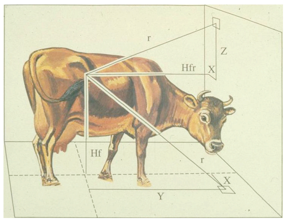
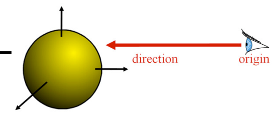
Local vs. Global Illumination & Radiosity

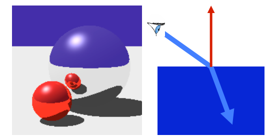


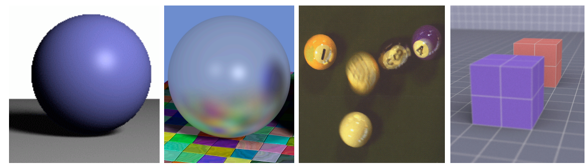
An early application of radiative heat transfer in stables.

Last Time?



- Ray Casting & Ray-Object Intersection
- Recursive Ray Tracing
- Distributed Ray Tracing



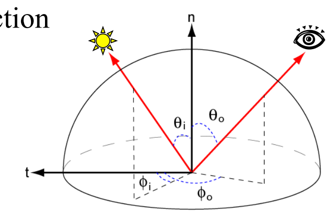


Today

- Local Illumination
  - BRDF
  - Ideal Diffuse Reflectance
  - Ideal Specular Reflectance
  - The Phong Model
- Why is Global Illumination Important?
- Radiosity Matrix
- Calculating the Form Factors
- Advanced Radiosity

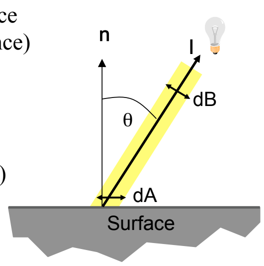
BRDF

- Ratio of light coming from one direction that gets reflected in another direction
- Bidirectional Reflectance Distribution Function
  - 4D
  - $R(\theta_i, \phi_i; \theta_o, \phi_o)$



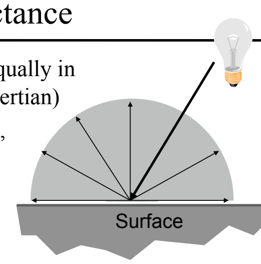
Incoming Radiance

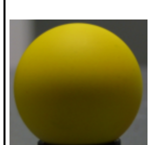
- The amount of light received by a surface depends on incoming angle
  - Bigger at normal incidence (Winter/Summer difference)
- By how much?
  - dB = dA cos θ
  - Same as: $\mathbf{l} \cdot \mathbf{n}$ (dot product with normal)

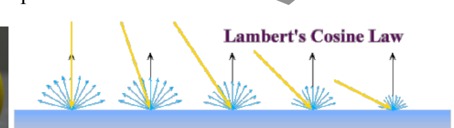


Ideal Diffuse Reflectance

- Assume surface reflects equally in all directions (a.k.a. Lambertian)
- An ideal diffuse surface is, at the microscopic level, a very rough surface
- Examples: chalk, clay, some paints

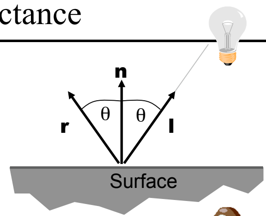






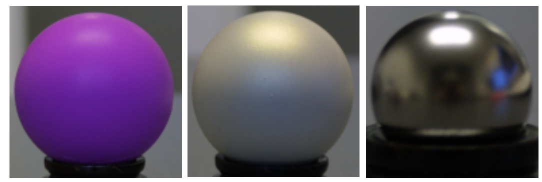
Ideal Specular Reflectance

- Assume surface reflects only in mirror direction
  - View dependent
- Microscopic surface elements are oriented in the same direction as the surface
- Examples: mirrors, highly polished metals



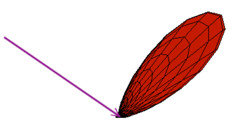
Non-Ideal Reflectors

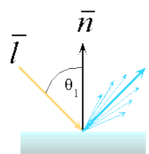
- Real materials tend to be *neither* ideal diffuse *nor* ideal reflective
- Highlight is blurry, looks glossy



Non-Ideal Reflectors

- Most light reflects in the ideal reflected direction
- Microscopic surface variations will reflect light just slightly offset
- How much light is reflected?



