Computer Graphics Seminar

Real-time Rendering of Refractive Objects in Participating Media

Michael Pfeuti

Universität Bern

Contents

UNIVERSITÄT BERN

Contents

Basis

ACM SIGGRAPH 2008 Paper

Interactive Relighting
Of
Dynamic Refractive Objects

by

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

b UNIVERSITÄT BERN

Goal

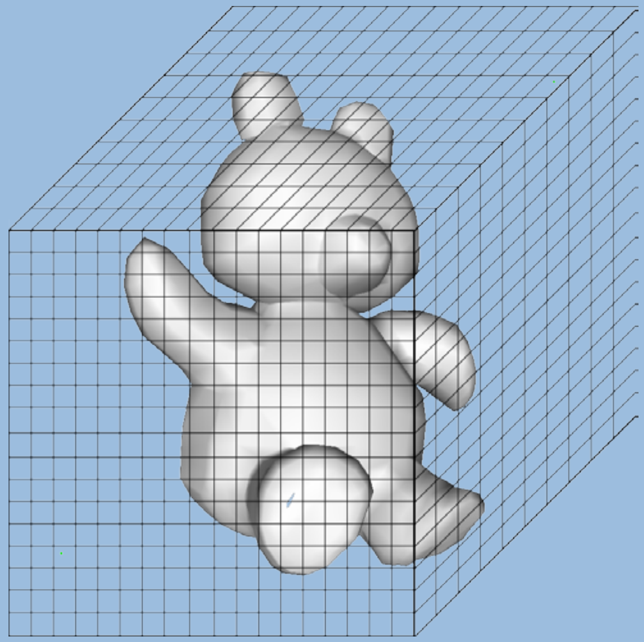
- Demo
- **Our goal**:
  - Reimplementation
  - Acceleration of pipeline

b UNIVERSITÄT BERN

Contents

Reference Pipeline

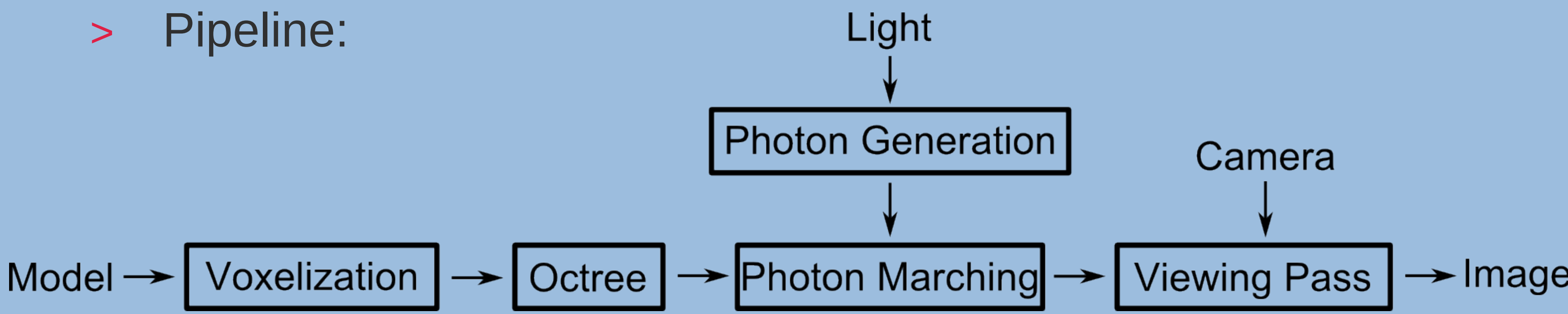
- Main idea:
  - Photon Marching
  - Store data in 3D arrays (voxel grids)
  - GPU implementation
- Data:
  - Scene (refraction indices/model, participating medium coefficients)
  - Illumination (radiance)
- Voxel grids cover participating media.



Voxel Grid

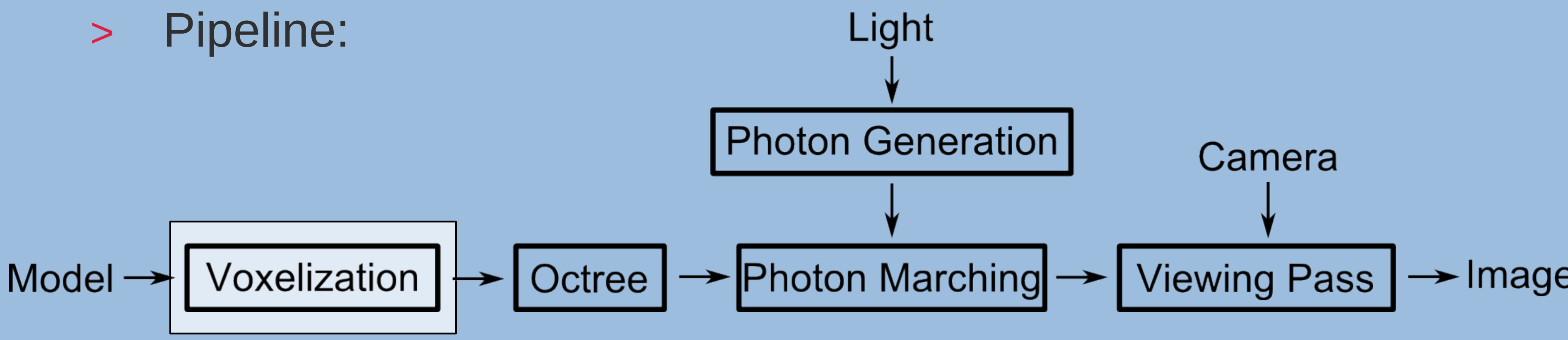
Reference Rendering Pipeline Passes

> Pipeline:



Reference Rendering Pipeline Passes

> Pipeline:



> Voxelization:

— Build voxel grids

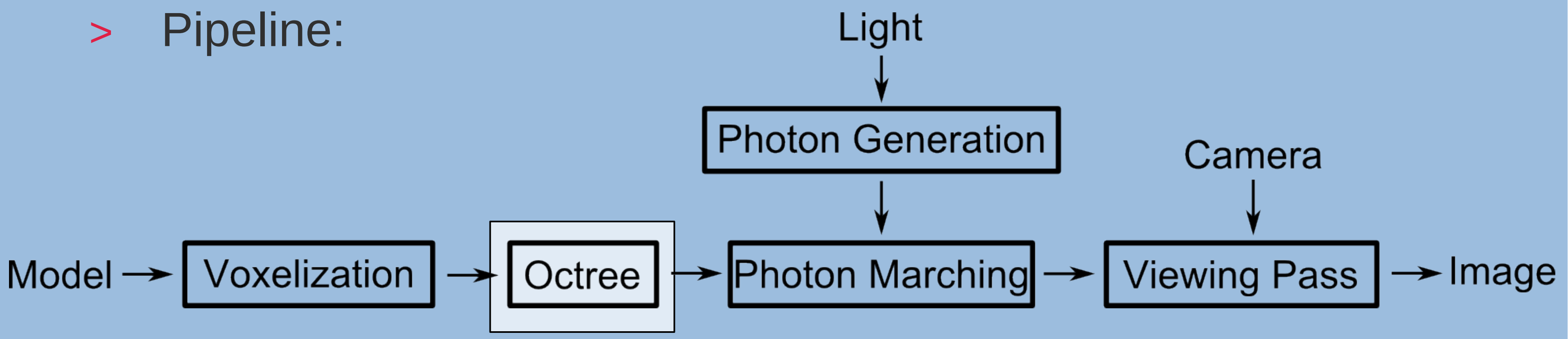
— Model represented through refraction indices

Model

Voxel Grid

Reference Rendering Pipeline Passes

> Pipeline:

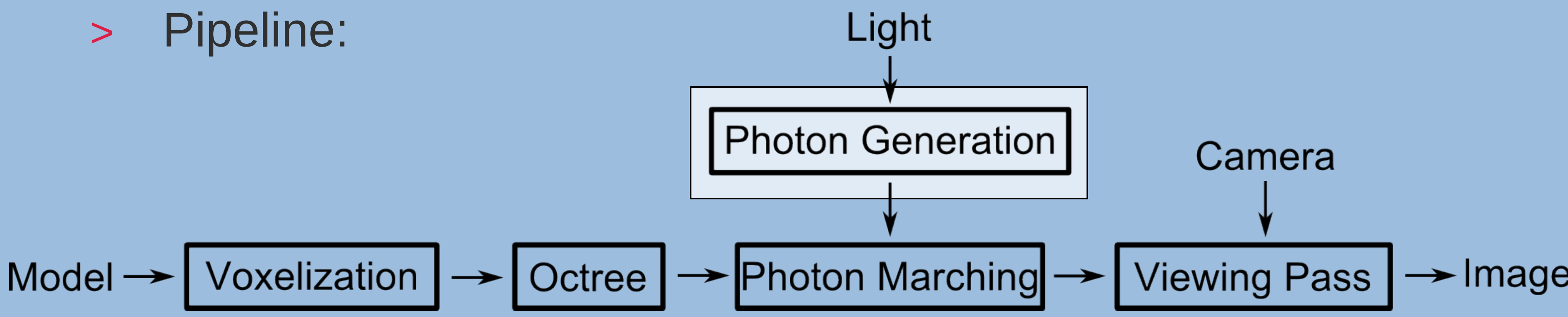


> Octree:
  — Acceleration structure for photon marching

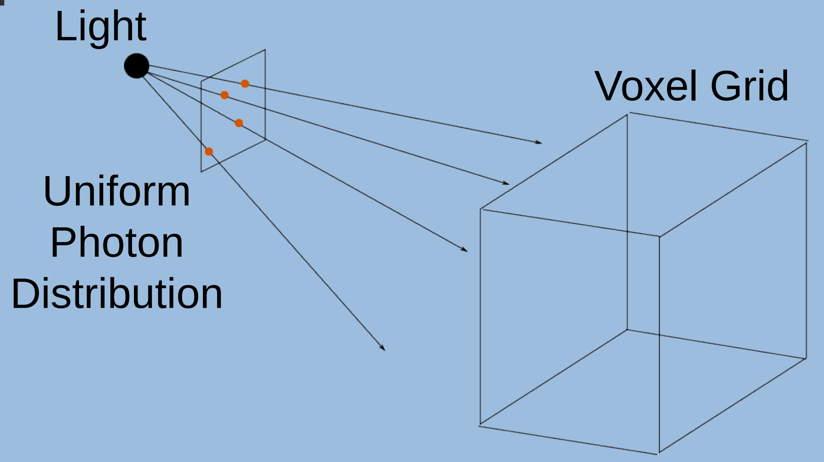
u^b UNIVERSITÄT BERN

Reference Rendering Pipeline Passes

> Pipeline:

