

Image-Based Lighting

Paul Debevec (organizer)
USC Institute for Creative Technologies

Dan Lemmon
Digital Domain

SIGGRAPH 2001 Course 14 (Half Day)
Sunday, August 12, 2000

Course Abstract

This course presents recently developed techniques for realistically integrating computer-generated imagery with live-action photography that use measurements of real-world lighting to illuminate CG objects. The course presents background material in high dynamic range photography, global illumination, and light reflection, and will describe both the theory and practice of acquiring measurements of light in the real world and injecting this light into computer animations. The techniques are illustrated with detailed breakdowns of the animations "Rendering with Natural Light" and "Fiat Lux" and shots from recent feature films. Examples of applying the techniques using commercial packages such as LightWave 3D are also presented. The course concludes with an overview of techniques to apply image-based lighting to people and objects in the real world.

Presenters

Paul Debevec

Executive Producer, Graphics Research
University of Southern California Institute for Creative Technologies
13274 Fiji Way, 5th Floor
Marina del Rey, CA 90292
(310) 574-7809
(310) 577-9140 fax
paul@debevec.org
<http://www.debevec.org/>

Paul Debevec earned degrees in Math and Computer Engineering at the University of Michigan in 1992 and a Ph.D. in Computer Science at UC Berkeley in 1996. He began doing research in computer graphics and vision in 1991 by deriving a textured 3D model of a Chevette from photographs for an animation project. At Interval Research Corporation he contributed to Michael Naimark's Immersion '94 virtual exploration of the Banff National forest and collaborated with Golan Levin on Rouen Revisited, an interactive visualization of the Rouen Cathedral and Monet's related series of paintings. Debevec's Ph.D. thesis presented an interactive method for modeling architectural scenes from photographs and rendering these scenes using projective texture-mapping. With this he led the creation of a photorealistic model of the Berkeley campus for his 1997 film *The Campanile Movie* whose techniques were later used to create the Academy Award-winning virtual backgrounds for the "bullet time" shots in the 1999 film *The Matrix*. Since his Ph.D. Debevec has worked on techniques for capturing real-world illumination and illuminating synthetic objects with real light, facilitating the realistic integration of real and computer generated imagery. His 1999 film *Fiat Lux* placed towering monoliths and gleaming spheres into a photorealistic reconstruction of St. Peter's Basilica, all illuminated by the light that was actually there. For real objects, Debevec has worked on developing the Light Stage, a device that allows objects and actors to be synthetically illuminated with any form of lighting. Debevec works at the University of Southern California's Institute for Creative Technologies, where he directs research in virtual actors, virtual environments, and applying computer graphics to creative projects.

Dan Lemmon

Technical Director
Digital Domain
300 Rose Avenue
Venice, CA 90291
310.314.2800 x.2303 (work)
310.314.2888 (fax)
dlemmon@d2.com

Dan Lemmon received a Bachelor's of Fine Arts in Industrial Design from Brigham Young University. While at BYU he began his work in Computer Graphics as an intern at Wavefront Technologies and Cinesite. Also while earning his BFA, Dan participated in three internships at James Cameron's visual effects company Digital Domain.

Dan's film work includes *The Fifth Element*, *Titanic*, *Fight Club*, *The Grinch*, and a variety of other films and television commercials. On *The Grinch* Dan led the creation of an automated tool for creating and clothing the huge library of characters to populate the town of Whoville. Dan has led recent efforts to incorporate image-based lighting, global illumination and high dynamic range rendering into the Digital Domain pipeline.

Course Schedule and Syllabus

1:30-2:15pm

1. Introduction (Debevec)
 - What is image-based lighting?
 - How does it differ from traditional techniques?
 - How can we use it with traditional techniques?
2. Global Illumination (A Brief Overview) (Lemmon)
 - The effects of global illumination
 - BRDFs: "The way light bounces off something"
 - Specular, diffuse, problems with the standard models
 - Shadows, highlights, retro-reflection, anisotropy, and Fresnel
 - Real-world examples
 - Radiosity: everything reflects everything else
 - Monte-Carlo integration
 - RADIANCE's method
 - Photon mapping
 - Not to be confused with Global Illumination: HDRI, image-based lighting, glare

2:15-3:00

3. Capturing, Representing, and Manipulating High Dynamic Range Imagery (HDRI) (Debevec)
 - How Cameras measure light
 - Lenses, Shutters, Apertures, CCD arrays, and ADCs
 - Luminance and color response
 - Imperfections: glare, flare, chromatic aberration
 - HDR: Taking a series of photographs
 - Deriving the response curve
 - Combining the photographs into a radiance map
 - HDR Shop: Creating, viewing and editing HDR imagery
 - High Dynamic Range image formats
 - LDR 8-bit and 16-bit formats
 - Nonlinear mappings and what's important to know about them.
4. Capturing Real-World Illumination (Debevec)
 - What to look for in a camera
 - Types of mirrored balls
 - Acquiring a light probe image using a mirrored ball
 - Alternate light probe techniques
 - Panoramic camera techniques
 - Low-dynamic range techniques
 - Real-time light probes
 - Rendering synthetic probe images
5. Illuminating Synthetic Objects with Real Light (Debevec)
 - Mapping a light probe image into a CG environment
 - Simulating light from the environment on the object
 - Converting light probe images to light source constellations
 - Real-time techniques: environment mapping and diffuse preconvolution



- High Dynamic Range Texture Maps
- 6. Making "Rendering with Natural Light" (SIGGRAPH 98 Electronic Theater) (Debevec)
 - Modeling the scene
 - Acquiring the light
 - Rendering the scene
 - Communicating the sense of brightness
 - Vignetting, Defocus, Glare, and Motion Blur
 - Using "pflare" to produce these effects
 - Re-rendering with Natural Light - creating the DVD edition
 - Improved sampling of curved surfaces



3:00-3:15 Break

- 7. Rendering Synthetic Objects into Real Scenes (Debevec)
 - Computing shadows, reflections, caustics, and interreflections
 - Compositing into background plates with differential rendering
 - Rendering into image-based environments
- 8. Image-Based Lighting in "Fiat Lux" (SIGGRAPH 99 Electronic Theater) (Debevec)
 - Combining Image-Based Lighting with Image-Based Modeling and Rendering
 - Creating "illum" light sources for direct lighting
 - Integrating animated objects
 - Avoiding flickering and sparkling
 - Inverse global illumination for the floor of St. Peter's
 - Distributed rendering



3:40-4:05pm

- 9. Image-Based Lighting with Commercial Renderers and in Commercial Production (Lemmon)
 - Image-Based Lighting with Lightwave 3D
 - Image-Based Lighting with Ray-Tracers
 - Image-Based Lighting with Scanline Rendering

4:15-5:00pm

- 10. Image-Based Lighting Real Objects and Actors
 - Acquiring 4D Reflectance Fields with a Light Stage
 - Illuminating reflectance fields
 - Interactive image-based lighting of human faces
 - Acquiring reflectance fields of objects
 - Combining with Environment Matting techniques
 - 8D Reflectance Fields
 - Reflectometry from reflectance fields and surface geometry

5:00pm End

Table of Contents

1. Course Slides 1

- Slides: 1. *Introduction: What is Image-Based Lighting?* (Debevec)
- Slides: 3. *Capturing, Representing, and Manipulating High Dynamic Range Imagery (HDRI)* (Debevec)
- Slides: 4. *Capturing Real-World Illumination* (Debevec)
- Slides: 5. *Illuminating Synthetic Objects with Real Light* (Debevec)
- Slides: 6. *Making "Rendering with Natural Light"* (Debevec)
- Slides: 7. *Rendering Synthetic Objects into Real Scenes* (Debevec)
- Slides: 8. *Image-Based Lighting in "Fiat Lux"* (Debevec)
- Slides: 9. *Image-Based Lighting with Commercial Renderers and in Commercial Production* (Lemmon)
- Slides: 10. *Image-Based Lighting Real Objects and Actors* (Debevec)

2. Course Slides 2

- Slides: 2. *Global Illumination: A Brief Overview* (Lemmon)

3. Supplemental Material

- Notes: *The Story of Reflection Mapping*
Paul Debevec
- Notes: *Illumination and Reflection Maps:
Simulated Objects in Simulated and Real Environments*
Gene S. Miller and C. Robert Hoffman, Course Notes for Advanced Computer Graphics
Animation, SIGGRAPH 84
- Notes: *Image-based Lighting in Lightwave 3D*
High Dynamic Range Images Exposed
Arnie Cachelin
- Sketch: *Image-Based Modeling, Rendering, and Lighting in Fiat Lux*
Paul Debevec, SIGGRAPH 99 Technical and Animation Sketch
- Notes: *Light Probe Image Gallery*
Paul Debevec, <http://www.debevec.org/Probes/>
- Sketch: *HDR Shop*
Chris Tchou and Paul Debevec, SIGGRAPH 2001 Technical Sketch (*draft*)
- Sketch: *A Real Time High Dynamic Range Light Probe*
Jamie Waese and Paul Debevec, SIGGRAPH 2001 Technical Sketch (*draft*)
- Sketch: *Light Stage 2.0*
Tim Hawkins, Jonathan Cohen, Chris Tchou, and Paul Debevec, SIGGRAPH 2001
Technical Sketch (*draft*)

4. Papers

- Paper: *Recovering High Dynamic Range Radiance Maps from Photographs.*
Paul E. Debevec and Jitendra Malik, Proc. SIGGRAPH 97

- Paper: *Overcoming Gamut and Dynamic Range Limitations in Digital Images*
Larson, G.W, Proc. Sixth Color Imaging Conference, November 1998
- Paper: *Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography*
Paul Debevec, Proc. SIGGRAPH 98
- Paper: *Acquiring the Reflectance Field of a Human Face*
Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar, Proc. SIGGRAPH 2000

5. CDROM Material: Images

- Image: *Rendering with Natural Light* still (1600 × 1200)
- Image: *Fiat Lux* still (3000 × 952)
- Image: Light Stage 2.0 still (2160 × 1440)
- Image: Image-Based Lighting in “Arnold” Mosaic (1280 × 1000)

6. CDROM Material: Animations

- Animation: *The Candlestick and Spheres on an Overcast Day* (1998)
- Animation: *The Space-Age Sepia Kitchen with Blur and Vignetting* (1998)
- Animation: *Synthetic Dominos on the Kitchen Table* (1998)
- Film: *Rendering with Natural Light* (SIGGRAPH 98 Electronic Theater)
Paul Debevec et al.
- Film: *Fiat Lux* (SIGGRAPH 99 Electronic Theater)
Paul Debevec et al.
- Film: *Image-Based Lighting* (SIGGRAPH 2000 Electronic Theater)
Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, and Westley Sarokin

7. CDROM Material: Light Probe Image Library

- Program: HDRView.exe Windows program for viewing and converting HDR images.
- Probes: Light Probe Images from *Rendering with Natural Light*, *Fiat Lux*, and the SIGGRAPH 98 image-based lighting paper. See also <http://www.debevec.org/Probes>

8. CDROM Material: Rendering with Natural Light Source Files

- Files: Everything needed to render the animation “*Rendering with Natural Light*” using the RADIANCE rendering system. See the README file for details.

Image-Based Lighting

SIGGRAPH 2001 Course #14

Paul Debevec

University of Southern
California

Dan Lemmon

Digital Domain



August 12, 2001

www.debevec.org/IBL2001

1. Introduction

What is Image-Based Lighting?

Traditional Lighting: Point Light Sources, and an Ambient Term

**Easy to set up, efficient to render, but
Not realistic, hard to match to real-world lighting**

What is Image-Based Lighting?

Image-Based Lighting: Light comes from everywhere – an *image* of incident illumination

**Realistic appearance, can be captured from the real world, but:
Slower to render**

HDRI, Image-Based Lighting, and Global Illumination

HDRI: A photographic method for capturing high dynamic range images of the real-world

IBL: Using images (usually HDRI) to illuminate CG objects, often using G.I.

GI: Rendering algorithms that properly simulate how light interreflects between surfaces in a scene, including diffuse interrefelction (e.g. radiosity, path tracing)

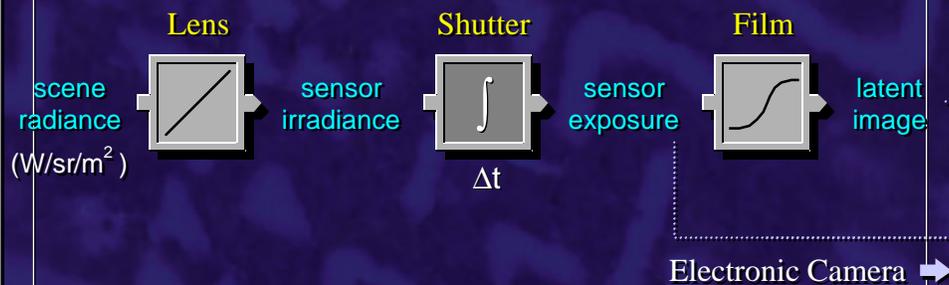
2. Global Illumination: A Brief Overview

Dan Lemmon

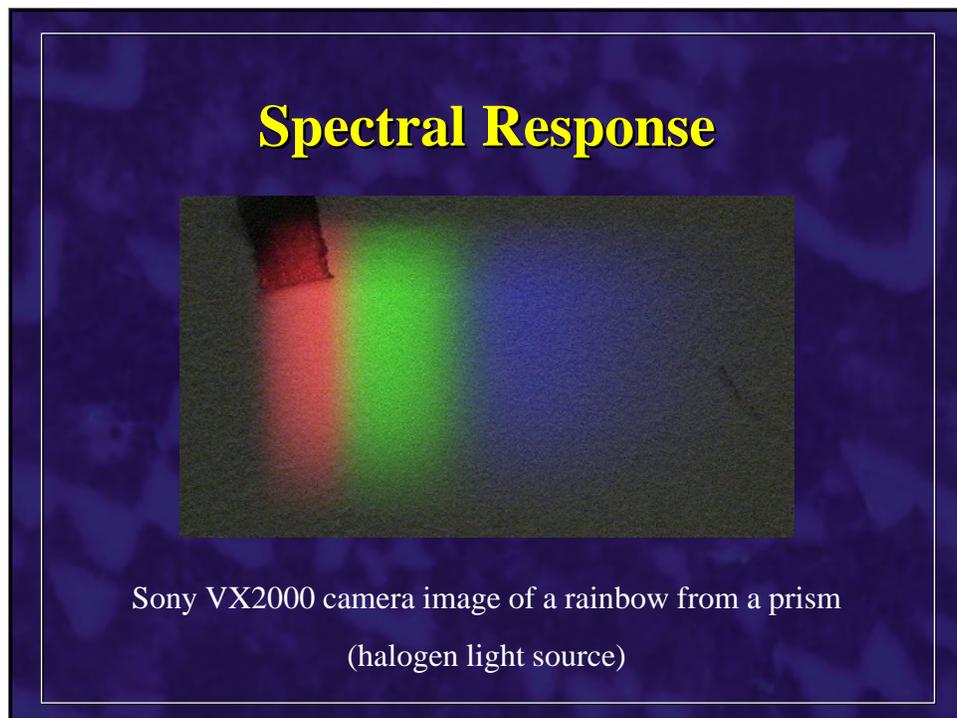
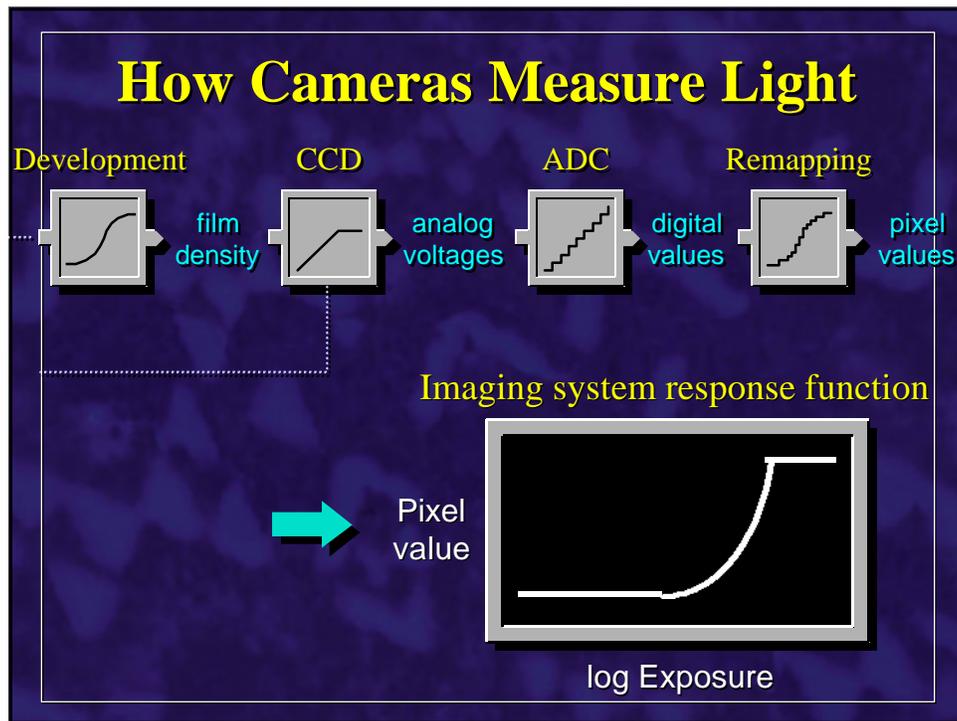
(see Dan Lemmon's notes)

3. Capturing, Representing, and Manipulating High Dynamic Range Imagery (HDRI)

How Cameras Measure Light



The Image Acquisition Pipeline



Camera Imperfections

Lens Flare

Vignetting

Sensor bleed

Noise

Reciprocity Failure

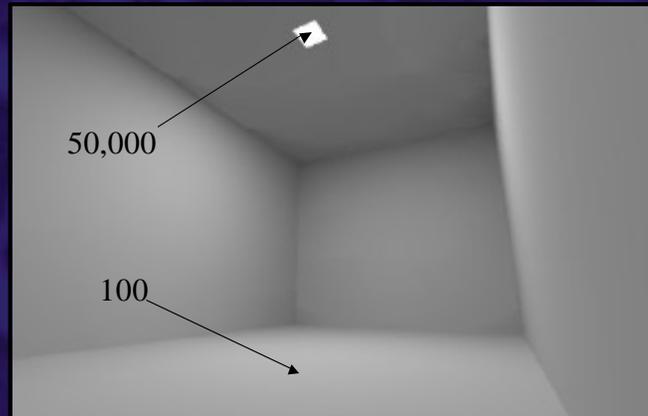
Problem: The World is High Dynamic Range

Dynamic Range is the ratio of the brightest accurately representable value to the dimmest

Light is measured in Radiance ($\text{W}/\text{sr}/\text{m}^2$);
Luminance regards only the *visible* energy.

The ratio of the luminance of the surface sun to the luminance of the numbers on a glow-in-the-dark watch is 10^{11} , or one hundred billion to one, or thirty-seven powers of 2.

Concentrated Light Sources generate High Dynamic Range

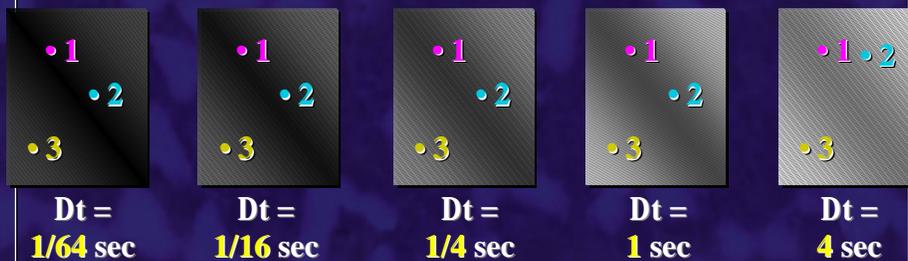


10' x 15' x 9' room, 9" by 9" light, 50% reflective walls

High Dynamic Range Photography

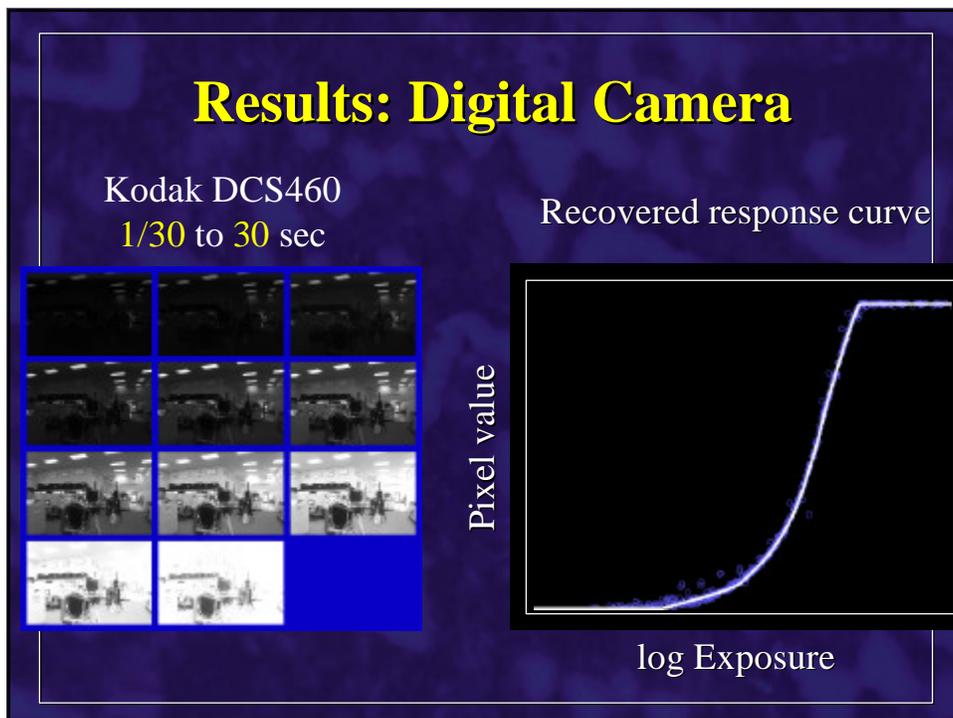
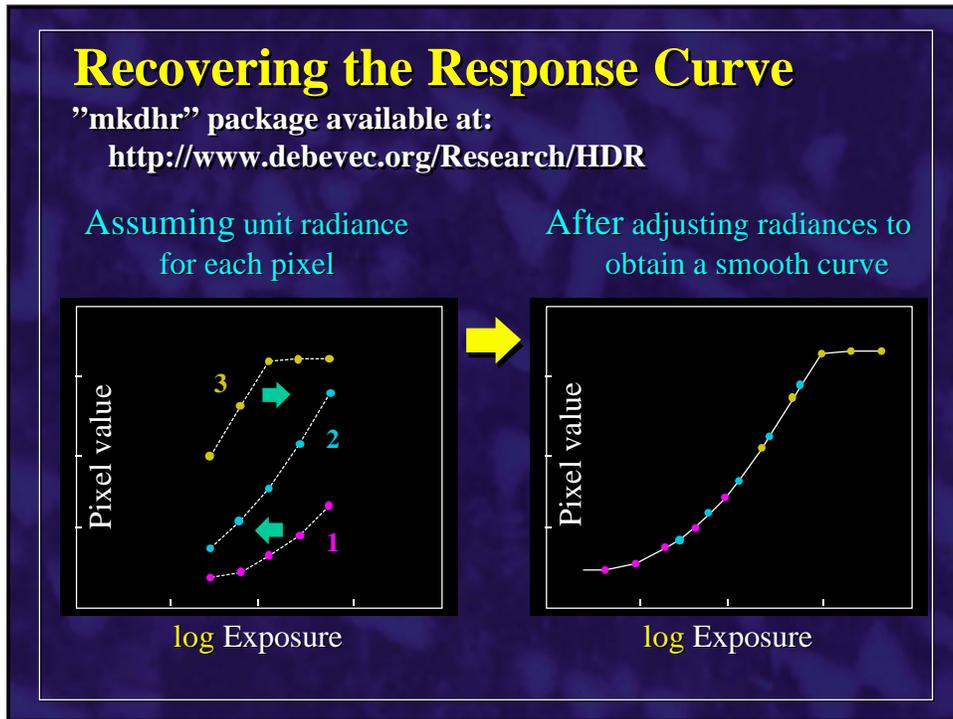
Debevec and Malik. Recovering High Dynamic Range Radiance Maps from Photographs. SIGGRAPH 97

