Lightcuts

Jan Kautz, 2009-2010

Lightcuts

- Efficient, accurate complex illumination

Environment map lighting & indirect
Time 111s

Textured area lights & indirect
Time 98s

(640x480, Anti-aliased, Glossy materials)

Scalable

- Scalable solution for many point lights
  - Thousands to millions
  - Sub-linear cost

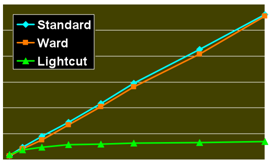


Tableau Scene

Complex Lighting

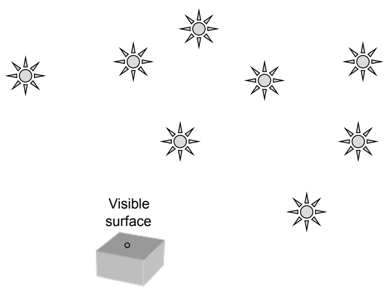
- Simulate complex illumination using point lights
  - Area lights
  - HDR environment maps
  - Sun & sky light
  - Indirect illumination
- Unifies illumination
  - Enables tradeoffs between components

Area lights + Sun/sky + Indirect

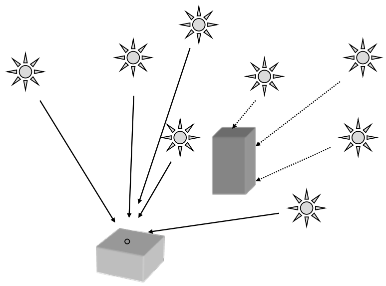
Overview

- Problem
- Pre-Process
  - Convert Illumination to Point Lights
  - Tree Building
- Run-Time
  - Cut

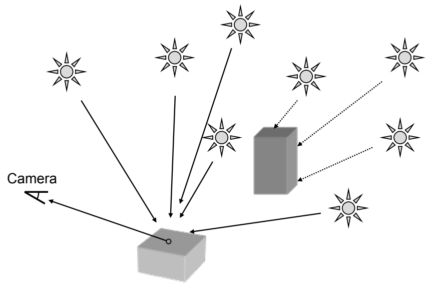
Lightcuts Problem



Lightcuts Problem

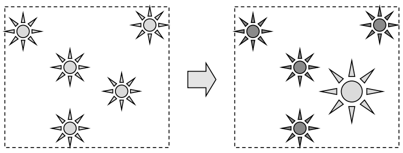


Lightcuts Problem



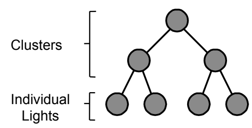
Key Concepts

- Light Cluster
  - Approximate many lights by a single brighter light (the representative light)



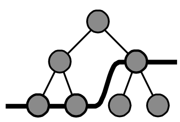
Key Concepts

- Light Cluster
- Light Tree
  - Binary tree of lights and clusters

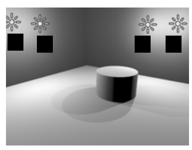


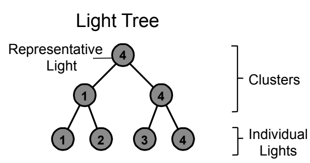
Key Concepts

- Light Cluster
- Light Tree
- A Cut
  - A set of nodes that partitions the lights into clusters

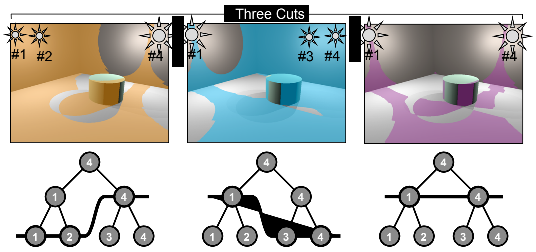


Simple Example

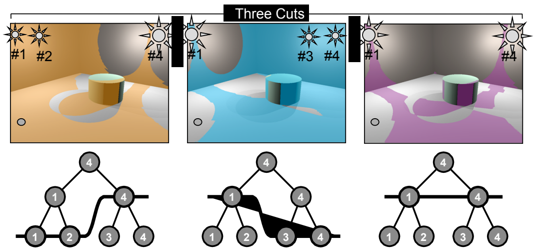




Three Example Cuts

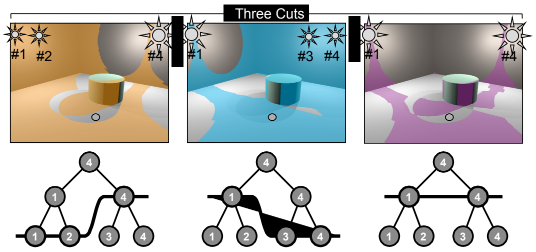


Three Example Cuts



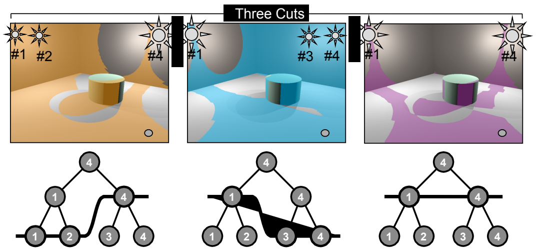
Good Bad Bad

Three Example Cuts



Bad Good Bad

Three Example Cuts



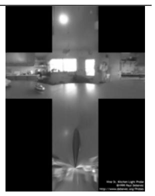
Good Good Good

Algorithm Overview

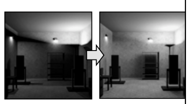
- Pre-process
  - Convert illumination to point lights
  - Build light tree
- For each eye ray
  - Choose a cut to approximate the illumination