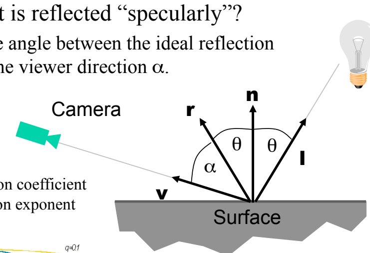


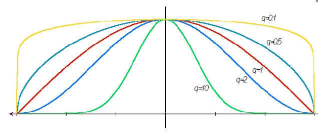
The Phong Model

- How much light is reflected "specularly"?
  - Depends on the angle between the ideal reflection direction and the viewer direction α.

$$L_o = k_s (\cos\alpha)^q \frac{L_i}{r^2}$$

ks: specular reflection coefficient
q : specular reflection exponent

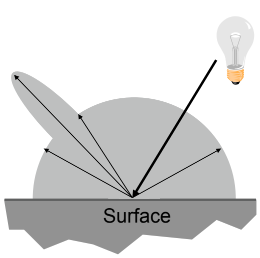


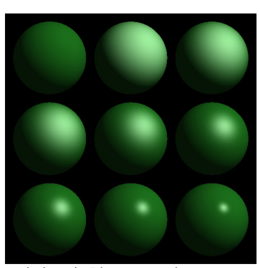


Effect of the *q* exponent

The Phong Model

- Sum of three components:
  diffuse reflection + specular reflection + "ambient".



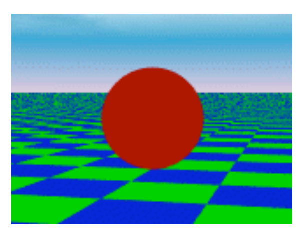


variations in Phong specular exponent

Ambient Illumination

- In a typical room, everything receives at least a little bit of light
- Ambient illumination represents the reflection of all indirect illumination

$$L(\omega_r) = k_a$$



- This is a total hack!

Questions?

Lightscape `http://www.lightscape.com`

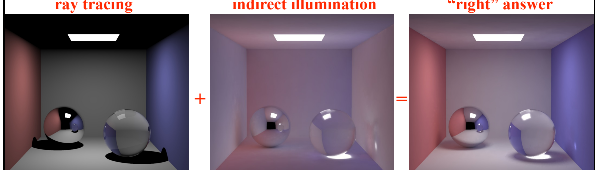
Today

- Local Illumination
- Why is Global Illumination Important?
  - The Cornell Box
  - Radiosity vs. Ray Tracing
- Radiosity Matrix
- Calculating the Form Factors
- Advanced Radiosity

Why Global Illumination?

- Simulate all light inter-reflections (indirect lighting)
  - in a room, a lot of the light is indirect: it is reflected by walls.
- How have we dealt with this so far?
  - Ambient term to fake some uniform indirect light

ray tracing + **indirect illumination** = **"right" answer**



(no ambient term) ***it is smooth, but not constant!***

Henrik Wann Jensen

Why Radiosity?

- Sculpture by John Ferren
- *Diffuse* panels

photograph:

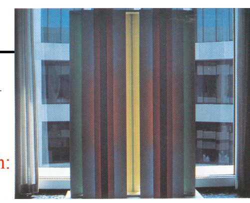
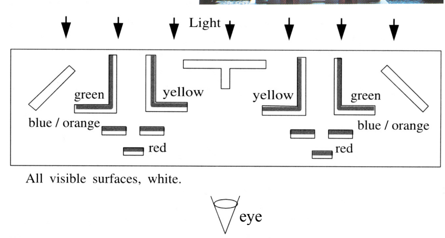


diagram from above:



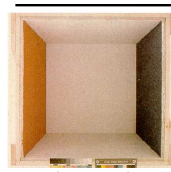
Radiosity vs. Ray Tracing

Original sculpture by John Ferren lit by daylight from behind.

