Hierarchical Point Based Light Transport

Michael Zollhöfer

Betreuer:

Prof. Dr.-Ing. Marc Stamminger

Institut für Informatik
Lehrstuhl für Graphische Datenverarbeitung
Friedrich-Alexander-Universität Erlangen-Nürnberg

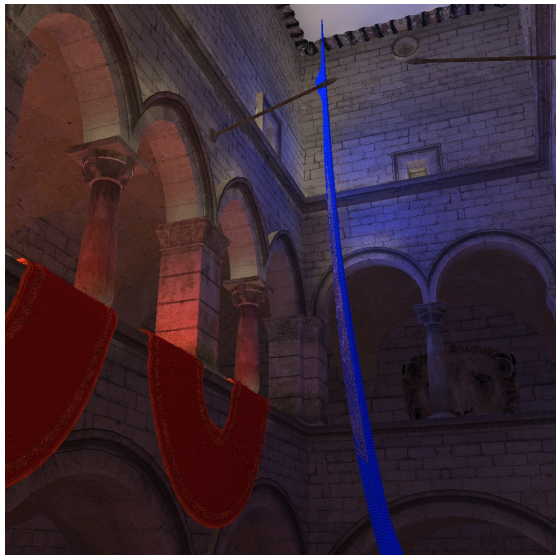
26.11.2010

Contents

Contents

Motivation

Direct Illumination



Indirect Illumination

Total Illumination

Related Work

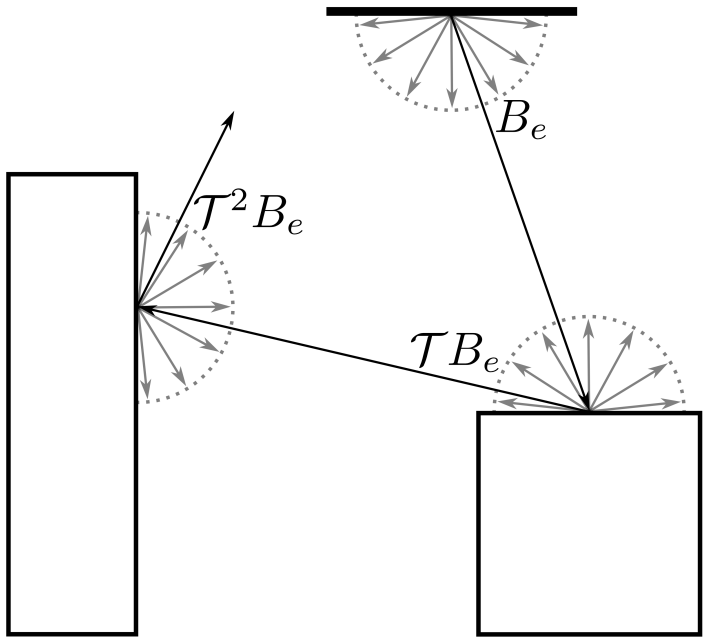
- Classical Radiosity
- Hierarchical Radiosity
- Clustering Radiosity
- Wavelets, Higher Order Basis Functions, ...
- Meshless Radiosity

Goals

- Implement Meshless Radiosity on the CPU
- Use the GPU to improve the performance
- Are there other ways to improve?

The Radiosity Equation

$$
\begin{aligned}
B_{out} &= B_e + \mathcal{T} B_{out} \\
&= B_e + \mathcal{T} B_e + \mathcal{T}^2 B_{out} \\
&= B_e + \mathcal{T} B_e + \mathcal{T}^2 B_e + \mathcal{T}^3 B_e + \dots
\end{aligned}
$$



The Discretized Radiosity Equation

- Apply a projection operator to project into a suitable function space
- Here we use the space panned by a Meshless Hierarchy

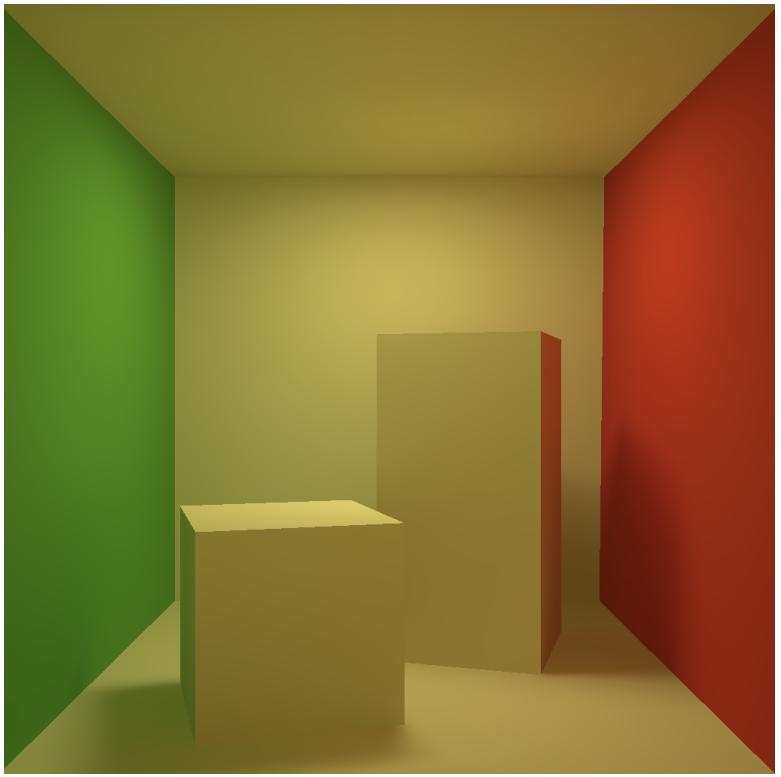
$$\mathcal{P}B_{out} = \mathcal{P}B_e + \mathcal{P}\mathcal{T}\mathcal{P}B_{out}$$

