Lecture 33 of 41

Ray Tracing, Part 2 of 2: Distributed RT & Radiosity/RT Hybrid Systems

William H. Hsu
Department of Computing and Information Sciences, KSU

KSOL course pages: http://bit.ly/hGvXlH / http://bit.ly/eVizrE
Public mirror web site: http://www.kddresearch.org/Courses/CIS636
Instructor home page: http://www.cis.ksu.edu/~bhsu

Readings:

Last class: **Ray Tracing Handout**
Today: Chapter 15, Eberly *2^e^* – see **http://bit.ly/ieUq45**; **Ray Tracing Handout**
Next class: **Tufte Handout 1**

Lecture Outline

- **Reading for Last Class: Ray Tracing Handout**
- **Reading for Today: Chapter 15, Eberly *2^e^*; Ray Tracing Handout**
- **Reading for Next Class: Tufte Handout 1**
- **Last Time: Ray Tracing Lab**
  - **ACM SIGGRAPH demo: http://bit.ly/cIlgx2**
  - **POV-Ray: http://www.povray.org**
- **Today: Ray Tracing, Part 2 of 2**
  - **Hybrid global illumination: RT with radiosity**
    - **Calculating specular exponents**
    - **Pre-rendering backgrounds**
    - **Progressive refinement**
  - **Other optimizations**
- **Next Class: Visualization, Part 1 of 3 – Data**
  - **Source: *The Visual Display of Quantitative Information, 2^e^***
  - **Applications: scientific visualization, information visualization**

Where We Are

21	**Lab 4a: Animation Basics**	**Flash animation handout**
22	Animation 2: Rotations; Dynamics, Kinematics	Chapter 17, esp. §17.1 – 17.2
23	Demos 4: Modeling & Simulation; Rotations	Chapter 10^1, 13^2, §17.3 – 17.5
24	**Collisions 1: axes, OBBs, Lab 4b**	**§2.4.3, 8.1, GL handout**
25	Spatial Sorting: Binary Space Partitioning	Chapter 6, esp. §6.1
26	**Demos 5: More CGA; Picking; HW/Exam**	**Chapter 7^2; § 8.4**
27	**Lab 5a: Interaction Handling**	**§ 8.3 – 8.4; 4.2, 5.0, 5.6, 9.1**
28	Collisions 2: Dynamic, Particle Systems	§ 9.1, particle system handout
	Exam 2 review; Hour Exam 2 (evening)	**Chapters 5 – 6, 7^2 – 8, 12, 17**
29	**Lab 5b: Particle Systems**	**Particle system handout**
30	**Animation 3: Control & IK**	**§ 5.3, CGA handout**
31	Ray Tracing 1: intersections, ray trees	Chapter 14
32	Lab 6a: Ray Tracing Basics with POV-Ray	RT handout
33	Ray Tracing 2: advanced topic survey	Chapter 15, **RT handout**
34	Visualization 1: Data (Quantities & Evidence)	Tufte handout (1)
35	**Lab 6b: More Ray Tracing**	**RT handout**
36	Visualization 2: Objects	Tufte handout (2 & 4)
37	Color Basics; Term Project Prep	**Color handout**
38	Lab 7: Fractals & Terrain Generation	**Fractals/Terrain handout**
39	**Visualization 3: Processes; Final Review 1**	Tufte handout (3)
40	**Project presentations 1; Final Review 2**	**–**
41	Project presentations 2	–
	Final Exam	**Ch. 1 – 8, 10 – 15, 17, 20**

Lightly-shaded entries denote the due date of a written problem set; heavily-shaded entries, that of a machine problem (programming assignment); blue-shaded entries, that of a paper review; and the green-shaded entry, that of the term project.

Green, **blue** and **red** letters denote **exam review**, **exam**, and **exam solution review** dates.

Acknowledgements: Advanced Ray Tracing & Radiosity

David K. Buck, Aaron Collins, *et al.*

Developers

Persistence of Vision Raytracer (POV-Ray)

http://www.povray.org

C. Khambamettu & Mani V. Thomas

Professor, Department of Computing and Information Sciences, University of Delaware &

Ph.D. graduate (University of Delaware) / Research Scientist, Canfield Scientific

http://www.cis.udel.edu/~chandra/

http://vims.cis.udel.edu/~mani/

Mario Costa Sousa

Associate Professor

Department of Computer Science

University of Calgary

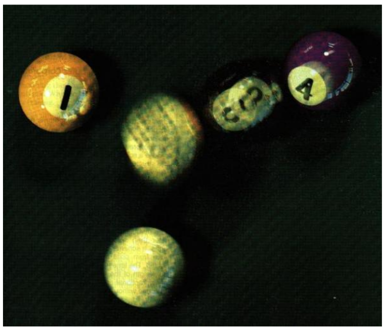
http://pages.cpsc.ucalgary.ca/~mario/

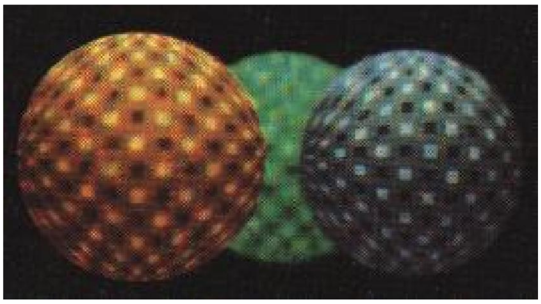
Review [1]: Recursive Ray Tracing Algorithm

- Compute 3D ray into scene for each 2D image pixel
- Compute 3D intersection point of ray with nearest object in scene
  - Test each primitive in the scene for intersection
  - Find nearest intersection
- Recursively spawn rays from point of intersection
  - Shadow Rays
  - Reflected rays
  - Transmitted rays
- Accumulate color from each spawned ray at point of intersection

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

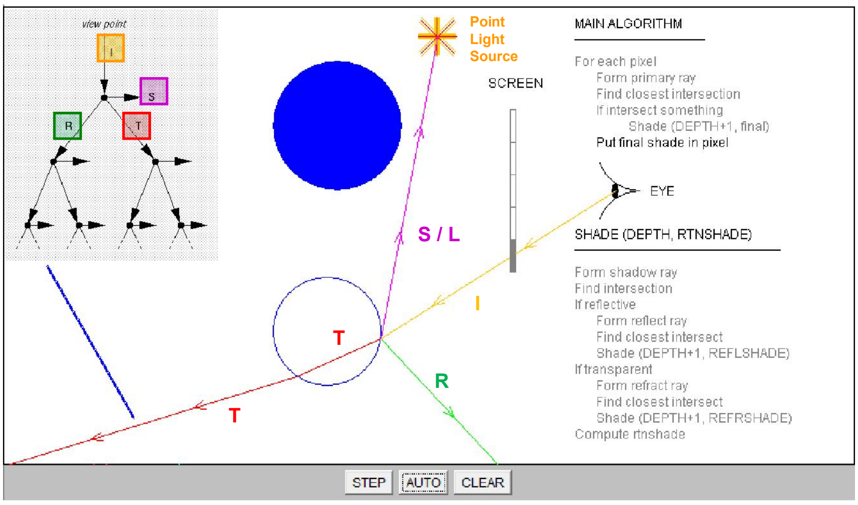
Review [2]: Distributed Ray Tracing





Adapted from slides ♥ 2001 D. Shreiner & B. Grantham, SCU
COEN 290: Computer Graphics I, Winter 2001 – http://bit.ly/hz1kfU

