Lecture 15: Many Lights

CS 6620, Spring 2009

Kavita Bala
Computer Science
Cornell University

Many Lights

- Most techniques work for a single light source
- Many light sources
  - For environment maps
  - For indirect illumination
- Treat it is a single integration domain
  - Importance sample lights
  - Importance sampling (with visibility) still hard problem

Research on many lights

- Ward ‘91
- Shirley, Wang, Zimmerman ‘94
- Fernandez, Bala, Greenberg ’02
  - Donikian, Fernandez.. ‘06
- Lightcuts ‘05

Shirley, Wang, Zimmerman ‘94

- Try to avoid linear cost of evaluating lights
- Separate lights into
  - Set of important lights (a small set)
  - Set of “dim” lights (large set)
- Construct pdf using:
  - all important lights
  - 1 out of all the dim lights
- Importance sample these lights

Shirley, Wang, Zimmerman ‘94

- Region of influence for important lights
  - Octree cells in region of influence have light in important set
- However, the partitioning into important and dim sets remains hard
- Also, still are not taking visibility into account

Lightcuts [SIGGRAPH ’05,’06]

- Walter, Fernandez, Arbree, Bala
- 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

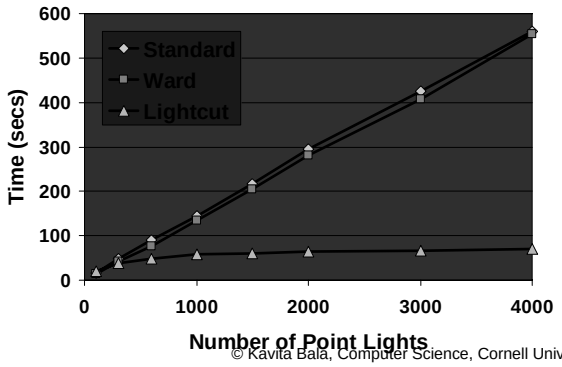


Tableau Scene

GI as many-point lights

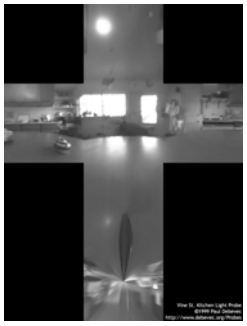
- Approach: unify illumination
  - Area lights
  - HDR environment maps
  - Sun & sky light
  - Indirect illumination
- Convert to point lights

Kitchen light: area, sun/sky, indirect

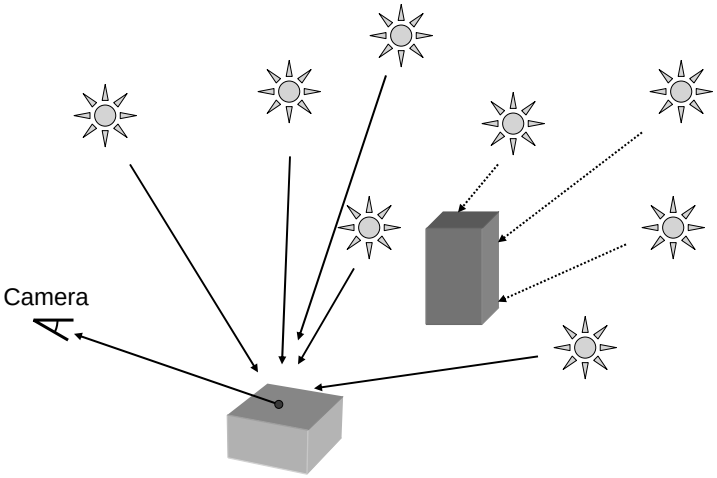
$$P = \int_{hemisphere} (indirect + direct)$$