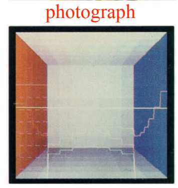
Ray traced image. A standard ray tracer cannot simulate the interreflection of light between diffuse surfaces.

Image rendered with radiosity. note color bleeding effects.

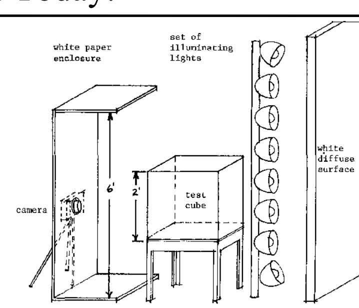
Reading for Today:



photograph



simulation

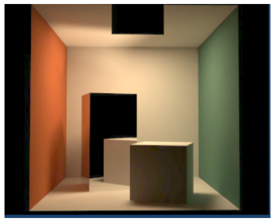


Goral, Torrance, Greenberg & Battaile
Modeling the Interaction of Light Between Diffuse Surfaces
SIGGRAPH '84

The Cornell Box

- Careful calibration and measurement allows for comparison between physical scene & simulation

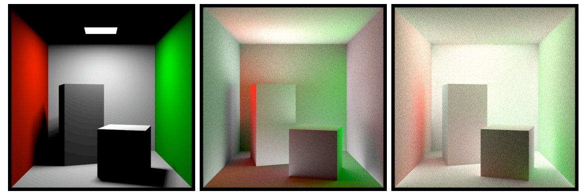
photograph



simulation

Light Measurement Laboratory
Cornell University, Program for Computer Graphics

Visualizing Inter-reflections…

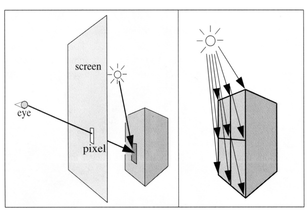


direct illumination (0 bounces) | 1 bounce | 2 bounces

images by Micheal Callahan
http://www.cs.utah.edu/~shirley/classes/cs684_98/students/callahan/bounce/

Radiosity vs. Ray Tracing

- Ray tracing is an *image-space* algorithm
  - If the camera is moved, we have to start over
- Radiosity is computed in *object-space*
  - View-independent (just don't move the light)
  - Can pre-compute complex lighting to allow interactive walkthroughs

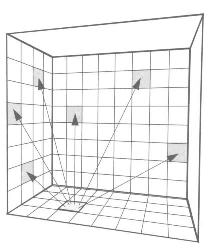


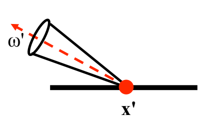
Today

- Local Illumination
- Why is Global Illumination Important?
- Radiosity Matrix
- Calculating the Form Factors
- Advanced Radiosity

Radiosity Overview

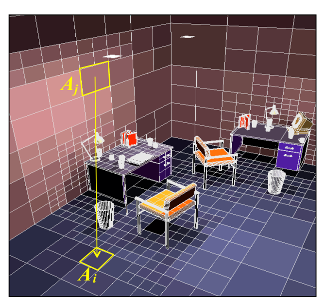
- Surfaces are assumed to be perfectly Lambertian (diffuse)
  - reflect incident light in all directions with equal intensity
- The scene is divided into a set of small areas, or patches.
- The radiosity, B_i, of patch *i* is the total rate of energy leaving a surface. The radiosity over a patch is constant.
- Units for radiosity: Watts / steradian * meter^2





Discrete Radiosity Equation

Discretize the scene into *n* patches, over which the radiosity is constant



$$B_i = E_i + \rho_i \sum_{j=1}^{n} F_{ij} B_j$$

- B_i: light leaving patch *i*
- ρ_i: material reflectivity
- E_i: light emitted from patch *i*
- F_{ij}: form factor

The equation is recursive, but it can be solved iteratively

Radiosity in Matrix Form

$$B_i = E_i + \rho_i \sum_{j=1}^{n} F_{ij} B_j$$

n simultaneous equations with n unknown B_i values can be written in matrix form:

solve for B_i

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & & \\ \vdots & & \ddots & \\ -\rho_n F_{n1} & \cdots & \cdots & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

A solution yields a single radiosity value B_i for each patch in the environment, a view-independent solution.