Review [3]: Java-Based 2-D RT Demo, SIGGRAPH



```
MAIN ALGORITHM

For each pixel
    Form primary ray
    Find closest intersection
    If intersect something
        Shade (DEPTH+1, final)
    Put final shade in pixel
```

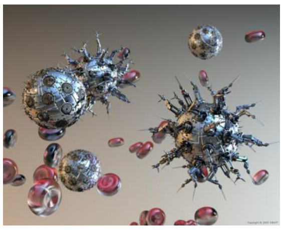
```
SHADE (DEPTH, RTNSHADE)

Form shadow ray
Find intersection
If reflective
    Form reflect ray
    Find closest intersect
    Shade (DEPTH+1, REFLSHADE)
If transparent
    Form refract ray
    Find closest intersect
    Shade (DEPTH+1, REFRSHADE)
Compute rtnshade
```

Screenshots from Java program ♥ 2001 G. S. Owen & Y. Liu, GSU
ACM SIGGRAPH Ray Trace Java Demo – http://bit.ly/cllgx2

Review [4]: POV-Ray

"The Wet Bird" © 2001 Gilles Tran
http://bit.ly/gMBuGt



"Dissolution" © 2005 Newt
http://bit.ly/fVqj5d

"Thanks for all the fish" © 2008 Robert McGregor
http://bit.ly/fE04gm

Images ♥ respective authors, generated using *POV-Ray*
© 1991 – 2011 D. K. Buck *et al.* – http://www.povray.org

Formulas: Ray-Object Intersection

- **Intersection with plane**
  - ✷ Implicit form

    $$F(x,y,z) = ax + by + cz + d = \mathbf{n}\cdot\mathbf{x} + d$$

  - ✷ Intersection

    $$\mathbf{n}\cdot\mathbf{S} + (\mathbf{n}\cdot\mathbf{c})t + d = 0 \Rightarrow t = \frac{-d - \mathbf{n}\cdot\mathbf{S}}{\mathbf{n}\cdot\mathbf{c}}$$

- **Intersection with sphere**
  - ✷ Implicit form

    $$|\mathbf{p} - \mathbf{p}_c|^2 = r^2 \quad \mathbf{p} = (x,y,z), \mathbf{p}_c = (a,b,c)$$

  - ✷ Intersection

    $$t = -\mathbf{c}\cdot(\mathbf{S} - \mathbf{p}_c) \pm \sqrt{(\mathbf{c}\cdot(\mathbf{S} - \mathbf{p}_c))^2 - |c|^2(|\mathbf{S} - \mathbf{p}_c|^2 - r^2)}$$

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Formulas: Light Vectors aka Shadow Rays

- **“Shadow feelers”**
  - ✷ Spawn ray from P to each light source
  - ✷ If there is intersection of shadow ray with any object then P is in shadow
- **Reflection**
  - ✷ Angle of incidence = angle of reflection

$$\begin{aligned}\mathbf{m} &= \left(\mathbf{a}\cdot\frac{\mathbf{n}}{|\mathbf{n}|}\right)\frac{(-\mathbf{n})}{|\mathbf{n}|} = -\frac{\mathbf{a}\cdot\mathbf{n}}{|\mathbf{n}|^2}\mathbf{n} \\ &= -(\mathbf{a}\cdot\hat{\mathbf{n}})\hat{\mathbf{n}} \qquad |\mathbf{n}| = 1 \\ &= -(|\mathbf{a}||\hat{\mathbf{n}}|\cos(180 - \theta_1))\hat{\mathbf{n}} = (\mathbf{a}\cdot\hat{\mathbf{n}})\hat{\mathbf{n}}\end{aligned}$$

$$\begin{aligned}\mathbf{r} &= \mathbf{e} + (-\mathbf{m}) = (\mathbf{a} - \mathbf{m}) + (-\mathbf{m}) = \mathbf{a} - 2\mathbf{m} \\ &= \mathbf{a} - 2(\mathbf{a}\cdot\hat{\mathbf{n}})\hat{\mathbf{n}}\end{aligned}$$

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Formulas:
Refraction of Transmitted Ray

- Ray passing through two media
  - Different refractive indices
  - Ray bends towards/away from normal
- Snell's Law
  - n_i and n_r are refractive indices of two media

$$n_i \sin\theta_i = n_r \sin\theta_r$$

- Transmitted ray

$$\mathbf{T} = \frac{n_i}{n_r}\mathbf{u} - \left(\cos\theta_r - \frac{n_i}{n_r}\cos\theta_i\right)\mathbf{n}$$

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

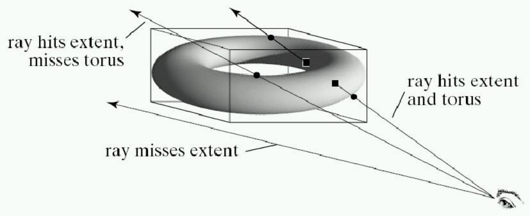
Speeding Up RT Using Extents/BVs [1]:
Motivation

- Ray tracing slow, performs same functions repeatedly
- Most time spent in computing intersections
  - Each ray should be intersected with every object in scene
  - Each ray spawns out reflected/transmitted rays which have to be interested with objects in scene

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Speeding Up RT Using Extents/BVs [2]: Definition & Basic Idea

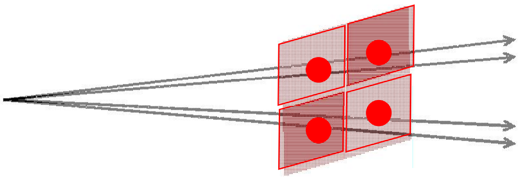
- Extent (*aka* bounding volume) of object: shape that encloses it
- Compute complicated intersections if and only if ray hits extent
- Two shapes most commonly used as extents
  - Sphere – specified by center and radius (*C*, *r*)
  - Box – specified by sides aligned to coordinate axis
    - Axis-aligned bounding box (AABB) – more typical for RT
    - Oriented bounding box (OBB)



Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Super-Sampling

- Instead of shooting one ray per pixel, shoot four rays through corners of pixel
- Adaptive super-sampling (Whitted's approach)
  - Shoot more rays through corners with higher intensity variation
  - Compute final color as weighted average rather than regular average



Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

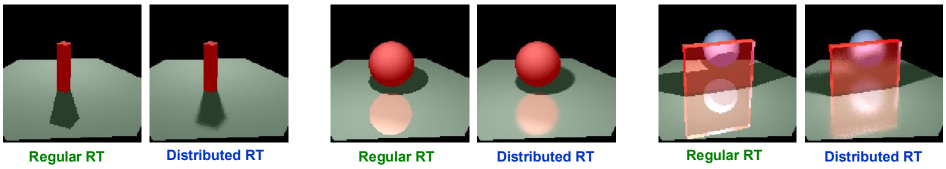
Advanced Topics in RT: Road Map

- Monte Carlo Methods: Distributed RT
- Bidirectional Ray Tracing: Caustics
- POV-Ray
- Hybrid Global Illumination
  - RT: good for
    - Specular highlights (highlights)
    - Point-to-point interobject reflectance, shadows
  - Radiosity: good for
    - Diffuse reflectance (matte effects)
    - Patch-to-patch interobject reflectance
  - Best of both worlds
    - RT for exponents
    - Radiosity for backgrounds

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [1]: What Is It?

