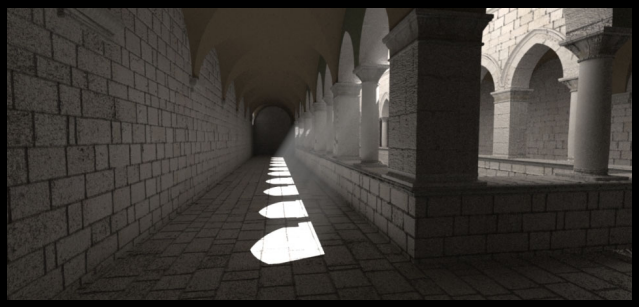
Global Illumination II



70001 – Advanced Computer Graphics: Photographic Image Synthesis

Abhijeet Ghosh

Lecture 15, Feb. 23rd 2024

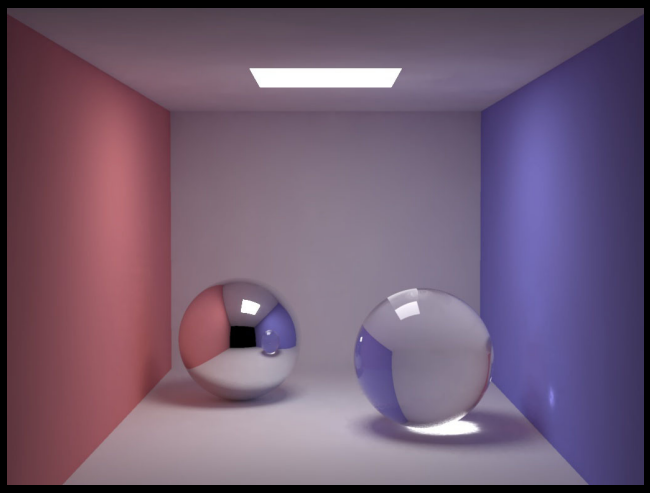
1

Biased approaches

- Reuse previously computed results
  - Assuming slow change in radiance
- Efficient computation of smooth noise-free images
  - Price: unpredictable behavior!
- Irradiance & radiance caching
- Photon mapping

2

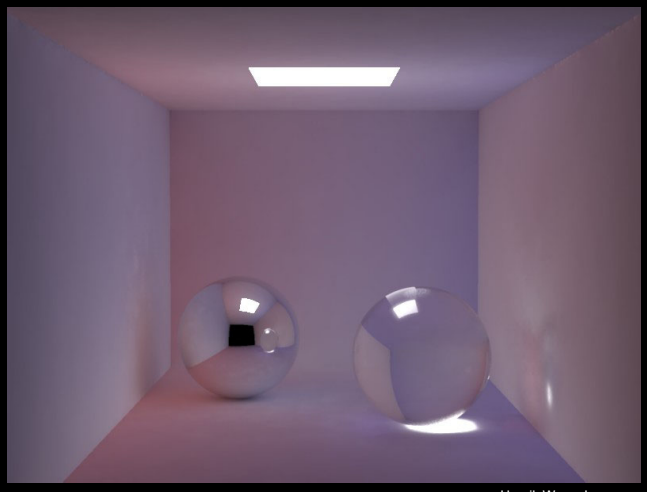
Global Illumination



Henrik Wann Jensen

3

Indirect Irradiance



Henrik Wann Jensen

4

Irradiance caching – Ward et al. 98

- Indirect lighting changes slowly compared to direct lighting
  - especially for diffuse surfaces
- Compute at sparse points and interpolate!
- Two issues:
  - How to store an irradiance estimate?
  - When to sample new estimate?

5

Irradiance caching

- Irradiance

  $E(x,n) = \int_\Omega L_i(x,\omega_i) \cos\theta_i \, d\omega_i.$

- Radiance

  $L_r(x,\omega_r) = \int_\Omega f_r(x, \omega_r, \omega_i) \, L_i(x,\omega_i) \cos\theta_i \, d\omega_i.$

- Diffuse surface: BRDF constant c

  $L_r(x,\omega_r) = c \int_\Omega L_i(x,\omega_i) \cos\theta_i \, d\omega_i = c \cdot E(x,n)$

6

Irradiance caching

- First compute E(x,n) with MC path tracing

$$E(x,n) = \frac{1}{N} \sum_j \frac{L_i(x,\omega_j) \cos\theta_j}{p(\omega_j)}$$

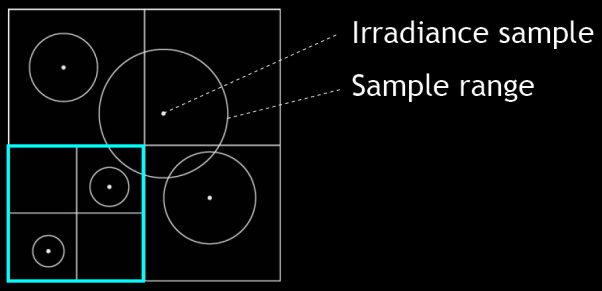
- Sample from cosine weighted distribution $p(\omega_j) = \cos\theta/\pi$

$$E(x,n) = \frac{\pi}{N} \sum_j L_i(x,\omega_j)$$

- Irradiance estimates stored in an octree data-structure

7

Octree example



- Adaptive subdivision such that each cell contains < 3 samples

8

Irradiance sampling

- Each sample is assigned a range for interpolation
- Rate of change of irradiance decides interpolation distance
  - inversely proportional
- Depends on distance to visible surfaces
- Harmonic Mean $\frac{N}{\sum_{i=1}^{N} 1/d_i}$
  - d_i is the distance to first intersection along path i

9

Irradiance sample

```
struct irradiance_sample {
  vector3 E  // irradiance
  vector3 n  // normal
  vector3 p  // position
  float r    // range
}
```

10

Irradiance interpolation

- Compute the error in estimate when using a cached sample

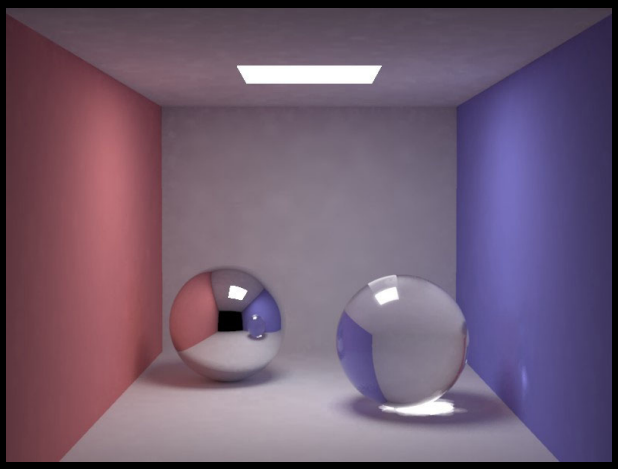
$$w_j(\mathbf{x}) = \frac{1}{\frac{\|\mathbf{x}-\mathbf{x}_j\|}{r_j} + \sqrt{1 - \mathbf{n}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}_j)}} \approx \epsilon_j(\mathbf{x})$$