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



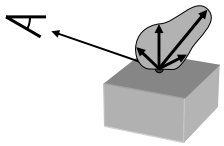
Lightcuts Problem



Illumination Equation

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

Material term | Geometric term | Visibility term | Light intensity

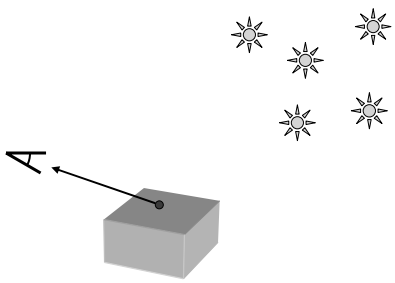


Currently support diffuse, phong, and Ward

Illumination Equation

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

Material term | Geometric term | Visibility term | Light intensity



Illumination Equation

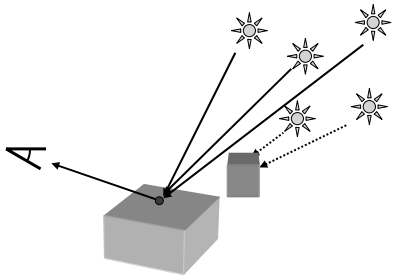
$$\text{result} = \sum_{\text{lights}} M_i \; G_i \; V_i \; I_i$$