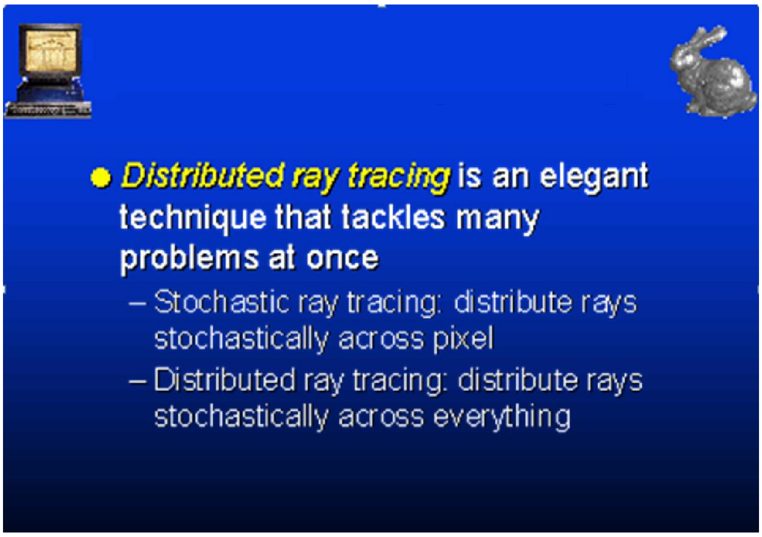
- Distributed ray tracing: not RT on distributed systems!
- Ray tracing method
  - based on randomly distributed oversampling
  - to reduce aliasing artifacts
  - in rendered images
- Reference
  - Allen Martin, Worcester Polytechnic Institute (WPI)
  - Examples for shadows, reflection, transparency



"Distributed Ray Tracing" © 1995 A. Martin – http://bit.ly/ex5ZUm

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [2]: From Stochastic RT to Distributed RT



- *Distributed ray tracing* is an elegant technique that tackles many problems at once
  - – Stochastic ray tracing: distribute rays stochastically across pixel
  - – Distributed ray tracing: distribute rays stochastically across everything

Adapted from slide ♥ 2005 D. Luebke, University of Virginia
CS 551-0003/651-0001: Advanced CG, Spring 2005 – http://bit.ly/eTWYAo

Computer Science at the University of Virginia

Distributed Ray Tracing [3]: Stochastic Oversampling (Cook, 1984)

- Developed by Cook *et al.* ("*Distributed Ray Tracing*", *Computer Graphics*, vol. 18, no. 3, pp 137-145, 1984)
- Stochastic Oversampling: http://bit.ly/eTWYAo
  - ✷ Pixel for antialiasing
  - ✷ Light source for soft shadows
  - ✷ Reflection function for soft (glossy) reflections
  - ✷ Time for motion blur
  - ✷ Lens for depth of field

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

University of Delaware

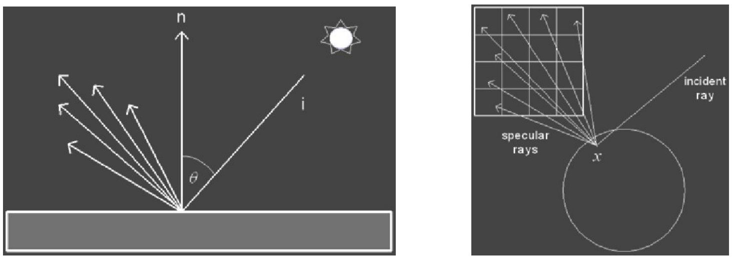
Distributed Ray Tracing [4]: Gloss

- Partially reflecting surfaces
- Traditional ray tracing
  - reflections look identical to scene they are reflecting
  - reflections are always sharp
- Randomly distributing rays reflected by surface
- Send out packet of rays around reflecting direction
- Actual value of reflectance is statistical mean of the values returned by each of these rays

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [5]: Perturbing Specular Reflection Ray

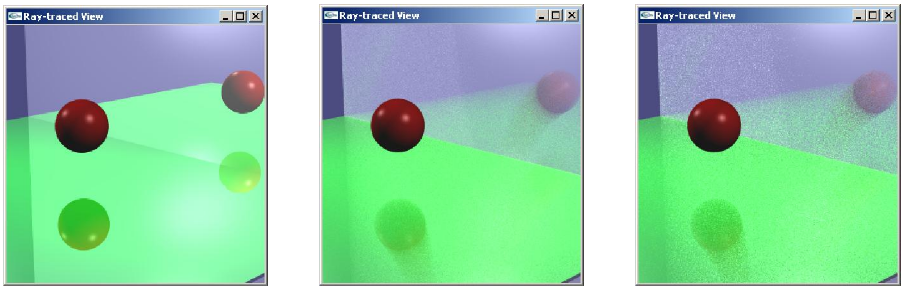
- Distributing set of reflection rays by randomly perturbing ideal specular reflection ray
- Spread of distribution determines glossiness where wider distribution spread models rougher surface



"Realistic Raytracing" © 2003 Z. Waters & E. Agu, WPI
http://bit.ly/gUNeGr

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [6]: Multiple Reflected Rays

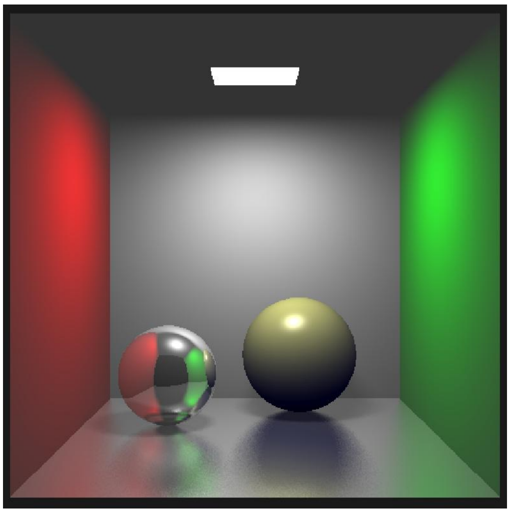


- First image is from traditional ray tracer
- Second one uses 16 rays in place of single reflected ray
- Third image uses 64 rays

"Monte Carlo Ray Tracing"
© 2005 D. Strock, University of Wisconsin Milwaukee
http://bit.ly/fSIYsL

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [7]: Soft Shadows & Reflection



