SIGGRAPH 98
25th International Conference
on Computer Graphics and
Interactive Techniques

COURSE NOTES 5
A BASIC GUIDE TO GLOBAL ILLUMINATION

Sunday, July 19, 1998
Half Day Course

ORGANIZER
Holly Rushmeier
IBM TJ Watson Research Center

LECTURERS

David Banks
NSF Engineering Research Center
Mississippi State University

Holly Rushmeier
IBM TJ Watson Research Center

Peter Shirley
Department of Computer Science
University of Utah

ABSTRACT

Images of the real world are formed by visible light being scattered by surfaces and volumes. The goal of global illumination methods is to simulate the path of light in an environment through the image plane in order to compute realistic images. Not all applications require the accuracy attainable with global illumination methods, and not all global illumination methods are good for all possible lighting effects. In this course the audience will be given a vocabulary and taxonomy for understanding global illumination. Insight into the basic methods will be provided using comparison to physical experiments. The target audience includes: people who are new to graphics who want to be generally informed, people who teach graphics courses but specialize in some other area of graphics, and/or people who think they may need global illumination for their application and want to understand how these methods differ from other rendering techniques.

ABOUT THE SPEAKERS

David C. Banks

Assistant Professor of Computer Science
NSF Engineering Research Center
Mississippi State University, MS 39762
601/325-0528 (voice)
601/325-8997 (fax)
banks@cs.msstate.edu

David Banks received his PhD from the University of North Carolina at Chapel Hill and held a post-doctoral position at the Institute for Computer Applications in Science and Engineering (ICASE) at NASA Langley Research Center. His research interests include applying computer graphics to study large-dimensional problems. He teaches graphics and visualization courses for undergraduate and graduate students.

Holly Rushmeier
Research Staff Member
IBM TJ Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532
914/784-7252 (voice)
914/784-7667 (fax)
holly@watson.ibm.com

Holly Rushmeier is a research staff member at the IBM TJ Watson Research Center. She received the BS(1977), MS(1986) and PhD(1988) degrees in Mechanical Engineering from Cornell University. Since receiving the PhD, she has held positions at Georgia Tech, and at the National Institute of Standards and Technology. In 1990, she was selected as a National Science Foundation Presidential Young Investigator. In 1996, she served as the Papers chair for the ACM SIGGRAPH conference, and she is currently Editor-in-Chief of ACM Transactions in Graphics. She has published numerous papers in the areas of data visualization, computer graphics image synthesis and thermal sciences. In the area of global illumination she has worked on the problems of comparing real and synthetic images, imaging participating media, and combining ray tracing and radiosity methods. Most recently she has worked on global illumination methods suitable for image based rendering, accurate tone reproduction for high dynamic range images, and systems for acquiring physical data for realistic rendering.

Peter Shirley
Assistant Professor
3190 Merrill Engineering Building

Department of Computer Science
University of Utah
Salt Lake City, UT 84112
801/581-5290 (voice)
801/581-5843 (fax)
shirley@cs.utah.edu

Peter Shirley is an Assistant Professor at the University of Utah. He has a BA in Physics from Reed College and a Ph.D. in Computer Science from the University of Illinois. He worked at Indiana University and the Cornell Program of Computer Graphics before joining Utah. His research interests include realistic rendering, illustration, and visualization. He has taught several undergraduate and graduate courses on computer graphics in general, and global illumination in particular.

SYLLABUS

1:30 - 2:15 pm
Motivation and Definitions
David Banks and Holly Rushmeier
- presentation notes pp. 2-1 to 2-13

2:15-3:00 pm
Ray Tracing
David Banks
- presentation notes pp. 3-1 to 3-18

(3:00-3:15 pm Break)

3:15 - 4:00 pm
Radiosity
Peter Shirley
- presentation notes pp. 4-1 to 4-6

4:00 - 4:45 pm
Current Trends
Holly Rushmeier
- presentation notes pp. 5-1 to 5-7

4:45 pm Questions and Answers

TABLE OF CONTENTS

A Basic Guide to
Global Illumination

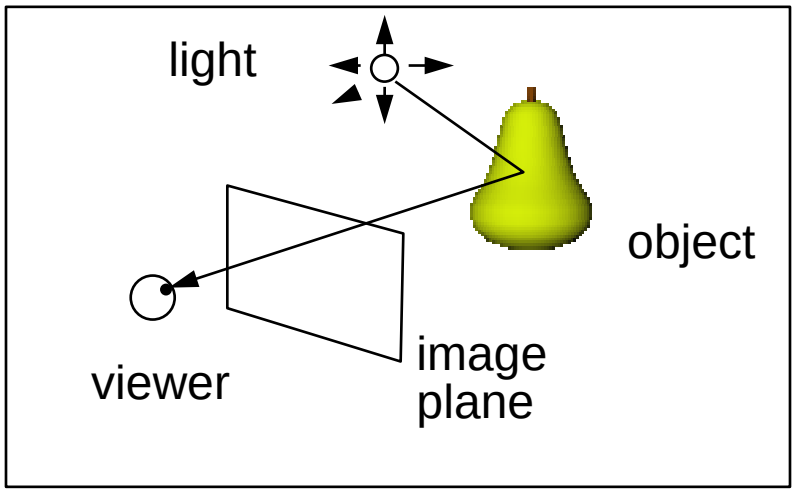
Motivation and Definitions

The purpose of this course is to give an overview of the area of computer graphics that has come to be known as "global illumination". The goal of global illumination is to make images of scenes which are defined numerically (may not physically exist yet) The images are predictive −− they are intended to show how the scene would appear if it were actually built. This is as opposed to artistic images or diagrams which may illustrate an individual's idea of what a scene would look like. Global illumination simulates the physical phenomenon of light transport.

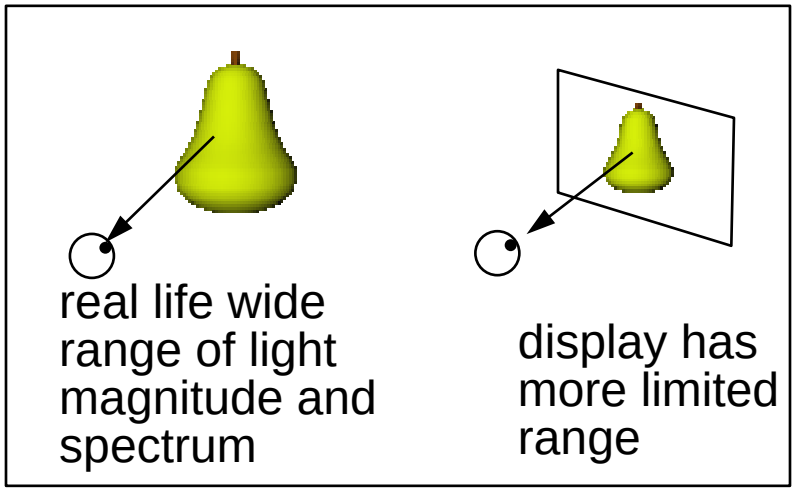
Applications:

- Product appearance design
- Safety design
- Artistic effect acheived by physical means.

There are many reasons to make images. In many cases it is desirable to make "non−photorealistic" images that emulate artistic techniques such as sketching and painting. Global illumination is used when accurate predictions are needed for applications such as: what will the car look like in the showroom? Will the dashboard be visible to a driver at night? Will this theatrical lighting setup achieved the desired dramatic effect for a performance?



We see things as a result of how they interact with visible light To form an image we select a view point, view direction, image plane and image resolution. We color each pixel in the image according to what object would be visible through that pixel, and what quantity of light would be leaving that object in the direction of the viewer.



The RGB (red,green,blue) values we ultimately choose will not produce the same quantity of light on our display as we would encounter in real life. Displays have limited color gamuts and dynamic ranges. Mappings are needed to convert the quantity of light we predict to something displayable. These mappings use models of the human visual system. So, unfortunately to completely understand the formation of a realistic image, some knowledge of both the physics of light and the psychophysics of humans is needed.

Isn’t lighting built into graphics hardware for making shaded images?

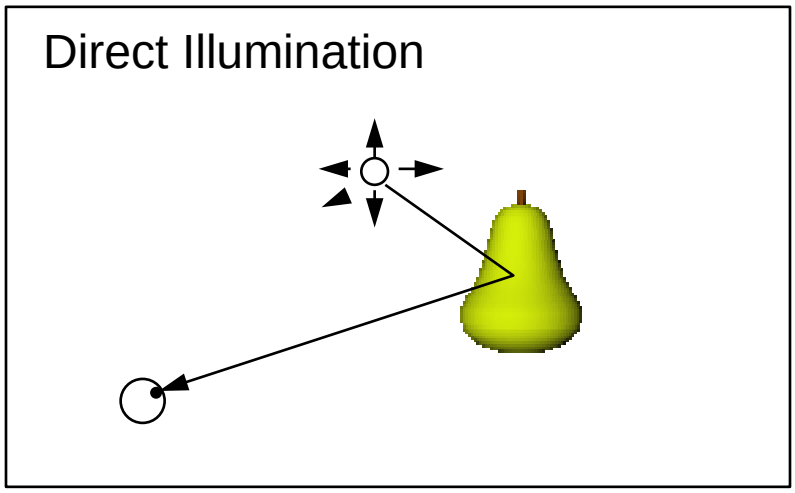
Systems such as Open GL and VRML have "lighting models" that are heuristics that emulate some lighting effects to render objects with shape and texture. However, these systems do not allow the definition of real light sources, physically realizable reflectances, do not include the "inverse square law", often have no shadows (or just sharp ones) and do not account for interreflections

---

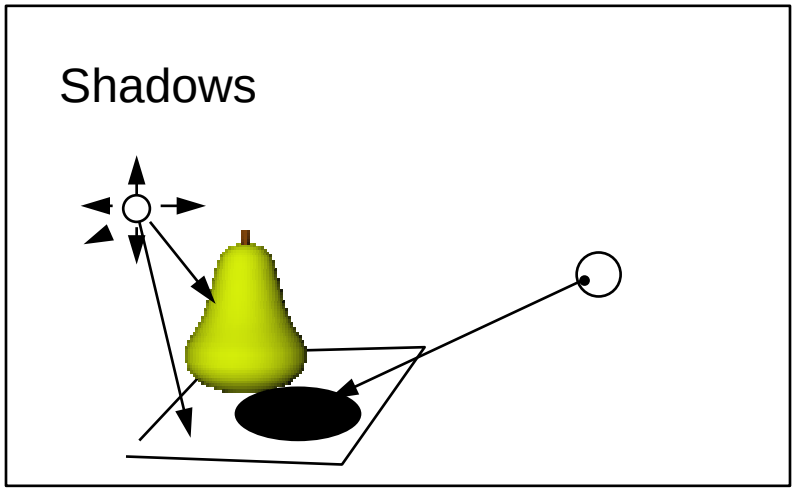
Fundamental Effects:

- Direct Illumination
- Shadows
- Interreflections
- Volumes

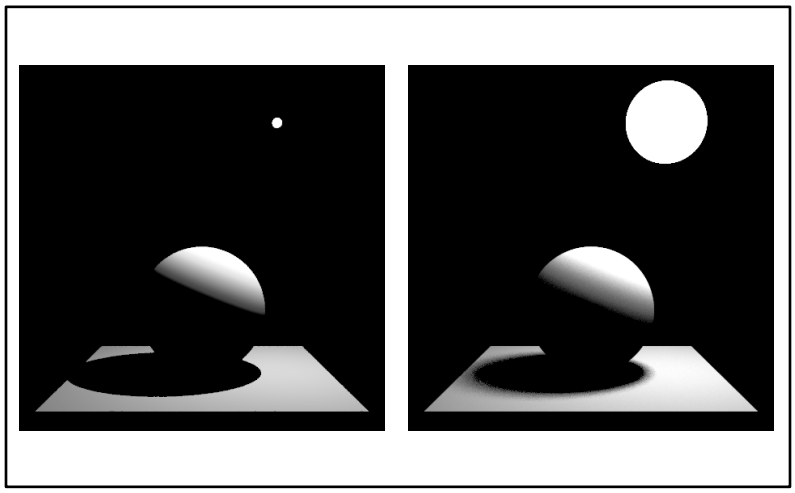
Let’s examine the fundamental effects that are achieved with global illumination that are not achieved by heuristic graphics lighting. Not all of these effects are equally important in every application, and they are certainly not all equally easy to compute. Sometimes they can be approximated by simple methods, but in some cases extensive calculations are required to get an adequate image. It is important to understand the effects critical to an application to determine the most cost effective approach to computing an accurate image.



"Direct Illumination" refers to light that arrives at an object directly from the light source and then is reflected to the viewer. To accurately compute direct illumination, appropriate definitions of the geometry, directional, and spectral composition of the light source and the reflectance function of the object are needed. Modeling light sources and reflectances is sometimes referred to as "local illumination." See the section "Global Illumination Input" for more details.

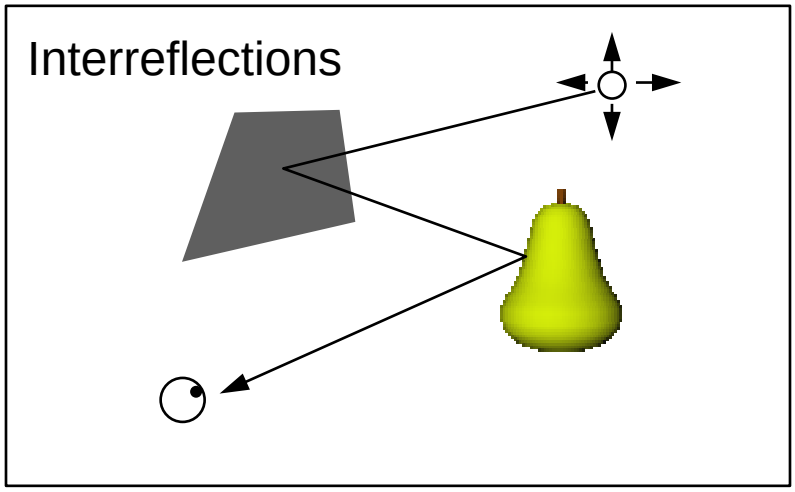


It is also important to find where light from the source does not reach an object. Shadows are an important cue to object locations −− we have the sense that the pear is floating above the plane becase of the location of the black ellipse used to represent its shadow.

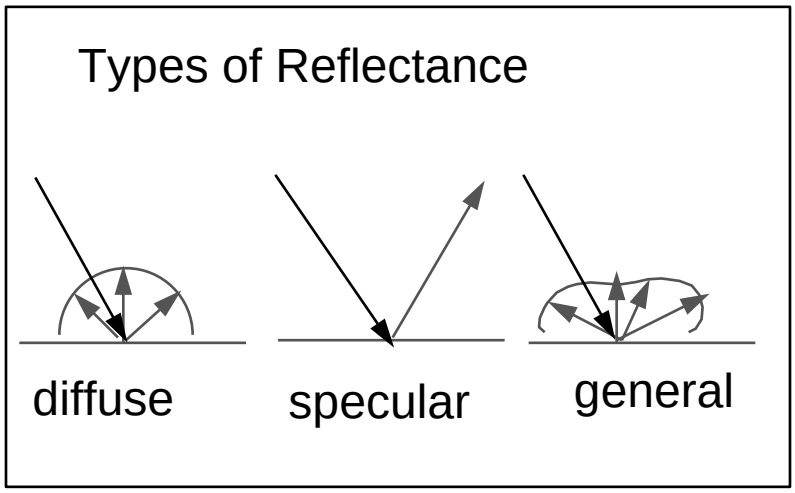


The shadow area where no light reaches is the umbra. There are some points where parts of the light source only are seen. called penumbra, which make the edges of the shadow look fuzzy. Whether the shadow is fuzzy or not depends on the sizes of the source and occlusion relative to the distances to the source and occlusion.

Many algorithms treat just the problem of how to compute shadows.. Classic techniques are Crow's shadow volumes (Crow82) and Williams' shadow maps (Williams78).

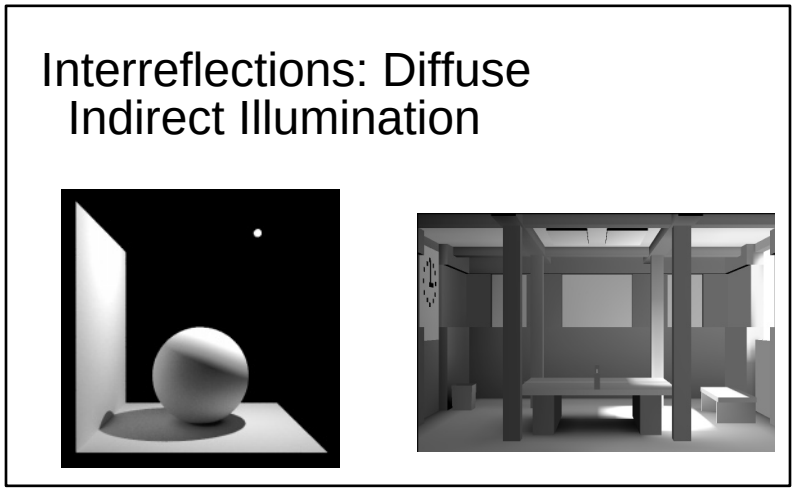


Interreflections are the "global" part of global illumination. The light that ultimately reaches the eye and has an effect on the image often goes through more than one bounce. Interreflections are expensive to compute −− in some scenes where they are not important it may be possible to neglect them or approximate crudely with simple calculations.



The effect of interreflections depends on the directional properties of the surfaces involved. Diffuse (a.k.a. Lambertian, matte) surfaces are characterised by the fact that you can't make out any objects reflected in the surface. Specular (a.k.a mirror−like) surfaces are characterized by the fact that you can see reflected objects clearly −− i.e. that's why we use them for mirrors ;) Many surfaces are neither of the idealized cases −− and reflections of objects may be seen dimly or fuzzed−out in a general surface

---

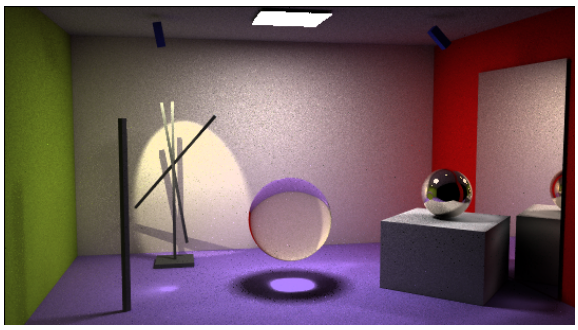


Indirect illumination, generally reflection off diffuse surfaces may cause surfaces which have no direct view of the light source to be illuminated. This is dramatic when there are many surfaces in the scene with no direct view of the source, as in these examples. However these interreflections are expensive to compute, and if most surfaces have a view of a light source, the effect of interreflections might be adequately approximated by a constant value. Good early examples of the effect of indirection illumination are shown in Nishita and Nakamae's 1985 SIGGRAPH paper

Interreflections: Diffuse Color Bleeding

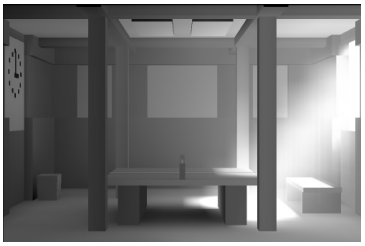
The color of objects depends on their spectral reflectance, and the spectrum of incident light. (This is made a little more comples by the "color constancy" human vision phenomenon −− see "From Solution to Image" in the appendix). If a white wall, for example, is illuminated by indirect illumination from a red wall, the white wall will look somewhat red. This is generally a subtle effect, and was illustrated by the "Cornell Box" (Goral84) shown above in a early incarnation.

Interreflections: Caustics Bright Spots



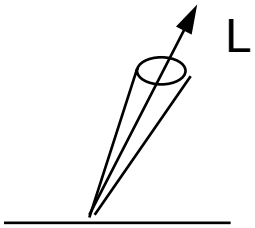
In graphics, "caustics" refer to bright spots that are the result of a path of reflection or multiple reflections, from the light source, by several specular surfaces, and then finally hitting a diffuse surface. An extreme example is shown here where a spot light on the ceiling at the right is aimed at a mirror which reflects light through a crystal ball which focusses light into a bright spot to the left of the ball on the floor. This is in addition to the bright spot that results from the crystal ball focussing the main ceiling light onto the floor. Combining different paths of interreflection was discussed in Chen et al ’91, in which this image appears

Fundamental Effects: Volumes



Besides surfaces, volumes of media can interact with light. Most of the time the volumetric medium in our environment (air) does not "participate" in the radiative transfer of visible light. However if there are water droplets (fog or clouds) or particulates (dust or smoke particles) in the air, these volumes of media partipate in the light transfer, and are called "participating media". Details on input and descriptions of participating media can be found in the appendix "Input for Participating Media."

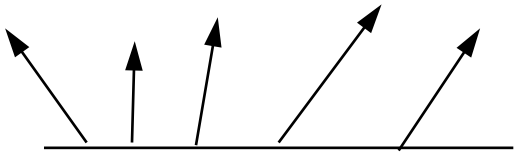
Definitions: RADIANCE



$$\frac{\text{energy}}{\text{time*projected-area*solid-angle}}$$

To form and solve equations for global illumination, we need to get specific on how to define a quantity of light. The basic quantity we want to solve for is radiance L The spectral radiance (i.e. radiance for various wavelengths of light) convolved with spectral functions related to the spectral sensitivities of the human visual system, will ulitimate be what we use to set the value of each pixel in an image.

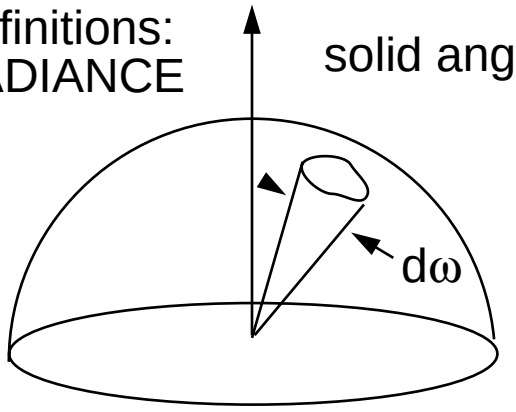
Definitions:
RADIANCE time and area



energy transport is continuous in time, and distributed over areas

Why this definition? Energy is continually being transferred in our problem. In a still image the rate that visible light per unit time is constant, but it is being transferred continually. A point has no dimension, so strictly speaking there is zero energy leaving a point. We can discuss energy/time at a point though if we express it as energy/(time*area).

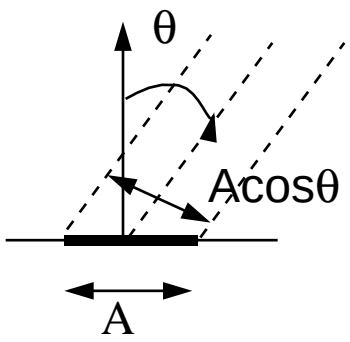
Definitions:
RADIANCE solid angle



The light leaving a point may be different in each direction. All of the directions around a point are included in a hemisphere over the point. The hemisphere is said to subtend 2π steradians over the surface, analogous to a half circle subtending π radians.
A solid angle is a chunk of that hemisphere of directions. By integrating over all solid angles we can account for either all of the light leaving the surface per unit time and area, or all of the light energy incident.

Definitions:
RADIANCE

projected area

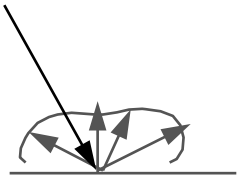


Why projected area and not just area in the radiance definition? When the surface is viewed at an angle, its area is forshortened by the cosine of the angle. By divided by projected area radiance expresses the quantity of light in terms of the effective surface area in the given direction

NOTE: Radiance is defined with respect to a surface, but not necessary a physical solid surface. Radiance is defined for any infinitesmal area specified by a location and surface normal, anywhere in space.

Reflectance: BRDF

Bidirectional
Reflectance
Distribution
Function



general

The other key definition is to precisely define the function that describes what happens to light when it is reflected from a surface. A reflectance is the ratio of reflected to incident light energy. A more general function expresses the directionaliuty of reflectance and is NOT a ratio that ranges from 0 to 1, but a distribution function that takes on any non−zero real value.

Reflectance: BRDF

$$f_r(\theta_i,\phi_i,\theta_r,\phi_r) = \frac{L_r(\theta_r,\phi_r)}{L_i(\theta_i,\phi_i)\cos\theta_i d\omega_i}$$

radiance/ energy flux density

The BRDF relates the reflected radiance in a particular direction (indicated here in spherical coordinates −− theta is the polar angle, phi is the azimuthal), to the incident energy flux density. For a general surface fr has a non−zero value for all pairs of incident and reflected directions.

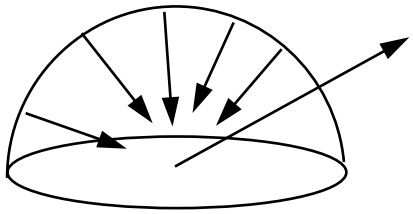
Special BRDF:

$f_r = \rho_d/\pi$ DIFFUSE

$f_r=\rho_s\,\delta(\theta_m)/\cos\theta$ SPECULAR

The BRDF for the idealized surface reflectances have a simple form. A diffuse surface has a BRDF that is the same for all incident and reflected directions. The value ρd is the ratio of reflected to incident light energy. π is in the denominator for the diffuse surface as a result of integrating all directions with a cosθ weighting factor. A specular surface reflects light in only one direction for a given incident direction, so its BRDF is a delta function

Reflected Radiance



$$L_r(\theta_r,\phi_r) = \int f_r(\theta_i,\phi_i,\theta_r,\phi_r)\, L_i(\theta_i,\phi_i)\cos\theta_i d\omega_i$$

To compute the radiance reflected from a point on a surface, we need to account for the fact that light may be incident from all directions, so we need to integrate over the entire incident hemisphere.

The Rendering Equation

$$L_o(\theta_r,\phi_r) = L_e(\theta_r,\phi_r) + \int f_r(\theta_i,\phi_i,\theta_r,\phi_r)\, L_i(\theta_i,\phi_i)\cos\theta_i d\omega_i$$

radiance from object

radiance emitted from object

radiance reflected from object

An object may emit and/or reflect light. The complete rendering equation gives the radiance leaving an object accounting for both effects. This is just the well known equation of radiative transfer as used in heat transfer, illumination engineering, and various area of physics. The seminal paper "The Rendering Equation" by Kajiya in 1986 pointed out that this is the equation we want to solve to generate accurate images, and that in fact all of the approximations that had been made in an attempt to make realistic images were in some way an attempt to solve this equation.

PUBLICATIONS REFERRED TO IN
MOTIVATION and DEFINITIONS

F.C. Crow, Shadow algorithms for computer graphics. In J.C. Beatty and K.S. Booth (eds.), Tutorial: Computer Graphics, Silver Spring, MD: IEEE Comput. Soc. Press, 1982.

L. Williams, Casting curved shadows on curved surfaces. Proc. of SIGGRAPH ’78 , Computer Graphics 12(3):270−274, 1978.

C.M. Goral, K.E. Torrance, D.P. Greenberg and B. Battaile Modeling the interaction of light between diffuse surfaces. Proc. of SIGGRAPH ’84, Computer Graphics, 18(3):213−222, 1984.

Tomoyuki Nishita and Eihachiro Nakamae Continuous tone representation of three−dimensional objects taking account of shadows} and interreflection, Computer Graphics (ACM SIGGRAPH ’85 Proceedings)}, 19, (3) pages = 23−−30.

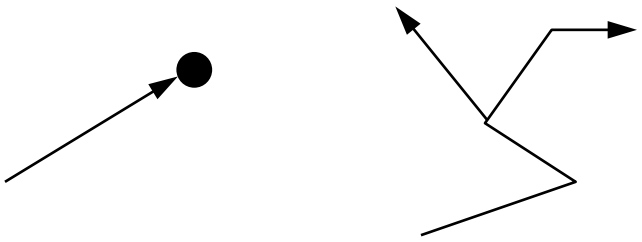
Shenchang Eric Chen and Holly E. Rushmeier and Gavin Miller and Douglass Turner, A progressive multi−pass method for global illumination Computer Graphics (ACM SIGGRAPH ’91 Proceedings). 25(4) 164−−174.

James T. Kajiya, The rendering equation, Computer Graphics (ACM SIGGRAPH ’86 Proceedings) 20(4), 143−−150.

Ray Tracing

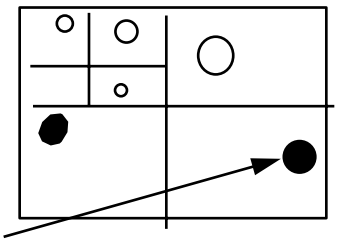
One of the basic classes of global illumination solutions is ray tracing. Ray tracing involves following paths or trees of line segments through the scene to compute the effects of typical light paths.

Ray Casting vs. Ray Tracing



The terms "ray casting" and "ray tracing" are sometimes confused. Ray casting refers to intersecting an individual ray with objects in the scene to find the first visible surface. Ray tracing involves finding paths or trees of line segments −− ray casting is a basic tool in finding these paths.

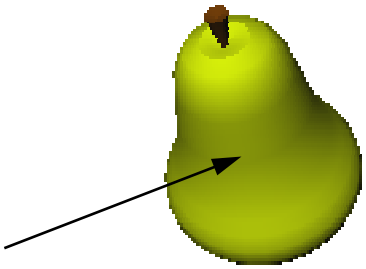
Ray Casting



Spatial organization for culling

A couple of notes on ray casting are worthwhile since it is a fundamental operation performed millions of times in a ray traced image. One area of research in ray casting has been avoiding as many intersection tests as possible. A typical approach is to sort surfaces into a spatial structure, such as an octree (Glassner84). Intersection tests are only performed on surfaces in the cells traversed by the ray. In this two−d example, no tests are needed for the hollow circles −− just for the two solid circles.

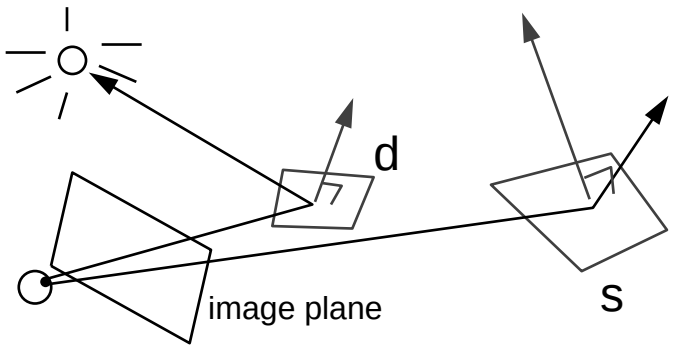
Ray Casting



Efficient intersections for NURBs etc.

Another area of interest in ray casting is efficiently intersecting a ray with a particular surface. Once it is known that a ray intersects a surface such as a NURB (non−uniform rational B−spline), or a quadric surface (e.g. ellipsoid) the intersection point should be computed with a minimal number of operations

Ray Tracing
Classic "Whitted−style"



The original ray tracing method involved tracing rays from the eye, through the image plane and into the scene. When a diffuse surface was hit by the ray, a ray was cast at the light source, and if it was visible the point was lit proportional to the cosine of the angle between the ray to the source and the surface normal. When a specular surface was hit, no light source test was made, instead a new ray was cast in the direction of specular reflection.

Ray Tracing
Classic

Diffuse surfaces shaded by orientation to light.

Specular surfaces look like mirrors.

Constant "ambient" added to avoid pure black.

Ray tracing was originally published by Whitted (Whitted81). Although conceptually simple, it was a computationally expensive method at the time, even for simple scenes. Despite its simplicity, the approach captured the important features that help make a scene look real −− particularly in scenes dominated by specular objects.

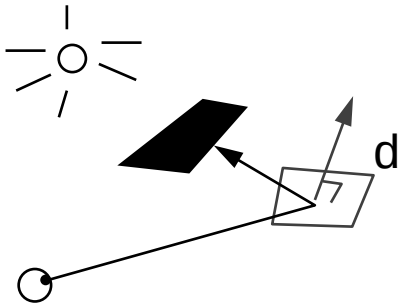
Ray Tracing Classic

$$L_o(\theta_o,\phi_o) = k_s L_{sp}(\theta_{sp},\phi_{sp}) +$$

$$k_d \cos\theta_{so} L_{e,so} + k_a L_a$$

This is the equation that classic ray tracing is effectively solving. The BRDF doesn't appear −− just coefficients that are related to reflectance −− ks and kd for specular and diffuse, and ka for "ambient." La is an ambient radiance that accounts for the effect of all interreflections, and is uniform throughout the scene. The coefficient ka allows the user to modify the effect of La surface by surface.

Ray Tracing Classic



shadows produced by testing the path of the ray to the light source

Shadows are produced in ray tracing by testing the visibility of the light source from a location on a diffuse surface.

Ray Tracing
Classic

PLUSES:
- shadows
- shiny objects
- arbitrary geometry types, anything that you can intersect with a ray

Ray Tracing
Classic

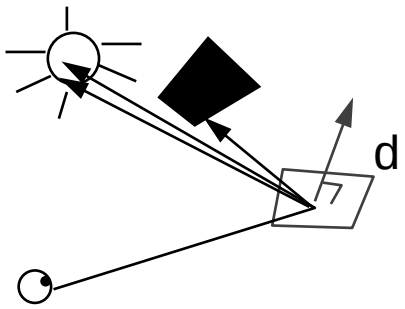
MINUSES:
- no $1/r^2$ fall−off of light
- diffuse interreflections an arbitrary constant
- sharp shadows, sharp specular reflections

Distribution Ray Tracing

INSIGHT: Generating a realistic image requires integration in many directions.

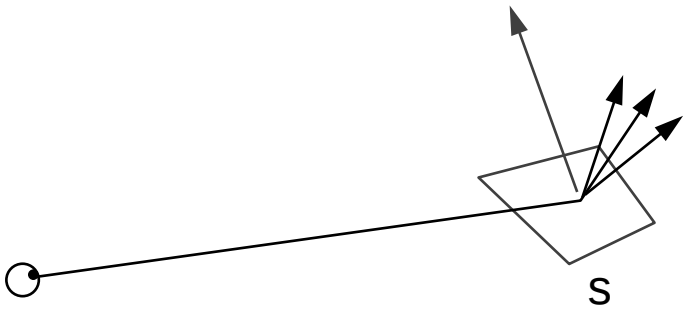
Distribution ray tracing was introduced by Cook et al in 1984. Originally it was called distributed ray tracing, but that name is now referred to ray tracing on parallel processors. The great contribution of distribution ray tracing is the recognition that integrals have to be performed over area, direction, and time to produce realistic images. A single ray for each surface isn't enough −− many rays have to be distributed over the area, or direction, or time, to compute the illumination effect.

Integrate across light source to get penumbra



To compute penumbra, rather than casting a single ray to the light source, an integral of the light from the source over the area of the source is needed. The integral is evaluated numerical by taking many sample points on the source and casting rays towards each of the points

Integrate over a cone around mirror angle to get fuzzy reflections.



To compute the effect of "fuzzy" specular reflection, and integral needs to be computed to collect the effect of light coming in from a cone of directions around the direction of specular reflection. In this case, many rays are cast within the cone of directions.

Distribution Ray Tracing

Also:

integrate across pixel to antialias

integrate across time to simulate motion blur

Integrating by taking many point samples can be used to solve many problems in rendering. "Jaggies" or stairstep edges can be "antialiased" by sampling many points on the area of a pixel and integrating the effect of all the surfaces visible through the sample points. If an animated scene is being rendered, the phenomenon of motion blure can be simulated by casting rays for into the scene for many points in time, and integrating the effect of all of the objects that are visible at different times.