- If error is acceptable, use a weighted sum of interpolated irradiance

$$E(\mathbf{x}) = \frac{\sum_i w_i(\mathbf{x}) E(\mathbf{x}_i)}{\sum_i w_i(\mathbf{x})}$$

11

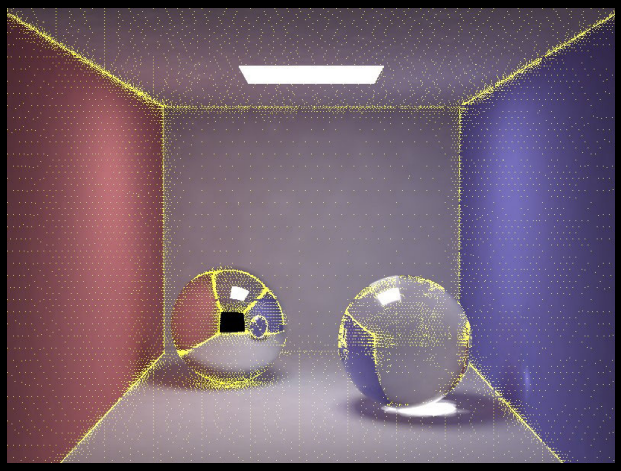
Irradiance gradients



1000 sample rays, w > 10

12

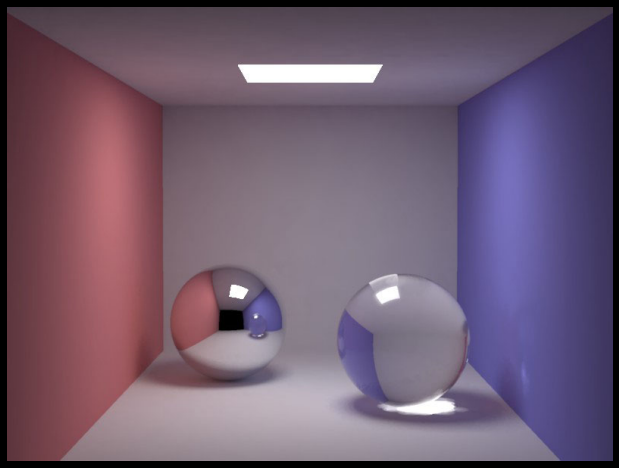
Irradiance cache positions



1000 sample rays, w > 10

13

Irradiance gradients



5000 sample rays, w > 10

14

Glossy surfaces?

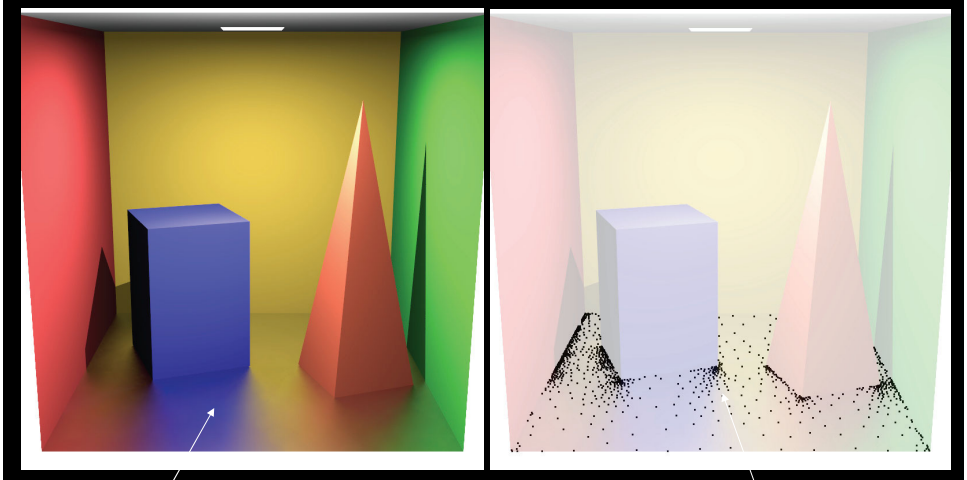
- Irradiance caching works well for diffuse surfaces
- For glossy surfaces we need to account for the directional dependence of the BRDF!

$$L_r(x,\omega_r) \approx \left(\int_\Omega f_r(x, \omega_r, \omega_i)\, d\omega_i\right)\left(\int_\Omega L_i(x,\omega_i) \cos\theta_i\, d\omega_i\right)$$

$$\approx \rho_{hd}(\omega_r) \cdot E(x,n)$$

15

Radiance caching – Křivánek et al. 05

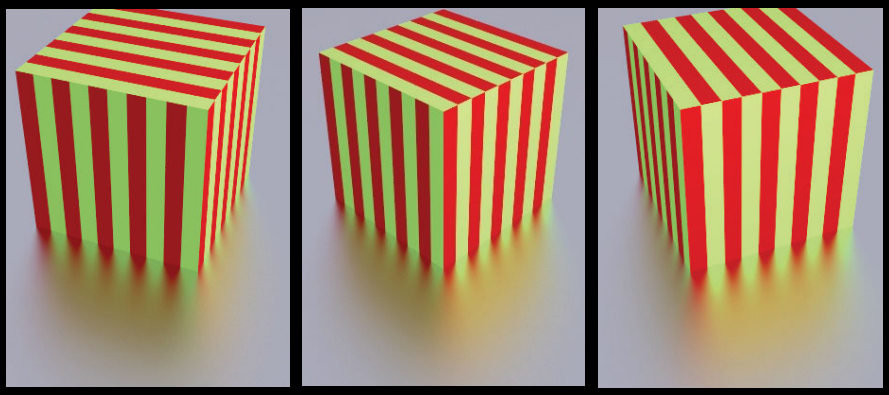


smooth indirect term

sparse computation and interpolation

16

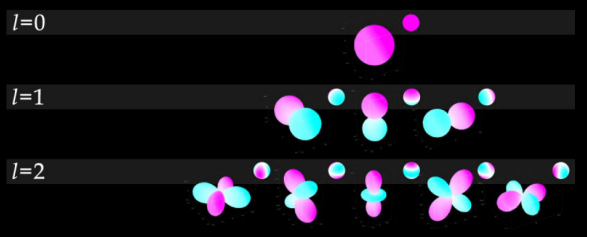
View dependence



17

Incoming radiance representation

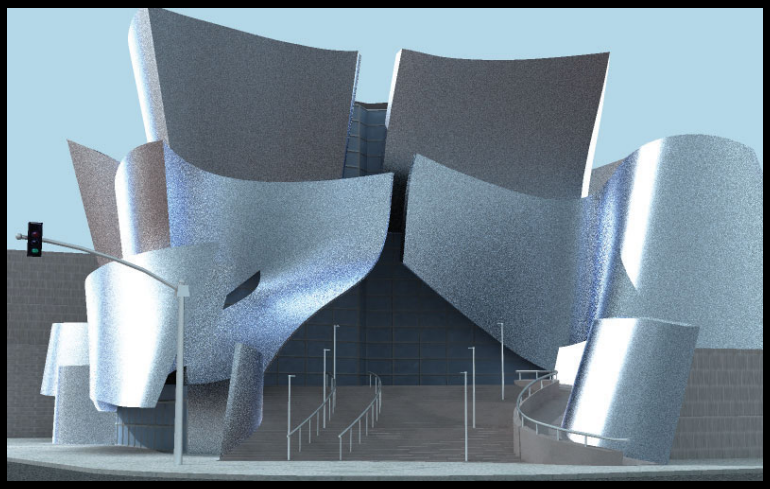
- Spherical Harmonics
  - basis functions on a sphere
- Approximate $\rho_{hd}(\omega_r)$ in the SH basis
  - 3D rotations of SH coefficients for change in viewpoint
- Dot with SH irradiance coefficient