"Realistic Raytracing" © 2003 Z. Waters & E. Agu, WPI
http://bit.ly/gUNeGr

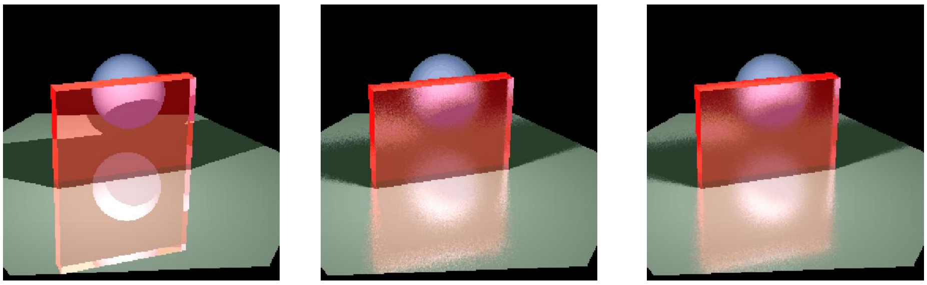
Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [8]: Fuzzy Translucency

- Same as glossy reflections, but jitter refracted ray
- Analytical function similar to shading
  - Transmission function is used instead of reflectance function
  - Light is gathered from other side of surface.
- Cast randomly distributed rays in general direction of transmitted ray from traditional ray tracing
- Average value computed from each of these rays: true translucent component

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [9]: Soft Reflection & Transparency

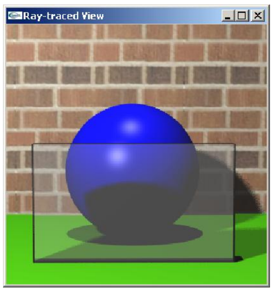


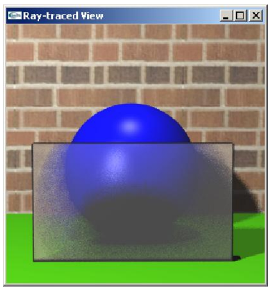
- First image is obtained from traditional ray tracer
- Second image uses 10 rays for transmitted ray
- Third image uses 20 rays

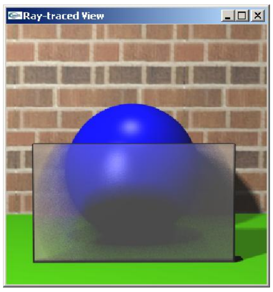
"Distributed Ray Tracing" © 1995 A. Martin – http://bit.ly/ex5ZUm

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [10]: Shadows, Reflection, Transparency







- First image is from traditional ray tracer
- Second one uses 16 rays in place of single reflected ray
- Third image uses 64 rays

"Monte Carlo Ray Tracing"
© 2005 D. Strock, University of Wisconsin Milwaukee
http://bit.ly/fSIYsL

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [11]: Penumbras (Soft Shadows)

- Traditional ray tracing shadows: discrete
  - Shadow feelers used to check if point is in shadow with respect to point light source
- Incorrect for large light sources and/or light sources that are close to object
- Transition from fully shadowed to partially shadowed is gradual
  - Due to finite area of real light sources
  - Also due to scattering of light of other surfaces

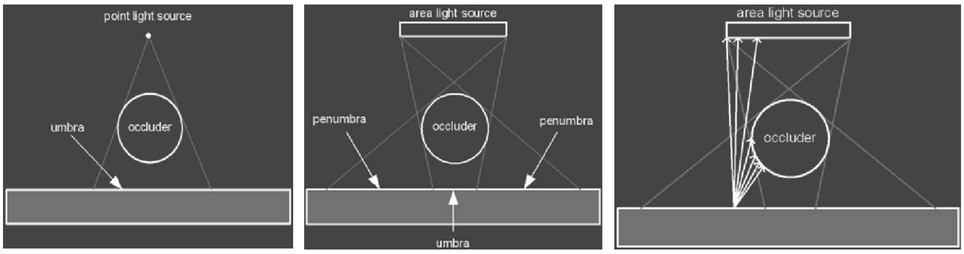
Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [12]: Tracing Penumbras (Soft Shadows)

- Set of rays cast about projected area of light source
  - Projected area helps tackle large area light source
- Amount of light transmitted by: ratio of number of rays that hit source to number of rays cast

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [13]: Shadow Feelers & Penumbras

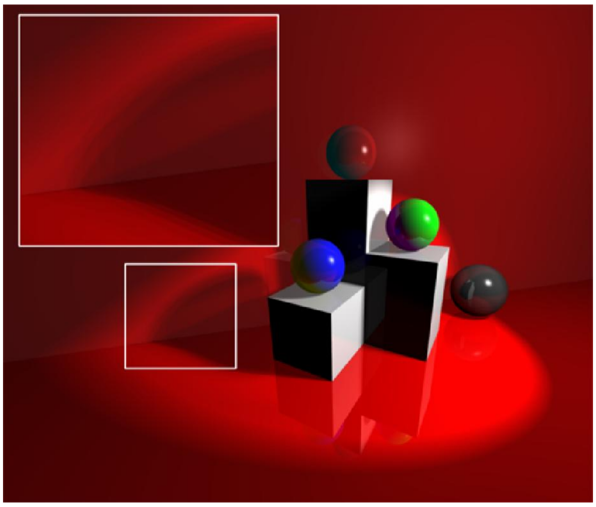


"Realistic Raytracing" © 2003 Z. Waters & E. Agu, WPI
http://bit.ly/gUNeGr

- In case of point source, occluder would create sharp shadow boundary
- In area light source or if light source is closer to object
  - Creation of penumbra region
- Sending out shadow feelers to capture penumbra region

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

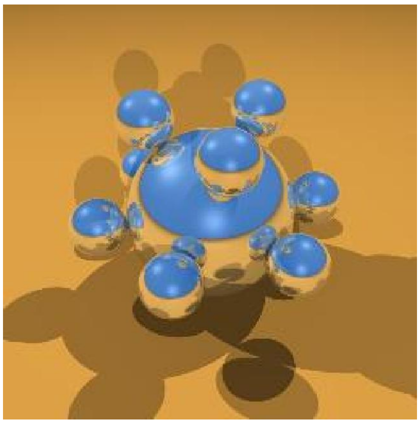
Distributed Ray Tracing [14]: Example – Transitions inside Penumbra



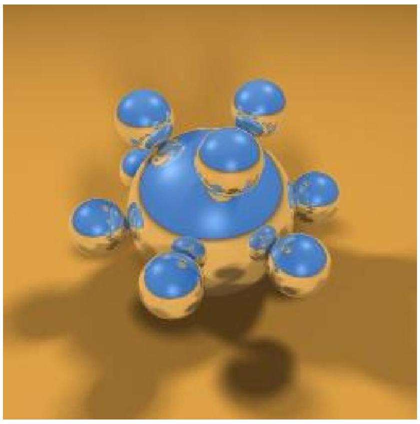
"Realistic Raytracing" © 2003 Z. Waters & E. Agu, WPI
http://bit.ly/gUNeGr

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [15]: Example – Soft Shadows



Regular RT



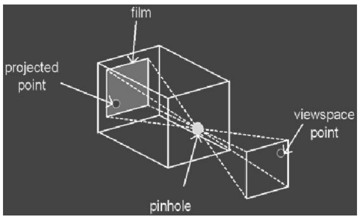
Distributed RT

♥ 2000 A. G. Zaferakis, UNC Chapel Hill
COMP 238, Advanced Image Generation – http://bit.ly/fGYzgw / http://bit.ly/dNQHtH