$$\mathbf{b}_{out} = \mathbf{b}_e + \mathbf{T}\mathbf{b}_{out}$$

Contents

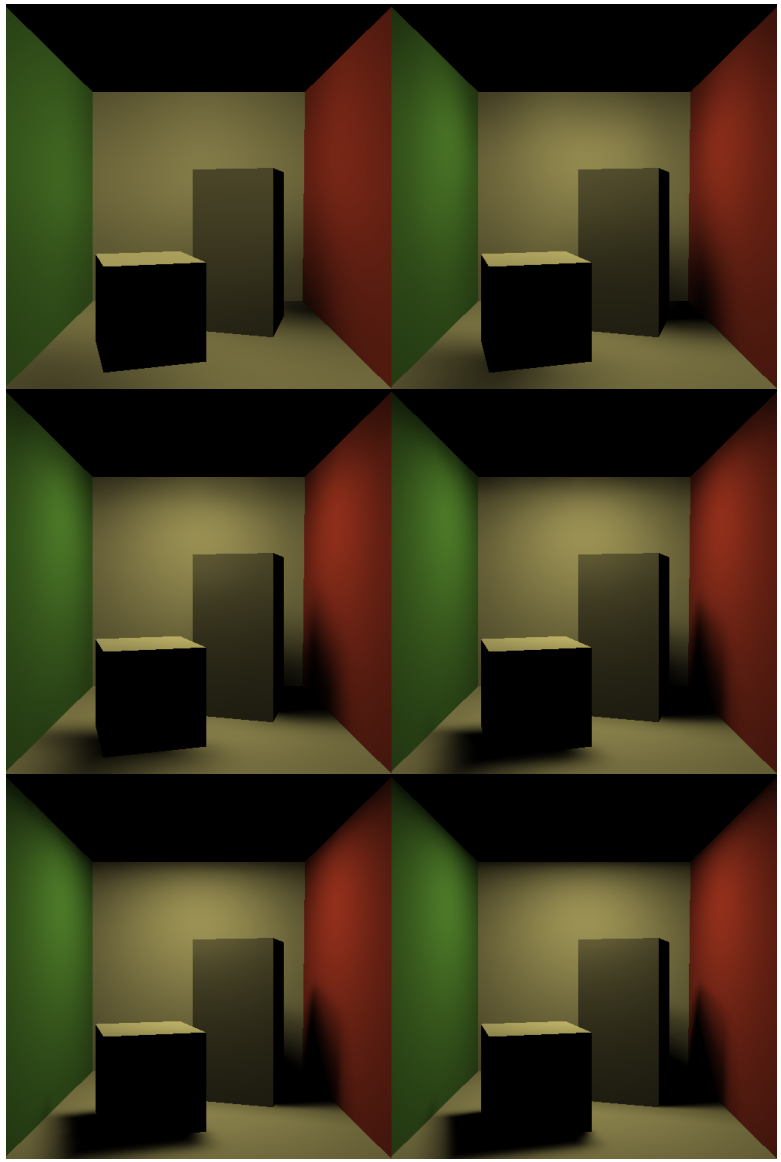
Advantages

- Hierarchical approach
- Point based representation
- Smooth solutions



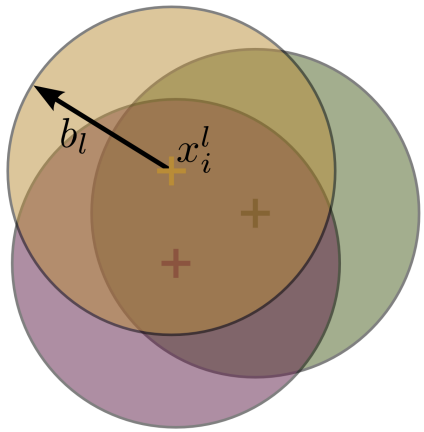
Meshless Hierarchy

- Use incident irradiance instead of radiosity
- Multi-level Shepard Approx.
  - Coarsest level encodes coarse approximation
  - Finer levels store detail as deltas
- Leads to a basis function expansion



