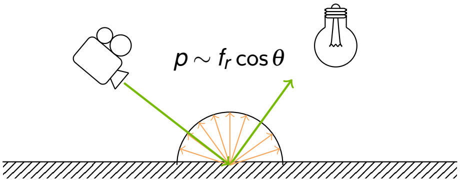
# Guiding and Shadow Rays

Alexander Keller, Ken Dahm, Nikolaus Binder, Thomas Müller

# Guiding and Shadow Rays

## Importance sampling of many light sources

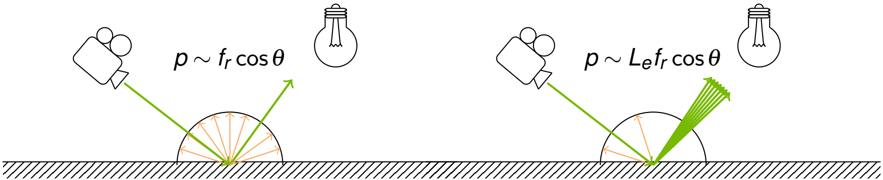
- sampling proportional to integrand



# Guiding and Shadow Rays

## Importance sampling of many light sources

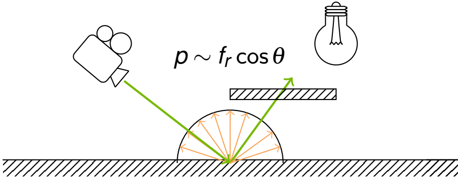
- sampling proportional to integrand

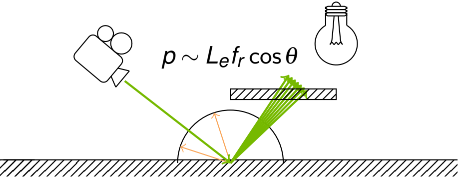


# Guiding and Shadow Rays

## Importance sampling of many light sources

- sampling proportional to integrand
  - requires to include visibility

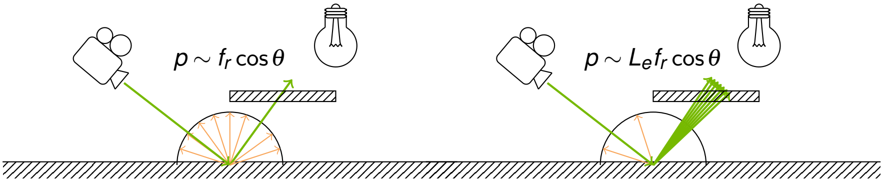


# Guiding and Shadow Rays

## Importance sampling of many light sources

- sampling proportional to integrand
  - requires to include visibility




- goals
  - massively parallel
  - linear in number of paths
  - constant time

# Guiding and Shadow Rays

## Previous work

- sorting lights by their unoccluded contribution, keeping record of their average visibility
  - Adaptive shadow testing for ray tracing [War91]

# Guiding and Shadow Rays

## Previous work

- sorting lights by their unoccluded contribution, keeping record of their average visibility
  - Adaptive shadow testing for ray tracing [War91]
- potentially visible sets
  - Visibility computations in densely occluded polyhedral environments [Tel92]

# Guiding and Shadow Rays

## Previous work

- sorting lights by their unoccluded contribution, keeping record of their average visibility
  - Adaptive shadow testing for ray tracing [War91]
- potentially visible sets
  - Visibility computations in densely occluded polyhedral environments [Tel92]
- spatial subdivision referencing important lights per stratum (besides many other useful things)
  - Monte Carlo techniques for direct lighting calculations [SWZ96]

# Guiding and Shadow Rays

## Previous work

- sorting lights by their unoccluded contribution, keeping record of their average visibility
  - Adaptive shadow testing for ray tracing [War91]
- potentially visible sets
  - Visibility computations in densely occluded polyhedral environments [Tel92]
- spatial subdivision referencing important lights per stratum (besides many other useful things)
  - Monte Carlo techniques for direct lighting calculations [SWZ96]
- contributing light sources determined by photons
  - Efficient importance sampling techniques for the photon map [KW00]