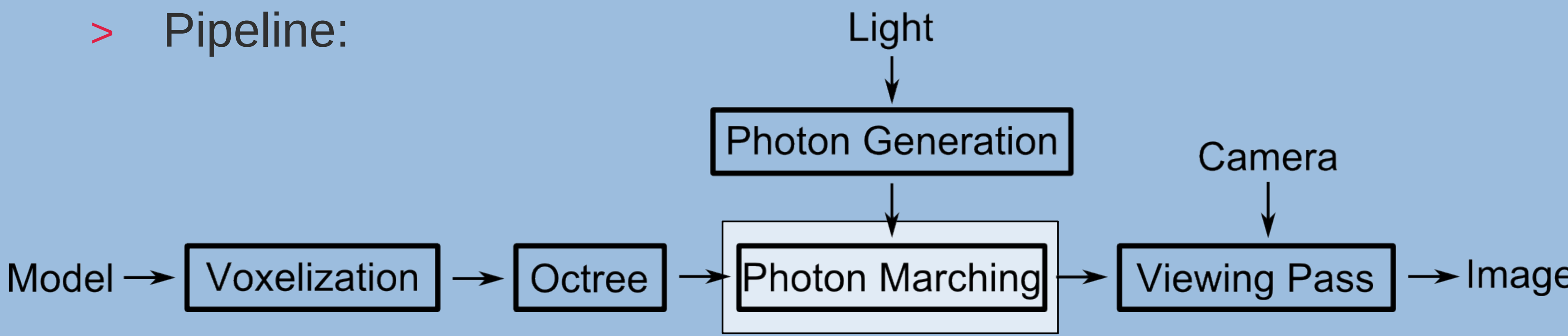


> Photon Generation:



Reference Rendering Pipeline Passes

> Pipeline:

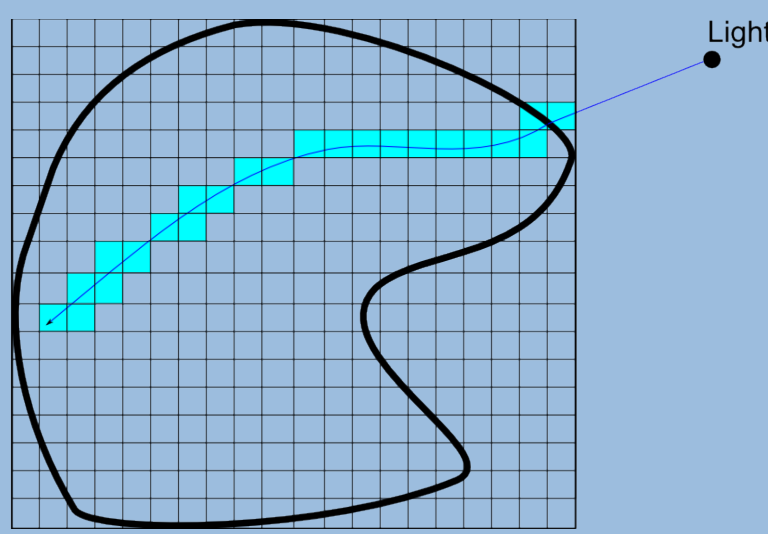


> Photon Marching:

— Move photons stepwise

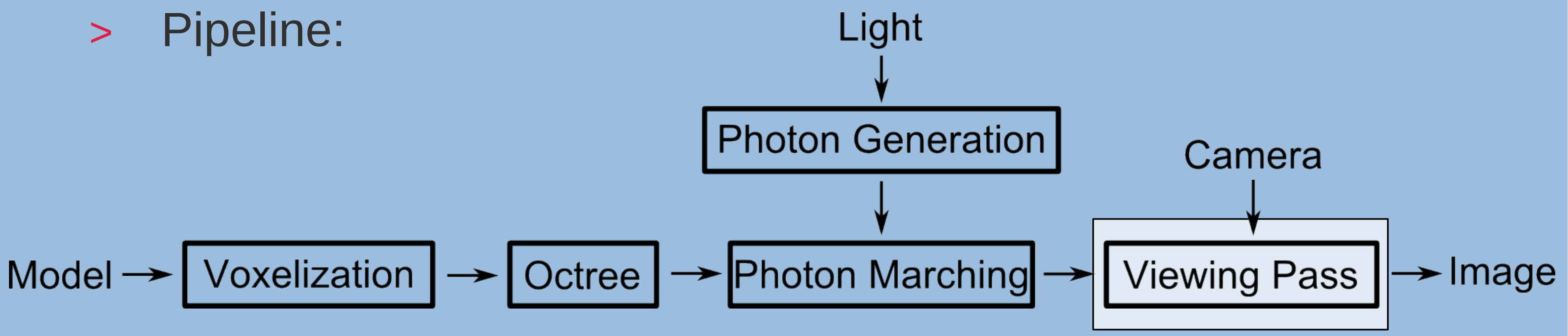
— Thread for each photon

⟶ atomic write

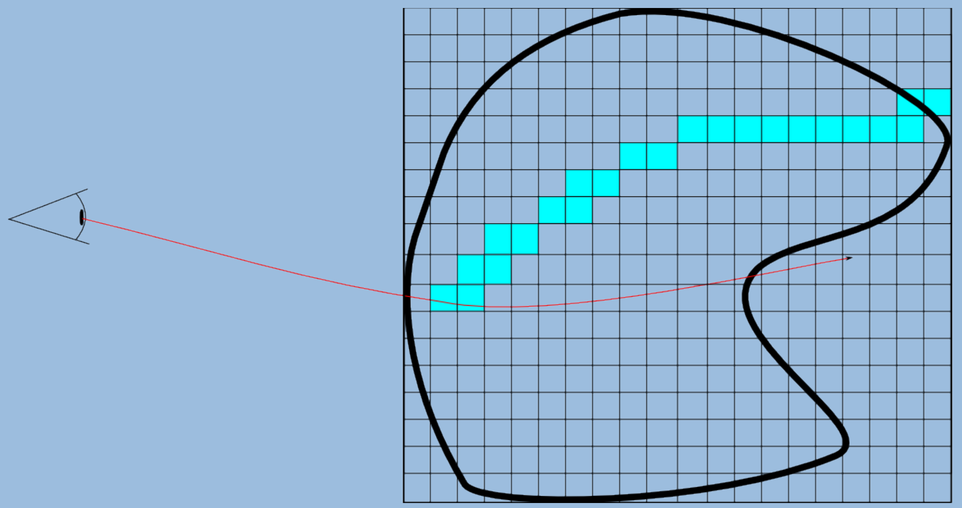


Reference Rendering Pipeline Passes

> Pipeline:



> Viewing Pass:
- — Ray marching
- — Radiance lookup after each step

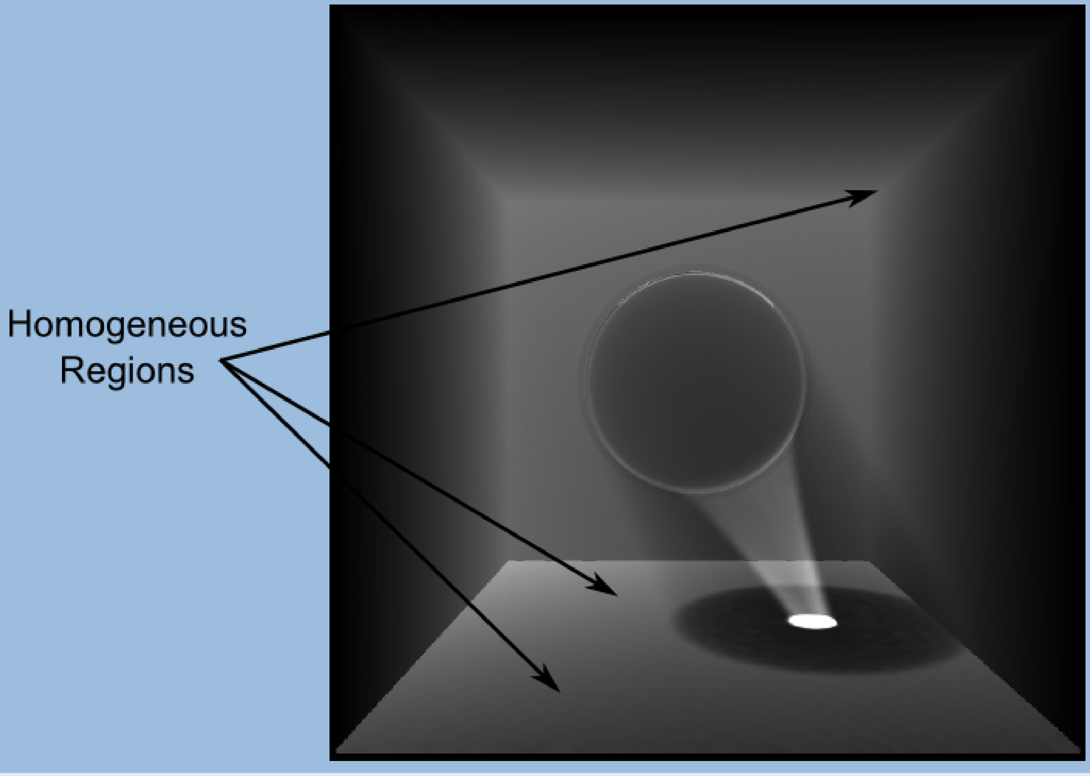


b UNIVERSITÄT BERN

Contents

Adaptive Rendering Pipeline

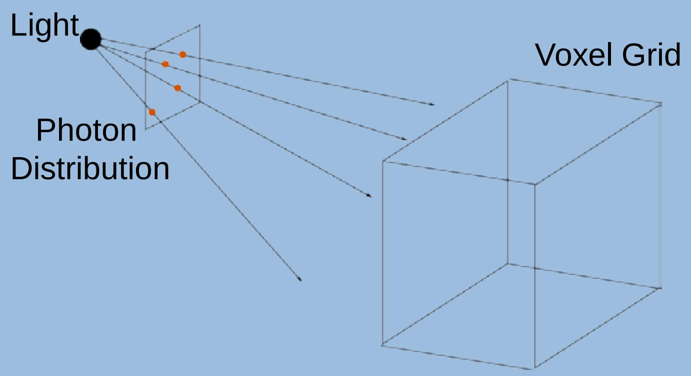
- Main idea: Reduce photons to accelerate pipeline.
- Shoot less photons through homogeneous regions.



Adaptive Rendering Pipeline

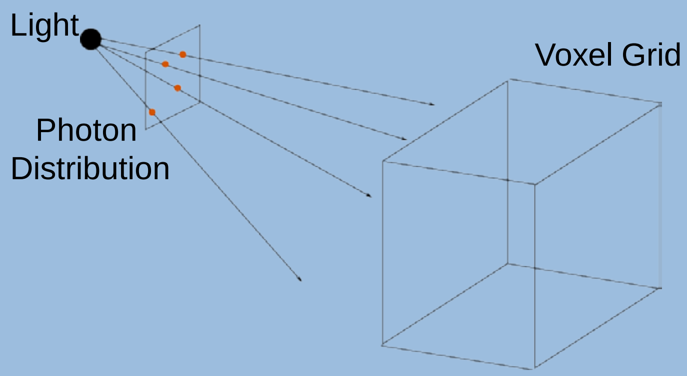
UNIVERSITÄT BERN

> This means distributing the photons **non-uniformly**.