Material term

Geometric term

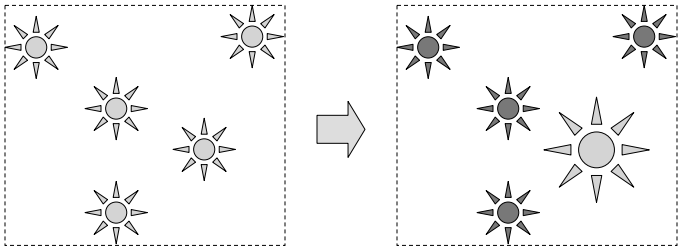
Visibility term

Light intensity



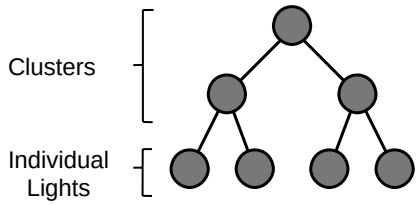
Key Concepts

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



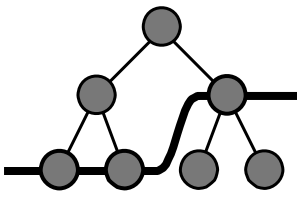
Key Concepts

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

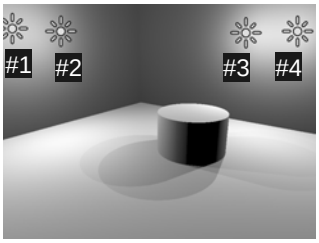


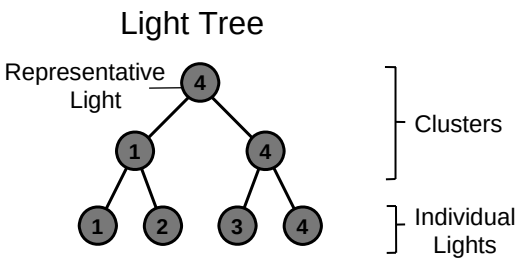
Key Concepts

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

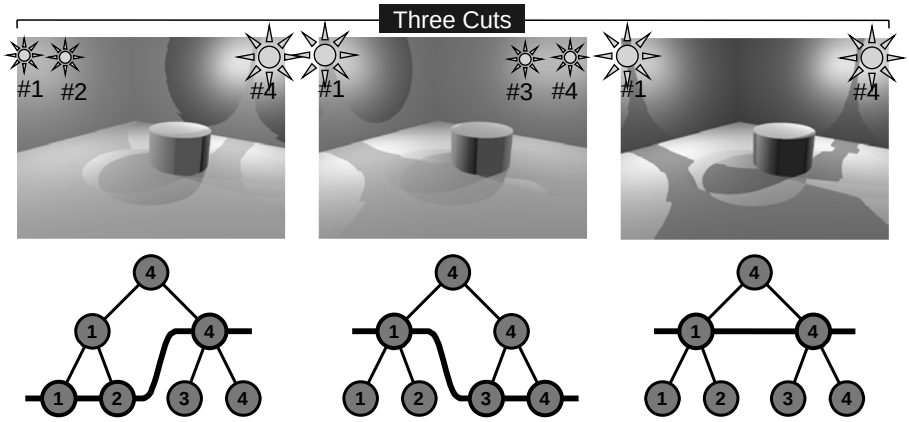


Simple Example

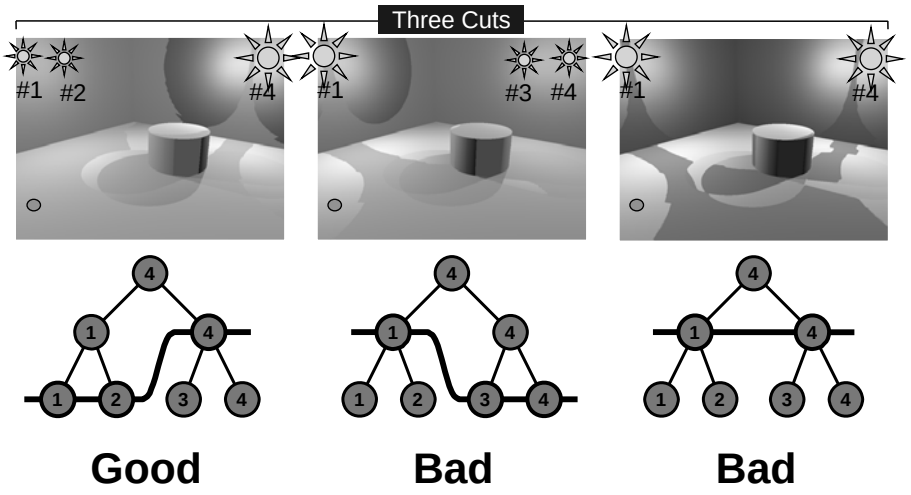




