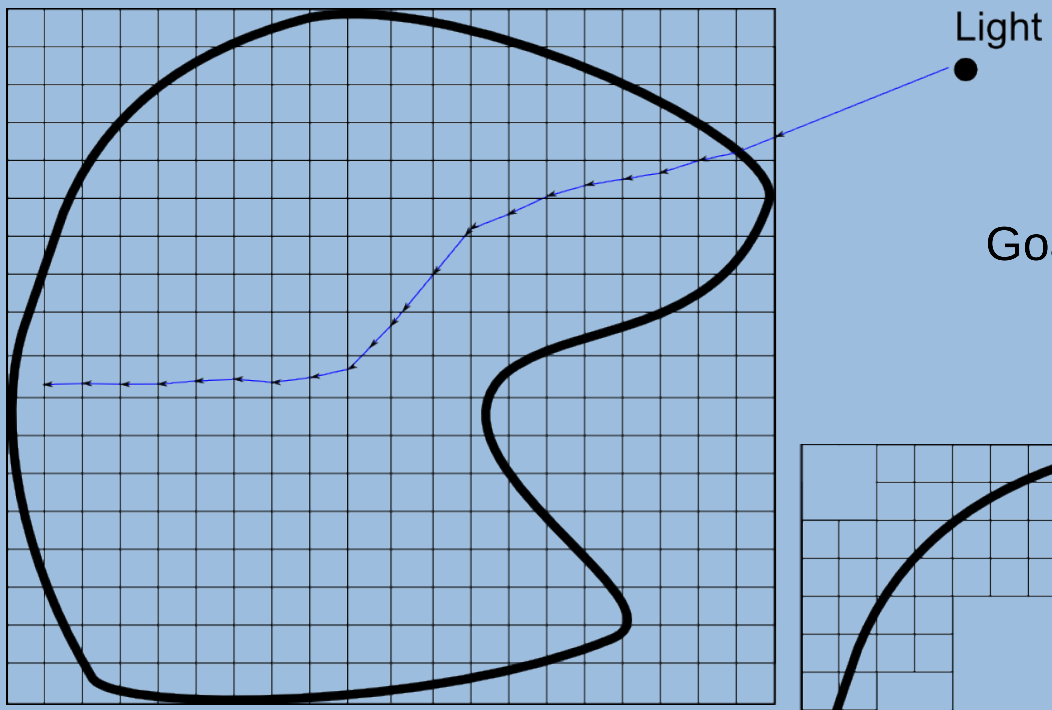
Final Voxel Grid
(after additional filtering)