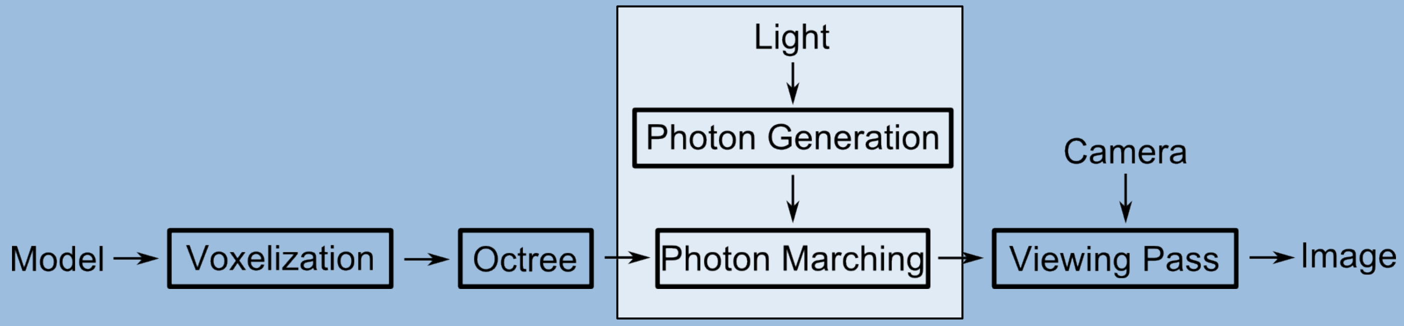


Adaptive Rendering Pipeline

> This means distributing the photons **non-uniformly**.

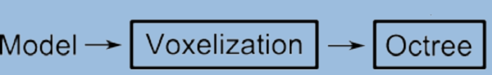


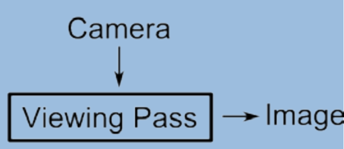
> We substitute the Photon Generation and Tracing pass.



Adaptive Rendering Pipeline

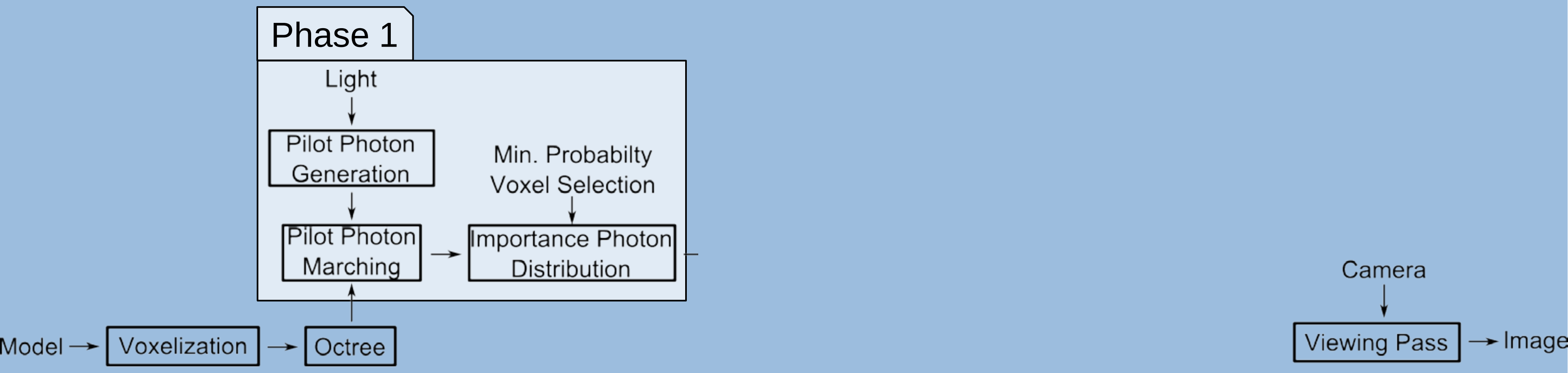
> Adaptive photon tracing





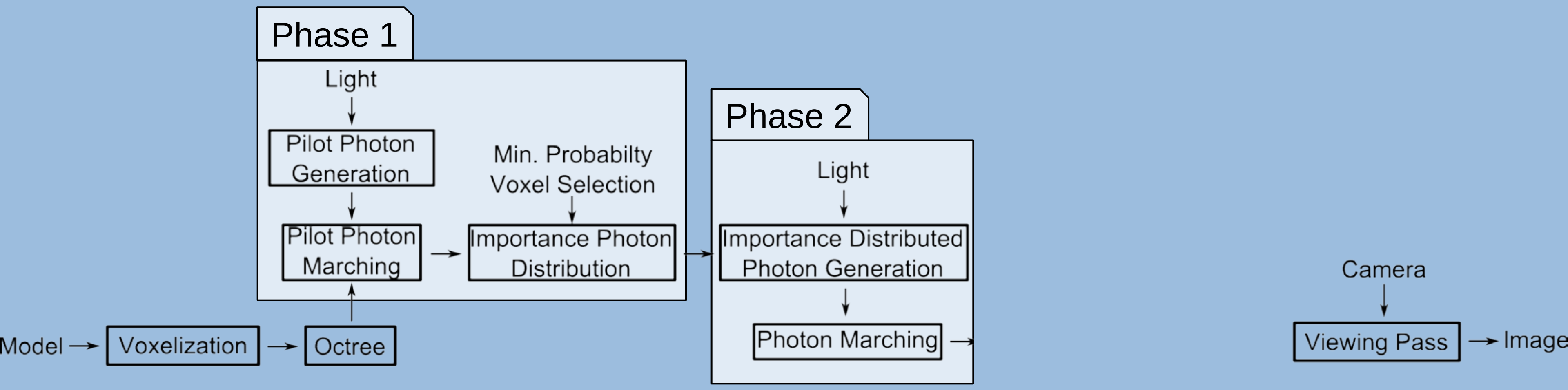
Adaptive Rendering Pipeline

> Adaptive photon tracing
  — Phase 1: Determine distribution



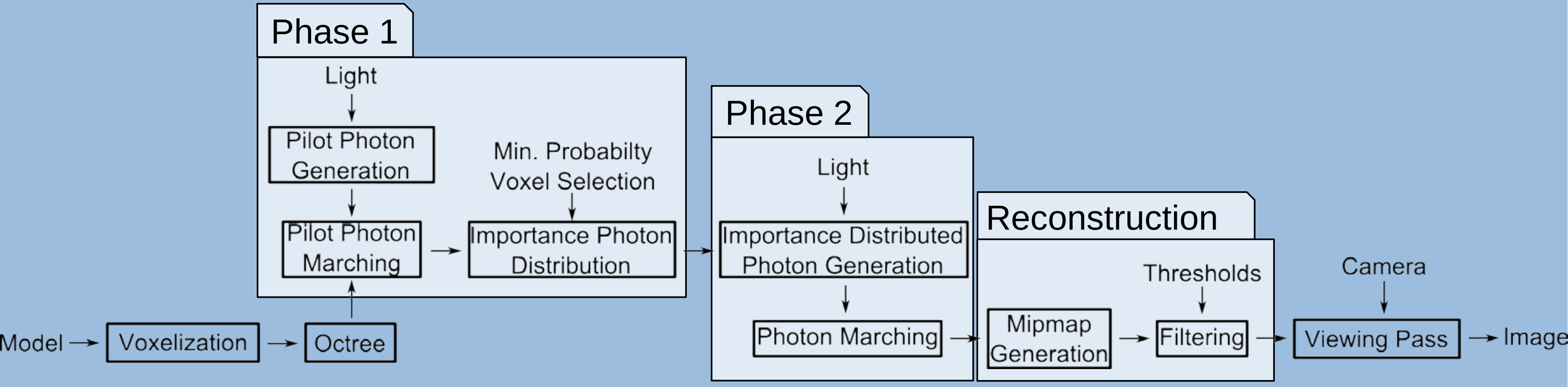
Adaptive Rendering Pipeline

> Adaptive photon tracing
  - Phase 1: Determine distribution
  - Phase 2: Compute illumination in voxel grid



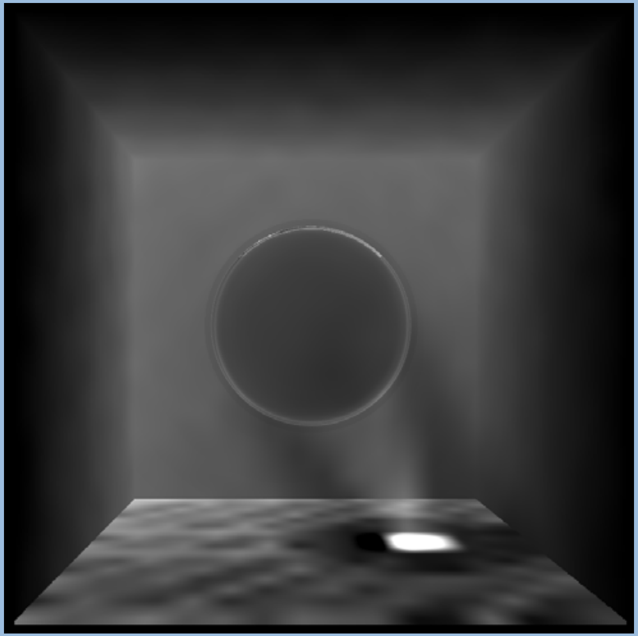
Adaptive Rendering Pipeline

> Adaptive photon tracing
  - Phase 1: Determine distribution
  - Phase 2: Compute illumination in voxel grid
  - Reconstruction: Compute a smooth radiance distribution



Phase 1

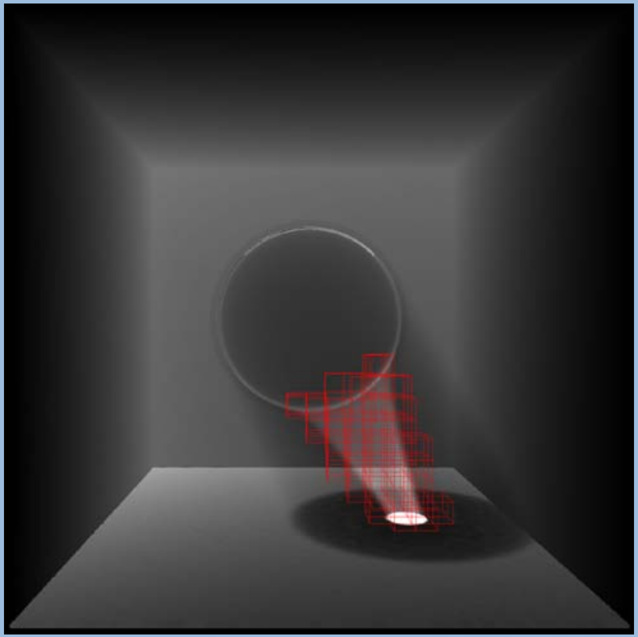
- Find in-/homogeneous regions.
- Shoot few “pilot” photons (~4000 photons)
- Select voxels for tight sampling