Image series



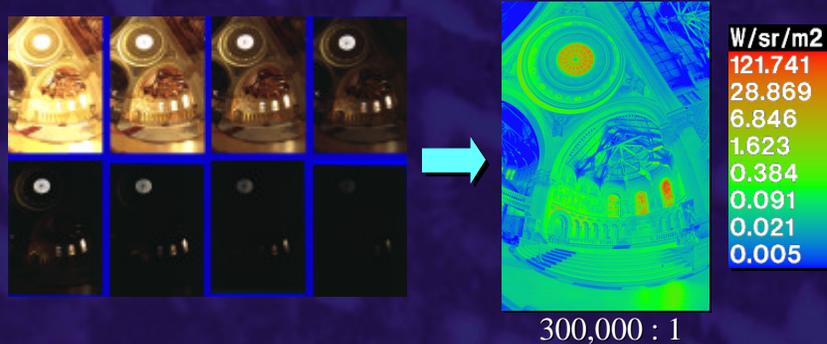
$$\text{Exposure} = \text{Radiance} \times \Delta t$$

$$\log \text{Exposure} = \log \text{Radiance} + \log \Delta t$$



High-Dynamic Range Photography

"mkdhr" beta package available at:
<http://www.debevec.org/Research/HDR>



HDR Shop Features

All images are internally represented linearly with three 32-bit floating point numbers per pixel

LDR images are imported using linearization

Displayed images are shown with proper gamma mapping, as are exported images

All image operations occur in linear space

User can interactively adjust the displayed exposure of the image

High Dynamic Range Images may be assembled from a series of varying exposures

Images may be saved in HDR and LDR formats

HDR Shop Features (2)

LDR slices of HDR images may be edited in standard image editing programs and re-integrated into the HDR image

Can convert between different panoramic mappings

Can assemble and manipulate light probe images

Can perform diffuse integrations of spherical mappings

High Dynamic Range Image Formats

Greg Ward's RADIANCE format (.pic, .hdr)

Raw Floating Point (.float)

Portable FloatMap (.pfm)

Newtek Flexible Image Format (.flx)

Floating-Point TIFF (.tif)

Greg Larson's LogLuv TIFF (.tif)

RADIANCE Format

Greg Larson's "Real Pixels" format



$$(145, 215, 87, 149) =$$

$$(145, 215, 87) * 2^{(149-128)} =$$

$$(1190000, 1760000, 713000)$$

$$(145, 215, 87, 103) =$$

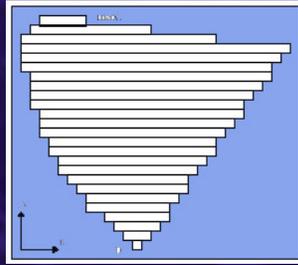
$$(145, 215, 87) * 2^{(103-128)} =$$

$$(0.00000432, 0.00000641, 0.00000259)$$

Ward, Greg. "Real Pixels," in Graphics Gems IV, edited by James Arvo, Academic Press, 1994

LogLuv Format

based on human color perception



24 bits: 10 for log luminance
 14 for chromaticity index
 32 bits: 15 log luminance
 8 u chrominance
 8 v chrominance
 1 sign

Larson, G.W., "Overcoming Gamut and Dynamic Range Limitations in Digital Images," Proceedings of the Sixth Color Imaging Conference, November 1998.

<http://positron.cs.berkeley.edu/~gwlarson/pixformat/tiffluv.html>

Portable FloatMap

12 bytes per pixel, 4 for each channel



Text header similar to Jeff Poskanzer's .ppm images:

```
PF
768 512
1
<binary image data>
```

8-bit Images

8-bit images are useful for representing pixels on screen or on a printer

If encoded linearly, they aren't perceptually useful

$$z = \text{floor}(255.9 \bar{\cdot} * \max(L, 1))$$

If encoded non-linearly, they can represent low-dynamic range images reasonably well

$$z = \text{floor}(255.9 * \max(g(L), 1))$$

8-bit Images

Linear encoding: for any pixel value < 100 , the relative brightness steps are $> 1\%$ and thus noticeable. Above 100, we sample more finely than necessary.

Thus we encode pixels non-linearly. What might make the most sense is a logarithmic encoding, where the next pixel value is 1% brighter than the previous. But this would only allow a dynamic range of 12.6:1

$$g(L) = k \log_{1.01}(L)$$

8-bit Images

If we takes steps of 2.1%, we can encode a dynamic range of 200:1 in 8 bits, but the steps will begin to be noticeable, and there's a precipitous drop in accuracy at the low end

In practice, a different nonlinear encoding called a *gamma curve* is used

$$g(L) = L^{1/\gamma}$$

8-bit Images

Most often, we encode images with $\gamma = 2.2$. This is done for most TIFs, BMPs, and JPEGs.

This gives a better than 1% relative accuracy near the high end, which degrades smoothly until the low end; by pixel value 23 the relative accuracy is 10%.

But since a pixel value of 255 is two hundred times as bright as a pixel value of 23, the steps are often unnoticeable.

However, for dark scenes on good displays, the steps will be noticeable

8-bit Images

As it turns out, CRT monitors automatically “undo” the nonlinear gamma mapping, and that’s in fact the original reason for having it in the first place

In fact, monitors by default somewhat overcorrect with a gamma of $\gamma = 2.5$

This contrast enhancement has been found to be desirable for viewing images on a monitor in an illuminated room

8-bit Images

Very Important: It is necessary to re-map nonlinear pixel value encodings into a linear space before performing mathematical transformations.

$$a^\gamma + b^\gamma \neq (a+b)^\gamma$$

$$\alpha(b^\gamma) \neq (\alpha b)^\gamma$$

Unfortunately, this is done incorrectly almost all the time throughout computer graphics.

8-bit Images

Also Very Important: It is necessary to map linear pixel value encodings into a gamma-corrected space before displaying on a monitor.

Unfortunately, this is also done incorrectly almost all the time throughout computer graphics.

8-bit Images

And finally, also Very Important: most image transformations either require or benefit from high-precision, high dynamic range pixel values.

16-bit Images

Supported to some extent by many image-editing programs (Photoshop, GIMP)

These greatly increase the precision that can be represented, eliminating roundoff problems

But the highest value (65535) is still taken to mean “white” => no High Dynamic Range

4. Capturing Real-World Illumination

Mirrored Ball - Records light in all directions

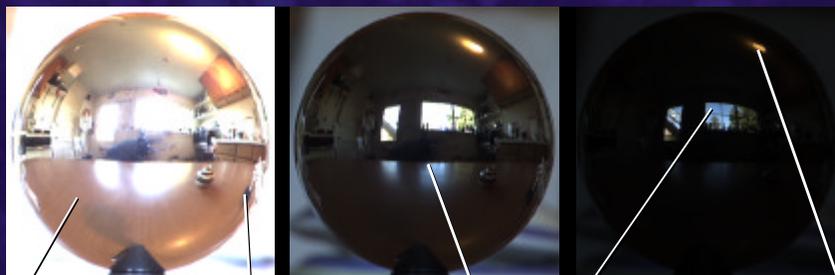


Brightest regions
are saturated



Intensity and color
information lost

Radiance Map of a Mirrored Ball



(60,40,35)

(18,17,19)

(620,890,1300)

(5700,8400,11800)

(11700,7300,2600)

Assembled from ten digital images,
 $\Delta t = 1/4$ to $1/10000$ sec

Sources of Mirrored Balls

2-inch chrome balls < \$20 ea.

King Bearing, Inc.

Applied Industrial Technologies

(many locations nationally, check www.bigbook.com)

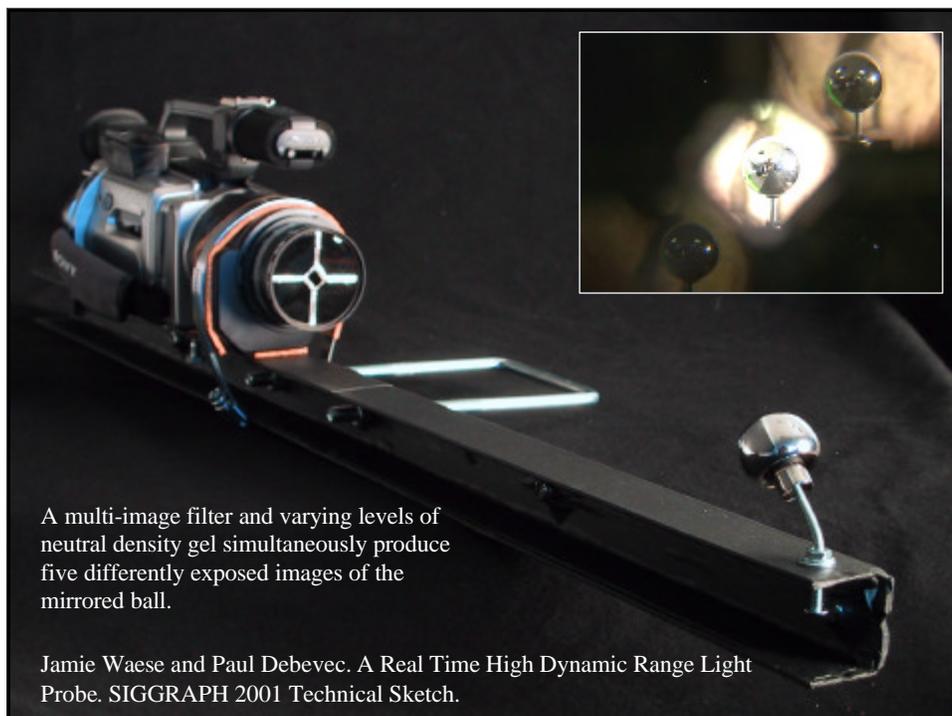
6-12 inch large gazing balls (blown glass)

Baker's Lawn Ornaments

570 BERLIN PLANK ROAD

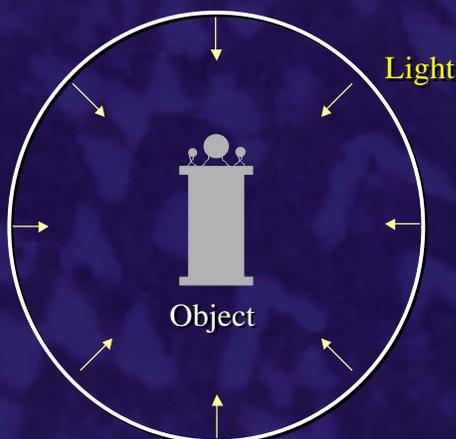
SOMERSET, PA 15501-2413

814-445-7028



5. Illuminating Synthetic Objects with Real Light

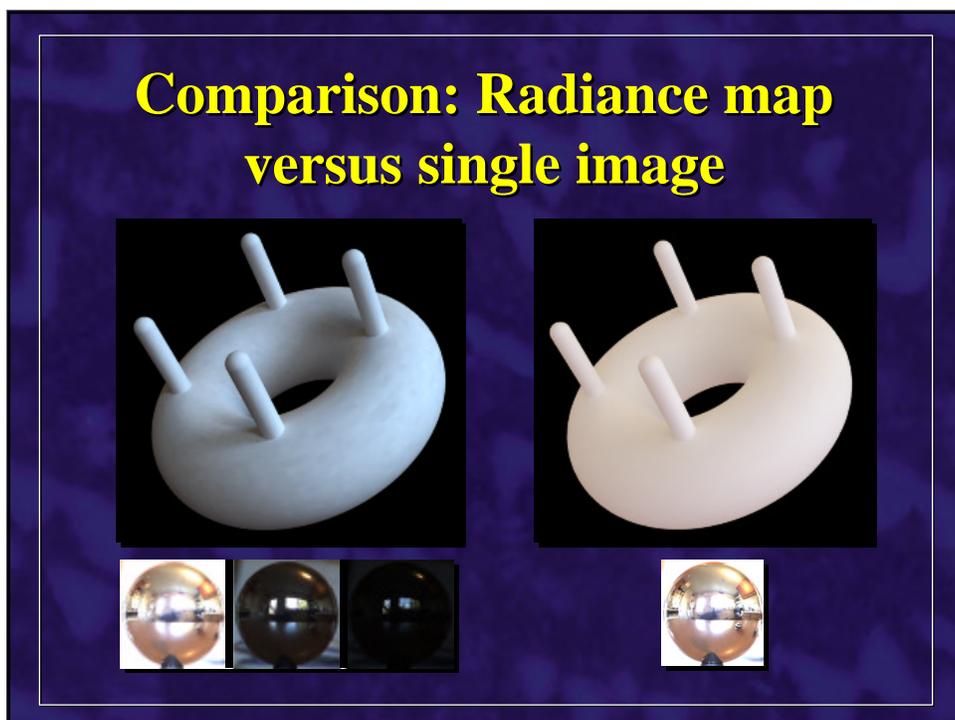
Illuminating Objects using Measurements of Real Light

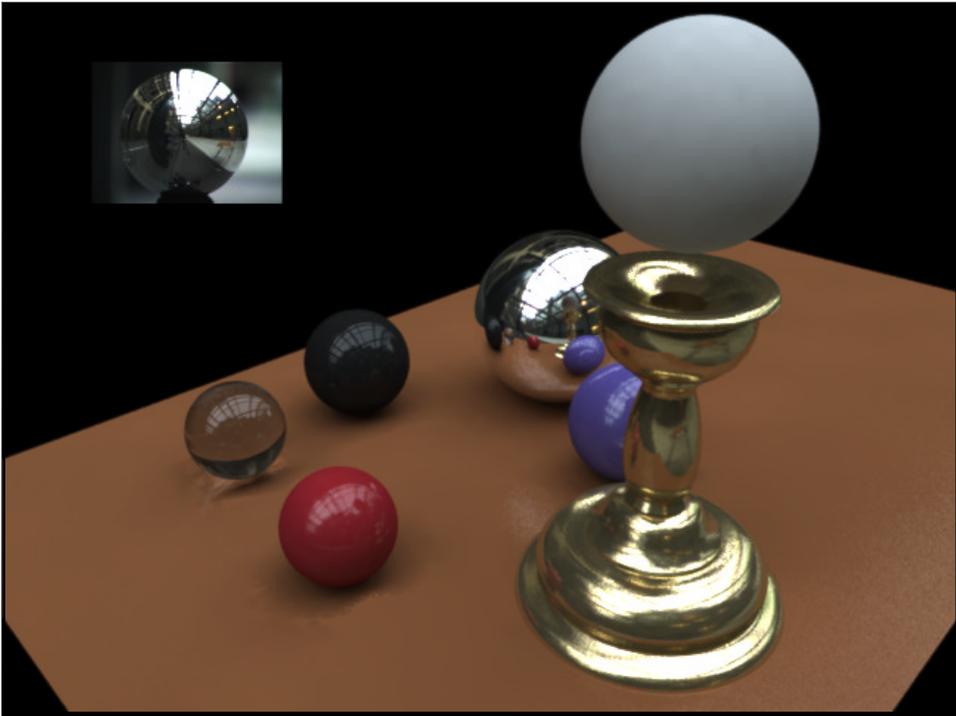
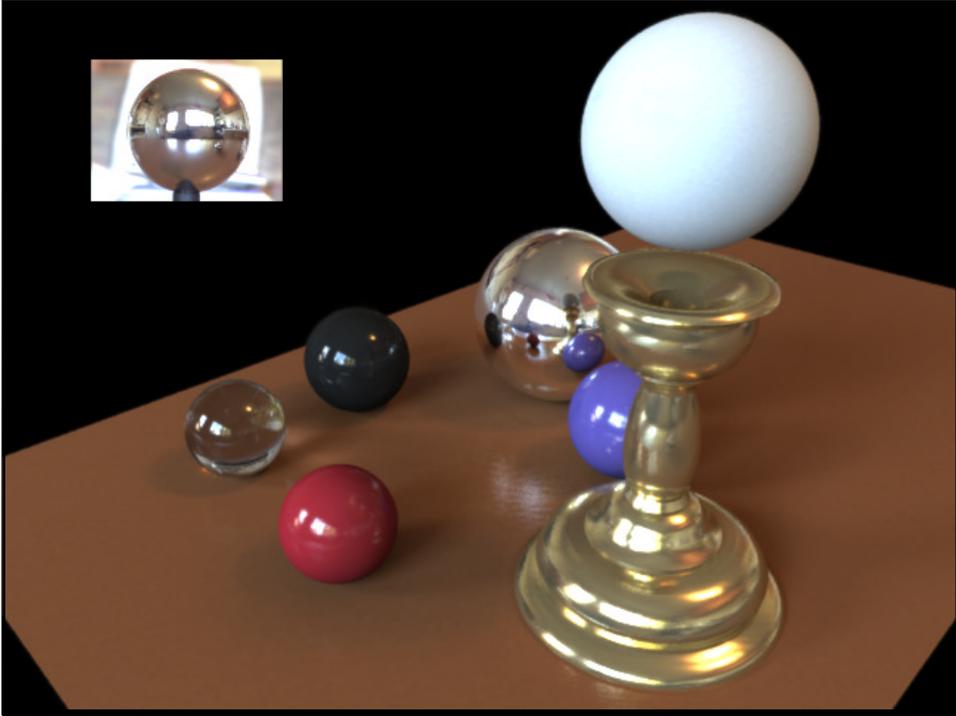


*Environment assigned "glow" material property in Greg Larson's **RADIANCE** system.*

<http://radsite.lbl.gov/radiance/>

See also: Larson and Shakespeare, "Rendering with Radiance", 1998





Reflection Mapping - 1982



Mike Chou and Lance Williams

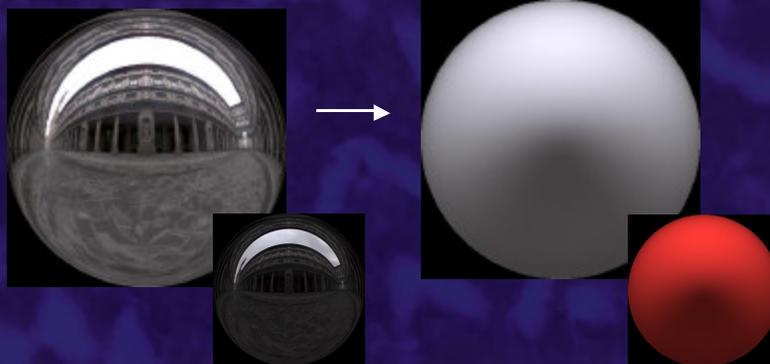


Gene Miller and Ken Perlin

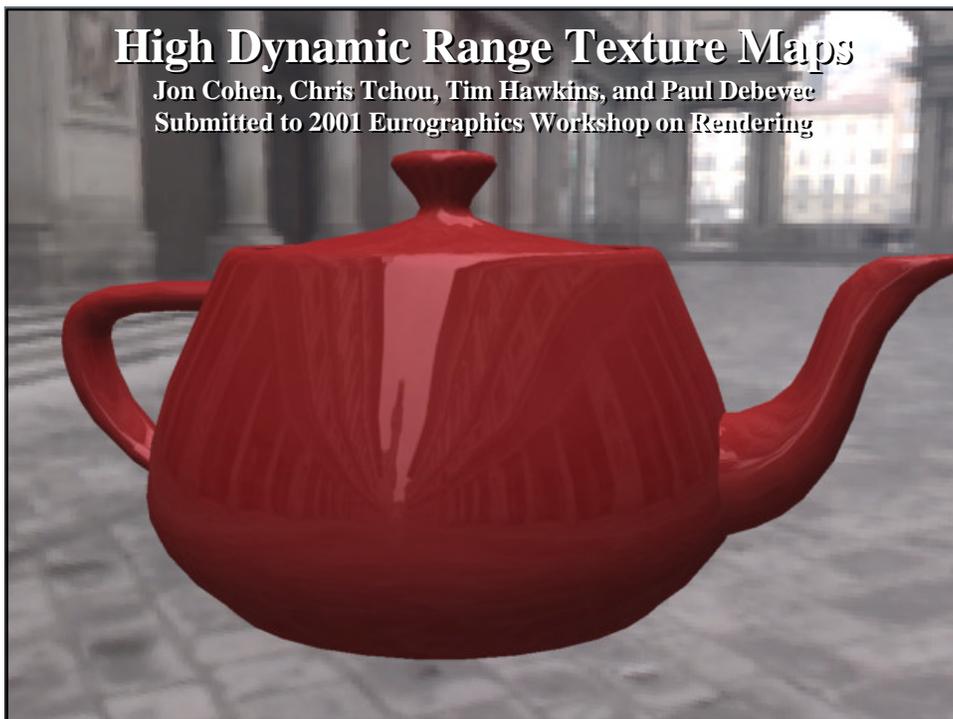
Today: can perform in real time with graphics hardware