Distribution Ray Tracing

$$L_o(\theta_o,\phi_o) =$$

$$(1/\Omega_{cone})\int k_s L_{sp}(\theta_{sp},\phi_{sp})\, d\omega +$$

$$k_d \int \cos\theta_{so}\cos\theta_{fs} L_{e,so}/r^2\, dA + k_a L_a$$

This is the approximation of the rendering equation being solved by distribution ray tracing. By properly formulating the integral over the area of the light source, the solid angle subtended by the light source is included in the approximation. Because the solid angle subtended decreases as the distance to the source r squared increases, this formulation will capture the familiar inverse square law of light propagation.

Integrals too complex for analytical solution.

General integration tool:

Monte Carlo

Many numerical quadrature methods could be used for performing the integrals in distribution ray tracing. The most general approach that can be used for evaluating all of the different integrals is Monte Carlo integration

Example: $y = \int_0^1 x^2\, dx$

Choose 4 random values of x between 0 and 1:
.981 .097 .503 .299

$y \approx (.981^2 + .097^2 + .503^2 + .299^2)/4$

$= .329$

Here is a simple example of using Monte Carlo to evaluate an integral. In global illumination deciding how to sample the independent variable (in this case x) is not as obvious, but the principle is the same. See the article in the appendix on Monte Carlo methods for more detail.

$(1/\Omega_{cone}) \int k_s L_{sp}(\theta_{sp},\phi_{sp})\, d\omega$

- Choose N directions in cone,
- Evaluate $k_s L_{sp}(\theta_{sp},\phi_{sp})$ for each
- Take the average of the evaluated terms

Here is an example of splitting up the integral for fuzzy specular reflection. The key problem is understanding how to sample the space of directions subtended by the solid angle in an unbiased manner.

Distribution Ray Tracing

PLUSES:

- fuzzy reflections
- fuzzy shadows
- $1/r^2$ fall−off

Distribution Ray Tracing

MINUS:

- still rely on arbitrary ambient term for the effects of many types of interreflections.

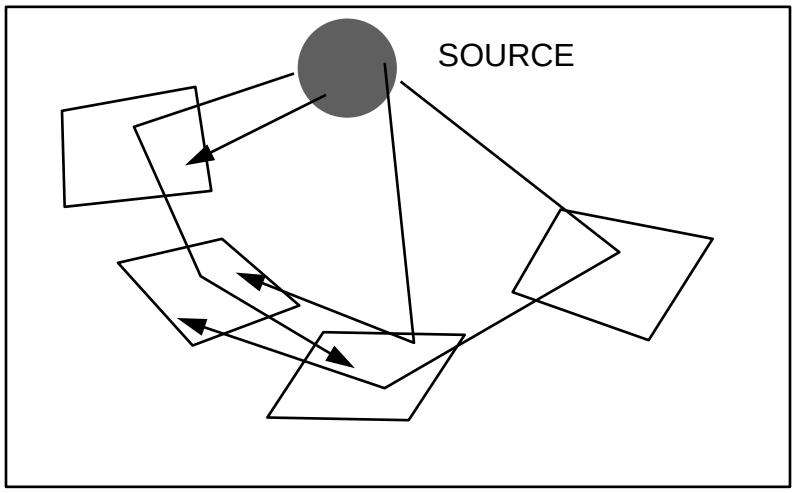
Monte Carlo Path Tracing

Insight: extend distribution ray tracing to account for all possible interreflections.

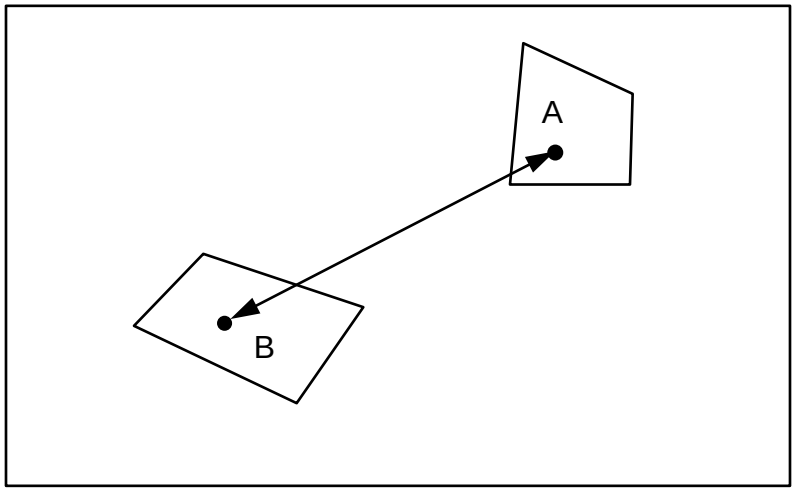
The rendering equation is an integral equation −− a Fredholm integral equation of the second kind. The idea of integration used in distribution ray tracing can be extended, but instead of tracing a lot of rays from each point to evaluate the integrals, a recursive approach is needed, because we want to estimate all radiances, rather than using the arbitrary La term. This idea was presented in Kajiya's 1986 paper on the rendering equation.

Monte Carlo Path Tracing

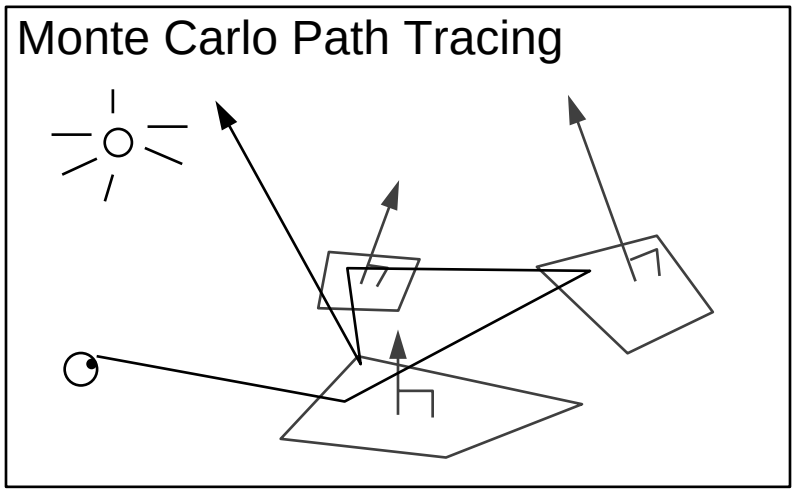
Naive approach: at each surface, follow a random direction to recursively estimate the incident light, until you happen to hit a light source.



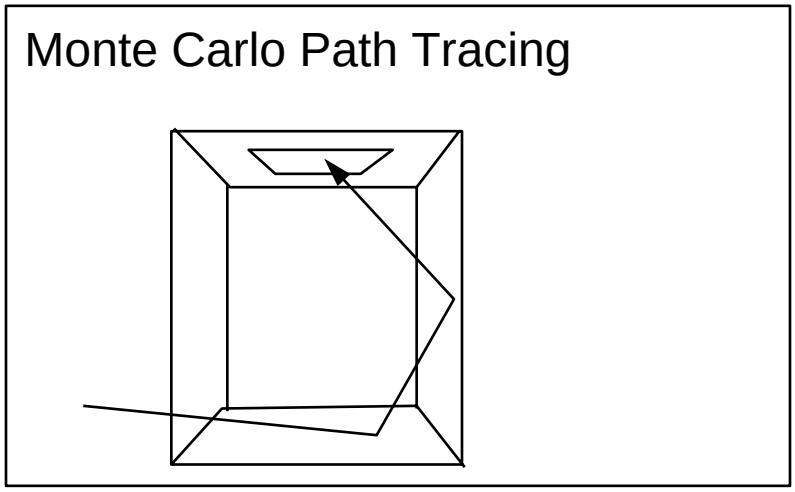
In other disciplines such as radiative heat transfer, Monte Carlo simulation is found to follow many light paths from a source to a receiver.



Because light can travel either way (if light can reach point B from point A, it can also reach point A starting at B), we can also find all the important paths by just reversing the direction (swapping source and receiver). In our problem the source is the light source, and the receiver is the eye.



In Monte Carlo Path tracing, paths of rays are found starting with the eye. When the first surface is hit, a random direction is chosen to estimate the incident radiance for that surface. When the next surface is hit, another random directionis chosen to estimate the incident radiance for that surface. This continues until a light source is hit for which the radiance is known, and no further integration is needed.



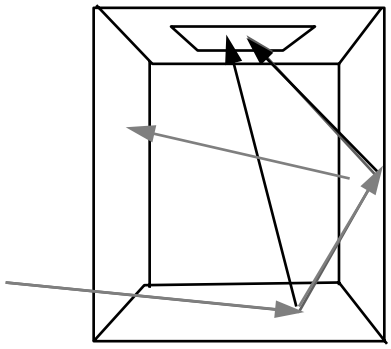
It may take a long time to reach a light source, or many rays may never hit a source. For enclosed spaces with large light sources though, naive Monte Carlo path tracing can produce an image of sorts after a reasonable wait...

Monte Carlo Path Tracing

Insight: At each point estimate TWO not just one integral: one over the light sources and one over everything else visible in the hemisphere.

To improve the convergence of the method, instead of just doing a reverse simulation, split the integral of incident illumination into two parts By evaluating the integral over light sources every time a surface is hit, non−zero values along the path are evaluated much more quickly.

Monte Carlo Path Tracing



By estimating the light source every time, the rays aren't just in a single path, but there are branches of the path to the light source from every surface hit along the way.

Monte Carlo Path Tracing

When does the path stop?

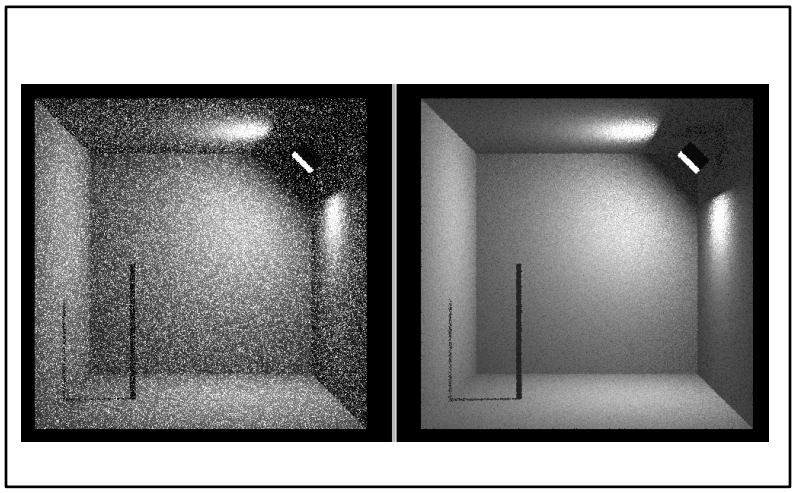
- arbitrary depth
- fixed threshhold
- Russian Roullette

Since light sources are hard to hit, some paths may go on indefinitely. However, for every surface in the path there is less light accumulated, since every surface absorrbs a little light. Paths can be terminated by limiting the depth, or ignoring paths that can contribute no more than a particular threshhold to the current result. An unbiased method stochastic method for terminating the paths is known as Russian Roullette.

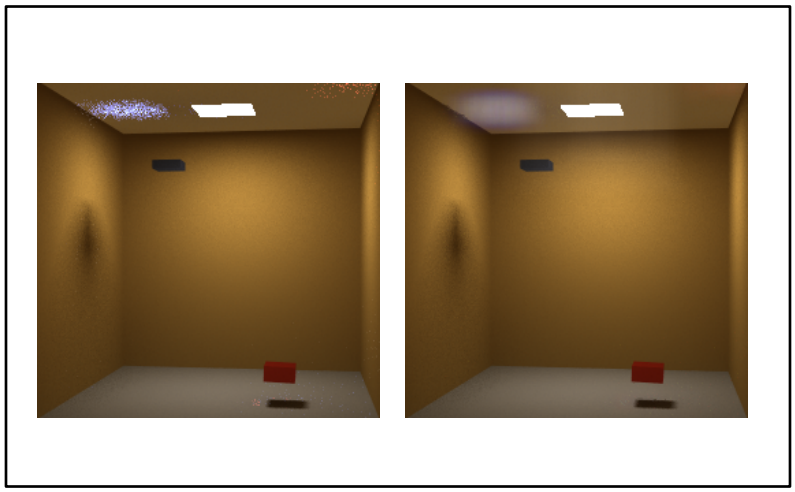
Monte Carlo Path Tracing

PLUS: Complete Solution to the Rendering Equation

MINUS: Noise, unless you have A LOT of samples.



Here are a couple of examples of the noise you get in a Monte Carlo solution. These are computed with the shot at the light source (the noise would be worse in a naive Monte Carlo solution.)The noise level in a Monte Carlo solution decreases with the square root of the number of samples taken. To reduce the noise by a factor of 2 (reducing it by any smaller factor will hardly be noticeable to your eye), 4 times as many samples are needed.



A common question is, why not just filter out the noise? The problem is differentiated the noise and the signal. Just blurring the image to get rid of the bright spots also blurs all of the real features in the image. In some cases, after a lot of sampling it may be able to detect isolated areas of noise and filter them. This is from Rushmeier and Ward 1994, and shows a caustic cast by a small blue specular surface onto the ceiling being filtered

Backwards Ray Tracing

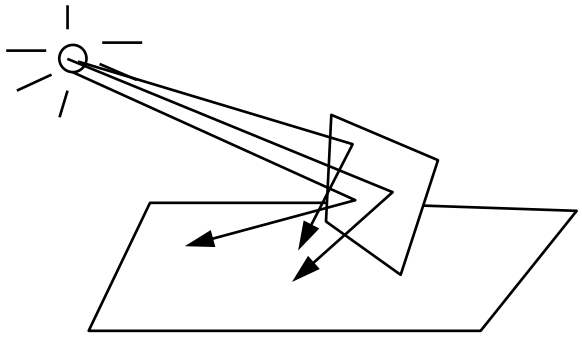
for finding caustics

aim from source to specular surfaces

collect samples on diffuse surface

Caustics are a feature that are difficult to capture. With Monte Carlo path tracing, the paths are hard to find, so take a large number of samples to hit.. An alternative approach, originally proposed by Arvo, is to trace the rays from the light source. Actually, this should be "forward" ray tracing, since it is how light naturally travels. In graphics though we normally start at the eye, so this is "backwards" compared to other ray tracing techniques.

Backwards Ray Tracing



In backwards ray tracing packets of light are deposited onto diffuse surfaces showing where caustic paths end. To make a picture at the end, some method of smoothing these point samples has to be used to make a smooth bright spot.

The RADIANCE Algorithm

Modified Monte Carlo−type path tracing −
− reorganizes sampling
− reuses samples

Included in comprehensive software package

A freely available ray tracing system is Radiance, written by Greg Ward Larson. It is a comprehensive lighting package that allows accurate , physical definition of the scene, and incorporates the common graphics primitives such as meshes, polygons, bump maps, texture maps. It also has many postprocessing features such as interpolation for image based rendering, and filters for tone mapping. It can be downloaded freely from the net, a course is being presented at SIGGRAPH 98 in using the software, and a book (with CD−ROM) is a available to learn about the package.

PUBLICATIONS REFERRED TO IN RAY TRACING:

Andrew Glassner, "Space subdivision for fast ray tracing" IEEE Computer Graphics and Applications, 4(10):15−22, October 1984.

Turner Whitted, "An improved illumination model for shaded display" Communications of the ACM, 23(6):343−349, June 1980.

Robert L. Cook and Thomas Porter and Loren Carpenter, "Distributed ray tracing", Computer Graphics (ACM SIGGRAPH ’84 Proceedings) 18(3): 137−−145.

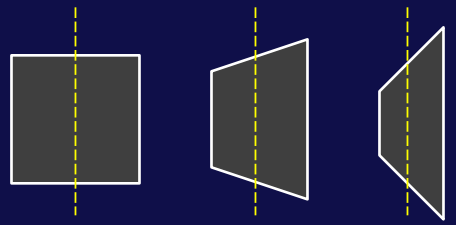
James T. Kajiya, "The rendering equation," Computer Graphics (ACM SIGGRAPH ’86 Proceedings), 20(4)143−150.

Holly Rushmeier and Greg Ward "Energy Preserving Non−Linear Filters" Computer Graphics (ACM SIGGRAPH ’94 Proceedings) pages 131−138.

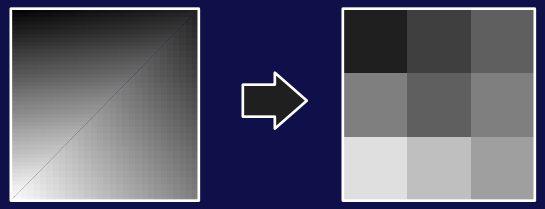
Radiosity

view independent solutions
for diffuse scenes

Diffuse surfaces have one radiance (color) for all viewing directions



Finite approximation



Rendering equation

$$L_r(\theta_r,\phi_r) = \int f_r(\theta_i,\phi_i,\theta_r,\phi_r)\, L_i(\theta_i,\phi_i) \cos\theta_i \, d\omega_i$$

Diffuse rendering equation

$$L_r = \int (\rho_d/\pi)\, L_i(\theta_i,\phi_i) \cos\theta_i \, d\omega_i$$

Diffuse rendering equation

$$L_r = \int (\rho_d/\pi)\, L_i(\theta_i,\phi_i) \cos\theta_i \, d\omega_i$$

L_r is just ρ_d times the average radiance indicent at a point.

Diffuse rendering equation

$$L_r = \int (\rho_d/\pi)\, L_i(\theta_i,\phi_i) \cos\theta_i \, d\omega_i$$

Suppose we assume that we break the environment up into N patches and that each patch has a single reflectance and radiance R_j and L_j.

Diffuse rendering equation

$$L_r = \int (\rho_d/\pi)\, L_i(\theta_i,\phi_i) \cos\theta_i \, d\omega_i$$

Diffuse rendering equation-- finite approximation

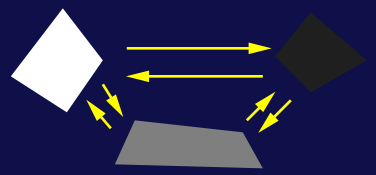
$$L_j = \int (R_j/\pi)\, L_k \cos\theta_{jk}\, (g_{ik} dA_k \cos\theta_{kj}/D_{jk}^2)$$

Diffuse rendering equation-- finite approximation

$$L_j = \int (R_j/\pi)\, L_k \cos\theta_{jk}\, (g_{ik} dA_k \cos\theta_{kj}/D_{jk}^2)$$

Diffuse rendering equation-- finite approximation

$$L_j = \sum_{k=1}^{N} R_j\, c_{jk}\, L_k$$



Diffuse rendering equation-- finite approximation

$$L_1 = E_1 + R_1 c_{11} L_1 + R_1 c_{21} L_2 + R_1 c_{31} L_3$$
$$L_2 = E_2 + R_2 c_{12} L_1 + R_2 c_{22} L_2 + R_2 c_{32} L_3$$
$$L_3 = E_3 + R_3 c_{13} L_1 + R_3 c_{23} L_2 + R_3 c_{33} L_3$$

$$\begin{bmatrix} 1-R_1c_{11} & -R_1c_{21} & -R_1c_{31} \\ R_2c_{12} & 1-R_2c_{22} & -R_2c_{32} \\ R_3c_{13} & -R_3c_{21} & 1-R_3c_{33} \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \begin{bmatrix} E_1 \\ E_3 \\ E_3 \end{bmatrix}$$

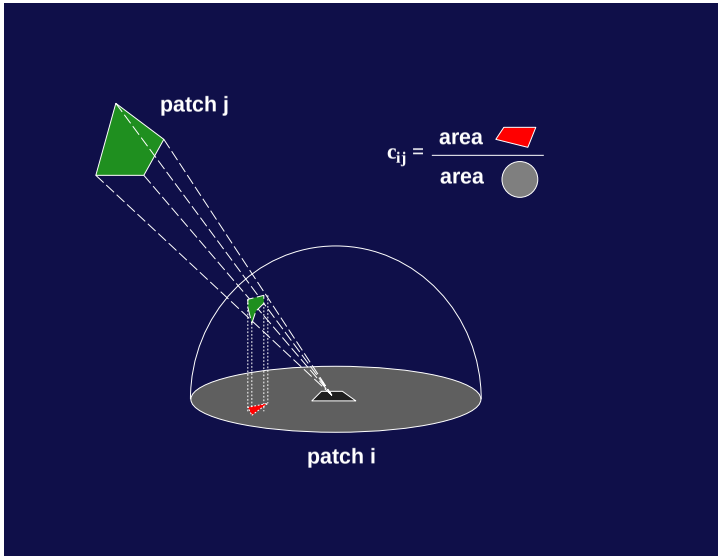
$$\begin{bmatrix} 1-R_1c_{11} & -R_1c_{21} & -R_1c_{31} \\ R_2c_{12} & 1-R_2c_{22} & -R_2c_{32} \\ R_3c_{13} & -R_3c_{21} & 1-R_3c_{33} \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \begin{bmatrix} E_1 \\ E_3 \\ E_3 \end{bmatrix}$$

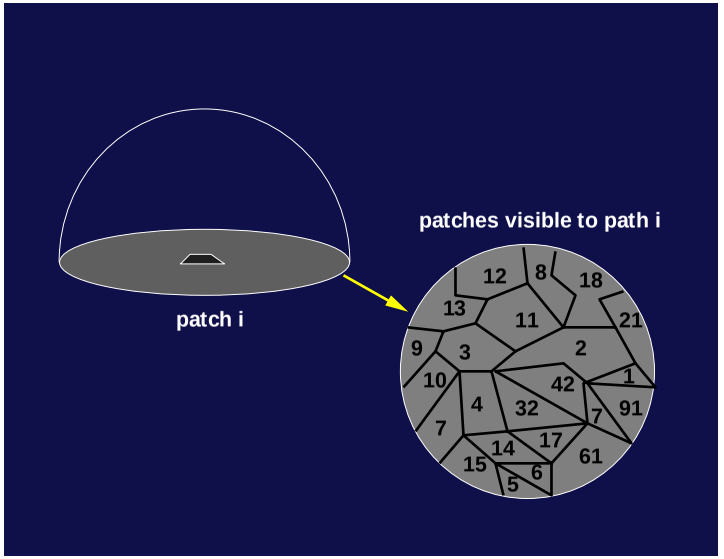
c_{ii} **is zero--**
a polygon does not illuminate itself.

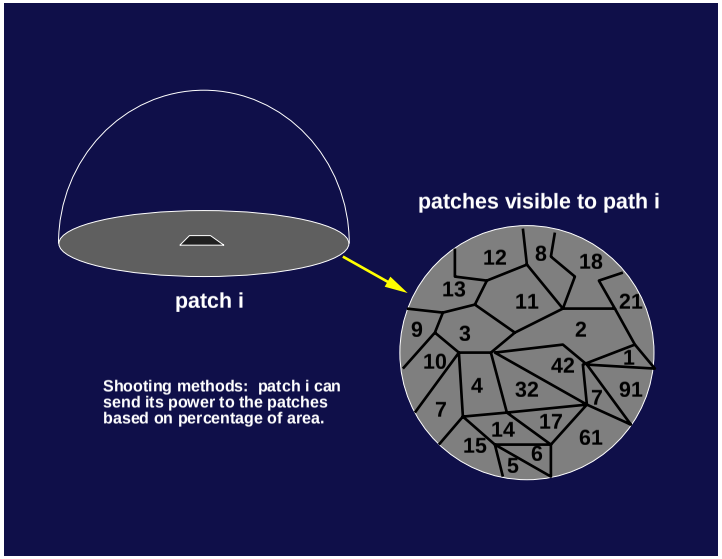
$$\begin{bmatrix} 1 & -R_1c_{21} & -R_1c_{31} \\ -R_2c_{12} & 1 & -R_2c_{32} \\ -R_3c_{13} & -R_{13}c_{23} & 1 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix}$$

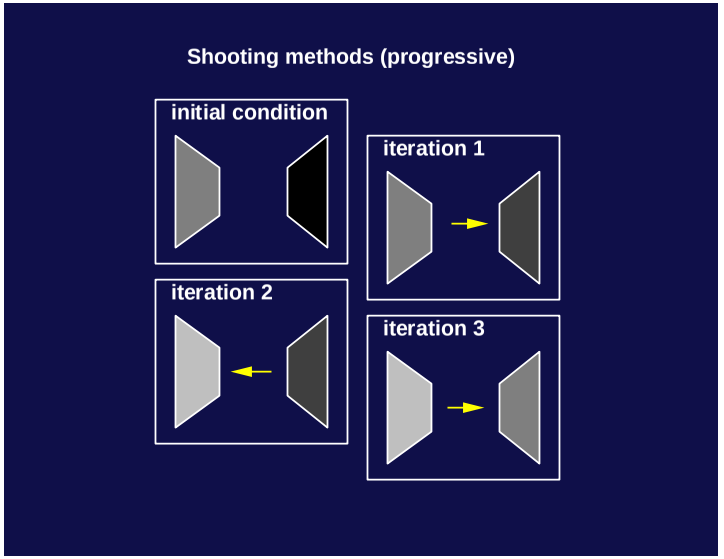
Compute all c_{jk} and then solve system....

BUT $O(N^2)$ storage and $O(N^3)$ computation time.

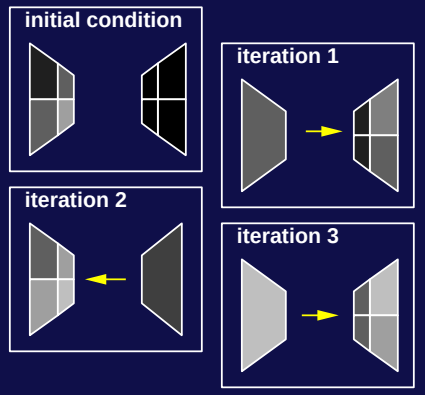

patch j
c_{ij} = area / area
patch i


patches visible to path i
patch i
12
8
18
13
11
21
9
3
2
10
42
1
4
32
7
91
7
14
17
61
15
6
5

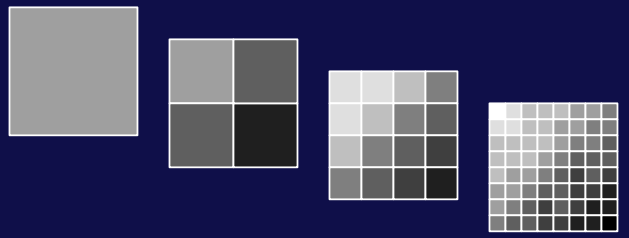

patches visible to path i
patch i
Shooting methods: patch i can send its power to the patches based on percentage of area.
12
8
18
13
11
21
9
3
2
10
42
1
4
32
7
91
7
14
17
61
15
6
5


Shooting methods (progressive)
initial condition
iteration 1
iteration 2
iteration 3

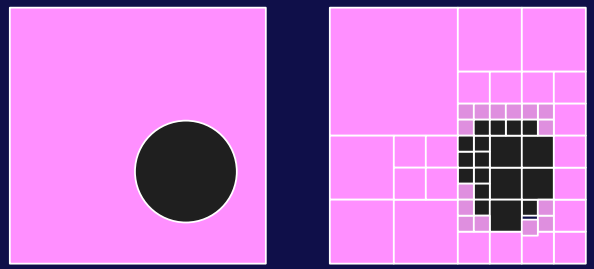
Hierarchical methods



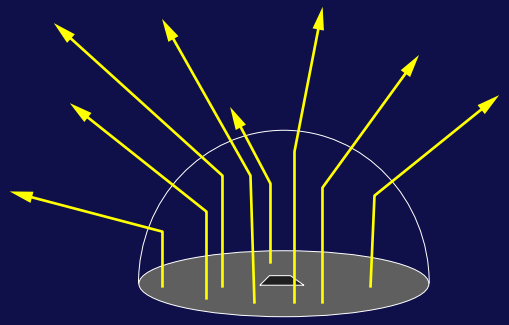
For hierarchical methods, each path is multi−resolution



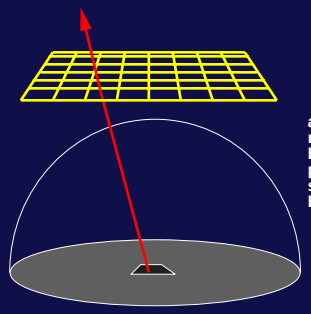
Usually an adaptive hierarchy is used



c_{ij} = fraction of rays that hit patch j



Z−buffers can work too....



area distortion must be corrected by a weight for each pixel. And five screens should be used.

patch i

Displaying solutions

real constant linear

Illumination textures

tiled reflectance texture map

illumination texture map

composite

Where to go from here

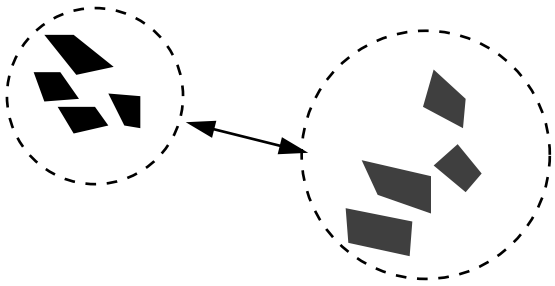
- Examine how solutions scale
- Non−polygonal objects
- Optimization of display meshes
- Memory is usually the bottleneck

Current Trends

A great deal of work has been done beyond the basics covered in this course and continues to go on. A good way to track the progress in global illumination research is to browse through the Proceedings of the Eurographics Rendering Workshop that has been held annually since 1990. The Proceedings are published as a volume by Springer−Verlag.
In this section we look at some of the areas that are still open problems, or that are new approaches that are being considered

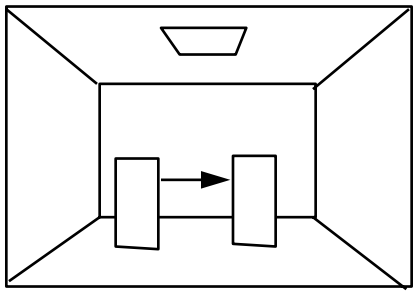
---

Radiosity Methods: Clustering



For complex environments, it is not practical to consider solving the simultaneous equations for individual surface to surface or object to object interchange. The transfer of light between clusters of objects is a way to deal with this. Smits et al. presented a clustering technique at SIGGRAPH 1994 for diffuse surfaces using different levels of links, while Sillion presented a method at the Eurographics Rendering Workshop that year that modelled clusters of objects as voluemes of particpating media. Recently Christenson et al. presented a clustering method for glossing surfaces in ACM Transactions on Graphics, 1997.

Radiosity: Real Time Object Motion

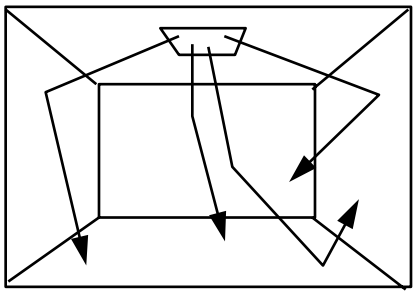


Radiosity solutions can be displayed in real time as long as the viewer only moves. If objects move, and new radiosity solution is needed. Since Baum et al.s 1986 Visual Computer paper on exploiting temporal coherence, many techniques have been proposed. Most recently Forsyth et al. (ERW94), Shaw (Computer Graphics Forum 1996), and Drettakis and Sillion (SIGGRAPH 97) have developed efficient techniques for updating hierarchical radiosity solutions for animated scenes.

Radiosity: Meshing Techniques

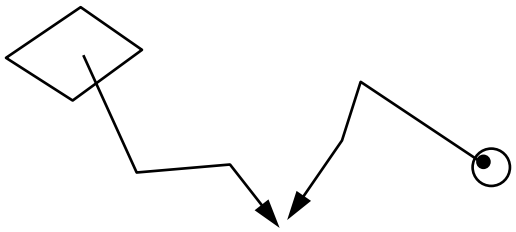
Work continues to make discontinuity meshing more efficient. The problem is to adequately discretize to capture visible discontinuities, but not to discretize for discontinuities that will not ultimately be visible in the final image.

Particle Tracing



The basic idea of particle tracing is to do a traditional forward radiative transfer from the light source. Like backwards ray tracing, some sort of smoothing, or more properly signal reconstruction, has to be performed to convert the deposited particles into a coherent image. Pattanaik presented a particle tracing technique in his 1993 dissertation, and recently a highly refined version of the method has been presented by Walter et al. in ACM Transactions on Graphics, 1997 An alternative approach called "photon maps" was developed by Jensen to use particle tracing for part of the illumination solution, and was presented at the ERW in 1996. A paper on using photon maps for volumes is being presented here at SIGGRAPH 98.

Bidirectional Ray Tracing



It was clear for many years, that sometimes it is efficient to trace from the light, and sometimes from the eye. The problem was to formalize this idea into robust methods Robust bidirectional path methods were independently developed by Veach and Guibas (SIGGRAPH 95) and LaForturne and Willems (Computer Graphics Forum, 1994). Eric Veach has developed a method beyond this to more efficiently find important light paths using a variation of simulated annealing known as the Metropolis Algorithm (SIGGRAPH 97).

Tone Mapping

mapping the dynamic range

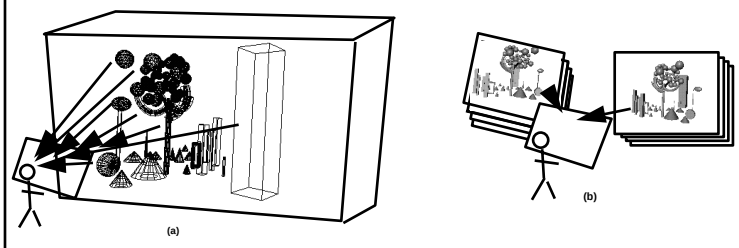
100000:1 −−−> 30:1

illumination solution

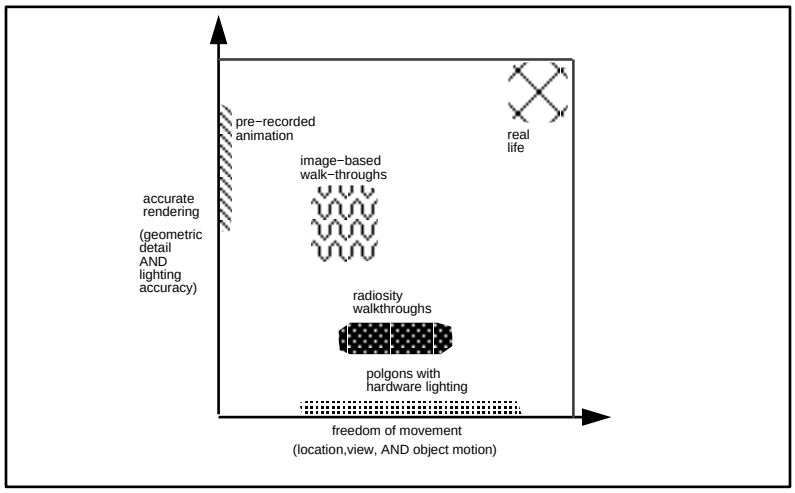
CRT

Global illumination techniques produce images with a much higher dynamic range that can be displayed by CRT's or prints. Mappings for going from one range to another are discussed in more detail in the Appendix from Solution to Image. Understanding what the mapping will be though can be used to reduce the calculations in the illumination solution. An example of this is presented by Gibson and Hubbold, in Computer Graphics Forum 1997.

Image Based Rendering



The idea of image based rendering is to make images from images, rather than from explicit geometric descriptions.