Pilot photon result

Phase 1

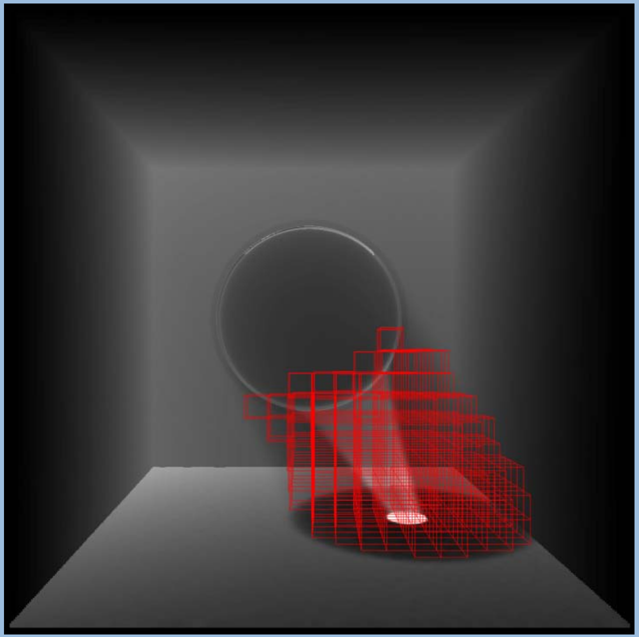
- Find in-/homogeneous regions.
- Shoot few “pilot” photons (~4000 photons)
- Select voxels for tight sampling
  - Gradient threshold for selecting voxel with larger gradients



Selection in reference image

Phase 1

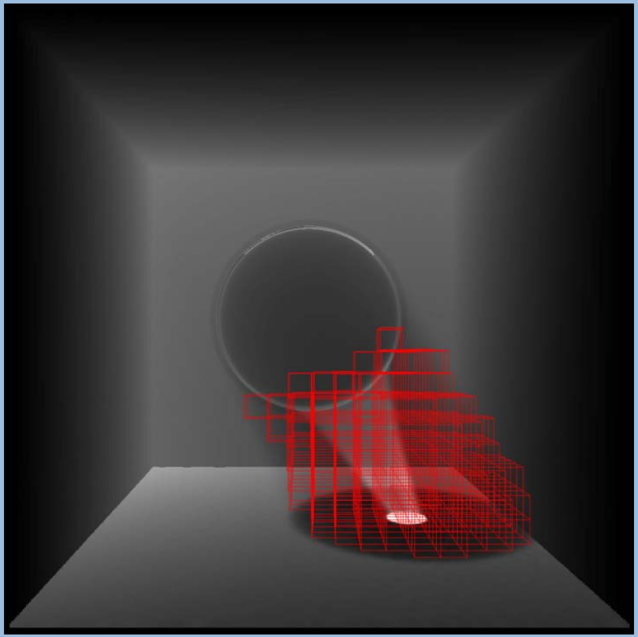
- Find in-/homogeneous regions.
- Shoot few “pilot” photons (~4000 photons)
- Select voxels for tight sampling
  - Gradient threshold for selecting voxel with larger gradients
  - Luminance threshold for selecting voxels in shadow



Selection in reference image

Phase 1

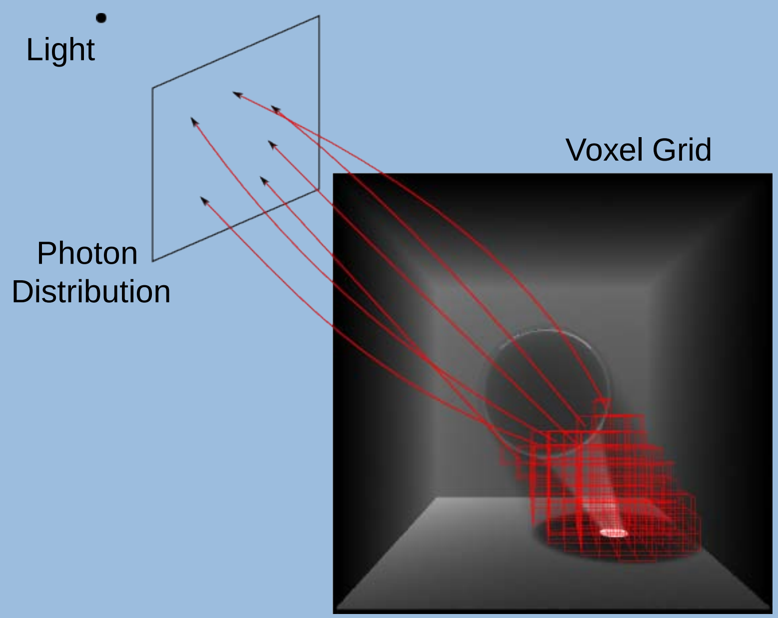
- Find in-/homogeneous regions.
- Shoot few “pilot” photons (~4000 photons)
- Select voxels for tight sampling
  - Gradient threshold for selecting voxel with larger gradients
  - Luminance threshold for selecting voxels in shadow
- Pilot Photon Marching:
  - Store **origin data** per voxel



Selection in reference image

Phase 1 (cont.)

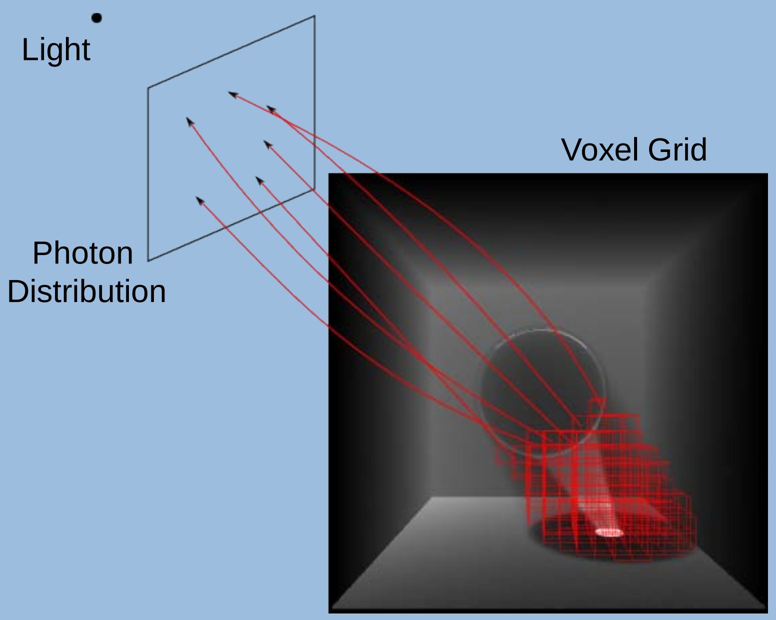
> Compute probability density function for photon distribution.



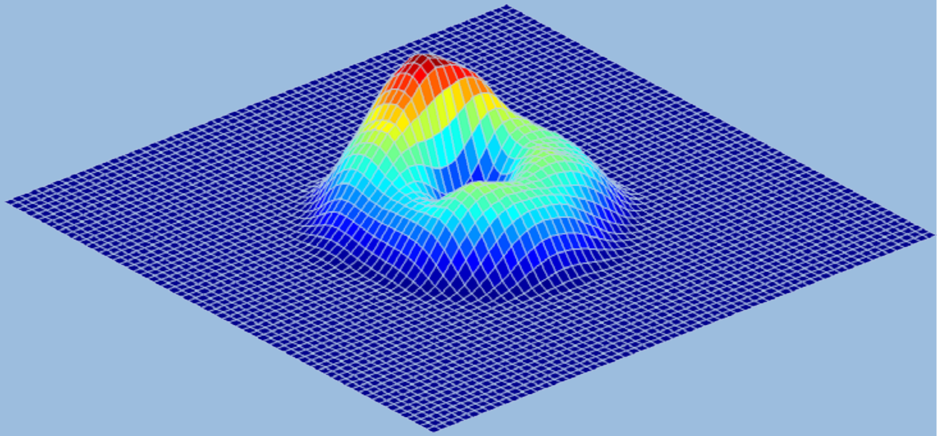
Origin Data for each Voxel

Phase 1 (cont.)

> Compute probability density function for photon distribution.



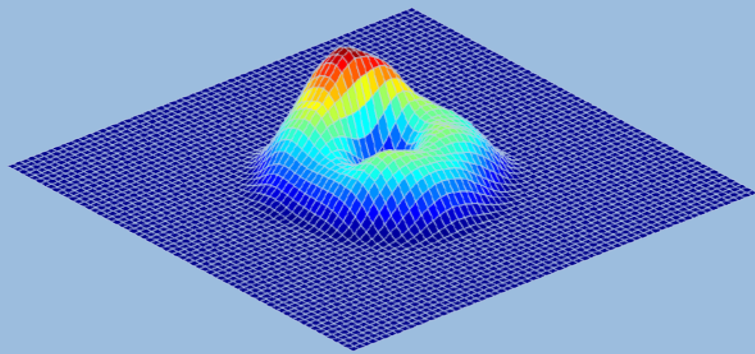
Origin Data for each Voxel



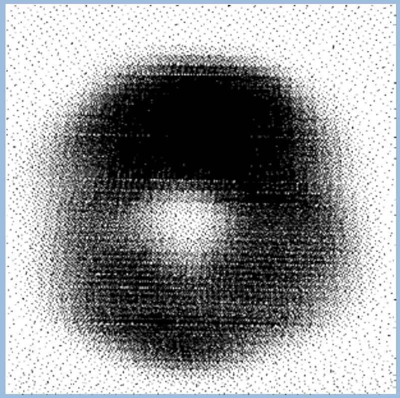
2D PDF
for photon sampling

Phase 2

- Photon generation according to the photon distribution from Phase 1



2D PDF

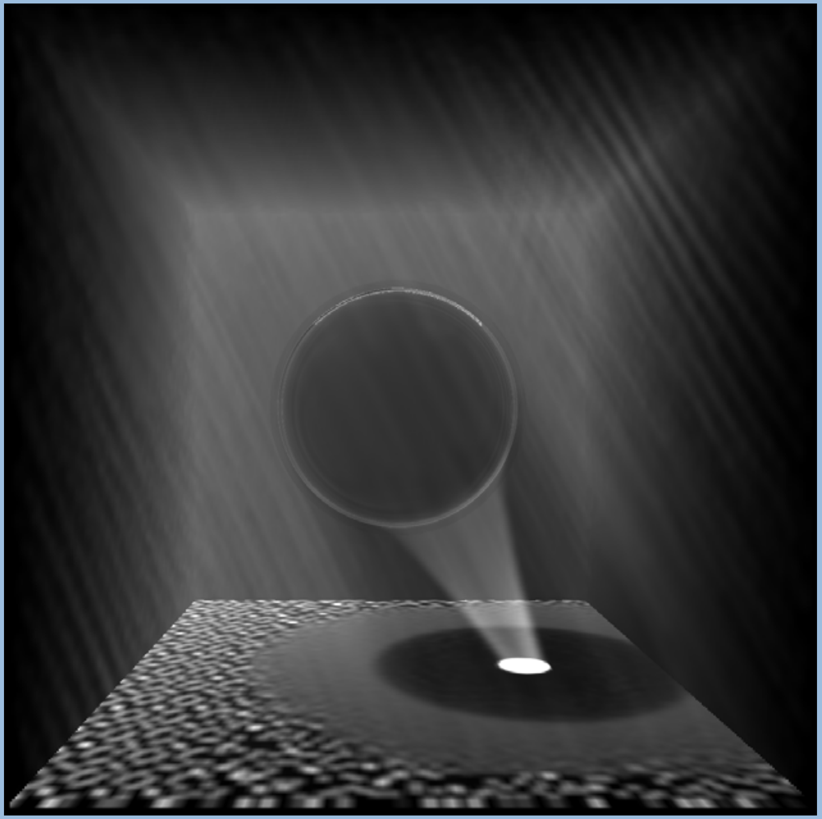


Distributed Photons

- Photon marching as in reference except:
  - Collect photon count per voxel
  - Collect average photon weight per voxel (from Monte-Carlo Integration)