[Green 2003]

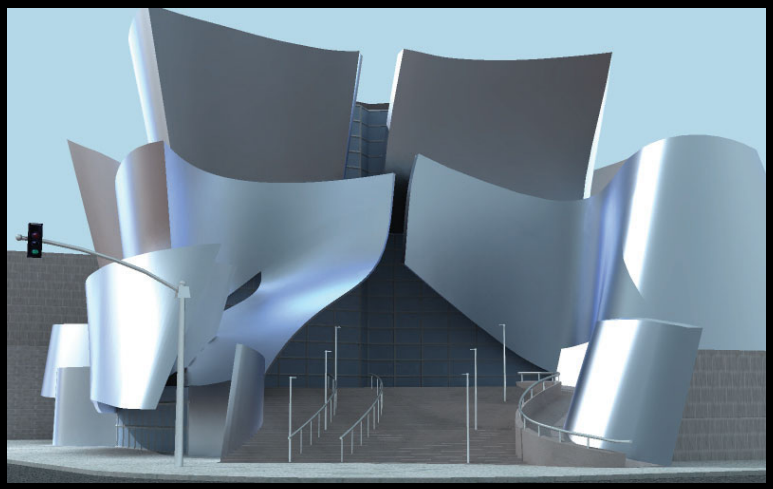
18

MC Importance Sampling



19

Radiance caching



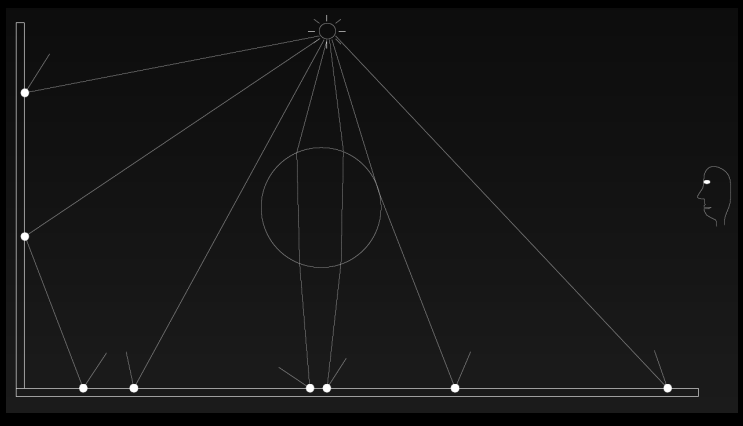
20

Indirect specular?

- Caustics are still hard to simulate with radiance caching ☹
- Solution: Photon Mapping [Jensen 2001]
- 2 pass algorithm
  - trace photon into the scene from light source
  - gather photon density from camera

21

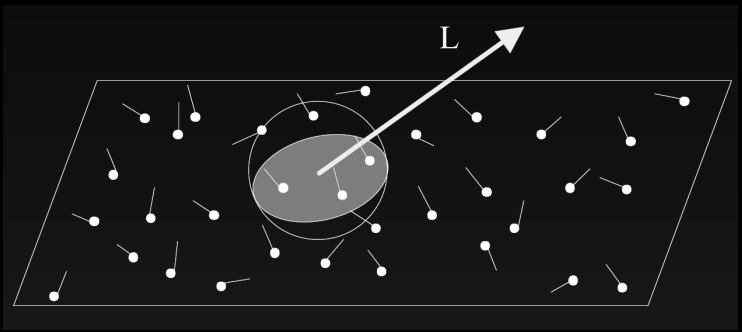
Photon tracing



Henrik Wann Jensen

22

Photon density



Henrik Wann Jensen

23

Radiance estimate

$$
\begin{aligned}
L(x,\vec{\omega}) &= \int_{\Omega} f_r(x,\vec{\omega}',\vec{\omega}) L'(x,\vec{\omega}') \cos\theta' \, d\omega \\
&= \int_{\Omega} f_r(x,\vec{\omega}',\vec{\omega}) \frac{d\Phi^2(x,\vec{\omega}')}{\cos\theta' \, d\omega \, dA} \cos\theta' d\omega \\
&= \int_{\Omega} f_r(x,\vec{\omega}',\vec{\omega}) \frac{d\Phi\ (x,\vec{\omega}')}{dA} \\
&\approx \sum_{p=1}^{n} f_r(x,\vec{\omega}'_p,\vec{\omega}) \frac{\Delta\Phi_p(x,\vec{\omega}'_p)}{\pi r^2}
\end{aligned}
$$

24

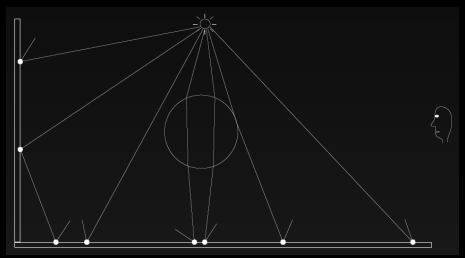
Photon map data-structure

- Balanced kd-tree

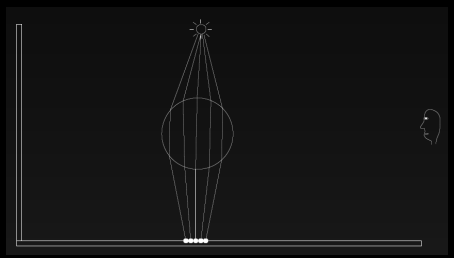
```
struct photon = {
  float position[3];
  rgbe power;         // power packed as 4 bytes
  char phi, theta;    // incoming direction
  short flags;
}
```