Convert Illumination

- HDR environment map
  - Apply captured light to scene
  - Convert to directional point lights using [Agarwal et al. 2003]



- Indirect Illumination
  - Convert indirect to direct illumination using Instant Radiosity [Keller 97]
    - Caveats: no caustics, clamping, etc.
  - More lights = more indirect detail



Algorithm Overview

- Pre-process
  - Convert illumination to point lights
  - Build light tree
- For each eye ray
  - Choose a cut to approximate the local illumination
    - Cost vs. accuracy
    - Avoid visible transition artifacts

Perceptual Metric

- Weber's Law
  - Contrast visibility threshold is fixed percentage of signal
  - Used 2% in our results
- Ensure each cluster's error < visibility threshold
  - Transitions will not be visible
  - Used to select cut

Illumination Equation

$$\text{result} = \sum_{\text{lights}} M_i \; G_i \; V_i \; I_i$$

M_i – Material term; G_i – Geometric term; V_i – Visibility term; I_i – Light intensity

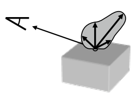
More formal:

$$L(x,\omega_r) = [L_e(x,\omega_r)] + \sum_i f_r(\omega_r,\omega_i)\,\omega_i \cdot n \frac{1}{\| y_i - x \|^2} V(x,\omega_i) I(y_i)$$

Illumination Equation

$$\text{result} = \sum_{\text{lights}} M_i \; G_i \; V_i \; I_i$$

M_i – Material term; G_i – Geometric term; V_i – Visibility term; I_i – Light intensity

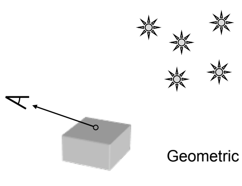


Material term is the BRDF: $f_r(\omega_r,\omega_i)\,\omega_i \cdot n$

Illumination Equation

$$\text{result} = \sum_{\text{lights}} M_i \; G_i \; V_i \; I_i$$

M_i – Material term; G_i – Geometric term; V_i – Visibility term; I_i – Light intensity

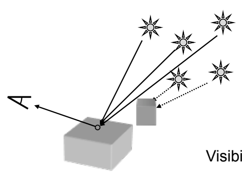


Geometric term is (for point lights): $\frac{1}{\| y_i - x \|^2}$

Illumination Equation

$$\text{result} = \sum_{\text{lights}} M_i \; G_i \; V_i \; I_i$$

M_i – Material term; G_i – Geometric term; V_i – Visibility term; I_i – Light intensity

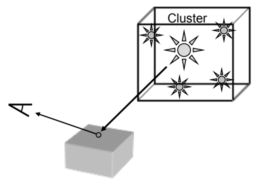


Visibility is defined as: $V(x,\omega_i) = \begin{cases} 1 \\ 0 \end{cases}$

Cluster Approximation

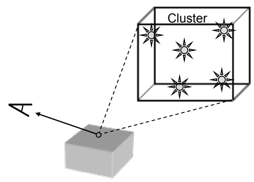
$$\text{result} \approx M_j \; G_j \; V_j \sum_{\text{lights}} I_i$$

j is the representative light



Cluster Error Bound

$$\text{error} \leq M_{ub} \; G_{ub} \; V_{ub} \sum_{\text{lights}} I_i$$

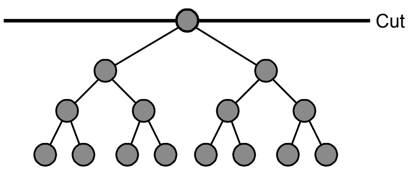


- Bound each term
  - Visibility <= 1 (trivial)
  - Intensity is known
  - Bound material and geometric terms using cluster bounding volume

ub == upper bound

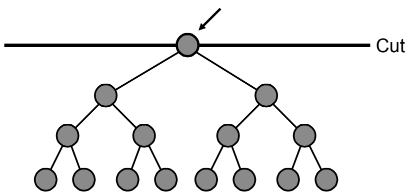
Cut Selection Algorithm

- Start with coarse cut (eg, root node)