<http://www.debevec.org/ReflectionMapping/>

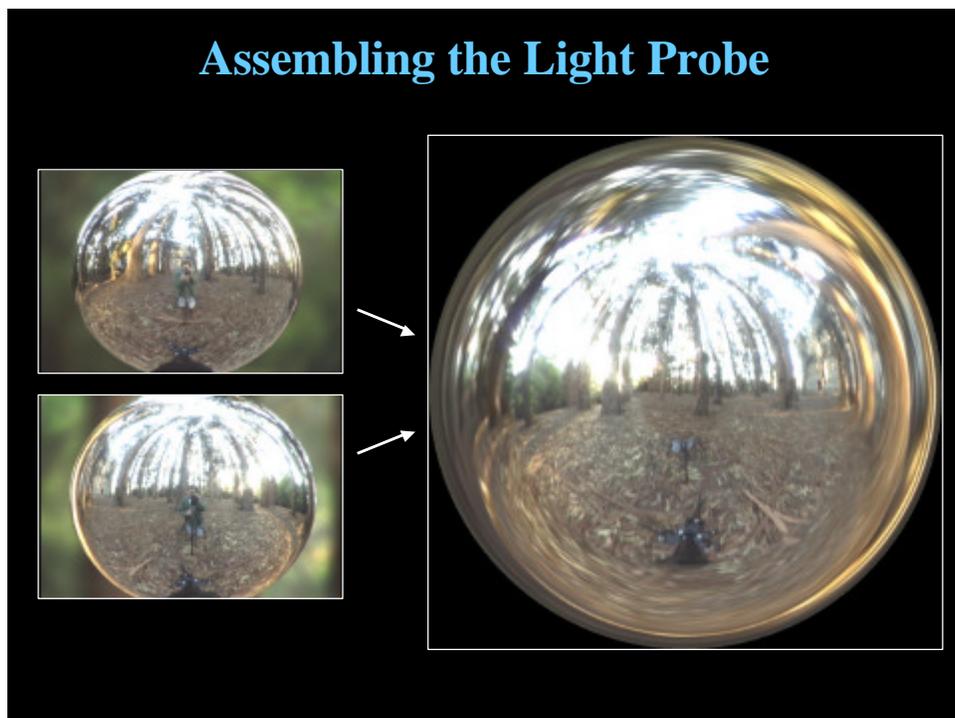
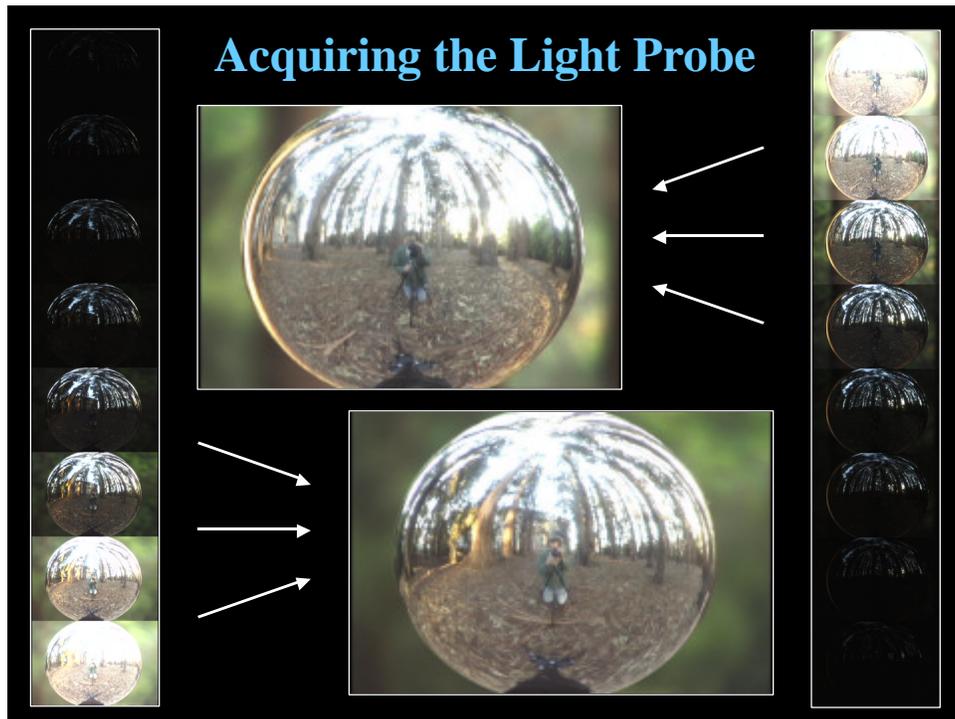
Real Time: Diffuse Pre-Convolution

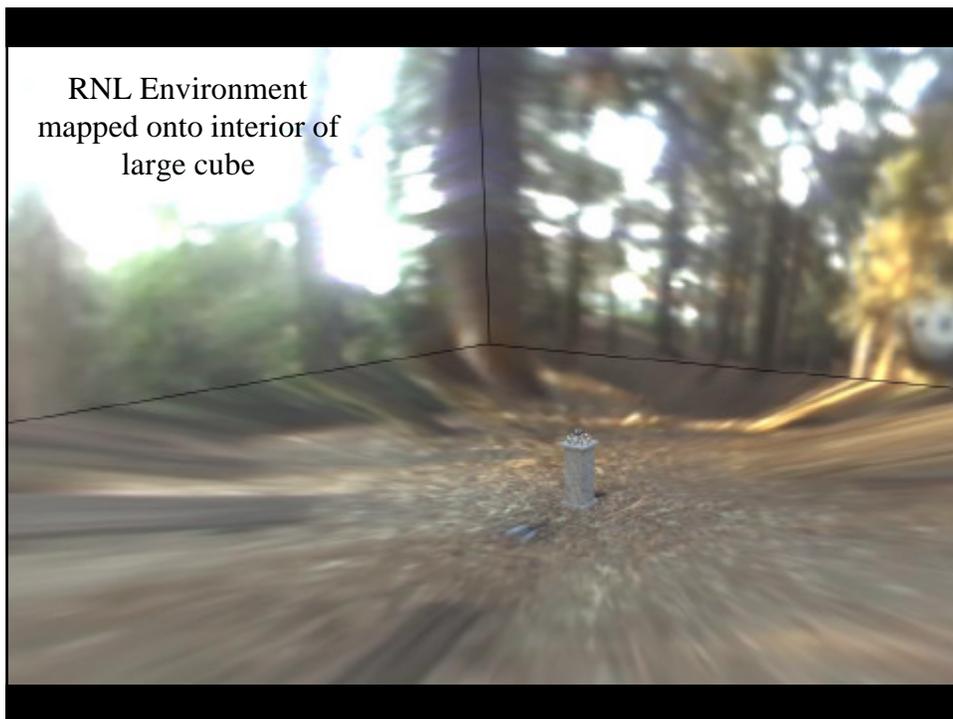


See: Gene S. Miller and C. Robert Hoffman. *Illumination and Reflection Maps: Simulated Objects in Simulated and Real Environments*. Course Notes for Advanced Computer Graphics Animation, SIGGRAPH 84.



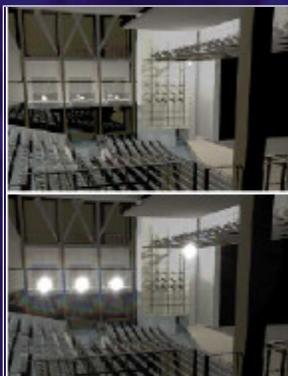
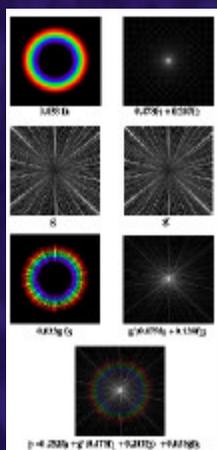
SIGGRAPH 98 Electronic Theater

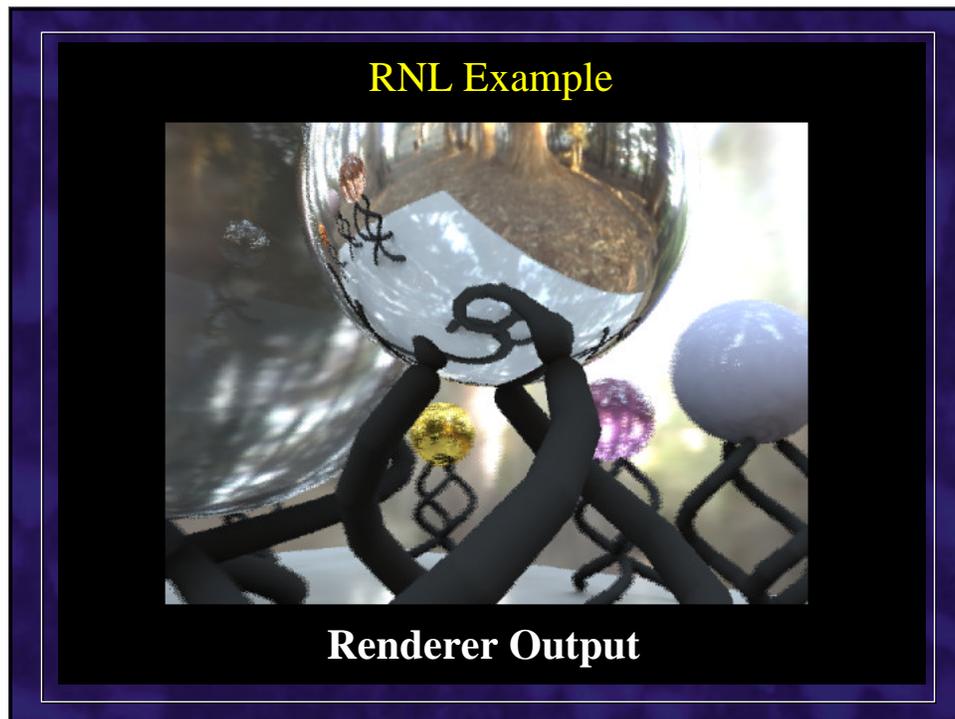




Simulating Glare

Greg Spencer, Peter Shirley, Kurt Zimmerman,
and Donald Greenberg. Physically-based glare
effects for digital images. SIGGRAPH 95.





RNL Example



Soft Focus Added

RNL Example



Light Falloff (Vignetting) Added

Motion Blur



Normal digitized photo Synthetic blur added

This slide illustrates the effect of synthetic motion blur. On the left is a sharp, normal digitized photograph of the interior of a large cathedral, showing a dome with a wooden structure and several arched windows. On the right is the same image with synthetic motion blur applied, resulting in horizontal streaking and a loss of fine detail.

Motion Blur



Blurred radiance map, virtually rephotographed Actual blurred photograph

This slide compares synthetic motion blur with actual motion blur. On the left is a blurred radiance map, which is a virtual representation of the scene's lighting and color, showing horizontal streaking. On the right is an actual blurred photograph of the same cathedral interior, which shows a similar but more natural-looking motion blur effect.

7. Rendering Synthetic Objects into Real Scenes

CGI / Background Plate Compositing

Need to match:

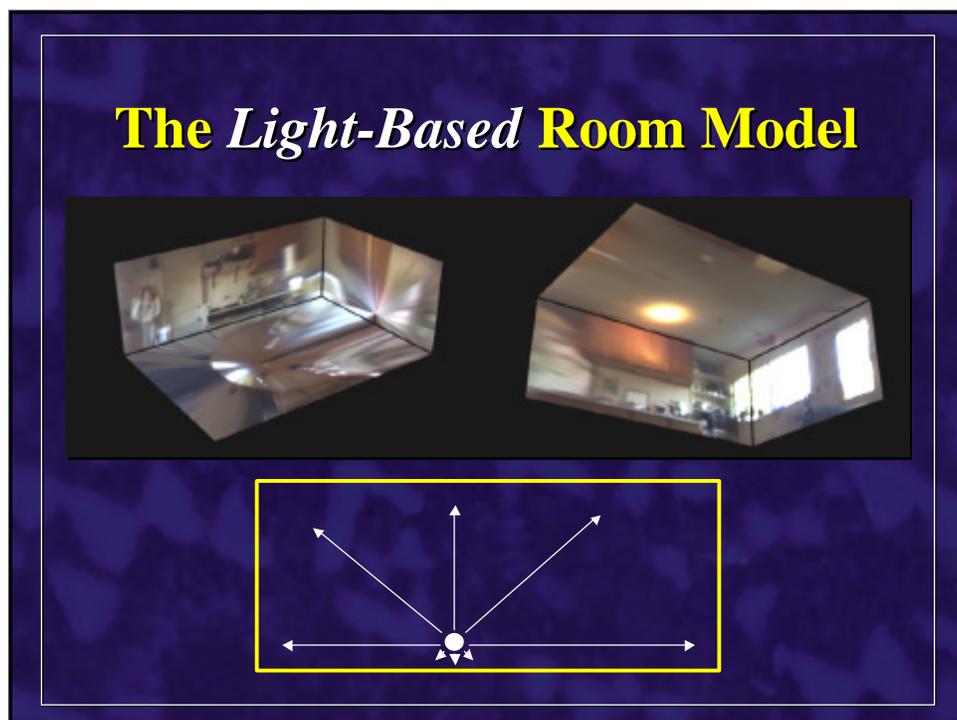
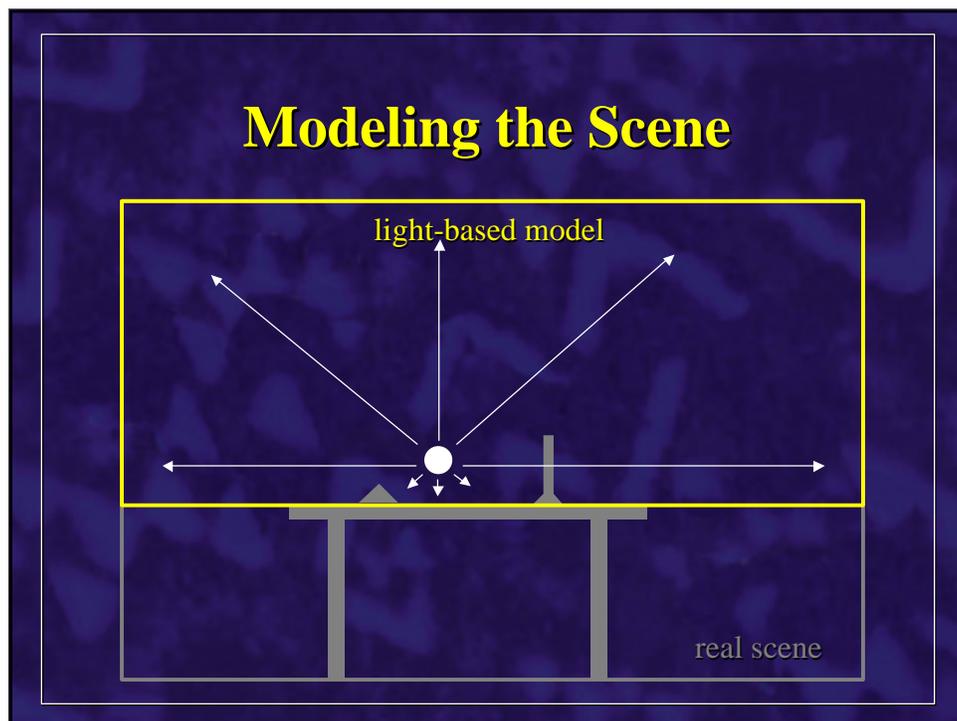
- **Camera Parameters**
 - Pose, Focal length, Distortion, Focus
- **Film Response**
 - Contrast, Toe & Shoulder, Spectral Transfer
- **MTF / Film Grain**
 - Modulation Transfer Function, Ag Particles
- ➡ • **Illumination**
 - Highlights, Reflections, and Shadows

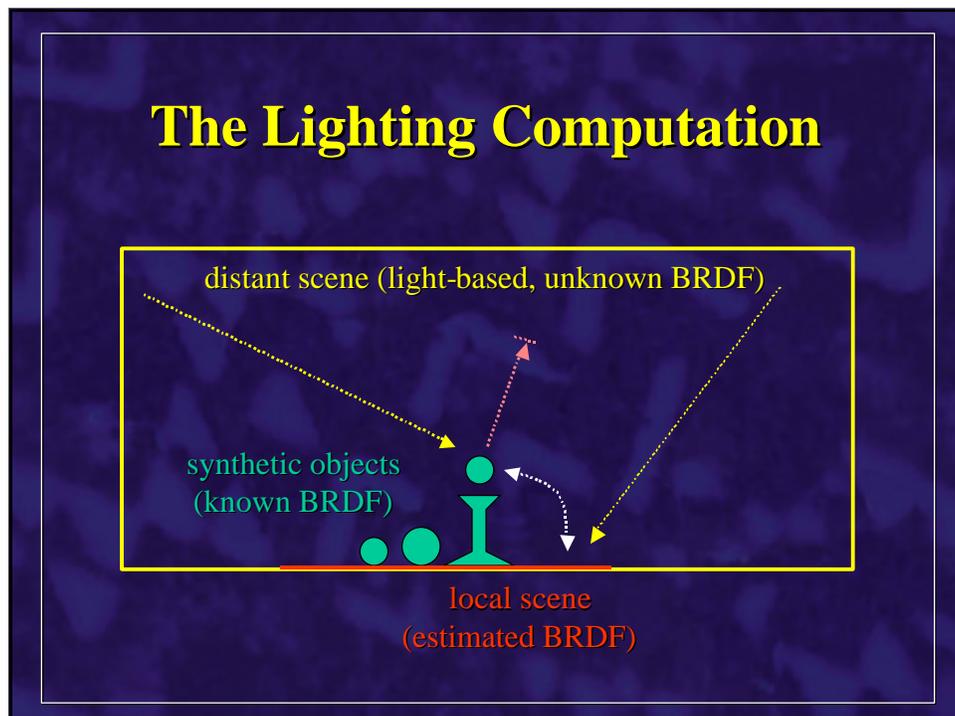
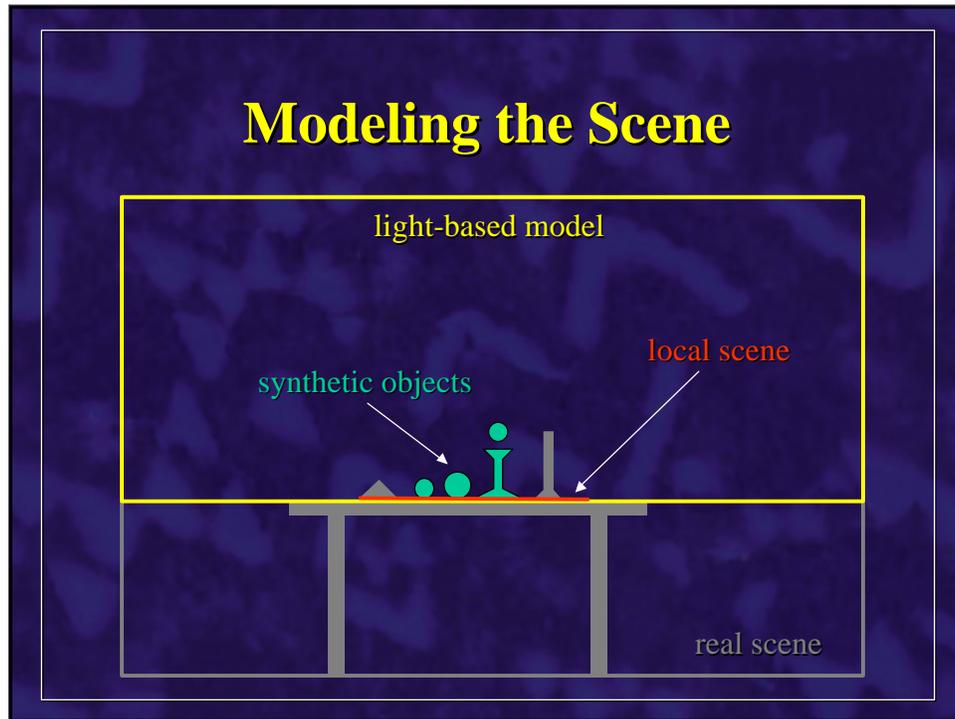
Compositing Objects into a Scene



Light Probe / Calibration Grid





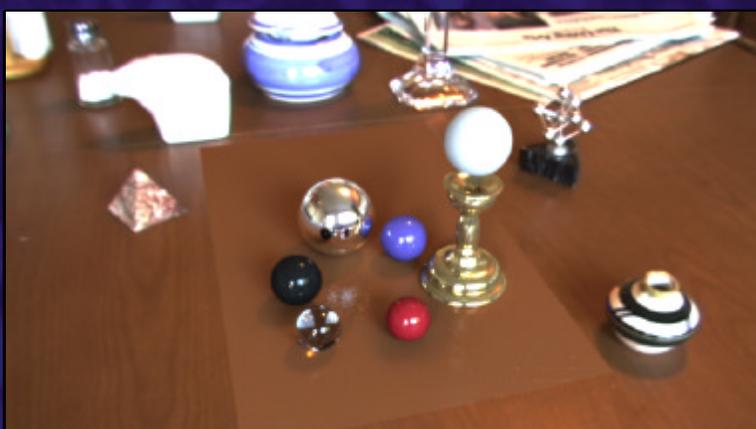


Rendering into the Scene



Background Plate

Rendering into the Scene



Objects and Local Scene matched to Scene

Differential Rendering



Local scene w/o objects, illuminated by model

Differential Rendering (2) Difference in local scene



Differential Rendering (3)