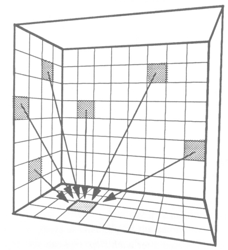
Solving the Radiosity Matrix

The radiosity of a single patch i is updated for each iteration by *gathering* radiosities from all other patches:

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_i \\ \vdots \\ E_n \end{bmatrix} + \begin{bmatrix} & & & \\ \rho_i F_{i1} & \rho_i F_{i2} & \cdots & \rho_i F_{in} \\ & & & \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{bmatrix}$$

Radiosity values on iteration *t+1*

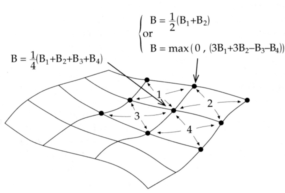
Radiosity values on iteration *t*



This method is fundamentally a Gauss-Seidel relaxation

Interpolating Vertex Radiosities

- B_i radiosity values are constant over the extent of a patch.
- How are they mapped to the vertex radiosities (intensities) needed by the renderer?
  - Average the radiosities of patches that contribute to the vertex
  - Vertices on the edge of a surface are assigned values extrapolation



Questions?

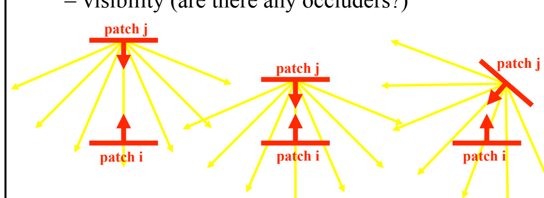
Factory simulation. Program of Computer Graphics, Cornell University. 30,000 patches.

Today

- Local Illumination
- Why is Global Illumination Important?
- Radiosity Equation/Matrix
- Calculating the Form Factors
- Advanced Radiosity

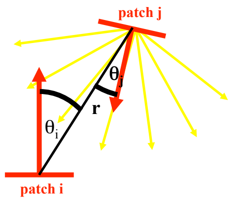
Calculating the Form Factor F_{ij}

- F_{ij} = fraction of light energy leaving patch j that arrives at patch i
- Takes account of both:
  - geometry (size, orientation & position)
  - visibility (are there any occluders?)



Calculating the Form Factor F_{ij}

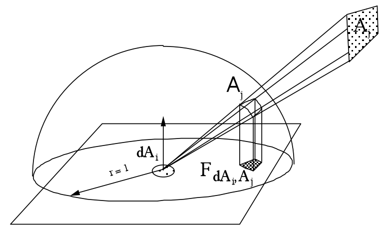
- F_{ij} = fraction of light energy leaving patch j that arrives at patch i



$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} V_{ij}\, dA_j\, dA_i$$

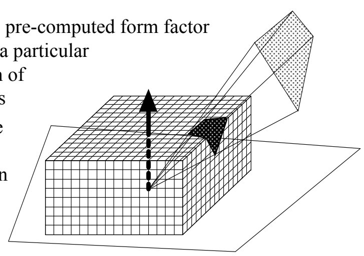
Form Factor Determination

The Nusselt analog: the form factor of a patch is equivalent to the fraction of the unit circle that is formed by taking the projection of the patch onto the hemisphere surface and projecting it down onto the circle.



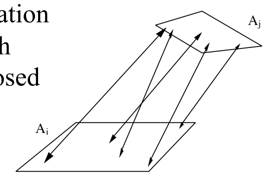
Hemicube Algorithm

- A hemicube is constructed around the center of each patch
- Faces of the hemicube are divided into "pixels"
- Each patch is projected (rasterized) onto the faces of the hemicube
- Each pixel stores its pre-computed form factor The form factor for a particular patch is just the sum of the pixels it overlaps
- Patch occlusions are handled similar to z-buffer rasterization



Form Factor from Ray Casting

- Cast *n* rays between the two patches
  - Compute visibility (what fraction of rays do not hit an occluder)
  - Integrate the point-to-point form factor
- Permits the computation of the patch-to-patch form factor, as opposed to point-to-patch



Questions?