Final voxel grid will then be transformed in refraction indices

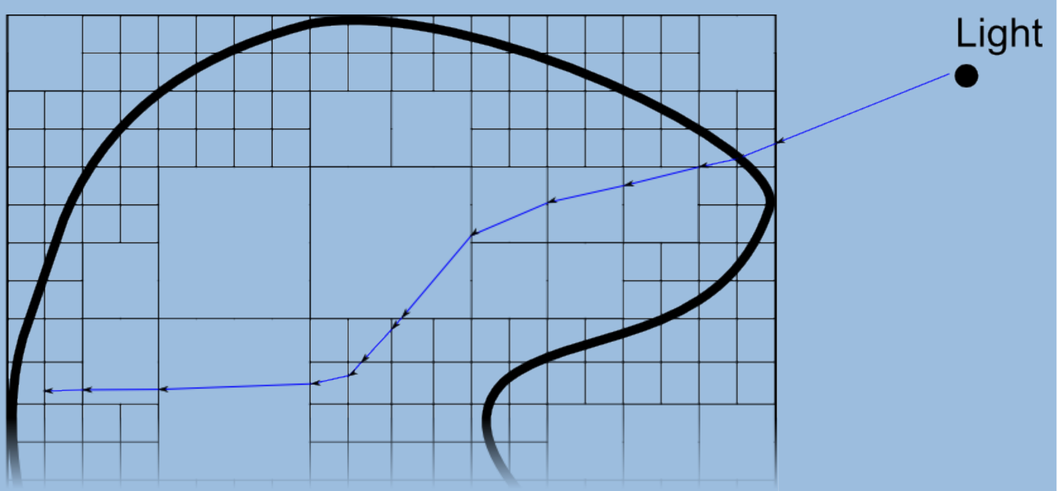
UNIVERSITÄT BERN

Contents

Octree

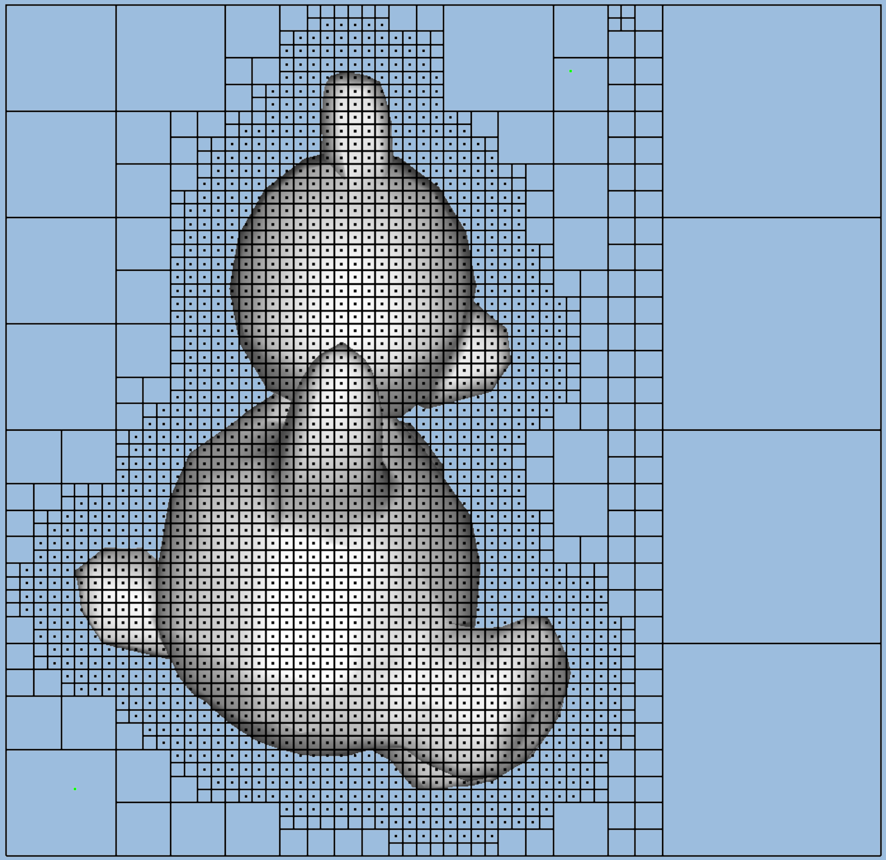


Goal: Accelerate Photon Tracing

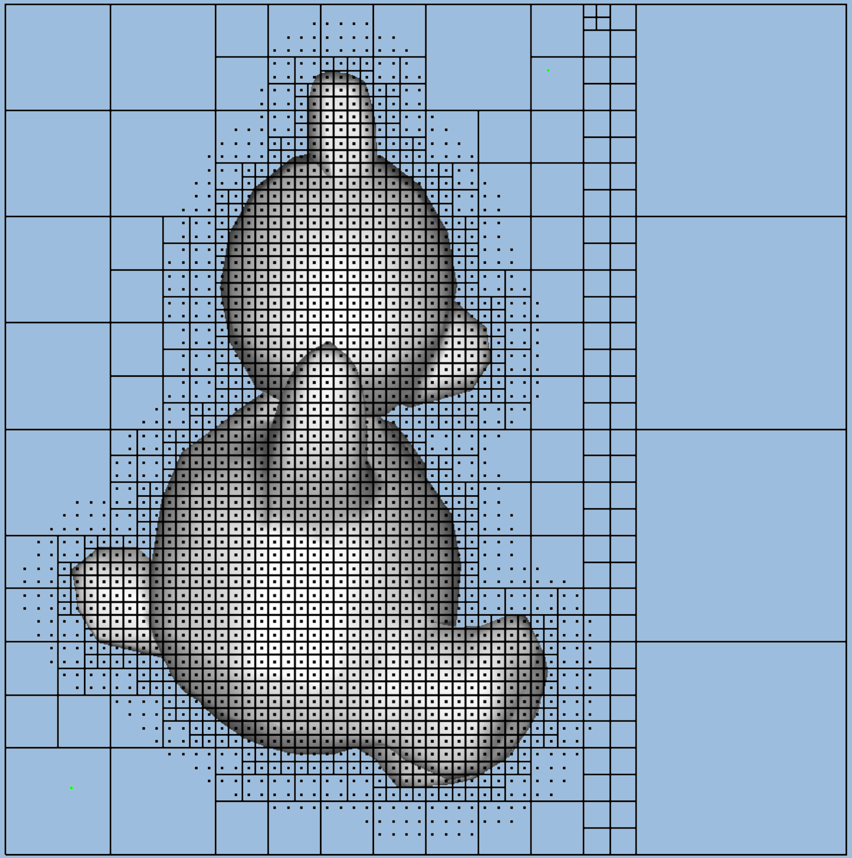


Octree Resolution

> User defined threshold ε.