Basis Functions

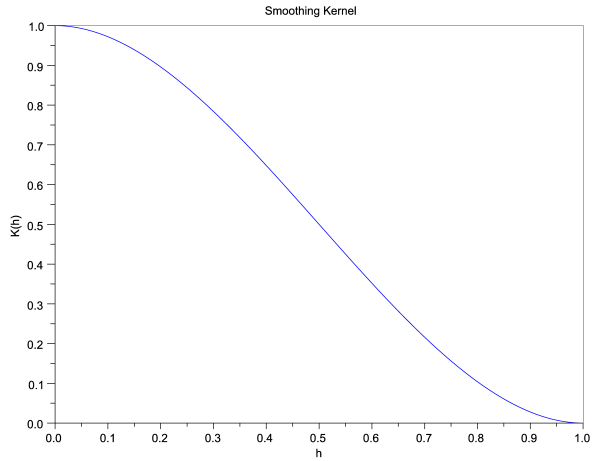
$$B_i^l(\mathbf{p}) = \frac{w_i^l(\mathbf{p})}{\sum_{j=0}^{N_l} w_j^l(\mathbf{p})}$$



$$w_i^l(\mathbf{p}) = K_i^l(MahaDist(\mathbf{p})) \cdot \max(0, \langle n\,, n_i^l \rangle)^2$$

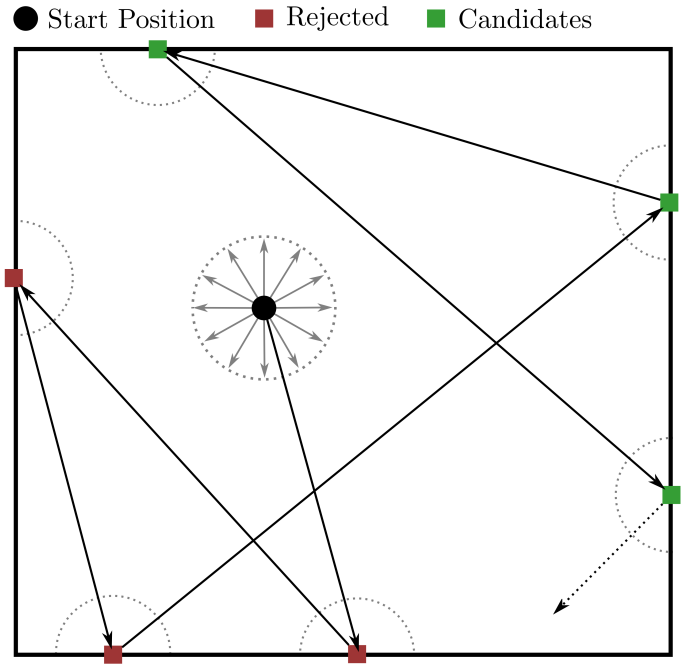
$$K_i^l(d) = K(\frac{d}{b_l})$$

$$K(h) = \begin{cases} 2h^3 - 3h^2 + 1 & h \leq 1 \\ 0 & h > 1 \end{cases}$$



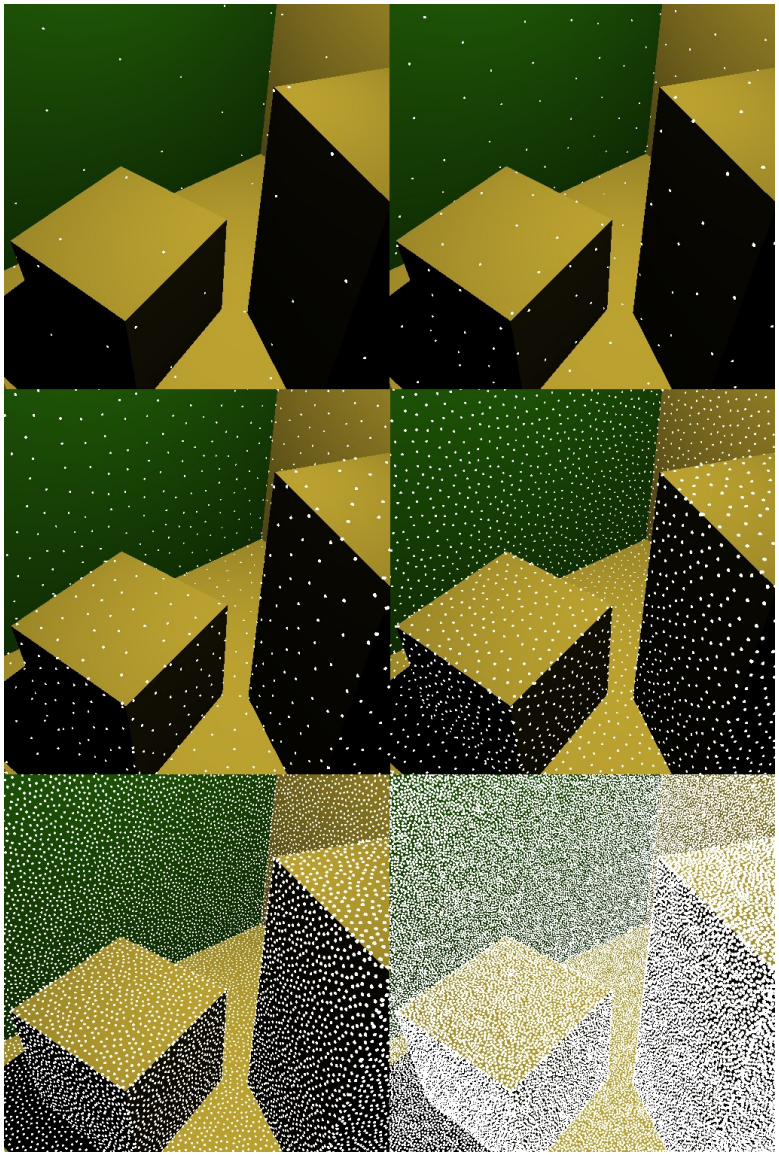
Placing the Basis Functions

- Basis functions have to overlap the whole domain
- Sample the scene using ray tracing
- Compute a Poisson Disk Distributions per level