Lightscape `http://www.lightscape.com`

Today

- Local Illumination
- Why is Global Illumination Important?
- Radiosity Equation/Matrix
- Calculating the Form Factors
- Advanced Radiosity
  - Progressive Radiosity
  - Adaptive Subdivision
  - Discontinuity Meshing
  - Hierarchical Radiosity

Stages in a Radiosity Solution

Input Geometry → **Form Factor Calculation** — > 90% ← **Why so costly?**

Calculation & storage of n^2 form factors

(n^3 for naive visibility calculation)

Emittance & Reflectance Properties → **Solve the Radiosity Matrix** — < 10%

↓

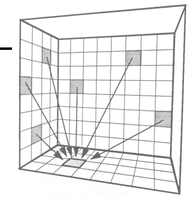
Radiosity Solution

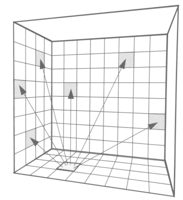
Camera Position & Orientation → **Visualization (Rendering)** — ~ 0%

↓

Radiosity Image

Progressive Refinement



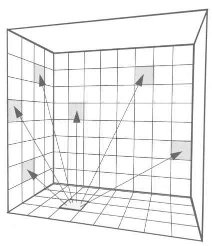


- Goal: Provide frequent and timely updates to the user during computation
- Key Idea: Update the entire image at every iteration, rather than a single patch
- How? Instead of summing the light received by one patch, distribute the radiance of the patch with the most *undistributed radiance*.

Reordering the Solution for PR

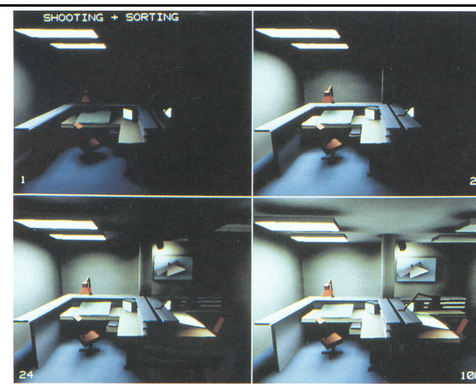
Shooting: the radiosity of all patches is updated for each iteration:

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ \vdots \\ B_n \end{bmatrix} + \begin{bmatrix} \cdots & \rho_1 F_{1i} & \cdots \\ \cdots & \rho_2 F_{2i} & \\ & & \\ \cdots & \rho_n F_{ni} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ B_i \\ \vdots \end{bmatrix}$$

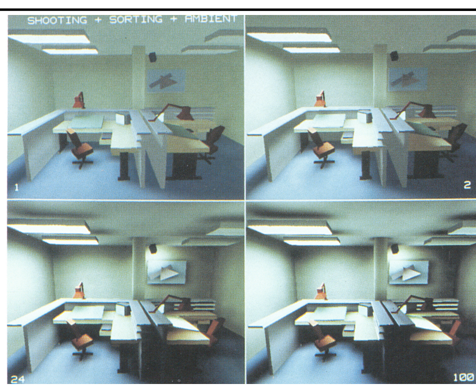


This method is fundamentally a Southwell relaxation

Progressive Refinement w/out Ambient Term



Progressive Refinement with Ambient Term



Questions?