25

Two photon maps



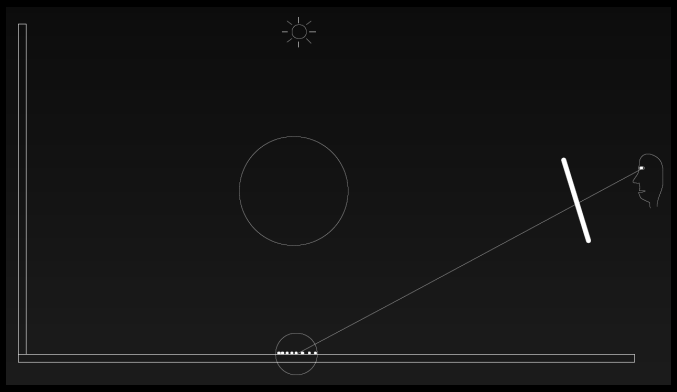
Global photon map



Caustic photon map

26

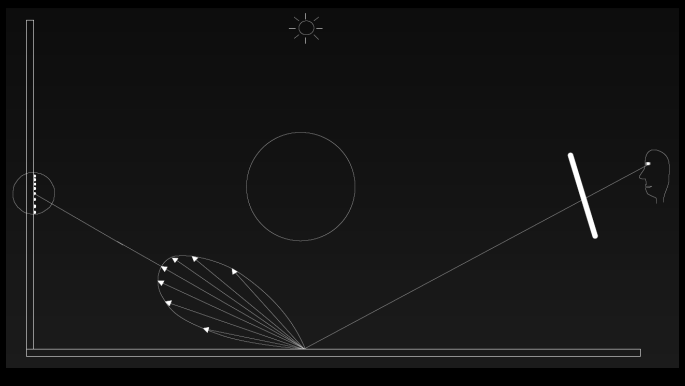
Rendering caustics



1st diffuse bounce

27

Rendering indirect illumination

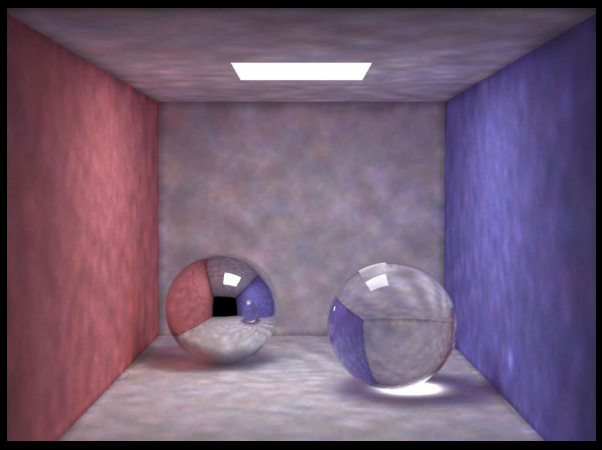


2nd diffuse bounce

- Also known as final gather!

28

Photon map visualization



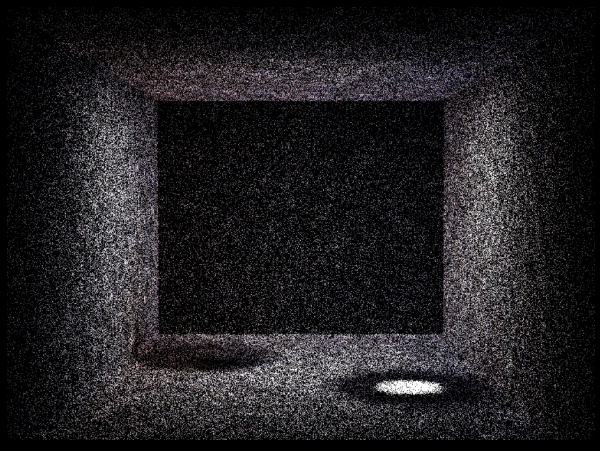
Henrik Wann Jensen

100,000 photons, 50 photons for density estimate

(1st bounce density evaluation)

29

Photon map visualization

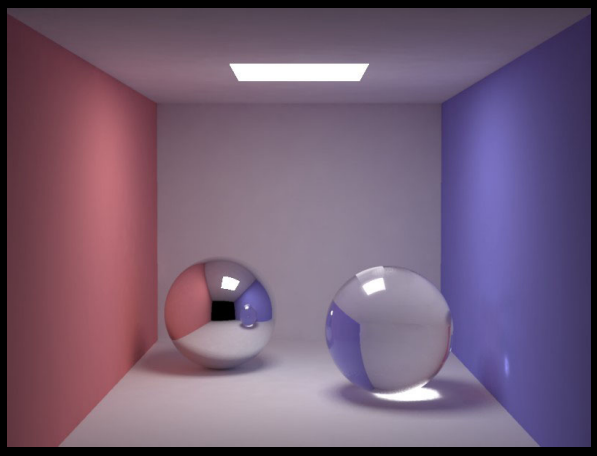


Henrik Wann Jensen

100,000 photons, 50 photons for density estimate

30

Rendering with final gather



Henrik Wann Jensen

- 2nd bounce density evaluation for indirect illumination

31

Photon Mapping - Cognac glass

32

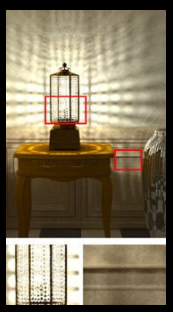
Photon Mapping – Mies house

33

Photon Mapping – subsurface scattering

34

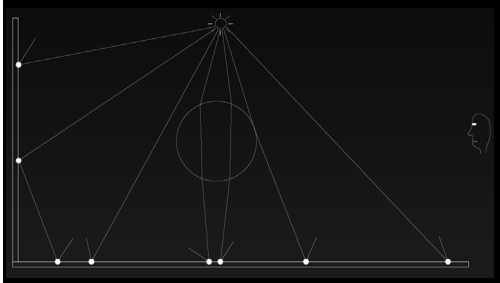
Caustics dominated scene

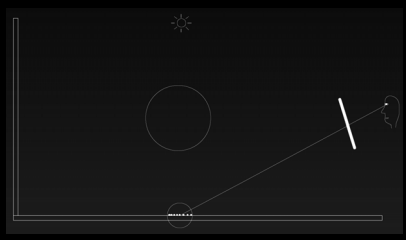


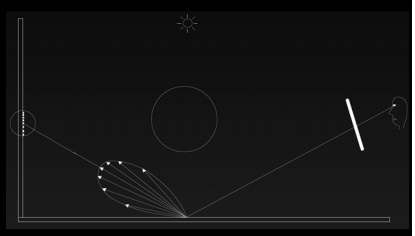
- Photon mapping still too expensive
  - too much memory consumption
- Solution: Progressive Photon Mapping [Hachisuka et al. 08]
  - resolve SDS paths with arbitrary accuracy with finite memory!
- Multiple photon tracing steps for incremental estimate
  - no storage of previously traced photons

35

Regular Photon Mapping



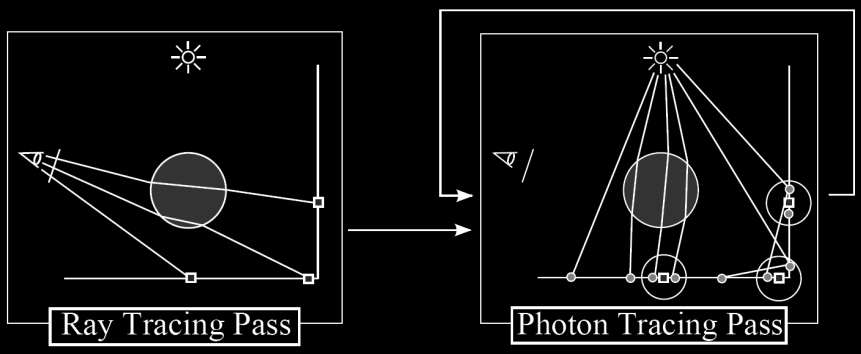




- 1st pass photon tracing from light source to diffuse surfaces
- 2nd pass of ray tracing from camera to scene hit points for evaluation of photon density estimate