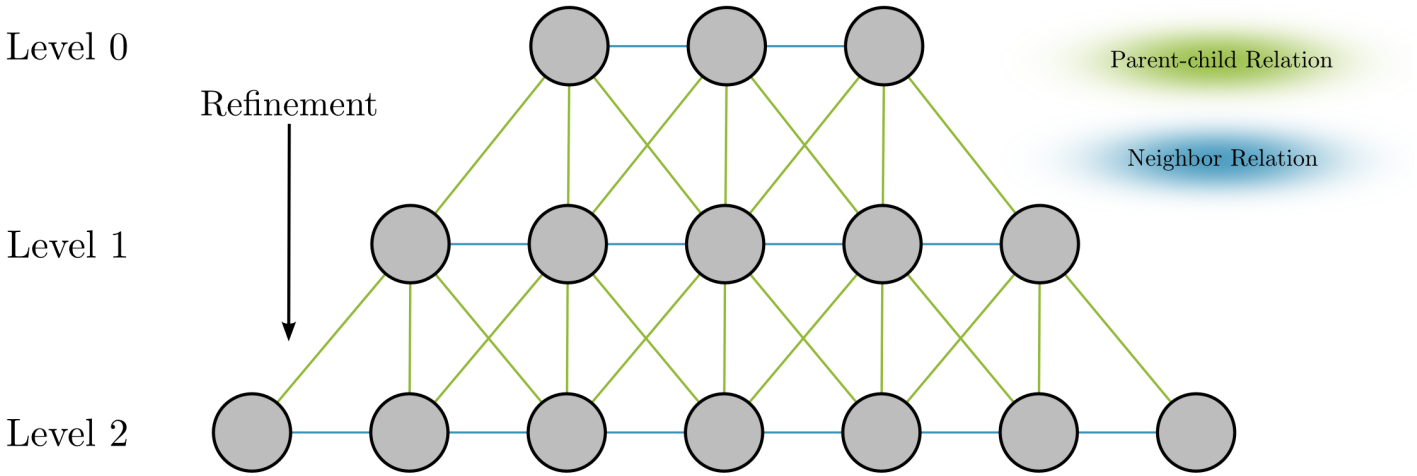


Placing the Basis Functions

- Basis functions have to overlap the whole domain
- Sample the scene using ray tracing
- Compute a Poisson Disk Distributions per level



Basis Function Expansion



$$F(\mathbf{p}) = \sum_{l=0}^{m-1} \sum_{j=0}^{N_l} \alpha_j^l \cdot B_j^l(\mathbf{p}) \qquad \alpha_i^l = \begin{cases} f_i^l & l = 0 \\ f_i^l - F_{l-1}(\mathbf{p}_i^l) & l > 0 \end{cases}$$