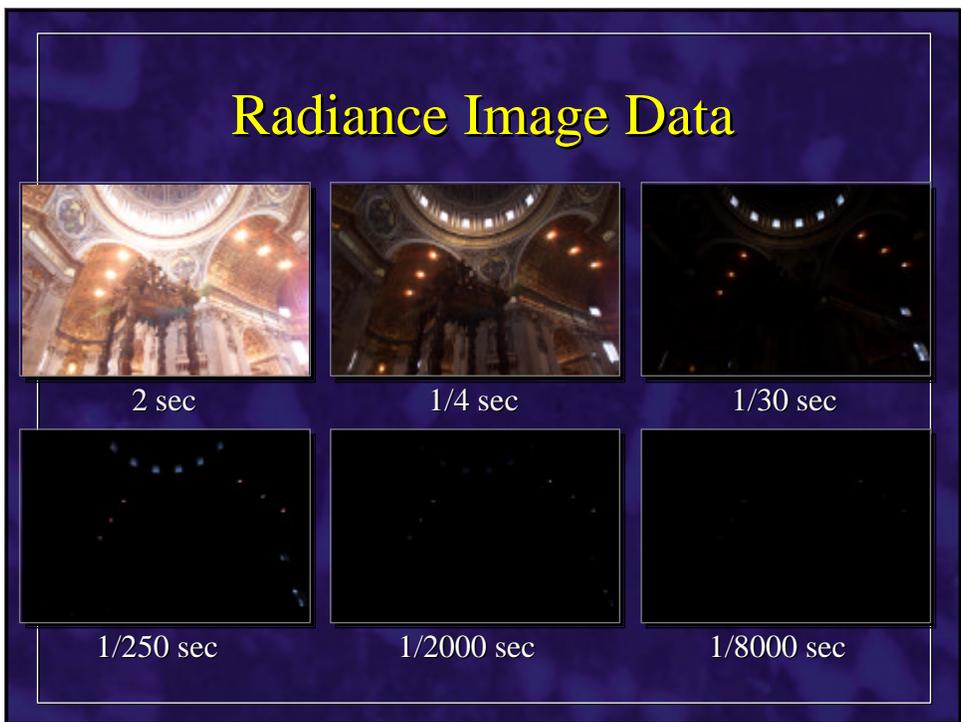
Final Result

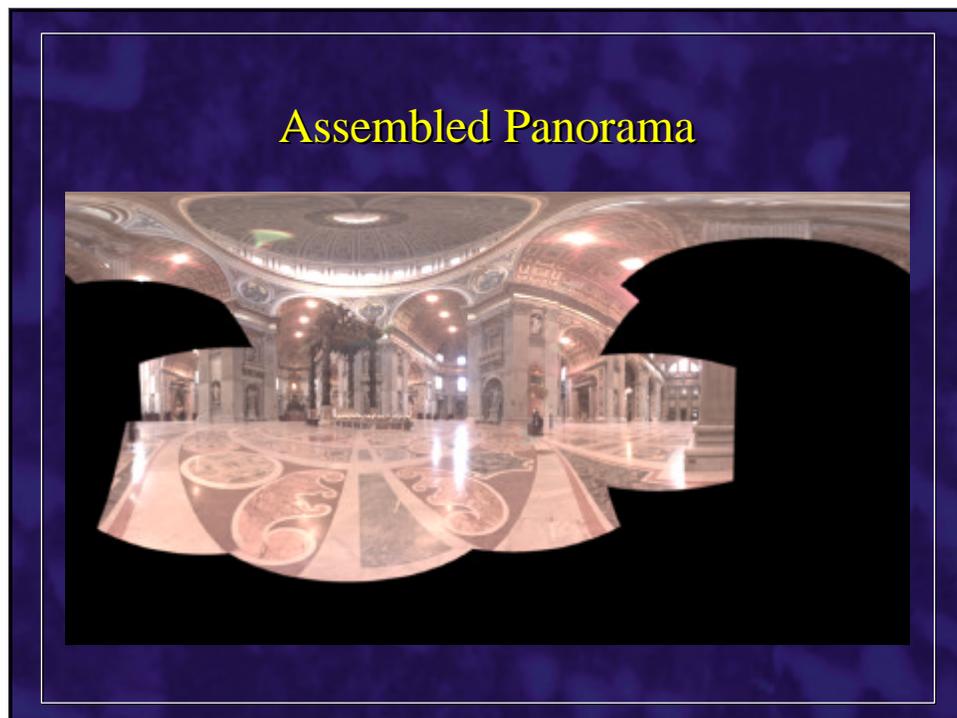


Video

**Domino animation rendered by
Son Chang and Christine Waggoner**

8. Image-Based Lighting in *Fiat Lux*



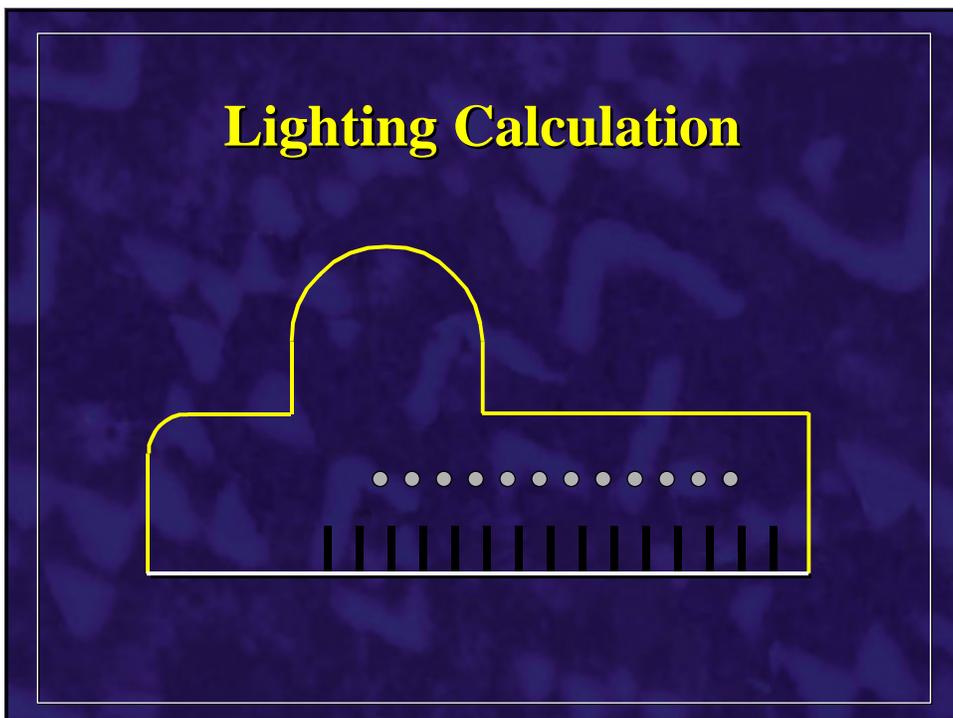
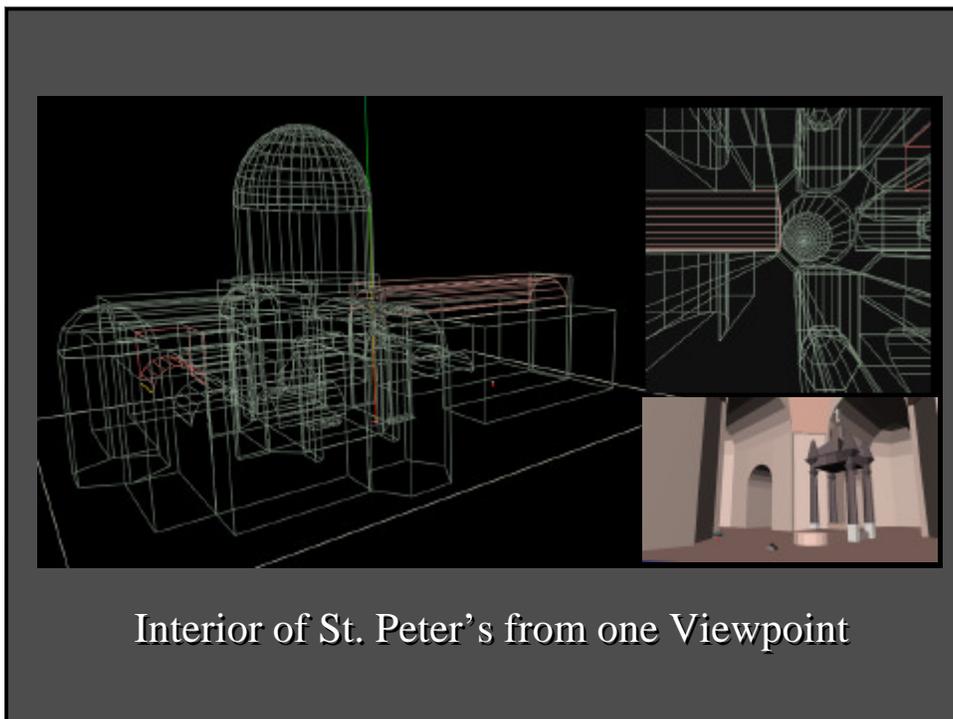


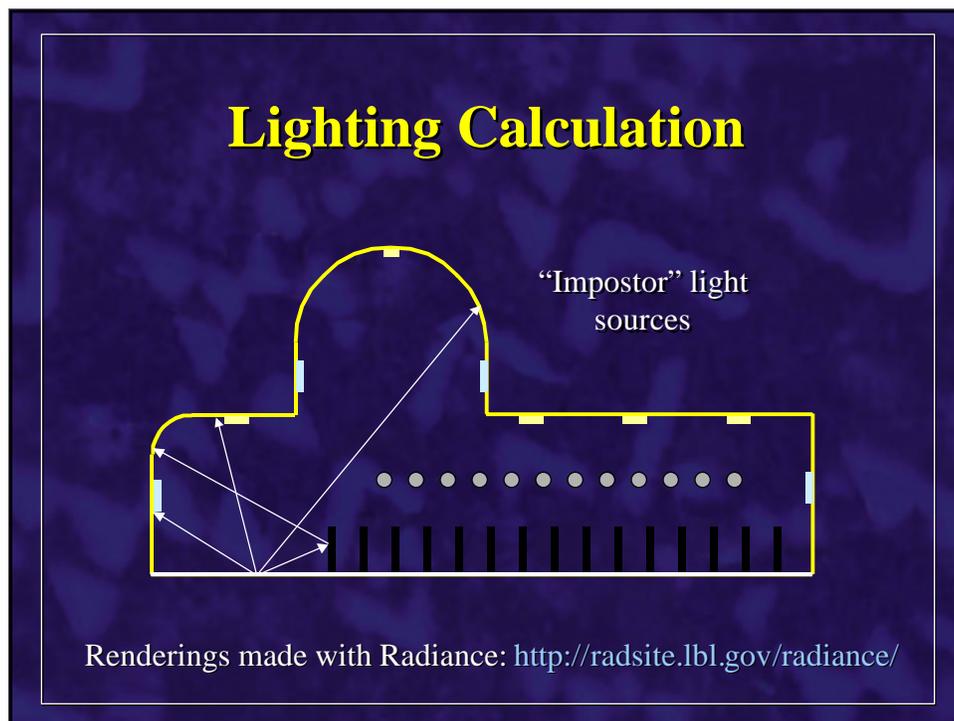
Processed Panorama

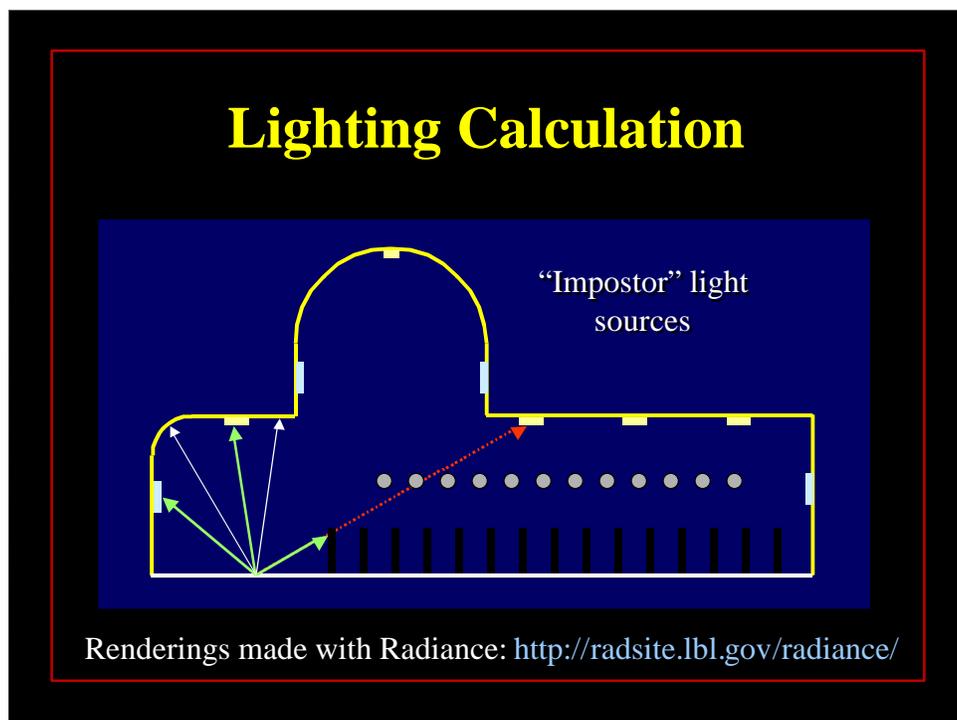
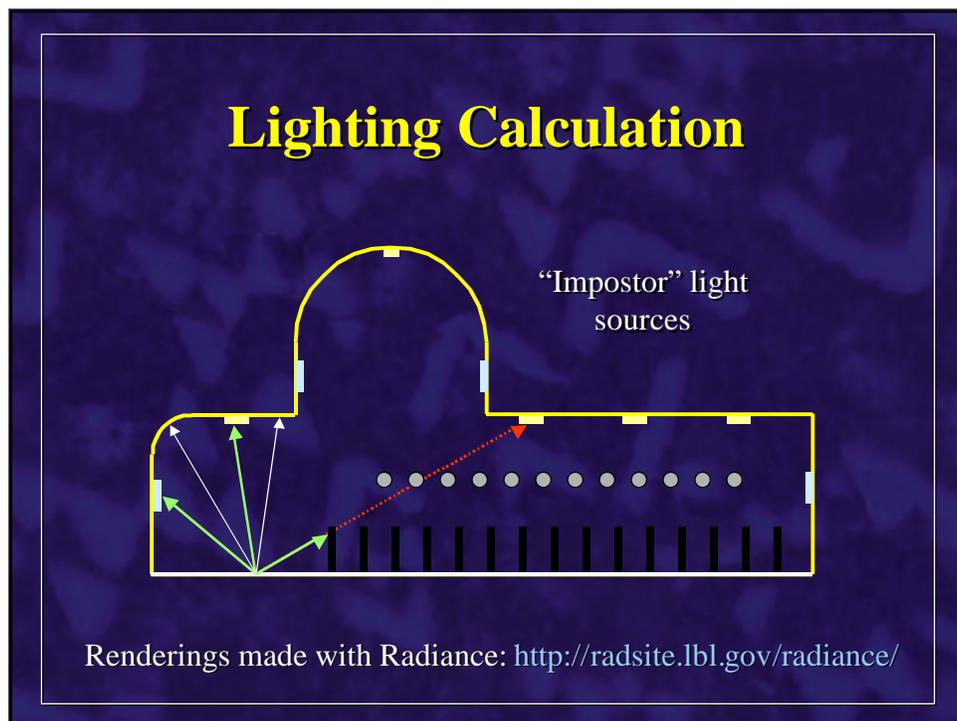


Light Probe Images









9. Image-Based Lighting with Commercial Renderers and in Commercial Production

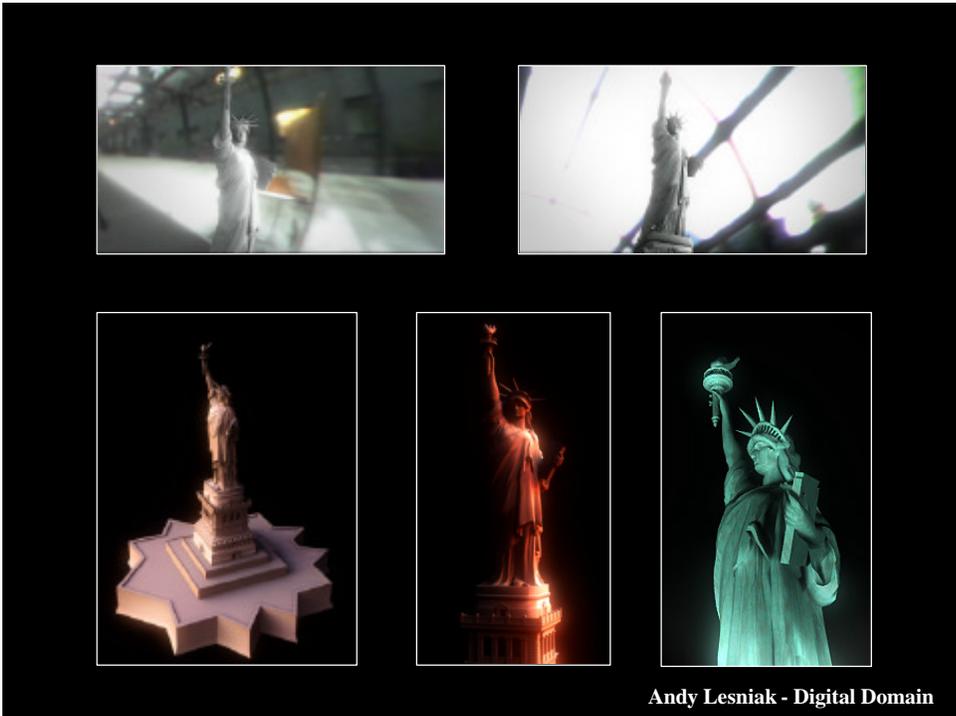
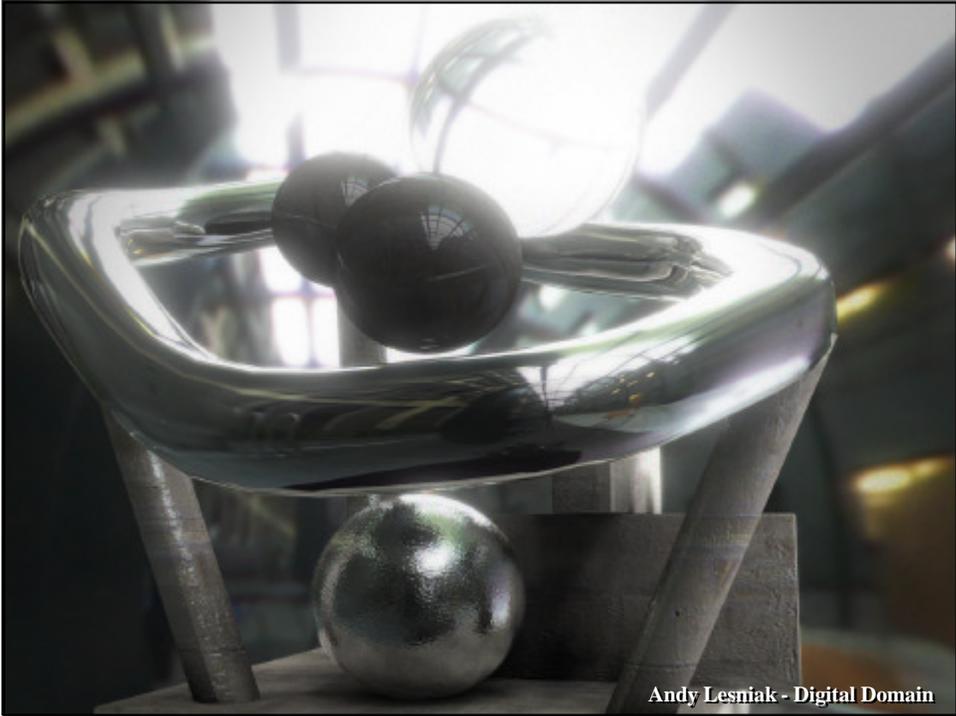


**Andy
Lesniak**
Digital Domain



Lightwave 3D 5.5

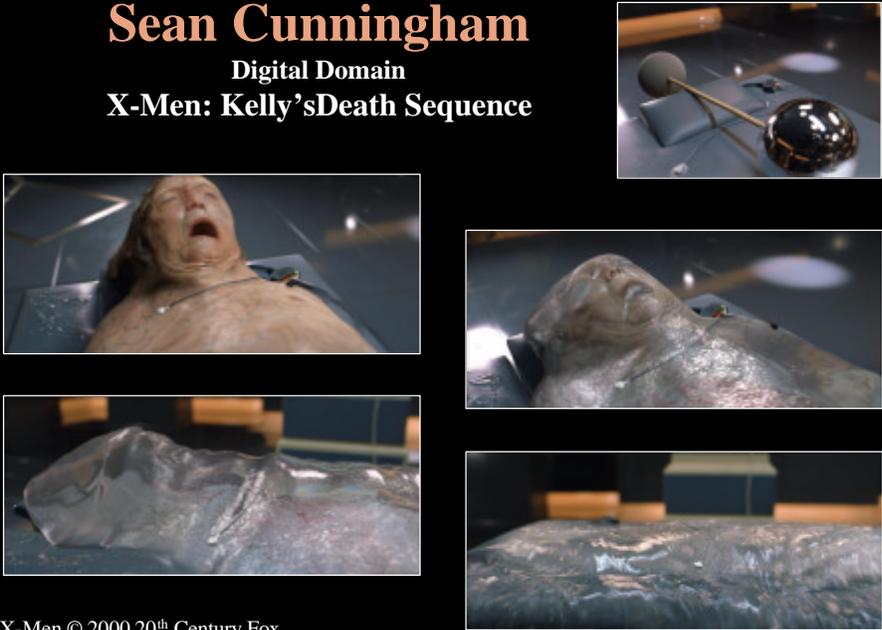






Sean Cunningham

Digital Domain
X-Men: Kelly's Death Sequence



X-Men © 2000 20th Century Fox

Joshua Kolden

Digital Domain
Experience Music Project



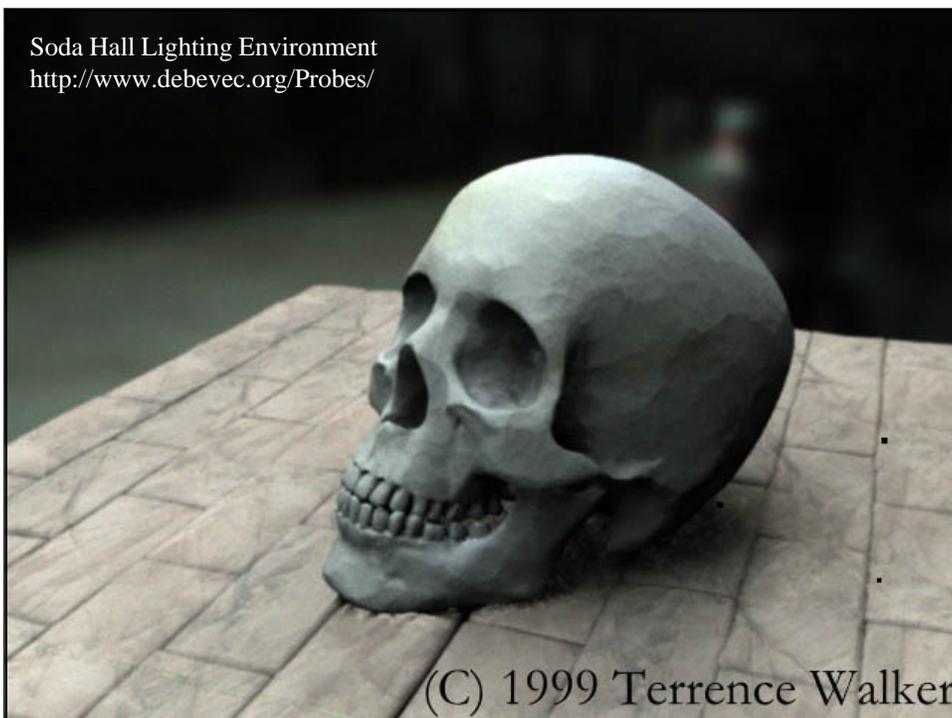
James Brown: Putting a New Face on the Godfather of Soul
Joshua Kolden, Mark David Brown, Andre Bustanoby