Reconstruction

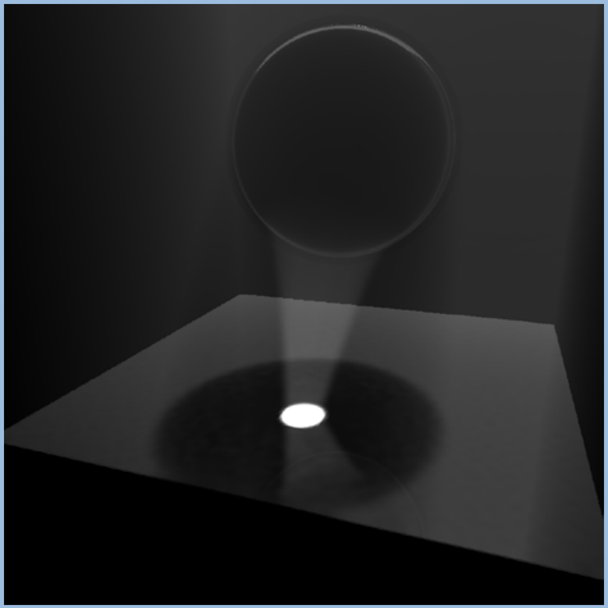
- Radiance voxel grid needs filtering.
- Tightly sampled regions
  - Hardly any filtering necessary
- Sparsely sampled regions
  - Much filtering needed

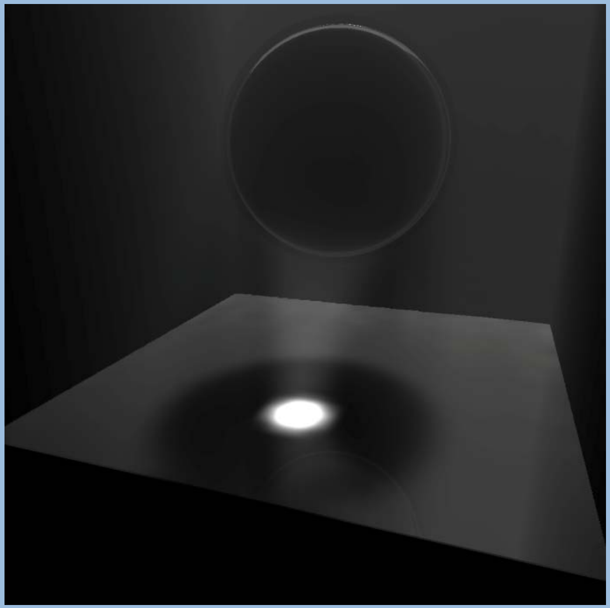


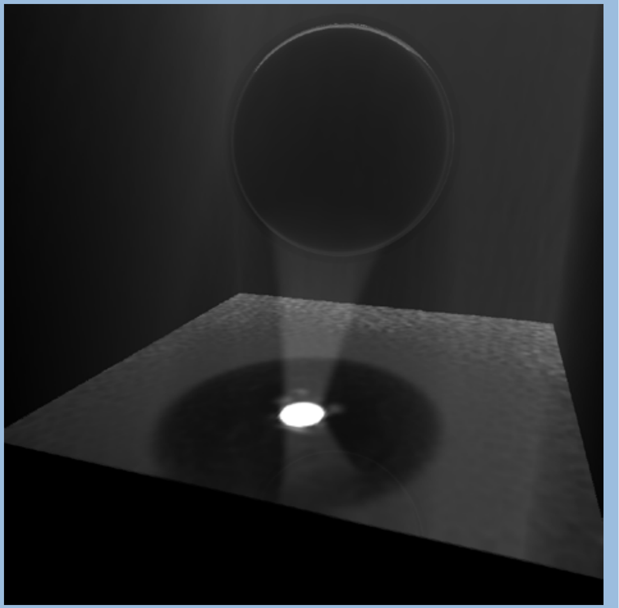
Noise in Voxel Grid

Reconstruction

> Simple approaches do not work (constant filter radius, k-NN).


Reference


Constant Radius


1-NN

Advanced Filtering

UNIVERSITÄT BERN

> We built a new filter:
  — Based on average photon weight
  — Center-surround filter

UNIVERSITÄT BERN

Advanced Filtering

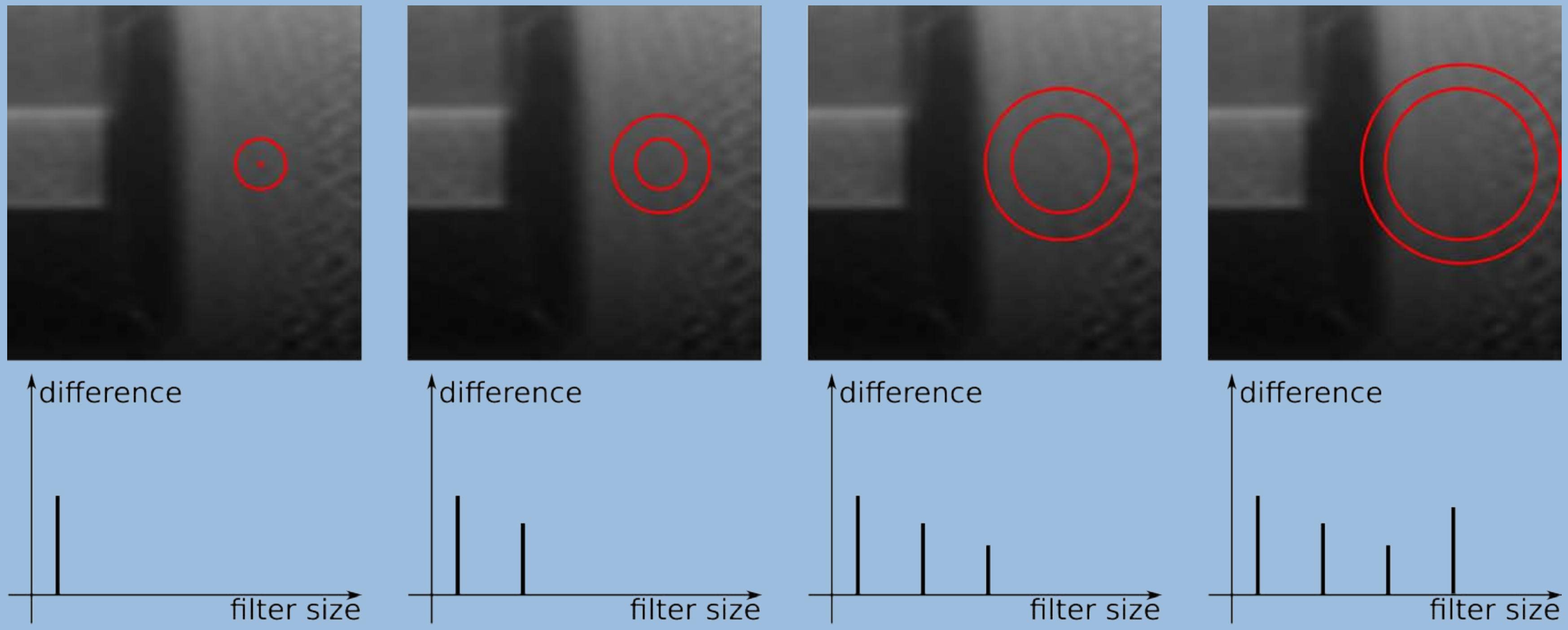
- We built a new filter:
  - Based on average photon weight
  - Center-surround filter

- Photon Weight Filter:
  - When photon weight
    - Large $\longrightarrow$ Sparse sampled region $\longrightarrow$ large filter radius
    - Small $\longrightarrow$ Tight sampled region $\longrightarrow$ small filter radius
  - Linear scaling of weight determines filter radius

$$radius = \alpha \cdot photon\_weight$$

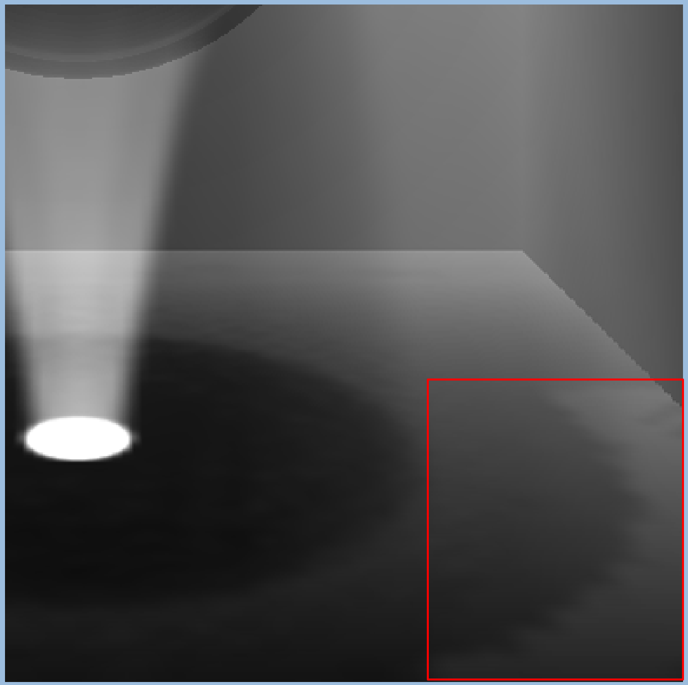
Center-Surround Filter

- Compares two filter radii
  - Increase radius as long as difference decreases
  - Stop at increase

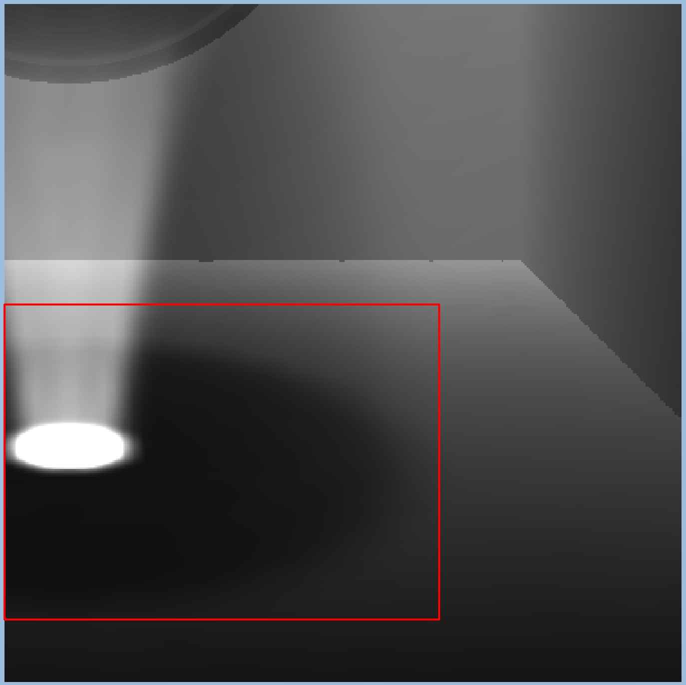


Combination

> Neither on its own is sufficient
  - Photon-Weight: Too large radii in sparse sampled regions
  - Center-Surround: Too large radii in tight sampled regions