$\varepsilon = 0.001$



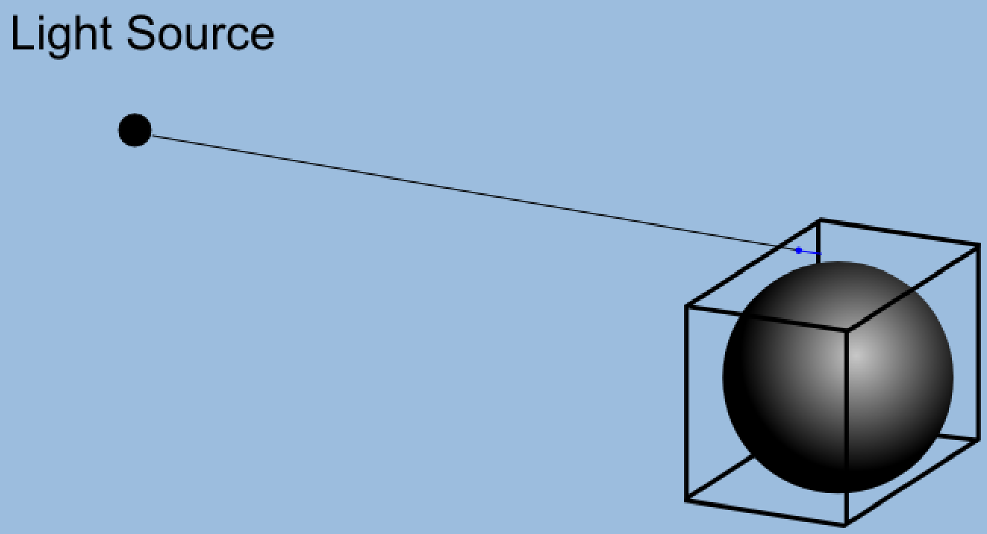
$\varepsilon = 0.1$

Contents

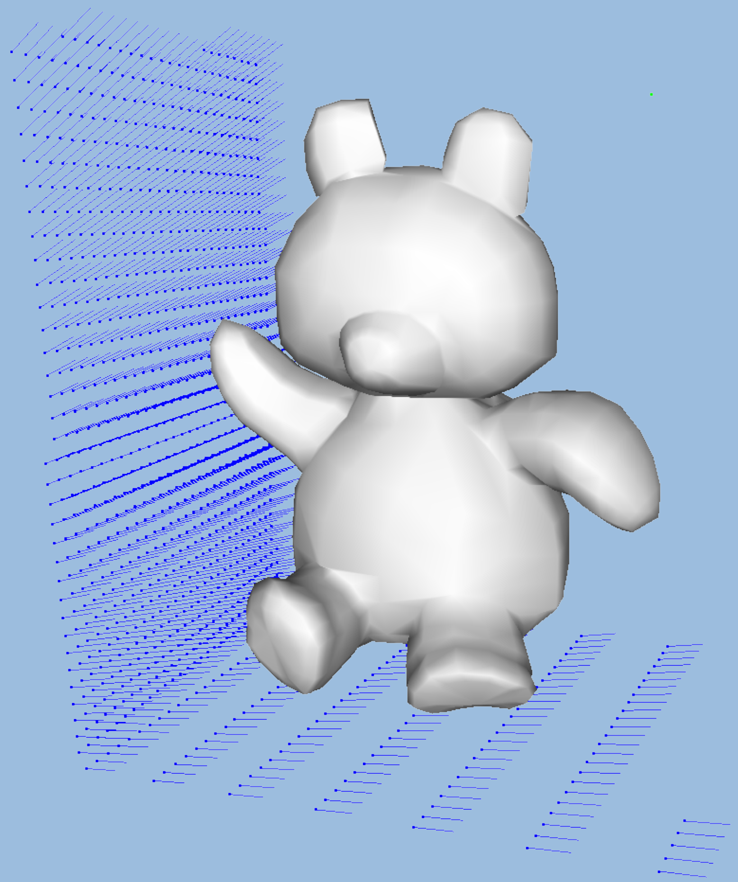
UNIVERSITÄT BERN

Photon Generation

- Goal: Determine initial position, direction and radiance of each photon.
- Use axis aligned bounding box to place the photons in the vicinity of the object.
- Place photons as close as possible to the object.
- Improvement: Use bounding geometry which fits the object better.



Photon Generation (cont.)



Photon from a point light

Contents

Photon Tracing

> Photons move through space/object according to the following equation:

