Many-lights rendering

UCSD CSE 272
Advanced Image Synthesis
Tzu-Mao Li
Rendering with many lights
Virtual point lights

- we can deposit point lights at light subpaths in bidirectional path tracing
Goal: importance sample lights

- important lights are:
  - closer
  - at BSDF peak
  - not occluded
  - high intensity
Naive approach: importance sample light intensity

contribution = $L \cdot \rho \cdot G \cdot V$

- camera
- shading point
- occluder

Terms:
- intensity
- geometry term
- BSDF
- visibility
Ideas

hierarchical clustering

data-driven

matrix formulation
[Hasan 2007, Ou 2011, Huo 2015, …]

spatial-temporal reuse + resampling
[Benedikt 2020]
Ideas

hierarchical clustering

matrix formulation
[Hasan 2007, Ou 2011, Huo 2015, …]

data-driven

spatial-temporal reuse + resampling
[Benedikt 2020]
Idea: group light into clusters

- important lights = small clusters
- unimportant lights = large clusters

approximate each cluster with one sample
Use a tree to cluster lights

- important lights = small clusters
- unimportant lights = large clusters
Determining light cut

- start at the root
Determining lightcut

- estimate the importance of the cluster using an upper bound of all contributions
- visibility is ignored

$L \cdot \rho \cdot G \cdot V$
Determining lightcut

- Refine the node with the highest importance and repeat

$L \cdot \rho \cdot G \cdot V$
Determining lightcut

- refine the node with the highest importance and repeat

\[ L \cdot \rho \cdot G \cdot V \]
Sampling light from the lightcut

- uniformly pick a cluster
- sample a light from the cluster by traversing the tree
Sampling light from the lightcut

- uniformly pick a cluster
- sample a light from the cluster by traversing the tree
Sampling light from the lightcut

- uniformly pick a cluster
- sample a light from the cluster by traversing the tree
Sampling light from the lightcut

- uniformly pick a cluster
- sample a light from the cluster by traversing the tree

Stochastic Lightcuts
Lightcuts-like ideas are widely used in practice

Estevez’s

Our method

Energy based sampling

Importance Sampling of Many Lights with Adaptive Tree Splitting

ALEJANDRO CONTY ESTEVEZ, Sony Pictures Imageworks
CHRISTOPHER KULLA, Sony Pictures Imageworks
Lightcuts-like ideas are widely used in practice
Lightcuts-like ideas are widely used in practice

Estevez's

Our method

Importance Sampling of Many Lights with Adaptive Tree Splitting

ALEJANDRO CONTY ESTEVEZ, Sony Pictures Imageworks
CHRISTOPHER KULLA, Sony Pictures Imageworks
Multi-dimensional lightcuts: also cluster the shading points
Multi-dimensional lightcuts: also cluster the shading points
Multi-dimensional lightcuts: also cluster the shading points
Multi-dimensional lightcuts:
also cluster the shading points
Multi-dimensional lightcuts: also cluster the shading points
Multi-dimensional lightcuts:
also cluster the shading points
Multi-dimensional lightcuts: also cluster the shading points
Multidimensional lightcuts is (was?) used by Autodesk

with lightcuts

without lightcuts

Ideas

hierarchical clustering
[Shirley 1996,
Paquette 1998,
Walter 2005, ...]

data-driven
[Donikian 2006,
Vevoda 2018,
Wang 2021]

matrix formulation
[Hasan 2007,
Ou 2011,
Huo 2015, ...]

spatial-temporal reuse + resampling
[Benedikt 2020]
Lightcuts' issue: ignore visibility
A pathological case for lightcuts
Idea: estimate importance of clusters using samples

- improve estimation as we render
Inappropriate clustering leads to noisy sampling

[Yuksel 2019] Stochastic Lightcut
(30 sec rendering)
Our method learns a good light clustering progressively.
Algorithm

group shading
points into cells
Algorithm

group shading points into cells

build light hierarchy & init clustering
group shading points into cells

build light hierarchy & init clustering

sample a cluster using importance
Algorithm

- Group shading points into cells
- Build light hierarchy & init clustering
- Sample a cluster using importance
- Sample a light within the cluster [Yuksel 2019]
group shading points into cells

build light hierarchy & init clustering

sample a cluster using importance

sample a light within the cluster

[Yuksel 2019]

update importance & variance
Algorithm

- Group shading points into cells
- Build light hierarchy & init clustering
- Sample a cluster using importance
- Sample a light within the cluster [Yuksel 2019]
- Split cluster if variance is large
- Update importance & variance
Algorithm

group shading points into cells

build light hierarchy & init clustering

sample a cluster using importance

split cluster if variance is large

sample a light within the cluster

update importance & variance

[Yuksel 2019]
How do we initialize/update the importance?