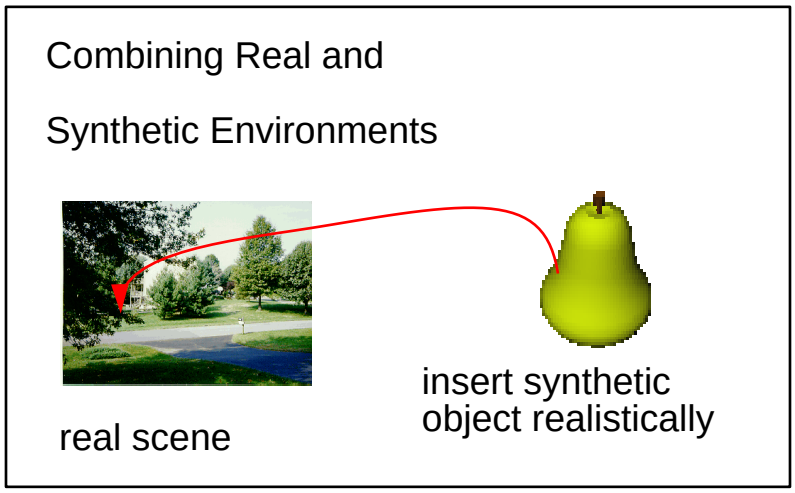
There are a number of trade−offs between polygonal representations, and image based representations.

Types of Image Based Representations

Range Images: store radiance and distance at each pixel

Light Field/ Lumigraph: no geometry, just radiances as a function of direction

There are different types of possible image based representations of a scene. The problem for global illumination research is, what is the most efficient method to use for to generate a particular representation. This question is discussed for range images in Nimeroff et al. IEEE TVCG 1996.



Perhaps the greatest application area for global illumination methods will be placing synthetic objects into representations of real scenes. Illumination has to be consistent between the real and the synthetic for the result to be believable. In the past, the adjust ment to make the combinations look right have been done in an ad hoc manner. Early work was done by Fournier et al (Graphics Interface 1993) combining video captured images and radiosity solutions. With computer graphics and computer vision becoming closer in the area of graphics input acquisition, and image based rendering, a lot of progress will be made in this area in the next couple of years.

PROCEEDINGS OF THE EUROGRAPHICS RENDERING WORKSHOP:

Bouatouch, K. and C. Bouville, Eds. 1992. Photorealism in Computer Graphics. Berlin, Germany: Springer−Verlag. (Proceedings of 1990 Workshop)

Brunet, P. and F. W. Jansen. Eds. 1994. Photorealistic Rendering in Computer Graphics. Berlin, Germany: Springer−Verlag. (Proceedings of 1991 Workshop)

Cohen, M., C. Puech, and F. Sillion. Eds., 1993. Fourth EUROGRAPHICS Workshop on Rendering, Eurographics Technical Report Series EG 93 RW. Aire−la−Ville, Switzerland: Eurographics Association. ISSN 1017−4656.

Sakas, Shirley, Müller, Eds., Photorealistic Rendering Techniques Springer−Verlag, Berlin, Germany. (Proceedings of 1994 Workshop.)

Hanrahan and Purgathofer, Eds., Rendering Techniques’95, Springer Wien. 1995.

Pueyo and Schroeder, Eds., Rendering Techniques’96 Springer Wien1996.

Dorsey and Slusallek, Eds., Rendering Techniques’97 Springer Wien,1997.

INDIVIDUAL ARTICLES REFERRED TO IN CURRENT TRENDS:

Brian Smits and James Arvo and Donald Greenberg, A clustering algorithm for radiosity in complex} environments Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings), 435−442.

Francois Sillion, Clustering and volume scattering for hierarchical radiosity calculations, Fifth Eurographics Workshop on Rendering, 105−117.

Per H. Christensen and Dani Lischinski and Eric J. Stollnitz and David H. Salesin, Clustering for glossy global illumination, ACM Transactions on Graphics 16(1) 3−−33.

Daniel R. Baum and John R. Wallace and Michael F. Cohen and Donald P. Greenberg, The back−buffer algorithm: an extension of the radiosity method to dynamic environments, The Visual Computer, 2 (5), 298−306.\

David A. Forsyth and Chien Yang and Kim Teo, Efficient radiosity in dynamic environments, Fifth Eurographics Workshop on Rendering, 313−323.

Erin Shaw, Hierarchical radiosity for dynamic environments, Computer Graphics Forum, 16 (2)107−118.

George Drettakis and Francios Sillion Interactive update of global illumination using a line−space hierarchy Proceedings of SIGGRAPH 97, 57−64.

Sumanta N. Pattanaik, Computational methods for global illumination and visualisation of complex 3D environments, Birla Institute of Technology & Science, Computer Science Department, Pilani, India, Ph.D. thesis

Bruce Walter and Philip M. Hubbard and Peter Shirley and Donald P Greenberg. Global illumination using local linear density estimation, ACM Transactions on Graphics 16(3) 217−259.

Henrik Wann Jensen, Global illumination using photon maps, Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering), 21−30.

Eric Veach and Leonidas Guibas, Optimally combining sampling techniques for Monte Carlo rendering SIGGRAPH 95, pp. 419−428.

Eric P. Lafortune and Yves D. Willems, Bi−directional path tracing , Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)}, 145−153.

Eric Veach and Leonidas Guibas Metropolis light transport, Proceedings of SIGGRAPH 97, pp. 65−76.

Simon Gibson and R. J. Hubbold, Perceptually driven radiosity, Computer Graphics Forum 16(2), 119−128/

Jeffry Nimeroff and Julie Dorsey and Holly Rushmeier, Implementation and analysis of an image−based global illumination framework for animated environments, IEEE Transactions on Visualization and Computer Graphics} 2(4) 283−298.

Alain Fournier and Atjeng S. Gunawan and Chris Romanzin, Common illumination between real and computer generated scenes, Proceedings of Graphics Interface '93,254−262.

Developing the Rendering Equations

Kurt Zimmerman
Indiana University

As stated, physically based rendering simulates the movement of light throughout an environment. It is important that we understand the units involved in measuring light. As we will see, it is sometimes useful to use different units depending on the application. This also provides us with mathematical framework for describing the rendering process.

We will assume geometric optics in our measurements. This means that we will use the particle theory of light. We can get away with this because most visual phenomenon can be modeled with this assumption in place, diffraction and interference being the notable exceptions. We will also assume that the speed of light is infinite, which implies that any simulation is in a steady state. This is usually appropriate since the time it takes light to travel in common scenes is not perceivable.

The following sections touch briefly on several important concepts, which are handled in much detail by Glassner [3].

1 Solid Angles

Key concepts in the radiometric definitions are the ideas of solid angle and projection. When we think of a solid angle we usually think of some object projected onto a unit sphere. This projection is the solid angle of the object as view from the center of the sphere (Figure 1). The units for solid angles are steradians, sr, which are actually unitless but are usually left in for clarity.

The relationship between a differential area on a sphere and the corresponding differential solid angle can be described in the following way: A differential area, dA, on a unit sphere is equal to its solid angle, $d\hat{\omega}$. If dA is on a non-unit sphere, then the difference between the two is an r^2 term where r is the radius of a sphere. In Figure 2 describes this in detail. Here we see two hemispheres. The inside hemisphere has $r = 1$. Since dA has a horizontal side of length $r\ \sin\theta\ d\phi$ and a vertical side of length $r\ d\theta$ the differential area is:

$$dA = r^2 \sin\theta\ d\theta\ d\phi \tag{1}$$

and the differential solid angle is: $d\hat{\omega} = \sin\theta\ d\theta\ d\phi$

2 Projections

The relationship between the area of surface element dA and the projection of that surface onto a plane is:

$$\text{proj}_A = \cos\theta\ dA\ , \tag{2}$$

as shown in Figure 3.

Finally, we can consider a differential area dA' which does not lie on a great sphere. Projecting this onto a sphere is equivalent to projecting it onto a plane which is perpendicular to the ray running

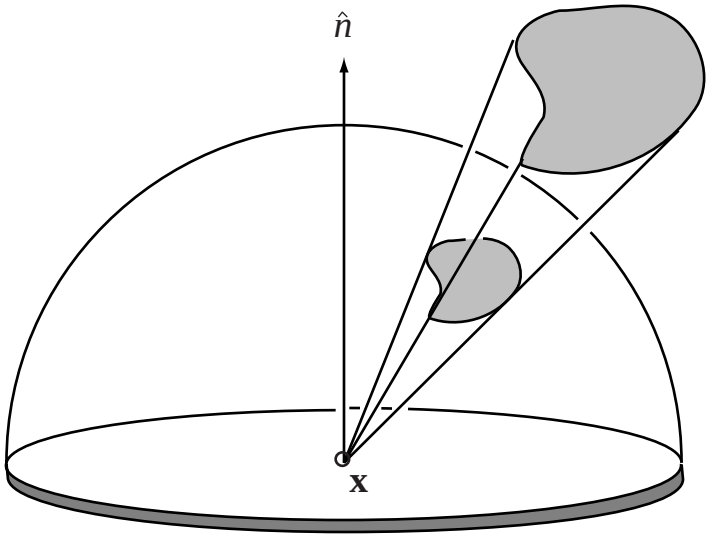


Figure 1: Solid Angle of an object viewed from **x**

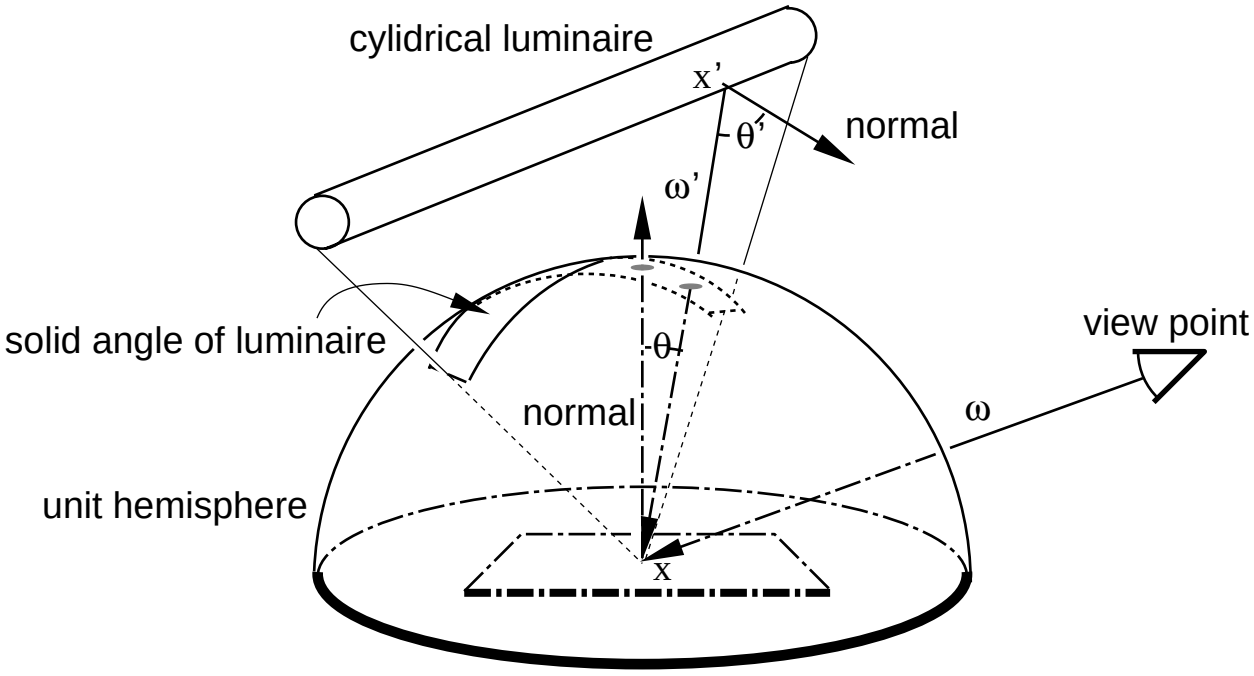


Figure 2: Relationship between area and solid angle on a sphere

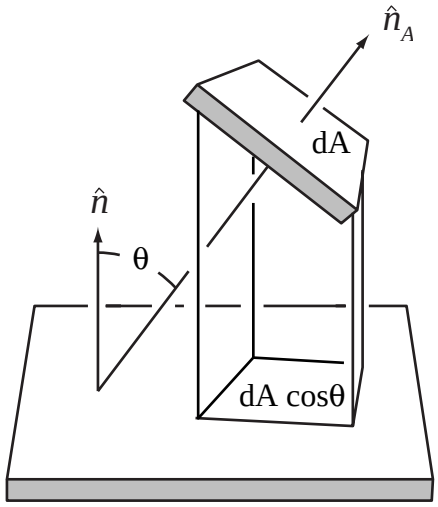


Figure 3: Projection of a surface element onto a plane

from the center of the sphere to the center of dA'. Thus from Equations 1, 3 and 2 we get the relationship between a differential solid angle $d\hat{\omega}'$ and an arbitrarily oriented differential area dA':

$$d\hat{\omega}' = \frac{dA' \cos\theta'}{||\mathbf{x}' - \mathbf{x}||^2} , \tag{3}$$

where $\mathbf{x}$ is the sphere center and $\mathbf{x}'$ is the center of dA'.

3 Radiometry

In general, physically based computer graphics algorithms do not chase light particles or photons around the environment. Usually the computational quantity of flow that is measured throughout an environment is *radiant flux* or *radiant power* which is generally denoted by the Greek letter Φ and measured in Watts. Radiant power has no meaning at a particular point in an environment, therefore we need different quantities to represent the interaction of radiant power and surfaces. The most important of these quantities is *radiance*.

4 Radiance

Radiance is a fundamental quantity usually associated with a light ray. The radiance leaving or arriving at a given point, $\mathbf{x}$, traveling in a given direction, $\hat{\omega}$, can be defined as the power per unit projected area perpendicular to the ray per unit solid angle in the direction of the ray. Following notation similar to the IES[1] standard we have:

$$L(\mathbf{x}, \hat{\omega}) = \frac{d^2\Phi(\mathbf{x}, \hat{\omega})}{dA \ \cos\theta \ d\hat{\omega}} , \tag{4}$$

[1]The Illumination Engineering Society or IES notation is the standard for illumination engineering. Notation and definitions can be found in the ANSI/IES report [5].

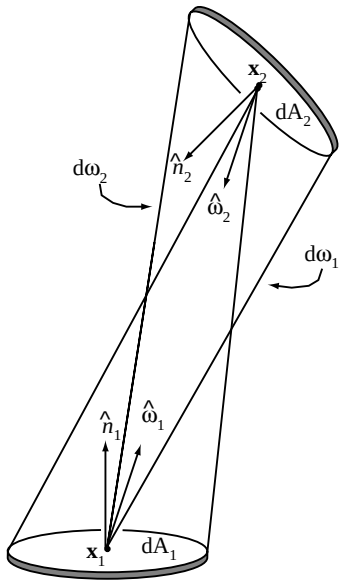


Figure 4: Radiance between differential surfaces.

where Φ is power, dA is the differential area surrounding $\mathbf{x}$, θ is the angle between the ray and the surface normal at $\mathbf{x}$, and $d\hat{\omega}$ is the differential solid angle in the direction of the ray.[2]

Radiance is a convenient quantity to associate with a light ray because it remains constant as it propagates along a direction (assuming a vacuum). To see that this is true we need to look closely at the definitions. We can reorganize the above definition in terms of radiant flux:

$$d\Phi(\mathbf{x}, \hat{\omega}) = L(\mathbf{x}, \hat{\omega}) \cos\theta d\hat{\omega} dA \ . \tag{5}$$

Using the geometry of Figure 4 and assuming a vacuum, the law of conservation of energy says that the flux leaving surface one in the direction of surface two, must arrive at surface two, more concisely:

$$d\Phi(\mathbf{x}_1, \hat{\omega}_1) = d\Phi(\mathbf{x}_2, \hat{\omega}_2) \ .$$

Thus

$$L(\mathbf{x}_1, \hat{\omega}_1) \cos\theta_1 d\hat{\omega}_1 dA_1 = L(\mathbf{x}_2, \hat{\omega}_2) \cos\theta_2 d\hat{\omega}_2 dA_2 \ . \tag{6}$$

From the previous definitions we see that $d\hat{\omega}_1 = (dA_2 \cos\theta_2)/r^2$ and $d\hat{\omega}_2 = (dA_1 \cos\theta_1)/r^2$ where $r^2 = \mathbf{x}_1 - \mathbf{x}_2^2$, $\theta_1 = (\hat{n}_1 \cdot \hat{\omega}_1)$ and $\theta_2 = (\hat{n}_2 \cdot \hat{\omega}_2)$. Dividing each side of Equation 6 by $dA_1(\cos\hat{\omega}_1 \ dA_2 \cos\hat{\omega}_2)/r^2$ we see that $L(\mathbf{x}_1, \hat{\omega}_1) = L(\mathbf{x}_2, \hat{\omega}_2)$. Notice that the definition of radiance lends itself to some confusion about the direction of flow. For this reason Arvo [1] uses the term *surface radiance*, $L_s(\mathbf{x}, \hat{\omega})$, to refer to light leaving $\mathbf{x}$ in direction $\hat{\omega}$ and *field radiance*, $L_f(\mathbf{x}, \hat{\omega})$, to refer to light arriving at $\mathbf{x}$ from direction $\hat{\omega}$.

Radiance is considered a fundamental quantity not only because it is convenient but because all other radiometric and photometric quantities can be derived from it as can be seen in the appendix.

[2]Note that Equation 4 should be written as a second order partial derivative in the form $\frac{\partial^2 \Phi}{\partial A \ \cos\theta \ \partial\hat{\omega}}$, but we will stick with convention.

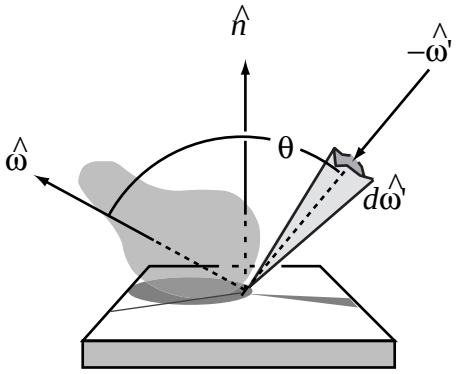


Figure 5: Geometry for BRDF.

5 BRDF and BTDF

Now that we have radiance to characterize the flow of light traveling between two surfaces a function is needed to describe the reflection of light off a surface. We would expect that the reflection of light off a surface is proportional to the light arriving at the surface. The function that describes this proportionality is the *bidirectional reflectance distribution function* or BRDF, Figure 5

$$f_r(\mathbf{x}, \hat{\omega}', \hat{\omega}) = \frac{dL_r(\mathbf{x}, \hat{\omega})}{L_f(\mathbf{x}, \hat{\omega}') \cos\theta d\hat{\omega}'} , \tag{7}$$

where L_f is the field radiance and L_r is the reflected radiance. Note that L_r is used instead of the surface radiance L_s. The reason for this distinction will become clear in the next section. Note also that the denominator of Equation 7 is irradiance as described in the appendix. A *physically plausible* BRDF maintains two important properties:

1. The BRDF must follow the *Helmholtz reciprocity principle*. This states that the BRDF will be the same if the incident and reflected light is reversed. Stated,

$$f_r(\mathbf{x}, \hat{\omega}', \hat{\omega}) = f_r(\mathbf{x}, \hat{\omega}, \hat{\omega}') \tag{8}$$

2. The BRDF must uphold the law of conservation of energy. Therefore the outgoing radiance must be less than or equal to the incoming radiance. If the BRDF is integrated over the hemisphere of reflected directions we will get the total reflectance for an incoming direction $\hat{\omega}'$. This value must be less than or equal to one:

$$R(\mathbf{x}, \hat{\omega}') = \int_\Omega f_r(\mathbf{x}, \hat{\omega}', \hat{\omega}) \cos\theta d\hat{\omega}' \leq 1.0 . \tag{9}$$

Several models for BRDF are described in Glassner [3] including the most commonly used models of Lambert and Phong, as well as more complicated models employing Fresnel equations and the empirical models of Ward [11]. An additional model which is not covered by Glassner but deserves mention is the modified Phong model of Lafortune and Willems [7]. Lafortune and Willems modify the Phong model so that it obeys the Helmholtz reciprocity principle. As pointed out by Shirley [10] it is difficult to tell whether or not it is necessary to have a physically plausible BRDF in order to produce realistic images.

For some surfaces that transmit light, the BRDF must be combined with the *bidirectional transmission distribution function*, BTDF. This allows us to render images of glass, lamp shades and ultra-thin metals.

6 The Rendering Equation

Previously, radiance was defined as means of expressing the light traveling between two surface. In the previous section, the BRDF was defined as the interaction of light with a surface. These two ideas can be combined to form an equation that describes the flow of light throughout an environment. Notice that by rewriting Equation 7 we get the following:

$$dL_r(\mathbf{x}, \hat{\omega}) = f_r(\mathbf{x}, \hat{\omega}', \hat{\omega}) L_f(\mathbf{x}, \hat{\omega}') \cos\theta d\hat{\omega}'$$

This is the reflected radiance in terms of the incoming radiance from one ray and the BRDF. The total reflected radiance at a point, $\mathbf{x}$, in direction, $\hat{\omega}$, combine with any emitted radiance, L_e, to form surface radiance, L_s:

$$L_s(\mathbf{x}, \hat{\omega}) = L_e(\mathbf{x}, \hat{\omega}) + \int_{\Omega_i} f_r(\mathbf{x}, \hat{\omega}', \hat{\omega}) L_f(\mathbf{x}, \hat{\omega}') \cos\theta d\hat{\omega}' \ , \tag{10}$$

where $\cos\theta = (\hat{n} \cdot -\hat{\omega}')$. This is the *rendering equation* in terms of directions as first introduced by Immel et al.[4]. Sometimes it is more convenient to express Equation 10 in terms of surfaces. We can do this by using the definition from Equation 3 to get:

$$L_s(\mathbf{x}, \hat{\omega}) = L_e(\mathbf{x}, \hat{\omega}) + \int_A g(\mathbf{x}, \mathbf{x}') f_r(\mathbf{x}, \hat{\omega}, \hat{\omega}') L_f(\mathbf{x}, \hat{\omega}') \frac{\cos\theta \cos\theta' dA}{||\mathbf{x}' - \mathbf{x}||^2} \ , \tag{11}$$

where $||\mathbf{x}' - \mathbf{x}||$ is the distance from $\mathbf{x}$ to $\mathbf{x}'$, $\cos\theta' = (\hat{n}' \cdot \hat{\omega}')$, and

$$g(\mathbf{x}, \mathbf{x}') = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is visible to } \mathbf{x}' \\ 0 & \text{otherwise .} \end{cases}$$

This geometry term is necessary since some surfaces might be blocked. Equation 11 is the form similar to that of Kajiya's landmark paper[6]. The geometry for the rendering equation can be seen in Figure 6.

We must keep in mind that $L_f(\mathbf{x}, \hat{\omega}') = L_s(\mathbf{x}', \hat{\omega}')$ in Equations 11 and 10 . By replacing L_f with L_s we see that Equations 11 and 10 are integral equations.

A Appendix: Radiometry and Photometry

This appendix was written in an attempt to clarify the relationship between radiometry and photometry. This clarification was necessary because our ray tracer associates a value of radiance with each ray traced. However, the illumination engineering community specifies luminaires with photometric values.

In order to use the value associated with a luminaire sample, we had to transform it into spectral radiance. It should be noted that in the literature the term *radiance* usually implies *spectral* radiance, averaged over a band of wavelengths (such as the red, green, or blue portions of the visible spectrum).

The first step was to understand the radiometric and photometric terminology according to ANSI/IES (1986)[5].

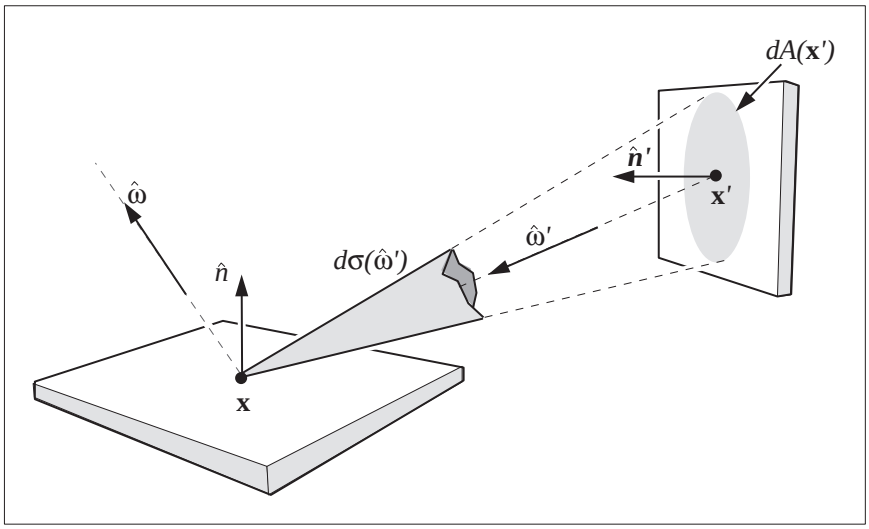


Figure 6: Geometry for the rendering equation

A.1 Important Radiometric Terms

1. **Radiant energy,** Q**.** Energy traveling in electro-magnetic waves, measured in joules.

   (a) **Spectral radiant energy,** $Q_\lambda = dQ/d\lambda$**,** measured in joules per nanometer, $joules/nm$.

2. **Radiant Flux (radiant power),** $\Phi = dQ/dt$**.** The time rate of flow of radiant energy, measured in joules per second or watts $= W$.

   (a) **Spectral Radiant Flux,** $\Phi_\lambda = d\Phi/d\lambda$**,** measured in W/nm.

3. **Radiant flux density,** $d\Phi/dA$**.** The quotient of the radiant flux incident on or emitted by a differential surface element dA at a point, divided by the area of the element. The preferred term for radiant flux density leaving a surface is exitance, M. The preferred term for radiant flux density incident on a surface is irradiance, E. Measured in watts per square meter, W/m^2.

   (a) **Spectral radiant flux density,** $d\Phi_\lambda/(dA\ d\lambda)$**.** In terms of exitance it is $M_\lambda/d\lambda$. In terms of irradiance it is $E_\lambda/d\lambda$. Measured in $W/(m^2\ nm)$.

4. **Radiant intensity,** $I = d\Phi/d\omega$**.** The radiant flux proceeding from a source per unit solid angle in a given direction. Measured in watts per steradian, W/sr.

   (a) **Spectral radiant intensity,** $I_\lambda = dI/d\lambda$**.** Measured in $W/(sr\ nm)$.

5. **Radiance,** $L = d^2\Phi/[d\omega(dA\cos\theta)]$**.** Power per unit projected area perpendicular to the ray per unit solid angle in the direction of the ray. Measured in $W/(m^2\ sr)$.

   (a) **Spectral radiance,** L_λ**.**
   $L_\lambda = d^3\Phi/[d\omega(dA\cos\theta)d\lambda]$. Measured in $W/(m^2\ sr\ nm)$.

A.2 Important Photometric Terms

Note that the symbols for radiometric and the corresponding photometric terms are the same. In cases where the terms might be confused radiometric terms will be identified by the subscript e and photometric terms will be identified by the subscript v.

1. **Luminous flux Φ.** Radiant flux evaluated in terms of a standardized visual response. Measured in lumens, lm.

$$\Phi_v = K_m \int_\Lambda \Phi_{e,\lambda} V(\lambda)\, d\lambda$$

where

Φ_v	=	lumens
$\Phi_{e,\lambda}$	=	watts per nanometer
λ	=	nanometers
$V(\lambda)$	=	the spectral luminous efficiency
K_m	=	the spectral luminous efficacy in lumens per watt (lm/W)

The above definition of luminous flux is for photopic vision and K_m has the value $683\, lm/W$. For scotopic vision $V(\lambda)$ is replaced by $V'(\lambda)$ and K_m is replaced by $K_{m'} = 1754\, lm/W$.

2. **Luminous flux density,** $d\Phi/dA$ This item is usually referred to as illuminance, E, if luminous flux density is incident on a surface element, and luminous exitance, M, if luminous flux density is leaving a surface element. Measured in lm/m^2

3. **Luminous intensity, $I = d\Phi/d\omega$.** The luminous flux per unit solid angle in a certain direction. Measured in lm/sr or candelas.

4. **Luminance, $L = d^2\Phi/[d\omega(dA\cos\theta)]$.** The definition is the same as radiance. The units are $lm/(m^2\ sr)$.

A.3 Deriving Everything from Radiance

All of the above definitions can be derived from spectral radiance. This is an important exercise which will help clarify the relationship between radiance and the other radiometric and photometric terms. In the following list, spectral radiance will be referred to as the function $L_e(x, \omega, \lambda)$.[3]

1. **Spectral Radiometry**

   - **Spectral radiant energy**

$$Q_{e,\lambda} = \int_T \int_\Omega \int_{x \in A} L_e(x, \omega, \lambda) \cos\theta\ dA\ d\omega\ dt$$

   - **Spectral radiant flux**

$$\Phi_{e,\lambda} = \int_\Omega \int_{x \in A} L_e(x, \omega, \lambda) \cos\theta\ dA\ d\omega$$

[3] We define only spectral radiometry since the corresponding radiometric terms can be found by integrating the spectral radiometric terms over the appropriate range of the light spectrum

- **Spectral radiant flux density** (in terms of irradiance)

$$E_{e,\lambda} = \int_{\Omega} L_e(x, \omega, \lambda) \cos\theta \; d\omega$$

- **Spectral radiant intensity**

$$I_{e,\lambda} = \int_{x \in A} L_e(x, \omega, \lambda) \; dA$$

2. **Photometry**

- **Luminous flux**

$$\Phi_v = K_m \int_{\Lambda} \int_{\Omega} \int_{x \in A} L_e(x, \omega, \lambda) V(\lambda) \cos\theta \; dA \; d\omega \; d\lambda$$

- **Luminous flux density**(in terms of illuminance)

$$E_v = K_m \int_{\Lambda} \int_{\Omega} L_e(x, \omega, \lambda) V(\lambda) \cos\theta \; d\omega \; d\lambda$$

- **Luminous intensity**

$$I_v = K_m \int_{\Lambda} \int_{x \in A} L_e(x, \omega, \lambda) V(\lambda) \; dA \; d\lambda$$

- **Luminance**

$$L_v = K_m \int_{\Lambda} L_e(x, \omega, \lambda) V(\lambda) \; d\lambda$$

A.4 IES Luminaires and Spectral Radiance

The IES photometric data file format [8] defines the three-dimensional distribution of light emitted by a luminaire. The distribution is defined for a point light source even though most luminaires are clearly not point sources. The file format specifies luminous intensities I_v for a set of vertical and horizontal directions, thus allowing for non-uniform distributions. To compute spectral radiance from this information we must make two assumptions: the distance from the luminaire to a point on the illuminated surface satisfies the "five-times" rule, and the spectral output of the luminaire is known. The five-times rule states that the luminaire can be modeled as a point source if distance from the luminaire to the point on the illuminated surface is greater than five times the maximum projected width of the luminaire as seen from the point. (In other words, the luminaire must not exceed a subtended angle of 0.2 radians as seen from the point.) If this rule is satisfied, the error for the predicted illuminance will be less than ± 1 percent [2].

The five-times rule allows us to model the luminaire as a photometrically homogeneous luminous aperture. That is, any point on the luminous surface of the luminaire will exhibit the same three-dimensional photometric distribution of luminous intensity as does the point source being used to represent the luminaire in the IES photometric data file.

Usually the type of lamp used in the luminaire will be defined in the IES file (although different lamps may be often be used when luminaire is installed). By maintaining a database of spectra that correspond to particular lamp types, we can satisfy the second assumption. Spectra from a number of generic lamp types are presented in the IES Lighting Handbook [9], while spectra for specific

lamps are often available from the lamp manufacturers. These spectra are given in terms of watts per nanometer, or spectral radiant flux ($\Phi_{e,\lambda}$). This allows us to derive the spectral radiant exitance $L_{e,\lambda}$ as follows:

The known quantities are luminous intensity $I_v = d\Phi_v/d\omega$, spectral radiant flux $\Phi_{e,\lambda}$, the maximum spectral luminous efficacy $K_m = 683$, and the photopic luminous efficiency curve $V(\lambda)$. The goal is spectral radiance $L_{e,\lambda}$.

Based on our assumption that the luminous surface of the luminaire is photometrically homogeneous, we have:

$$L_{e,\lambda} = \frac{d\, I_{e,\lambda}}{d\, A \cos\theta} = \frac{I_{e,\lambda}}{A \cos\theta} \tag{12}$$

where A is the luminous surface area of the luminaire as seen from the point on the illuminated surface and θ is the mean angle between the luminous surface normal and the direction of the point. (Remember that we are modeling the luminaire as a point source.) Therefore, we will have a solution for $L_{e,\lambda}$ if we can solve for the spectral radiant intensity $I_{e,\lambda}$.

We also have:

$$L_v = \frac{dI_v}{dA \cos\theta} = \frac{I_v}{A \cos\theta} \tag{13}$$

Now it is evident that the luminance L_v at the point on the surface is directly proportional to the amount of luminous flux Φ_v received at that point. The same argument must therefore hold for spectral radiance: $L_{e,\lambda}$ is directly proportional to the spectral radiant flux $\Phi_{e,\lambda}$. This gives us:

$$\frac{L_{e,\lambda}}{L_v} = \frac{\Phi_{e,\lambda}}{\Phi_v} \tag{14}$$

Rearranging terms gives us:

$$L_{e,\lambda} = \frac{L_v \Phi_{e,\lambda}}{\Phi_v} = \frac{I_v \Phi_{e,\lambda}}{(A \cos\theta)\Phi_v} \tag{15}$$

However:

$$\Phi_v = K_m \int_\lambda \Phi_{e,\lambda} V(\lambda)\, d\lambda \tag{16}$$

and so spectral radiance can be defined as:

$$L_{e,\lambda} = \frac{I_v \Phi_{e,\lambda}}{(A \cos\theta) K_m \int_\lambda \Phi_{e,\lambda} V(\lambda)\, d\lambda} \tag{17}$$

References

[1] James Arvo. *Analytic Methods for Simulated Light Transport*. PhD thesis, Yale University, December 1995.

[2] Ian Ashdown. *Radiosity: A Programmer's Perspective*. John Wiley & Sons, New York, 1994. includes C++ source code for fully functional radiosity renderer.

[3] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan-Kaufman, San Francisco, 1995.

[4] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. *Computer Graphics*, 20(4):133–142, August 1986. ACM Siggraph '86 Conference Proceedings.

[5] American National Standard Institute. Nomenclature and definitions for illumination engineering. ANSI Report (New York), 1986. ANSI/IES RP-16-1986.

[6] James T. Kajiya. The rendering equation. *Computer Graphics*, 20(4):143–150, August 1986. ACM Siggraph '86 Conference Proceedings.

[7] Eric P. Lafortune and Yves D. Willems. The ambient term as a variance reducing technique for Monte Carlo ray tracing. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 163–172, 1995.

[8] Illumination Engineering Society of North America. Ies standard file format for electronic transfer of photometric data and related information. IES Lighting Measurement Series, 1991. IES LM-63-1991.

[9] Mark S. Rea, editor. *The Illumination Engineering Society Lighting Handbook*. Illumination Engineering Society, New York, NY, 8th edition, 1993.

[10] Peter Shirley. *Physically Based Lighting Calculations for Computer Graphics*. PhD thesis, University of Illinois at Urbana-Champaign, January 1991.

[11] Gregory J. Ward. Measuring and modeling anisotropic reflection. *Computer Graphics*, 26(4):265–272, July 1992. ACM Siggraph '92 Conference Proceedings.

Global Illumination Input

Holly E. Rushmeier

This is an updated version of "Radiosity Input" that appeared in the course notes for "Making Radiosity Practical" at SIGGRAPH 93

1 General Remarks

A method for computing global illumination requires as input a geometric description of objects in an environment and their radiative properties. Restrictions on the geometries and properties (e.g. polygons only, perfect diffuse surfaces) obviously depend on the particular method and particular implementation of the method.