# Guiding and Shadow Rays

## Previous work

- sorting lights by their unoccluded contribution, keeping record of their average visibility
  - Adaptive shadow testing for ray tracing [War91]
- potentially visible sets
  - Visibility computations in densely occluded polyhedral environments [Tel92]
- spatial subdivision referencing important lights per stratum (besides many other useful things)
  - Monte Carlo techniques for direct lighting calculations [SWZ96]
- contributing light sources determined by photons
  - Efficient importance sampling techniques for the photon map [KW00]
- contribution of light sources estimated by sampling some paths across the image
  - Interactive global illumination in complex and highly occluded environments [WBS03]

# Guiding and Shadow Rays

## Previous work

- importance resampling
  - Importance resampling for global illumination [TCE05]

# Guiding and Shadow Rays

## Previous work

- importance resampling
  - Importance resampling for global illumination [TCE05]
- cache points referencing contributing lights
  - Importance caching for complex illumination [GKPS12]

# Guiding and Shadow Rays

## Previous work

- importance resampling
  - Importance resampling for global illumination [TCE05]
- cache points referencing contributing lights
  - Importance caching for complex illumination [GKPS12]
- probabilistic traversal of light hierarchy
  - Efficient sampling of many lights [Ces14], see `https://ompf2.com/viewtopic.php?t=1938`

# Guiding and Shadow Rays

## Previous work

- importance resampling
  - Importance resampling for global illumination [TCE05]
- cache points referencing contributing lights
  - Importance caching for complex illumination [GKPS12]
- probabilistic traversal of light hierarchy
  - Efficient sampling of many lights [Ces14], see `https://ompf2.com/viewtopic.php?t=1938`
- caching importance records
  - Probabilistic connections for bidirectional path tracing [PRDD15]

# Guiding and Shadow Rays

## Previous work

- importance resampling
  - Importance resampling for global illumination [TCE05]
- cache points referencing contributing lights
  - Importance caching for complex illumination [GKPS12]
- probabilistic traversal of light hierarchy
  - Efficient sampling of many lights [Ces14], see `https://ompf2.com/viewtopic.php?t=1938`
- caching importance records
  - Probabilistic connections for bidirectional path tracing [PRDD15]
- generalization to guiding by probability hierarchies
  - The Iray light transport simulation and rendering system [KWRSvAKK17]

# Guiding and Shadow Rays

## Previous work

- refined bounds and clustering
  - Importance sampling of many lights with adaptive tree splitting [CK18]

# Guiding and Shadow Rays

## Previous work

- refined bounds and clustering
  - Importance sampling of many lights with adaptive tree splitting [CK18]
- cache points referencing lights accounting 97% of the energy
  - The design and evolution of Disney's Hyperion renderer [BACDHKKKT18]

# Guiding and Shadow Rays

## Previous work

- refined bounds and clustering
  - Importance sampling of many lights with adaptive tree splitting [CK18]
- cache points referencing lights accounting 97% of the energy
  - The design and evolution of Disney's Hyperion renderer [BACDHKKKT18]
- learning importance of clusters in a hierarchy combined with separation of the main part
  - Bayesian online regression for adaptive direct illumination sampling [VKK18]

# Guiding and Shadow Rays

## Previous work

- refined bounds and clustering
  - Importance sampling of many lights with adaptive tree splitting [CK18]
- cache points referencing lights accounting 97% of the energy
  - The design and evolution of Disney's Hyperion renderer [BACDHKKKT18]
- learning importance of clusters in a hierarchy combined with separation of the main part
  - Bayesian online regression for adaptive direct illumination sampling [VKK18]
- data structures
  - Efficient data structures and sampling of many light sources for next event estimation [Mik18]
  - see `https://github.com/AndiMiko/masterthesis/releases`