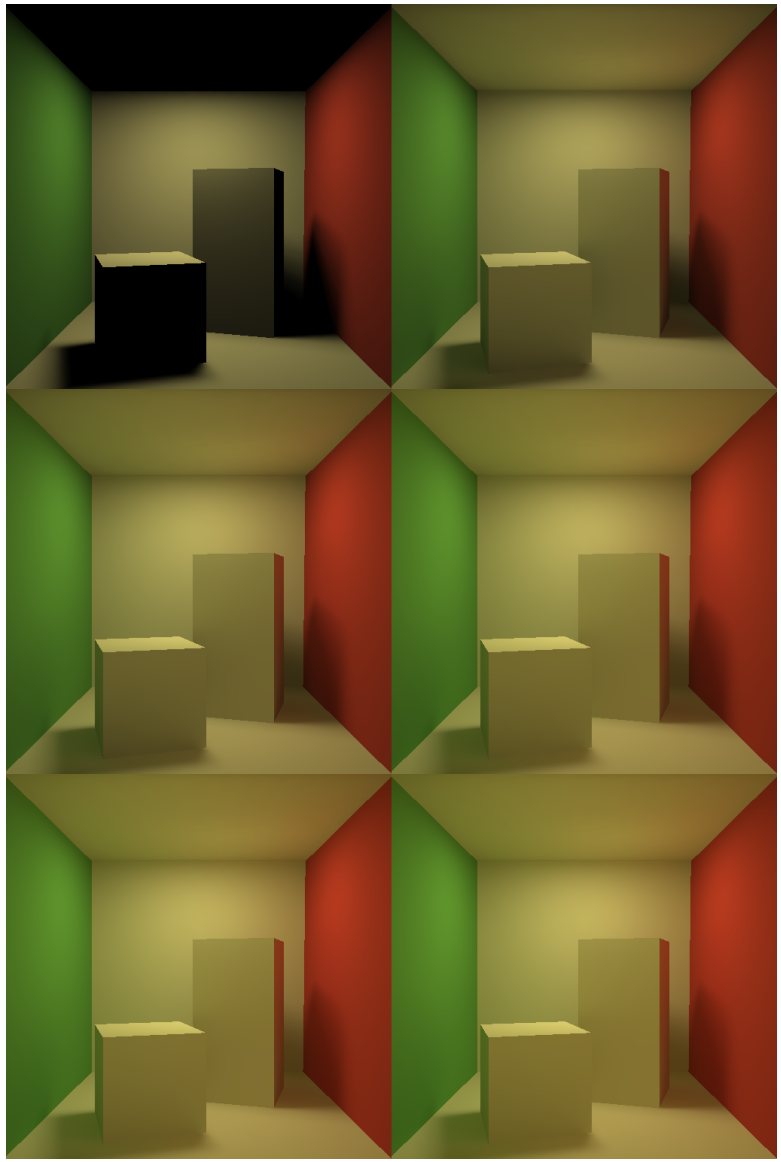
Evaluating the Basis Function Expansion

- Brute force evaluation too expensive
- Basis Functions have local support
  - Only few functions contribute
  - Use a KD-tree per level

$$F(\mathbf{p}) = \sum_{l=0}^{m-1} \sum_{j=0}^{N_l} \alpha_j^l \cdot B_j^l(\mathbf{p})$$

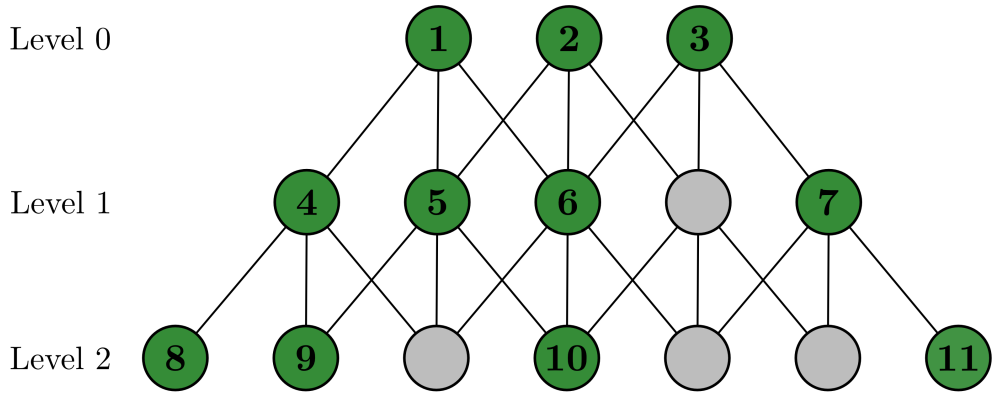
Basic Approach

1. Project the direct illumination into the basis spanned the Meshless Hierarchy
2. Iteratively compute the light bounces
   - Precompute transport links
   - Compute transport on-the-fly
   - Space vs time
3. Visualize the global illumination solution



Transport Operator

1. Encode “un-shot” energy
2. Start on the coarsest level
3. Gather incident illumination
   - Locate influencing basis functions (KD-tree)
   - Interpolate coefficients
4. Refine if necessary