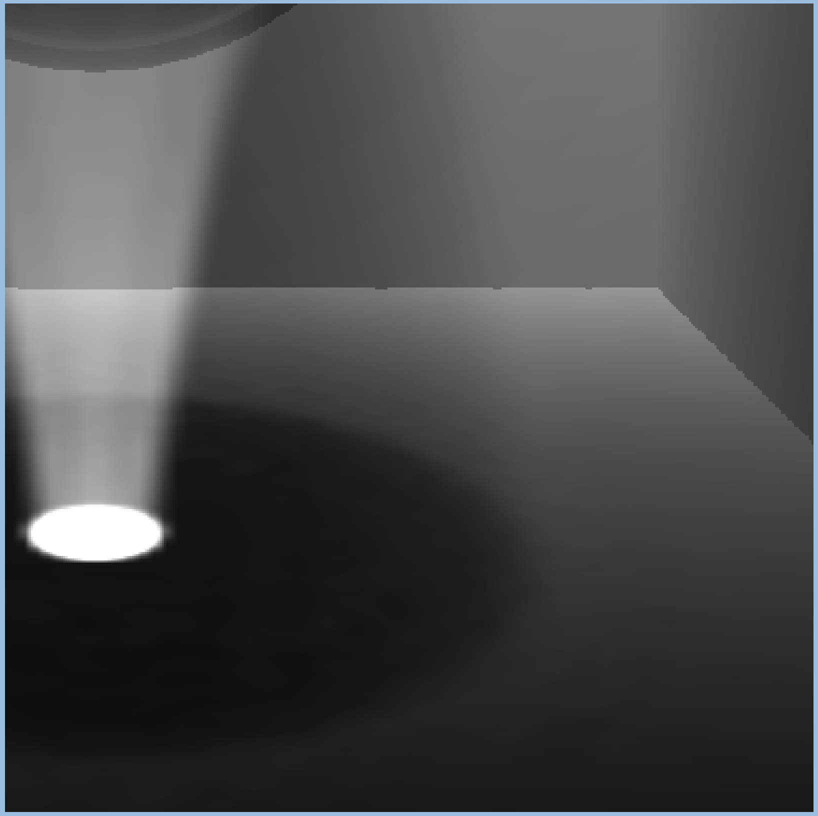
Photon-Weight



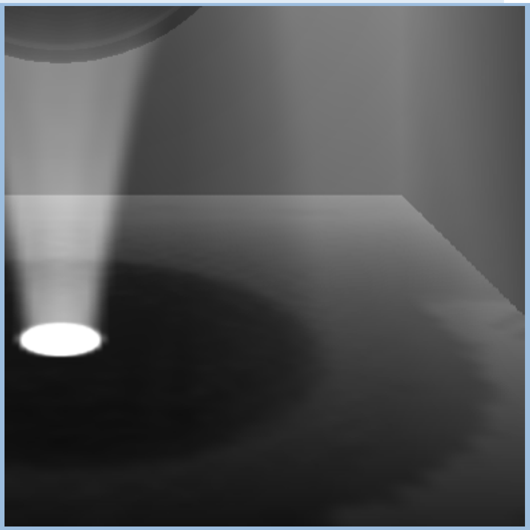
Center-Surround

Combination

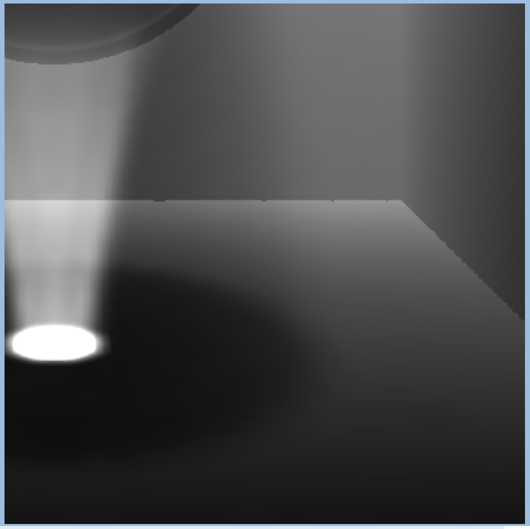
- Take minimum of the two.



Combination



Photon-Weight



Center-Surround

Contents

UNIVERSITÄT BERN

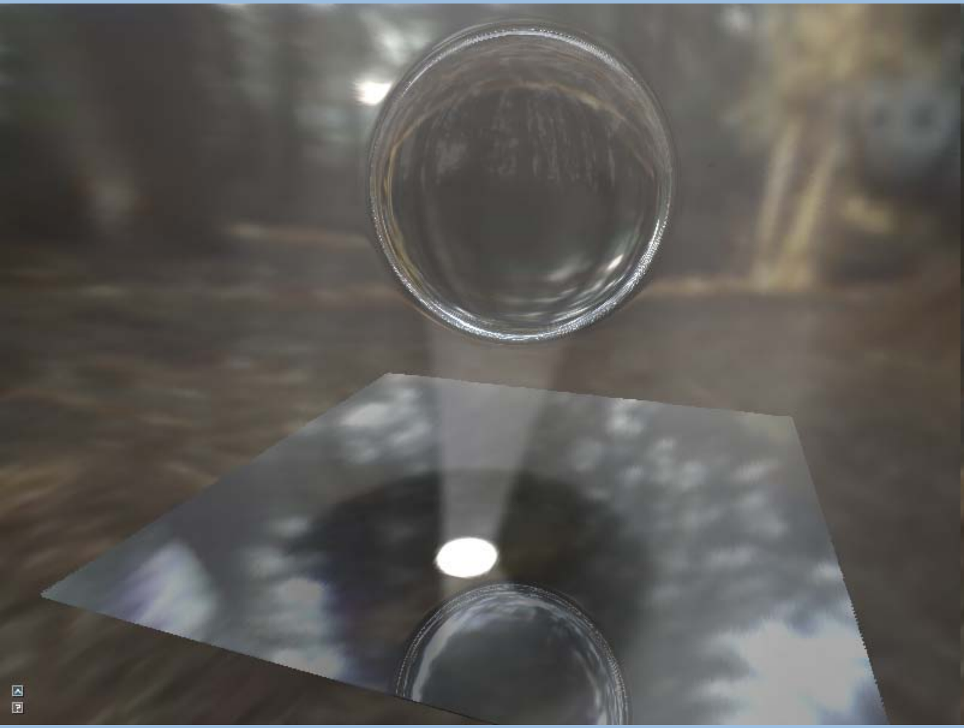
Evaluation

> Hardware:
  — nVdidia GTX 260, 896MB RAM

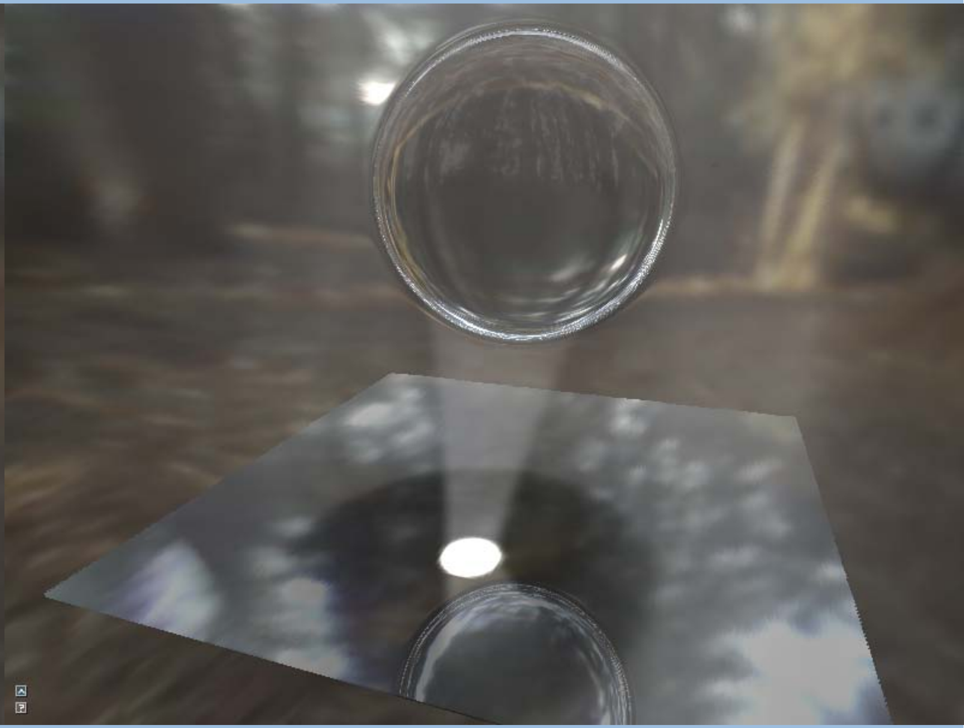
> General Settings:
  — 128x128x128 voxel grids
  — 800x600 images