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [16]: Depth of Field

- Distance at which objects appear in focus
- Objects too far away or too close appear unfocused, blurry
- Pinhole camera model does not truly mimic real world
  - Pinhole assumed to be infinitely small
  - Changing focal length changes field of view but does not change focus



"Realistic Raytracing" © 2003 Z. Waters & E. Agu, WPI
http://bit.ly/gUNeGr

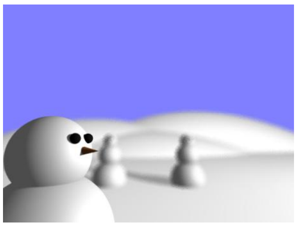
Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [17]: Creating Depth of Field

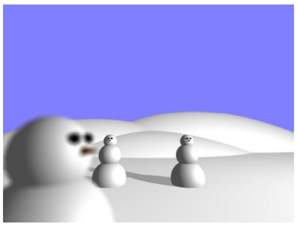
- Distributed RT: places artificial lens in front of view plane
- Randomly distributed rays: used once again to simulate blurring of depth of field
  - First ray cast not modified by lens
  - Focal point of lens is at fixed distance along this ray
  - Rest of rays sent out for same pixel scattered about surface of lens
  - Points in scene close to focal point of lens: in sharp focus
  - Points closer or further away: blurred

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

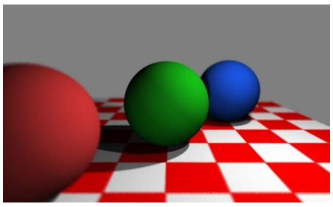
Distributed Ray Tracing [18]: Example – Depth of Field



Low Focal Distance, © 2004 S. C. Mikula (http://bit.ly/fK6ckX)



High Focal Distance, © 2004 S. C. Mikula (http://bit.ly/fKA3b3)



Intermediate Focal Distance, © 2003 A. Bair (http://bit.ly/faXUFh)

♥ 2001-2006 respective authors, University of Illinois at Urbana-Champaign
CS 419, Advanced Computer Graphics – http://bit.ly/e0UfsN

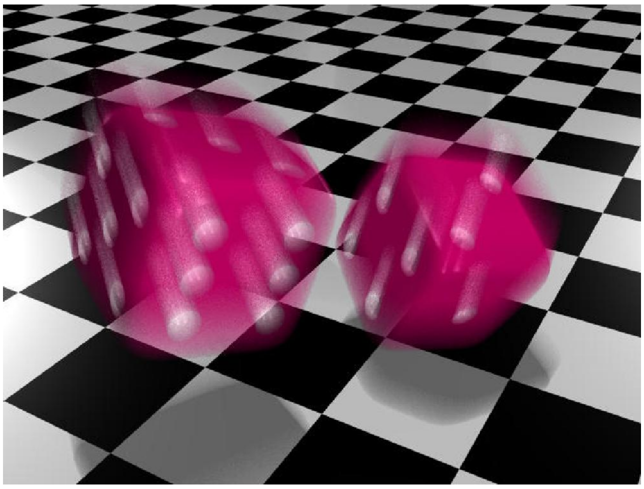
Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [19]: Motion Blur

- Temporal sampling rather than spatial sampling
- Frame represents average of scene during time that camera shutter is open
- Before each ray is cast, objects are translated or rotated to their correct position for that frame
- Rays are averaged to give actual value
- Objects with most motion will have most blurring in rendered image

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Distributed Ray Tracing [20]: Example – Motion Blur



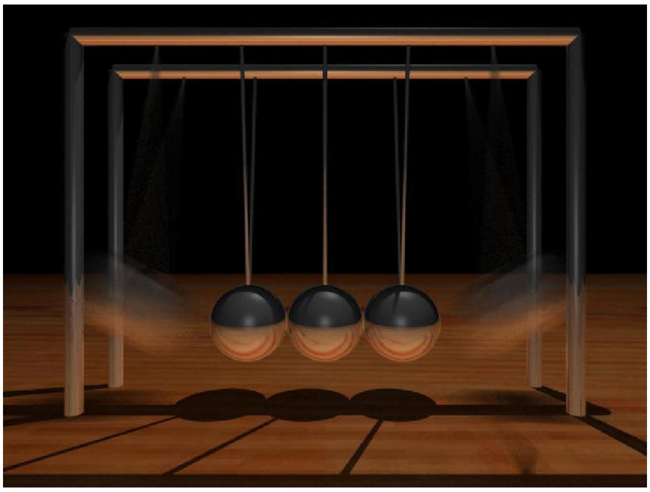
♥ 2005 C. M. Cameron, University of Illinois at Urbana-Champaign
CS 419, Advanced Computer Graphics – http://bit.ly/hmZU3x

ILLINOIS

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

UNIVERSITY OF DELAWARE

Distributed Ray Tracing [21]: Example – Soft Shadows



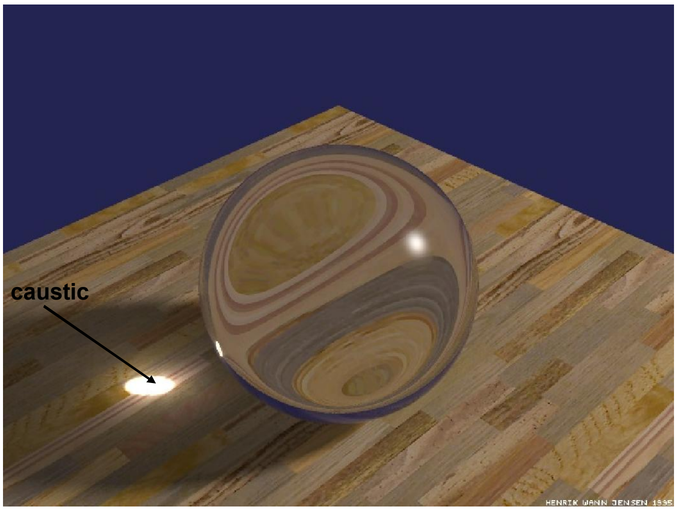
♥ 2005 M. A. Townsend, University of Illinois at Urbana-Champaign
CS 419, Advanced Computer Graphics – http://bit.ly/dL8GrH

ILLINOIS

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

UNIVERSITY OF DELAWARE

Bidirectional Ray Tracing [1]: Example – Caustic

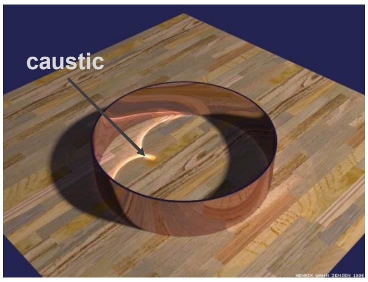


© 1996 H. W. Jensen, University of California, San Diego
http://graphics.ucsd.edu/~henrik/

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Bidirectional Ray Tracing [2]: Example – Caustic

- Caustic – (concentrated) specular reflection/refraction onto diffuse surface
- Standard ray tracing cannot handle caustics

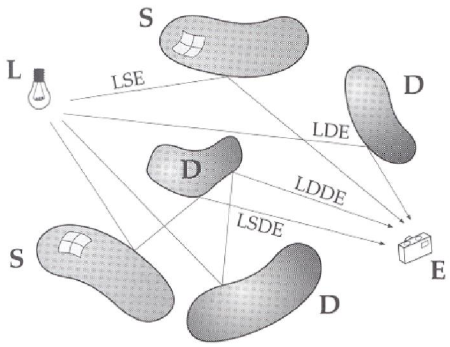


© 1996 H. W. Jensen, University of California, San Diego
http://graphics.ucsd.edu/~henrik/