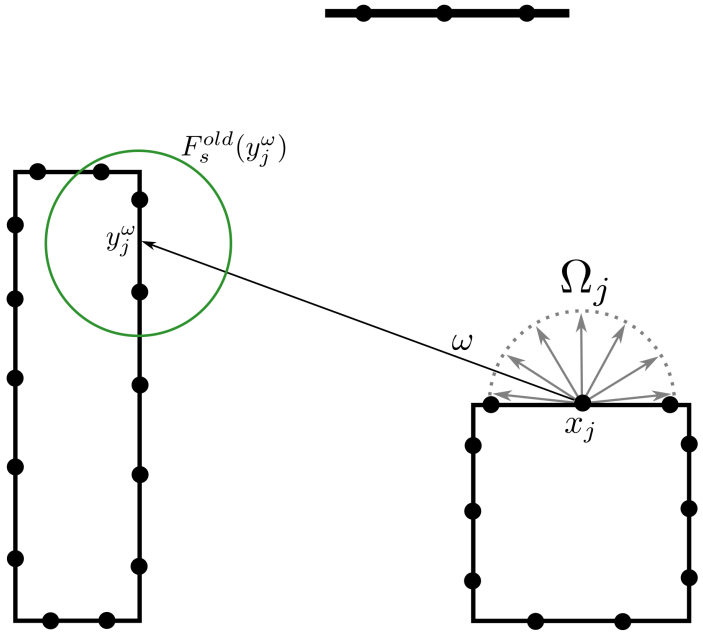


Incident Illumination

$$y_j^\omega = r(x_j, \omega)$$

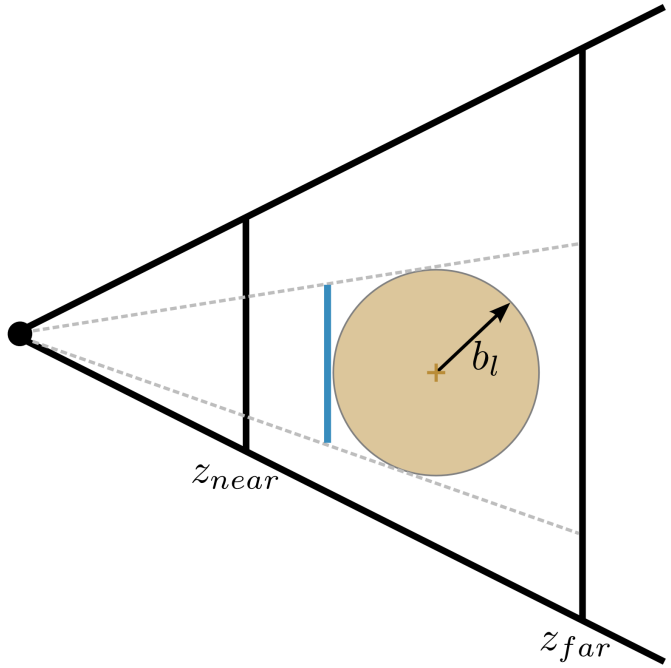
$$I'(x_j) = \frac{1}{\pi} \int_{\Omega_j} F_s^{old}(y_j^\omega) \rho(y_j^\omega) \langle n_j \, , \omega \rangle d\omega$$

$$I(x_j) = I'(x_j) - F_{l-1}^{new}(p_j)$$



Visualization

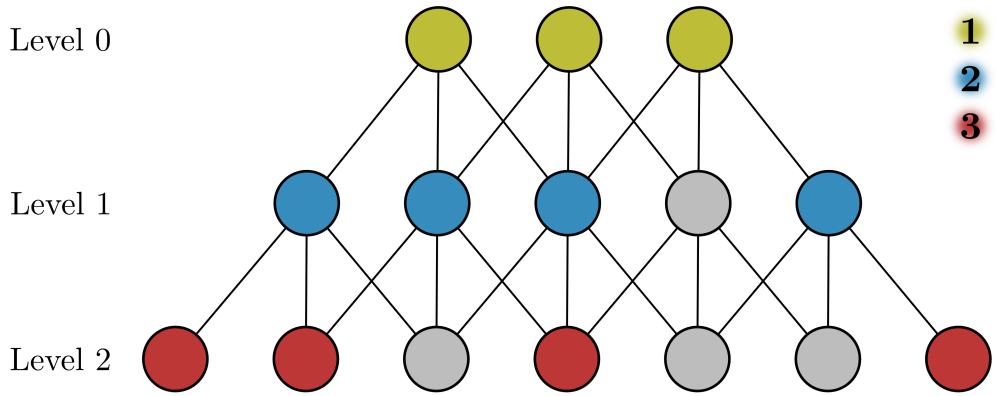
- Computed illumination can be visualized directly
  - Create G-Buffer
  - Basis function splatting
  - Per-pixel evaluation
- Each level has to be splatted separately
- Flatten hierarchy to reduce overdraw



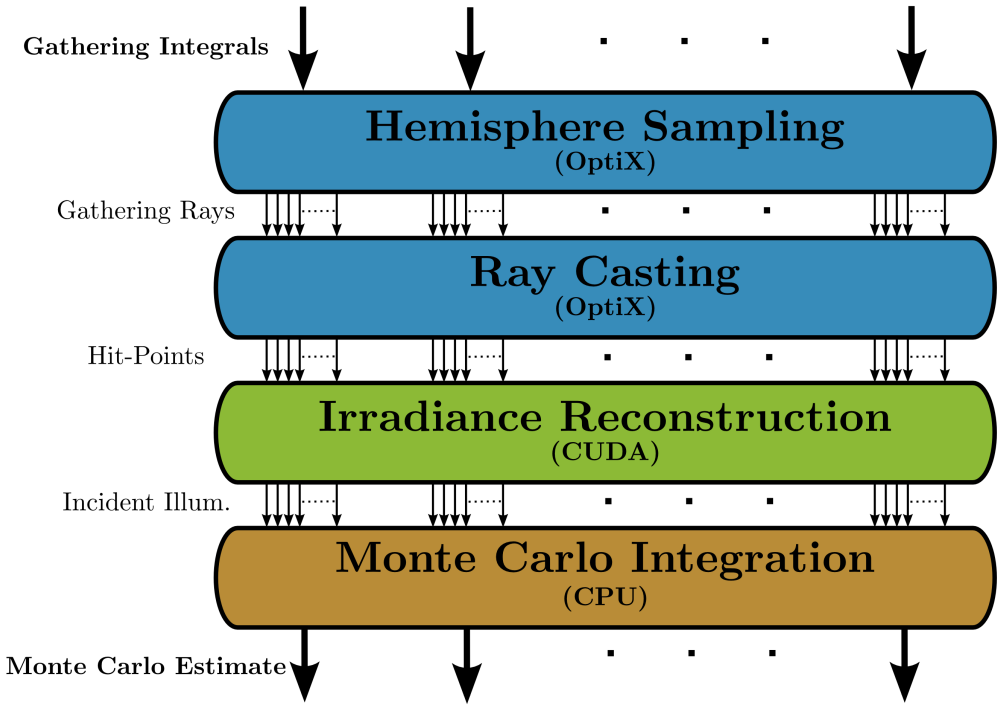
Contents

Analysis

- Integrals on a common level are independent
- Integrals on different levels are dependent
- Each integral itself can be parallelized