Geometry: One brief observation – an image will not appear realistic unless the geometric description is realistic. Remarkably realistic images can be synthesized with accurate geometry and direct illumination alone. Besides actually measuring geometries yourself (either with a measuring stick or more sophisticated three-dimensional scanner), typical dimensions for common architectural spaces and furniture can be obtained from handbooks such as [40]. Some sample geometry is available for free download at the Materials and Geometry Format website, at *http://radsite.lbl.gov/*. Commercial companies such as Viewpoint Datalabs sell libraries of three dimensional models.

Also, geometry can be modelled at different levels of detail, as discussed in [21]. At the largest scale are geometric representations such as triangle meshes, quadric surfaces and NURBs. At a finer scale are mappings such as bump maps and height fields. A method for changing between these representations is discussed in [6]. Bump maps and height fields can be obtained by processing scanned point clouds [23] or can be captured directly [30].

Color: Radiosity methods do not take colors as input, and they do not explicitly calculate colors. Radiosity methods take as input spectral data for light source emission and surface reflectances/transmittances at a series of wavelengths in the visible band. Essentially the wavelengths are chosen so that an accurate estimate of the continuous spectral radiance distribution leaving a surface can be made. A discussion of determining appropriate sample wavelengths can be found in [25].

Global illumination methods calculate radiances for each wavelength independently. The determination of the color associated with the calculated spectral radiance distribution is performed after the solution is complete, and the radiance distributions are mapped to the display device.

2 Emission

There are two major types of light sources – artificial and natural light (i.e. daylight). For a discussion of selection of sources for a particular environments see [19] or [20].

2.1 Artificial Light – Electrical Fixtures

Data on artificial lighting can be obtained from lighting manufacturers. In particular, the Ledalite Company (web site *http://www.ledalite.com/*) has data for their products, an excellent series of papers on the measurement of light sources by Ian Ashown ([1],[2], [3], [4], [5]) and many other resources for computing lighting accurately.

2.2 Natural Light

The spectral distribution and luminance for natural light depends on time of day, latitude and sky conditions (i.e. clear or over cast). Sample values can be found in the [20] or [9]. Note that different values for luminance and for the spectral distribution apply for direct (direct line to the sun) and indirect (from the hemisphere of the sky). Rough approximations of relative spectral distributions would be for a clear sky a blackbody at 15000K, for an overcast sky a blackbody at 6500 K, and for direct sunlight a blackbody at 5800K. A typical value for the the incident light due to indirect natural light is on the order of 1000 to 5000 cd/m^2. The magnitude of direct solar radiation is on the order of 1300 W/m^2. integrated over the entire electromagnetic spectrum (i.e. not weighted by luminous efficiency). A detailed example of applying the characterics of natural light to the generation of synthetic images can be found in [35].

Extensive work in simulating natural light using computer graphics global illumination calculations has been down by John Mardaljevic, and he has prepared a chapter on the topic for [37], and has a web page describing his work *http://www.iesd.dmu.ac.uk/~jm/*

3 Surface Reflectance/Transmittance

The spectral/directional data required to define bidirectional reflectance/transmittance distribution functions (BRDF/BTDF) for architectural materials is more difficult to find than the light source data. The BRDF/BRTF depends both on the chemical composition of the surface and on the surface condition (e.g.. perfectly smooth, rough, oxidized, etc.) Furthermore, many common materials do not have spatially uniform BRDF's (i.e. consider describing the BRDF for wood grain, or speckled formica).

A few electronic databases of BRDF data have recently become available. One is the Columbia-Utrecht data base at *http://www.cs.columbia.edu/CAVE/curet* that has measured data for 61 real world surfaces. Because the BRDF of a real world surface such as bread or straw varies with position, the data base introduces the concept of a bidirectional texture

function for representing the data. A description of the data collected and its application to computer vision can be found in [11].

Another electronic source is the Nonconventional Exploitation Factors Data Systems data base originally developed by the National Imagery and Mapping Agency. It is currently in the process of being made available by the US National Institute of Standards and Technology at *http://math.nist.gov/mcsd/Staff/RLipman/brdf/nefhome.html*. The database appears to include materials and characteristics that would be of particular interest in defense applications.

A database of BRDF for remote sensing from the department of geography at the University of Zurich is located at *www.geo.unizh.ch/~sandi/BRDF/about.html*. The goniometer used to measure this data is very large – so that it can measure the BRDF of a large patch of grass (for example.)

Non-electronic sources for reflectance/transmittance data include [36] and [8]. These are excellent references for materials with important thermal engineering applications – data for the chemical elements and common chemical compounds (e.g. silver iodide, silicon nitrate, etc.) can be found. However, you won't find data for many common architectural surfaces such as "simulated wood grained formica". Furthermore, even for well defined chemical compounds, full spectral BRDF data is not available. Generally spectral data is given for normal incidence and hemispherical reflectance or for reflection in the mirror direction for one specific angle of incidence. [32] contains spectral data (much of it in the infrared) for similar materials. However[32] also includes some spectral data for some building materials such as asphalt and brick, and plants such as lichen. Also included is the reflectance assorted foods such as the brown crust of baked bread (.06 at 400 and 500 nm, .14 at 600 nm and .38 and 700 nm.)

Handbooks for different fields contain a small amount of data for selected materials. For example [14], along with the spectral distributions for specular reflections for freshly evaporated silver and gold mirrors, also lists a spectral distribution for a ripe peach (.1 at 400 and 500 nm, .41 at 600 nm and .42 at 700nm) versus a green peach (.18 at 400nm, .17 at 500 nm, .62 at 600 nm and .63 at 700 nm). Data for other fruit are not given. [33] lists spectral reflectance for reflections from the water surfaces, as well as the spectral absorption of light by sea water.

Since full BRDF data is difficult to obtain, one alternative is to calculate a physically feasible BRDF from various local models given the complex index of refraction and surface roughness distribution (e.g. [10], [17] [27]). Complex indices of refraction can be found in handbooks such as [14]. Some sample roughness distribution functions are discussed in [16]. BRDF data can also be computed by casting rays at a mathematically defined surface microstructure [39] [15]. For imperfect and weathered surfaces Dorsey et al. have developed some techniques for representing the reflectances [12] [13].

Another alternative is to measure BRDF. This can be done (at non-trivial expense) at a commercial laboratory. The description of less expensive measurements of BRDF for can be found in [35] and [38]. More recently, methods for measuring BRDF have been developed that use inexpensive video capture systems. Karner et al. describe a system for

measuring the BRDF of flat samples [22]. Sato et al. describe a system for measuring the BRDF for which the shape has been measured by a range finding system [31]. Devices that are sold for print and monitor calibration, such as the Colortron *http://www.ls.com* can be used to measure spectral, if not directional, reflectances.

For the purposes of making some trial images here are some "reasonable" room values for total (i.e. averaged over the visible spectrum) diffuse reflectances (based on information in [20]):

- ceiling : 0.6 to 0.9, walls: 0.50 to 0.8, floor: 0.15 to 0.35
- furniture: 0.3 (dark wood) to 0.5 (blond wood)

Some typical values for specular materials:

- polished mirror: 0.99, polished aluminum: 0.65

For transmitting materials:

- clear glass: 0.80 to 0.99 basically "specular", solid opal glass : 0.15 to 0.40 basically "diffuse"

For trial purposes, a complete set of input data for a simple environment can be found in [24]. A larger set of sample data for a simple room comparison described in [29] can be found on-line at *http://radsite.lbl.gov/mgf/compare.html*

References

[1] Ashdown, I.“Making Near-Field Photometry Practical,”*1997 IESNA Annual Conference*, Seattle.

[2] Ashdown, I. “Near-Field Photometry: A New Approach,” *Journal of the Illuminating Engineering Society* 22(1):163-180 (Winter).

[3] Ashdown, I. “Virtual Photometry,” *Lighting Design+Application* 23(12):33-39 (December).

[4] Ashdown, I. “Photometry and Radiometry: A Tour Guide for Computer Graphics Enthusiasts,” Course 1: Realistic Input for Realistic Images, I. Ashdown, Ed. 1995 ACM SIGGRAPH Annual Conference, August 6-11, Los Angeles, CA.

[5] Ashdown, I. “Near-Field Photometry: Measuring and Modeling Complex 3-D Light Sources,” 1995 ACM SIGGRAPH Annual Conference, August 6-11, Los Angeles, CA.

[6] Becker, B., and Max, N. “Smooth transitions between bump rendering algorithms.” *Computer Graphics (SIGGRAPH ’93 Proceedings)* (1993), 183–189.

[7] K.R. Boff and J.E. Lincoln. *Engineering Data Compendium: Human Perception and Performance*, Vol. 1. Harry Armstrong Aerospace Medical Research Laboratory, Wright-Pattterson Air Force Base, 1988.

[8] J.F. Chaney, V. Ramidas, C.R. Rodriguez, and M.H. Wu, eds. *Thermophysical Properties Research Literature Retrieval Guide 1900-1980*. IFI/Plenum, New York, 1982.

[9] Committee on Colorimetry. *The Science of Color*. Optical Society of America, Washington, DC, 1963.

[10] R.L. Cook and K.E. Torrance. “A Reflectance Model for Computer Graphics” *ACM Transactions on Graphics*, January 1982, pp. 7-24.

[11] K.J. Dana, B. van Ginneken, S.K. Nayar, J.J. Koenderink, “Reflectance and Texture of Real World Surfaces”, *Proceedings of Computer Vision and Pattern Recognition ’97*.

[12] J. Dorsey and P. Hanrahan. “Modelling and Rendering of Metallic Patinas”*Proceedings of SIGGRAPH 96*, pp. 387-396.

[13] J. Dorsey, H. Pedersen and P. Hanrahan. “Flow and Changes in Appearance” *Proceedings of SIGGRAPH 96*, pp. 411-420.

[14] Dwight and Gray, eds. *American Institute of Physics Handbook*. McGraw Hill, New York, 1972.

[15] J. Gondek, G. Meyer and J. Newman, “Wavelength Dependent Reflectance Functions” *Proceedings of SIGGRAPH 94*, pp. 213-220.

[16] R. Hall. *Illumination and Color in Computer Generated Imagery*. Springer- Verlag, New York 1989.

[17] X.D. He, K.E. Torrance, F.X. Sillion and D.P. Greenberg. “A Comprehensive Physical Model for Light Reflection” *Proceedings of SIGGRAPH 1991*, pp. 175-185.

[18] H. Hewitt and A. S. Vause, Ed. *Lamps and Lighting*. American Elsevier, New York, 1964.

[19] R.G. Hopkinson. *Architectural Physics: Lighting*. Her Majesty’s Stationery Office, London, 1963.

[20] *IES Lighting Handbook*, 1981 Reference Edition.

[21] Kajiya, J. “Anisotropic reflection models.” *Computer Graphics (SIGGRAPH ’85 Proceedings)* 19, 3 (July 1985), 15–22.

[22] Karner, K., Mayer, H., and Gervautz, M. “An image based measurement system for anisotropic reflection.” *Proceedings of EUROGRAPHICS 96, Computer Graphics Forum 15*, 3 (1996), 119–128.

[23] Krishnamurthy, V., and Levoy, M. “Fitting smooth surfaces to dense polygon meshes.” *Computer Graphics (SIGGRAPH ’96 Proceedings)* (August 1996), 313–324.

[24] G.W. Meyer, H.E. Rushmeier, M.F. Cohen, D.P. Greenberg and K.E. Torrance. “An Experimental Evaluation of Computer Graphics Imagery.” *ACM Transactions on Graphics*, Jan. 1986, pp. 30-50.

[25] G.W. Meyer. “Wavelength Selection for Synthetic Image Generation.” *Computer Vision, Graphics, and Image Processing*,1988, p. 57-79.

[26] E. Nakamae, K. Kaneda, T. Okamoto and T. Nishita. “ A Lighting Model Aiming at Drive Simulators,” *Proceedings of SIGGRAPH 1990*, pp. 395-404.

[27] M. Oren and S. K. Naya, “Generalization of Lambert’s Reflectance Model” *Proceedings of SIGGRAPH 94* pp. 239-246.

[28] *The Photonics Design and Applications Handbook*, Book2. Lariun Publishing Company, Pittsfield, MA, 1988.

[29] H. Rushmeier, G. Ward, C. Piatko, P. Sanders and B. Rust, “Comparing Real and Synthetic Images: Some Ideas about Metrics”, *1995 Eurographics Workshop on Rendering.*

[30] Rushmeier, H., Taubin, G., and Guéziec, A. “Applying shape from lighting variation to bump map capture.” *Proceedings of the Eighth Eurographics Rendering Workshop* (June 1997), 35–44.

[31] Sato, Y., Wheeler, M., and Ikeuchi, K. “Object shape and reflectance modeling from observation.” *Computer Graphics (SIGGRAPH ’97 Proceedings)* (August 1997), 379–388.

[32] A. Sala. *Radiant Properties of Materials*. Elsevier, Amsterdam, 1986.

[33] F.G. W. Smith ed. *CRC Handbook of Marine Science*. CRC Press, Cleveland, 1974.

[34] *Sweet’s Catalog File: Products for Engineering*. McGraw-Hill, NY, 1981.

[35] A. Takagi, H. Takaora, T. Oshima and Y. Ogata. “Accurate Rendering Technique Based on Colorimetric Conception,” *Proceedings of SIGGRAPH 1990*, pp. 263-272.

[36] Y.S. Touloukian and D.P. DeWitt.*Thermophysical Properties of Matter, Vols. 7 and 8: Thermal Radiative Properties*. IFI/Plenum, New York, 1972.

[37] G. Ward Larson and R. Shakespeare, *Rendering with Radiance: The Art and Science of Lighting Visualization* Morgan Kaufmann, 1998.

[38] G.J. Ward. “Measuring and Modeling Anisotropic Reflection” *Proceedings of SIGGRAPH 1992*, pp. 265-272.

[39] S.H. Westin, J.R. Arvo, K.E. Torrance. “Predicting Reflectance Functions from Complex Surfaces,” *Proceedings of SIGGRAPH 1992*, pp. 255-264.

[40] W. Woodson. *Human Factors Design Handbook*, McGraw-Hill, New York 1981.

Input for Participating Media

Holly Rushmeier

This originally appeared in the SIGGRAPH 95 course notes on input for global illumination solutions

1 Introduction

Images of radiatively participating media are aesthetically appealing – curls of smoke, sunsets, fires and clouds. Generating physically accurate, rather than artistic, images of participating media is an extremely challenging computational problem. In computer graphics, significant effort has gone into developing computational methods to account for attenuation and multiple scattering in participating media (e.g. [27], [4], [32],[18],[5],[24], [3],[31], [34],[33]). While such methods are still extremely time consuming, the problem is well understood. However, far less attention has been given into obtaining and/or modeling appropriate input for rendering participating media. In many cases, getting realistic input data is much more difficult than computing the light scattering. In this section we will consider what data is needed and some possible approaches for getting it.

2 Defining The Problem

A reasonable place to begin is to define the problem of physically accurate rendering of participating media. The geometry of rendering a scene containing a participating medium is shown in Fig. 1.

As in rendering any realistic scene, the image is computed by finding the radiance (energy per unit time, solid angle and projected area) $L(s)$ which would pass through an image pixel to the eye. To form a final image, a weighted average of this value must be found across the pixel (for antialiasing) and the spectral radiance distribution must be mapped to the gamut of the display.

Unlike the surface problem, in which it is adequate to find the radiance of the closest visible surface, in the presence of a participating medium an integral along the line of sight must be evaluated. Along the line of sight, four processes may occur, absorption, out-scattering, in-scattering and emission.

2.1 Absorption

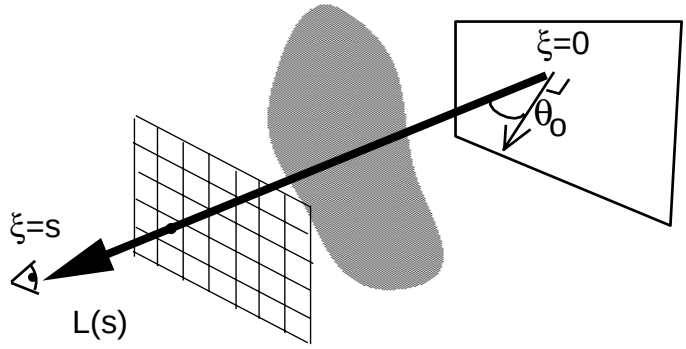


Figure 1: The geometry of rendering a scene with a participating medium. An image is formed by computing the radiance L(s) that reaches the eye through a pixel by integrating along the line of sight.

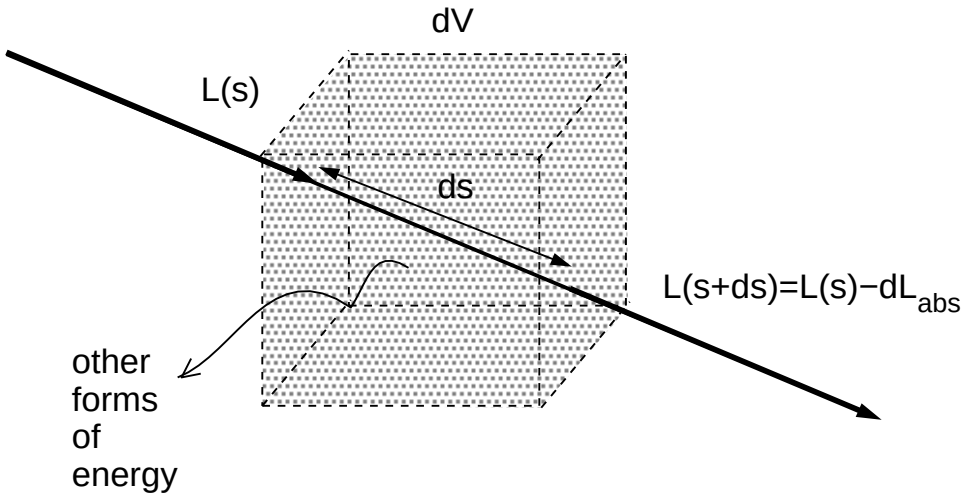


Figure 2: Absorption in a participating medium. Some of the the incident light energy leaves the path in another form.

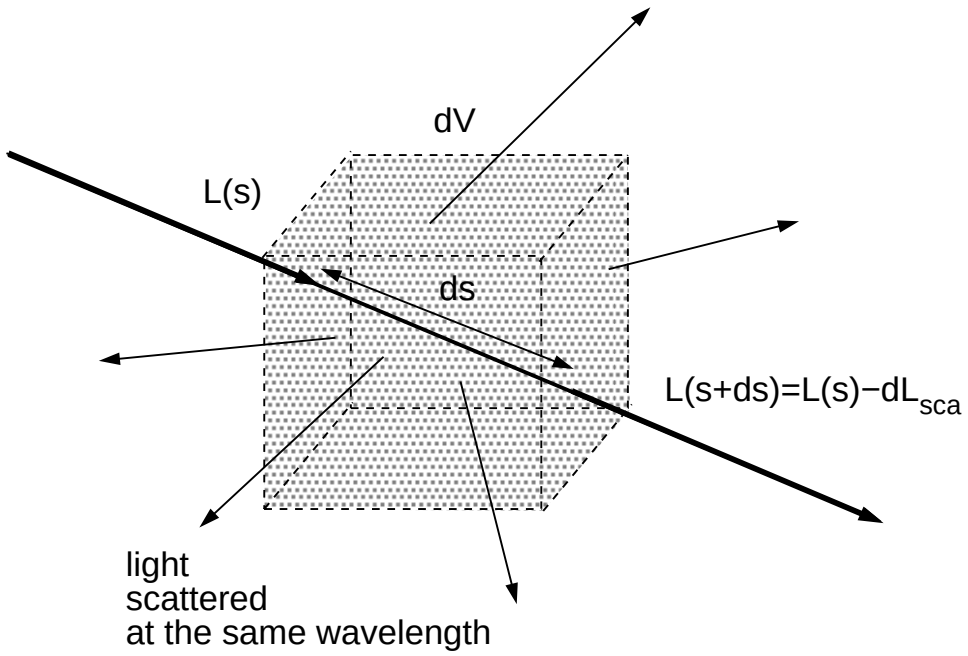


Figure 3: Scattering out of participating medium. Some of the the incident light energy leaves the path as light traveling in a different direction.

Figure 2 shows absorption – some fraction of the beam of light is absorbed by the medium. The light energy does not disappear, it is converted into another form. The energy transferred to the medium causes it to increase in temperature, or the energy is conducted or convected away. The ability of the medium to absorb light is expressed as the absorption coefficient σ_a, the fraction by which the beam of light is reduced by absorption *per unit length* traveled along the line of sight.

$$\left.\frac{dL(s)}{ds}\right|_{abs} = -\sigma_a L(s) \tag{1}$$

2.2 Out-Scattering

Figure 3 shows out-scattering – some fraction of the beam of light is scattered by the medium. This light is absorbed by the medium and immediately reradiated, but in directions that are different from the original path. The ability of the medium to scatter light out of the path is expressed as the scattering coefficient σ_s, the fraction by which the beam light is reduced by scattering *per unit length* traveled along the line of sight.

$$\left.\frac{dL(s)}{ds}\right|_{sca} = -\sigma_s L(s) \tag{2}$$

Bohren gives an example of a simple experiment that illustrates the difference between attenuation due to absorption and attenuation due to scattering. Referring to Fig. 4, place

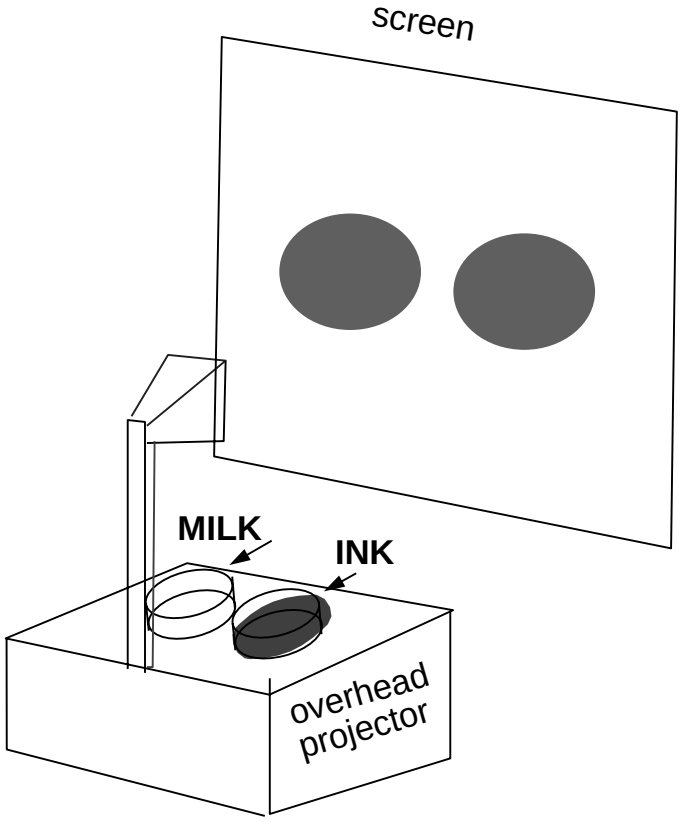


Figure 4: An experiment described by Bohren, illustrating attenuation by absorption and by out-scattering

two glass dishes of water on an overhead projector. Add ink to one dish, and milk to the other. Its possible to add ink and milk at rates such that the projection through the two dishes is the same on the screen - they have each attenuated the beam from the projector by the same fraction. However, the dish of ink will look much darker than the dish of milk. The ink has attenuated the beam by absorption, the milk has attenuated the beam by scattering. Bohren's book *Clouds in a Glass of Beer*[6] describes many other simple experiments that help develop a physical understanding of the interaction of visible light with participating media.

Because they both attenuate the radiance of a beam of light, the absorption and scattering coefficients are frequently combined into the extinction coefficient, σ_{ext}:

$$\sigma_{ext} = \sigma_a + \sigma_s \tag{3}$$

$$\frac{dL(s)}{ds} = -\sigma_{ext} L(s) \tag{4}$$

The effect of scattering relative to the effect of outscattering is expressed as the single

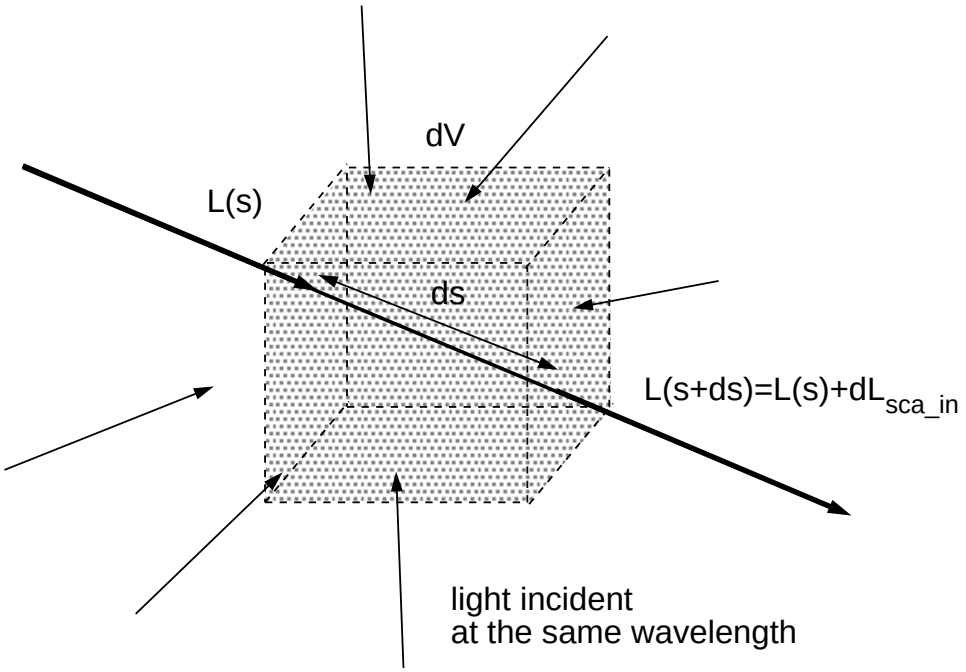


Figure 5: Scattering into the participating medium. Some incident light is scattered into the path.

scatter albedo Ω of a medium:

$$\Omega = \frac{\sigma_s}{\sigma_a + \sigma_s} \tag{5}$$

Referring back to the milk and ink experiment, the two media have similar extinction coefficients. The milk has a high albedo relative to the ink.

2.3 In-Scattering

Scattering can also result in augmentation of the beam of light, as diagrammed in Fig. 5. In-scattering from beams of light from other directions can increase the radiance along a line of sight. When discussing out-scattering, the directionality of scattering was unimportant – all that mattered was that light left the path. For in-scattering, the directionality of scattering is important to understand to what extent light from other directions is scattered into the path.

The directionality of scattering is expressed by the scattering phase function $P(\theta)$, where θ is the angle between the direction of scattering and the original path, as shown in Fig. 6. That is, forward scattering is in the direction for which θ is nearly zero. The phase function is a dimensionless quantity which is equal to the ratio of the radiance scattered in a particular direction $dL(\theta)$ to the radiance that would be scattered if the medium were

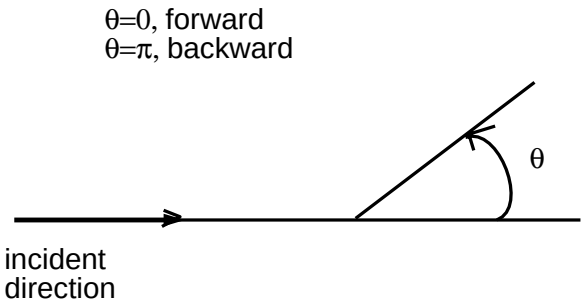


Figure 6: Definition of the angle in the scattering phase function.

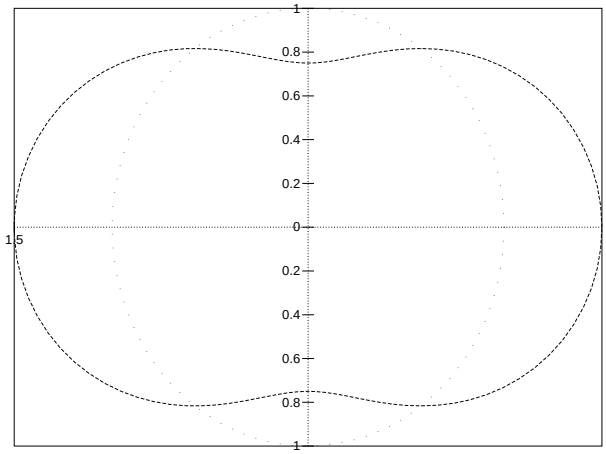


Figure 7: Rayleigh scattering phase function (isotropic shown with very light dotted line).

isotropic (i.e. if the medium scattered equally in all 4π directions $d\omega$):

$$P(\theta) = \frac{dL(\theta)}{\frac{1}{4\pi}\int dL(\theta^*)d\omega} \tag{6}$$

Two things to note about the phase function are that:

- the value of $P(\theta)$ is not bounded
- the function of $P(\theta)$ is normalized:

$$\frac{1}{4\pi}\int P(\theta)d\omega = 1 \tag{7}$$

Scattering phase functions are shown in polar plots in Figs. 7 and 8.

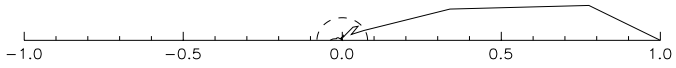


Figure 8: Mie scattering for a 525 nanometer radius sphere with index of refraction 1.5 (not normalized.)

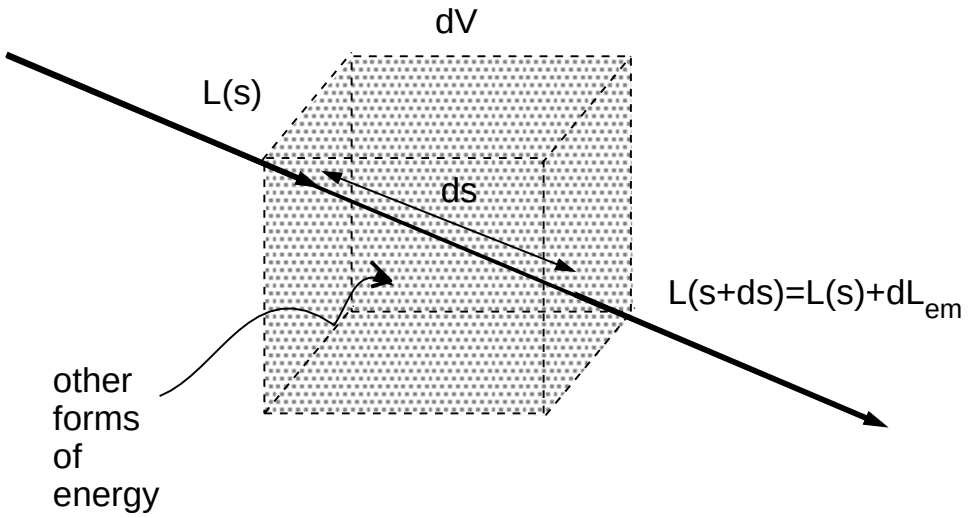


Figure 9: Emission in the participating medium. Energy in a form other than visible light enters the volume and causes the emission of visible light into the path.

The increase in radiance along a direction s then, due to scattering from a beam of radiance $L'(\theta)$ from direction θ to path s is $\frac{\sigma_s L'(\theta) P(\theta) ds}{4\pi}$. Adding up all of the contributions from all directions gives the increase in direction s as:

$$\frac{dL}{ds}_{in\_scat} = \frac{\sigma_s}{4\pi} \int L(\theta) P(\theta) d\omega \tag{8}$$

2.4 Emission

Finally, radiance may increase in a path due to emission within a volume, as shown in Fig. 9. If the emission is due to thermal agitation of the medium, the increase is given by the product of the absorption coefficient and the blackbody temperature of the medium L_b:

$$\frac{dL}{ds}_{em} = \sigma_a L_b \tag{9}$$

The reason the absorption coefficient appears in both absorption and emission terms is based on thermodynamics. Briefly, suppose a volume of medium at temperature T is in an black (totally absorbing) environment T. Both the volume and the environment emit radiation at a rate governed by T. If the volume didn't emit radiation at temperature T

at the same rate it absorbed, it would spontaneously change temperature – violating the laws of thermodynamics. This basic idea underlies the various reciprocity relationships in radiation(e.g. form factor reciprocity and reciprocity of the BRDF, see [36]).

Thermal emission is not the only type of emission that we see day to day. A notable exception are the fluorescent gases in fluorescent light fixtures. The emission can be expressed in the same form as Eq. 9, but the expression for obtaining L_b is not the same.

Putting together the four contributions to change in radiance along a path, the equation of transfer in a participating medium is:

$$\frac{dL(s)}{ds} = -\sigma_a L(s) - \sigma_s L(s) + \sigma_a L_b + \frac{\sigma_s}{4\pi}\int L(\theta)P(\theta)d\omega \tag{10}$$

In terms of extinction coefficient and albedo, this can also be written:

$$\frac{dL(s)}{ds} = -\sigma_{ext} L(s) + \sigma_{ext}(1-\Omega)L_b + \frac{\sigma_{ext}\Omega}{4\pi}\int L(\theta)P(\theta)d\omega \tag{11}$$

The product $\sigma_{ext}ds$ is a dimensionless length in the medium called the optical differential thickness. Setting the function $d\kappa$ equal to this dimensionless length, Eq. 11 can also be written:

$$\frac{dL(s)}{d\kappa} = -L(\kappa) + (1-\Omega)L_b + (\Omega/4\pi)\int L(\theta)P(\theta)d\omega \tag{12}$$

The optical thickness $\kappa(s)$ (also called optical depth or opacity) of a path through the medium is just the integral of the optical differential thickness:

$$\kappa(s) = \int_o^s \sigma_{ext} ds^* \tag{13}$$

The extinction coefficient expresses the effect a differential volume has on the incident light. The optical thickness of a medium expresses the effect of the entire extent of the medium. The optical thickness of a medium is a dimensionless length that can be used to compare the effects of volumes of medium. For example, a glass of milk of diameter 5 cm will attenuate a beam of light much more than the same glass filled with cigarette smoke at a density typically found in a restaurant. However, a volume of milk with optical thickness 1 will attenuate a beam of light exactly as much as a volume of cigarette smoke with optical thickness 1.

Looking at attenuation only, the radiance after traveling along a path s in a medium from a starting point at 0 is:

$$L(s) = L(0)e^{-\int_o^s \sigma_{ext} ds^*} \tag{14}$$

The quantity $e^{-\int_o^s \sigma_{ext} ds*}$ represents the fraction of light that emergences after traveling through a finite extent of a medium, and is generally referred to as the transmittance τ. Note that τ is a function of a finite extent of a medium, it is not a function of a differential volume at a point in the medium.

2.5 Summary of the Input Needed

In addition to the input data required for a surface-only problem, the definition of a problem containing a participating medium requires the definitions of L_b, $P(\theta)$, σ_{ext}, and Ω as functions of position in the medium. Unlike the surface problem in which geometry and reflectance properties are treated entirely separately, the definition of the geometry of a participating medium and its properties are closely coupled. If σ_{ext} is given directly as a function of location, the geometry of the medium is implied. The distribution of the medium may also be specified by giving partial pressure, volume fraction, or the density of the medium as a function of location. The values of σ_{ext} are computed by converting these quantities to densities, and using the mass coefficients of extinction (i.e. (fraction extinction/length)/(mass density)). The spatial distribution of scattering particles an gases may be constructed (e.g. by thoroughly mixing milk into water), but more often in environments of interest in graphics, they are determined by complex natural processes. For most media, the spatial modeling problem is closer to the complexity of modeling plants and animals than it is to the complexity of modeling a chair or a desk.