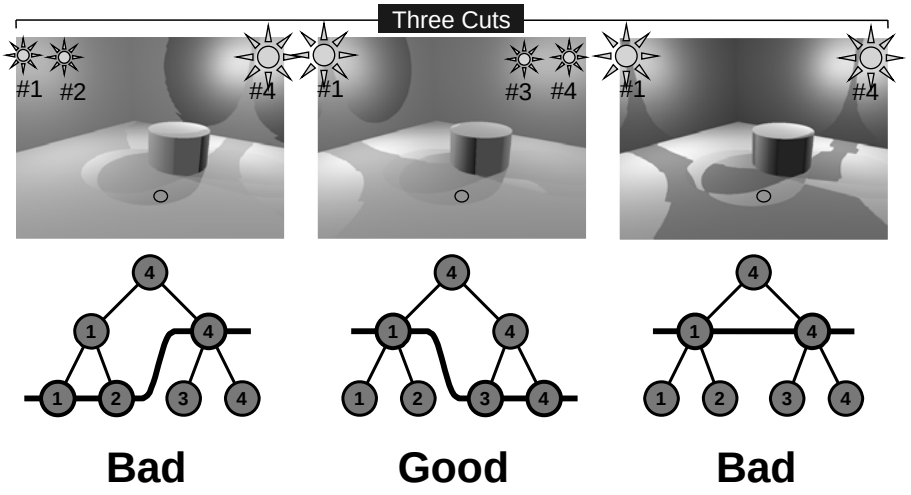
Three Example Cuts



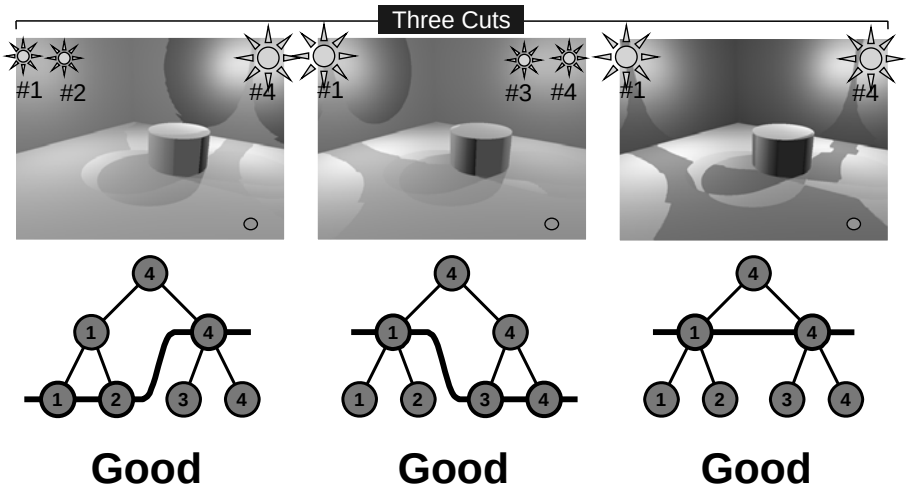
Three Example Cuts



Three Example Cuts



Three Example Cuts



Good **Good** **Good**

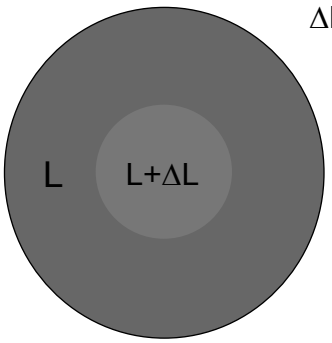
Algorithm Overview

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

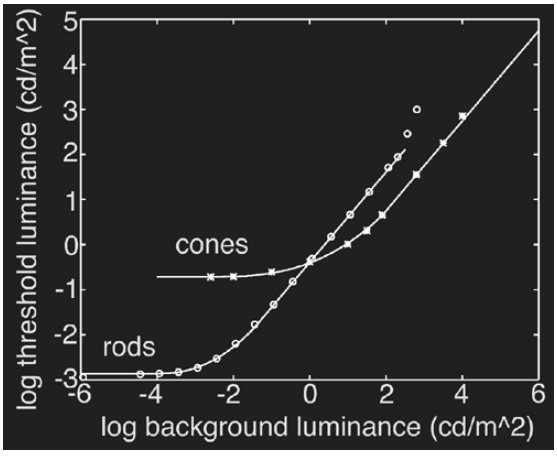
Algorithm Overview

- 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

Threshold visibility



ΔL



ΔL = kL Weber's law

TVI functions

L

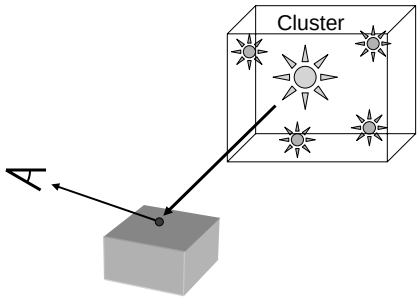
Perceptual Metric