- key idea: initialize the importance using lightcut upper bound, update with data
Key idea: initialize the importance using lightcut upper bound, update with data

\[ Q_0(c) = \text{lightcuts importance} \]

\[ Q_{t+1}(c) = (1 - \alpha_t)Q_t(c) + \alpha_t(\text{sampling contribution}) \]

goal: \( Q_t(c) \) converges to the sum of contributions of lights in cluster \( c \)
Key idea: initialize the importance using lightcut upper bound, update with data

\[ Q_0(c) = \text{lightcuts importance} \]

\[ Q_{t+1}(c) = (1 - \alpha_t)Q_t(c) + \alpha_t(\text{sampling contribution}) \]

goal: \( Q_t(c) \) converges to the sum of contributions of lights in cluster \( c \)
Need to be very careful with the “learning rate” $\alpha_t$

$$Q_{t+1}(c) = (1 - \alpha_t)Q_t(c) + \alpha_t({\text{sampling contribution}})$$

converges to the sum of contribution when

$$\sum_{t=1}^{\infty} \alpha_t = \infty \text{ and } \sum_{t=1}^{\infty} \alpha_t^2 < \infty$$

($\alpha_t = \text{constant doesn't work!}$)

Stochastic Approximation
[Robbins and Monro 1951]
Need to be very careful with the “learning rate” $\alpha_t$

\[ Q_{t+1}(c) = (1 - \alpha_t)Q_t(c) + \alpha_t \text{(sampling contribution)} \]

converges to the sum of contribution when

\[ \sum_{t=1}^{\infty} \alpha_t = \infty \text{ and } \sum_{t=1}^{\infty} \alpha_t^2 < \infty \]

we pick $\alpha_t = \frac{1}{at^b}$

($\alpha_t = \text{constant doesn’t work!}$)  

Stochastic Approximation  
[Robbins and Monro 1951]
Using a constant $\alpha_t$ can lead to visual artifacts!

$$Q_{t+1}(c) = (1 - \alpha_t)Q_t(c) + \alpha_t(\text{sampling contribution})$$
We made data-driven methods robust.

4776 lights, direct lighting only

<table>
<thead>
<tr>
<th>method:</th>
<th>relMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>stochastic lightcuts [Yuksel 2019]</td>
<td>0.152</td>
</tr>
<tr>
<td>Bayesian online [Vevoda 2018]</td>
<td>0.095</td>
</tr>
<tr>
<td>variance-aware Bayesian [Rath 2020]</td>
<td>0.101</td>
</tr>
<tr>
<td>reinforcement lightcuts [Pantaleoni 2019]</td>
<td>0.065</td>
</tr>
<tr>
<td>ours</td>
<td>0.057</td>
</tr>
<tr>
<td>ref</td>
<td></td>
</tr>
</tbody>
</table>
We made data-driven methods robust

indirect illumination rendered with 71311 virtual point lights

<table>
<thead>
<tr>
<th>method</th>
<th>relMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>stochastic lightcuts [Yuksel 2019]</td>
<td>0.352</td>
</tr>
<tr>
<td>Bayesian online [Vevoda 2018]</td>
<td>1.034</td>
</tr>
<tr>
<td>variance-aware Bayesian [Rath 2020]</td>
<td>0.476</td>
</tr>
<tr>
<td>reinforcement lightcuts [Pantaleoni 2019]</td>
<td>0.404</td>
</tr>
<tr>
<td>ours</td>
<td>0.050</td>
</tr>
<tr>
<td>ref</td>
<td></td>
</tr>
</tbody>
</table>
We made data-driven methods robust