Used Pipeline



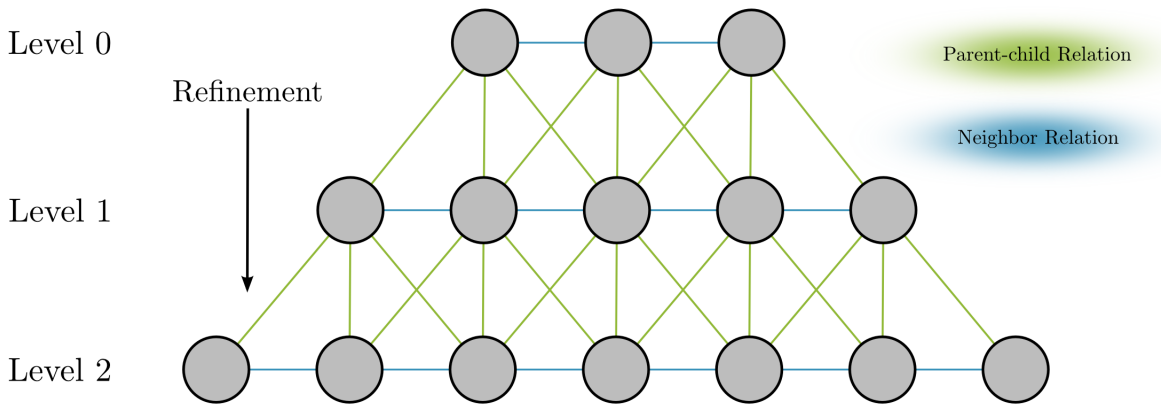
Implementation Details

- Batch Processing
  - Memory constraints
  - Not all integrals can be computed in parallel
  - Divide whole task into batches
- GPU KD-tree
  - Median split
  - 8-Byte nodes
  - Per-query-point stack in local memory

Contents

Problems

- Deltas coefficients are problematic
  - Parent transport has to be subtracted
  - The reconstruction requires multiple levels
  - Hierarchy has to be flattened
- Remove deltas, **but** stay adaptive



New Version of the Meshless Hierarchy

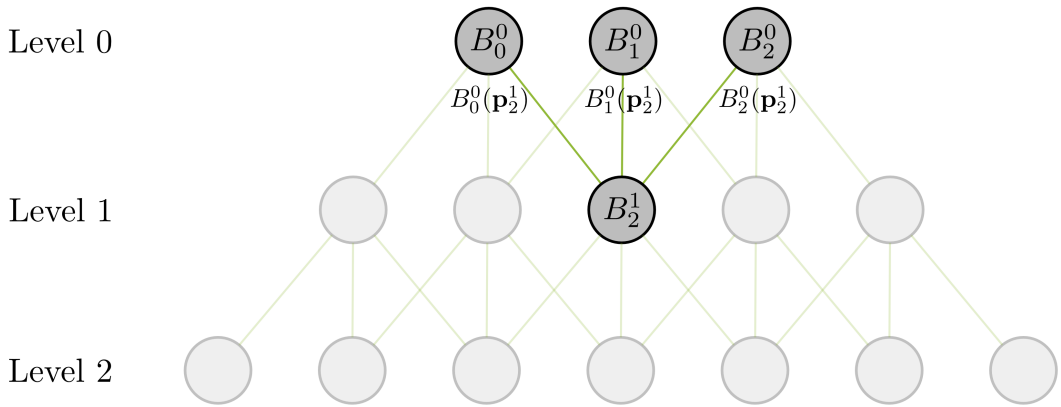
- Absolute Shepard Approximation on all levels
- Coefficients correspond to the illumination

$$F^l(\mathbf{p}) = \sum_{j=0}^{N_l} \alpha_j^l \cdot B_j^l(\mathbf{p})$$

$$\alpha_i^l = f_i^l$$

Illumination propagation

- Store influence in the parent-child relation
- Push the gathered illumination down the hierarchy



Approximate Reconstruction

- Do not reconstruct the illumination exactly
- Use illumination at the approximate nearest neighbor
- Saves basis function evaluations
- Error cancels, because of high gathering ray number

Contents

Timings (Core i7 860 @ 2.80 GHz, GeForce GTX 480)

	Cornell	Happy	Sponza
Primitives	36	1.05M	279k
Levels	6	6	6
Sender Levels	3	3	3
Threshold	0.016	0.016	0.15
Functions L0	218	582	1438
Functions L5	167k	288k	356k

Tabelle: Scene properties and used parameters

	Cornell	Happy	Sponza
1. Bounce	21s	499s	567s
2. Bounce	2.9s	154s	82s
3.Bounce	2.4s	44s	24s
Integrals	13k	67k	37k
Transport	**32s**	**728s**	**743s**

Tabelle: Performance of the original version on the CPU.