- 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

Cluster Approximation

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

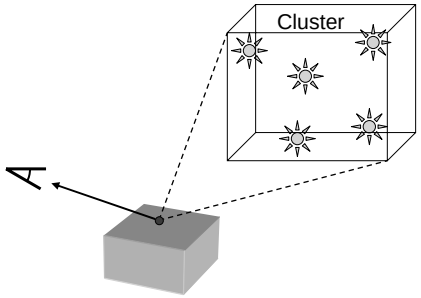
j is the representative light



Cluster Error Bound

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

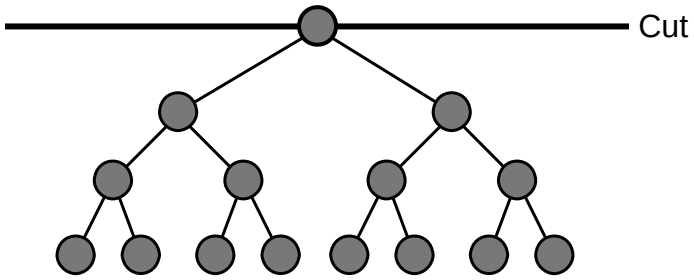
- Bound each term
  - Visibility <= 1 (trivial)
  - Intensity is known
  - Bound material and geometric terms using cluster bounding volume



ub == upper bound

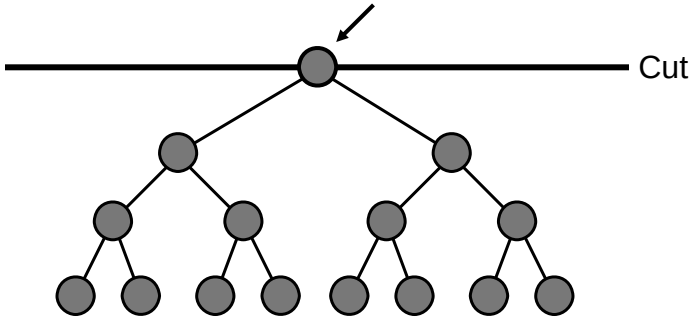
Cut Selection Algorithm

- Start with coarse cut (eg, root node)