36

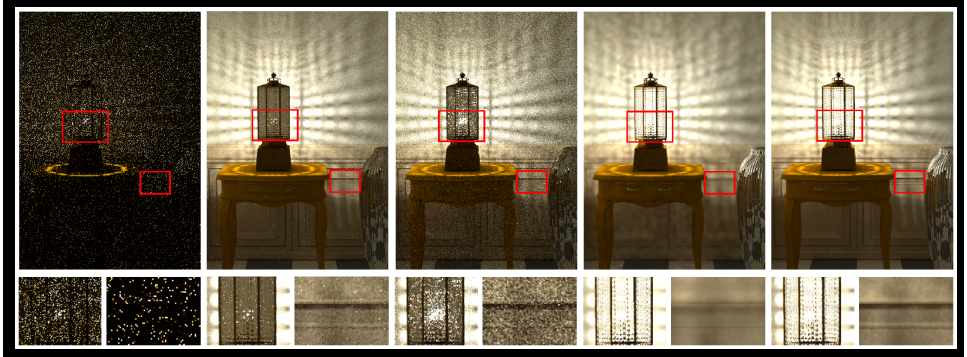
Progressive Photon Mapping



- 1st pass ray trace from camera to scene hit points
- 2nd pass (multiple) of photon tracing from light source to evaluate photon density at camera hit points. No need to store older photons! Update of density estimate after each pass.

37

Progressive photon mapping



[Hachisuka et al. 08]

38

Photon density estimation

- Source of bias:

$$L(x,\vec{\omega}) \approx \sum_{p=1}^{n} \frac{f_r(x,\vec{\omega},\vec{\omega}_p)\phi_p(x_p,\vec{\omega}_p)}{\pi r^2},$$

  - radiance estimate converges with increase in density
  - infinite density in the limit

- Trace N photons, estimate with N^{β} photons, β in [0,1]:

$$L(x,\vec{\omega}) = \lim_{N\to\infty} \sum_{p=1}^{\lfloor N^{\beta} \rfloor} \frac{f_r(x,\vec{\omega},\vec{\omega}_p)\phi_p(x_p,\vec{\omega}_p)}{\pi r^2},$$

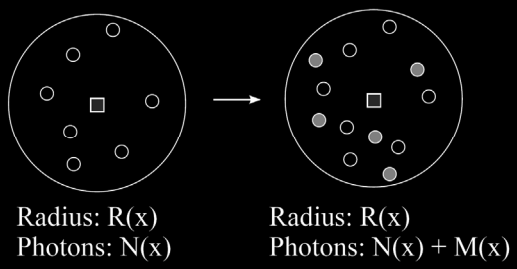
39

Progressive density estimate

- Density $d(x) = n/\pi r^2$
- If radius r remains constant, then density estimate is averaged
  - suboptimal
- Key idea is to progressively reduce r with each photon tracing pass
  - while increasing photon accumulation!

40

Radius reduction

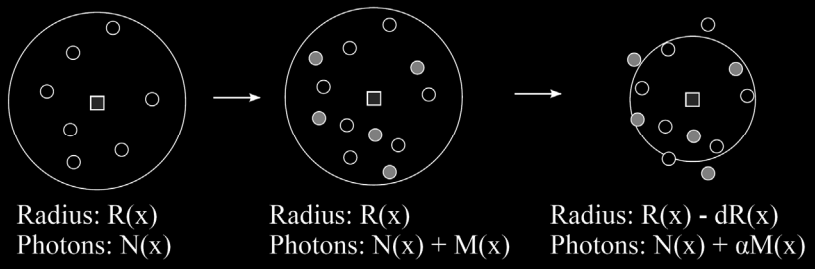


- After second round of photon tracing, local density with constant radius R(x)

  $d'(x) = (N(x) + M(x))/(\pi * R(x)^2)$

41

Radius reduction

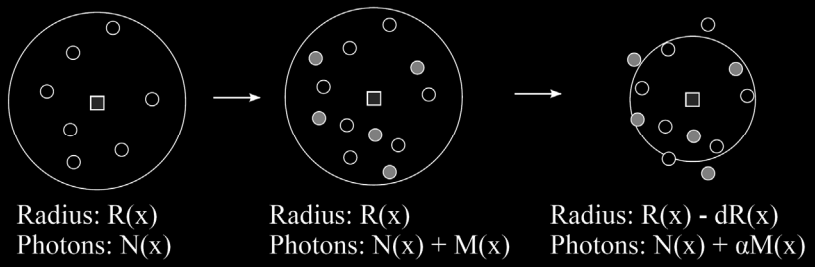


$$dR(x) = R(x) - R(x)\sqrt{\frac{N(x) + \alpha M(x)}{N(x) + M(x)}}.$$

- $\alpha \rightarrow [0, 1]$

42

Radius reduction

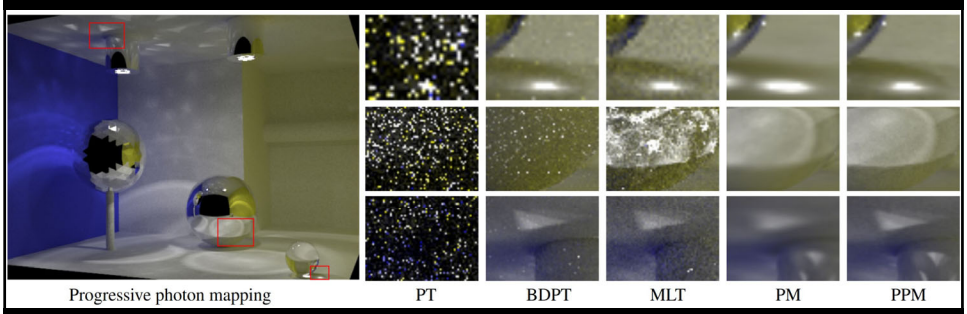


$$dR(x) = R(x) - R(x)\sqrt{\frac{N(x) + \alpha M(x)}{N(x) + M(x)}}.$$

- d'(x) = (N(x) + αM(x))/(π * (R(x) - dR(x))^2)

43

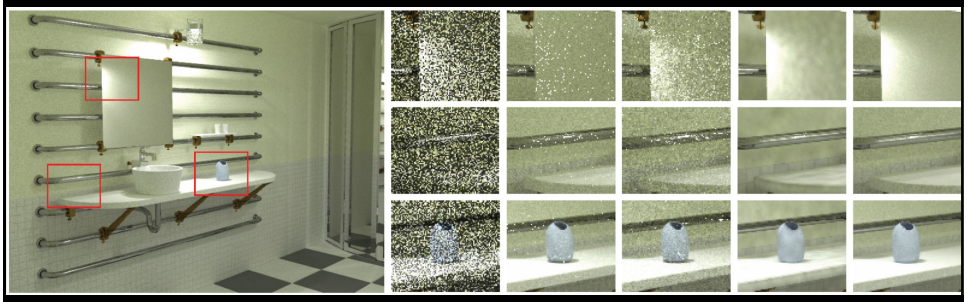
Progressive PM Results



[Hachisuka et al. 08]