90862 lights, direct illumination only

<table>
<thead>
<tr>
<th>method</th>
<th>relMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>stochastic lightcuts [Yuksel 2019]</td>
<td>0.153</td>
</tr>
<tr>
<td>Bayesian online [Vevoda 2018]</td>
<td>0.766</td>
</tr>
<tr>
<td>variance-aware Bayesian [Rath 2020]</td>
<td>0.480</td>
</tr>
<tr>
<td>reinforcement lightcuts [Pantaleoni 2019]</td>
<td>0.237</td>
</tr>
<tr>
<td>ours</td>
<td></td>
</tr>
<tr>
<td>ref</td>
<td>0.047</td>
</tr>
</tbody>
</table>
Ideas

hierarchical clustering

data-driven

matrix formulation
[Hasan 2007, Ou 2011, Huo 2015, ...]

spatial-temporal reuse + resampling
[Benedikt 2020]
Many-lights rendering = estimating the light transport matrix

figure from Ou 2011
Many-lights rendering = estimating the light transport matrix

observation: the light transport matrix is low-rank!

figure from Ou 2011
Idea: reconstruct the light transport matrix by sampling rows and columns.
Row/column sampling can be done using rasterization/shadow mapping!

column sampling = render a point light for all pixels

column sampling = render a pixel with all lights
Result: high-quality global illumination only using rasterization!

2.2m triangles: 300 rows, 900 columns, 16.9 s
388k triangles: 432 rows, 864 columns, 13.5 s
869k triangles: 100 rows, 200 columns, 3.8 s
Followup: applying matrix completion algorithms for light transport matrix estimation

A Matrix Sampling-and-Recovery Approach for Many-Lights Rendering

Yuchi Huo  Rui Wang*  Shihao Jin  Xinguo Liu  Hujun Bao*
State Key Lab of CAD&CG, Zhejiang University

Matrix Recovery by Matrix Separation. Matrix separation has been recently developed [Candès et al. 2011; Shen et al. 2014]. Specifically in our scenario, the reduced lighting matrix $L$ can be separated from the corrupted matrix $D$ with a sparse error matrix $Z$, $D = L + Z$, by solving the following minimization:

$$\min_{L,Z} \|L\|_* + \lambda \|Z\|_1$$

$$\text{s.t.} \quad P_{T_l}(L + Z) = P_{T_l}(D)$$

(5)
Ideas

hierarchical clustering

data-driven

matrix formulation
[Hasan 2007, Ou 2011, Huo 2015, …]

spatial-temporal reuse + resampling
[Benedikt 2020]
Motivation: real-time rendering

Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting

BENEDIKT BITTERLI, Dartmouth College
CHRIS WYMAN, NVIDIA
MATT PHARR, NVIDIA
PETER SHIRLEY, NVIDIA
AARON LEFOHN, NVIDIA
WOJIECH JAROSZ, Dartmouth College
Idea: reuse neighboring pixels’ sampling results

- each pixel starts with a single light sampled (can use lightcuts or whatever)
Idea: reuse neighboring pixels’ sampling results

- each pixel starts with a single light sampled (can use lightcuts or whatever)

- for the center pixel, pick the unoccluded lights from neighbor pixels
Idea: reuse neighboring pixels’ sampling results

• each pixel starts with a single light sampled (can use lightcuts or whatever)

• for the center pixel, pick the unoccluded lights from neighbor pixels

• sample from these lights using probability proportional to $L \cdot \rho \cdot G$
Idea: reuse neighboring pixels' sampling results

- each pixel starts with a single light sampled (can use lightcuts or whatever)
- for the center pixel, pick the unoccluded lights from neighbor pixels
- sample from these lights using probability proportional to $L \cdot \rho \cdot G$
- can propagate the information to the next frame
Idea: reuse neighboring pixels’ sampling results

- benefits
  - occluded lights have low probability to be sampled
  - $\rho$ & $G$ are considered during reuse
  - sampling distribution is improved over time

- each pixel starts with a single light sampled (can use lightcuts or whatever)

- for the center pixel, pick the unoccluded lights from neighbor pixels

- sample from these lights using probability proportional to $L \cdot \rho \cdot G$

- can propagate the information to the next frame
What are the connections between these ideas?

Hierarchical clustering

Data-driven

Matrix formulation
[Hasan 2007, Ou 2011, Huo 2015, …]

Spatial-temporal reuse + resampling
[Benedikt 2020]
Next: ReSTIR and Path-reusing