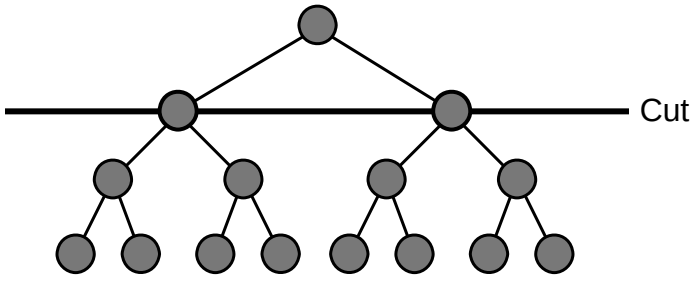
Cut Selection Algorithm

- Select cluster with largest error bound

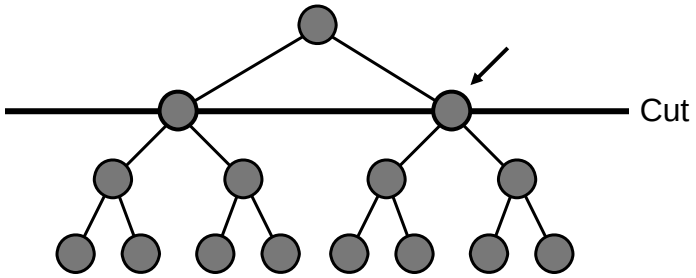


Cut Selection Algorithm

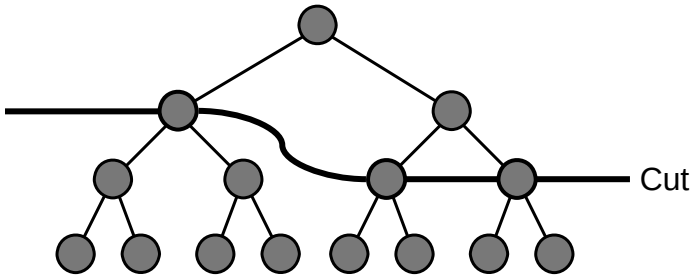
- Refine if error bound > 2% of total



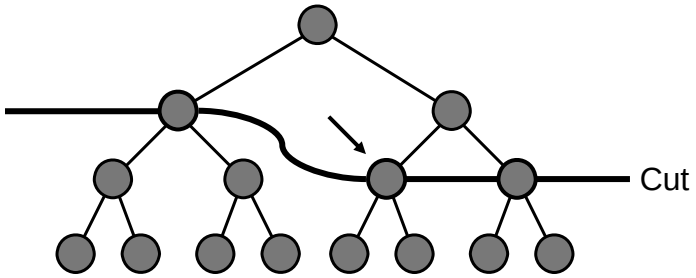
Cut Selection Algorithm



Cut Selection Algorithm

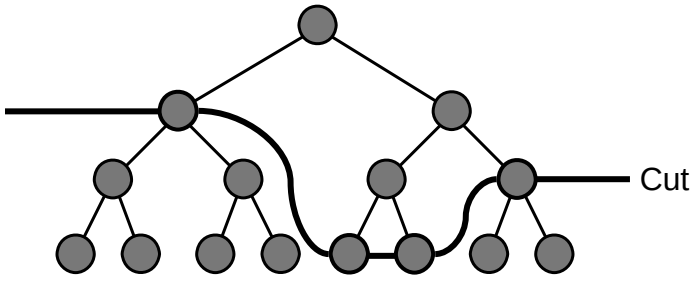


Cut Selection Algorithm



Cut Selection Algorithm

- Repeat until cut obeys 2% threshold



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

Kitchen, 388K polygons, 59,672 Lights

Kitchen, shadow ray false color

Combined Illumination

Lightcuts 128s

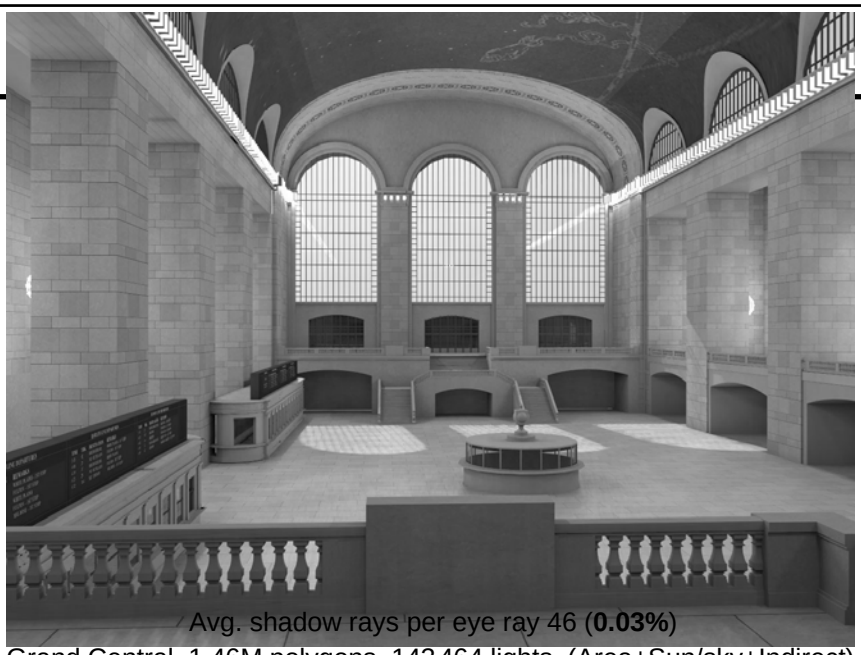
4 608 Lights
(Area lights only)

Avg. 259 shadow rays / pixel

Lightcuts 290s

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

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



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

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

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

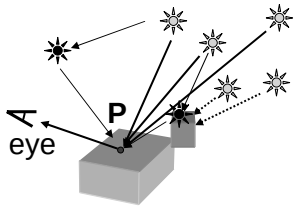
Tableau, 630K polygons, 13000 lights, (EnvMap+Indirect)

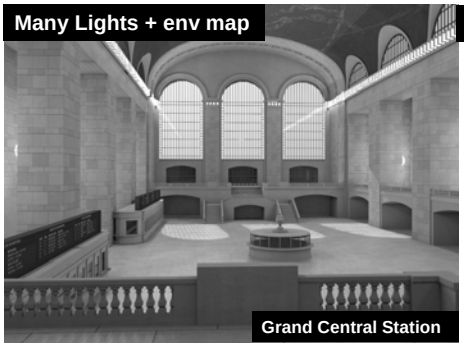
Avg. shadow rays per eye ray 17 (**0.003%**)

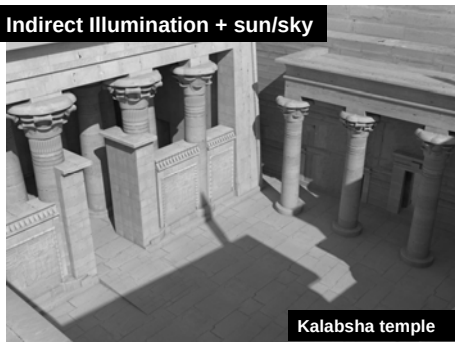
Bigscreen, 628K polygons, 639528 lights, (Area+Indirect)