Timings (Core i7 860 @ 2.80 GHz, GeForce GTX 480)

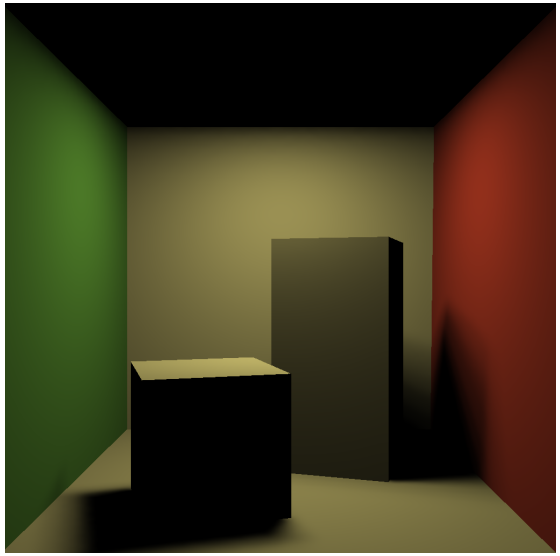
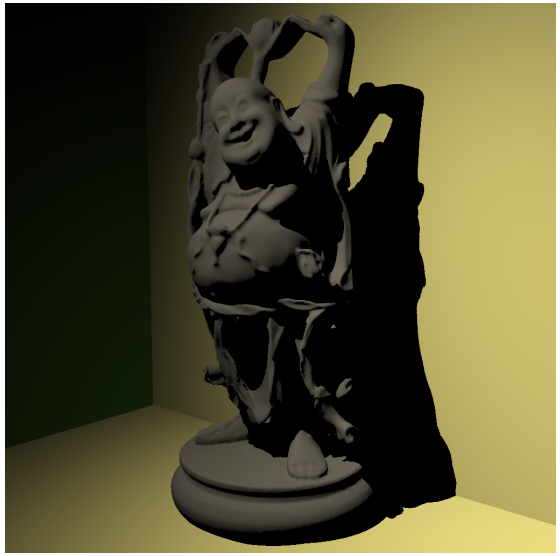
	Cornell	Happy	Sponza
1. Bounce	1.2s	15s	19s
2. Bounce	0.2s	4.1s	3.0s
3.Bounce	0.15s	1.3s	1.0s
Integrals	13k	66k	36k
Transport	1.9s	21s	26s
Speedup	**16.8x**	**34.7x**	**28.6x**

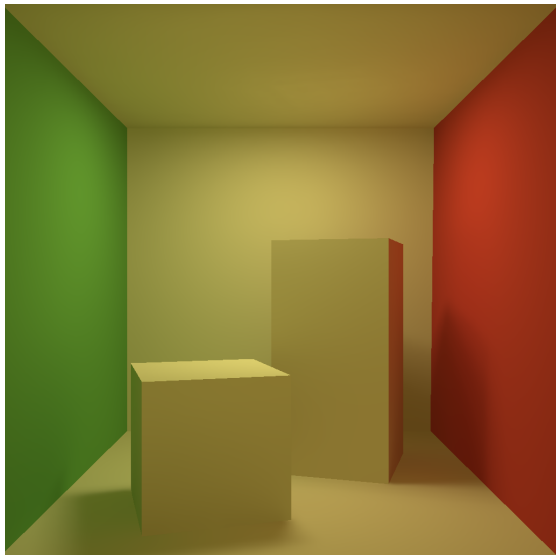
Tabelle: Performance of the original version on the GPU

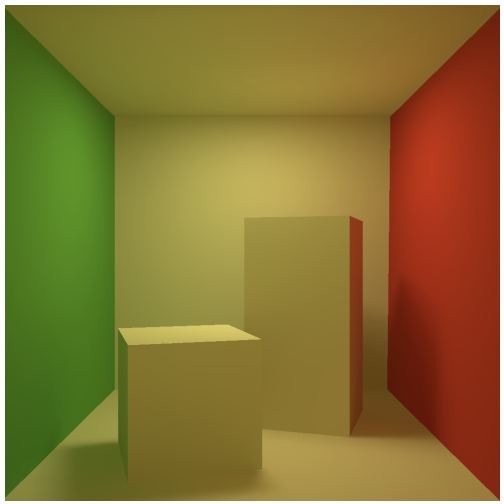
	Cornell	Happy	Sponza
1. Bounce	0.49s	3.5s	3.0s
2. Bounce	0.15s	1.0s	0.4s
3.Bounce	0.1s	0.4s	0.3s
Integrals	14k	68k	37k
Transport	1.0s	5.6s	4.5s
Speedup	**32x**	**130x**	**165x**

Tabelle: Performance of the optimized version on the GPU

Renderings





Any Questions?