Terrence Walker:
HDRI and Image-Based Lighting
in LightWave 3D 6.0

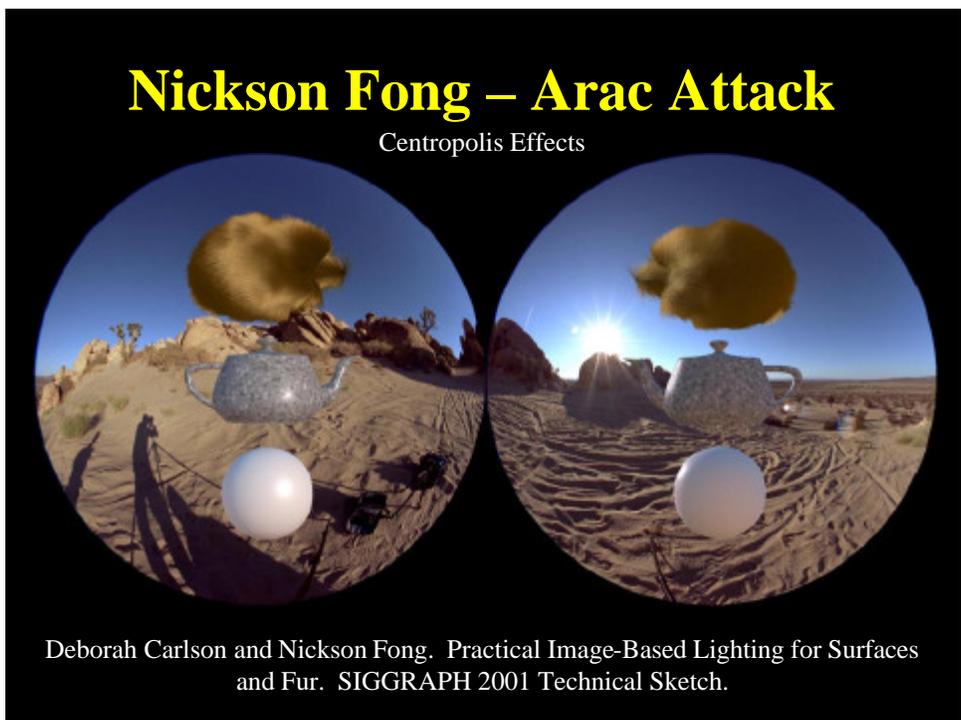


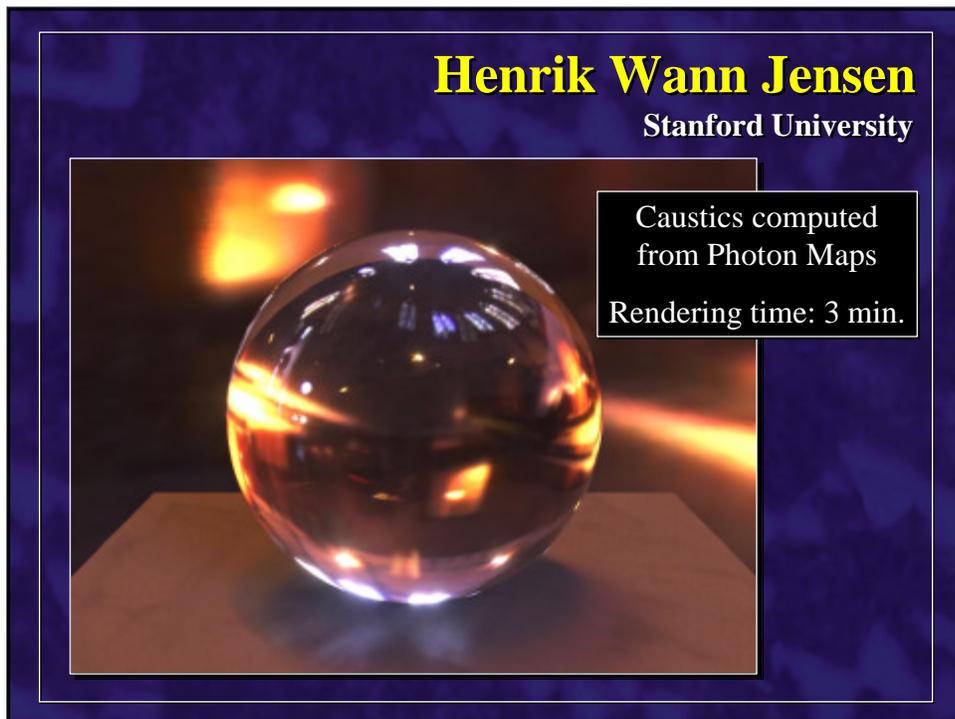
Lighting Environments from the [Light Probe Image Gallery](http://www.debevec.org/Probes/)
<http://www.debevec.org/Probes/>

Soda Hall Lighting Environment
<http://www.debevec.org/Probes/>

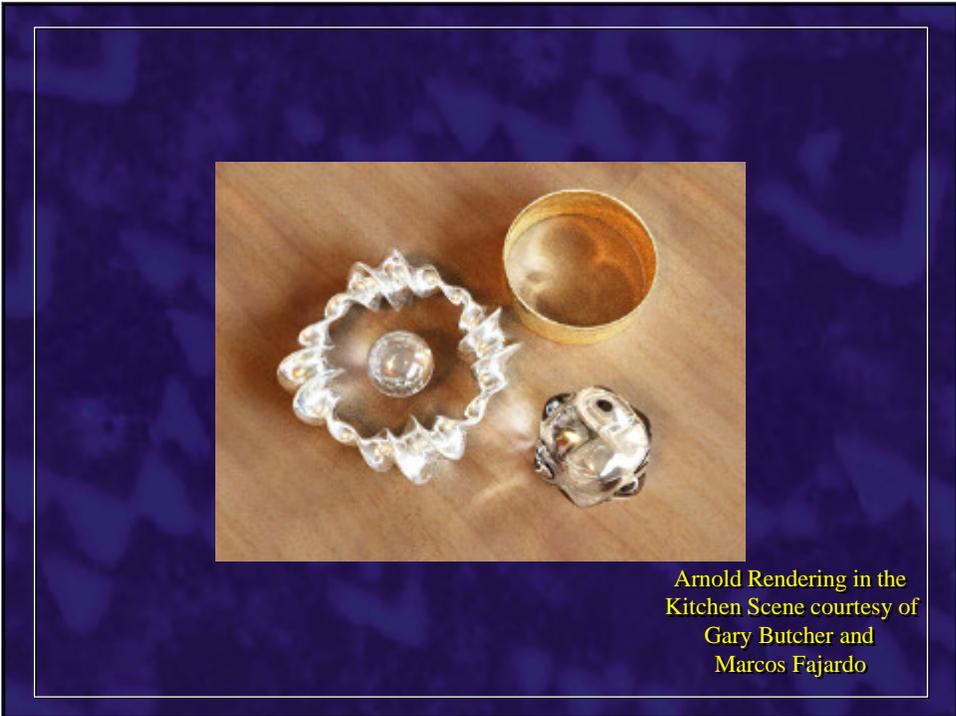
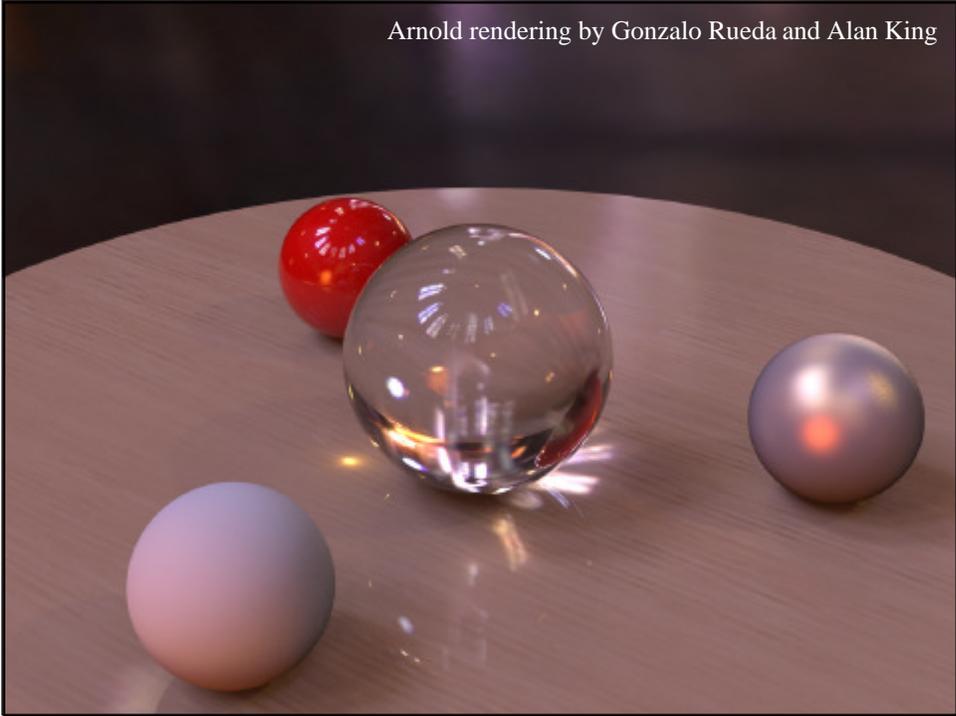


(C) 1999 Terrence Walker









Arnold Rendering in the Kitchen Scene courtesy of Gary Butcher and Marcos Fajardo



10. Image-Based Lighting Real Objects and Actors

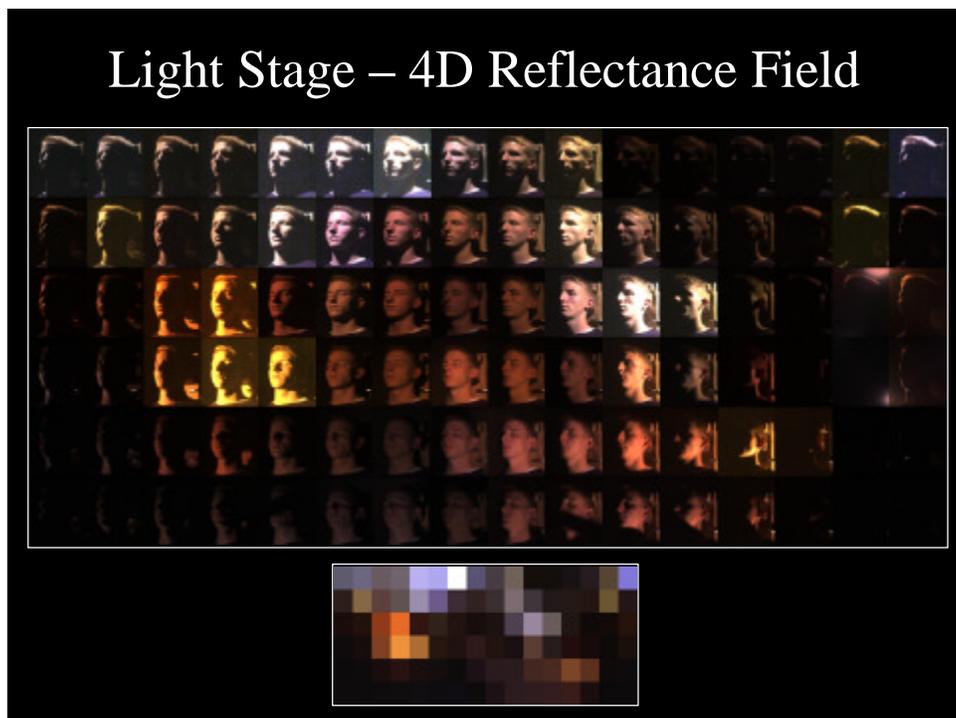
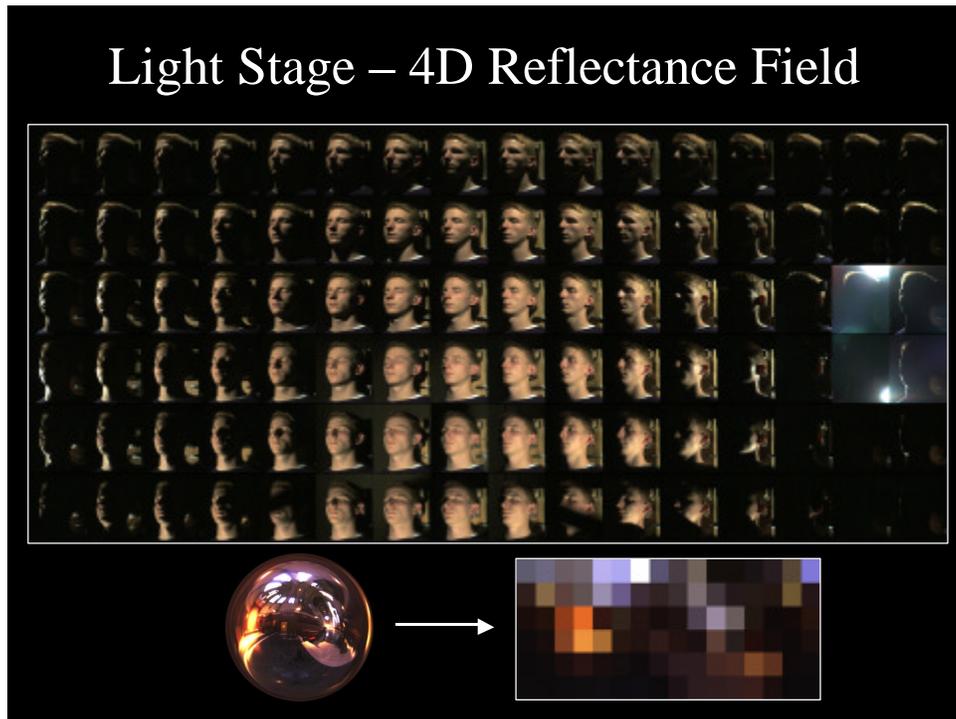
The Light Stage

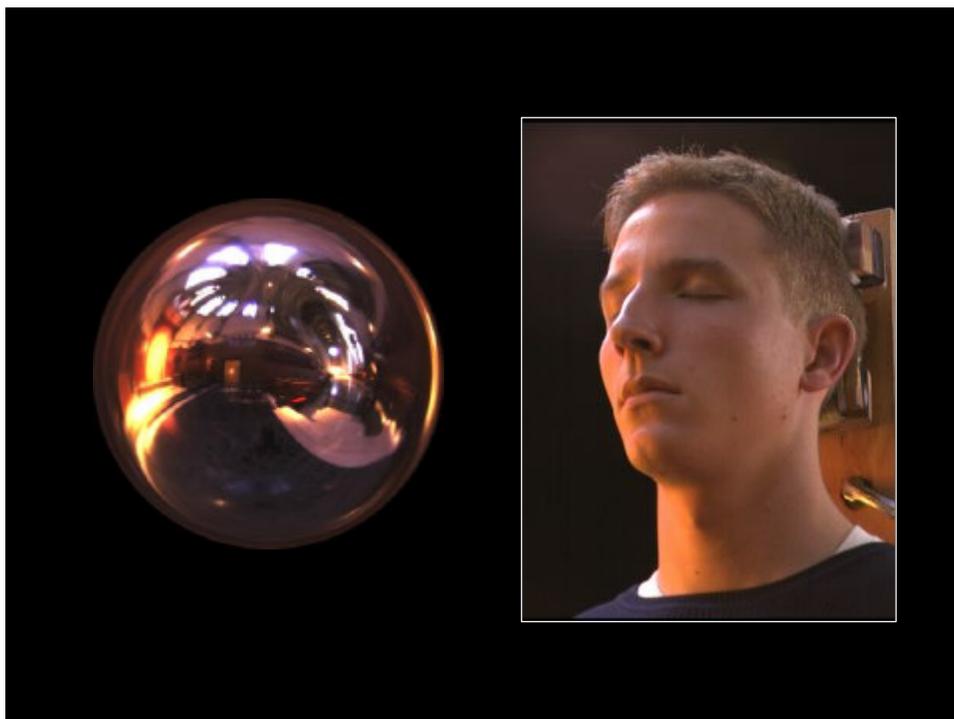


“Acquiring the Reflectance Field of a Human Face”
Wednesday, Late Afternoon, Hall E1-3

SIGGRAPH 2000 Creative Applications Laboratory





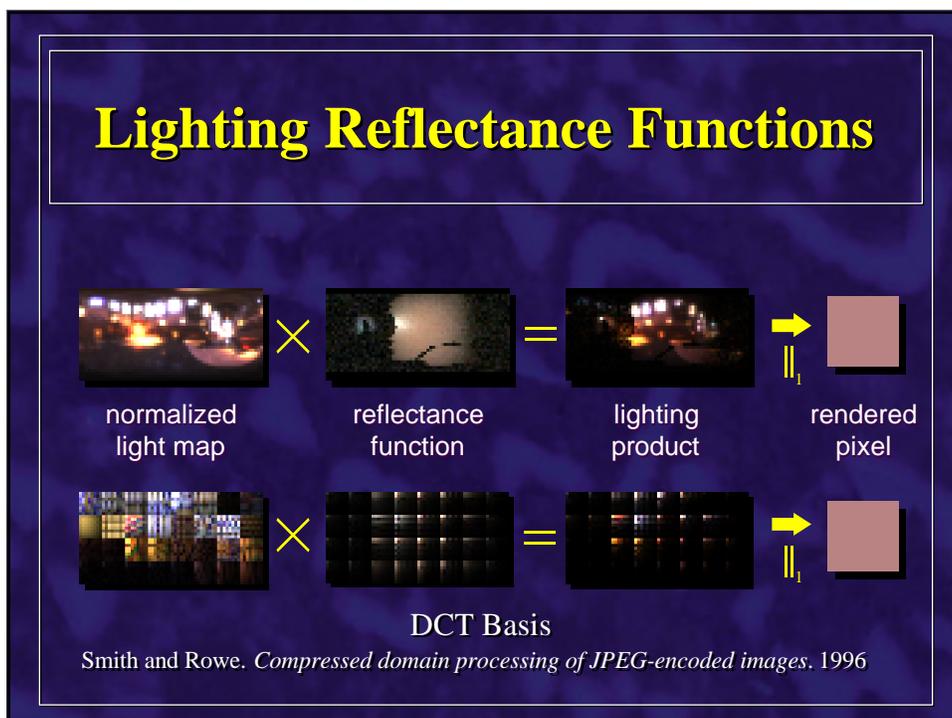
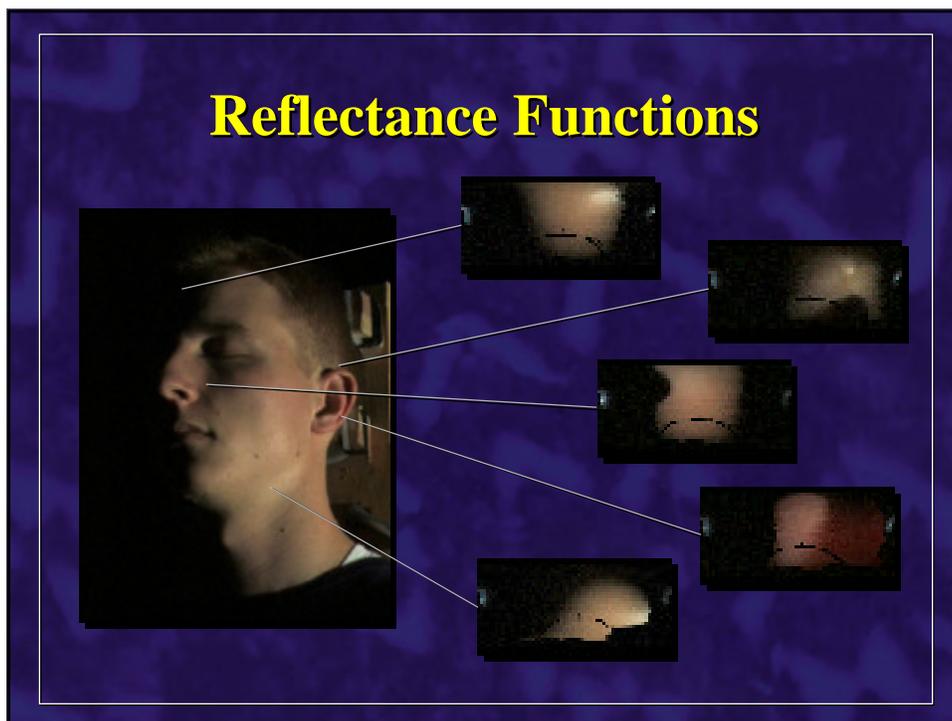


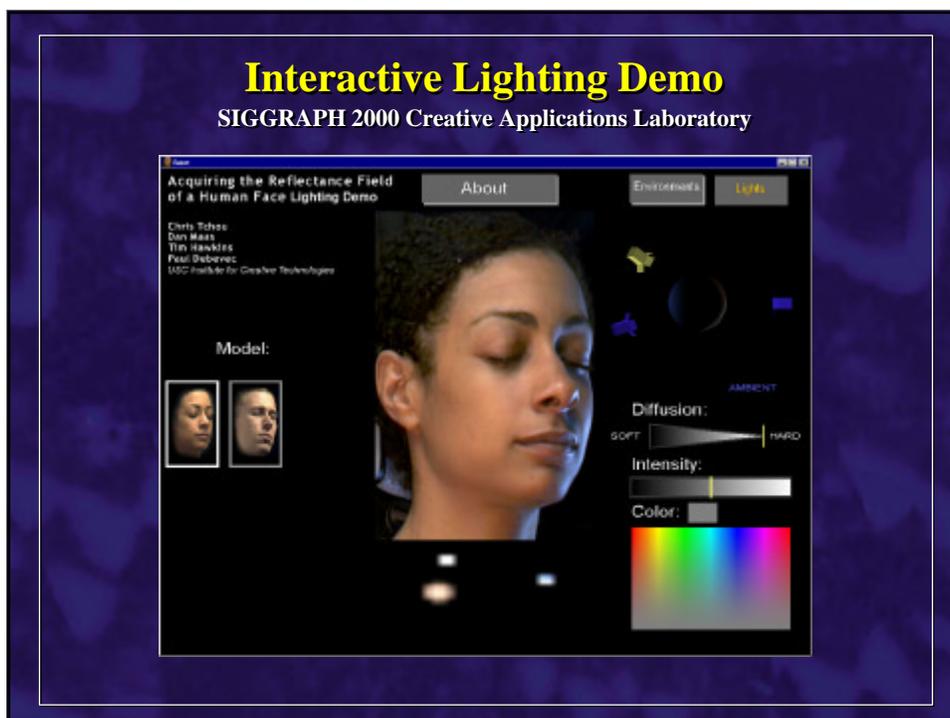
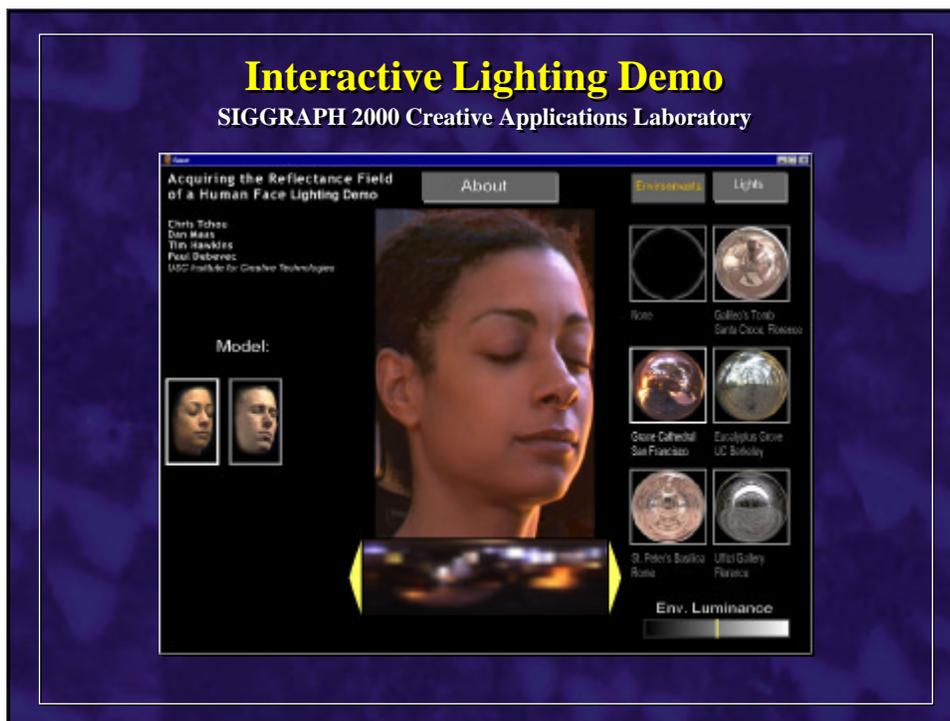
The Light Stage