> Adaptive Pipeline Setting:
  — ~4000 pilot photons

Sphere Scene

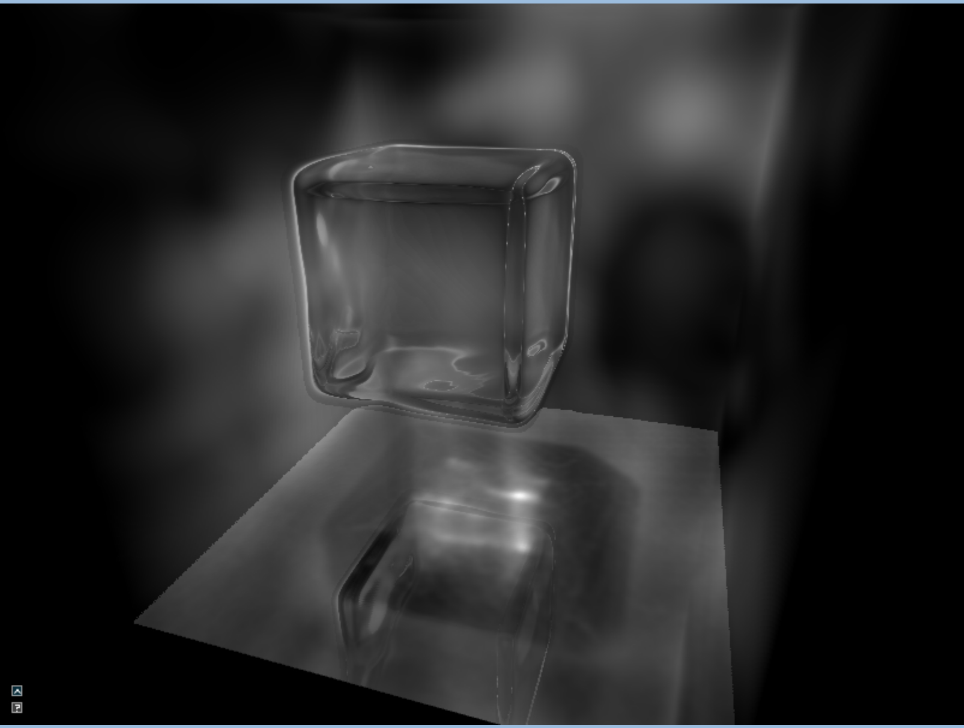


Reference
~122’000 photons
~2.6 FPS

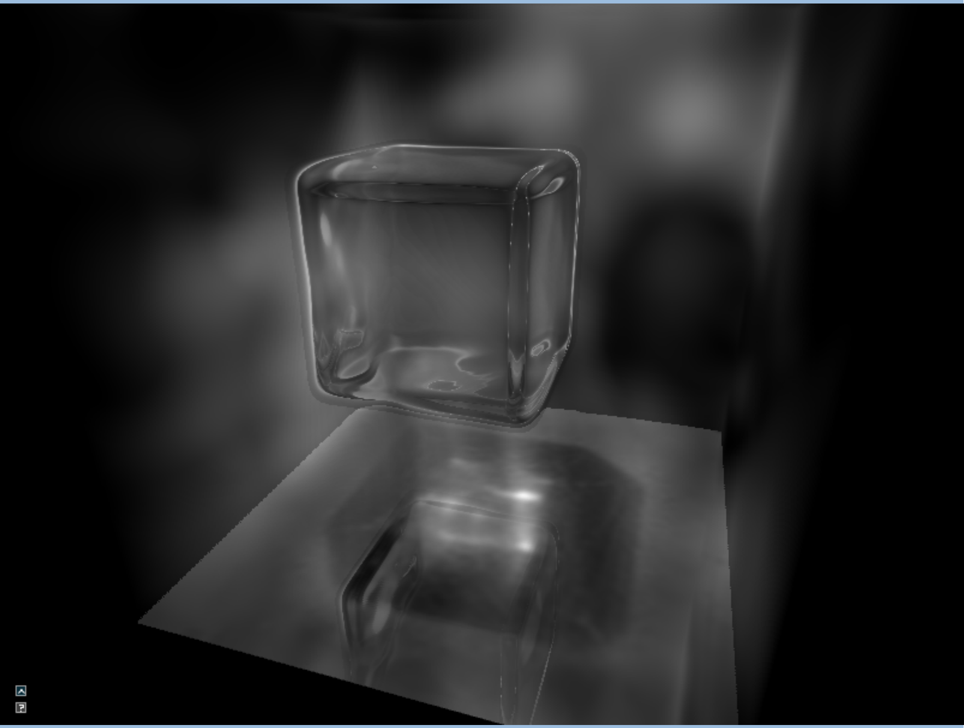


Adaptive
~50’000 photons
~2.5 FPS

Cube Scene



Reference
~125’000 photons
~2.2 FPS



Adaptive
~46’000 photons
~2.2 FPS

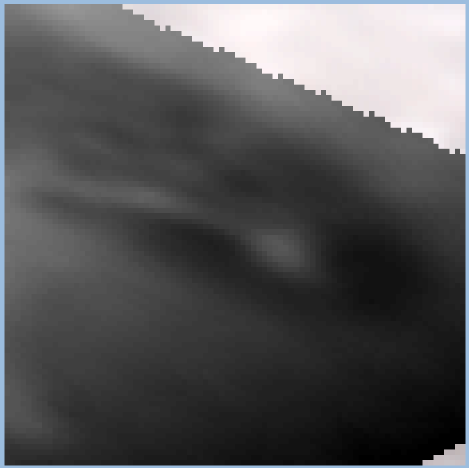
Armadillo Scene

Reference
~106’000 photons
~2.1 FPS

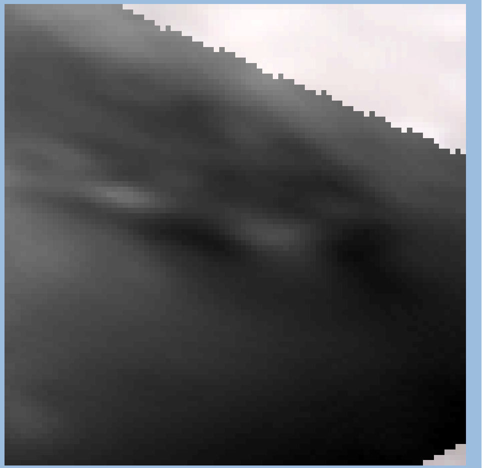
Adaptive
~49’000 photons
~2.0 FPS

Armadillo Close-Ups

Scene



Reference



Adaptive

Performance

Sphere Scene

	Reference	**Adaptive**
Voxelization	25 ms	25 ms
Octree	5 ms	5 ms
Photon Generation	< 1 ms	< 1 ms
Photon Marching	110 ms	10 ms
Photon Distribution	-	5 ms
Photon Generation	-	2 ms
Photon Marching	-	85 ms
Filtering	-	30 ms
Viewing Pass (+copy)	189 (+ 44) ms	189 (+ 44) ms
	373 ms	**395 ms**

47 ms overhead !!!

Problem

> Half of the photons does not lead to a photon marching twice as fast.

> Reason:
  - Atomic conflicts in photon marching (we trace the photon that are most likely to cause collisions)
  - Collection of additional data (photon weight, photon count)

> Additional passes take more time than we could save with photon reduction
  - Adaptive pipeline is faster when reference pipeline uses >200’000 photons

b
UNIVERSITÄT
BERN

Contents

UNIVERSITÄT BERN

Conclusion

> Implemented Sun et al. pipeline
> Extended Sun et al. pipeline with adaptive photon sampling
> We were able to halve the number of photons with equal quality.
> We were **not** able to accelerate the pipeline
  — Additional passes are too costly
  — Photon marching more expensive than expected

Future Work

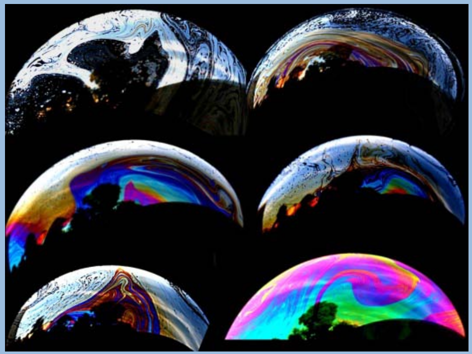
> Overhead minimization

> Atomic conflict reduction

> Automatic threshold adjustment

> Management of large scenes

> Multiple scattering, thin-film interference, …



Thin-film interference

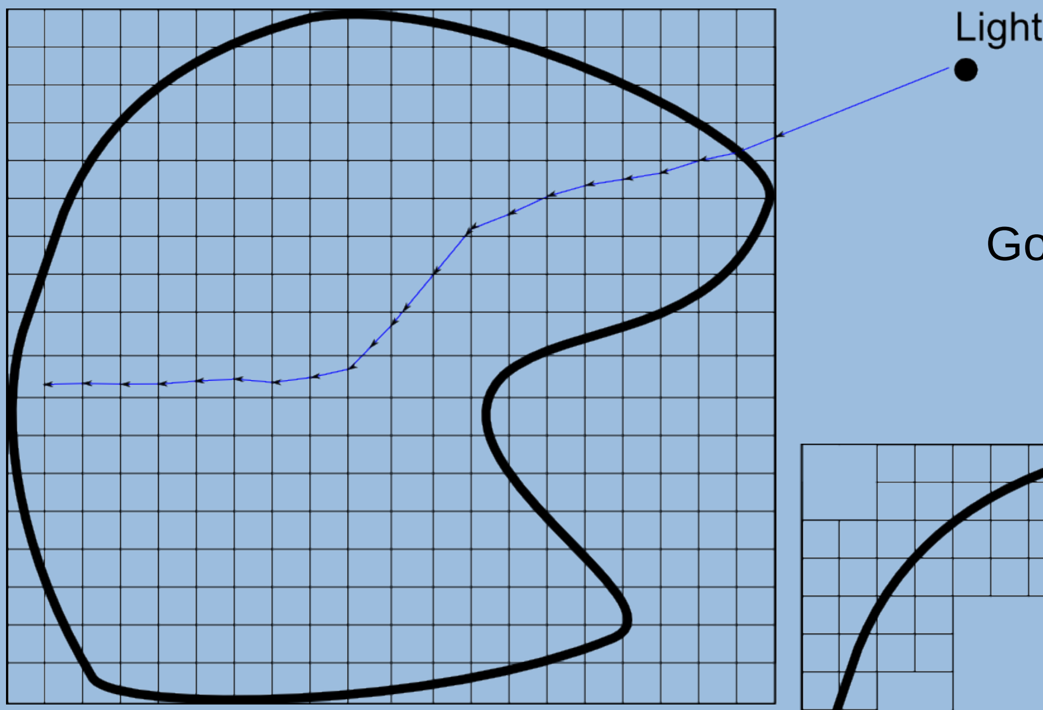
b
UNIVERSITÄT
BERN

Thank you for your attention!

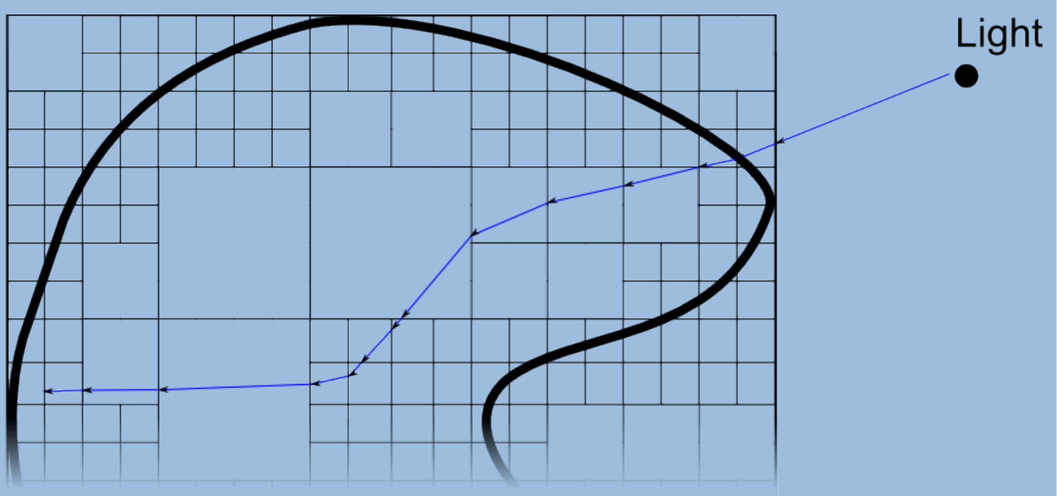
Questions?