# Importance Sampling

## Partial cumulative distribution function (CDF)

- index set $I := \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ of references $i_j$ to (point) light sources

# Importance Sampling

## Partial cumulative distribution function (CDF)

- index set $I := \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ of references $i_j$ to (point) light sources
- probability density function storing only the $q_i$ for $i \in I$

$$p_i := \begin{cases} (1-b) \cdot q_i + b \cdot \frac{1}{n} & \text{for } i \in I \\ b \cdot \frac{1}{n} & \text{for } i \notin I \end{cases}$$

# Importance Sampling

## Partial cumulative distribution function (CDF)

- index set $I := \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ of references $i_j$ to (point) light sources
- probability density function storing only the $q_i$ for $i \in I$

$$p_i := \begin{cases} (1-b) \cdot q_i + b \cdot \frac{1}{n} & \text{for } i \in I \\ b \cdot \frac{1}{n} & \text{for } i \notin I \end{cases}$$

- sampling
  - proportional to $q_i$ with probability $1-b$, using cumulative distribution $Q_k := \sum_{j=1}^{k} q_{i_j}$
  - uniform with probability $b$

# Importance Sampling

## Partial cumulative distribution function (CDF)

- index set $I := \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ of references $i_j$ to (point) light sources
- probability density function storing only the $q_i$ for $i \in I$

$$p_i := \begin{cases} (1-b) \cdot q_i + b \cdot \frac{1}{n} & \text{for } i \in I \\ b \cdot \frac{1}{n} & \text{for } i \notin I \end{cases}$$

- sampling
  - proportional to $q_i$ with probability $1-b$, using cumulative distribution $Q_k := \sum_{j=1}^{k} q_{i_j}$
  - uniform with probability $b$

# Importance Sampling

## Partial cumulative distribution function (CDF)

- index set $I := \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ of references $i_j$ to (point) light sources
- probability density function storing only the $q_i$ for $i \in I$

$$p_i := \begin{cases} (1-b) \cdot q_i + b \cdot \frac{1}{n} & \text{for } i \in I \\ b \cdot \frac{1}{n} & \text{for } i \notin I \end{cases}$$

- sampling
  - proportional to $q_i$ with probability $1 - b$, using cumulative distribution $Q_k := \sum_{j=1}^{k} q_{i_j}$
  - uniform with probability $b$

# Importance Sampling

## Partial cumulative distribution function (CDF)

- index set $I := \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ of references $i_j$ to (point) light sources
- probability density function storing only the $q_i$ for $i \in I$

$$p_i := \begin{cases} (1-b) \cdot q_i + b \cdot \frac{1}{n} & \text{for } i \in I \\ b \cdot \frac{1}{n} & \text{for } i \notin I \end{cases}$$

- multiple importance sampling
  - proportional to $q_i$ with probability $1-b$, using cumulative distribution $Q_k := \sum_{j=1}^{k} q_{i_j}$
  - light hierarchy with probability $b$

# Importance Sampling

## Partial cumulative distribution function (CDF)

- index set $I := \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ of references $i_j$ to (point) light sources

- probability density function storing only the $q_i$ for $i \in I$

$$p_i := \begin{cases} (1-b) \cdot q_i + b \cdot \frac{1}{n} & \text{for } i \in I \\ b \cdot \frac{1}{n} & \text{for } i \notin I \end{cases}$$

- multiple importance sampling
  - proportional to $q_i$ with probability $1-b$, using cumulative distribution $Q_k := \sum_{j=1}^{k} q_{i_j}$
  - light hierarchy with probability $b$

- discrete density simulation, see `https://arxiv.org/abs/1901.05423`

# Importance Sampling

## Finding the probabilities $q_i$

- $q_i$ as normalized accumulated flux

$$\Phi = \frac{\max\{\cos\omega, 0\} \cdot L(x,\omega)}{p(x) \cdot p(s,\omega)}$$