We will now look into determining input for participating media – properties of and emission from a differential volume of medium, and the spatial distribution of participating media.

3 Properties and Emission for a Differential Volume

Essentially there are two types of quantities we need at a differential volume at some point in space – the properties of the medium, σ_{ext}, Ω and $P(\theta)$, and the emission L_b. We will discuss properties first, and then turn to emission.

Similar to the study of surface reflectance, measured values of gas or particulate absorption and scattering properties may be used directly, or analytical models may be used to calculate them from more fundamental measurements of optical properties and microscopic geometry. We begin with the analytical approaches.

3.1 Analytical Models for Properties

Similar to approximations of reflectance at surfaces, there are two common approaches to modeling the properties of volumetric media – geometric optics for particles that are large relative to the wavelength of light (e.g. as used in [9] for large surface roughness scales), and physical optics for smaller particles (e.g. as used in He [19] for smaller surface roughness scales).

3.1.1 Geometric Optics

Large Specular Spheres The geometry of light intersecting a large specular sphere is shown in Fig. 10a. The reflectance of a specular reflecting surface as a function of angle of incidence $\rho(\beta)$ is given by the Fresnel equations. Integrating over all incident angles gives

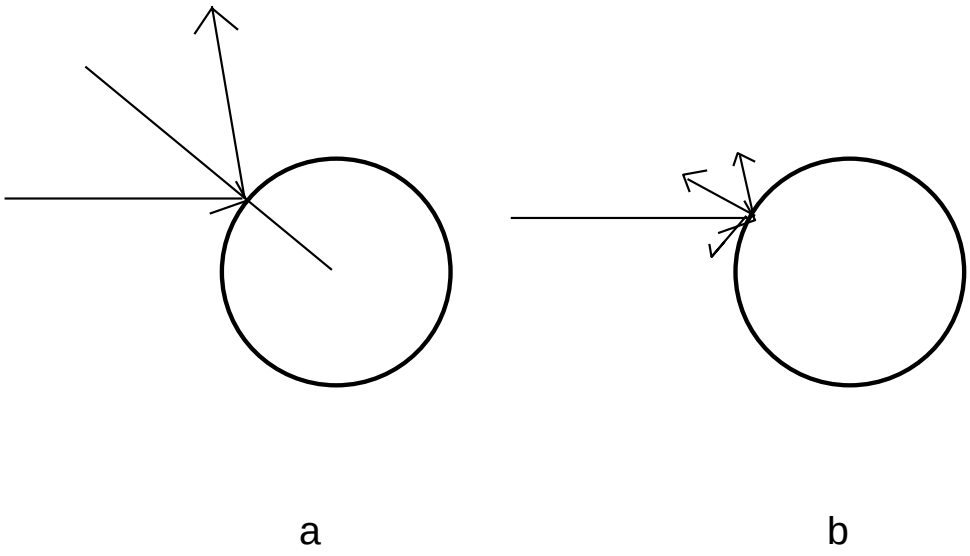


Figure 10: A ray striking a large specular (a) and diffuse(b) sphere.

the hemispherical reflectance ρ_h. Using the Fresnel reflectance, the properties for a cloud large specular spheres, with a size distribution of N(R) spheres of radius R per unit volume, are [36]:

$$\sigma_s = \rho_h \int_0^\infty \pi R^2 N(R) dR \tag{15}$$

$$\sigma_a = (1 - \rho_h) \int_0^\infty \pi R^2 N(R) dR \tag{16}$$

$$P(\theta) = \frac{\rho((\pi - \theta)/2)}{\rho_h} \tag{17}$$

The scattering and absorption coefficients depend only on the number density of the particles (which gives the cross sectional area along the path which is blocked by particles) and the hemispherical reflectance (which determines which fraction of the light which hits particles is absorbed and which is scattered).

Large Diffuse Spheres The geometry of light intersecting a large diffuse sphere is shown in Fig. 10b. The values of σ_s and σ_a are the given by the same expressions as for the specular case. However, the change in directional variation of the reflectance results in the following the scattering phase function [36]:

$$P(\theta) = \frac{8}{3\pi}(sin\theta - \theta \cos\theta) \tag{18}$$

Rainbows Geometric optics can also be used to approximate the scattering that results in rainbows [7]. Raindrops have diameters on the order of a millimeter, over 1000 times the wavelength of visible light. Raindrops essentially do not absorb the visible light, and scattering occurs as a result of internal reflections and transmissions. In particular consider rays that are refracted, internally reflectance and refracted again as in Fig. 11. Because

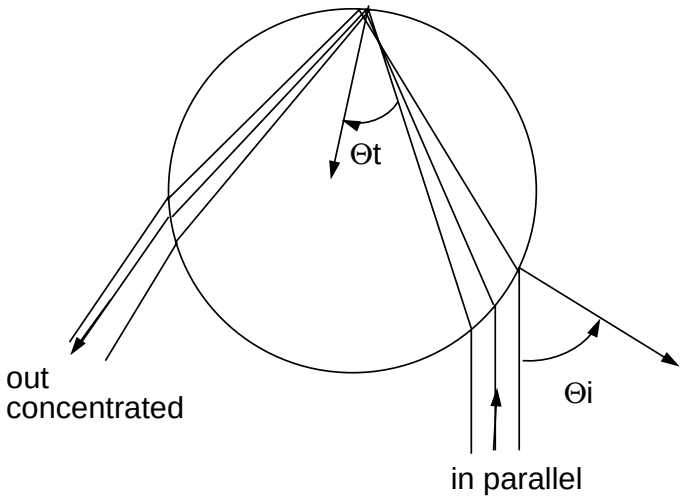


Figure 11: Ray paths that result in a rainbow.

of the curved surface of the raindrop, and the fact that the index of refraction of water is greater than that of air, the rays are concentrated, or a *caustic* is formed. Because the index of refraction is different for different wavelengths, these concentrations are at different positions for different wavelengths, and we see a bow of colors, rather than just a bow of bright light. Rainbows occur when the angle of incidence Θ_i to the surface of the drop is equal to:

$$cos(\Theta_i) = \sqrt{\frac{m^2 - 1}{3}} \tag{19}$$

The angle of scatter θ after a single internal reflection is:

$$\theta = 2\Theta_i - 4\Theta_t + \pi \tag{20}$$

Geometric optics cannot predict the correct radiance for a rainbow (the geometric optics theory breaks down, and a value of infinity is obtained). However, Eqs.19 and 20 can be used to determine when rainbows can occur, and from which vantage points they will be visible.

3.1.2 Physical Optics

For particles of arbitrary size, electromagnetic theory must be used to accurately develop an analytical expression for absorption and scattering. A solution of Maxwell's equations needs to be found for the electric and magnetic fields inside the particle of interest, and outside of it. The absorption coefficient, scattering coefficient and scattering phase function can be found from this solution.

The most famous solution of the problem is for the intersection of a plane wave with a sphere with an arbitrary radius a and complex index of refraction $n + ik$, as shown in

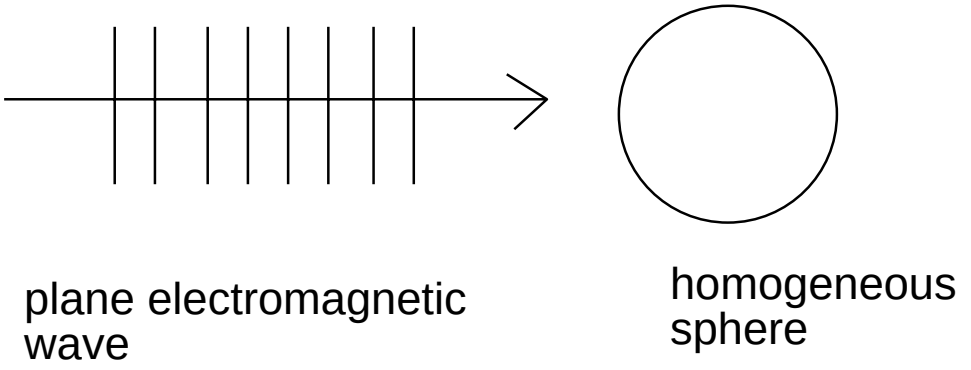


Figure 12: Geometry of the Mie solutions.

Fig. 12. This solution is referred to as the *Mie scattering theory*. Detailed descriptions of the solution are given in [7] and [39].

The Mie solutions are generally given in terms of cross sections, C_{sca} for scattering cross section and C_{ext} for extinction cross section. The scattering and extinction coefficients are found from these cross sections by multiplying by number density N of particles:

$$\sigma_s = C_{sca} N \tag{21}$$

The solution is most compactly expressed using *Riccati-Bessel* functions, ψ and ξ, and expressing the radius as a the ratio of $x = \frac{2\pi(n+ik)a}{\lambda}$ and the ratio m of the indices of refraction of the sphere to its surroundings. Note that these are both complex numbers. The coeffiecients which appear in the series solution for the electric fields are then:

$$a_n = \frac{m\psi_n(mx)\psi_n'(x) - \psi_n(x)\psi_n'(mx)}{m\psi_n(mx)\xi_n'(x) - \xi_n(x)\psi'(mx)} \tag{22}$$

$$b_n = \frac{\psi_n(mx)\psi_n'(x) - m\psi_n(x)\psi_n'(mx)}{\psi_n(mx)\xi_n'(x) - m\xi_n(x)\psi'(mx)} \tag{23}$$

where the prime indicates the derivative of the function. In terms of these coefficients:

$$C_{sca} = \frac{2\pi}{k^2}\sum_1^\infty (2n+1)(|a_n^2| + |b_n^2|) \tag{24}$$

$$C_{ext} = \frac{2\pi}{k^2}\sum_1^\infty (2n+1) Re(a_n + b_n) \tag{25}$$

Letting Q_n^1 denote Legendre polynomials (since we have already used $P()$ for the phase function), for unpolarized light the scattering phase function is given by:

$$P(\theta) = \frac{1}{2}(|\sum \frac{2n+1}{n(n+1)}(a_n \frac{Q_n(\theta)}{\sin\theta} + b_n \frac{dQ_n(\theta)}{d\theta})|^2 + |\sum \frac{2n+1}{n(n+1)}(b_n \frac{Q_n(\theta)}{\sin\theta} + a_n \frac{dQ_n(\theta)}{d\theta})|^2) \tag{26}$$

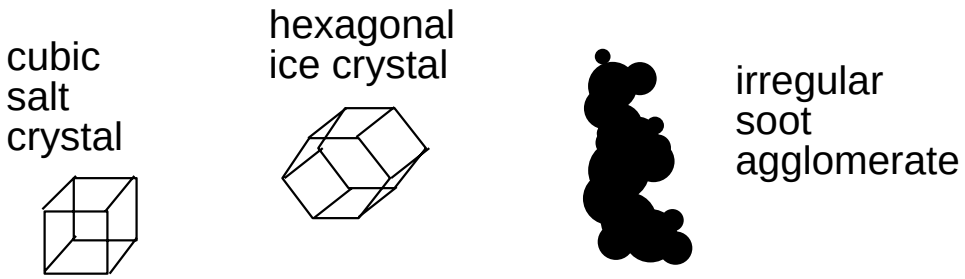


Figure 13: Common particle shapes that are not modeled well by Mie theory.

The Mie results are stated in this short form here not as a guide to computation, but to show that the solution is completely known, and that solutions can be obtained by computing enough terms in the infinite series – which are convergent. Code is available from many places to compute the Mie results, such as in the appendix to Bohren. Figure 8 shows results computed with this code.

Just having a code to compute Mie scattering doesn't solve the input problem. The complex index of refraction of the media being modeled is required, as is a size distribution of the particles in the medium. Furthermore, although it is quite a detailed solution, it does require the assumption of spherical particles. This is probably a good assumption for atmospheric clouds composed of water droplets. Water has a complex index of refraction of $1.33 + i10^{-8}$. Reference ([30], p. 187) gives values for the size and number density of droplets, with radii of 4 μm and number densities of 300 per cm^{-3} being typical for atmospheric clouds composed of liquid water droplets (as opposed to clouds composed of ice crystals).

Frequently, input for Mie calculations is not given directly, but must be extracted from reports. For example, a special issue of *The Journal of Geophysical Research* had several papers recording measurements of the smoke plumes from the Kuwaiti oil fires, eg. [13], [21]. Overall black plumes were found to be composed of elemental carbon particles, with a typical diameter of $0.5\mu m$ and density of 1000 $\frac{\mu gm}{m^3}$. This must be coupled with the information that elemental carbon has a complex index of refraction of $1.59 + .66i$, and the mass density of solid carbon is $2g/cm^3$. White smokes were found to be composed primarily of salts, with particles of 0.2 μm diameter and density of 1000 $\frac{\mu gm}{m^3}$. This must be coupled with a typical complex index of refraction of salt of $1.5 + 0i$ and mass density of solid salt of $2.2g/cm^3$.

Mie theory doesn't give good results for some particles of interest in rendering, such as those shown in Fig. 13. Clouds composed of ice crystals are not well modeled with Mie theory [30]. Dobbins et al. [10] show that for irregularly shaped soot agglomerates, Mie scattering theory gives results for cross sections that can err by as much as a factor of two.

A special case of Mie scattering theory is scattering from very small particles, generally known as Rayleigh scattering. For this case the series expansion for the scattering cross

section of particles is:

$$C_{sca} = \frac{8}{3}\frac{\pi D^2}{4}(\frac{\pi D}{\lambda})^4|\frac{(n+ik)^2-1}{(n+ik)^2+2}|^2 \tag{27}$$

and the scattering phase function is:

$$P(\theta) = \frac{3}{4}(1+cos^2\theta) \tag{28}$$

Cigarette smoke consists of particles with diameters less than 0.1 μm, and can be modeled as Rayleigh scatterers. Number densities of particulates in a room with a couple of smoldering cigarettes is on the order of 50,000 cm^{-3} [29]. Since the scattering cross section is proportional to $\frac{1}{\lambda^4}$, much more light is scattered at short wavelengths (the blue end of the visible spectrum) than at longer wavelengths. As a result, scattered light from cigarette smoke generally looks bluish.

Molecular scattering has the same phase function. However, rather than modeling a molecule as a particle with diameter D, the scattering cross section is given by [30],p. 166:

$$C_{sca} = 1.06\frac{8}{3}\pi^3\frac{n^2-1^2}{\lambda^4 N^2} \tag{29}$$

where index of refraction is approximated by:

$$(n-1)10^8 = 6430 + \frac{2,950,000}{146-\lambda^{-2}} + \frac{25,500}{41-\lambda^{-2}} \tag{30}$$

(λ in microns.)

A typical value for the number density of molecules in the atmosphere N is $2.55\text{x}10^{19} cm^{-3}$.

The attenuation coefficient for molecular scattering becomes significant only over distances of kilometers. In the atmosphere, the $\frac{1}{\lambda^4}$ dependence in Eq. 29 is apparent in the blue color of the sky.

3.2 Measured Properties

Because measuring the shape, size distribution and optical properties of particles of common participating media can be extremely difficult, it is often easier to rely on measured values for scattering and absorption coefficients.

For example [17] describes a workshop on measuring the interaction of light with aerosol particles. Measurements of absorption coefficients, mass of particles per unit volume of air, and albedo are given for various test cases using soot, methylene blue, salt and Arizona road dust. For example the samples of Arizona road dust had typical values of about 7 x 10^{-6} m^{-1} for absorption coefficient, and 0.7 for albedo.

To describe measured scattering distributions, fitting Mie parameters would be very tedious. Instead the Heyney-Greenstein function is generally used:

$$P_{HG}(\theta, g) = \frac{1-g^2}{(1+g^2-2gcos\theta)^{\frac{3}{2}}} \tag{31}$$

The parameter g indicates the asymmetry of the distribution. Reference [30] gives typical values of σ_{ext}, Ω and g for cirrus clouds composed of ice crystals. For example, for cirrus uncinus, these values are 2.61 km^{-1}, .9999, and 0.84 respectively.

3.3 Emission

Emission from volumetric media is generally rarer in rendering problems than absorption and scattering. One of the most prominent examples of emission is flames. Most visible light from flames comes from emission due to thermal agitation from soot. The "blackbody" radiance is given by Planck's equation:

$$L_b = \frac{2C_1}{\lambda^5(exp(\frac{C_2}{\lambda T}) - 1)} \tag{32}$$

where C_1 is approximately 0.59544x10^{-16} Wm^2, and C_2 is 14,388 μmK.

A feature of the Planck distribution is that the product of the wavelength of peak emission and the temperature, $\lambda_{max}T$ is a constant. This is known as Wien's displacement law. It means that the higher the temperature the lower the peak wavelength. For low temperatures, like room temperature, λ_{max} is in the long, infrared wavelengths. The spectrum of sunlight is approximately the same as a blackbody at 5600 K, with a peak around blue in the visible spectrum.

Obviously, the temperatures in fires differ. A typical pool fire temperature is on the order of 1000 K. In this temperature range the flame will tend to look orange or yellow.

Other types of particles may also have emission due to thermal agitation. Siegel and Howell [36] cite an example in rocket design in which aluminum oxide particles are introduced into exhaust, and contribute to the luminosity of the plume.

Generally other colors in common flames – the blue color of a methane flame – do not come from thermal emission, but from electron transitions.

4 Spatial Distribution of Absorbing/Scattering Media

Similar to properties and emission, the spatial distribution of a medium can be computed from a model, or can be obtained from measurements.

4.1 Fluid Mechanics

In general, the distribution of participating media can be modeled analytically using the principles of fluid mechanics. There is neither space nor time to discuss particular methods for solving problems in fluid mechanics, and in this section we simply present some basic ideas and vocabulary for understanding literature in this area.

A fluid is any substance that moves continuously under a shear stress. Both gases and liquids are fluids. A fluid is said to be Newtonian if this shear stress is linearly proportional to the velocity gradient in the medium. The constant of proportionality is the viscosity. The

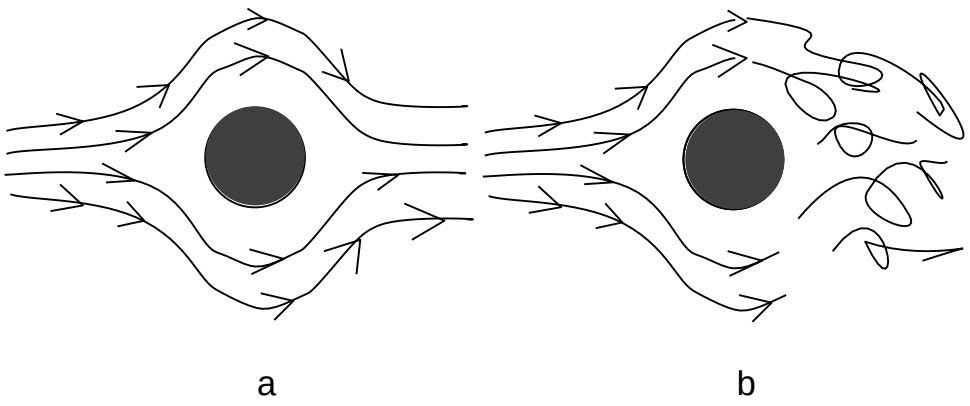


Figure 14: Laminar (a) and turbulent (b) flow over a sphere.

viscosity indicates how thick the fluid is – in the sense that maple syrup is much thicker than water.

The motion of a fluid is governed by the equations of conservation of mass and energy and the Navier-Stokes equations. The conservation of mass equation is frequently referred to as the continuity equation. The Navier-Stokes equations express conservation of momentum in the fluid. Derivations of these non-linear differential equations can be found in any standard fluid mechanics or heat transfer undergraduate textbook (e.g. [14], [23]), or in more advanced texts such as [26] and [1]. Full solutions of the Navier-Stokes and mass and energy equations are rarely required for practical problems. For example some problems are isothermal, so the energy equation is not needed. In some problems viscous forces are very small compared to inertial forces, so inviscid equations can be used. When viscous forces are high relative to inertial forces, creep flow equations can be used.

Generally the fluids literature refers to two regimes of flow – laminar and turbulent, diagrammed in Fig. 14. Laminar flow is orderly and layered, while turbulent flow is characterized by rapid fluctuations.Many flows of interest in rendering – such as the smoke plume from a large fire – are turbulent.

Flows are characterized by the Reynolds number, Re, which quantifies the importance of inertial to viscous forces. Re is defined as $\frac{\rho v L}{\mu}$ where ρ is the mass density, v is velocity L is a characteristic length of the flow and μ is the viscosity. The characteristic length is measured differently for different types of flows, as shown in Fig. 15. The transition from laminar to turbulent flow occurs at a critical value of Re. In flows below this value, perturbations are damped out before the flow becomes unstable. In flows with Re larger than the critical value,the perturbations grow. The Re number for which the transition occurs depends on the particular flow geometry – it takes on much different values for pipe flows than for flows over a flat plate. Futhermore, the Re for transition is not a sharp cut off – transitions are experimentally observed over a range of numbers. Because of the complicated nature of the Navier-Stokes equations, critical values of Re have not been derived analytically.

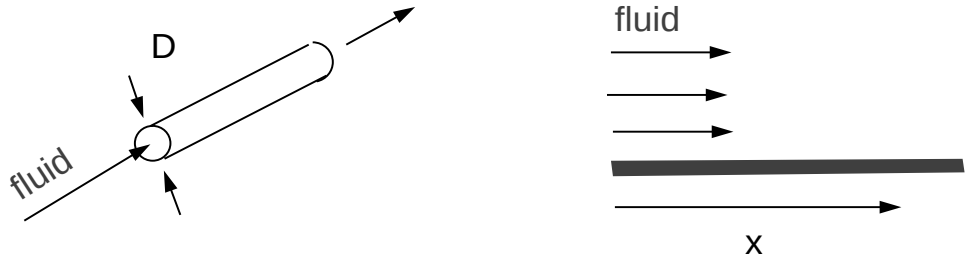


Figure 15: Definition of characteristic lengths for the Reynold's number for different types of flow.

Turbulence is the result of perturbations introduced into the flow when the Reynold's number is high enough – such as irregularities caused by surface roughness in a pipe. Generally pipe flow becomes turbulent at about Re equal to 2000, but when conditions are carefully controlled, laminar flow has been observed at Re up to 40,000 [35]. In fact there is no known upper limit to the Re at which laminar flow could be observed, if no perturbations were introduced to the flow.

Interesting laminar flows can be computed by direct solution of the Navier-Stokes equations. Mathematically, there are two ways that a flow can be characterized. One way, referred to as the Lagrangian method, is to follow fluid particles through time. The other way, the Eulerian method, is to solve for the velocity at each point in space as a function of time. Generally, a solution for the main fluid (typically water or air) is computed with the Eulerian approach. A grid with velocities as a function of time is computed. The distribution of particulates (e.g. water droplets, soot, dust) which scatter and absorb light can then be found by following them as Lagrangian particles in this flow field.

In principle, solutions for turbulent flow, like laminar flow, could be computed by direct numerical solution of the Navier-Stokes equations. The problem is that the non-linearity of the equations requires that the numerical grid be capable of capturing the fluid flow at a extremely wide range of length scales, to capture both the large scales of the flow (e.g. the entire length of the fluid being studied) to the small scales at which fluctuations are finally damped out by viscous dissipation. The range of length scales required grows with Re. Reference [28] gives the example of a small wind tunnel problem, where the length scale ranges from 50 mm for the size of the tunnel to 0.1 mm for the dissipation length scales. A solution would be needed at $(50/.1)^3$ or approximately 10^8 points. For atmospheric phenomenon, solutions would be required on grids of on the order of 10^{20} points.

One alternative to direct numerical simulation is referred to as Large Eddy Simulations (LES)[2]. In this case an additional model is introduced to account for the effects of turbulence at subgrid length scales. The results of an LES calculation is show in Fig. 16. The use of subgrid models is limited, because the non-linearity of the fluid equations prevents a complete decoupling of the various length scales. This approach is only useful for the range of flows for which the additional model of subgrid turbulence has been validated.

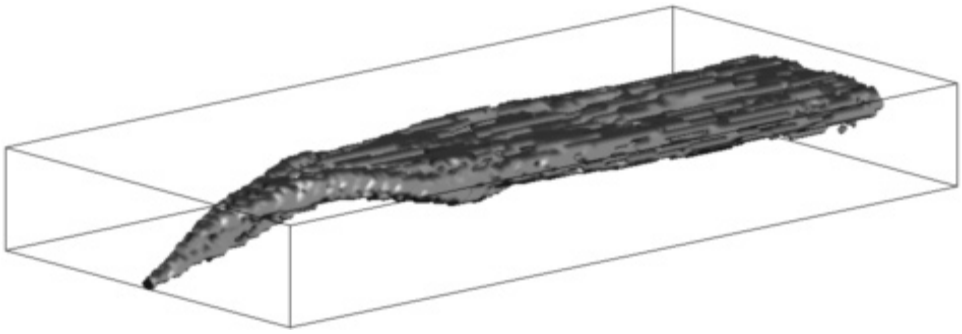


Figure 16: An isosurface of the mass density distribution for smoke plume computed with a Large Eddy Simulation.

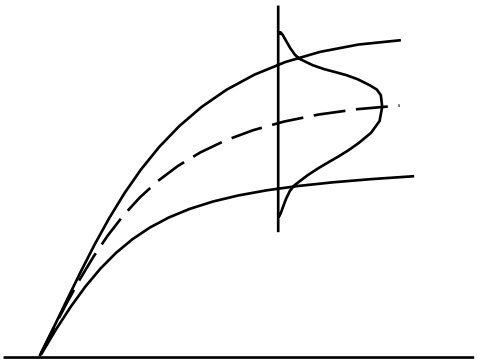


Figure 17: Baseline battlefield plume definition used by Hoock. The centerline is defined with the height equal to downwind direction raised to a power. The concentration in the plume is a Gaussian of distance from the centerline.

Because of the length scale problem in computing direct solutions, alternative approaches have been developed for modeling turbulence. In particular, statistical methods and dimensional arguments have been used [38]. For example a flow can be viewed as being composed of eddies of various lengths – ranging from a characteristic length in the problem, to the length scale of viscous dissipation. One model is that energy is transferred from the largest scale eddies to the smallest, without loss. This process is referred to as "the energy cascade." Using statistical arguments for the special case of homogeneous turbulence, the energy of eddies in this cascade scale according to the wavenumber of the eddy raised to a power. This power law can be used to determine a realistic spectrum of spatial and temporal variations to simulate homogeneous turbulence.

A complete example of successfully using the energy cascade approach is given by Hoock [20] to a Gaussian plume which has an overall shape (centerline and width) shown in Fig. 17. The basic plume centerline is modeled with the plume height equal to downwind

distance raised to a power – with the coefficient and power based on experimental observations of various plumes. The basic centerline is then perturbed according to a model of wind conditions. The basic particulate concentration of the plume is a Gaussian distribution from the centerline. The width of the distribution increases along the centerline, and depends on an estimated rate of entrainment of ambient fluid into the plume. The base distribution is then perturbed by sinosoidal fluctuations in concentration. These fluctuations simulate turbulent eddies of various length scales. The amplitude of these fluctuations is inversely related to their spatial frequency to emulate the observed "energy cascade" in turbulent flows.

In computer graphics, Stam and Fiume [37] have applied the approach of using a power law relationship between energy and length scales to compute realistic looking particulate distributions.

Of course, as in the case of electromagnetic solutions for scattering, being able to solve a fluid mechanics problem doesn't solve the input problem for participating media. If a fluids model is to be used, the appropriate input for that computation has to be found – i.e. initial and boundary conditions for the the velocities and pressures in the field.

4.2 Measured Density Distributions

Because of the difficulties in finding solutions for the fluid flows that frequently of interest in rendering, an alternative is to use measured distributions. As illustrated in the case of battle field plumes, experimental data can be found to model at least the overall spatial distribution of the participating media.

Numerous studies in the fire science literature are available giving the mass distribution of smoke particles as crude (i.e. not well spatially resolved) functions of height and time [8], or the optical thickness in an enclosure as a function of height and time [11].

Liou [30] gives data for overall size distribution for atmospheric clouds, as well as data for the number density and composition droplet/crystal in the cloud. For example the size distribution of cumulus clouds per km^2 surface area observed from satellite photographs is given. This type of bulk data coupled with mathematical functions which mimic observed cloud shape, as presented by Gardner [15], could be used to rendering physically realistic clouds.

There is no one combined source for data on spatial distributions of participating media. However, both particle characteristics and spatial distributions for particular types of flows can be assembled for a particular problem from data presented in journals such as the *The Journal of Geophysical Research*(e.g. [13]), *Atmospheric Environment* (e.g. [40]), and *Journal of Aerosol Science* (e.g. [29]).

4.3 Fire

The complexity of computing the spatial distribution of emitting, absorbing and scattering media is compounded in the case of fires. Not only are most fires turbulent, but the chemistry of the combustion process must be included in any type of physical simulation.

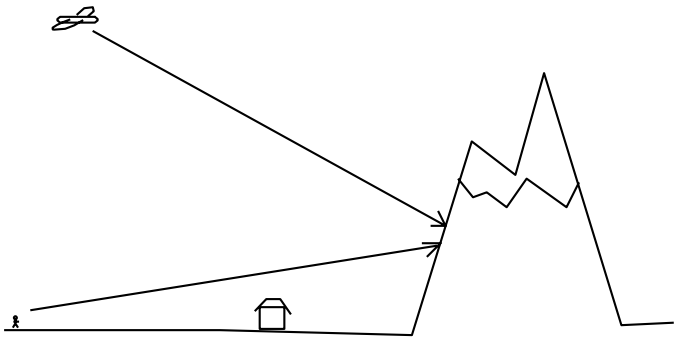


Figure 18: The LOWTRAN code is typically used to compute the effect of the atmosphere on individual, kilometers long, lines of sight.

To an even greater extent, input for accurate models of fires needs to be obtained from observations and measurement. In computer graphics, this approach has been used by Inagake [22], using descriptions of flame structure from Gaydon and Wolfard [16], and Faraday [12].

5 Existing Codes – LOWTRAN

Because attenuation and scattering through the atmosphere is so important in remote sensing applications, there is an extensive body of literature on this topic. For computation, many of the models for the transport of radiation have been included in the program LOWTRAN [25]. LOWTRAN is one of the most-used large scale scientific programs, and is cited widely in the remote sensing literature. Typical, kilometers long, lines of sight for which LOWTRAN is used to compute transmittance and radiance through the atmosphere are shown in Fig. 18.

LOWTRAN was developed over decades at the Air Force Geophysics Laboratory (AFGL) to include a wide range of phenomena. A variety of model atmospheres can be selected, e.g. tropical, subartic summer, etc. Many different types of aerosols can be included such as fog, volcanic dust and typical desert aerosols. Clouds of different types can be specified. Different models for rain can be used. The various models used for the properties of various atmospheric components are detailed in a long series of technical reports from AFGL.

The name "LOWTRAN" comes from the relatively low spectral sampling for many atmospheric applications. The sampling is at 5 cm^{-1}, which is a low rate at the far infrared (wavenumber 20 cm^{-1} at λ of 500 μm). In the visible range (wave numbers on the order of 20,000 cm^{-1}), it is a relatively high sampling rate for graphics researchers accustomed to sampling 3 wavelengths.

LOWTRAN covers many phenomena and wavelengths (into the ultraviolet and out into the infrared) which are not of interest in visible image synthesis. And, as a FORTRAN

program which has evolved over many years, the code itself is unwieldy to work with. However, for building a renderer for atmospheric effects, the LOWTRAN documentation is a good starting point for understanding the important effects to model, and the LOWTRAN code could be used to check the accuracy of line intergration of a visible image renderer.

6 Summary

For most problems of interest in rendering, it is essentially impossible to obtain completely accurate input data for the properties, emission and spatial distribution of participating media. While there are detailed analytical solutions, such as the Mie scattering theory, for some aspects of the problem, these solutions require restrictive assumptions and input data that may also be difficult to obtain. Obtaining a physically accurate set of input data requires using a mix of analysis and measured data that are appropriate for the particular rendering problem at hand.

In the introduction to *A First Course in Turbulence*[38], Tennekes and Lumley write:

"In turbulence the equations do not give the entire story. One must be willing to use (and capable of using) simple physical concepts based on experience to bridge the gap between the equations and the actual flows. We do not want to imply that the equations are of little use; we merely want to make it unmistakably clear that turbulence needs spirited inventors just as badly as dedicated analysts."

Similarly, for the entire problem of modeling input for participating media, invention based on the simple physical concepts is required as well as detailed mathematical analysis.

References

[1] G.K.S. Batchelor. *An Introduction to Fluid Dynamics.* Cambridge University Press, 1967.

[2] H.R. Baum, K.B. McGrattan, and R.G. Rehm. Simulation of smoke plumes from large pool fires. In *The Proceedings of the Twenty-fifth International Symposium on Combustion*. The Combustion Institute, 1994.

[3] N. Bhate. Application of rapid hierarchical radiosity to participating media. In *Proceedings of AATRV-93: Advanced Techniques in Animation, Rendering and Visualization*, pages 43–53, 1993.

[4] P. Blasi, B. Le Saëc, and C. Schlick. A rendering algorithm for discrete volume density objects. In *Proceedings of Eurographics 1993*, 1993.

[5] J.F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. In *Proceedings of Siggraph 1982*, pages 21–29. ACM SIGGRAPH, 1982.

[6] C. Bohren. *Clouds in a Glass of Beer*. John Wiley and Sons, 1987.

[7] C. F. Bohren and D. R. Huffman. *Absorption and Scattering of Light by Small Particles*. Wiley, 1983.

[8] B. Collins. Visibility of exit signs in clear and smoky conditions. *Journal of the Illumination Engineering Society*, pages 69–83, Winter 1992.

[9] R. Cook and K.E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1:7–24, 1982.

[10] R.A. Dobbins, G.W. Mulholland, and N.P. Bryner. Comparison of a fractal smoke optics model with light extinction measurements. *Atmospheric Environment*, 28(5):889–897, 1994.

[11] E. Braun et al. Comparison of full scale fire tests and a computer fire model of several smoke ejection experiments. NIST Internal Report 4961, 1992.

[12] M. Faraday. *The Chemical History of a Candle*. Larlin, 1978.

[13] R.J. Ferek, P. V. Hobbs, J.A. Herring, K.K. Laursen, and R.E. Weiss. Chemical composition of emissions from the kuwait oil fires. *Journal of Geophysical Research*, 97(D13):14483–14489, 1992.

[14] R.W. Fox, , and A.T. McDonald. *Introduction to Fluid Mechanics*. Wiley, 1973.

[15] G.Y. Gardner. Visual simulation of clouds. In *Proceedings of Siggraph 1985*, pages 297–303. ACM SIGGRAPH, 1985.

[16] A.G. Gaydon and H.G. Wolfhard. *Flames, Their Structure, Radiation and Temperature*. Chapman and Hall, 1979.

[17] H.E. Gerber and E.E. Hindman. *Light Absorption by Aerosol Particles*. Spectrum Press, 1982.

[18] S. Haas and G. Sakas. Methods for efficient sampling of arbitrary distributed volume densities. In *Proceedings of the Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pages 215–227. INRIA-IRISA, 1990.

[19] X.D. He, K.E. Torrance, F.X. Sillion, and D.P. Greenberg. A comprehensive physical model for light reflection. In *Proceedings of Siggraph 1991*, pages 175–186. ACM SIGGRAPH, 1991.

[20] D.W. Hoock. Modeling time-dependent obscuration for simulated imaging of dust and smoke clouds. In *Characterization, Propagation and Simulation of Sources and Backgrounds*, pages 164–175. SPIE, 1991.

[21] W.R. Cofer III and et al. Kuwaiti oil fires: Compositions of source smoke. *Journal of Geophysical Research*, 97(D13):14521–14525, 1992.

[22] M. Inakage. A simple model of flames. In *Proceedings of Computer Graphics International*, pages 71–81, 1989.