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Light Paths: Abbreviated Notation

- Shown: interactions of light ray
- Can be expressed using regular expressions
  - ✸ L: light source
  - ✸ E: eye/camera
  - ✸ D: diffuse surface
  - ✸ S: specular surface

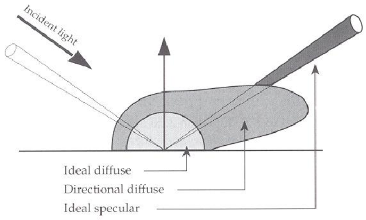


Sillion, F. X. & Puech, C. (1994). ***Radiosity and Global Illumination.*** San Francisco, CA: Morgan-Kaufmann.
Amazon: http://amzn.to/evNBJH

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

BRDF Revisited: Diffuse Surfaces [1]

- Uncertainty in direction photon will take for diffuse surfaces
- Specular surfaces: Bidirectional Reflectance Distribution Function (probability that incoming photon will leave in particular direction) has thin profile
  - ✸ Can predict direction of outgoing photon
- For ideal diffuse surfaces, BRDF would be spherical
  - ✸ Photon can travel along any direction with equal probability

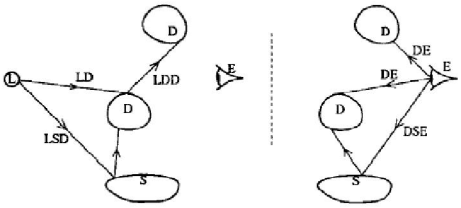


Sillion, F. X. & Puech, C. (1994). ***Radiosity and Global Illumination.*** San Francisco, CA: Morgan-Kaufmann.
Amazon: http://amzn.to/evNBJH

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

BRDF Revisited: Diffuse Surfaces [2]

- Idea: Trace forward light rays into scene as well as backward eye rays
  - At diffuse surfaces, light rays additively “deposit” photons in radiosity textures, or “rexes”, where they are picked up by eye rays
- Rays of forward and backward pass "meet in middle" to exchange information

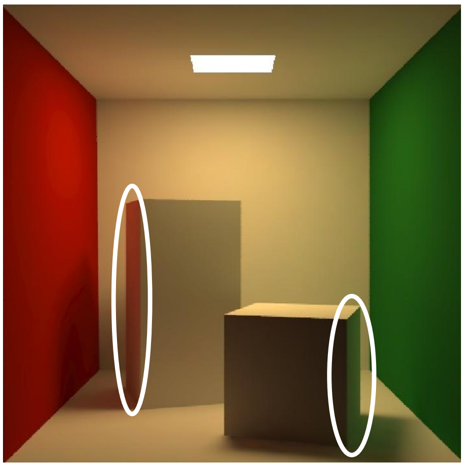


Heckbert, P. S. (1990). “Adaptive radiosity textures for bidirectional ray tracing ”, *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques* (Scott 1990).

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Radiosity

- Handling cases such as LD*E
- “Color Bleeding”



Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Software Packages

- Two excellent, full-featured rendering & modeling packages
- *POV-Ray* (http://www.povray.org/)
  - *Persistence of Vision Ray Tracer*
  - Free rendering tool (not modeling tool)
  - Uses text-based scene description language (SDL)
  - Available on Windows, Linux, Mac OS
- *Blender* (http://www.blender3d.org)
  - Modeling, animation, rendering tool
  - Especially useful in 3-D game creation
  - Available for Windows, Linux, Irix, Sun Solaris, FreeBSD or Mac OS X under GPL

Adapted from slides ♥ 2005 M. Thomas & C. Khambamettu, U. Del.
CISC 440/640: Computer Graphics, Spring 2005 – http://bit.ly/hz1kfU

Radiosity [1]: Basic Idea

- “Radiosity" method: basis is field of thermal heat transfer
- Heat transfer theory describes radiation as transfer of energy from surface when that surface has been thermally excited

Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

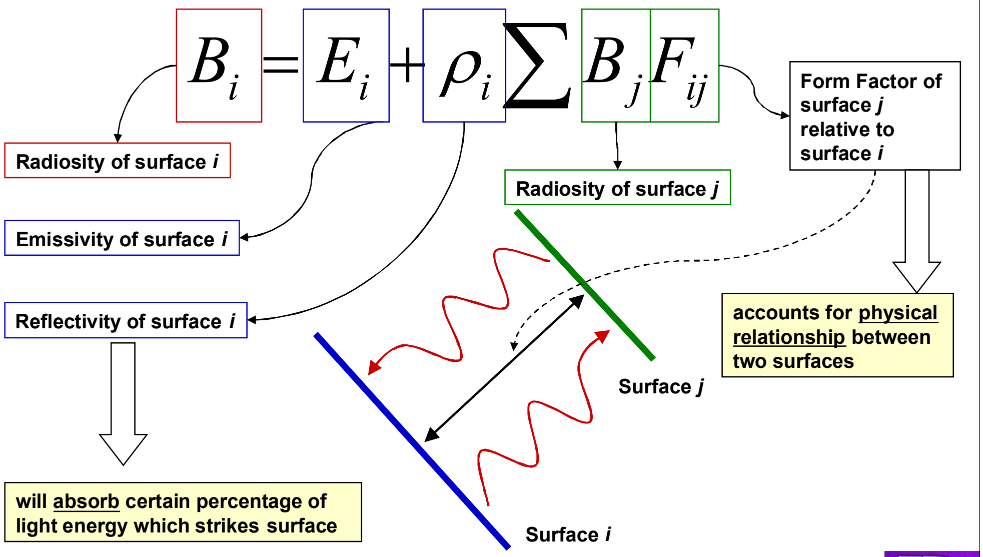
Radiosity [2]: Derivation of Radiosity Equation

- **Radiosity equation** describes **amount of energy** which can be emitted from surface, as sum of
  - energy inherent in surface (*e.g.*, light source)
  - energy which strikes surface, being emitted from some other surface
- Energy which leaves surface *j* and strikes another surface *i* is attenuated by two factors
  - "form factor" between surfaces *i* and *j*, which accounts for physical relationship between two surfaces
  - the reflectivity of surface *i*, which will absorb some percentage of light energy striking surface

Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

Radiosity [3]: Radiosity Equation

$$B_i = E_i + \rho_i \sum B_j F_{ij}$$



Radiosity of surface *i*

Emissivity of surface *i*

Reflectivity of surface *i*

will absorb certain percentage of light energy which strikes surface

Radiosity of surface *j*

Form Factor of surface *j* relative to surface *i*

accounts for physical relationship between two surfaces

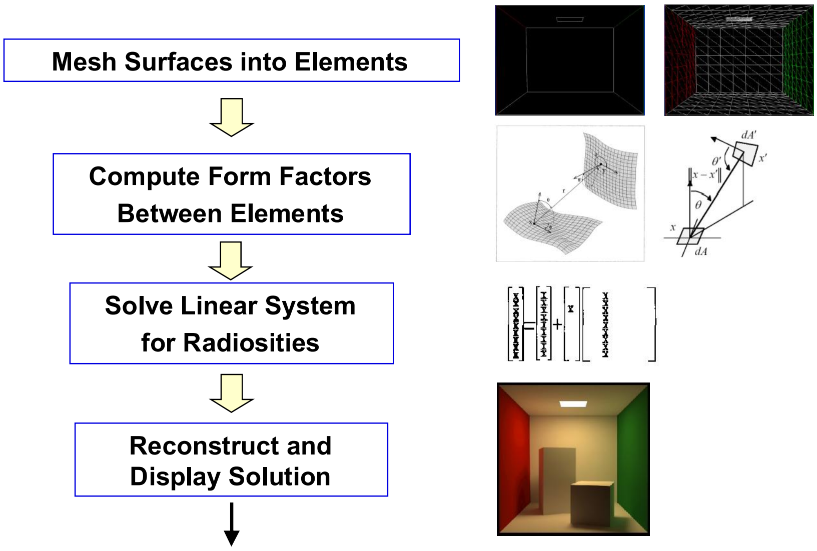
Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

Radiosity [4]: Implementation, Pros & Cons

- Classic radiosity = finite element method
- Assumptions
  - Diffuse reflectance
  - Usually polygonal surfaces
- Advantages
  - Soft shadows and indirect lighting
  - View independent solution
  - Precompute for set of light sources
  - Useful for walkthroughs

Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

Radiosity [5]: Classic Radiosity Algorithm



Mesh Surfaces into Elements

Compute Form Factors Between Elements

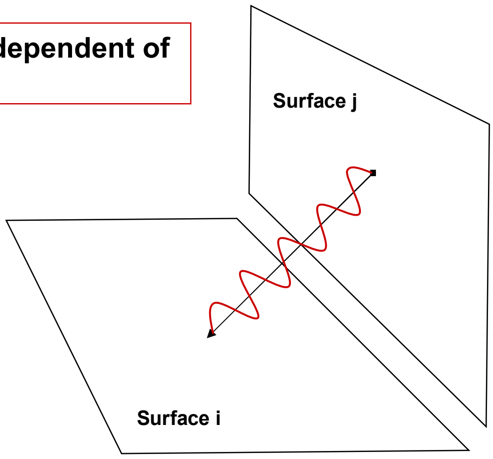
Solve Linear System for Radiosities

Reconstruct and Display Solution

Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

Radiosity [6]: Form Factors Illustrated

Purely geometric relationship, independent of viewpoint or surface attributes



Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

Radiosity [7]: Formula – Form Factor

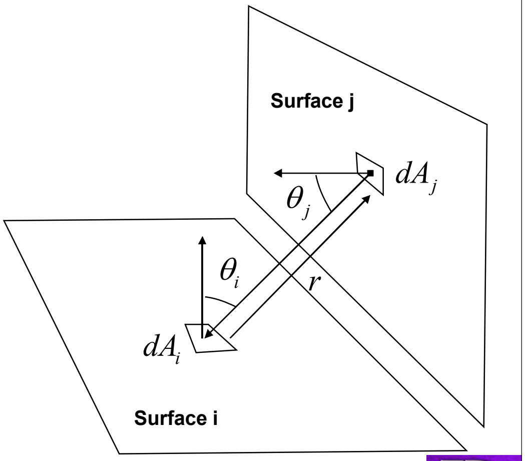
$$\boxed{FdA_jdA_j} = \frac{\boxed{\cos\theta_i}\boxed{\cos\theta_j}}{\pi|r|^2}$$

differential area of surface *I, j*

angle between Normal$_i$ and ***r***

angle between Normal$_j$ and ***r***

vector from dA$_i$ to dA$_j$

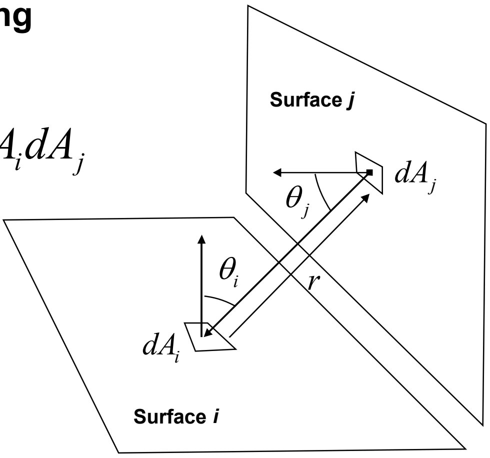


Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

Radiosity [8]: Formula – Overall Form Factor

Overall form factor between *i* and *j* is found by integrating

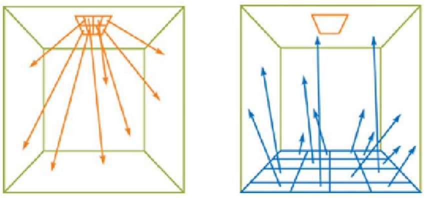
$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi |r|^2} dA_i dA_j$$



Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

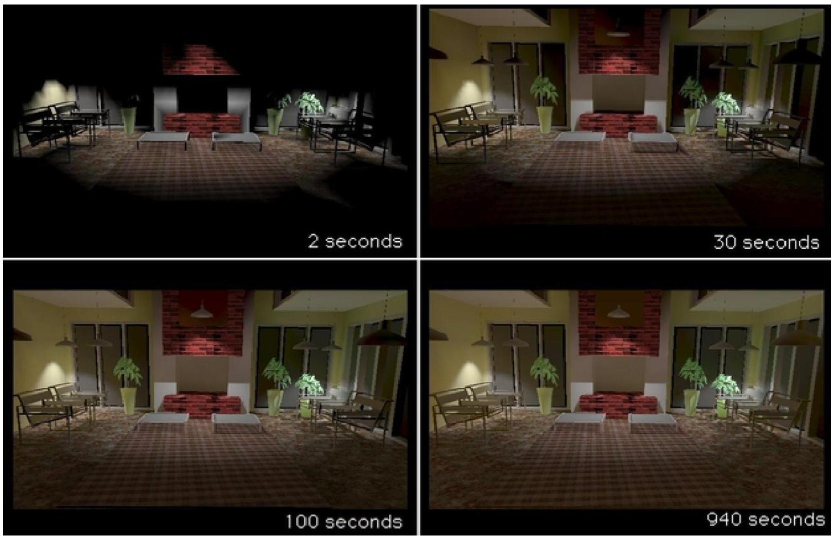
Radiosity [9]: Progressive Refinement

- Each element in scene maintains two energy values
  - Accumulated
  - Residual (“unshot”)
- Choose one element as shooter
- Test visibility of every other element from this shooter
  - If visible, calculate shooter-to-receiver energy transfer
  - Based on shooter’s residual, receiver’s reflectance
- Progressive refinement: reset residual, repeat with new shooter
- Terminate when shooter residuals below threshold



Adapted from *GPU Gems 2*, Chapter 39, “Global Illumination Using Progressive Refinement Radiosity”, © 2005 G. Coombe & M. Harris, nVidia Corporation – http://bit.ly/hXQ8Zd