$$\frac{d}{ds}\left(n\frac{dx}{ds}\right) = \nabla n$$

$$\xrightarrow{v = n\frac{dx}{ds}}$$

$$\frac{dx}{ds} = \frac{v}{n}$$

$$\frac{dv}{ds} = \nabla n$$

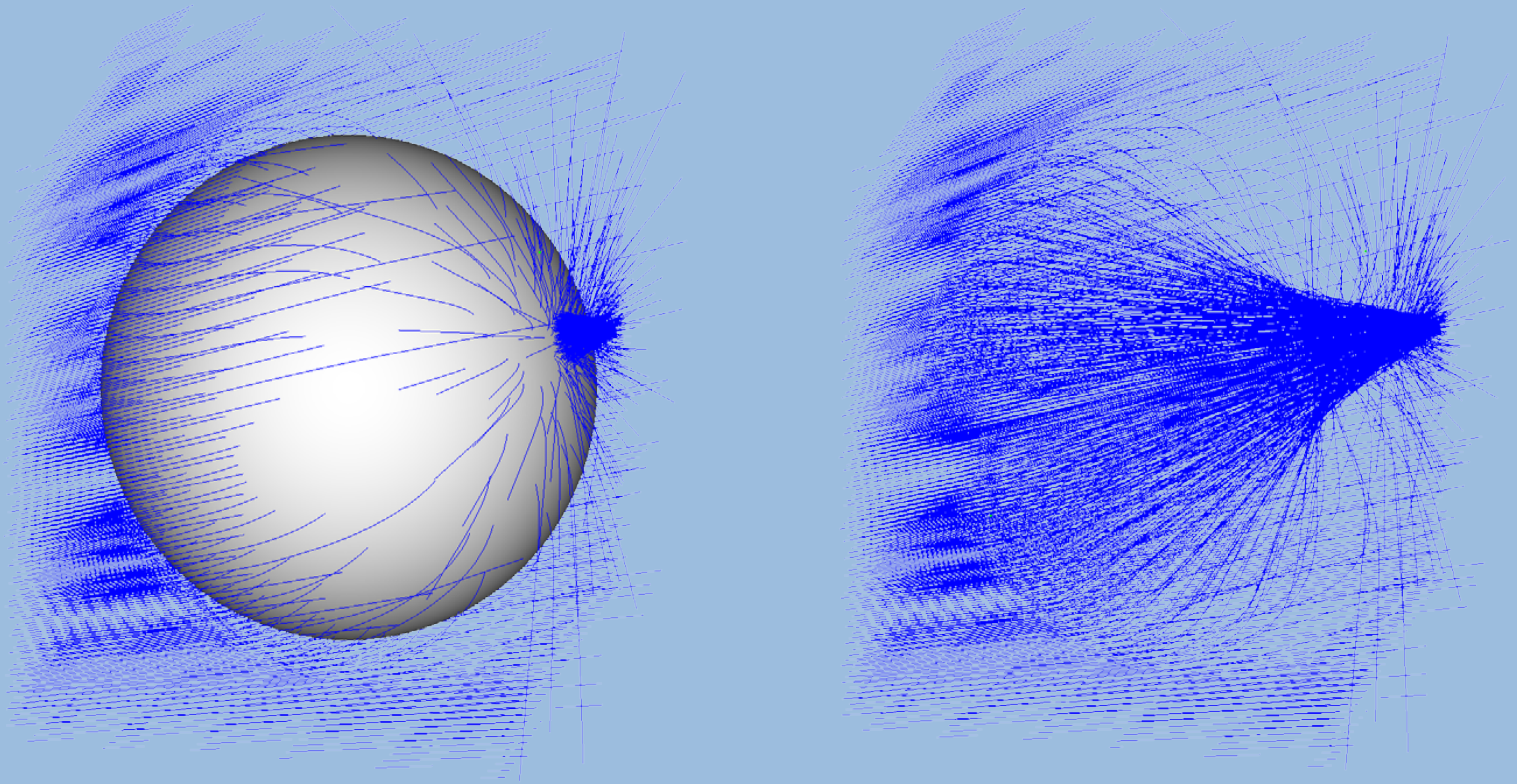
$$\xrightarrow{\text{discretize}}$$

$$x_{i+1} = x_i + \frac{\Delta s}{n} v_i$$

$$v_{i+1} = v_i + \Delta s \nabla n$$

> We use the octree to determine the largest step size Δs.

Photon Tracing (cont.)

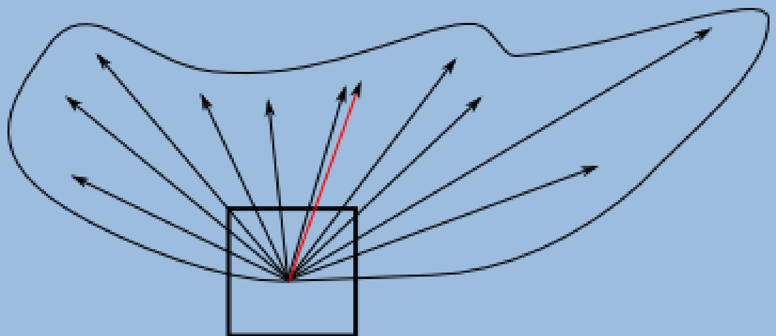


Photon Paths

Photon Tracing (cont.)

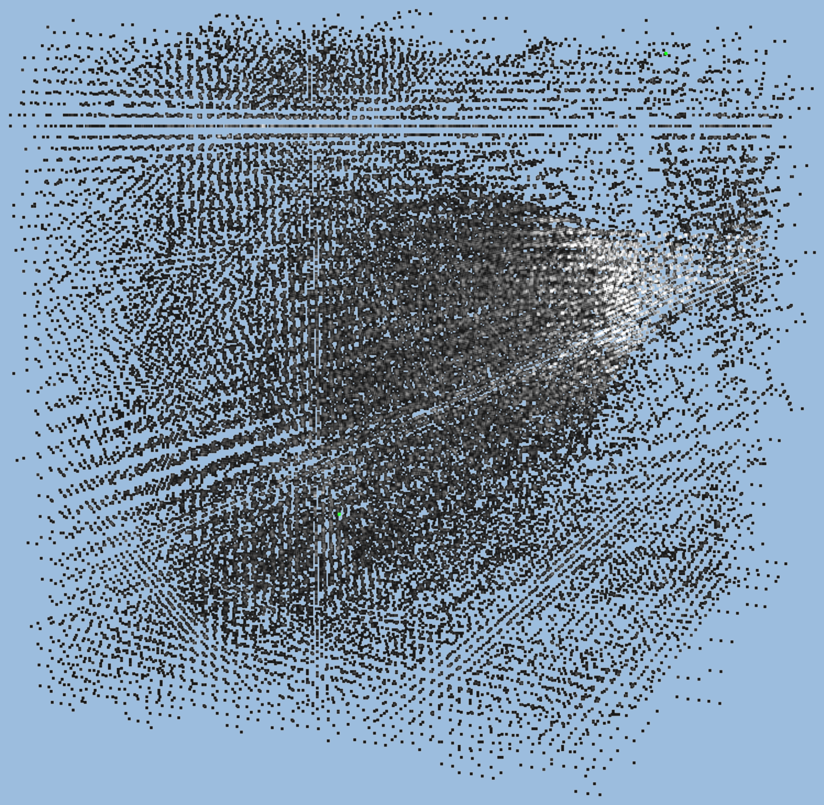
- When moving the photons through space/object each photon deposits a certain amount of light/energy in the voxels.