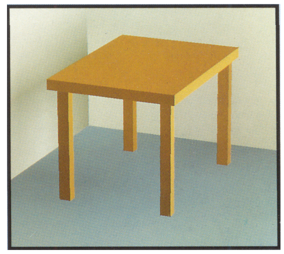
Lightscape `http://www.lightscape.com`

Today

- Local Illumination
- Why is Global Illumination Important?
- Radiosity Equation/Matrix
- Calculating the Form Factors
- Advanced Radiosity
  - Progressive Radiosity
  - Adaptive Subdivision
  - Discontinuity Meshing
  - Hierarchical Radiosity

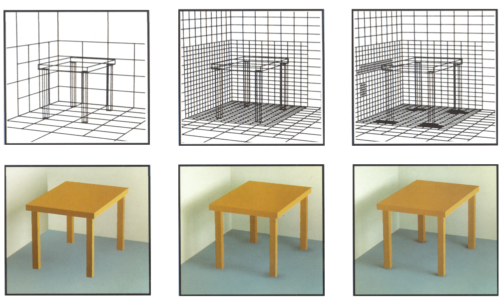
Increasing the Accuracy of the Solution

What's wrong with this picture?



- Image quality is a function of patch size
- Compute a solution on a uniform initial mesh, then refine the mesh in areas that exceed some error tolerance:
  - shadow boundaries
  - other areas with a high radiosity gradient

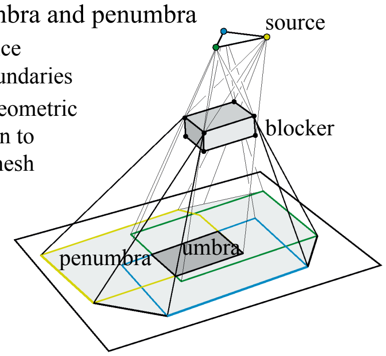
Adaptive Subdivision of Patches



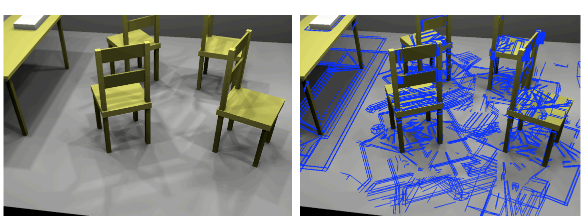
Coarse patch solution (145 patches) | Improved solution (1021 subpatches) | Adaptive subdivision (1306 subpatches)

Discontinuity Meshing

- Limits of umbra and penumbra
  - Captures nice shadow boundaries
  - Complex geometric computation to construct mesh



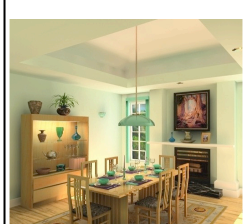
Discontinuity Meshing

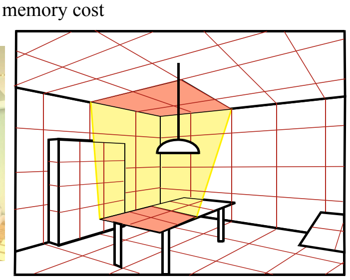


"Fast and Accurate Hierarchical Radiosity Using Global Visibility"
Durand, Drettakis, & Puech 1999

Hierarchical Radiosity

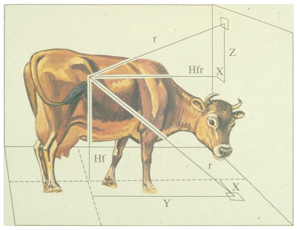
- Group elements when the light exchange is not important
  - Breaks the quadratic complexity
  - Control non trivial, memory cost





Practical Problems with Radiosity

- Meshing
  - memory
  - robustness
- Form factors
  - computation



Cow-cow form factor?

- Diffuse limitation
  - extension to specular takes too much memory

Questions?

Lightscape `http://www.lightscape.com`

Readings for Tuesday 3/4 *(pick one)*:

- "The Rendering Equation", Kajiya, SIGGRAPH 1986
  $L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA$
- "A Theoretical Framework for Physically Based Rendering", Lafortune and Willems, Computer Graphics Forum, 1994.

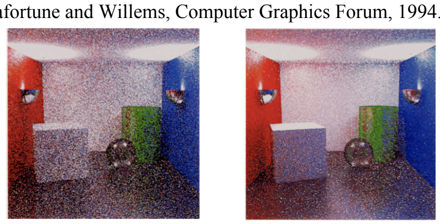


Figure B: *An indirectly illuminated scene rendered using path tracing and bidirectional path tracing respectively. The latter method results in visibly less noisefor the same amount of work.*