Problem: Many Lights and GI

$$\int_{hemisphere} (indirect + direct)$$

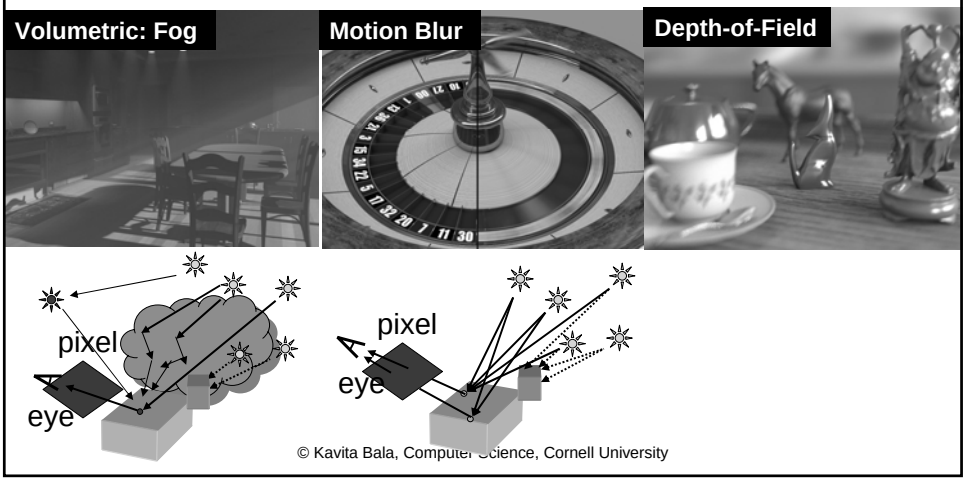






AA, Volumetric, Motion Blur, DOF

Pixel = $\int_{aperture}$ $\int_{time}$ $\int_{volume}$



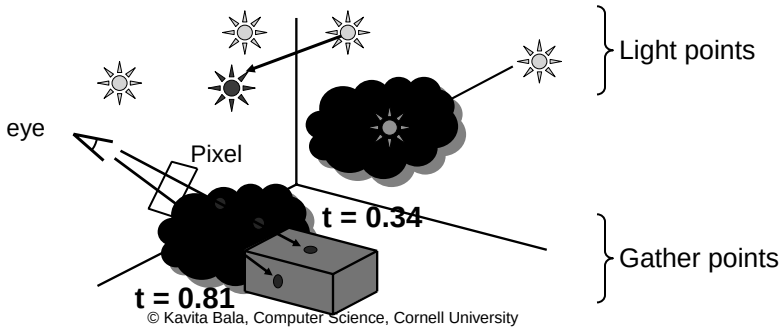
How to scale to complexity?

More complexity is Less visually salient

Multidimensional LC[SIG'06]

Pixel = $\int_{aperture} \int_{time} \int_{volume}$

- Discretize full integral into 2 point sets
  - Light points ($\mathbf{L}$)
  - Gather points ($\mathbf{G}$)

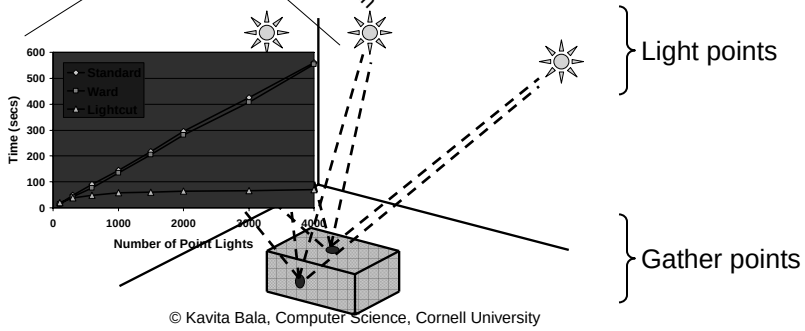


Multidimensional problem

- Sum of all pairs of gather and light points

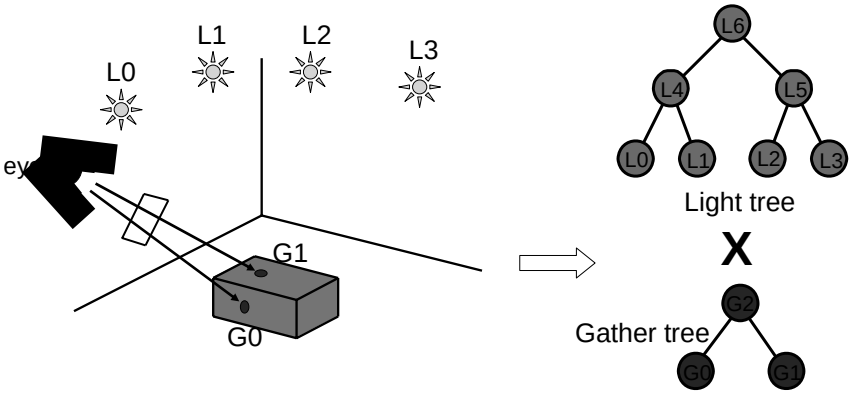
$$\text{Pixel} = \sum_{(j,i) \in \mathbf{G} \times \mathbf{L}} S_j \; k_{ji} \; I_i$$