- Approximate *radiance distribution*
  - Average direction of photons
  - Average radiance value of photons



- This pass produces *volume caustics* because the radiance distribution is stored in the voxel grid.

Photon Tracing (cont.)



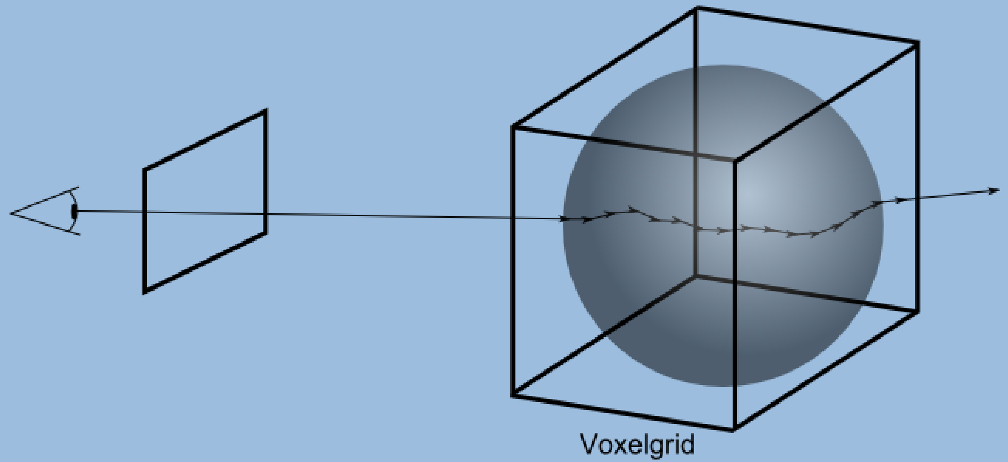
Radiances stored in voxels

Contents

Viewing Pass

- Idea:
  - Ray Tracing on the outside of the object
  - Ray Marching on the inside of the object

- Reminder: Ray Marching = Ray Tracing with fixed step size (here length of one voxel)



- To determine the path of the ray we use also the $\frac{d}{ds}(n\frac{dx}{ds}) = \nabla n$ equation.

Viewing Pass (cont.)

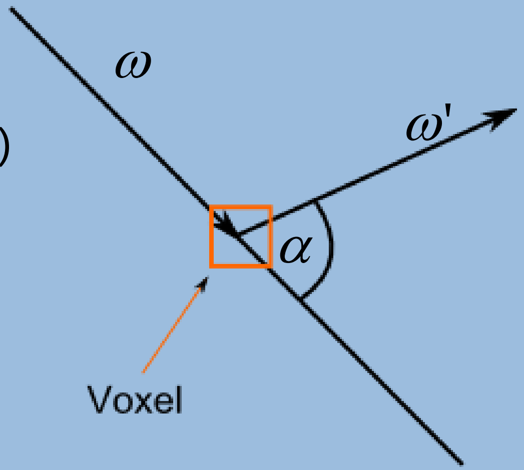
UNIVERSITÄT BERN

- The radiances from the photon tracing pass are used in each ray marching step to calculate the color value for the ray.

- To achieve *single scattering* we use the *scattering phase function* in each step.

  - Examples: $p(\omega \rightarrow \omega') = \frac{1}{4\pi}$ (Isotrop)

    $$p(\omega \rightarrow \omega') = \frac{1}{4\pi} \frac{1-k^2}{(1-k\cos(\alpha))^2}$$ (Schlick)