44

Progressive PM Results



[Hachisuka et al. 08]

45

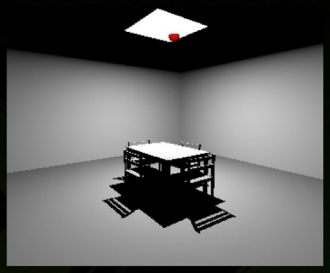
Many light methods

- Virtual point lights (VPLs) [Keller 97]
  - Approximate GI as contribution from a few virtual point lights
- Light Cuts [Walter et al. 05]
  - Hierarchical culling of point light contributions for efficient approximation

46

Instant Radiosity with VPLs [Keller 97]

- Approximate GI as contribution from a few virtual point lights
- Hardware (OpenGL) implementation using the accumulation buffer!
  - Soft shadowing computed as average of many images rendered with a point light placed at different positions on the area light

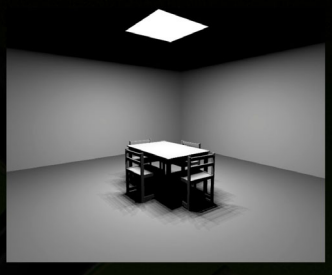
$\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1}$  $=$

i

47

Instant Radiosity with VPLs [Keller 97]

- Approximate GI as contribution from a few virtual point lights
- Hardware (OpenGL) implementation using the accumulation buffer!
  - Soft shadowing computed as average of many images rendered with a point light placed at different positions on the area light

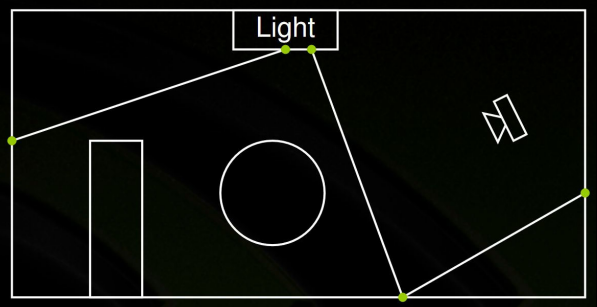
$\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1}$  $=$ 

i

48

Instant Radiosity with VPLs [Keller 97]

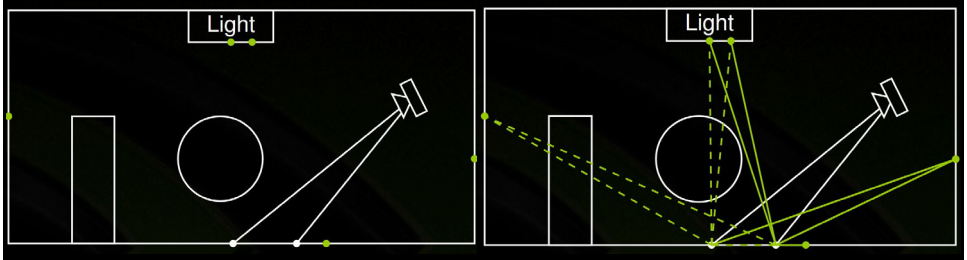
- Light path vertices stored as point lights
  - similar to a small photon map!



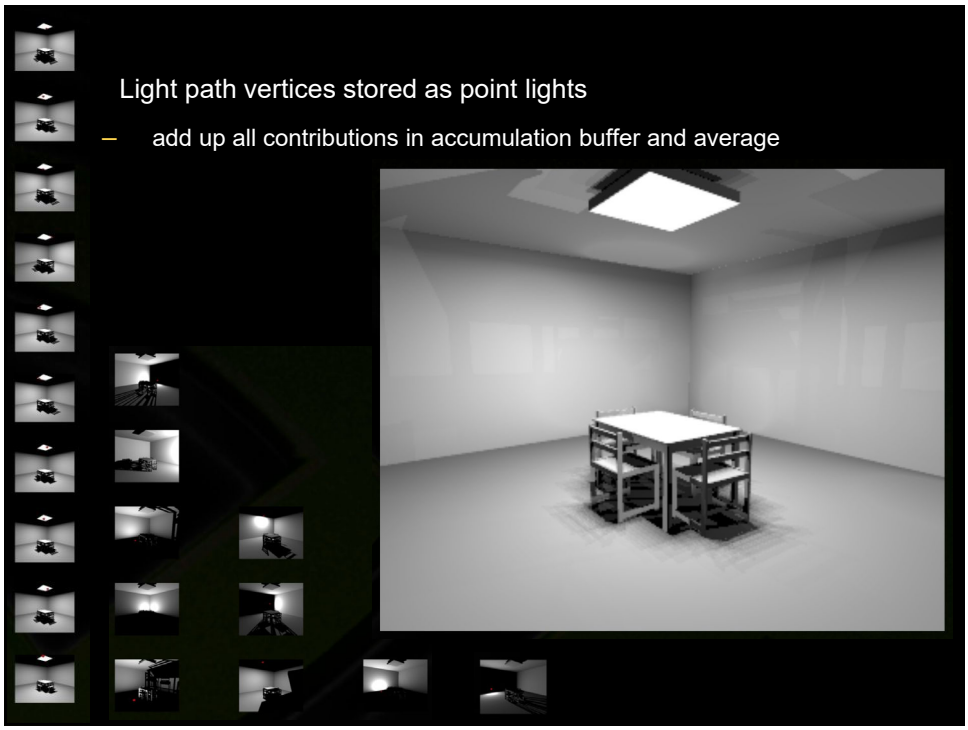
49

Instant Radiosity with VPLs [Keller 97]