  - probability $p(x)$ of selecting location $x$ (unless point light source)
  - probability $p(s,\omega)$ of selecting direction $\omega$ in shading point $s$

# Importance Sampling

## Finding the probabilities $q_i$

- $q_i$ as normalized accumulated flux

$$\Phi = \frac{\max\{\cos\omega, 0\} \cdot L(x, \omega)}{p(x) \cdot p(s, \omega)}$$

  - probability $p(x)$ of selecting location $x$ (unless point light source)
  - probability $p(s, \omega)$ of selecting direction $\omega$ in shading point $s$

- learning probabilities $q_i$ during path tracing
  - simple inclusion of path guiding for scattering

# Importance Sampling

## Finding the probabilities $q_i$

- $q_i$ as normalized accumulated flux

$$\Phi = \frac{\max\{\cos\omega, 0\} \cdot L(x, \omega)}{p(x) \cdot p(s, \omega)}$$

  - probability $p(x)$ of selecting location $x$ (unless point light source)
  - probability $p(s, \omega)$ of selecting direction $\omega$ in shading point $s$

- learning probabilities $q_i$ during path tracing
  - simple inclusion of path guiding for scattering

- photon-based next event estimation
  - origins of photons within search radius to determine set $l$ of (point) light sources

# Efficient Implementation

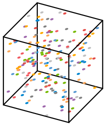
## Linear complexity

- store one partial CDF per hashed cell
- stochastic interpolation
- accumulate probabilities similar to Massively Parallel Path Space Filtering
  - see `https://arxiv.org/abs/1902.05942`

# Efficient Implementation

## Hashing instead of searching

- descriptors for selected vertices include

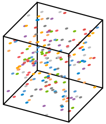


world space location $x$

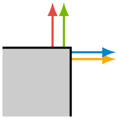
# Efficient Implementation

## Hashing instead of searching

- descriptors for selected vertices include



world space location $x$

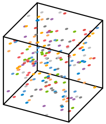


and optionally normal $n$,

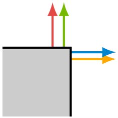
# Efficient Implementation

## Hashing instead of searching

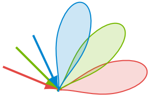
- descriptors for selected vertices include



world space location $x$



and optionally normal $n$,

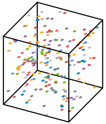


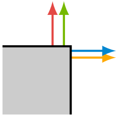
incident angle $\omega$,

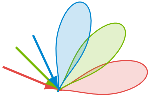
# Efficient Implementation

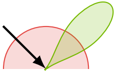
## Hashing instead of searching

- descriptors for selected vertices include


world space location $x$


and optionally normal $n$,

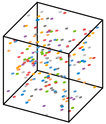
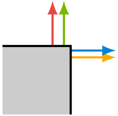
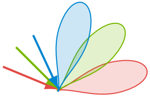
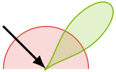

incident angle $\omega$,


and BRDF layer

# Efficient Implementation

## Hashing instead of searching

- descriptors for selected vertices include

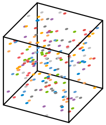
world space location $x$ and optionally normal $n$, incident angle $\omega$, and BRDF layer

- storing and loading data using hashed quantized descriptors
  - trade a larger hash table size for faster access (proportional to number of paths)

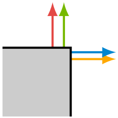
# Efficient Implementation

## Hashing instead of searching

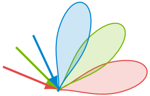
- descriptors for selected vertices include



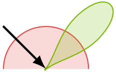
world space location $x$



and optionally normal $n$,



incident angle $\omega$,



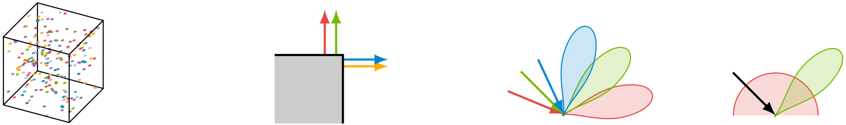
and BRDF layer

- storing and loading data using hashed quantized descriptors
  - trade a larger hash table size for faster access (proportional to number of paths)
  - use a second hash of the descriptor instead of storing full keys