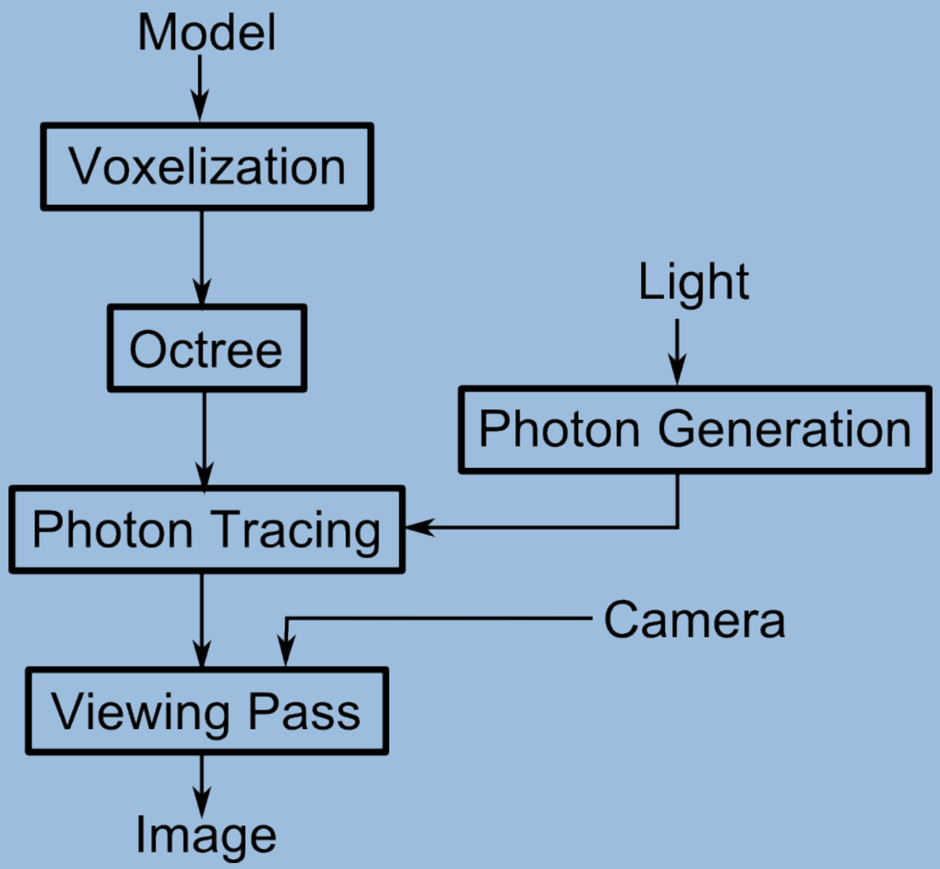
- Volume Caustics

UNIVERSITÄT BERN

Contents

Rendering Pipeline Recap

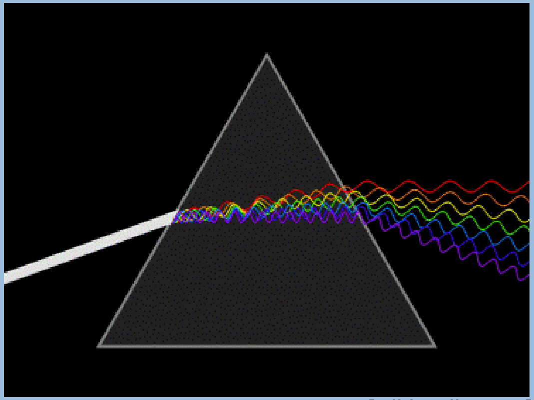
- Input: A watertight triangle mesh (OBJ Format)
- Output: Rendered image with caustics (within milliseconds)



Contents

Where to Go From Here

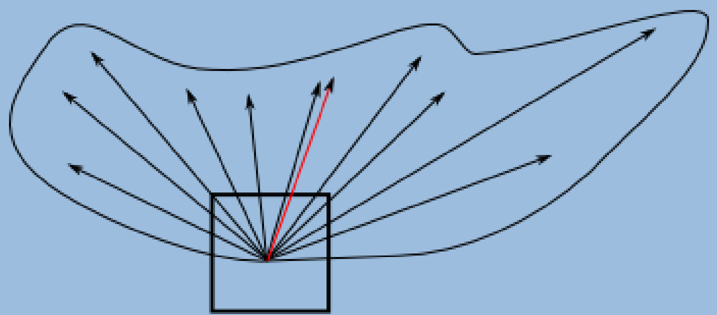
- Now: With refractions all wavelength bend with the same angle.
  - No prism effect
- Future work: Treat wavelength differently
  - Issue: Impossible to deal with all wavelength in real-ime



[wikipedia.com]

Where to Go From Here (cont.)

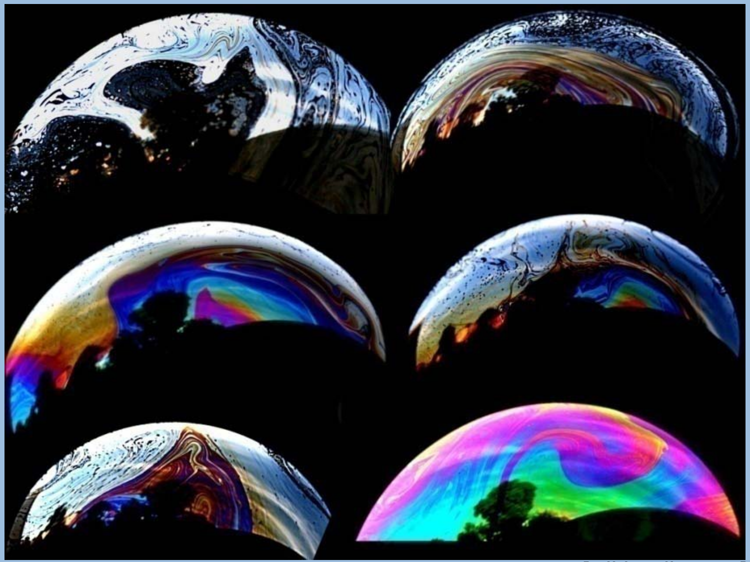
- Now: Radiance distribution is stored in each voxel as the average of all incoming directions.
  - Very coarse approximation



- Future work: Better approximation (e.g. Spherical harmonics)
  - Issue: Does the image quality benefit from that?