[23] F.P. Incropera and D.P. DeWitt. *Fundamentals of Heat Transfer.* Wiley, 1981.

[24] J.T. Kajiya and B.P. Von Herzen. Ray tracing volume densities. In *Proceedings of Siggraph 1984*, pages 165–174. ACM SIGGRAPH, 1984.

[25] F.X. Kneizys, E.P. Shettle, G.P. Anderson, L.W. Abreu, J.H. Chetwynd, J.E.A. Selby, S.A. Cloug, and W.O. Gallery. *LOWTRAN 7 COMPUTER CODE : USER'S MANUAL AFGL-TR-88-0177*. Air Force Geophysics Laboratory, Hanscom AFB, MA, 1988.

[26] L.D. Landau and E.M. Lifshitz. *Fluid Mechanics.* Pergamon Press, 1989.

[27] E. Languénou, K. Bouatouch, and M. Chelle. Global illumination in presence of participating media with general properties. In *Proceedings of the 5th Eurographics Workshop on Rendering*, 1994.

[28] M. Lesieur. *Turbulence in Fluids*. Kluwer, 1990.

[29] C.S. Li, F.T. Jenq, and W.H. Lin. Field characterization of submicron aerosols from indoor combustion sources. *Journal of Aerosol Science*, 23(S1):S547–S550, 1992.

[30] K.N. Liou. *Radiation and Cloud Processes in the Atmosphere.* The Oxford University Press, New York, 1992.

[31] N. Max. Efficient light propagation for multiple anisotropic volume scattering. In *Proceedings of 5th Eurographics Workshop on Rendering*, 1994.

[32] T. Nishita, Y. Miyawaki, and E. Nakamae. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. In *Proceedings of Siggraph 1987*, pages 303–308. ACM SIGGRAPH, 1987.

[33] H. Rushmeier. *Realistic Image Synthesis for Scenes with Radiatively Participating Media*. PhD thesis, The Sibley School of Mechanical and Aerospace Engineering, Cornell University, 1988.

[34] H. Rushmeier and K.E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. In *Proceedings of Siggraph 1987*, pages 293–302. ACM SIGGRAPH, 1987.

[35] D.G. Shepherd. *Elements of Fluid Mechanics.* Harcourt, Brace and World, 1965.

[36] R. Siegel and J. Howell. *Thermal Radiation Heat Transfer*. Hemisphere Publishing Corporation, 1981.

[37] J. Stam and E. Fiume. Turbulent wind fields for gaseous phenomena. In *Proceedings of Siggraph 1993*, pages 369–376. ACM SIGGRAPH, 1993.

[38] H. Tennekes and J.L. Lumley. *A First Course in Turbulence*. The MIT Press, Cambridge, MA, 1972.

[39] H.C. van de Hulst. *Light Scattering by Small Particles.* Dover, 1981.

[40] W.H. White, D.J. Moore, and J.P. Lodge, editors. *Proceedings of the Symposium on Plumes and Visibility: Measurement and Model Components*, 1980. in a special issue of *Atmospheric Environment*, vol. 15,1981.

Monte Carlo Methods in Rendering

Peter Shirley
University of Utah

Revised May 15, 1996 and April 15, 1998. The original version of this paper appeared as: Shirley, P. 1994. "Hybrid Radiosity/Monte Carlo Methods," ACM SIGGRAPH '94 Advanced Topics in Radiosity Course Notes, Chapter 11, pp. 1–24.

1 Introduction

Monte Carlo methods refer to any method that uses averages of random computations to get an approximate answer to a problem. In computer graphics Monte Carlo techniques can be used to perform radiosity calculations and can be used in distribution ray tracing for effects such as soft shadows and motion blur. These notes serve as an introduction to the tools of Monte Carlo, but a broader treatment on Monte Carlo methods for rendering can be found in Glassner's recent two-volume book [14].

In these notes I will cover the basics of both *Monte Carlo simulation*, where a physical system is modeled, *Monte Carlo integration*, where random numbers are used to approximate integrals, and *Quasi-Monte Carlo integration*, where non-random numbers are used. This discussion will cover the general techniques, and will use global illumination problems as examples.

One appeal of using Monte Carlo methods is that they are easy to design and use. However, it is not so easy to design a *good* Monte Carlo method, where the computation can be completed to the desired accuracy relatively quickly. Here both cleverness and some analytic skills are required. Fortunately, the analytic skills are fairly narrow in scope, so many of them can be covered in this short tutorial. A more formal discussion of Monte Carlo simulation can be found in the neutron transport literature (e.g. [53]) and an extremely current survey of Monte Carlo integration for practical applications can be found in the survey article by Spanier and Maize [54].

2 Background and Terminology

Before getting to the specifics of Monte Carlo techniques, we need several definitions, the most important of which are *continuous random variable*, *probability density function*, *expected value*, and *variance*. This section is meant as a review, and those unfamiliar with these terms should consult an elementary probability theory book (particularly the sections on continuous, rather than discrete, random variables).

Loosely speaking, a *continuous random variable* x is a scalar or vector quantity that 'randomly' takes on some value from a continuous space S, and the behavior of x is entirely described by the distribution of values it takes. This distribution of values can be quantitatively described by the *probability density function*, p, associated with x (the relationship is denoted $x \sim p$). If x ranges over a space S, then the probability that x will take on a value in some region $S_i \subset S$ is given by

the integral:

$$Prob(x \in S_i) = \int_{S_i} p(x) d\mu \quad (p : S \to \mathcal{R}^1). \tag{1}$$

Here $Prob(\textit{event})$ is the probability that *event* is true, so the integral is the probability that x takes on a value in the region S_i. The measure μ is the measure on our probability space. In graphics S is often an area ($d\mu = dA = dxdy$), or a set of directions (points on a unit sphere: $d\mu = d\omega = \sin\theta d\theta d\phi$). Loosely speaking, the probability density function describes the relative likelihood of a random variable taking a certain value; if $p(x_1) = 6.0$ and $p(x_2) = 3.0$, then a random variable with density p is twice as likely to have a value "near" x_1 than it it to have a value near x_2. The density p has two characteristics:

$$p(x) \geq 0 \quad \text{(Probability is nonnegative)}, \tag{2}$$

$$\int_S p(x) d\mu = 1 \quad (\text{Prob}(x \in S) = 1). \tag{3}$$

As an example, the *canonical* random variable ξ takes on values between zero (inclusive) and one (non-inclusive) with uniform probability (here *uniform* simply means each value for ξ is equally likely). This implies that:

$$f(\xi) = \begin{cases} 1 & \text{if } 0 \leq \xi \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

The space over which ξ is defined is simply the interval $[0, 1)$. The probability that ξ takes on a value in a certain interval $[a, b] \in [0, 1)$ is:

$$Prob(a \leq \xi \leq b) = \int_a^b 1 dx = b - a.$$

As an example, a two dimensional random variable α is a uniformly distributed random variable on a disk of radius R. Here *uniformly* means uniform with respect to area, e.g., the way a bad dart player's hits would be distributed on a dart board. Since it is uniform, we know that $p(\alpha)$ is some constant. From Equation 3, and the fact that area is the appropriate measure, we can deduce that $p(\alpha) = 1/(\pi R^2)$. This means that the probability that α is in a certain subset S_1 of the disk is just:

$$Prob(\alpha \in S_1) = \int_{S_1} \frac{1}{\pi R^2} dA.$$

This is all very abstract. To actually use this information we need the integral in a form we can evaluate. Suppose S_i is the portion of the disk closer to the center than the perimeter. If we convert to polar coordinates, then α is represented as a (r, θ) pair, and S_1 is where $r < R/2$. Note that just because α is uniform does not imply that $theta$ or r are necessarily uniform (in fact, $theta$ is, and r is not unifrom). The differential area dA becomes $r\, dr\, d\theta$. This leads to:

$$Prob(r < \frac{R}{2}) = \int_0^{2\pi} \int_0^{\frac{R}{2}} \frac{1}{\pi R^2} r\, dr\, d\theta = 0.25.$$

The average value that a real function f of a one dimensional random variable will take on is called its *expected value*, $E(f(x))$:

$$E(f(x)) = \int_S f(x) p(x) d\mu.$$

The expected value of a one dimensional random variable can be calculated by letting $f(x) = x$. The expected value has a surprising and useful property: the expected value of the sum of two random variables is the sum of the expected values of those variables:

$$E(x + y) = E(x) + E(y),$$

for random variables x and y. Since functions of random variables are themselves random variables, this linearity of expectation applies to them as well:

$$E(f(x) + g(y)) = E(f(x)) + E(g(y)).$$

An obvious question is whether this property hold if the random variables being summed are correlated (variables that are not correlated are called *indepedent.* This linearity property in fact does hold *whether or not* the variables are independent! Since the sum of two random variables is itself a random variable, this principle generalizes. As an example of expectation, consider random points on the disk of radius R. What is the expected distance r from the center of the disk of radius R?

$$E(r) = \int_0^{2\pi} \int_0^R \left(\frac{1}{\pi R^2} r \right) r \, dr \, d\theta = \frac{2R}{3}.$$

The *variance*, $var(x)$, of a one dimensional random variable is the expected value of the square of the difference between x and $E(x)$:

$$var(x) = E([x - E(x)]^2).$$

Some algebraic manipulation can give the non-obvious expression:

$$var(x) == E(x^2) - [E(x)]^2 .$$

The expression $E([x - E(x)]^2)$ is more useful for thinking intuitively about variance, while the algebraically equivalent expression $E(x^2) - [E(x)]^2$ is usually convenient for calculations. The variance of a sum of random variables is the sum of the variances *if the variables are independent*. This summation property of variance is one of the reasons it is frequently used in analysis of probabilistic models. The square root of the variance is called the *standard deviation*, σ, which gives some indication of expected absolute deviation from the expected value.

Many problems involve sums of independent random variables x_i, where the variables share a common density f. Such variables are said to be *independent identically distributed* random variables. When the sum is divided by the number of variables, we get an estimate of $E(x)$:

$$E(x) \approx \frac{1}{N} \sum_{i=1}^{N} x_i.$$

As N increases, the variance of this estimate decreases. We want n to be large enough that we have confidence that the estimate is "close enough". However, there are no sure things in Monte Carlo; we just gain statistical confidence that our estimate is good. To be sure, we would have to have $n = \infty$. This confidence is expressed by *Law of Large Numbers*:

$$Prob \left[E(x) = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} x_i \right] = 1.$$

3 Monte Carlo Simulation

For some physical processes, we have statistical models of behavior at a microscopic level from which we attempt to derive an analytic model of macroscopic behavior. For example, we often think of a luminaire (a light emitting object) as emitting a very large number of random photons (really pseudo-photons that obey geometric, rather than physical, optics) with certain probability density functions controlling the wavelength and direction of the photons. From this a physicist might use statistics to derive an analytic model to predict how the luminaire distributes its energy in terms of the directional properties of the probability density functions. However, if we are not interested in forming a general model, but instead want to know about the behavior of a particular luminaire in a particular environment, we can just numerically simulate the behavior of the luminaire. To do this we computationally "emit" photons from the luminaire and keep track of where the photons go. This simple method is from a family of techniques called *Monte Carlo Simulation* and can be a very easy, though often slow, way to numerically solve physics problems. In this section simulation techniques are discussed, and methods for improving their efficiency are presented.

The first thing that you might try in generating a highly realistic image is to actually track simulated photons until they hit some computational camera plane or were absorbed. This would be very inefficient, but would certainly produce a correct image, although not necessarily while you were alive. In practice, very few Monte Carlo simulations model the full physical process. Instead, an *analog* process is found that is easier to simulate, but retains all the *important* behavior of the original physical process. One of the difficult parts of finding an analog process is deciding what effects are important.

An analog process that is almost always employed in graphics is to replace photons with set wavelengths with power carrying beams that have values across the entire spectrum. If photons are retained as an aspect of the model, then an obvious analog process is one where photons whose wavelengths are outside of the region of spectral sensitivity of the film do not exist.

Several researchers (e.g. [3]) have used Monte Carlo simulation of a simple analog of optics, where only Lambertian and specular surfaces are used. A lambertian surface is one with several simple properties. First, its radiance at any wavelength does not vary with viewing angle. Second, this radiance varies linearly according to the total incident power per unit area and the reflectance of the surface. Quantitatively this can be written:

$$L(\lambda) = \frac{\rho_d(\lambda)\Phi_{incoming}(\lambda)}{\pi A} \tag{4}$$

where $L(\lambda)$ is the spectral radiance at wavelength λ, $\rho_d(\lambda)$ is the reflectance of the surface at λ, $\Phi_{incoming}(\lambda)$ is the incident power per unit wavelength at λ, and A is the area of the surface being illuminated. What makes the Lambertian surface attractive is that if we can figure out how much light is hitting it (irrespective of where the light comes from), then we know its radiance for all viewing directions. Note that ρ_d is the reflectance, not the BRDF, of the surface. The BRDF is a constant function with value ρ_d/π.

As an example of an analog process, the *illumination ray tracing* of Arvo [3] assumed photons traveled as bundles with a spectral distribution. He further assumed that these bundles were attenuated when reflecting from a specular surface. Like almost all graphics programs, his also assumed that the optical properties of the scene were constant within the time interval the picture represented, and that this time interval was very large relative to the speed it takes light to travel any distances in the scene. This last assumption, usually taken for granted, makes it possible to treat light as moving instantaneously within our programs. Finally, Arvo assumed that diffuse surfaces can be broken

into zones whose radiances are described by the power incident to them (i.e. they obey Equation 4 and are constant within a small neigborhood defined by the illumination pixel).

These assumptions allowed Arvo to trace power-carrying rays and mark each zone with the accumulated power. Once the simulation was over, the radiance of each zone could be calculated using Equation 4. Although we often think of this as being a brute force physical simulation, it is important to remember that this is really the simulation of an analog process where all wavelengths follow the same paths, and time dependent behavior can be ignored.

The trickiest part of implementing Arvo's simulation method is tracking the power through the environment. A natural choice for tracking the power is to use ray tracing. However, it is not so obvious how many rays to send, or where to send them. This question has been examined in a fairly sophisticated way in [19], but even for a simple implementation the answer is non-obvious. The number of rays that must be sent is "enough". This depends on how much noise is acceptable in an image, and how small the zones are. In [47] it is argued that the number of rays should be linearly proportional to the number of zones, so doubling the number of zones implies that the number of rays should also be doubled. A visual example of this argument is shown in Figure 1 where an environment with four times as many zones seems to require four times as many rays for the same level of accuracy as the environment with fewer zones[1]. The other detail, where the rays should be sent, is easier. The rays should be generated randomly with the same distribution as the emitted power of the luminaire. Generating rays sets with such directional distributions is discussed in Appendix A. The rays should also be sent from points distributed on the surface of the luminaire. To do this, first choose a random point on the luminaire surface, and then choose a random direction based on the surface normal at that point.

Arvo's method can be extended to a radiosity [15] method by letting the Lambertian zones interreflect light [33, 1, 2, 44]. This is really just a ray tracing variant of the progressive refinement radiosity method [8]. In this method, reflectance ($\rho_{d,i}$) and emitted power ($\Phi_{e,i}$) of the ith zone are known, and the reflected power ($\Phi_{r,i}$) ($\Phi_{r,i} = \Phi_{incoming,i}$) is unknown. If we solve for $\Phi_{r,i}$, then we can find Φ_i, the total power coming from the ith patch.

Once the total power of each patch is found, it can be converted to radiance using Equation 4. These radiance values can then be interpolated to form a smooth appearance [9].

We first set our estimate of Φ_i to be $\Phi_{e,i}$ for all i. For each surface i that has non-zero $\Phi_{e,i}$, we can shoot a set of n_i energy packets each carrying a power of Φ_i^e/n_i. When a packet with power Φ hits a surface j, we can add $\rho_{d,j}\Phi$ for our estimate of Φ_j, and reflect a new energy packet with power $\rho_{d,j}\Phi$. This energy packet will bounce around the environment until it is depleted to a point where truncation is used. This basic energy packet tracing technique has been used in Heat Transfer [21, 13, 56], Illumination Engineering [55], and Physics [53, 23].

This method, which I call *reflection simulation* (see Figure 2), is problematic in that each reflection is followed by a ray intersection test to find the next surface hit. The later reflections will carry a relatively small amount of power, so tracing these later rays is somewhat wasteful in the sense that we have bad 'load-balancing': some rays do more work than others. One solution to this problem is to use "Russian roulette" and keep all particles with the same power by probabilistically absorbing them according to the albedo of the surface [5, 37]. Another solution to this problem of low energy particles is to replace the reflection model with an analog model where light is absorbed and immediately reemitted (after attenuation by the reflectance) (see Figure 3). A scene where light

[1]The number of rays sent can be thought of as the number of photons tracked in a certain time interval. The number of rays will be proportional to the time a "shutter" is open. Once the "exposure" is long enough, the noise will not be objectionable.

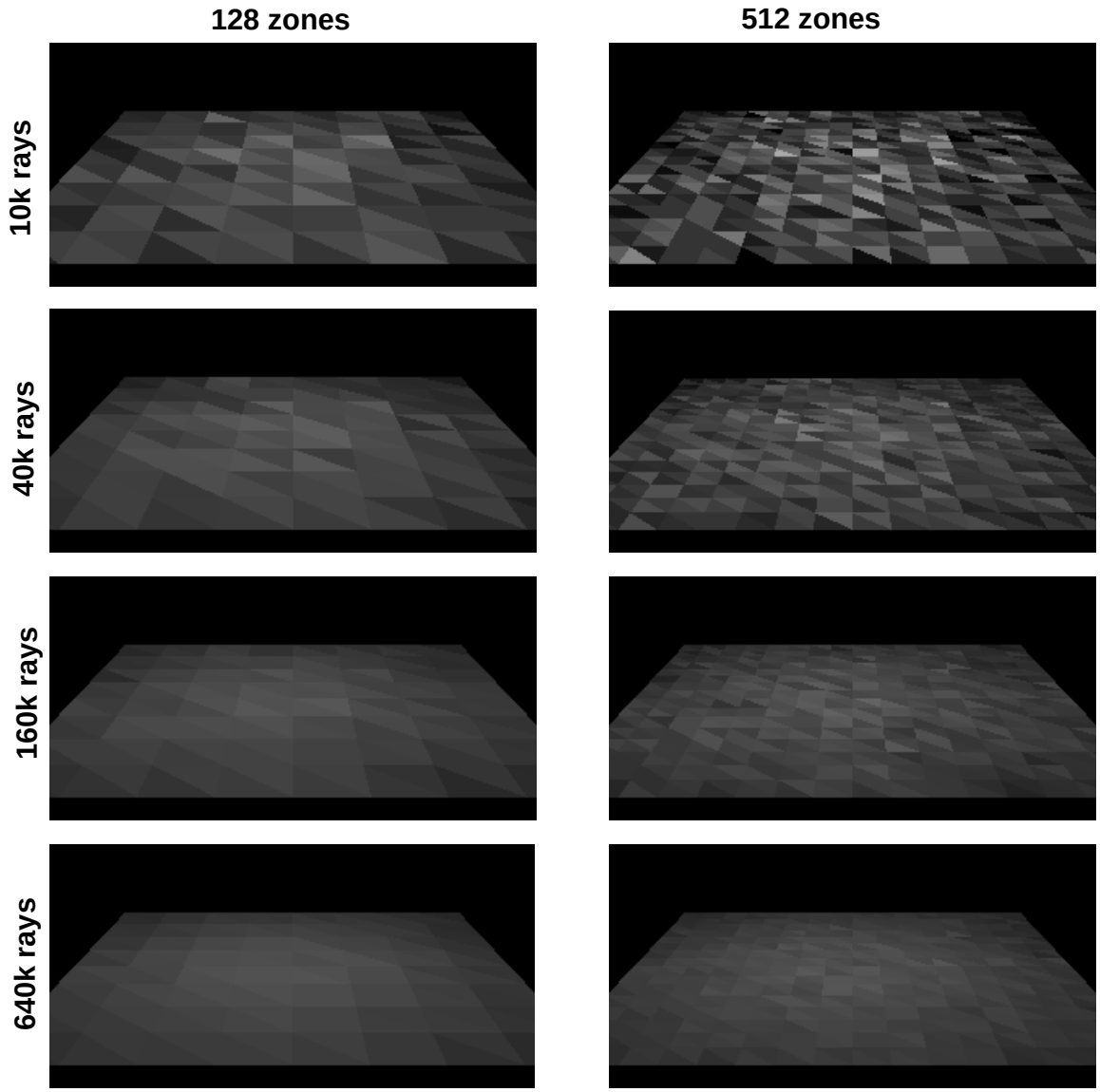


Figure 1: Noise reduction as the number of energy bundles increases. Note that the number of bundles needed is approximately inversely related to the surface area of each zone.

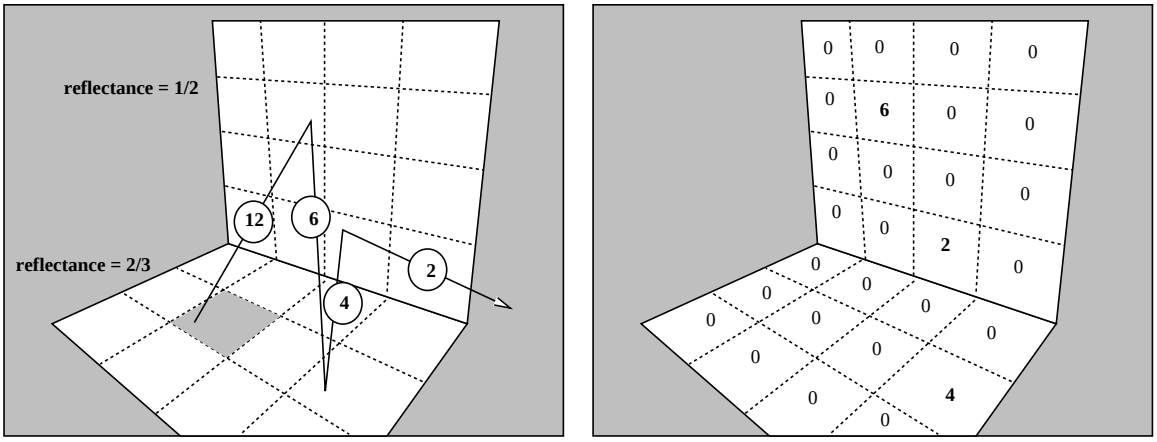


Figure 2: Reflection Simulation. The patch on the floor is a luminaire and emits a "photon" with 12 watts of power. Each reflection damps some of the power and scatters the photon according to a diffuse (cosine) distribution. On the right is the reflected power from each patch after the photon leaves the environment. The emitted power is also stored for each patch but is not shown.

is absorbed and reemitted in this way looks similar to a scene where light is reflected, so solving for the transport in either model will hopefully yield a similar solution. The difference is that now light may strike one side of a zone and later be reemited on the other side of the zone. This can give rise to objectional artifacts if the zone is partially in a dark area and partially in a light area[2] (e.g. goes under a door between the outside and a dark room). To solve for this absorb and reemit model, we can again send power in bundles from light sources. When a bundle carrying power Φ hits a surface j, the absorbed power that will later be reemitted by surface j can be scaled by $\rho_{d,j}\Phi$. After each light source emits its power, reflective surfaces can, in turn, emit their absorbed power. The efficiency of this method is best if surfaces with the greatest amount of power send their power first.

The reason that we have the freedom to let the zones emit in any order we choose is that our analog has lost its time dependence. We are lucky the speed of light is so fast! There are two points which are crucial to the implementation of this progressive refinement method. The first is that the number of rays emitted from a certain zone is proportional to the power being emitted in that iteration (each ray carries approximately the same amount of power). The other is that, unlike in [8], the zone with the most power is not searched for, or the time complexity of the method will increase from $O(N \log N)$ to $O(N^2)$, where N is the number of zones [47]. This problem can be avoided if a heap or similar structure is used to make the search for maximum $O(\log N)$ rather than $O(N)$. A more detailed discussion of the implementation of Monte Carlo radiosity can be found in [46].

Recently, Neumann et al. have compared various Monte Carlo strategies for radiosity on predefined meshes [36]. Interestingly, the straightforward particle tracing with Russian roulette converges faster inm their tests than "absorb and reemit", and that "absorb and reemit" can be improved by viewing it in a linear algebra context.

The biggest problem with these Monte Carlo radiosity methods is that small zones will be undersampled and will have large errors, or enough rays will be sent that the large area zones

[2]Thanks to Dani Lischinski for pointing this out. Earlier versions of this document said that absorb and reemit was asymptotically equivalent to the photon tracking model.

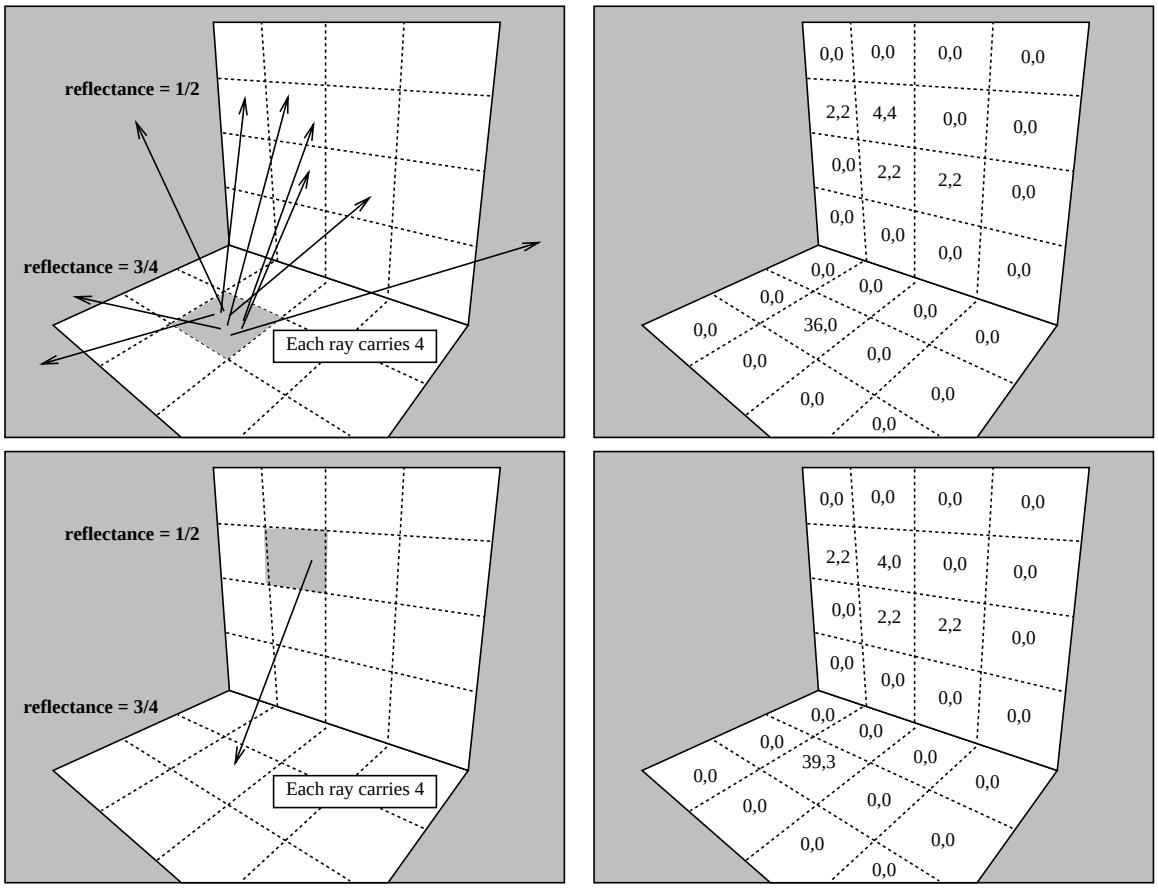


Figure 3: Absorb and reemit. The patch on the floor has 36 units of power that it has not distributed. Each patch has two numbers, the total reflected power (left of pair), and the power that still needs to be sent (right of pair).

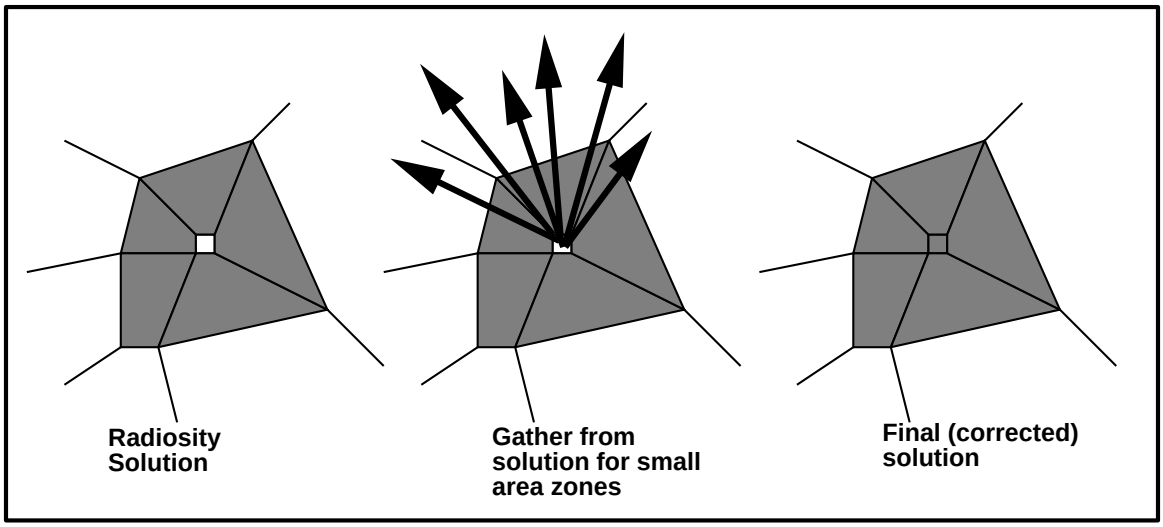


Figure 4: Zones with small areas have their radiance recalculated more accurately in a postprocess. The arrows indicate the direction of rays sent into the environment to find energy sources and thus flow against the direction of light transport.

are oversampled. This is only a problem in scenes with a large range of zone areas, but this is not uncommon. One way to get around this problem (that I have not yet tried) is to do a "gather" on small zones in the scene after the first radiosity solution is done. The radiance of the zone is simply its reflectance times the average radiance "seen" by the gather rays provided they are sent in a cosine distribution. This idea is illustrated in Figure 4.

This simple simulation method could also be used for diffuse transmission, in a manner similar to that of Rushmeier and Torrance [41]. Some of the simulation techniques discussed earlier can be extended to non-diffuse reflection types. The most important application is to scenes that include specular surfaces, but glossy surfaces are sometimes desirable too.

The simplest method of including specular reflection in a radiosity calculation is the *image method* [59, 41]. In the image method, a specular surface is replaced by a hole into a virtual environment. This method works only for planar mirrors, but performs very well for environments that have one important specular surface like a mirror or highly polished floor. Malley extended his Monte Carlo power transport method to account for zonal transport by specular surfaces [33]. He did this by allowing power carrying rays to reflect off specular surfaces as shown in Figure 5. The colors of specular surfaces can be determined in the viewing phase by standard ray tracing. Sillion and Puech used a similar technique to account for specular reflection, and included subdivision strategies for sampling more heavily where ray paths diverged [52].

Any non-diffuse reflectors can have zonal values, as long as each incoming power packet adds to a power *distribution function* that will be reemitted. In the viewing stage, this distribution can be queried with results depending on viewer position. The distribution functions could be stored in a Hemicube as done by Immel et al. [22], as spherical harmonics as done in [6, 51], or in hemispherical tables as done in [16, 42, 48]. These latter methods use Monte Carlo by generating outgoing power rays according to the shape of the unemitted power function as shown in Figure 6. My experience has been that non-diffuse radiosity does not work well for near mirrors because more zones are needed (the surfaces have detail visible in them) and each zone needs a larger table to represent the complicated outgoing power distribution.

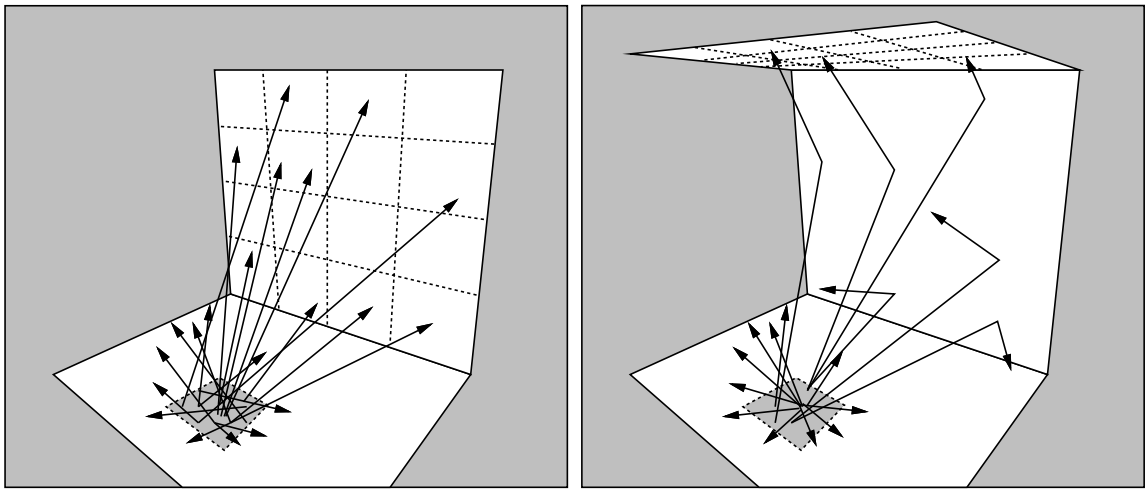


Figure 5: Monte Carlo emission of energy with and without specular reflection. On the left, energy is transported directly between diffuse zones. On the right, the vertical wall is a mirror, and light that hits it is reflected until it hits a diffuse surface.

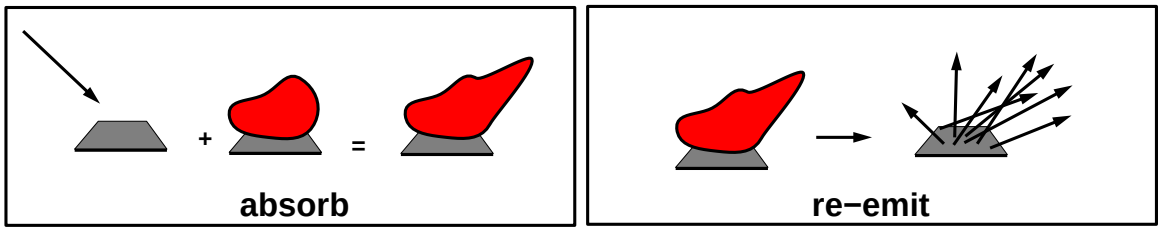


Figure 6: Absorb and re-emit strategy requires directional distribution at each zone and a way to directionally shoot power to directions where the accumulated distribution is large.

4 Monte Carlo Integration

In this section the basic Monte Carlo solution methods for definite integrals are outlined. These techniques are then straightforwardly applied to certain integral problems. All of the basic material of this section is also covered in several of the classic Monte Carlo texts. This section differs by being geared toward classes of problems that crop up in Computer Graphics. Readers interested in a broader treatment of Monte Carlo techniques should consult one of the classic Monte Carlo texts [18, 50, 17, 66].

From Section 2 we saw that for a function f and a random variable $x \sim p$, we can approximate the expected value of $f(x)$ by a sum:

$$E(f(x)) = \int_{x \in S} f(x)p(x)d\mu \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i). \tag{5}$$

Because the expected value can be expressed as an integral, the integral is also approximated by the sum. The form of Equation 5 is a bit awkward; we would usually like to approximate an integral of a single function g rather than a product fp. We can get around this by substituting $g = fp$ as the integrand:

$$\int_{x \in S} g(x)d\mu \approx \frac{1}{N}\sum_{i=1}^{N} \frac{g(x_i)}{p(x_i)}. \tag{6}$$

For this formula to be valid, p must be positive where g is nonzero.