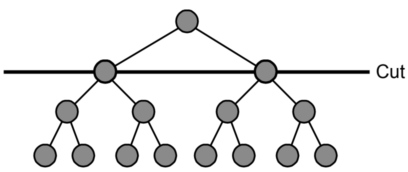
Cut Selection Algorithm

- Select cluster with largest error bound

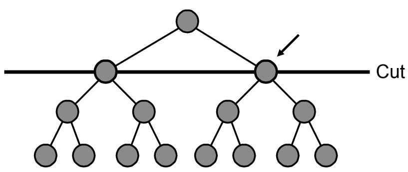


Cut Selection Algorithm

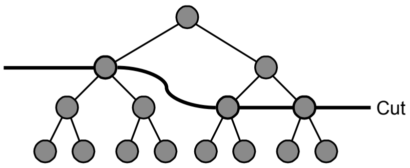
- Refine if error bound > 2% of total



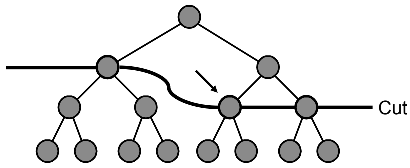
Cut Selection Algorithm



Cut Selection Algorithm

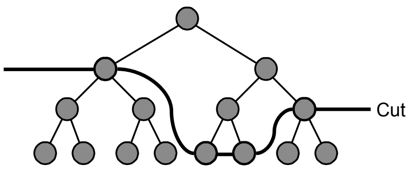


Cut Selection Algorithm



Cut Selection Algorithm

- Repeat until cut obeys 2% threshold



Kitchen, 388K polygons, 4608 lights (72 area sources)

Kitchen, 388K polygons, 4608 lights (72 area sources)

Combined Illumination

Lightcuts 128s	Lightcuts 290s
4 608 Lights (Area lights only)	59 672 Lights (Area + Sun/sky + Indirect)

Combined Illumination

Lightcuts 128s

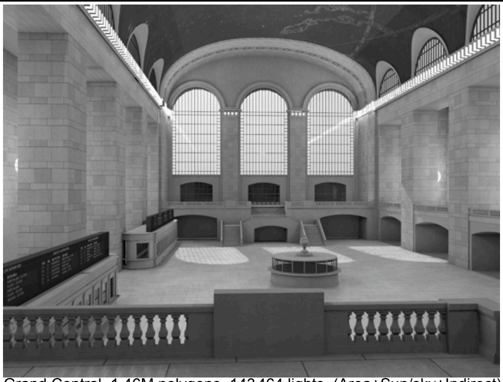
4 608 Lights
(Area lights only)

Avg. 259 shadow rays / pixel

Lightcuts 290s

59 672 Lights
(Area + Sun/sky + Indirect)

Avg. 478 shadow rays / pixel
(only 54 to area lights)

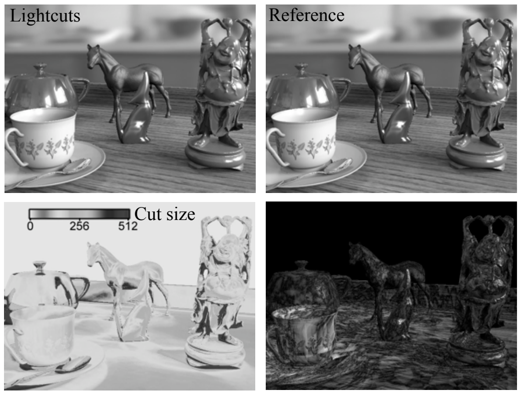


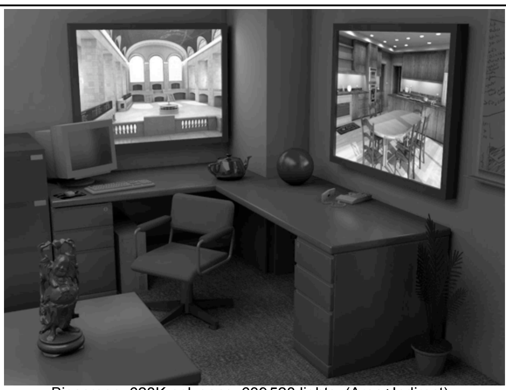
Grand Central, 1.46M polygons, 143464 lights, (Area+Sun/sky+Indirect)

Avg. shadow rays per eye ray 46 (**0.03%**)

Tableau, 630K polygons, 13000 lights, (EnvMap+Indirect)

Avg. shadow rays per eye ray 17 (**0.13%**)

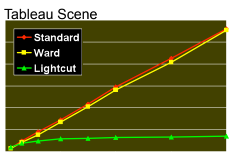


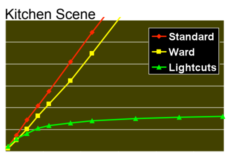


Bigscreen, 628K polygons, 639528 lights, (Area+Indirect)
Avg. shadow rays per eye ray 17 (**0.003%**)

Scalable

- Scalable solution for many point lights
  - Thousands to millions
  - Sub-linear cost





Summary

- Unified illumination handling
- Scalable solution for many lights
  - Locally adaptive representation (the cut)
- Analytic cluster error bounds
  - Most important lights always sampled
- Perceptual visibility metric

The End