Where to Go From Here (cont.)

- Implementation of thin film interference
- Management of large scenes
- Investigation of possible performance enhancements



[wikipedia.com]

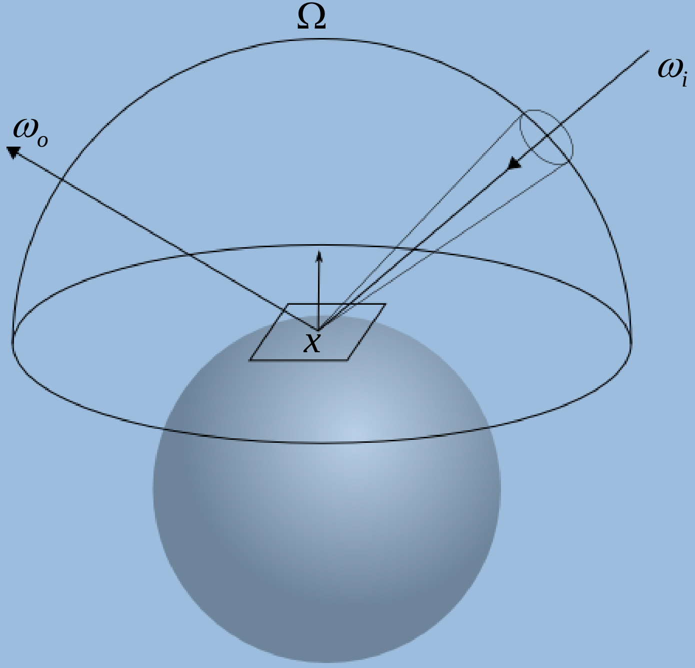
Thank you for your attention!

Questions?

Global Illumination (Rendering Equation)

- **Rendering Equation**:

$$L_o(x,\omega_o) = L_e(x,\omega_o) + \int_\Omega \underbrace{f(x,\omega_o,\omega_i)}_{\text{BRDF}} L_i(x,\omega_i)\,d\omega$$



- Assumption: Ray interacts only with surfaces.
- Evaluation only on surfaces:
  - No volumetric caustics
  - No scattering

Global Illumination (Volume Rendering Equation)

> **Volume Rendering Equation**:

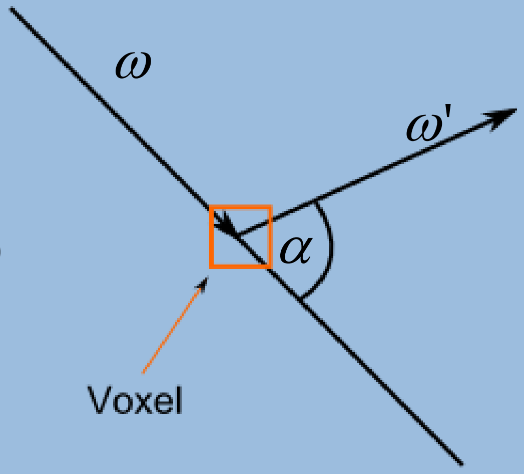
$$L_o(x,\omega_o) = \int_0^\infty \underbrace{e^{-\tau(s)}}_{\text{Extinction (Absorption, Out-Scattering)}} \left(\underbrace{\sigma_s(x+s\omega_i) \int_{S^2} p(\omega_i \to \omega_o) L(x+s\omega_i,\omega_i)\, d\omega_i}_{\text{In-Scattering}} + \underbrace{L_e(x+s\omega_i,\omega_o)}_{\text{Emission}} \right) ds$$

> $p(\omega_i \to \omega_o)$ is the *scattering phase function*

Global Illumination (Volume Rendering Equation)

- The *scattering phase function* specifies how light arriving at a voxel is scattered in a certain direction.
  - Examples: $p(\omega \to \omega') = \frac{1}{4\pi}$ (Isotrop)

    $p(\omega \to \omega') = \frac{1}{4\pi} \frac{1-k^2}{(1-k\cos(\alpha))^2}$ (Schlick)