# Efficient Implementation

## Hashing instead of searching

- descriptors for selected vertices include

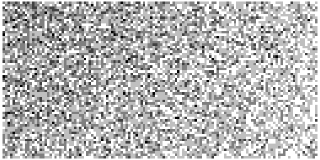


world space location $x$ and optionally normal $n$, incident angle $\omega$, and BRDF layer

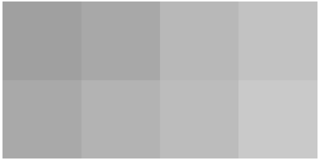
- storing and loading data using hashed quantized descriptors
  - trade a larger hash table size for faster access (proportional to number of paths)
  - use a second hash of the descriptor instead of storing full keys
  - linear probing for collision resolution

# Efficient Implementation

## Stochastic interpolation to resolve quantization artifacts



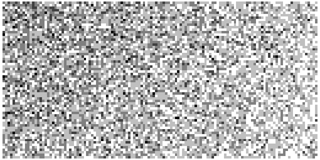
input



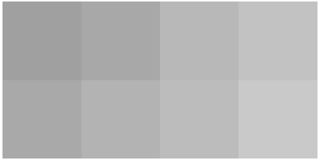
average per cell

# Efficient Implementation

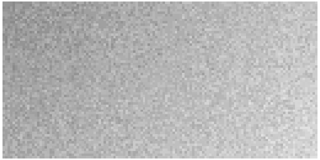
## Stochastic interpolation to resolve quantization artifacts



input



average per cell



with jittering

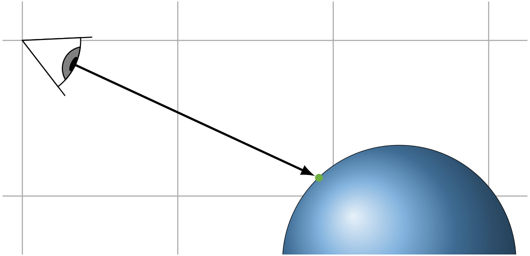
- jitter descriptor ($x_i$, ...) on store and load
  - resulting uniform noise amenable to (existing) post filtering


NVIDIA.

# Efficient Implementation

## Linear instead of quadratic complexity

- finding the hash table location $i$



$$i \leftarrow \text{hash}(\tilde{x}, \ldots) \ \% \ \text{table\_size}$$

for both averaging and querying

# Efficient Implementation

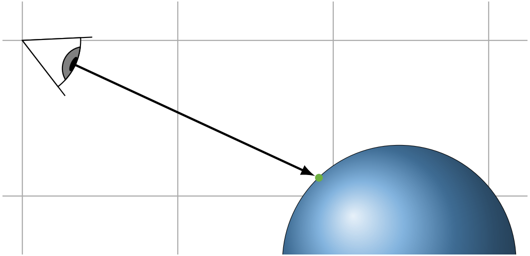
## Linear instead of quadratic complexity

- finding the hash table location $i$

$$i \leftarrow \text{hash}(\tilde{x}, \ldots) \;\; \% \text{ table\_size}$$

$$v \leftarrow \text{hash2}(\tilde{x}, n, \ldots)$$

for both averaging and querying



# Efficient Implementation

## Linear instead of quadratic complexity

- finding the hash table location $i$

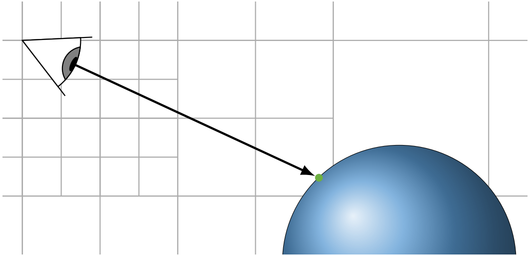
$$l' \leftarrow \text{level\_of\_detail}(|p_{\text{cam}} - x'|)$$