So to get a good estimate, we want as many samples as possible, and we want the g/p to have a low variance (g and p should have a similar shape). Choosing p intelligently is called importance sampling, because if p is large where g is large, there will be more samples in important regions. Equation 5 also shows the fundamental problem with Monte Carlo integration: *diminishing return*. Because the variance of the estimate is proportional to $1/N$, the standard deviation is proportional to $1/\sqrt{N}$. Since the error in the estimate behaves similarly to the standard deviation, we will need to quadruple N to halve the error.

Another way to reduce variance is to partition S, the domain of the integral, into several smaller domains S_i, and evaluate the integral as a sum of integrals over the S_i. This is called stratified sampling. Normally only one sample is taken in each S_i (with density p_i), and in this case the variance of the estimate is:

$$var\left(\sum_{i=1}^{N} \frac{g(x_i)}{p_i(x_i)}\right) = \sum_{i=1}^{N} var\left(\frac{g(x_i)}{p_i(x_i)}\right). \tag{7}$$

It can be shown that the variance of stratified sampling is never higher than unstratified if all strata have equal measure:

$$\int_{S_i} p(x)d\mu = \frac{1}{N}\int_{S} p(x)d\mu.$$

The most common example of stratified sampling in graphics is jittering for pixel sampling [12].

As an example of the Monte Carlo solution of an integral I set $g(x)$ to be x over the interval (0, 4):

$$I = \int_0^4 x\,dx = 8. \tag{8}$$

The great impact of the shape of the function p on the variance of the N sample estimates is shown in Table 1. Note that the variance is lessened when the shape of p is similar to the shape of g. The

method	*sampling function*	*variance*	*samples needed for standard error of 0.008*
importance	$(6-x)/(16)$	$56.8N^{-1}$	887,500
importance	$1/4$	$21.3N^{-1}$	332,812
importance	$(x+2)/16$	$6.3N^{-1}$	98,437
importance	$x/8$	0	1
stratified	1/4	$21.3N^{-3}$	70

Table 1: Variance for Monte Carlo Estimate of $\int_0^4 x\,dx$

variance drops to zero if $p = g/I$, but I is not usually known or we would not have to resort to Monte Carlo. One important principle illustrated in Table 1 is that stratified sampling is often *far* superior to importance sampling. Although the variance for this stratification on I is inversely proportional to the cube of the number of samples, there is no general result for the behavior of variance under stratification. There are some functions where stratification does no good. An example is a white noise function, where the variance is constant for all regions. On the other hand, most functions will benefit from stratified sampling because the variance in each subcell will usually be smaller than the variance of the entire domain.

4.1 Quasi-Monte Carlo Integration

Although distribution ray tracing is usually phrased as an application of Equation 6, many researchers replace the ξ_i with more evenly distributed (quasi-random) samples (e.g. [11, 34]). This approach can be shown to be sound by analyzing decreasing error in terms of some discrepancy measure [67, 65, 34, 43] rather than in terms of variance. However, it is often convenient to develop a sampling strategy using variance analysis on random samples, and then to turn around and use non-random, but equidistributed samples in an implementation. This approach is almost certainly correct, but its justification and implications have yet to be explained.

For example, when evaluating a one dimensional integral on $[0,1]$ we could use a set of N uniformly random sample points $(x_1, x_2, \cdots, x_N)$ on $[0,1]$ to get an approximation:

$$\int_0^1 f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i).$$

Interestingly, we can replace the points $(x_1, x_2, \cdots, x_N)$ with a set of non-random points $(y_1, y_2, \cdots, y_N)$, and the approximation will still work. If the points are too regular, then we will have aliasing, but having correlation between the points (e.g. using one dimension Poisson disk sampling), does not invalidate the estimate (merely the Monte Carlo argument used to justify the approximation!). In some sense, this quasi-Monte Carlo method can be thought of as using the equidistributed points to estimate the height of f. This does not fit in with the traditional quadrature approaches to numerical integration found in most numerical analysis texts (because these texts focus on one-dimensional problems), but is no less intuitive once you are used to the idea.

4.2 Multidimensional Monte Carlo Integration

Applying Equation 6 to multidimensional integrals is straightforward, except that choosing the multidimensional sampling points can be more involved than in the one dimensional case. More

specifics on this can be found in Appendix A.

As an example in two dimensions, suppose we want to integrate some function f on the origin centered square $[-1,1]^2$. This can be written down as a integral over a single two dimensional variable $\mathbf{x}$:

$$I = \int_{[-1,1]^2} f(\mathbf{x}) dA.$$

Applying Equation 6 to this gives us:

$$I \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)},$$

where each $\mathbf{x}_i$ is a two dimensional point distributed according to a two dimensional density p. We can convert to more explicit Cartesian coordinates and have a form we are probably more comfortable with:

$$I = \int_{y=-1}^{1} \int_{x=-1}^{1} f(x,y) dx dy \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i, y_i)}{p(x_i, y_i)}.$$

This is really no different than the form above, except that we see the explicit components of $\mathbf{x}_i$ to be (x_i, y_i).

If our integral is over the disk of radius R, nothing really changes, except that the sample points must be distributed according to some density on the disk. This is why Monte Carlo integration is relatively easy: once the sample points are chosen, the application of the formula is always the same.

For a more complicated example, we look at the four dimensional integral for the form factor between two surfaces S_1 and S_2:

$$F_{12} = \frac{1}{A_1} \int_{\mathbf{x}_1 \in S_1} \int_{\mathbf{x}_2 \in S_2} \frac{g(\mathbf{x}_1, \mathbf{x}_2) \cos\theta_1 \cos\theta_2 dA_1 dA_2}{\pi ||\mathbf{x}_1 - \mathbf{x}_2||^2}.$$

The sampling space is the four dimensional space $S_1 \times S_2$. A four dimensional point in this space is just an ordered pair $(\mathbf{x}_1, \mathbf{x}_2)$, where $\mathbf{x}_1$ is a point on S_1 and $\mathbf{x}_2$ is a point on S_2. The simplest way to proceed is to choose our four dimensional sample point as a pair of uniformly random points, one from each surface. The probability density function for this is the constant $1/(A_1 A_2)$, because $A_1 A_2$ is the four dimensional volume of the space, and this value just enforces Equation 3. If we use only one sample we have the estimate:

$$F_{12} \approx A_2 \frac{g(\mathbf{x}_1, \mathbf{x}_2) \cos\theta_1 \cos\theta_2}{\pi ||\mathbf{x}_1 - \mathbf{x}_2||^2}.$$

A ray would be sent to evaluate the geometry term g. If many samples were taken, we could increase our accuracy. Notice that the shape of the surfaces was never explicitly used. This formula is valid whenever we have a method to choose random points from a shape!

4.3 Weighted Averages

We often have integrals that take the form of a strictly positive weighted average of a function:

$$I = \int_S w(x) f(x) d\mu.$$

where w is a weighting function with unit volume. To solve this by Equation 6, the optimal choice for the probability function is $p(x) = Cw(x)f(x)$, but as is often pointed out, this choice requires us to already know the value of I. Instead, people often either choose uniform p, or set $p(x) = w(x)$ [11, 38, 31].

An example of a weighted average often used is pixel filtering. The color of a pixel $I(i,j)$ can be expressed as an integral:

$$I(i,j) = \int_S w(\mathbf{p})L(\mathbf{p})dA. \tag{9}$$

where $\mathbf{p}$ is a point on the viewport (or filmplane if a camera model is used), $L(\mathbf{p})$ is the radiance seen through the viewport at $\mathbf{p}$, and S is the non-zero region of the filter function w.

Rewriting with the assumption that the same origin-centered weighting function is used for every pixel yields the estimator :

$$I(i,j) \approx \frac{1}{N}\sum_{k=1}^{N} \frac{w(x_k,y_k)L(i+0.5+x_k, j+0.5+y_k)}{p(x_k,y_k)}. \tag{10}$$

This assumes a coordinate system where a pixel (i,j) has unit area and is centered at $(i{+}0.5, j{+}0.5)$ as suggested by Heckbert [20].

Once a w is chosen for filtering, implementation is straightforward with p proportional to w provided that w is strictly positive (as it must be if negative pixel colors are disallowed). But how do we choose *non-uniform* random points? As discussed in Appendix A, sample points can be chosen uniformly from $[0,1]^2$ and then a warping transformation can be applied to distribute the points according to w [50, 43, 31].

For several practical and theoretical reasons [45] we have chosen the width 2 weighting function that is non-zero on $(x,y) \in [-1,1]^2$:

$$w(x,y) = (1-|x|)\,(1-|y|)\,. \tag{11}$$

We generate random points with density equal to w by applying a transformation to a uniform random pair $(r_1, r_2) \in [0,1]^2$. The transformed sample point is just $(t(r_1), t(r_2))$, where the transformation function t is:

$$t(u) = \begin{cases} -0.5 + \sqrt{2u} & \text{if } u < 0.5 \\ 1.5 - \sqrt{2(1-u)} & \text{if } u \geq 0.5 \end{cases}$$

An important detail is that we do not really use uniform (r_1, r_2), but instead use jittered or an otherwise better distributed set of points. After warping, we still have a better than random distribution.

Another example of a weighted average is the radiance of a point $\mathbf{x}$ on a Lambertian surface:

$$L(\mathbf{x}) = \rho_d(\mathbf{x}) \int_{\text{incoming } \psi'} \frac{1}{\pi} L(\mathbf{x}, \psi') \cos\theta d\omega'.$$

Where $L(\mathbf{x}, \psi')$ is the incoming radiance seen at point $\mathbf{x}$ coming from direction ψ'. Again, we might be more comfortable with the explicit form:

$$L(\mathbf{x}) = \rho_d(\mathbf{x}) \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\frac{\pi}{2}} \frac{1}{\pi} L(\theta,\phi) \cos\theta \sin\theta d\theta d\phi.$$

The $\sin\theta$ term arises because the measure is solid angle (area on the unit sphere: $d\omega = dA = \sin\theta d\theta d\phi$). To solve this we just need to choose a random direction ψ to sample with a distribution according to the density function $\cos\theta/\pi$. This gives the estimator:

$$L(\mathbf{x}) = \rho_d(\mathbf{x}) L(\mathbf{x}, \psi).$$

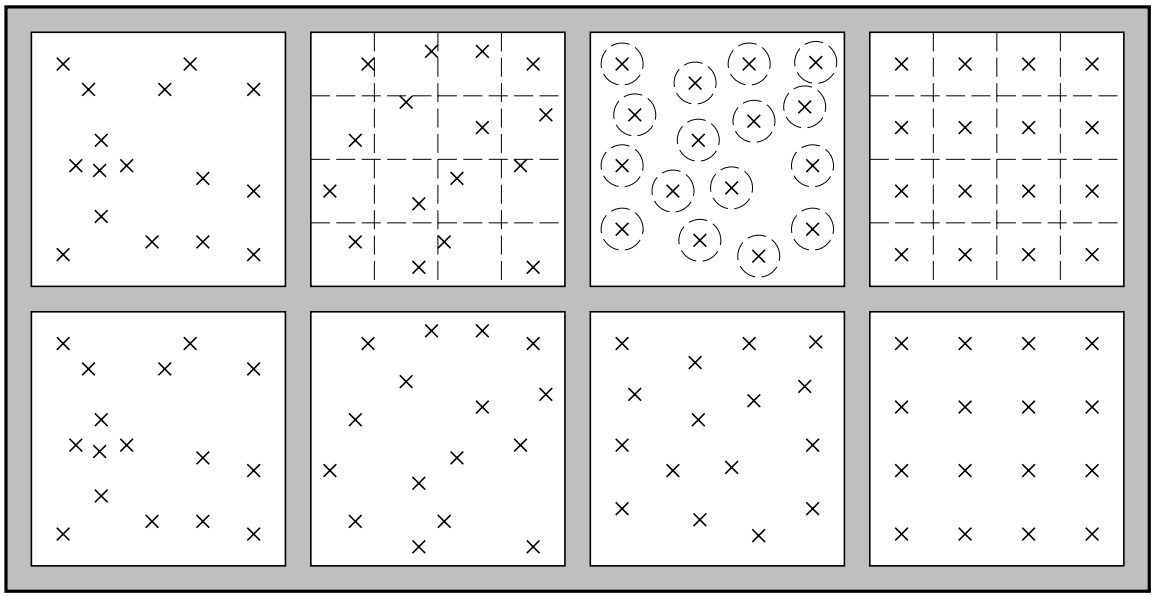


Figure 7: Random, Jittered, Dart-throwing, Regular.

This makes it easy to figure out the color of the ground in the midwest: it's the weighted average of the color of the sky times the reflectance of the ground!

4.4 Multidimensional Quasi-Monte Carlo Integration

Suppose we want to numerically estimate the value of an integral I on $[0,1]^2$:

$$I = \int_0^1 \int_0^1 f(x,y)dxdy.$$

For pure Monte Carlo we might use a set of uniform random points $(x_i, y_i) \in [0,1]^2$ and estimate I to be the average of $f(x_i, y_i)$. For stratified sampling we might partition $[0,1]^2$ into several equal-area rectangles and take one sample (x_i, y_i) in each rectangle and again average $f(x_i, y_i)$. Interestingly, we might also use "Poisson-disk" sampling to generate the points, or even just use points on a regular grid. No matter which of these point distributions (shown in Figure 7) we use, the estimate of I is the average of $f(x_i, y_i)$. Interestingly, only when we use one of the first two patterns are we doing Monte Carlo integration. With Poission-disk (dart-throwing), the samples are correlated, and in regular sampling they are deterministic.

As in the one-dimensional case, we can replace the random sample points with any set of samples that are in some sense uniform, and this is just quasi-Monte carlo integration. There is a rich literature on this topic, but Mitchell has indicated that the graphics community will not be able to find many useful answers there, because the patterns that are used in that literature are deterministic, which causes aliasing in images [35].

4.5 Direct Lighting

In this section the famous direct lighting calculation is discussed. Even if radiosity is used, it can often be used only for the indirect component and the direct component can be done using the machinery of this section.

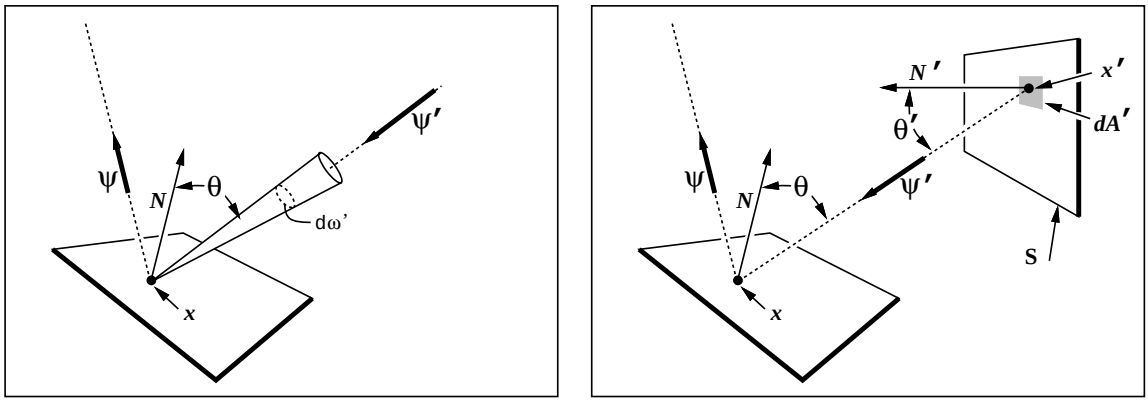


Figure 8: Definitions for the rendering equations.

The rendering equation can be written down in two basic ways. It can be written down in terms of all directions visible to $\mathbf{x}$ (as in [22]):

$$L(\mathbf{x}, \psi) = \int_{\text{incoming } \psi'} f_r(\mathbf{x}, \psi, \psi') L(\mathbf{x}, \psi') \cos\theta d\omega'. \tag{12}$$

where f_r is the BRDF, or it can written down as an integral over all surfaces (as in [25]):

$$L(\mathbf{x}, \psi) = \int_{\text{all } \mathbf{x}'} g(\mathbf{x}, \mathbf{x}') f_r(\mathbf{x}, \psi, \psi') L(\mathbf{x}', \psi') \cos\theta \frac{dA' \cos\theta'}{\|\mathbf{x}' - \mathbf{x}\|^2}. \tag{13}$$

When Equation 12 is used, we can view $f_r(\mathbf{x}, \psi, \psi') \cos\theta$ as a weighting function and sample according to it. Because there is some energy absorbed by a surface, this gives us the estimator:

$$L(\mathbf{x}, \psi) \approx R(\mathbf{x}, \psi) L(\mathbf{x}, \xi), \tag{14}$$

where ξ is a random direction with density proportional to $f_r(\mathbf{x}, \psi, \psi') \cos\theta$. The reflectivity term is simply:

$$R(\mathbf{x}, \psi) = \int_{\text{incoming } \psi'} f_r(\mathbf{x}, \psi, \psi') \cos\theta d\omega.$$

For an ideal specular surface, the ξ will always be the ideal reflection direction. For a dielectric, ξ can be chosen randomly between reflected and transmitted directions [5], or it can be split into two integrals as is done in a Whitted-style ray tracer [64]. For a diffuse surface, ξ will follow a cosine distribution: $p(\psi') = \cos\theta/\pi$.

When Equation 13 is used, the sampling takes place over all surfaces in the environment. In practice, only the direct lighting is calculated, so the integration space becomes all luminaire surfaces. This can be split into one integral for each surface [11], or can be viewed as a single sampling space [31, 49]. To simplify this discussion, we will assume only one luminaire, so the sampling space is just a single surface. Looking at Equation 13, an ideal estimator for diffuse luminaires would result if we sampled according to the density:

$$p(x') = C g(\mathbf{x}, \mathbf{x}') f_r(\mathbf{x}, \psi, \psi') \cos\theta \frac{\cos\theta'}{\|\mathbf{x}' - \mathbf{x}\|^2},$$

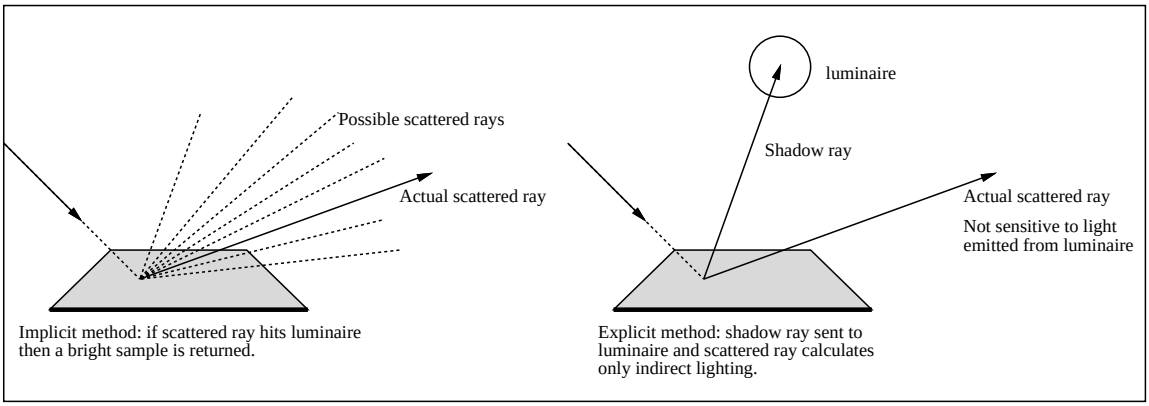


Figure 9: Implicit and explicit lighting calculation.

where C is a normalization constant. In practice, this isn't practical because the geometry term g and the BRDF f_r can be very difficult to characterize. Instead, many researchers [12, 25] sample uniformly within the solid angle subtended by the luminaire, which yields:

$$p(x') = C' \frac{\cos\theta'}{\|\mathbf{x}' - \mathbf{x}\|^2}. \tag{15}$$

This can be done for triangular luminaires [4], and for spherical luminaires [27, 61]. If Equation 15 is used to choose points on the luminaire, then radiance can be estimated to be:

$$L(\mathbf{x}, \psi) \approx g(\mathbf{x}, \mathbf{x}') f_r(\mathbf{x}, \psi, \psi') L(\mathbf{x}', \psi') \cos\theta\omega, \tag{16}$$

where ω is the total solid angle subtended by the luminaire as seen by $\mathbf{x}$.

We call the use of Equation 12 an *implicit* direct lighting calculation because any scattered ray that hits a luminaire will account for light from that luminaire. The use of Equation 13 is an *explicit* direct lighting calculation because each luminaire is explicitly queried using shadow rays (see Figure 9). Which should be used, an implicit or explicit direct lighting calculation? Clearly, the implicit method must be used for perfect mirrors, because that method implicitly evaluates the delta function BRDF. For a diffuse surface, the explicit method is usually used for direct lighting, and the implicit method is used only for indirect lighting [25, 63, 31]. To decide which method to use, variance should be analyzed, but the general rule is that specular surfaces should be dealt with using the implicit calculation and diffuse surfaces are treated explicitly.

If indirect lighting is to be added, then the surfaces that use the explicit direct lighting calculation can calculate indirect lighting implicitly with a scattered reflection ray [25]. This method, called *path tracing*, just recursively applies the direct lighting calculation and adds indirect lighting. If you implement this, be sure not to double count the indirect lighting!

5 Hybrid Methods

Many methods use some combination of view-dependent and view-independent methods. There are three basic approaches that have been used:

1. Generate a radiosity solution and view with ray tracing.

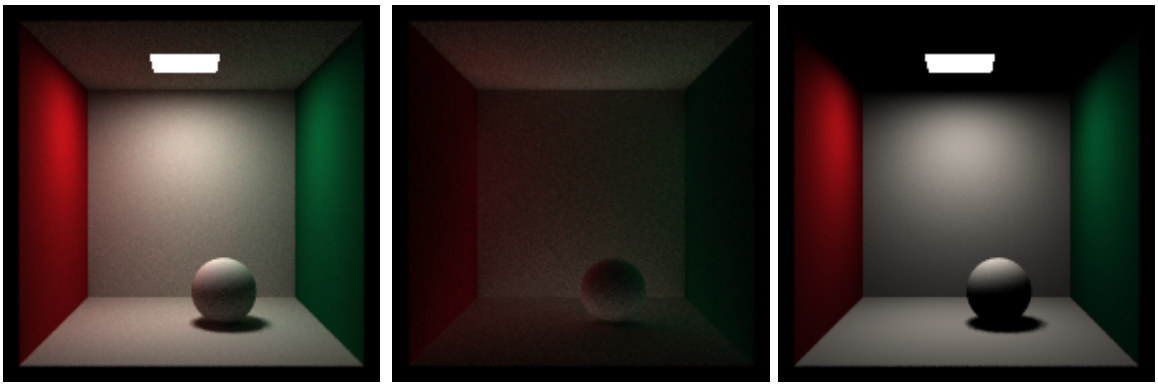


Figure 10: Combined, indirect, and direct lighting. Note that the the sharp shading changes are in the direct component

.

2. Generate a radiosity solution and use only for indirect lighting. Use ray tracing for direct lighting.

3. Generate a radiosity solution on a *low resolution* environment and use this in the viewing phase.

In method 1 the ray tracing is really just to accurately capture specular effects [59] and the radiosity phase my or may not include specular transport [33, 52] or directional diffuse transport [42, 48, 51]. Any problems with the meshing in high gradient areas will be very obvious in method 1, so some form of discontinuity meshing should be used [32].

In method 2 the fine detail caused by shadows (see Figure 10) is handled in the direct phase and the indirect lighting is handled by some precomputed values [44, 7, 29]. Ward's Radiance program [63, 62] is in the second family although the indirect information is calculated on the fly and cached, and the mesh is implicit.

In method 3, a zonal solution is carried out on a low-resolution version of the scene, and this is used as sources for gather phases at each pixel [40, 39, 28]. This in some sense is a generalization of the patch and elements approach [10]. The application of this technique and brute-force path tracing [25] is shown in Figure 11. The Rushmeier method ran eight times faster because it did not have to recursively fire rays. On complex scenes this advantage will only grow.

I am a fan of method 3 for many applications. I used to use method 2 (see [44]) but I found it cumbersome to have to mesh all diffuse objects. The beauty of method 3 is that it works even if the high resolution environment is an on-demand procedural model, it is easy to code, and that there are no smoothing issues. The radiosity solution does not have to look good! However, there are a number of open questions related to method 3:

- How should directional diffuse surfaces be handled?
- How should nearly specular surfaces be handled?
- How should caustics be handled?
- Should a hierarchy of various low-resolution environments be used?
- How should the low-resolution environment be created?

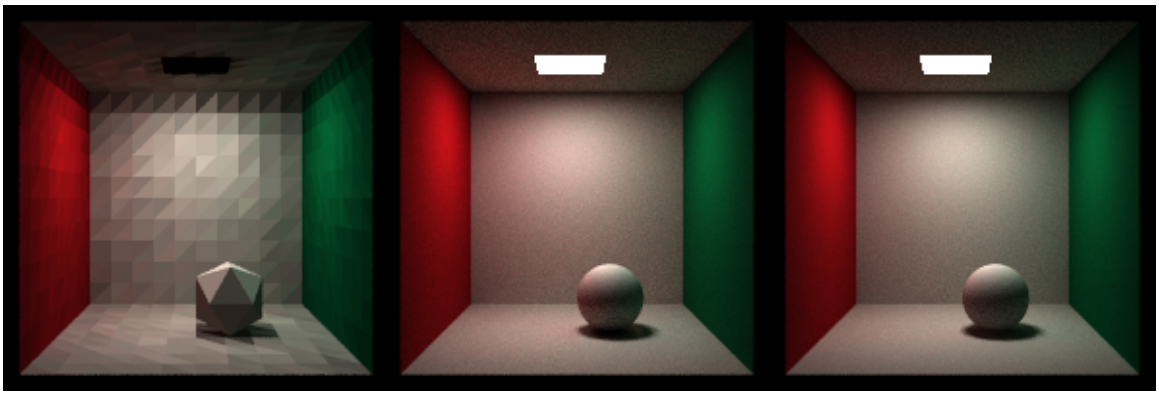


Figure 11: Left: Low-resolution radiosity solution. Middle: Rushmeier solution. Right: Path tracing.

6 Conclusion

I hope that this tutorial has revealed the elegance and simplicity of Monte Carlo methods. This elegance and simplicity allow the modeling and solution of many problems with very few assumptions. However, these benefits come with the price of long execution times. If you need speed, use other techniques, or supplement Monte Carlo techniques with other methods. A good example of this combined strategy is Ward's *Radiance* program described in the conference procedings [62].

Since these notes first appeared at SIGGRAPH '94, there has been much work in Monte Carlo rendering. Those that want to really delve into the subject should consult the dissertations of Eric Lafortune, Eric Veach, or Kurt Zimmerman. There are several others that have been written in the last several years but I haven't yet read them. I hope to keep up-to-date pointers on my own web page. Summarizing the big developments of late that are not treated well in the body of these notes:

Metropolis Algorithm. This algorithm [58] operates in path space and attempts to create a set of light transport paths that carry equal power density to the camera. It is a real bear to wrap your mind around, but is a really neat idea. I am working hard trying to do my own implementation of this and it is tough! To understand this algorithm, you have to really "dot your i's" on issues of measure and density. A good place to start before you read the Metropolis paper is the bidirectional path tracing work of Lafortune [30] and Veach [57].

Density Estimation. This algorithm [60] does a photon tracing phase and stores all interactions between photons and surfaces. On diffuse surfaces it looks at the pattern of photon hits and tries to infer (estimate) the continuous pattern of light (density). This algorithm is geared toward view-independent solutions of semi-complex diffuse scenes.

Photon Map. This algorithm [24] is similar to density estimation, except that photon positions and incident directions are used. This means that more storage is used but non-diffuse effects can be accounted for. This method can also be applied to participating media (se SIGGRAPH 98 proceedings).

When designing new Monte Carlo methods, we usually think in terms of variance reduction. Work by Arvo and Kirk [5, 26] has detailed that this can be a non-trivial and sometimes counter intuitive process. To add to the confusion, we usually use quasi-random sampling, so the variance calculations are only an approximation. In the end I find that developing a theory using straight Monte Carlo assumptions, and then adding to it using intuition works the best for me. The most common heuristic I use is that every sample should do about the same amount of work. This is

intuitively related to importance sampling, because the way to come close to this is to try to give every sample the same weight (load balancing).

The real key to a successful Monte Carlo method is the design of the probability density functions and stratification strategies used to generate the samples. This is where your efforts should be concentrated. There is a tendency to think that your work is done (and the computer's starts!) once you have chosen to use a Monte Carlo method, but the very freedom to choose any density function dooms us to look for a *better* choice!

7 Acknowledgments

Many people have helped me understand the material in these notes, and an incomplete list of those I owe thanks to is Willie Hunt, Kelvin Sung, Frederick Jansen, Greg Ward, Claude Puech, William Kubitz, Don Mitchell, James Kajiya, Holly Rushmeier, Andrew Glassner, Kenneth Chiu, Dani Lischinski, Jim Arvo, John Wallace, Atul Jain, Randy Bramley, and Changyaw Wang. Special thanks to Paul Heckbert, Steve Marschner, Sumant Pattanaik, and Eric Lafortune who found many errors in a previous version.

A Generating Random Numbers With Non-Uniform Densities

We often want to generate sets of random or pseudorandom points on the unit square for applications such as distribution ray tracing. There are several methods for doing this such as jittering and Poisson disk sampling. These methods give us a set of N reasonably equidistributed points on the unit square: (u_1, v_1) through (u_N, v_N).

Sometimes, our sampling space may not be square (e.g. a circular lens), or may not be uniform (e.g. a filter function centered on a pixel). It would be nice if we could write a mathematical transformation that would take our equidistributed points (u_i, v_i) as input, and output a set of points in our desired sampling space with our desired density. For example, to sample a camera lens, the transformation would take (u_i, v_i) and output (r_i, θ_i) such that the new points were approximately equidistributed on the disk of the lens.

If the density is a one dimensional $f(x)$ defined over the interval $x \in [x_{min}, x_{max}]$, then we can generate random numbers α_i that have density f from a set of uniform random numbers ξ_i, where $\xi_i \in [0, 1]$. To do this we need the cumulative probability distribution function $P(x)$:

$$Prob(\alpha < x) = P(x) = \int_{x_{min}}^{x} f(x')d\mu \tag{17}$$

To get α_i we simply transform ξ_i:

$$\alpha_i = P^{-1}(\xi_i) \tag{18}$$

where P^{-1} is the inverse of P. If P is not analytically invertible then numerical methods will suffice because an inverse exists for all valid probability distribution functions.

For example, to choose random points x_i that have the density $p(x) = 3x^2/2$ on $[-1, 1]$, we see that $P(x) = (x^3+1)/2$, and $P^{-1}(x) = \sqrt[3]{2x-1}$, so we can "warp" a set of canonical random numbers $(\xi_1, \cdots, \xi_N)$ to the properly distributed numbers $(x_1, \cdots, x_N) = (\sqrt[3]{2\xi_1 - 1}, \cdots, \sqrt[3]{2\xi_N - 1})$. Of course, this same warping function can be used to transform "uniform" Poisson disk samples into nicely distributed samples with the desired density.

If we have a random variable $\alpha = (\alpha_x, \alpha_y)$ with two dimensional density (x, y) defined on $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$ then we need the two dimensional distribution function:

$$Prob(\alpha_x < x \text{ and } \alpha_y < y) = F(x, y) = \int_{y_{min}}^{y} \int_{x_{min}}^{x} f(x', y') d\mu(x', y')$$

We first choose an x_i using the marginal distribution $F(x, y_{max})$, and then choose y_i according to $F(x_i, y)/F(x_i, y_{max})$. If $f(x, y)$ is separable (expressible as $g(x)h(y)$), then the one dimensional techniques can be used on each dimension.

For example, suppose we are sampling uniformly from the disk of radius R, so $p(r, \theta) = 1/(\pi R^2)$. The two dimensional distribution function is:

$$Prob(r < r_0 \theta < \theta_0) = F(r_0, \theta_0) = \int_0^{\theta_0} \int_0^{r_0} \frac{r dr d\theta}{\pi R^2} = \frac{\theta r^2}{2\pi R^2}$$

This means that a canonical pair (ξ_1, ξ_2) can be transformed to a uniform random point on the disk: $(r, \theta) = (R\sqrt{\xi_1}, 2\pi\xi_2)$.

To choose random points on a triangle defined by vertices p_0, p_1, and p_2, a more complicated analysis leads to the transformation $u = 1 - \sqrt{1 - \xi_1}$, $v = (1 - u)\xi_2$, and the random point p will is:

$$p = p_0 + u(p_1 - p_0) + v(p_2 - p_0).$$

To choose reflected ray directions for zonal calculations or distributed ray tracing, we can think of the problem as choosing points on the unit sphere or hemisphere (since each ray direction ψ can be expressed as a point on the sphere). For example, suppose that we want to choose rays according to the density:

$$p(\theta, \phi) = \frac{n + 1}{2\pi} \cos^n \theta \tag{19}$$

Where n is a Phong-like exponent, θ is the angle from the surface normal and $\theta \in [0, \pi/2]$ (is on the upper hemisphere) and ϕ is the azimuthal angle ($\phi \in [0, 2\pi]$). The distribution function is:

$$P(\theta, \phi) = \int_0^{\phi} \int_0^{\theta} p(\theta', \phi') \sin\theta' d\theta' d\phi' \tag{20}$$

The $\cos\theta'$ term arises because on the sphere $d\omega = \cos\theta d\theta d\phi$. When the marginal densities are found, p (as expected) is separable and we find that a (ξ_1, ξ_2) pair of canonical random numbers can be transformed to a direction by:

$$(\theta, \phi) = (\arccos((1 - r_1)^{\frac{1}{n+1}}), 2\pi r_2)$$

One nice thing about this method is that a set of jittered points on the unit square can be easily transformed to a set of jittered points on the hemisphere with a distribution of Equation 19. If n is set to 1 then we have a diffuse distribution needed for a Monte Carlo zonal method.

For a zonal or ray tracing application, we choose a scattered ray with respect to some unit normal vector $\vec{\mathbf{N}}$ (as opposed to the z axis). To do this we can first convert the angles to a unit vector a:

$$\vec{\mathbf{a}} = (\cos\phi \sin\theta, \sin\phi \sin\theta, \cos\theta)$$

We can then transform $\vec{\mathbf{a}}$ to be an $\vec{\mathbf{a}}'$ with respect to ψ by multiplying $\vec{\mathbf{a}}$ by a rotation matrix R ($\vec{\mathbf{a}}' = R\vec{\mathbf{a}}$). This rotation matrix is simple to write down:

$$R = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix}$$

where $\vec{\mathbf{u}} = (u_x, u_y, u_z)$, $\vec{\mathbf{v}} = (v_x, v_y, v_z)$, $\vec{\mathbf{w}} = (w_x, w_y, w_z)$, form a basis (an orthonormal set of unit vectors where $\vec{\mathbf{u}} = \vec{\mathbf{v}} \times \vec{\mathbf{w}}$, $\vec{\mathbf{v}} = \vec{\mathbf{w}} \times \vec{\mathbf{u}}$, and $\vec{\mathbf{w}} = \vec{\mathbf{u}} \times \vec{\mathbf{v}}$) with the constraint that $\vec{\mathbf{w}}$ is aligned with $\vec{\mathbf{N}}$:

$$\vec{\mathbf{w}} = \frac{\vec{\mathbf{N}}}{|\vec{\mathbf{N}}|}$$

To get $\vec{\mathbf{u}}$ and $\vec{\mathbf{v}}$, we need to find a vector $\vec{\mathbf{t}}$ that is not collinear with $\vec{\mathbf{w}}$. To do this simply set $\vec{\mathbf{t}}$ equal to $\vec{\mathbf{w}}$ and change the smallest magnitude component of $\vec{\mathbf{t}}$ to one. The $\vec{\mathbf{u}}$ and $\vec{\mathbf{v}}$ follow easily:

$$\vec{\mathbf{u}} = \frac{\vec{\mathbf{t}} \times \vec{\mathbf{w}}}{|\vec{\mathbf{t}} \times \vec{\mathbf{w}}|}$$

$$\vec{\mathbf{v}} = \vec{\mathbf{w}} \times \vec{\mathbf{u}}$$

As an efficiency improvement, you can avoid taking trigonometric functions of inverse trigonometric functions (e.g. $\cos \arccos \theta$). For example, when $n = 1$ (a diffuse distribution), the vector $\vec{\mathbf{a}}$ simplifies to

$$\vec{\mathbf{a}} = (\cos(2\pi\xi_1)\sqrt{\xi_2}, \sin(2\pi\xi_1)\sqrt{\xi_2}, \sqrt{1-\xi_2})$$

References

[1] John M. Airey and Ming Ouh-Young. Two adaptive techniques let progressive radiosity outperform the traditional radiosity algorithm. Technical Report TR89-20, Computer Science Department, University of North Carolina at Chapel Hill, August 1989.