“Acquiring the Reflectance Field of a Human Face”
Debevec, Hawkins, Tchou, Duiker, Sarokin, and Sagar

SIGGRAPH 2000





Light Stage Data Image (HDR)



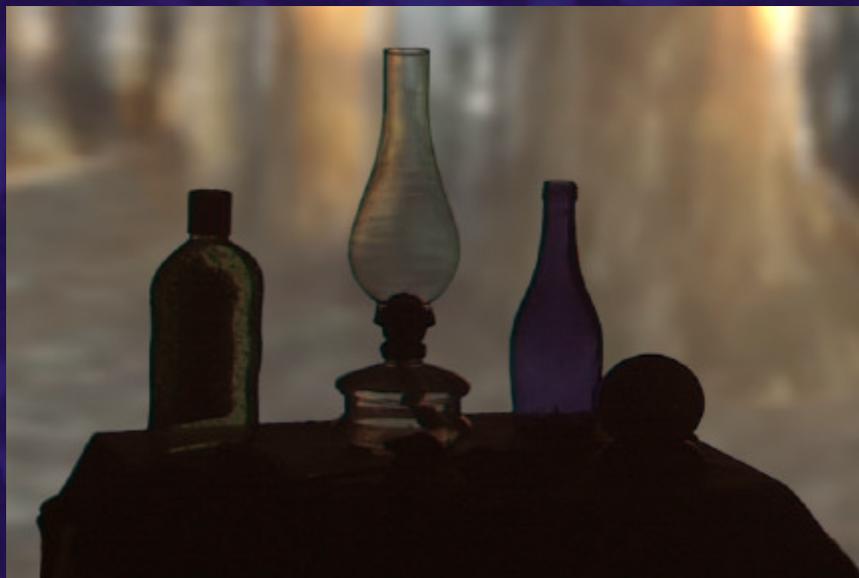
LS Rendering in RNL Environment



Environment Matte Acquisition



Environment Matte Rendering



LS Rendering in RNL Environment



LS Image w/o Background



Complete Rendering



Complete Rendering (Grace)



Basic Approach

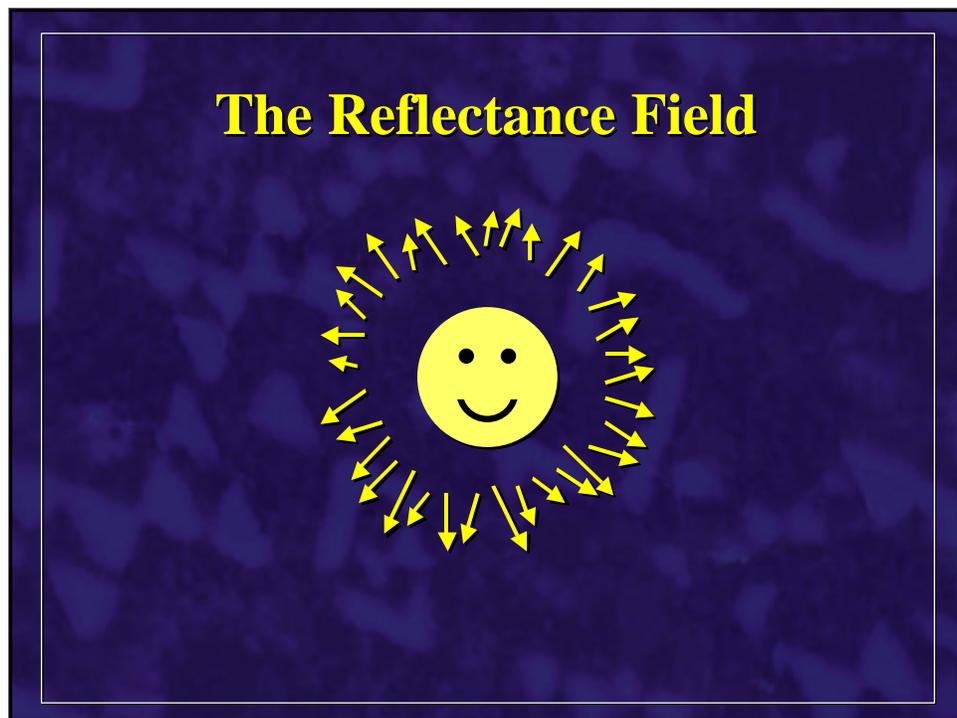
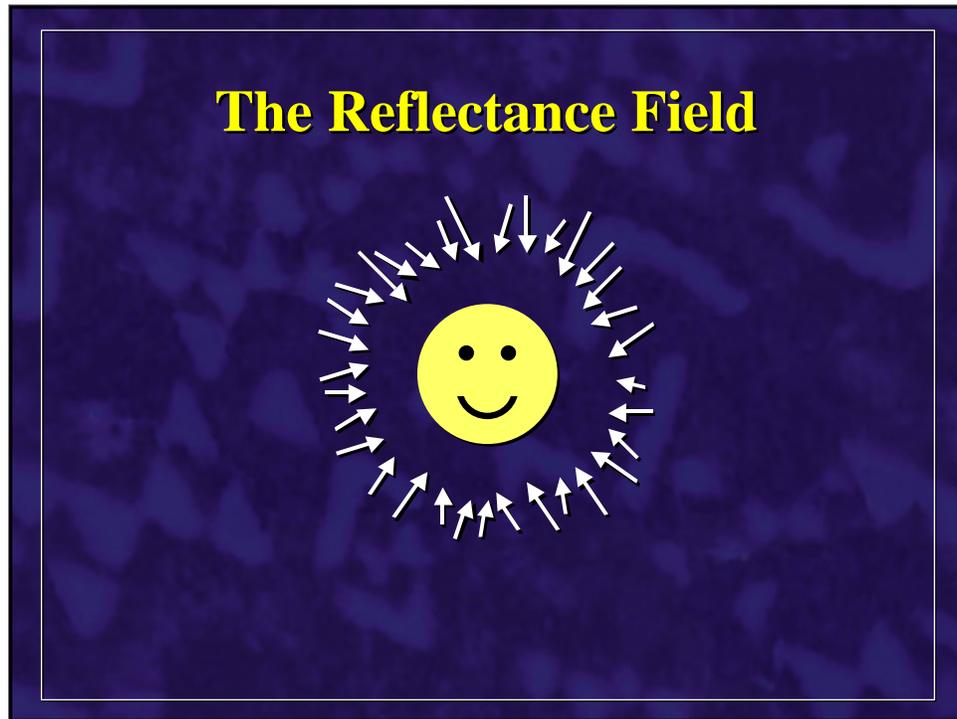
- Directly record how the face, as a whole, interacts with **light**, *i.e.*
- Record how the face transforms an **incident** field of illumination into a **radiant** field of illumination, *i.e.*
- Acquire the face's 8-D **Reflectance Field** as fully as possible



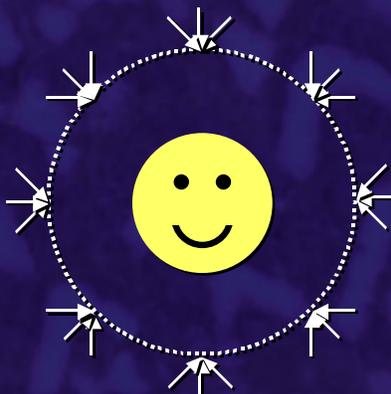
$$R = R(\mathbf{R}_i, \mathbf{R}_r) = R(u_i, v_i, q_i, \mathbf{f}_i; u_r, v_r, q_r, \mathbf{f}_r)$$

The Reflectance Field





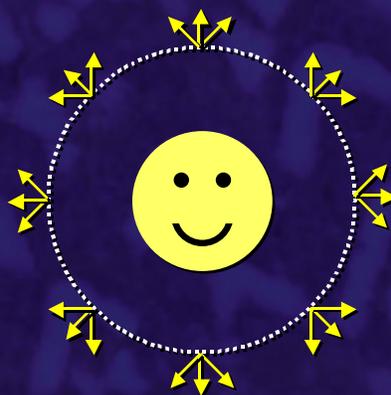
The Reflectance Field



$$R_i(u_i, v_i, q_i, f_i)$$

incident light field

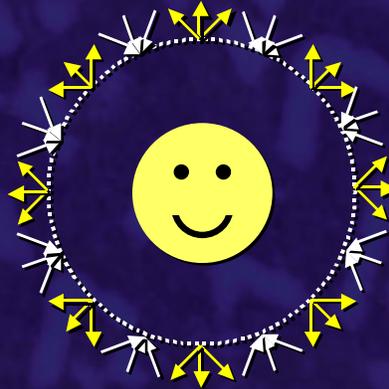
The Reflectance Field



$$R_i(u_i, v_i, q_i, f_i) \longrightarrow R_r(u_r, v_r, q_r, f_r)$$

incident light field radiant light field

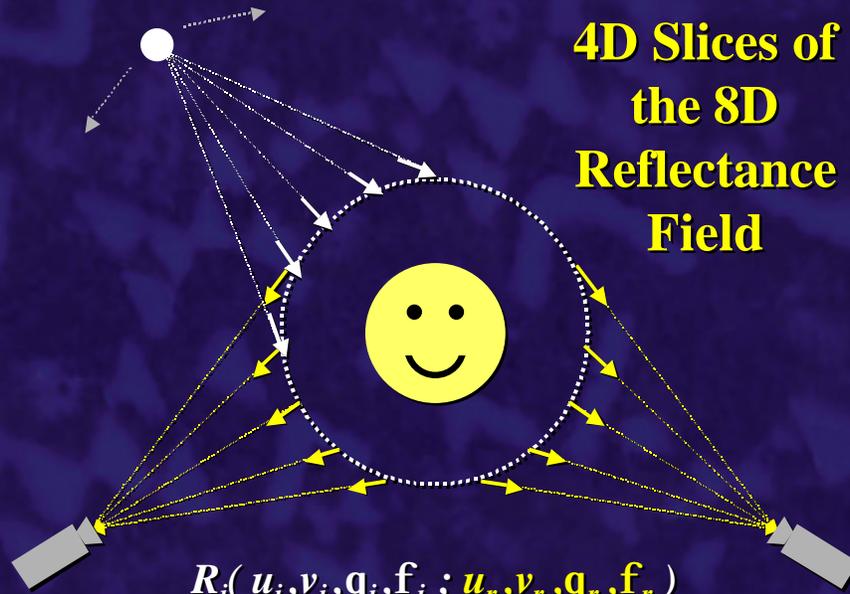
The Reflectance Field



$$R_i(u_i, v_i, q_i, f_i; u_r, v_r, q_r, f_r)$$

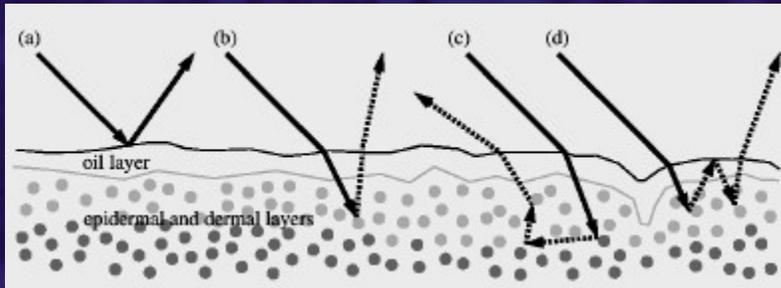
8D reflectance field

4D Slices of the 8D Reflectance Field



$$R_i(u_i, v_i, q_i, f_i; u_r, v_r, q_r, f_r)$$

Reflection of Light from Skin



Specular Component:

(After Hanrahan and Krueger SIGGRAPH 93)

Color of light, shiny, brighter near grazing,
maintains polarization

Subsurface Component:

Color of skin, diffuse, desaturated near grazing,
scrambles polarization

Separating Reflectance Components using Crossed Polarizers



Normal Image

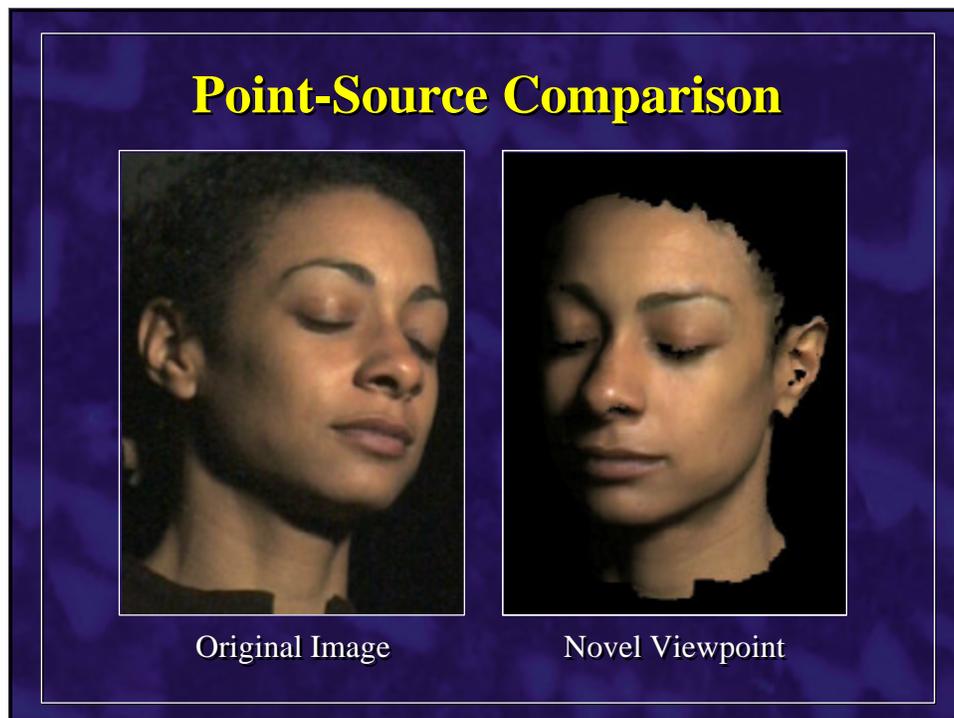
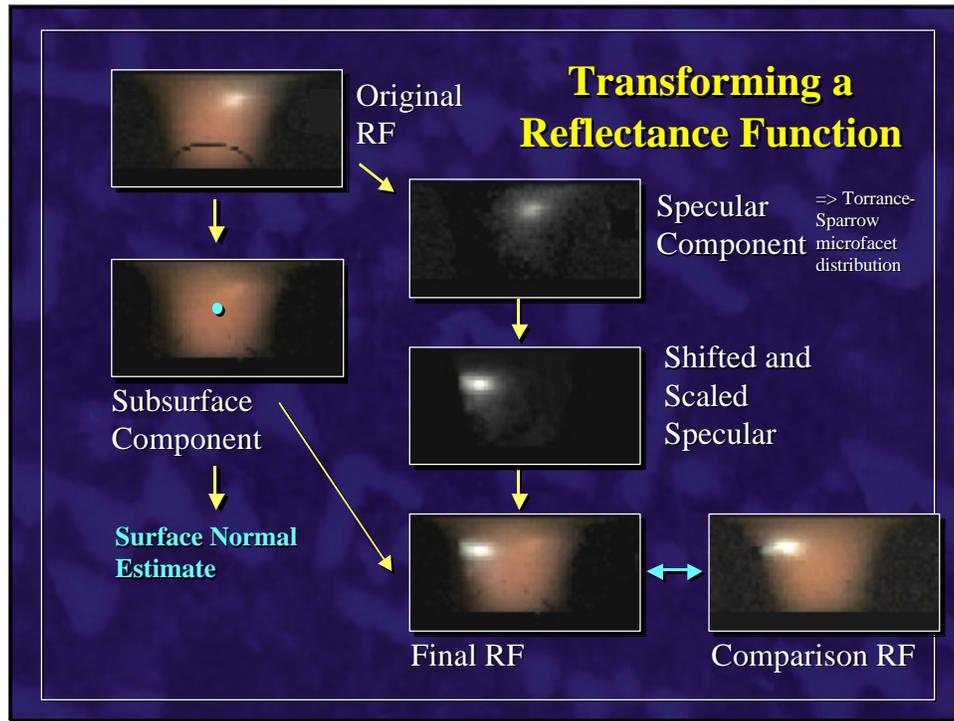


Subsurface
Component



Specular
Component

Colorspace techniques - Sato '94, Nayar '97

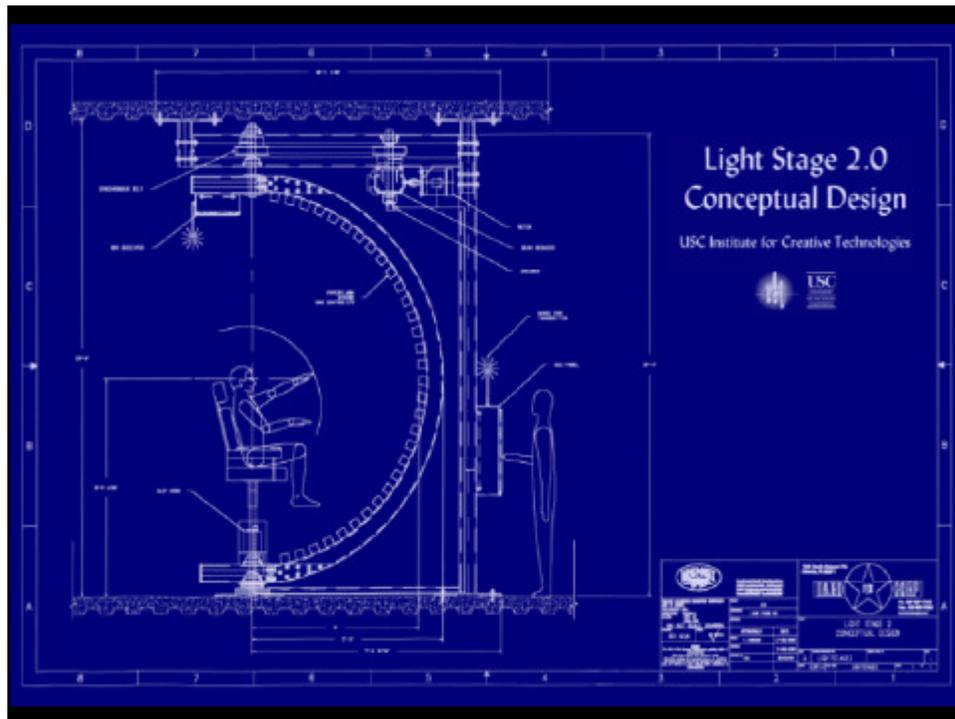


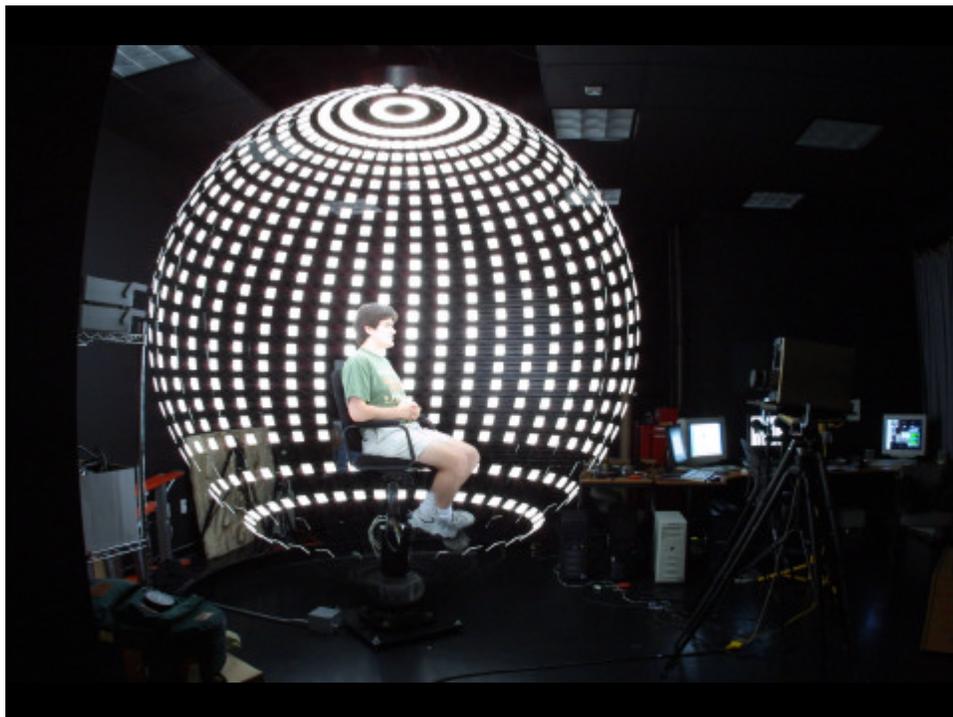
Spatially-Varying Reflectance Parameters

			
Surface Normals	Diffuse Albedo	Specular Intensity	Specular Roughness
n	r_d	k_s	a

Compositing Test

 Original Image	 Light Probe
 Rendered Face	 Composite





Light Stage 3.0

Computer controlled lighting all around the subject

Each light can assume a full range of colors and intensities

If the required lighting is known beforehand, live-action performance can be captured and composited into live-action footage or a virtual set with correct photometric interactions

Light Stage 4.0 (?)