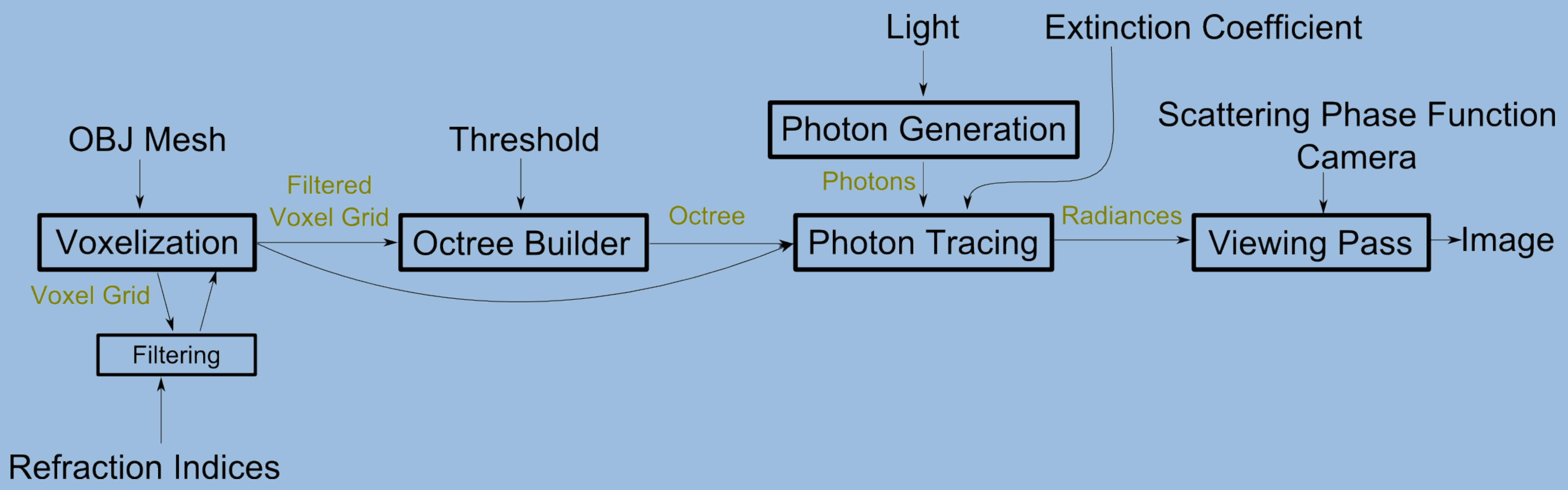


- Volume Caustics
- Full scattering

UNIVERSITÄT BERN

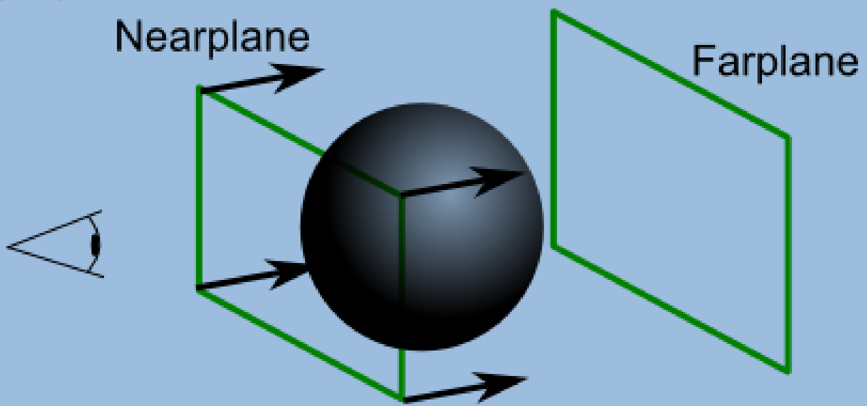
Rendering Pipeline Recap

- Input: A watertight triangle mesh (OBJ Format)
- Output: Rendered image with caustics (within milliseconds)



Voxelization (cont.)

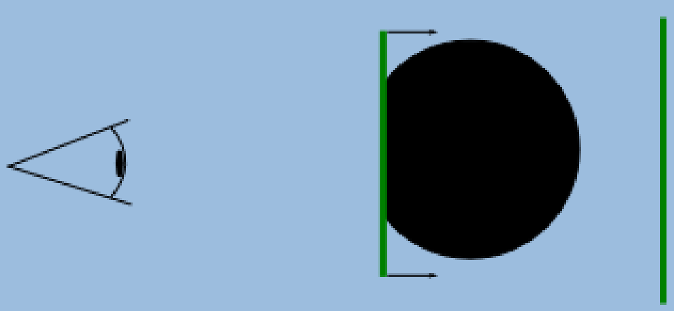
- Idea: Exploit Clipping Capability of OpenGL.
  We move the nearplane towards the farplane.



- Use coloring to distinguish inside from outside

- Apply (Gauss) filtering afterwards.

- Convert voxel grid to refraction indices.



Octree Construction

b
UNIVERSITÄT
BERN

> Goal: Accelerate photon tracing pass

> Idea: Group regions with approximately constant refraction indices

> In these regions the path of a photon changes only slightly.

> This allowes us to take larger steps for photon tracing.

In principle the octree should also respect the extinction coefficients. However, the visual impact of the extinction coefficients is hardly noticeable. (According to paper)

UNIVERSITÄT BERN

Photon Tracing (cont.)

- Problem: Hard to tell if paths are correct.