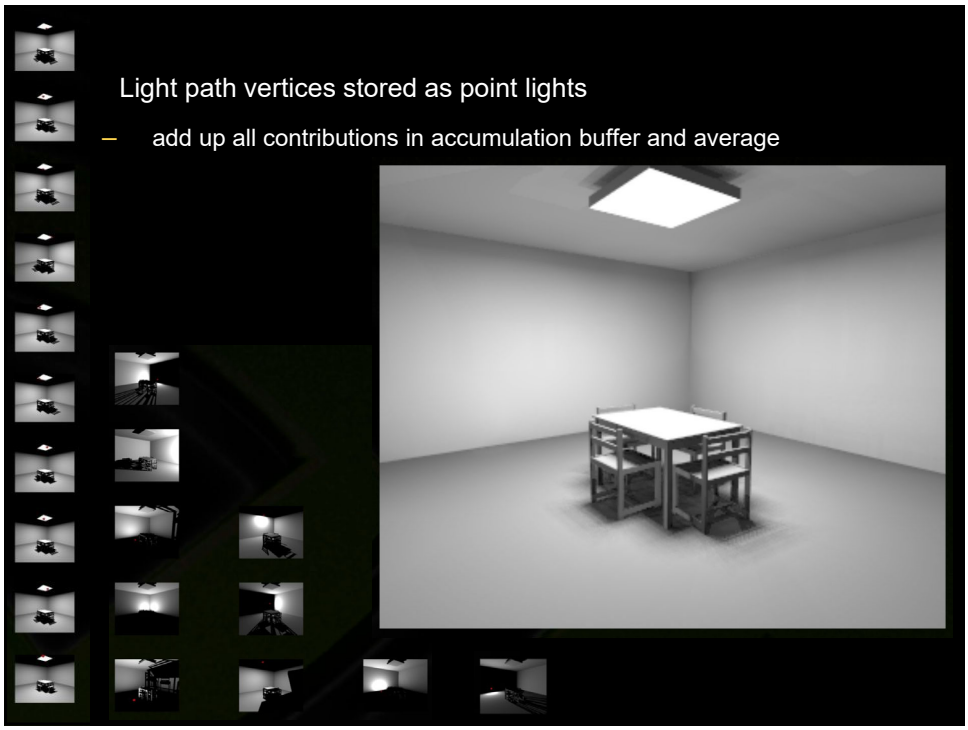
- Light path vertices stored as point lights
  - trace camera rays and then connect to VPLs with shadow rays



50


Light path vertices stored as point lights
– add up all contributions in accumulation buffer and average

51


Light path vertices stored as point lights
– add up all contributions in accumulation buffer and average

52

Scalability with many lights – Lightcuts
[Walter et al. 05]

- Simulate complex illumination using point lights
  - Area lights
  - HDR environment maps
  - Sun & sky light
  - Indirect illumination
- More lights ➔ more accurate
  - And more expensive
  - Naive cost linear in lights

Area lights + Sun/sky + Indirect

53

Accurate approximation with few lights
[Walter et al. 05]

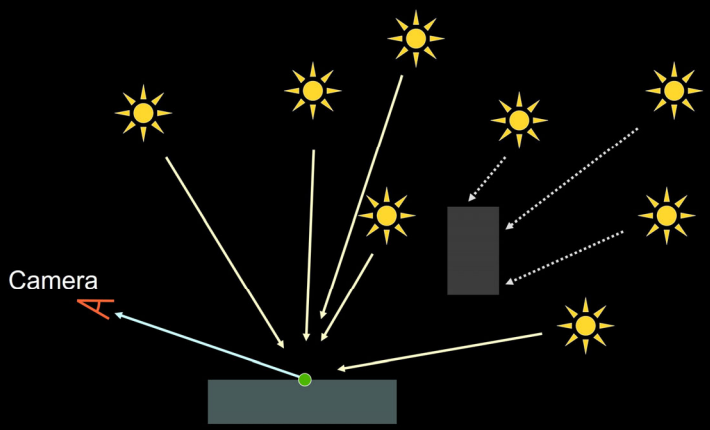
- Do we really need to evaluate all lights?
  - Thousands to millions of lights
  - Average contribution miniscule
  - Below human perceptual limits
    - Weber's Law (roughly 2%)
- Evaluate a small subset
  - Chosen adaptively
  - Perceptually accurate approximation

Environment lighting + Indirect

54

Lightcuts problem

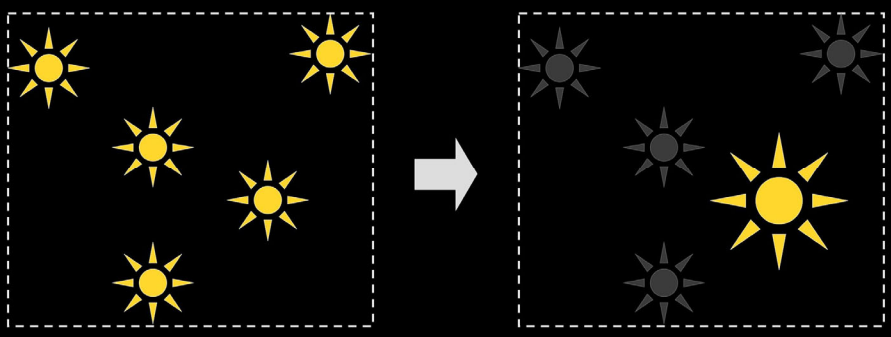
- All lights do not contribute equally to a surface
  - Some may be blocked or affected by BRDF



55

Lightcuts key idea - clustering

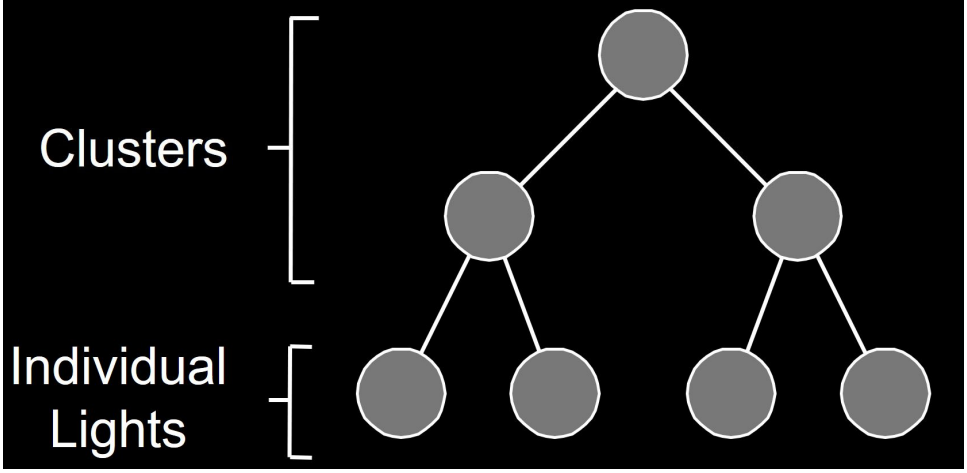
- Cluster a group of lights and represent them with a representative light