Gather strength · Weight of ray · Light intensity



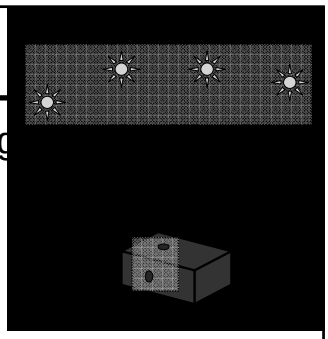
Product Graph

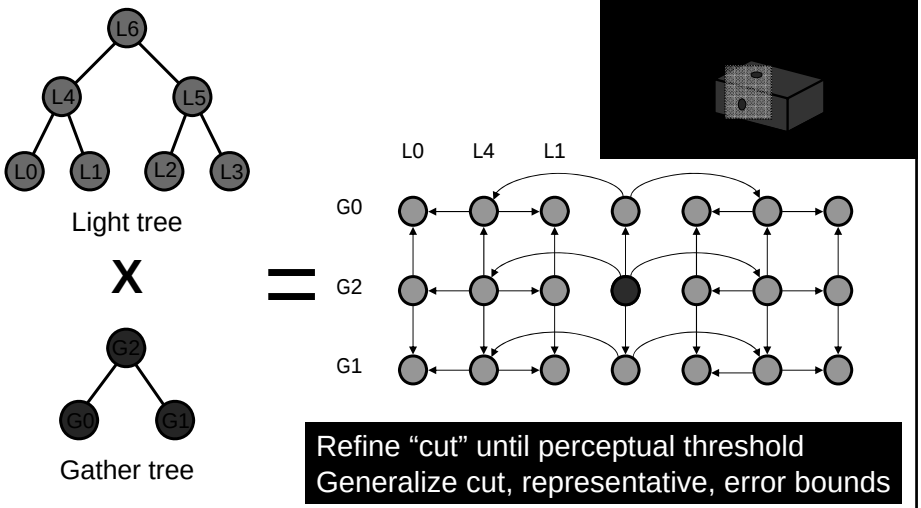
- Explicit hierarchy would be too expensive
- Use implicit hierarchy: product graph



Product Graph

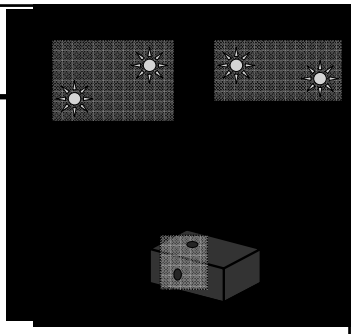
Cartesian product of two trees (g

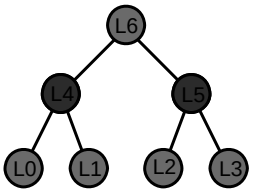




Refine “cut” until perceptual threshold
Generalize cut, representative, error bounds

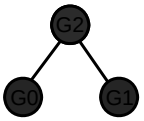
Product Graph





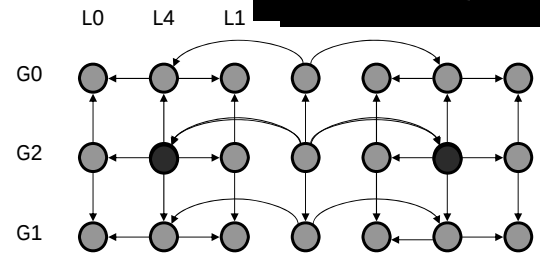
Light tree

X



Gather tree

=



Refine “cut” until perceptual threshold
Generalize cut, representative, error bounds

Summary

- Unified illumination handling
- Scalable solution for many lights
  - Locally adaptive representation (the cut)
- Perceptual visibility metric