Very fast strobe lighting all around the subject

If the strobes are very fast relative to the frame rate, and all the differently illuminated images are captured and stored, then performances can be arbitrarily re-illuminated in post-production

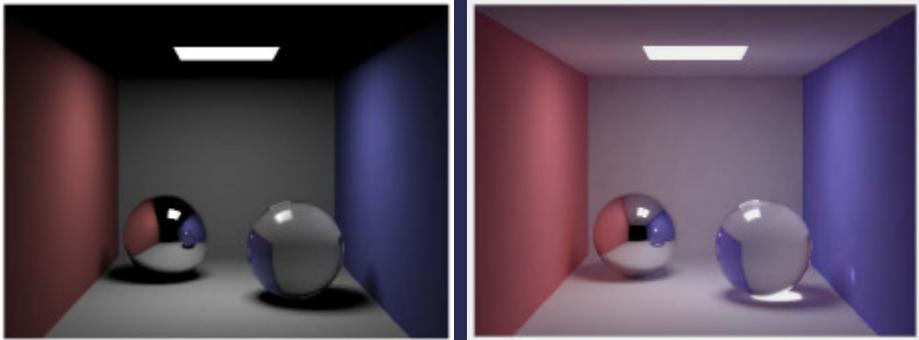
Global Illumination

A Brief Overview.

Dan Lemmon - Digital Domain - Siggraph 2001

The following is a “not so technical” overview of *Global Illumination*, a technique for simulating realistic lighting. It’s goals are to define Global Illumination and to give a brief introduction to some approaches for achieving Global Illumination.

Direct Illumination vs. Global Illumination

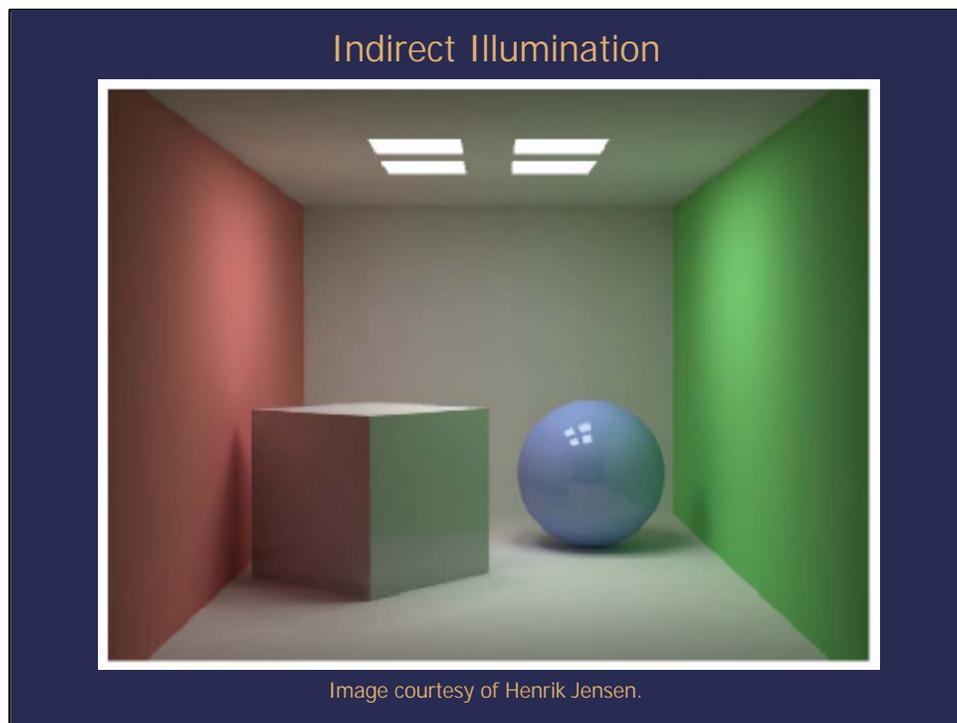


- discreet light source.
- efficient lighting calculations based on light and surface vectors (i.e. fast cheats).

- reflected, scattered and focused light (not discreet).
- physical-based light transport calculations modeled around *bidirectional reflective distribution functions* (BRDFs).

Images courtesy of Henrik Jensen.

In *raytracing* or *scanline-rendering* – the kind of rendering most of us do most often – we define discreet light sources at specific locations in a scene. We can take some shortcuts (i.e. cheat) if we only concern ourselves with light that comes straight from those discreet light locations to the surface that we are rendering. In other words, we ignore any indirect light that might be bouncing off of walls, reflecting off of shiny objects, or bouncing around under the surface of translucent surfaces. Because we are only paying attention to the light that comes directly from the light sources, we call this “*direct illumination*”. On the other hand “*global illumination*” tries to account for all of the light in an environment – indirect light bouncing off walls, reflecting off shiny objects, etc. Rather than using efficient algorithms based on discreet light sources, global illumination tries to perform more realistic simulations of light transport.



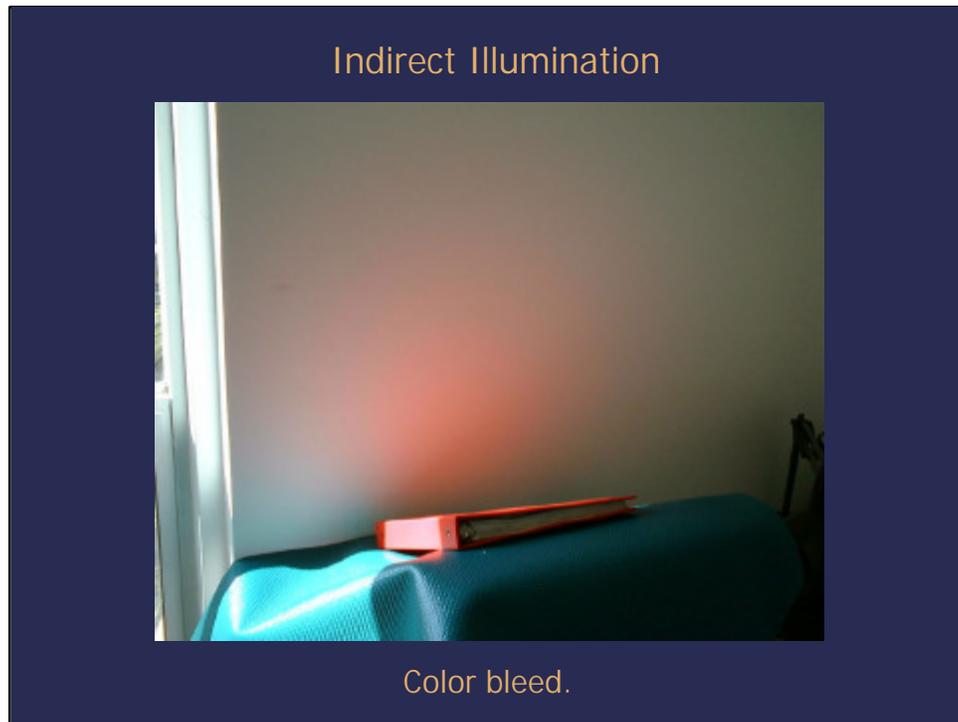
By “Indirect Illumination” we mean all the light that comes to a surface indirectly; not straight from the light source. This might be light scattering off of walls, light refracted and focused through glass or water, or even light bouncing around under the surface of a translucent object.

Indirect Illumination

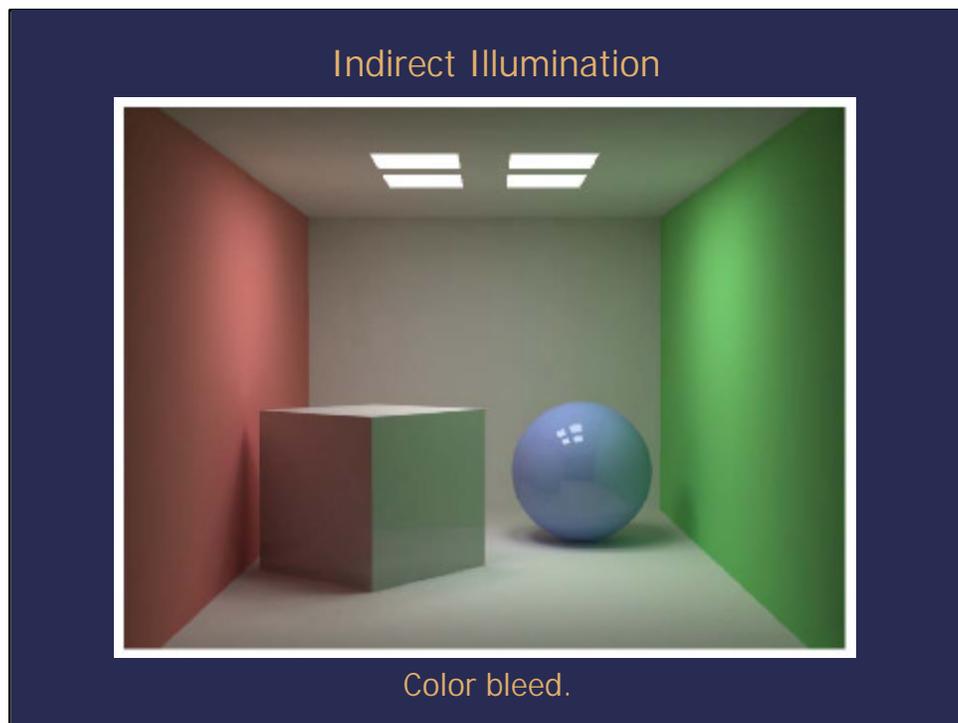


San Francisco Museum of Modern Art
Photograph courtesy of Philp Greenspun.

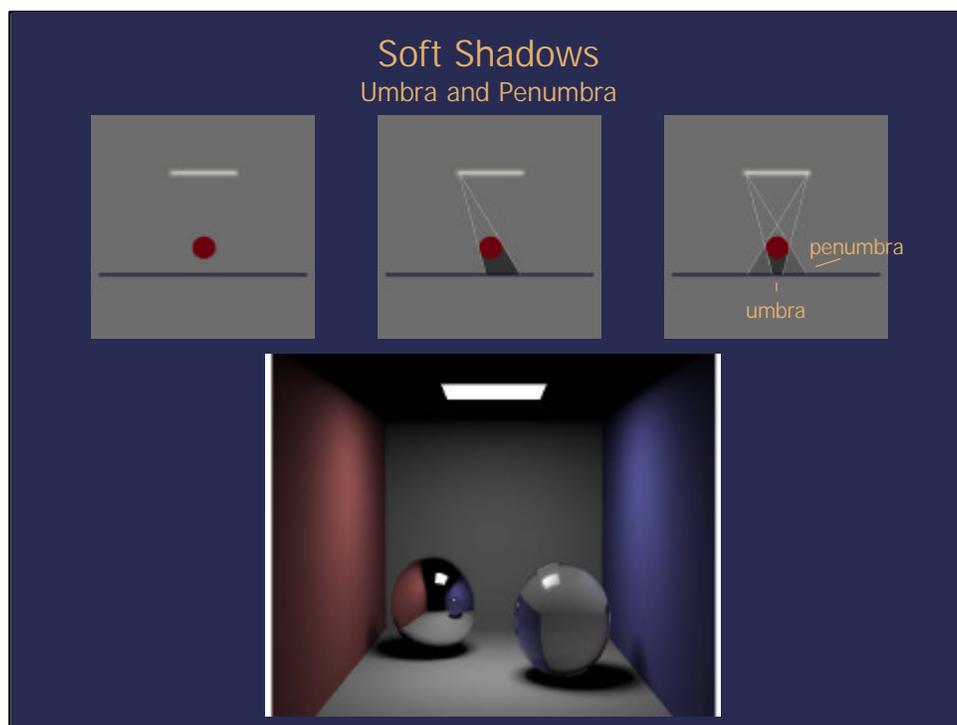
By “Indirect Illumination” we mean all the light that comes to a surface indirectly; not straight from the light source.



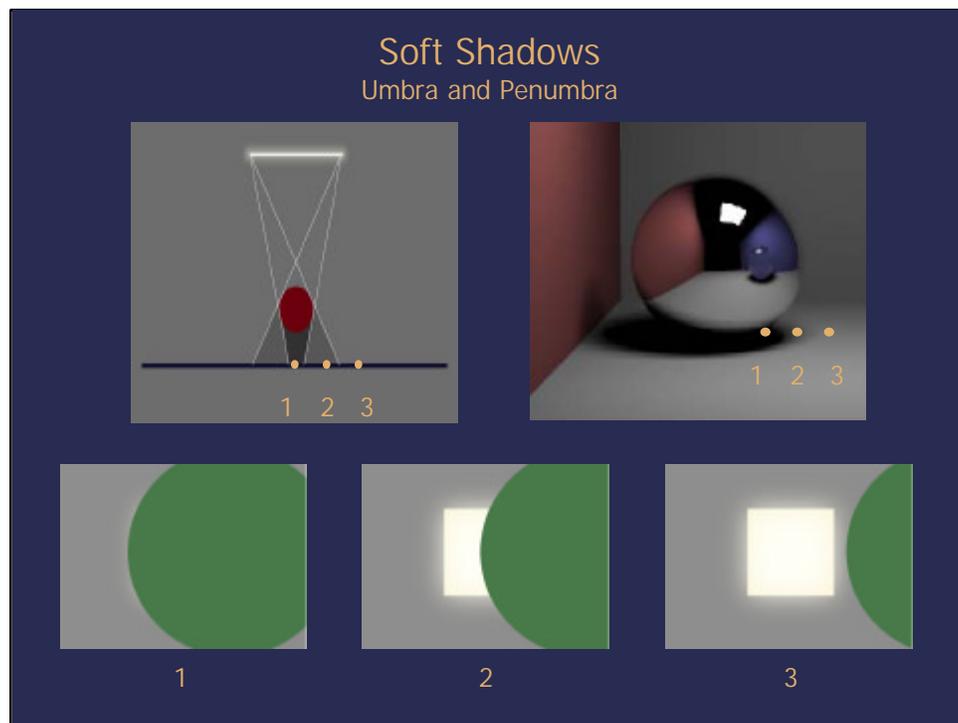
One of the most striking effects of indirect illumination is *color bleed*.



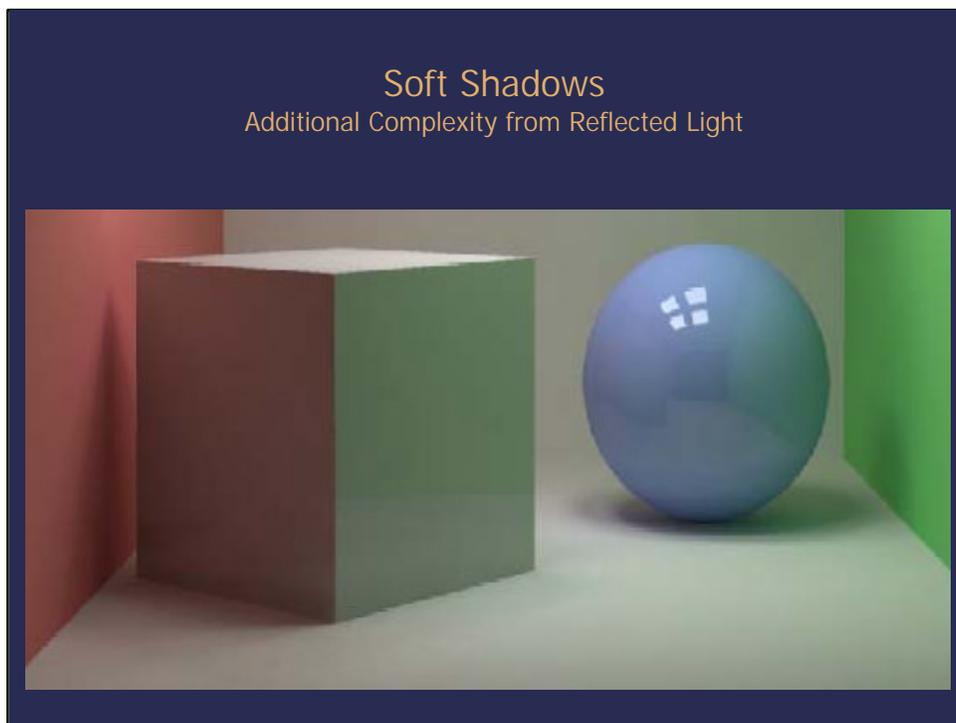
If you look at the ceiling in this image, you'll notice that the left side is slightly red and the right side is slightly green. This is because much of the light that reaches the ceiling bounces off of the colored walls. Sometimes people will use the term "indirect illumination" to refer specifically to the light that is scattered by diffuse surfaces (i.e. not the focused or crisply reflected light). Others call the same thing "radiosity" or "radiosity effects", although that can cause some confusion as *radiosity* can also refer to certain specific historical algorithms that can be used to calculate diffuse reflected light.



Soft shadows are not necessarily unattainable using direct illumination techniques. Area lights, while still discreet, at least provide for umbra and penumbra effects. The umbra is the area in the shadow which is darkest. In the umbra, none of the lightsource is visible. In the penumbra, however, the lightsource is *partially* obscured by the object casting the shadow, but still *partially* visible. Consequently, a certain percentage of light still makes it to the surface. Near the outside edge of the penumbra, the lightsource is nearly entirely visible. And near the inside edge of the penumbra, the lightsource is nearly entirely obscured. A shadow with a smooth, soft rolloff from light to shadow is the result.



In the umbra, none of the lightsource is visible (1). In the penumbra, however, the lightsource is *partially* obscured by the object casting the shadow, but still *partially* visible (2). Consequently, a certain percentage of light still makes it to the surface. Outside the penumbra the lightsource is entirely visible (3). A shadow with a smooth rolloff from light to shadow is the result.



However, the soft shadows that result from Global Illumination techniques are much more dynamic than their Direct Illumination counterparts. In addition to *color bleed* and rich fill lighting, the soft shadows of Global Illumination contain so-called *contact shadows*, not just a dark *umbra*. Notice the surface just under the sphere. The shadow gets much darker where the direct illumination as well as most of the indirect illumination is occluded. That dark *contact shadow* helps enormously in “sitting” the sphere in scene. *Contact shadows* are difficult to fake, even with area lights.

Focused and Reflected Light



Focused and reflected light, or *caustics*, are another feature of the real world that we lack in direct illumination.



Caustics are a striking and unique feature of Global Illumination.

Focused and Reflected Light

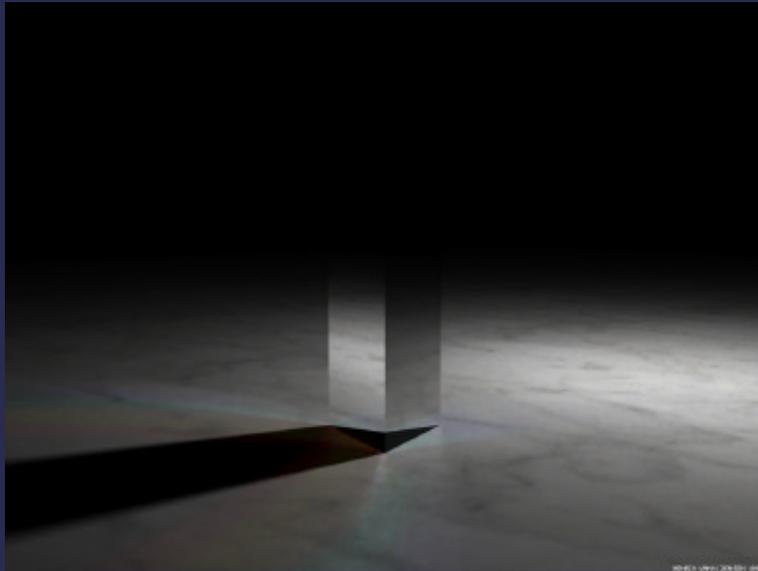


Image courtesy of Henrik Jensen.

This rendering of a prism illustrates how with Global Illumination you can actually treat white light as a sum of light of various wavelengths and refract those different wavelengths at different rates producing a “rainbow” spectral separation.

Subsurface Light Scattering



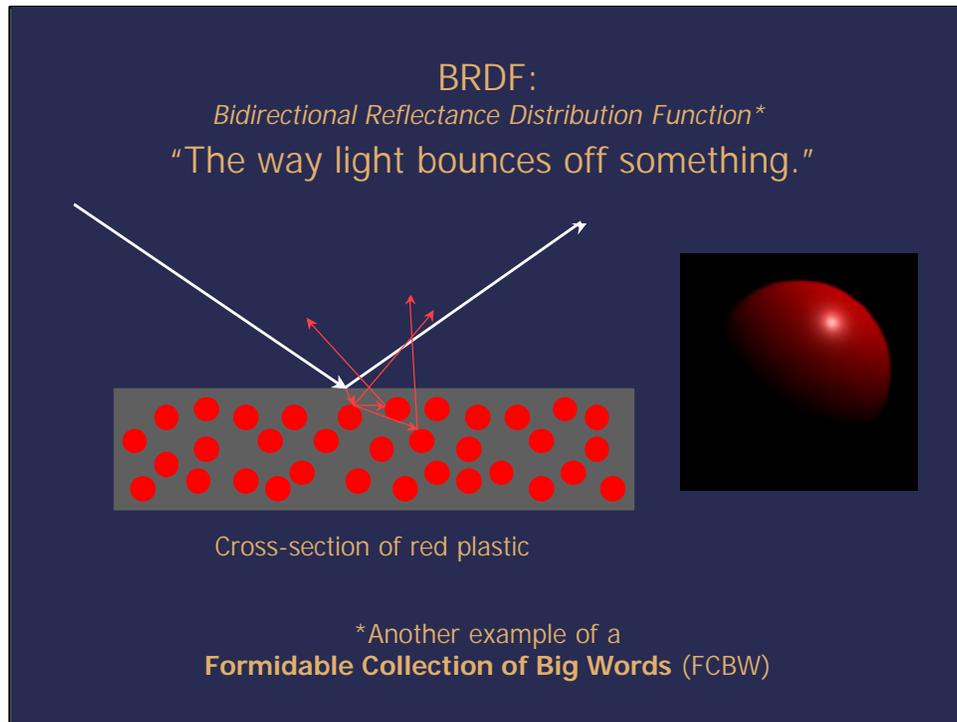
Another area in which direct illumination falls short of the real world is in subsurface scattering of light.