$$\tilde{x} \leftarrow \left\lfloor \frac{x'}{scale \cdot 2^{l'}} \right\rfloor$$

$$i \leftarrow \text{hash}(\tilde{x}, \ldots) \;\; \% \text{ table\_size}$$

$$v \leftarrow \text{hash2}(\tilde{x}, n, \ldots)$$

for both averaging and querying



# Efficient Implementation

## Linear instead of quadratic complexity

- finding the hash table location $i$

$$l \leftarrow \text{level\_of\_detail}(|p_{\mathrm{cam}} - x|)$$

$$x' \leftarrow x + \text{jitter}(n) \cdot scale \cdot 2^{l}$$

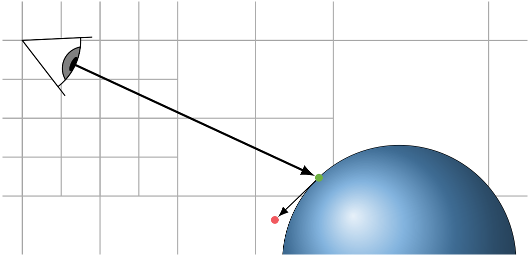
$$l' \leftarrow \text{level\_of\_detail}(|p_{\mathrm{cam}} - x'|)$$

$$\tilde{x} \leftarrow \left\lfloor \frac{x'}{scale \cdot 2^{l'}} \right\rfloor$$

$$i \leftarrow \text{hash}(\tilde{x}, \ldots) \ \% \ \text{table\_size}$$

$$v \leftarrow \text{hash2}(\tilde{x}, n, \ldots)$$

for both averaging and querying



# Efficient Implementation

## Linear instead of quadratic complexity

- finding the hash table location $i$

$$l \leftarrow \text{level\_of\_detail}(|p_{\text{cam}} - x|)$$

$$x' \leftarrow x + \text{jitter}(n) \cdot \mathit{scale} \cdot 2^{l}$$

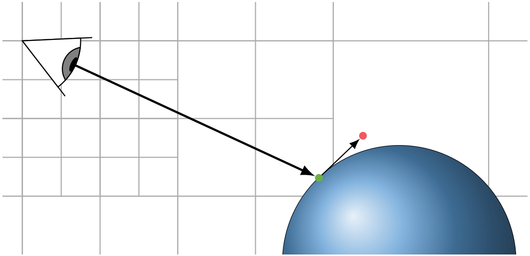
$$l' \leftarrow \text{level\_of\_detail}(|p_{\text{cam}} - x'|)$$

$$\tilde{x} \leftarrow \left\lfloor \frac{x'}{\mathit{scale} \cdot 2^{l'}} \right\rfloor$$

$$i \leftarrow \text{hash}(\tilde{x}, \ldots) \ \% \ \text{table\_size}$$

$$v \leftarrow \text{hash2}(\tilde{x}, n, \ldots)$$

for both averaging and querying



# Efficient Implementation

## Linear instead of quadratic complexity

- finding the hash table location $i$

$$
\begin{aligned}
l' &\leftarrow \text{level\_of\_detail}(|p_{\text{cam}} - x'|) \\
\tilde{x} &\leftarrow \left\lfloor \frac{x'}{scale \cdot 2^{l'}} \right\rfloor \\
i &\leftarrow \text{hash}(\tilde{x}, \ldots)\ \ \%\ \text{table\_size} \\
v &\leftarrow \text{hash2}(\tilde{x}, n, \ldots)
\end{aligned}
$$