56

Light tree

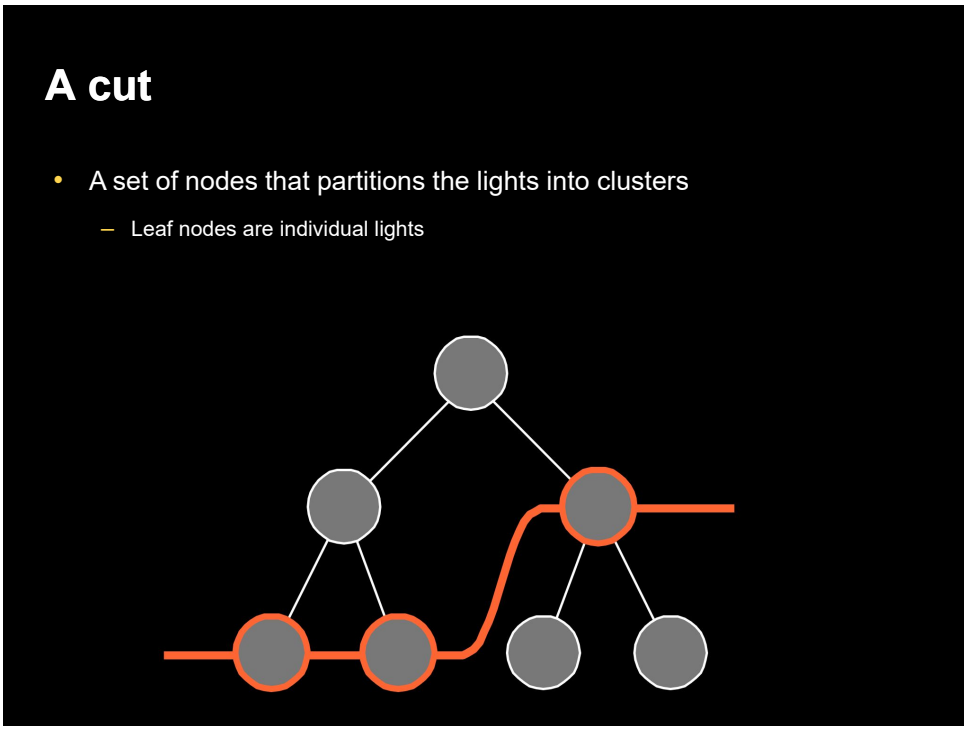
- A binary tree of lights and light clusters



57

A cut

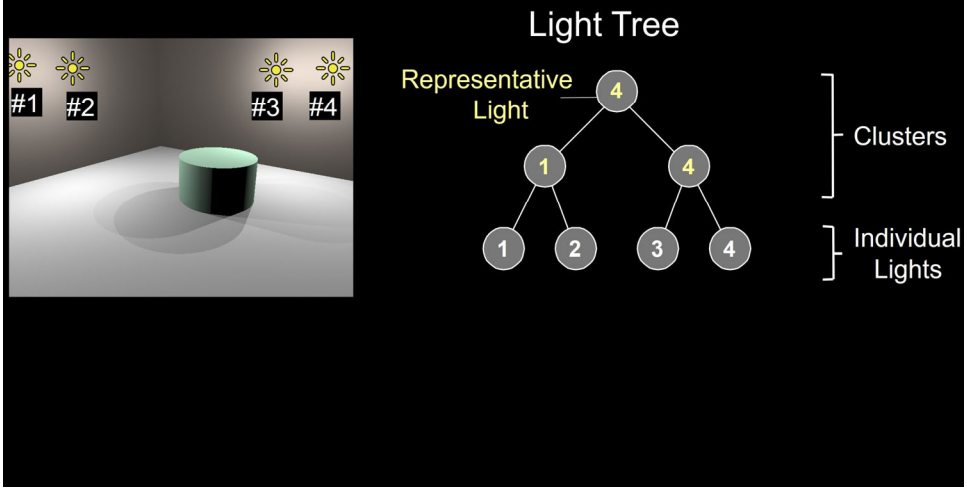
- A set of nodes that partitions the lights into clusters
  - Leaf nodes are individual lights



58

An example

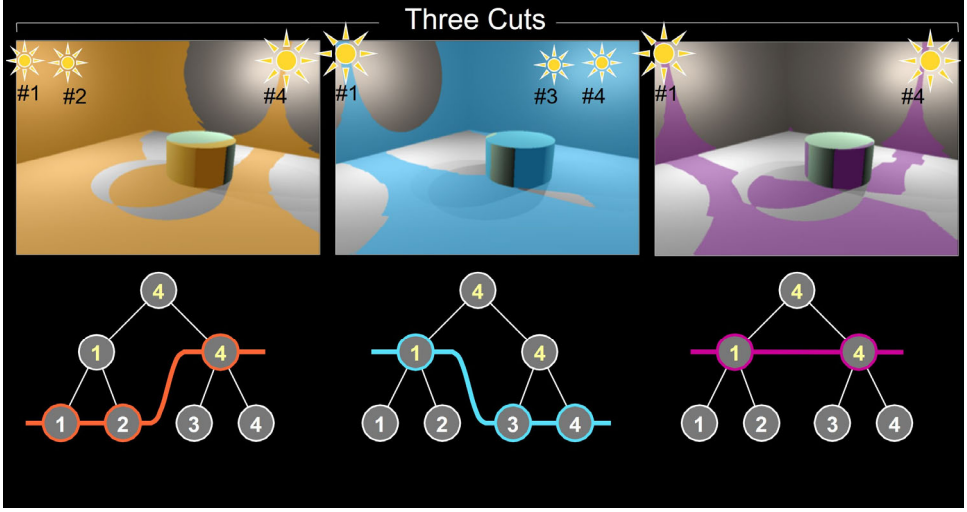
- A scene with 4 point lights



59

An example

- Three possible cuts of 3 lights

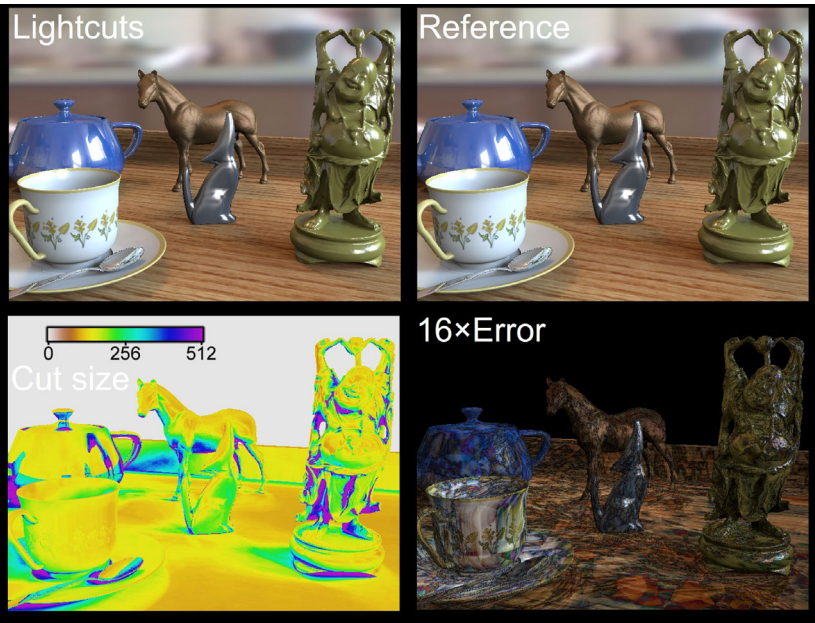


60

Lightcuts algorithm

- Pre-process
  - Convert illumination to point lights
  - Build light tree
- For each visible point
  - Choose a cut to approximate the local illumination
    - Bound maximum error of cluster approximation
    - Refine cluster if error bound is too large
- Ensure each cluster's error < perceptual threshold
  - Webber's Law 2% of signal

61



- 13000 VPLs for scene lit with environmental illumination + indirect
  - Cut uses only up to 512 lights for each shaded surface point

62