u^b UNIVERSITÄT BERN

Octree



Goal: Accelerate Photon Tracing



Photon Marching

- Photon paths are computed with

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

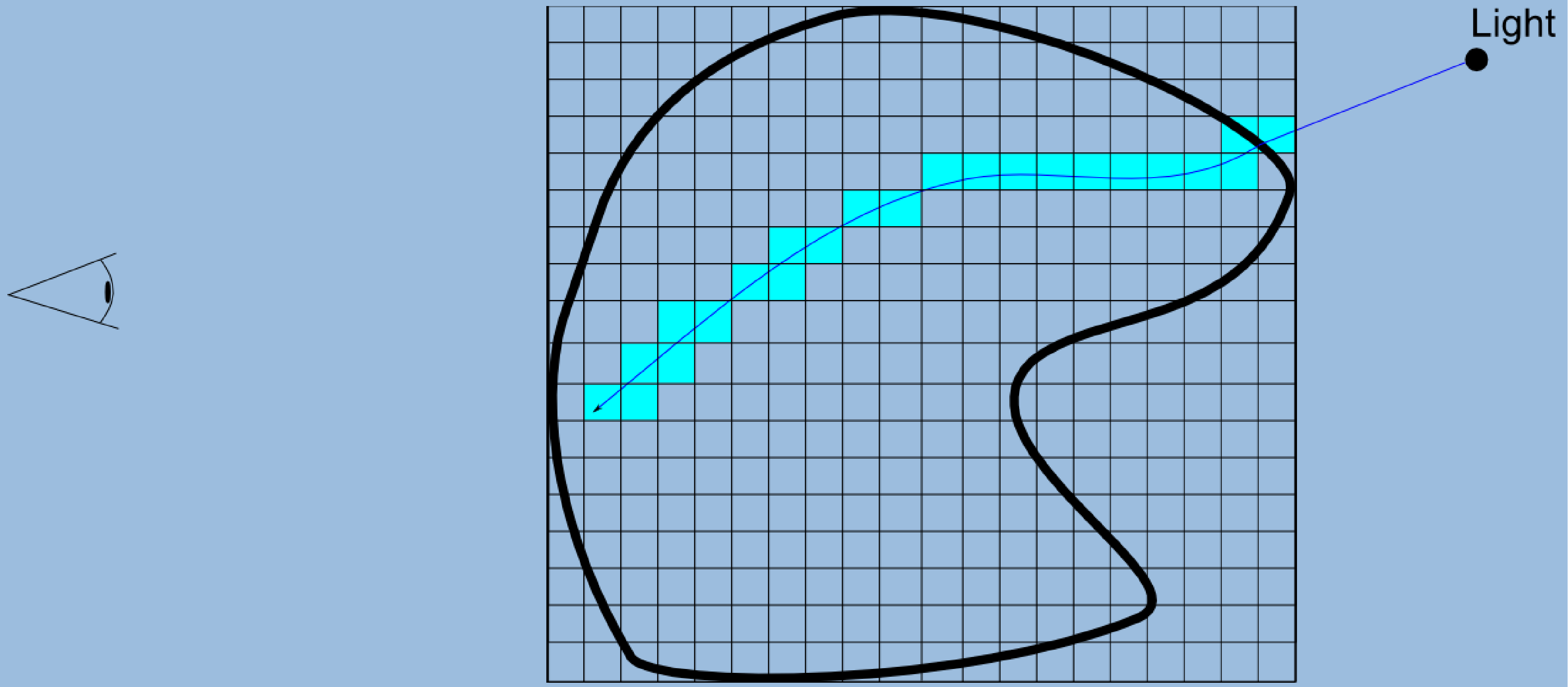
$$v_{i+1} = v_i + \Delta s \nabla n$$

- Arbitrary step size Δs (in our case given by octree)

- Formulas can be derived from the eikonal equation and the transport equation from geometric optics.

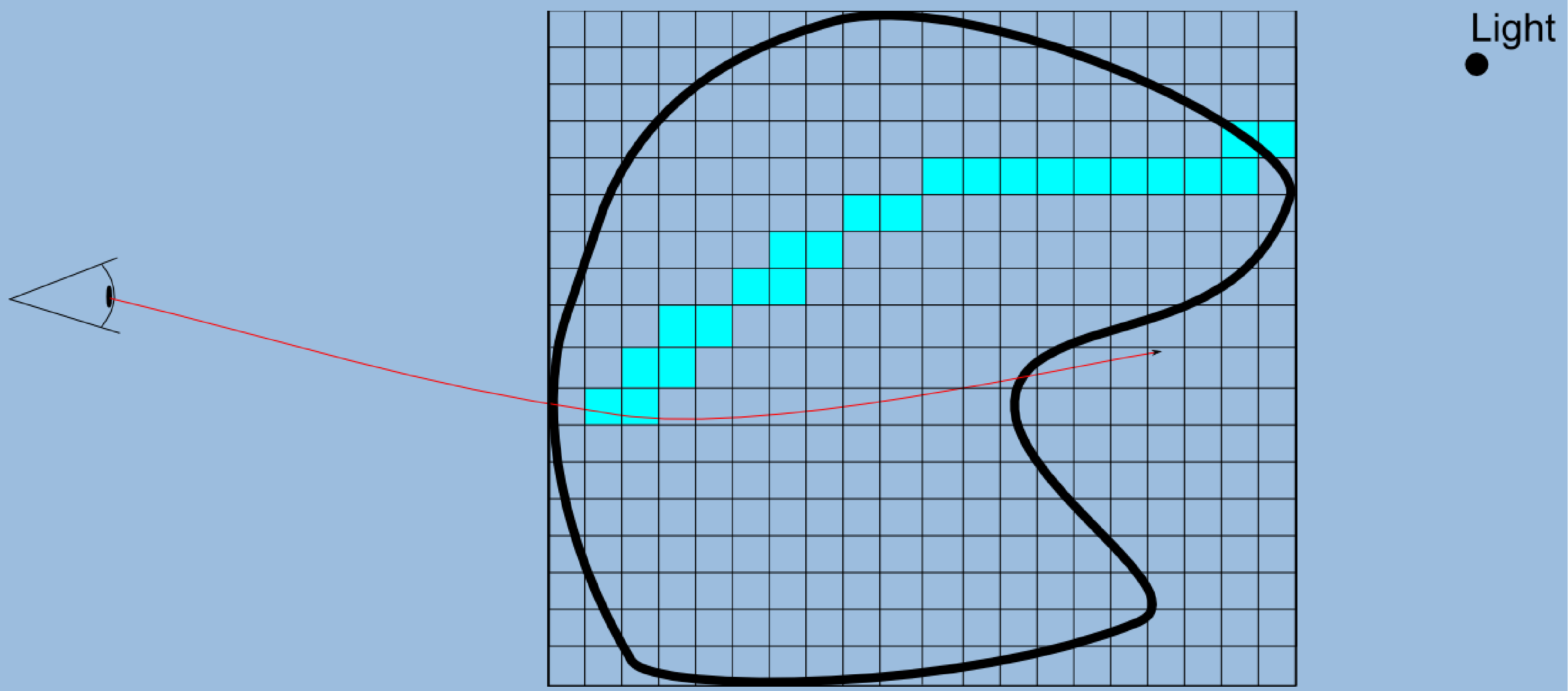
Photon Marching

- Update the radiance through every voxel a photon passed.
- Attenuate photon energy after each step (=> absorption).



Viewing Pass

- Shoot viewing rays and march as photons through grid.
- Evaluate scattering after each step (=> single scattering).

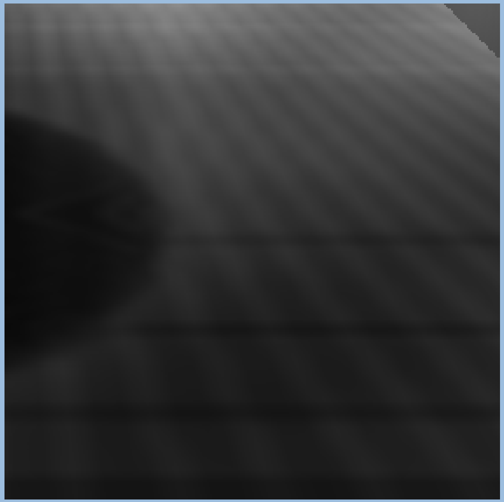


Performance (cont.)

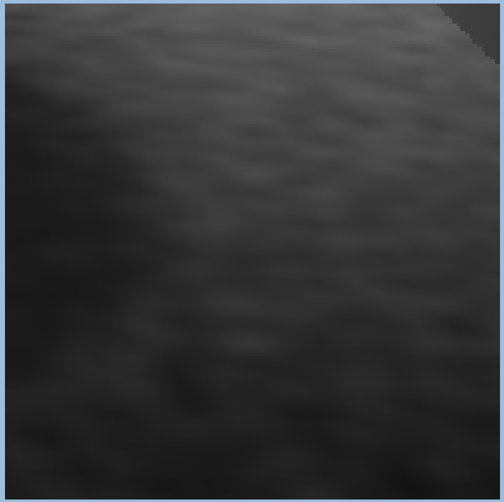
Performance and Sampling

- We use Hammersley sequence for sampling
  - Good visual results
  - Good performance

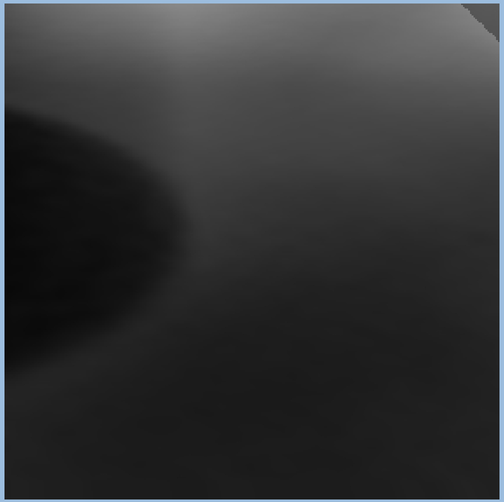
- Visual:
  - Hammersley shows hardly any noise



Grid



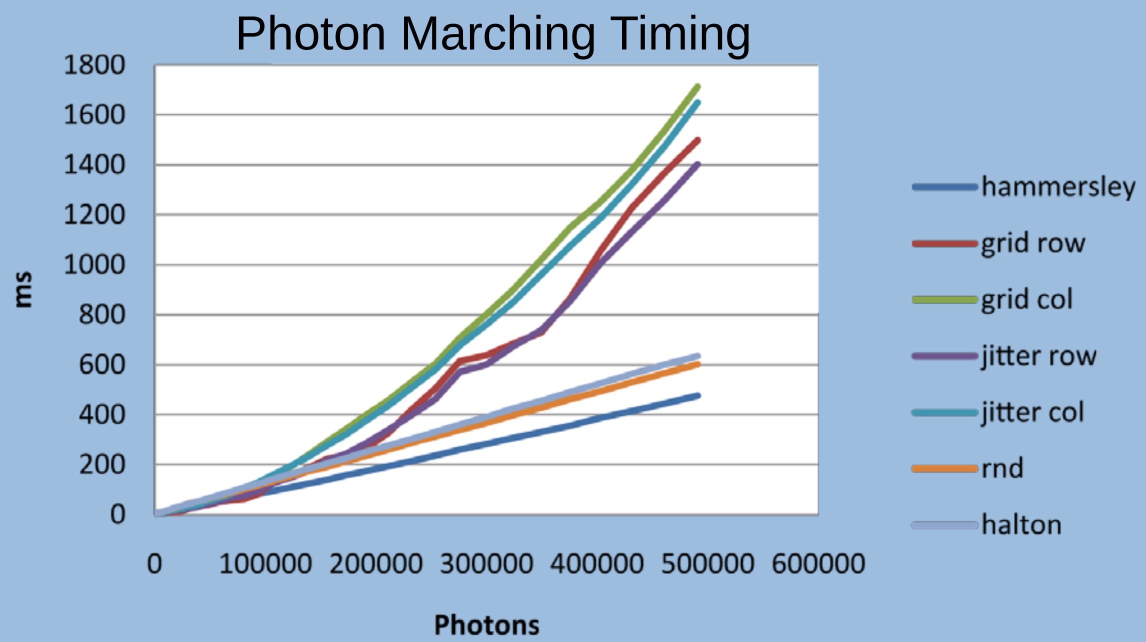
Random



Hammersley

Performance and Sampling (cont.)

- Performance:
  - Good trade-off between
    - Memory Access Pattern
    - Atomic Conflicts



UNIVERSITÄT BERN

Performance and Sampling (cont.)