for both averaging and querying

# Efficient Implementation

## Linear instead of quadratic complexity

- finding the hash table location $i$

$$l \leftarrow \text{level\_of\_detail}(|p_{\text{cam}} - x|)$$

$$x' \leftarrow x + \text{jitter}(n) \cdot scale \cdot 2^l$$

$$l' \leftarrow \text{level\_of\_detail}(|p_{\text{cam}} - x'|)$$

$$\tilde{x} \leftarrow \left\lfloor \frac{x'}{scale \cdot 2^{l'}} \right\rfloor$$

$$i \leftarrow \text{hash}(\tilde{x}, \ldots) \ \% \ \text{table\_size}$$

$$v \leftarrow \text{hash2}(\tilde{x}, n, \ldots)$$

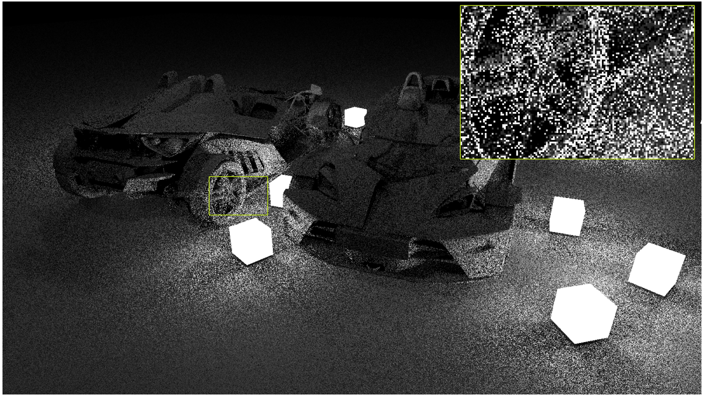
for both averaging and querying

- jittering before quantization hides discretization artifacts in uniform noise

# Results

## Comparison at 16 paths per pixel

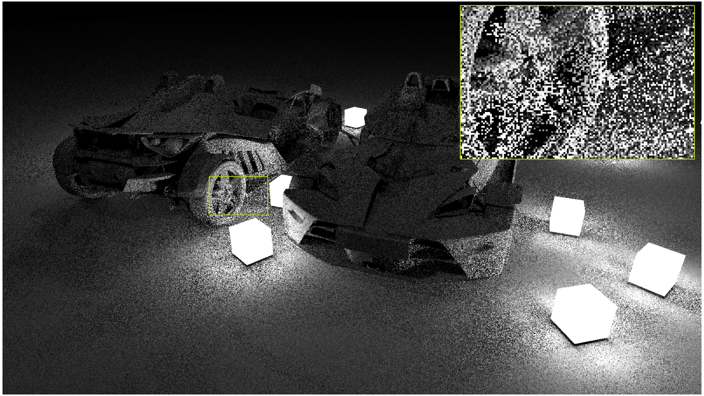
- uniform sampling



# Results

## Comparison at 16 paths per pixel

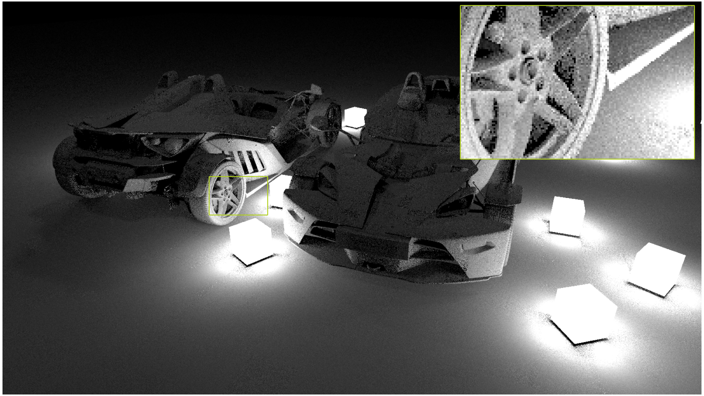
- light hierarchy



# Results

## Comparison at 16 paths per pixel

- multiple importance sampling combining partial CDF including visibility and light hierarchy