[2] John M. Airey, John H. Rohlf, and Frederick P. Brooks. Towards image realism with interactive update rates in complex virtual building environments. *Computer Graphics*, 24(1):41–50, 1990. ACM Workshop on Interactive Graphics Proceedings.

[3] James Arvo. Backward ray tracing. *Developments in Ray Tracing*, pages 259–263, 1986. ACM Siggraph '86 Course Notes.

[4] James Arvo. Stratified sampling of spherical triangles. In Rob Cook, editor, *Proceedings of SIGGRAPH '95 (Anaheim, California, August 6–11, 1995)*, Computer Graphics Proceedings, Annual Conference Series, pages 437–438, August 1995.

[5] James Arvo and David Kirk. Particle transport and image synthesis. *Computer Graphics*, 24(3):63–66, August 1990. ACM Siggraph '90 Conference Proceedings.

[6] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional reflectance functions from surface bump maps. *Computer Graphics*, 21(4):273–282, July 1987. ACM Siggraph '87 Conference Proceedings.

[7] Shenchang Eric Chen, Holly Rushmeier, Gavin Miller, and Douglass Turner. A progressive multi-pass method for global illumination. *Computer Graphics*, 25(4):165–174, July 1991. ACM Siggraph '91 Conference Proceedings.

[8] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. *Computer Graphics*, 22(4):75–84, August 1988. ACM Siggraph '88 Conference Proceedings.

[9] Michael F. Cohen and Donald P. Greenberg. The hemi-cube: a radiosity solution for complex environments. *Computer Graphics*, 19(3):31–40, July 1985. ACM Siggraph '85 Conference Proceedings.

[10] Michael F. Cohen, Donald P. Greenberg, David S. Immel, and Philip J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics & Applications*, 6(2):26–35, 1986.

[11] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 5(1):51–72, January 1986.

[12] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. *Computer Graphics*, 18(4):165–174, July 1984. ACM Siggraph '84 Conference Proceedings.

[13] R. C. Corlett. Direct Monte Carlo calculation of radiative heat transfer in vacuum. *Journal of Heat Transfer*, pages 376–382, November 1966.

[14] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan-Kaufman, San Francisco, 1995.

[15] Cindy M. Goral, Kenneth E. Torrance, and Donald P. Greenberg. Modeling the interaction of light between diffuse surfaces. *Computer Graphics*, 18(4):213–222, July 1984. ACM Siggraph '84 Conference Proceedings.

[16] David Edward Hall. An analysis and modification of Shao's radiosity method for computer graphics image synthesis. Master's thesis, Department of Mechanical Engineering, Georgia Institute of Technology, March 1990.

[17] John H. Halton. A retrospective and prospective of the Monte Carlo method. *SIAM Review*, 12(1):1–63, January 1970.

[18] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Wiley, New York, N.Y., 1964.

[19] Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *Computer Graphics*, 24(3):145–154, August 1990. ACM Siggraph '90 Conference Proceedings.

[20] Paul S. Heckbert. What are the coordinates of a pixel? In Andrew Glassner, editor, *Graphics Gems*. Academic Press, New York, NY, 1990.

[21] J. R. Howell and M. Perlmutter. Monte Carlo solution of thermal transfer through radiant media between gray walls. *Journal of Heat Transfer*, pages 116–122, February 1964.

[22] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. *Computer Graphics*, 20(4):133–142, August 1986. ACM Siggraph '86 Conference Proceedings.

[23] Theodore M. Jenkins, Walter R. Nelson, and Alessandro Rindi, editors. *Monte Carlo Transport of Electrons and Photons*. Plenum Press, New York, N.Y., 1988.

[24] Henrik Wann Jensen. Importance driven path tracing using the photon map. In *Rendering Techniques '95*. Springer-Verlag/Wien, 1995.

[25] James T. Kajiya. The rendering equation. *Computer Graphics*, 20(4):143–150, August 1986. ACM Siggraph '86 Conference Proceedings.

[26] David Kirk and James Arvo. Unbiased sampling techniques for image sysnthesis. *Computer Graphics*, 25(4):153–156, July 1991. ACM Siggraph '91 Conference Proceedings.

[27] David Kirk and James Arvo. Unbiased variance reduction for global illumination. In *Proceedings of the Second Eurographics Workshop on Rendering (Barcelona, May 1991)*, 1991.

[28] Arjan F. Kok. Grouping of patches in progressive radiosity. In *Proceedings of the Fourth Eurographics Workshop on Rendering*, pages 221–231, 1993.

[29] Arjan J. F. Kok and Frederik W. Jasen. Source selection for the direct lighting calculation in global illumination. In *Proceedings of the Second Eurographics Workshop on Rendering (Barcelona, May 1991)*, pages 75–82, 1991.

[30] Eric P. Lafortune and Yves D. Willems. Bidirectional path tracing. In *Proceedings of COMPUGRAPHICS*, pages 145–153, 1993.

[31] Brigitta Lange. The simulation of radiant light transfer with stochastic ray-tracing. In *Proceedings of the Second Eurographics Workshop on Rendering (Barcelona, May 1991)*, 1991.

[32] Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. *Computer Graphics*, pages 199–208, August 1993. ACM Siggraph '93 Conference Proceedings.

[33] Thomas J. V. Malley. A shading method for computer generated images. Master's thesis, Computer Science Department, University of Utah, June 1988.

[34] Don P. Mitchell. Spectrally optimal sampling for distributed ray tracing. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 157–164, July 1991.

[35] Don P. Mitchell. Ray tracing and irregularities of distribution. In *Proceedings of the Third Eurographics Workshop on Rendering*, pages 61–70, 1992.

[36] László Neumann, Martin Feda, Manfred Kopp, and Werner Purgathofer. The stochastic ray method for radiosity. In *Proceedings of the Sixth Eurographics Workshop on Rendering*, pages 206–218, June 1995.

[37] S. N. Pattanaik. *Computational Methods for Global Illumination and Visualisation of Complex 3D Environments*. PhD thesis, Birla Institute of Technology & Science, Computer Science Department, Pilani, India, February 1993.

[38] Werner Purgathofer. A statistical method for adaptive stochastic sampling. *Computers and Graphics*, 11(2):157–162, feb 1987.

[39] Holly Rushmeier, Charles Patterson, and Aravindan Veerasamy. Geometric simplification for indirect illumination calculations. In *Proceedings of Graphics Interface '93*, pages 227–236, Toronto, Ontario, Canada, May 1993. Canadian Information Processing Society.

[40] Holly E. Rushmeier. *Realistic Image Synthesis for Scenes with Radiatively Participating Media*. PhD thesis, Cornell University, May 1988.

[41] Holly E. Rushmeier and Kenneth E. Torrance. Extending the radiosity method to include specularly reflecting and translucent materials. *ACM Transaction on Graphics*, 9(1):1–27, January 1990.

[42] Bertrand Le Saec and Christophe Schlick. A progressive ray-tracing-based radiosity with general reflectance functions. In *Proceedings of the Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pages 103–116, June 1990.

[43] P. Shirley. Discrepancy as a quality measure for sample distributions. In Werner Purgathofer, editor, *Eurographics '91*, pages 183–194. North-Holland, September 1991.

[44] Peter Shirley. A ray tracing method for illumination calculation in diffuse-specular scenes. In *Proceedings of Graphics Interface '90*, pages 205–212, May 1990.

[45] Peter Shirley. *Physically Based Lighting Calculations for Computer Graphics*. PhD thesis, University of Illinois at Urbana-Champaign, January 1991.

[46] Peter Shirley. Radiosity via ray tracing. In James Arvo, editor, *Graphics Gems 2*. Academic Press, New York, NY, 1991.

[47] Peter Shirley. Time complexity of Monte Carlo radiosity. In *Eurographics '91*, pages 459–466, September 1991.

[48] Peter Shirley, Kelvin Sung, and William Brown. A ray tracing framework for global illumination systems. In *Proceedings of Graphics Interface '91*, pages 117–128, June 1991.

[49] Peter Shirley and Changyaw Wang. Direct lighting by Monte Carlo integration. In *Proceedings of the Second Eurographics Workshop on Rendering (Barcelona, May 1991)*, 1991.

[50] Y. A. Shreider. *The Monte Carlo Method*. Pergamon Press, New York, N.Y., 1966.

[51] François X. Sillion, James Arvo, Stephen Westin, and Donald Greenberg. A global illumination algorithm for general reflection distributions. *Computer Graphics*, 25(4):187–196, July 1991. ACM Siggraph '91 Conference Proceedings.

[52] François X. Sillion and Claude Puech. A general two-pass method integrating specular and diffuse reflection. *Computer Graphics*, 23(3):335–344, July 1989. ACM Siggraph '89 Conference Proceedings.

[53] Jerome Spanier and Ely M. Gelbard. *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley, New York, N.Y., 1969.

[54] Jerome Spanier and Earl H. Maize. Quasi-random methods for estimating integrals using relatively small samples. *SIAM Review*, 36(1):18–44, March 1994.

[55] Dan Stanger. Monte Carlo procedures in lighting design. *Journal of the Illumination Engineering Society*, pages 14–25, July 1984.

[56] J. S. Toor and R. Viskanta. A numerical experiment of radiant heat interchange by the Monte Carlo method. *International Journal of Heat and Mass Transfer*, 11:883–897, 1968.

[57] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 147–162, June 1994.

[58] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *SIGGRAPH 97 Conference Proceedings*, pages 65–76. ACM SIGGRAPH, August 1997.

[59] John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. A two-pass solution to the rendering equation: a synthesis of ray tracing and radiosity methods. *Computer Graphics*, 21(4):311–320, July 1987. ACM Siggraph '87 Conference Proceedings.

[60] Bruce Walter, Philip M. Hubbard, Peter Shirley, and Donald F. Greenberg. Global illumination using local linear density estimation. *ACM Transactions on Graphics*, 16(3):217–259, July 1997.

[61] Changyaw Wang. Physically correct direct lighting for distribution ray tracing. In David Kirk, editor, *Graphics Gems 3*. Academic Press, New York, NY, 1992.

[62] Gregory J. Ward. The RADIANCE lighting simulation and rendering system. *Computer Graphics*, 28(2):459–472, July 1994. ACM Siggraph '94 Conference Proceedings.

[63] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 85–92, August 1988.

[64] T. Whitted. An improved illumination model for shaded display. *CACM*, 23(6):343–349, June 1980.

[65] H. Wozniakowski. Average case complexity of multivariate integration. *Bulletin (New Series) of the American Mathematical Society*, 24(1):185–193, January 1991.

[66] Sidney J. Yakowitz. *Computational Probability and Simulation*. Addison-Wesley, New York, N.Y., 1977.

[67] S. K. Zaremba. The mathematical basis of Monte Carlo and quasi-Monte Carlo methods. *SIAM Review*, 10(3):303–314, July 1968.

From Solution to Image

Holly E. Rushmeier

updated from an article that appeared in the "Making Radiosity Practical" course notes in SIGGRAPH 93

1 General Remarks

Global illumination methods are techniques for accurately calculating the transport of radiation in an environment. In this course,we consider methods for calculating the transport of visible light for the purpose of generating realistic images. The accuracy of the final image depends not only on the specific method employed, but on the quality of the input data, and on the methods used to transform the results of the calculation to an image on a display device. In this section we will consider the problem of transforming global illumination results into a displayable image. This problem requires some understanding of a number of complex subject areas such as perception, colorimetry, etc. This section is intended only as a brief introduction, with a just a few references to the extensive literature available in these areas.

Global illumination methods take as input geometry and material properties of the environment. From this, methods compute radiosities or radiances for discrete values of location, direction and wavelength. For methods that don't compute images pixel by pixel, but in object space, continuous radiance distribution must be reconstructed from these discrete values. This continous distribution is then resampled for generating an image on a display device. Generally, the dynamic and spectral ranges of the radiosity results are not in the range of the typical display devices we use. We need to use perceptually based models to map results to the display device to obtain the best rendition of a scene. In this section we discuss how to convert the discrete values from an illumination calculation into an image.

The human visual system has been studied for centuries, and volumes of observations and theories can be found. Understanding how a spectral distribution of radiant energy is

converted into an idea in the human mind is a more complex problem that the light transport problem. Some insight into the the relevant issues can be found in relatively accessible form in references such as [1], [4], and [28].

In this section we present some very simplistic ideas on how principles of human vision can be applied to image generation. We include these only to give a flavor of how global illumination fits in to the overall image synthesis process. A much more careful consideration of the ideas discussed here is needed for any application in which highly accurate renderings of images are required.

We will consider three basic topics in vision – spatial variation, color, and brightness.

2 Spatial Variation

There are at least two problems that can be considered in this area – reconstructing continuous spatial variations of radiance/radiosity solutions from discrete samples, and resampling the distributions in image space.

2.1 Mach Banding

The simple way to reconstruct a point-sampled spatial radiosity solution is to bilinearly interpolate between samples. This can give relatively small percentage errors in radiosity or radiance at each location. However, simply reducing error in the reconstruction at each point is not adequate. The human eye is very sensitive to changes in the spatial gradiant of luminance, producing what are known as "Mach Bands." Changes in gradient can produce the perception of light or of dark bands, where no such bands exist in the radiance distribution. A crude explanation of this phenomenon is that receptors in the eye do not act independently. Receptor response depends not only on the incident illumination, but the illumination on neighboring receptors. An interesting observation is that the bands only occur where there are changes in luminance. If luminance doesn't change, spectral variations alone do not produce Mach bands.

In practice, most people probably reduce the mesh size after they see the Mach band artifacts in their image. Often when images are recorded on film the gradients become steeper, and the Mach bands become more noticeable. This makes the "fixing it after you see it" approach even less practical.

An example of a non-ad hoc method for adjust meshing taking into account perceptual effects is presented by Hedley et al. [9]. They discuss how to reduce the number of

discontinuities which force mesh subdivisions by taking into account how the solution will ultimately be mapped to the display device.

2.2 Anti Aliasing

Even global illumination calculations that are performed in object space ultimately have to be sampled in image space to determine pixel values. The problem of sampling so that the representation of the continuous image by discrete pixels does not produce visual artifacts has been studied extensively in computer graphics for many years (e.g. [5]).

A unique aspect of the antialiasing problem when physically accurate global illumination methods are used is that pixel sampling can be performed at two different steps in the image generation process. One option is to sample each pixel and determine the radiance, and then transform that radiance to monitor coordinates (i.e. 0 to 255 values for RGB). A disadvantage of this method is that there will be a very high variance in the radiance for pixels on the borders of light sources . As a result, if a stochastic sampling method is used to find radiance pixel values, the result will be ragged edges on light sources unless extremely high numbers of samples are used. An alternative is to transform all of the radiances to RGB values first, and then sample for pixel values. This is much more efficient, but sacrifices accuracy very slightly.

A study of filtering to attempt to avoid the ragged edges caused by the wide range of sample values without clipping the values first is given in [18].

2.3 Other Consequences of Spatial Variations

Mach banding is just one effect demonstrating that perception is a function of the spatial variation of luminance.

In a study of how to measure the similarity of real and synthetic images Rushmeier et al. [20] found that useful metrics included a filtering by the human spatial contrast sensitivity function. In that paper it was found to be more useful to compare images in the spatial frequency domain, rather than pixel by pixel.

Recently Ferwerda et al. [7] presented a paper discussing the varying sensitivity of the human visual system to spatial variations of luminance, and how that sensitivity varies with the content of the image. This is an effect that could possibly be exploited to compute images more efficiently without any degradation in the perceived image quality.

3 Color

3.1 Metamers

In the physical world, there are an infinite number of possible continuous spectral distributions for radiance. On a typical display device only a finite number of distributions can be displayed. Fortunately, the human eye can not distinguish between all possible spectral distributions. Many distributions appear to humans to have the same color. This phenomenon is known as color metamerism. The basic solution to displaying spectral distributions is to find an RGB triplet on the monitor you are using which is a metamer of the distribution calculated by the illumination calculations.

The physical mechanism behind metamerism is that the color we see is the result of the response of three types of receptors in the eye, each of which produce a signal which is the result of integrating the incident spectral distribution with a filter. The three receptors correspond to low, medium and high wavelength band filters. This suggests that for the purposes of human perception, spectral distributions can be represented in terms of three functions. This is the motivation behind the development of the CIE standard color matching functions $x(\lambda)$, $y(\lambda)$ and $z(\lambda)$. (Note however that these functions are not estimates of the receptor sensitivities.)

The $y(\lambda)$ function is essentially equal to the luminous efficiency function. Full sets of values can be found in [29], [10], etc. Integrating the spectral distribution weighted by the functions results in the X, Y, Z tristimulus values in the CIE colorimetric system. That is:

$$X = k \int L(\lambda) x(\lambda) d\lambda$$
$$Y = k \int L(\lambda) y(\lambda) d\lambda$$
$$Z = k \int L(\lambda) z(\lambda) d\lambda$$

where k is a constant.

If L is expressed in $W/m^2 str$ and k is chosen to be 1/680, Y is equal to the luminance of the distribution in cd/m^2. Chromaticity coordinates, (x, y), are defined by:

$$x = X/(X + Y + Z)$$
$$y = Y/(X + Y + Z)$$

Using the functions $x(\lambda)$, $y(\lambda)$, and $z(\lambda)$ device independent XYZ values can be computed for each spectral radiance distribution obtained from the illumination calculation. Spectral distributions with the same tristimulus values appear to a human viewer to be the same color. The use of this idea to to display images is described in detail in [14] and [8]. For a particular monitor, the values of x and y can be measured or obtained from the manufacturer for each of the phosphors, along with the values of Y for each phosphor when it is set at a unit value. As an example of the magnitude of these quantities, for the experiment described in [13], the chromaticities were found to be:

$$(x_r, y_r) = (.64, .33), (x_g, y_g) = (.29, .60), (x_b, y_b) = (.15, .06)$$

The ratios of luminances at the white point were found to be $Yr : Yg : Yb$ = .3142:1:.1009, and the total luminance of the white point was 82 cd/m^2.

From this information X, Y, and Z for a unit value of each of the RGB primaries can be found. Knowing these values, a transformation between RGB triplet and an XYZ value can be calculated by a simple matrix multiplication:

$$[X, Y, Z]^T = M[R, G, B]^T, M = [(X_r, X_g, X_b), (Y_r, Y_g, Y_b), (Z_r, Z_g, Z_b)]^T$$

For image synthesis, we want to determine RGB for a given XYZ. This can be computed then using:

$$[R, G, B]^T = M^{-1}[X, Y, Z]^T$$

Note that chromaticities and luminances vary for different classes of display device. The same RGB values on a desktop CRT and on a typical laptop will look quite different.

The three by three matrix multiplication, in theory, allows the display of any calculated radiance distribution on a monitor. However, undisplayable values may be obtained for R, G or B. This may be cause the chromaticity (x, y) is outside of the displayable range of the monitor. A discussion of this case is can be found in [8]. Another problem is that the luminance of the calculated radiance distribution may be outside of the range of the monitor. This problem is discussed in the following section on brightness.

3.2 Color Constancy

Unfortunately, even producing a completely accurate reconstruction of the spectral distribution and appropriately converting the distribution to XYZ and to RGB coordinates will not always produce a satisfactory image. For example, if you render a room illuminated

by a tungsten source the image will look oddly reddish. In one sense that is the correct result. If you were outside under a sodium street light at night, looking into the room, it would look reddish. However, if you are in the room and see only objects illuminated by the tungsten light, the reddish cast is gone. White objects look white. This is an example of "color constancy." Our visual system adjusts so that objects appear to be essentially the same color to us, even though the reflected spectral distribution from the object changes. Nobody has introduced a a completely robust way to deal with color constancy. The most common work around is to model light sources with a a flat white spectral distribution, rather than with their true spectral distribution.

One promising approach for color constancy is to apply Land's retinex theory [12]. These approaches have been developed in the image processing literature [11] for application to physically acquired, rather than computer generated, imagery.

Besides the color constancy effect, some researchers question whether the CIE color matching can be applied to match emitting and reflected light [21]. That is, it is not clear that an XYZ triplet emitted from a spot on a CRT will appear the same as a spot in the environment reflecting light with the same XYZ values.

3.3 Other Effects

It should also be noted that the perception of color is not independent of spatial variaions. The difference in a human's ablility to detect luminance variations versus color variations was used in developing the original color television broadcast standards. Exploiting these differences is also a topic of research in image synthesis.

4 Brightness

If the range of luminances in a scene were in the same range as those displayed by the monitor, the XYZ/RGB calculations described in the previous section would be all that is needed. However, real world luminances can vary from 10^{-6} to 10^{4} cd/m^2, while monitor luminances are in the range 1 to 100 cd/m^2. Some transformation is needed map real world luminances to the monitor luminances.

The first logical mapping most people think of is a linear mapping, setting the maximum real world luminance to the maximum monitor luminance. For example, if a white light source is in the scene, its RGB values are set to 255, 255,255. Unfortunately, because of the dynamic range of luminance in most common environments, the result of

this mapping will be an image in which the source is a white bright spot, and everthing else is black.

Several other alternative scaling factors are obvious, choosing the highest luminance that is not from a light source to be the maximum monitor luminance, or choosing the average luminance to be the average monitor luminance. All of these approaches have the disturbing characteristic that they are independent of the absolute light level. The image will look the same regardless of whether it is illuminated by fire flies or an aircraft searchlight.

One early approach, detailed in [24] is to try to solve the luminance mapping problem in the same spirit as the color reproduction problem. Just as human beings are not sensitive to precise spectral distributions of light, they are also not very good at judging the physical magnitude of light reaching the eyes. Humans are more sensitive to luminance variations than to absolute values. Just as we perceive colors rather than spectral distributions, we perceive brightness, rather than absolute luminance. In an initial attempt at this approach, only grey scale images are considered in [24].

For the color problem, the transformation to XYZ coordinates provided a mechanism to find distributions displayable on the monitor that are percieved to be the same color as the physical spectral distribution calculated by the radiosity method. We seek a similar transformation from luminance to brightness that will allow the display of luminance distributions on the monitor that will be preceived by the observer to have the same brightness as the physical scene. This type of transformation has been studied in photography, and is referred to as the tone reproduction operator.

Many different models for the tone reproduction operator could be used. As an example [24] describes the use of a model based on an experiment described in [23]. In the work described in [23], brightness is measured of units of brils, where a bril is the sensation of brightness from a fully dark adapted eye viewing a 5 degree target of 1 micro-lambert for one second. In the experiments, the brightness in brils was measured as a function of luminance for various adaptation levels of the eye. The result was a set of straight line curves of brightness versus lumninance for various luminant adaptation levels. The curve values can be expressed in the following equation:

$$B = 10^{\beta} L^{\alpha}$$

$$\alpha = 0.4 log_{10}(L_w) + 2.92, \beta = -0.4(log_{10}(L_w))^2 + (-2.584 log_{10}(L_w)) + 2.0208$$

B is brightness in brils, L is luminance, and L_w is the luminance of the adaptation level.

Equating brightnesses for the observers of the monitor and the physical scene can give a relationship between the luminance calculated by a radiosity solution L_{rw} and the luminance to be displayed L_d.

$$L_d = L_{rw}^{\alpha_{rw}/\alpha_d} 10^{(\beta_{rw}-\beta_d)/\alpha_d]}$$

To make a practical calculation, the luminances of the adaptation levels of the two observers need to be approximated. [24] suggests letting the real world adaptation level $L_{w(rw)}$ be:

$$log_{10}(L_{w(rw)}) = E[log_{10}(L_{rw})] + 0.84$$

where E is the statistical mean over the image, and the monitor adaptation level be the monitor peak luminance $L_{d,max}$. For typical monitors peak luminance is about 85 cd/m^2.

Also, a relationship is needed between framebuffer value n (assuming $0 < n < 1$) which specfies RGB and the luminance output. Letting γ be the correction for the non-linear relationship between gun voltage and luminous output and C_{max} be the maximum contrast ratio between screen luminances (usually around 35 for CRT's), the relationship between n and L_d is:

$$n = [(L_d/L_{d,max}) - (1/C_{max})]^{1/\gamma}$$

The transformation from real world to display luminances, expressions for luminance adaptation level and the relationship between frame buffer value and display luminance gives a complete mapping from simulated real world luminances to image values.

The above treatment of brightness is only one possible method, and was developed for gray scale images.

Another early approach, from the illumination engineering literature is discussed in [21]. This approach converts RGB calculated from the XYZ values associated with each point, into the gamut of the display device. The approach is based on a series of experiments in which observers compared images and scale models of the environment imaged. It takes into account the effect that regardless of their true spectrum, surfaces with relatively high luminances tend to appear white. There are two steps in the method.

First consider an orthogonal coordinate system in which the axes are R, G and B. Let θ_o be the angle between the values (R_o, G_o, B_o) to be converted, and the line that passes through (0,0,0) and (1,1,1).

RGB triplets are shifted towards white using the following:

$$\theta_1 = \theta_o(1 - (Y/\zeta Y_{max})^\eta)$$

This shift transforms (R_o, G_o, B_o) into (R_1, G_1, B_1). The new values are then scaled into the final display values using:

$$R_2 = R_1(Y/Y_{max})^\nu, G_2 = G_1(Y/Y_{max})^\nu, B_2 = B_1(Y/Y_{max})^\nu$$

The values of η, ζ and ν were found by perceptual experiments to be 0.75, 5. and 0.75 respectively.

Niether of these two early methods is perfect, particularly if an image has a very high dynamic range. A number of other methods for dealing with tone mapping have been developed in the past few years. Ward [26] developed a simple linear operator for preserving feature visibility. Chiu et al. [3] presented a non-uniform spatial scaling method for mapping images with very high dynamic ranges. Schlick [19] developed an alternative to this method with improved computational efficiency. The retinex methods that account for color constancy [11] also map the wide dynamic range of luminances to the range of the display device. Two of the most recent tone reproduction methods can be found in [27] and [25].

5 Other Effects

There are other important vision effects besides those discussed here. Very bright areas in real life produce glare. Simulating the effects of glare in images is discussed in [16] and in [22]. Lower light levels result in changes in spatial acuity and color sensitivity. These effects are discussed in [6] and are also incorporated in the model presented in [27].

The application of perceptual principles to realistic image synthesis is an active area of research. Two papers related to this area are being presented at SIGGRAPH 98 ([17], [2].)

References

[1] K.R. Boff and J.E. Lincoln. *Engineering Data Compendium: Human Perception and Performance*, Vol. 1. Harry Armstrong Aerospace Medical Research Laboratory, Wright-Pattterson Air Force Base, 1988.

[2] M. Bolin and G. Meyer, "A Perceptually Based Adaptive Sampling Algorithm" *Proceedings of SIGGRAPH 1998.*

[3] K. Chiu et al. “Spatially Nonuniform Scaling Functions for High Contrast Images,” *Proceedings of Graphics Interface*, 1993.

[4] Committee on Colorimetry. *The Science of Color*. Optical Society of America, Washington, DC, 1963.

[5] F.C. Crow “The Aliasing Problem in Computer-Generated Shaded Images,” *Communications of the ACM*Vol. 20, pp. 799, November 1977.

[6] J. Ferwerda, S. Pattanaik, P. Shirley, and D. Greenberg, “A Model of Visual Adaptation for Realistic Image Synthesis”, *Proceedings of SIGGRAPH 96*, pp. 249-258.

[7] J. Ferwerda, S. Pattaniak, P. Shirley, D. Greenberg, “A Model of Visual Masking for Computer Graphics”, *Proceedings of SIGGRAPH 97*, pp. 143-152.

[8] R. Hall. *Illumination and Color in Computer Generated Imagery*. Springer- Verlag, New York 1989.

[9] David Hedley, Adam Worrall, Derek Paddon, “Selective Culling of Discontinuity Lines” *Eighth Eurographics Workshop on Rendering*, pp. 69-80.

[10] *IES Lighting Handbook*, 1981 Reference Edition.

[11] D. J. Jobson, Z. Rahman, and G.A. Woodell, “Properties and Performance of a Center/Surround Retinex”, *IEEE Transactions on Image Processing* 6(3), pp 451-462 (March 1997).

[12] E. H. Land, “The Retinex Theory of Color Vision,” *The Scientific American* 237(6), 1977.

[13] G.W. Meyer, H.E. Rushmeier, M.F. Cohen, D.P. Greenberg and K.E. Torrance. “An Experimental Evaluation of Computer Graphics Imagery.” *ACM Transactions on Graphics*, Jan. 1986, pp. 30-50.

[14] G.W. Meyer. “Tutorial on Color Science.” *The Visual Computer*, 1986, pp. 278-290.

[15] G.W. Meyer. “Wavelength Selection for Synthetic Image Generation.” *Computer Vision, Graphics, and Image Processing*,1988, p. 57-79.

[16] E. Nakamae, K. Kaneda, T. Okamoto and T. Nishita. “A Lighting Model Aiming at Drive Simulators,” *Proceedings of SIGGRAPH 1990*, pp. 395-404.

[17] S. Pattanaik, J. Ferwerda, D. Greenberg and M. Fairchild, “A Multiscale Model of Adaptation and Spatial Vision for Realistic Imaging”, *Proceedings of SIGGRAPH 1998*

[18] H. Rushmeier and G. Ward “Energy Preserving Non-Linear Filters”, *Proceedings of SIGGRAPH 1994* pp. 131-8.

[19] C. Schlick,“Quantization Techniques for Visualization of High Dynamic Range Pictures,” *Photorealistic Rendering Techniques*, Springer-Verlag, 1995 pp. 7-20.

[20] H. Rushmeier, G. Ward, C. Piatko, P. Sanders and B. Rust, “Comparing Real and Synthetic Images: Some Ideas about Metrics”, *1995 Eurographics Workshop on Rendering*.

[21] M. Smith. “A New Method of Generating Accurate Color Renderings of Architectural Spaces,” *Journal of the Illuminating Engineering Society*, Winter, 1993, pp. 26-32.

[22] G. Spencer, P. Shirley, K. Zimmerman and D. Greenberg, “Physically-Based Glare Effects for Computer Generated Images”, *Proceedings of SIGGRAPH 95*, pp. 325-334.

[23] S. S. Stevens and J. C. Stevens “Brightness Function: Effects of Adaptation” *Journal of the Optical Society of America*, Volume 53, number 3, March (1963).

[24] J.T. Tumblin and H.E. Rushmeier. “Tone Reproduction for Realistic Computer Generated Images,” *IEEE Computer Graphics and Applications* 13(6)pp. 42-48.

[25] J.T. Tumblin, J. Hodges, and B. Guenter, “Two Methods for Display of High Contrast Images”, to appear in *ACM Transactions on Graphics*.

[26] G. Ward,“A Contrast-Based Scalefactor for Luminance Display” *Graphics Gemes IV* P. Heckbert, Ed., Academic Press.

[27] G. Ward Larson, H. Rushmeier, and C. Piatko, “A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes”, *IEEE Transactions on Visualization and Computer Graphics*, 3(4), October-December 1997, pp. 291-306

[28] W. Woodson. *Human Factors Design Handbook*. McGraw-Hill, New York 1981.

[29] G. Wyszecki and W. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, New York, 1982.

Further reading

An Introduction to Ray Tracing, *Edited by Andrew Glassner, Academic Press, 1989.* A survey of ray tracing and the techniques needed to implement it. Surprisingly relevent for such an old book!

Radiosity: A Programmer's Perspective, *Ian Ashdown, John Wiley & Sons, 1994.* Goes through a full and available C++ implementation of a radiosity program. Ashdown is a professional engineer, and gets the details right.

Radiosity and Realistic Image Synthesis, *Michael Cohen and John Wallace, Academic Press, 1993.* This is a technical overview of realistic rendering in general, and radiosity in particular. It is a good place to start getting the researcher's view of the field.

Radiosity and Global Illumination, *François Sillion and Claude Puech, Morgan Kaufmann, 1994.* Another technical overview of realistic rendering. More coverage of Monte Carlo techniques than the Cohen and Wallace books.

Illumination and Color in Computer Generated Imagery, *Roy Hall, Springer-Verlag, 1988.* An overview of the practicalities of light, optics and perception from a graphics viewpoint.

Light and Color in Nature and Art, *Samuel Williamson and Herman Cummins, Wiley, 1983,* A great all-around survey of light and color.

Thermal Radiation Heat Transfer, 3rd ed., *Robert Siegel and John Howell, Hemisphere, 1992.* Light transfer from the mechanical engineering perspective. These guys helped invent what we call the radiosity method in graphics. A very good read after you are fairly comfortable with the graphics side of the the problem.

The Illumination Engineering Society Lighting Handbook, *Edited by Mark Rea, Illumination Engineering Society, 1993.* This is a reference book that describes what lighting engineers need to know about light sources, optics, and perception. It is over $400, so get it at the library.

Principles of Digital Image Synthesis, *Andrew Glassner, Morgan Kaufmann, 1995 (2 vols).* Everything about rendering and more from the advanced perspective. A good graduate text for a full-year course. An errata sheet is available at:
http://www.research.microsoft.com/research/graphics/glassner/work/projects/pdis/errata.htm

Analytic Methods for Simulated Light Transport, *Jim Arvo, Yale University, 1995.* This

is a dissertation with a very careful discussion of the mathematics and physics of light transport. Although this document is extremely clear in its presentation, it is not for the mathematically timid. It is available online at: http://www.cs.caltech.edu/~arvo/.

Rendering with Radiance: The Art and Science of Lighting Visualization *Greg Ward Larson and Rob A. Shakespeare, with contributions from Peter Apian-Bennewitz, Charles Ehrlich, John Mardaljevic, a nd Erich Phillips, Morgan Kaufmann, 1998* Radiance is a UNIX software system for lighting design and rendering and it is freely available. This book is a complete description of how to use the software, how it works, and how to apply it to a variety of lighting design problems.

ONLINE RESOURCES

Websites come and go and change, but there are a couple "official" web sites that have been around for a while that offer a great deal of useful material.

Ian Ashdown maintains a number of bibliographies related to global illumination at
http://www.ledalite.com/library-/rrt.htm
including a Radiosity Bibliograpy, a list of Radiosity and Global Illumination Theses, and an Image-Based Rendering Bibliography.

Eric Haines publishes an electronic newsletter called "Ray Tracing News" that comes out about twice a year. Issues since 1987 are available at
http://www.acm.org/tog/resources/RTNews/html/index.html

Eric Haines also maintains a page of pointers to software tools that includes pointers to both ray tracing and radiosity packages at
htpp://www.acm.org/tog/Software.html