Radiosity [10]: Example – Progressive Refinement

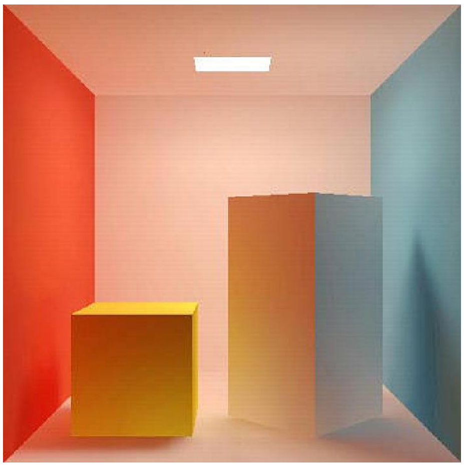


Myszkowski, K. (2001). ***Efficient and Predictive Realistic Image Synthesis.*** **Habilitation thesis, Warsaw University of Technology – http://bit.ly/gij9k6**

Politechnika Warszawska

Radiosity [11]: Example – Cornell Box

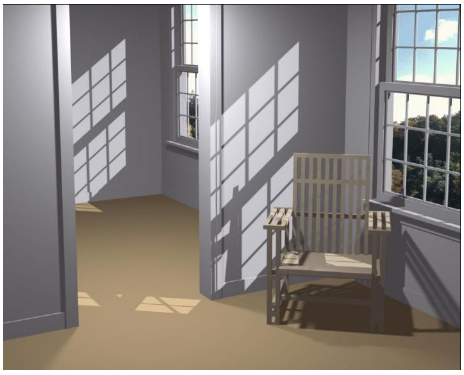
- **This simulation of the Cornell box was done by Michael F. Cohen and Donald P. Greenberg for the 1985 paper *The Hemi-Cube, A Radiosity Solution for Complex Environments*, Vol. 19, No. 3, July 1985, pp. 31-40.**
- **The hemi-cube allowed form factors to be calculated using scan conversion algorithms (which were available in hardware), and made it possible to calculate shadows from occluding objects.**



Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

Radiosity [12]: Example – Discontinuity Meshing

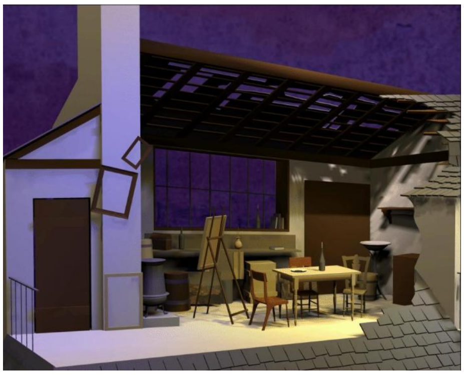
- Dani Lischinski, Filippo Tampieri and Donald P. Greenberg created this image for the 1992 paper *Discontinuity Meshing for Accurate Radiosity*.
- It depicts a scene that represents a pathological case for traditional radiosity images, many small shadow casting details.
- Notice, in particular, the shadows cast by the windows, and the slats in the chair.



Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

Radiosity [13]: Example – Focused Opera Lighting

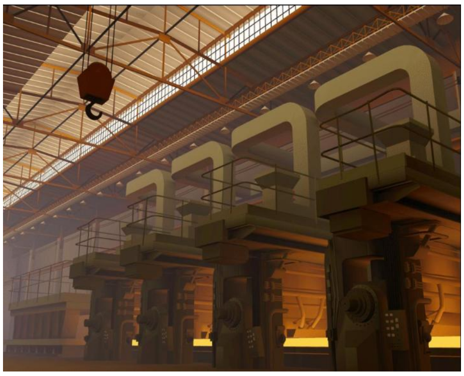
- This scene from *La Bohème* demonstrates the use of focused lighting and angular projection of predistorted images for the background.
- It was rendered by Julie O'B. Dorsey, Francois X. Sillion, and Donald P. Greenberg for the 1991 paper *Design and Simulation of Opera Lighting and Projection Effects*.



Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

Radiosity [14]: Formula – Overall Form Factor

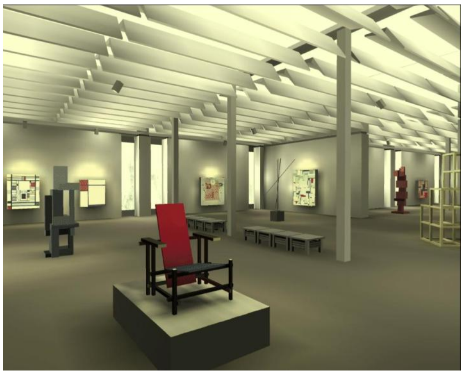
- These two images were rendered by Michael F. Cohen, Shenchang Eric Chen, John R. Wallace and Donald P. Greenberg for the 1988 paper *A Progressive Refinement Approach to Fast Radiosity Image Generation*.
- The factory model contains 30,000 patches, and was the most complex radiosity solution computed at that time.
- The radiosity solution took approximately 5 hours for 2,000 shots, and the image generation required 190 hours; each on a VAX8700.



Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

Radiosity [15]: Example – Cornell Virtual Museum

- Most of the illumination that comes into this simulated museum arrives via the baffles on the ceiling.
- As the progressive radiosity solution executed, users could witness each of the baffles being illuminated from above, and then reflecting some of this light to the bottom of an adjacent baffle.
- A portion of this reflected light was eventually bounced down into the room.
- The image appeared on the proceedings cover of SIGGRAPH 1988.



© 1988-1989 E. Chen & M. Cohen,
Cornell University – http://bit.ly/e7Y1tj

Cornell University
Department of Computer Science

Adapted from slides ♥ 2001 - 2005 M. C. Sousa, University of Calgary
CPSC 591/691: Rendering, Spring 2001 through Spring 2011 – http://bit.ly/ftMP0G

Summary

- Reading for Last Class: Ray Tracing Handout
- Reading for Today: Chapter 15, Eberly *2^e*; Ray Tracing Handout
- Reading for Next Class: Tufte Handout 1
- Last Time: Ray Tracing Lab
  - ACM SIGGRAPH demo: http://bit.ly/cllgx2
    - 2-D "screen"
    - Moveable objects: transparent, opaque (both reflective)
  - POV-Ray (http://www.povray.org) Example Renderings
- Next Class: Visualization Part 1 of 3

Terminology

- Caustic: Envelope of Light Rays Reflected/Refracted by Curved Object
- RT Direction
  - "Forward" RT: Light-to-Scene, Scene-to-Eye (Only for Caustics)
  - "Backward" RT: Eye-to-Scene, Scene-to-Light (Typical Order)
  - Bidirectional RT: both directions (meet in middle)
- Stochastic Jitter: Local Random Perturbations of Traced Rays
- Distributed RT: Nonlocal Randomization
- Penumbra: Region Where Only Part of Light Source Blocked
- Blurring
  - Soft shadows: blurred penumbras (achieved using shadow feelers)
  - Gloss: property of smooth surface material (multiple reflected rays)
  - Reflections: soften (also distributed RT)
  - Transparency: lensed caustic effect (also distributed RT)
- Form Factor: Fraction of Energy Leaving Surface *i* That Reaches *j*
- Radiosity: Heat Transfer-Based Global Illumination Method