# Results

## Comparison at 2 paths of length 3 per pixel with 4 shadow rays each bounce

- uniform sampling vs. light hierarchy vs. new method

# Results

## Comparison at 2 paths of length 3 per pixel with 4 shadow rays each bounce

- uniform sampling vs. light hierarchy vs. new method

# Results

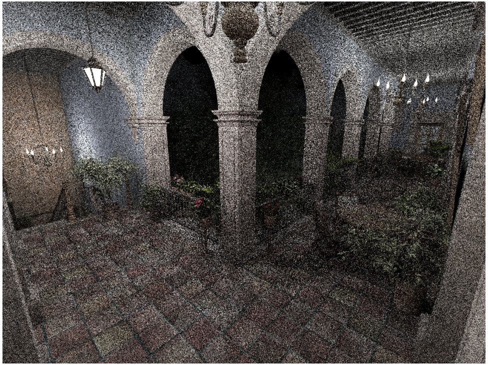
## Comparison at 2 paths of length 3 per pixel with 4 shadow rays each bounce

- uniform sampling vs. light hierarchy vs. new method

## Results

### Comparison at 8 paths of length 4 per pixel

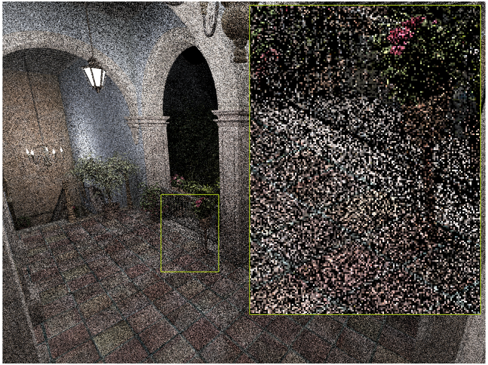
- light hierarchy vs. partial CDF



# Results

## Comparison at 8 paths of length 4 per pixel

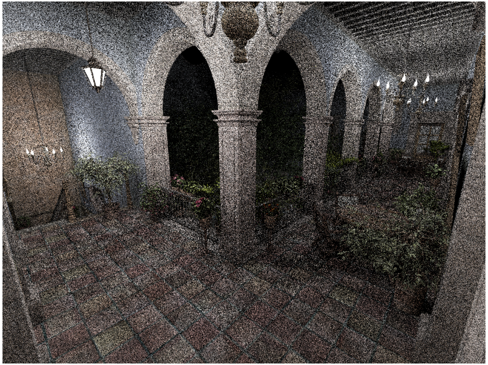
- light hierarchy vs. partial CDF



# Results

## Comparison at 8 paths of length 4 per pixel

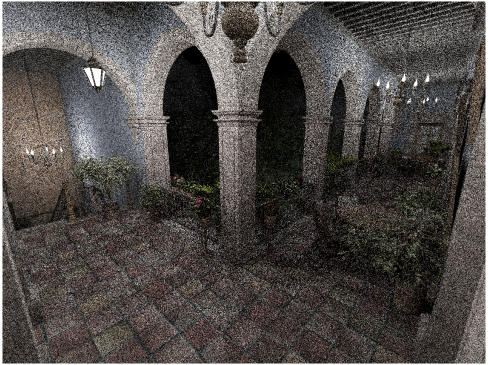
- light hierarchy vs. partial CDF



# Results

## Comparison at 8 paths of length 4 per pixel

- light hierarchy vs. partial CDF with path space filtering



# Guiding and Shadow Rays

## Sampling proportional to integrand including visibility

- level-of-detail hash of partial CDFs
  - light hierarchy as fallback
- stochastic interpolation
  - for both accumulation and sampling
- probabilities determined by either path tracing or light paths (photons)

# Guiding and Shadow Rays

## Sampling proportional to integrand including visibility

- level-of-detail hash of partial CDFs
  - light hierarchy as fallback
- stochastic interpolation
  - for both accumulation and sampling
- probabilities determined by either path tracing or light paths (photons)
- up next
  - learn *b* by gradient descent
  - include bidirectional scattering distribution function