Subsurface Light Scattering

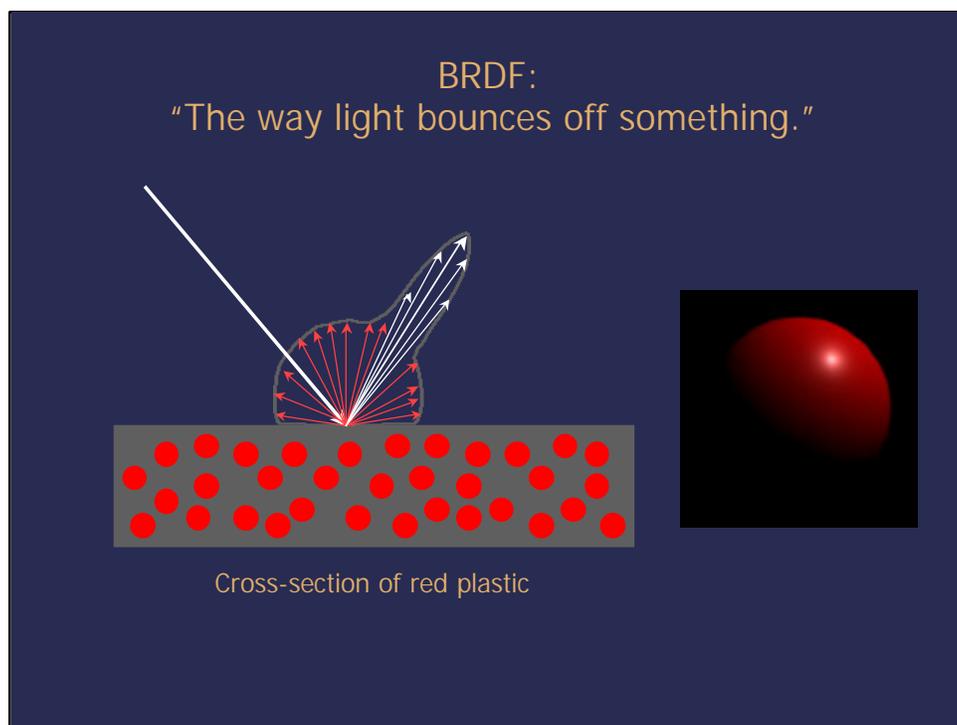


Image courtesy of Henrik Jensen.

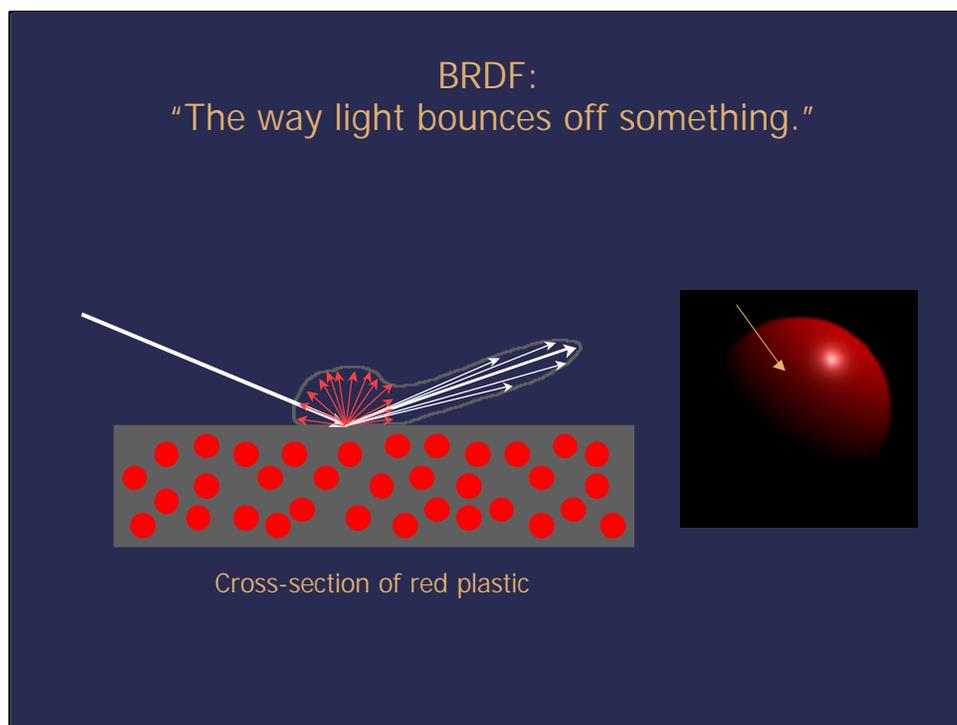
The effects of global illumination subsurface light scattering are striking. Notice the nose and hair bun of this CG marble bust. It exhibits translucent properties as some of the light that contacts the surface from behind is scattered through the surface and back to the camera. Pretty amazing!



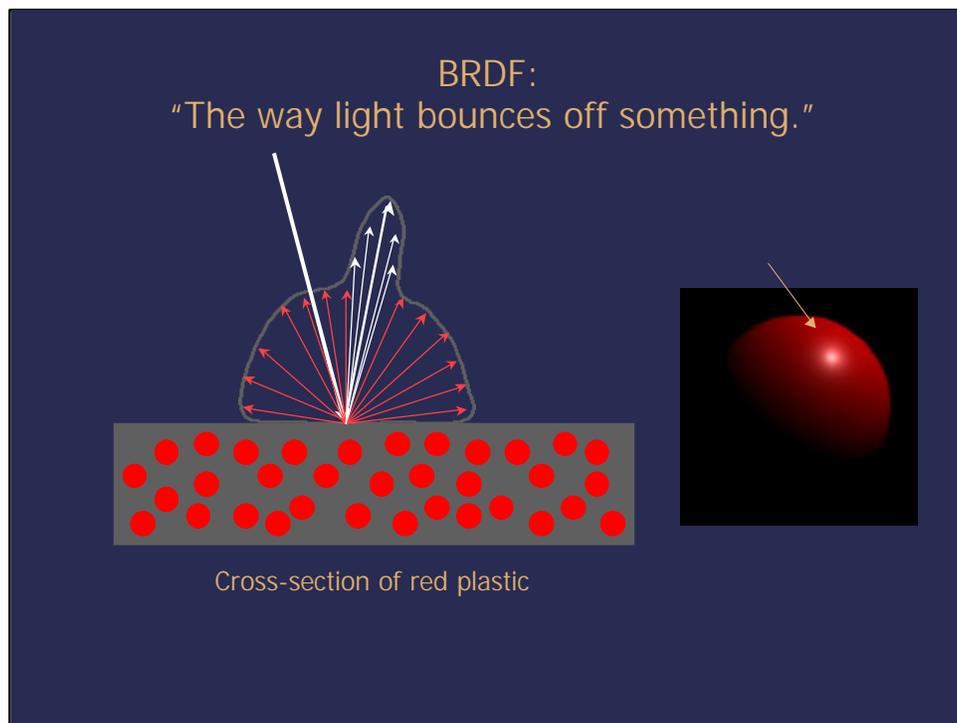
In order to continue talking about different rendering techniques, we need to conquer the term *BRDF* or "*Bidirectional Reflectance Distribution Function*." *BRDF* really just means "the way light bounces off of something." If we were talking about the *BRDF* of red plastic, for example, we'd want to note that some of the light bounces right off of the surface along the angle of reflection (equal and opposite to the angle of incidence) without really changing color much. (We sometimes call that light the *specular reflection* or *specular highlight*). We'd also want to note that some of the light gets refracted into the plastic and bounced around between red particles of pigment. Most of the green and the blue light is absorbed by the particles and only the red light makes it's way back out of the surface. Additionally, the red light that gets bounced back out of the plastic is scattered every which way with fairly equal probability. We call that scattered light *diffuse*.



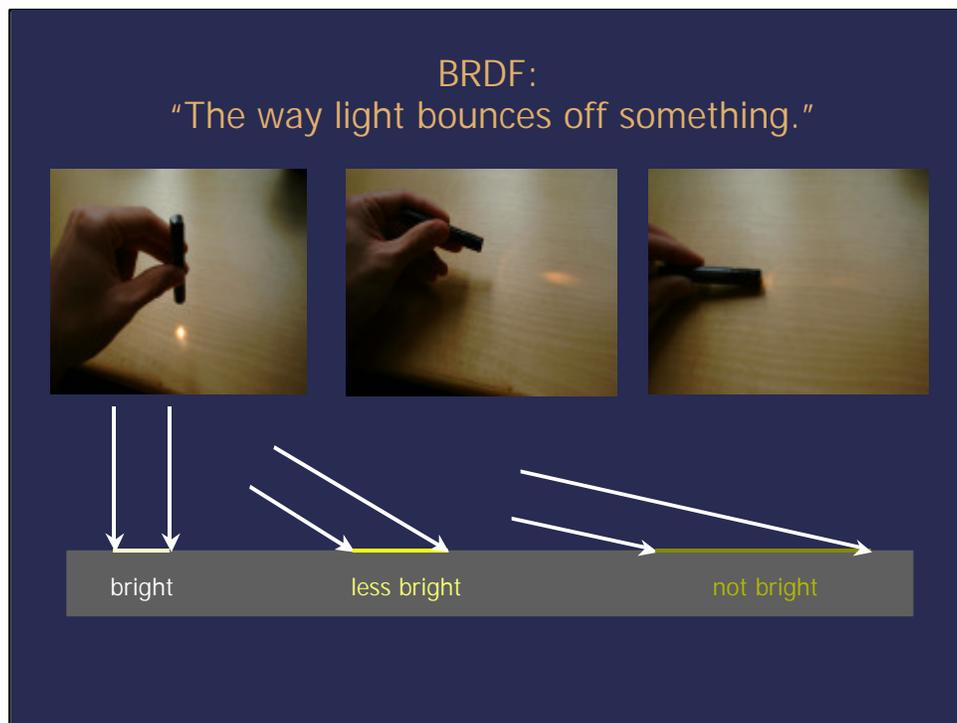
One convenient way to look at all that bouncing/reflecting information is to take all the light that comes through a single point (the point that we will shade) and look at the cross-section of bounced and reflected light at that point. We draw the light that is brightest as long lines, and we draw the dimmer light as shorter lines. So for red plastic, we'd have some long white lines along the angle of reflection as well as some shorter red lines radiating out in a sort of hemi-sphere. [Sometimes we just draw the outline.]



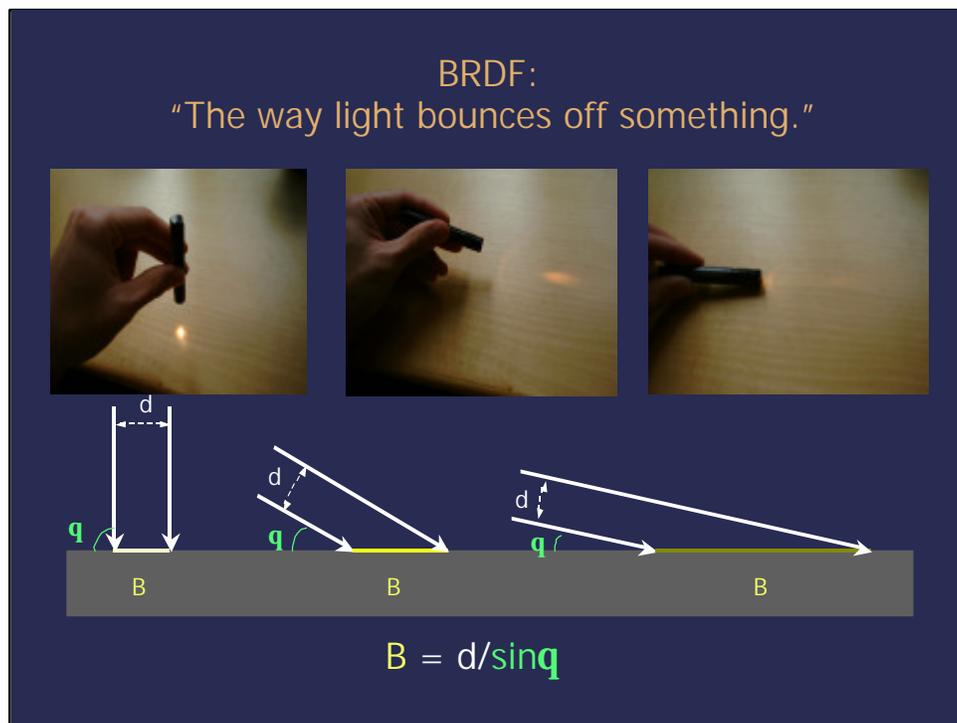
One thing that is important to note is that as the angle of incidence decreases, the cross-section of the light at that point changes. In the case of red plastic, the light which is scattered and colored red, the *diffuse* component, decreases.



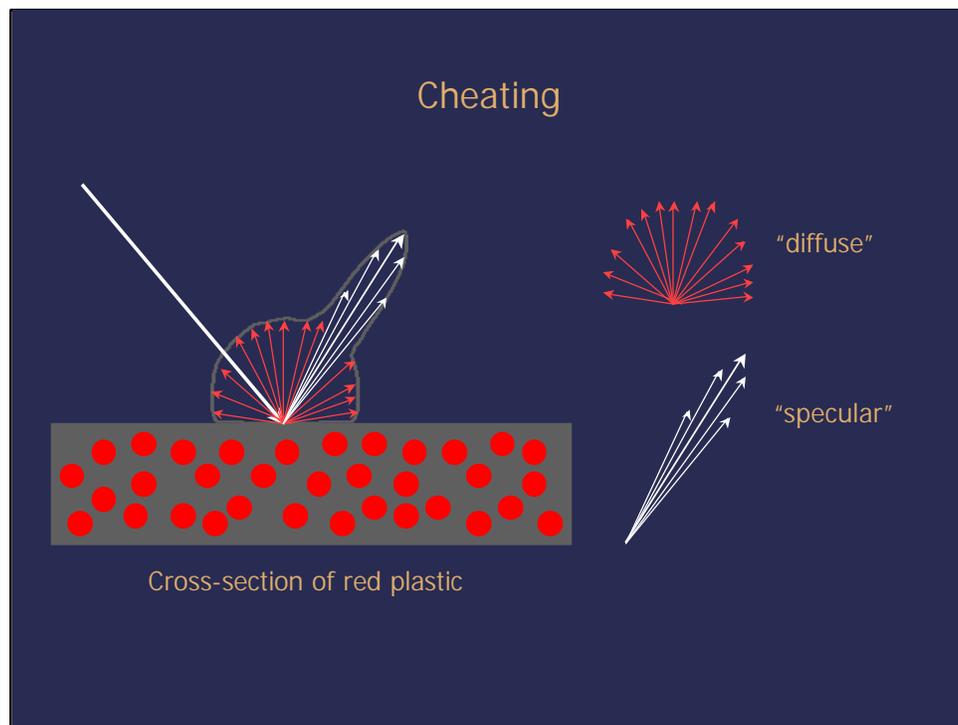
Similarly, as the angle of incidence increases, the cross-section of the light at that point also changes. In the case of red plastic, the light which is scattered and colored red, the *diffuse* component, increases.



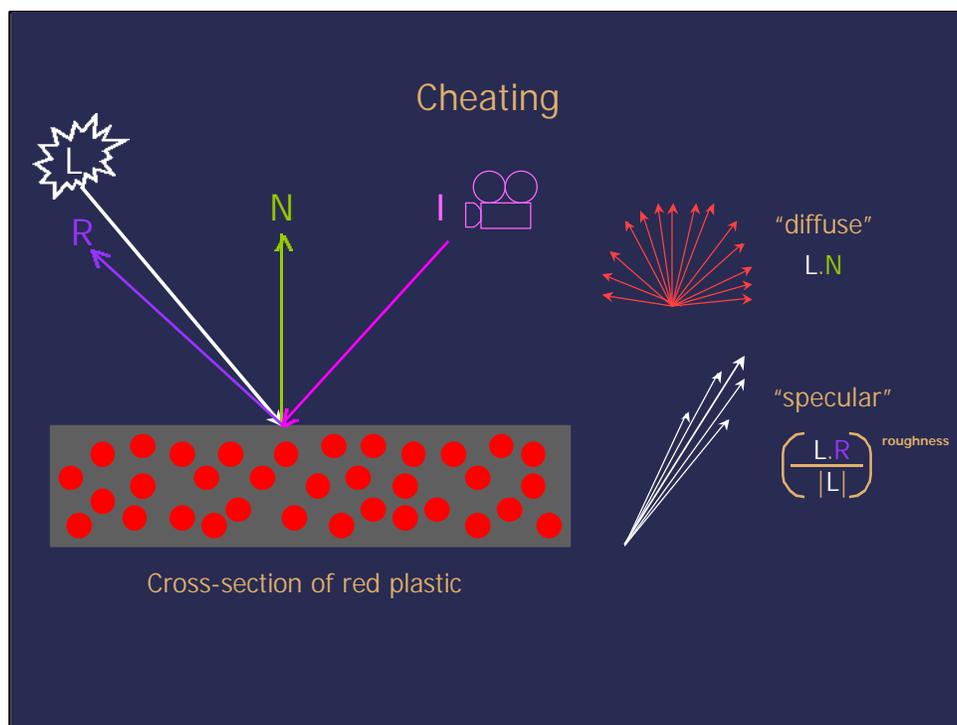
With a flashlight, we can visualize the reduction of the diffuse component relative to the angle of incidence.



We can figure out the length of B – and thus the reduction of radiance – with some simple trigonometry.



If we want to render our images quickly (usually a good idea) we can separate the specular and the diffuse components, use optimized algorithms to figure them out, and then sum the results when we are done.

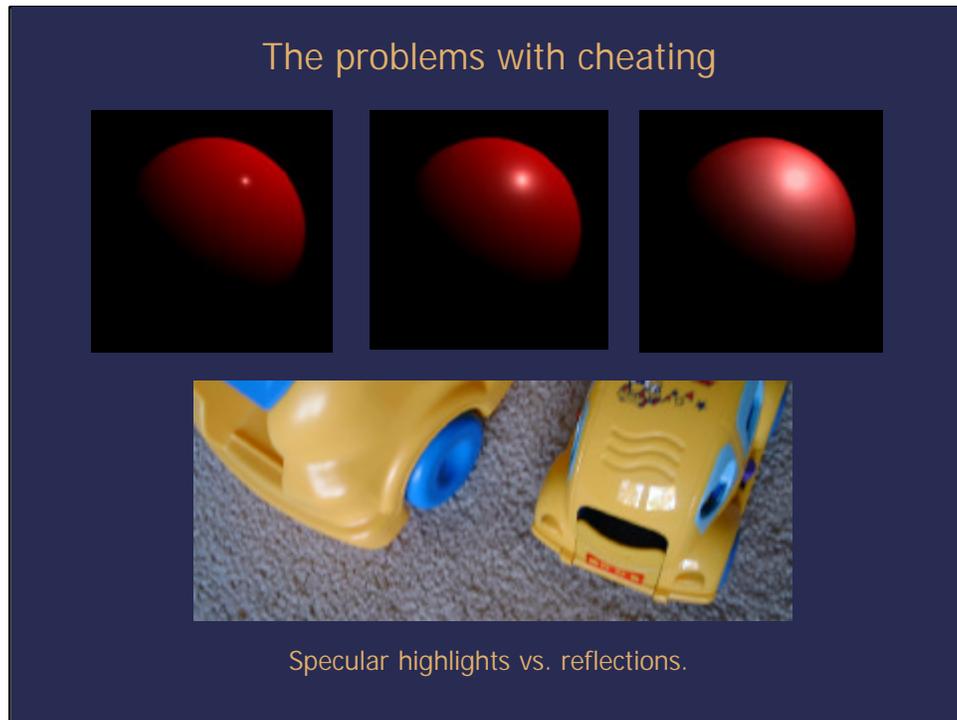


Two optimized algorithms for that we often use are Lambert's and Phong's algorithms for diffuse and specular components respectively. Used together they give fairly realistic results for the direct illumination of plastic and plastic-like (*die-electric*) materials. Amazingly, much of our work in visual effects to date has been rendered this way, and it has held up pretty well.



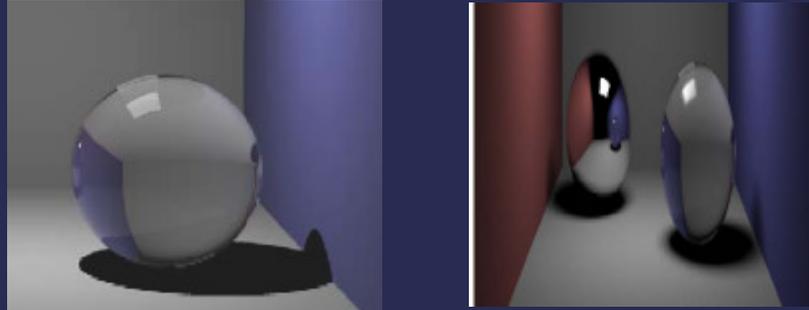
"Phong's a dude?"
- Howie Muzika

The use of these algorithms has become so pervasive that some people in the visual effects community have only recently learned that terms like “phong” and “blinn” actually reference the people who invented them.



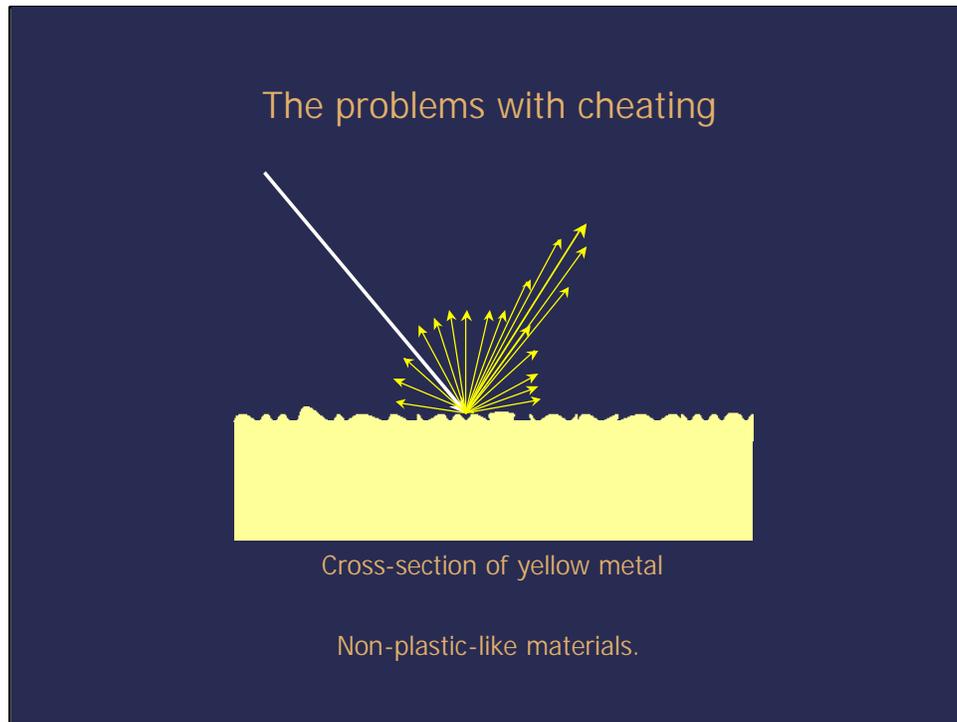
Unfortunately, this simplified approach breaks down in some places. For starters, there really is no difference between a “specular highlight” and a “reflection”. The “specular highlight” is really just the reflection of the lightsource. In Phong’s algorithm as a surface decreases in “roughness” the “specular highlight” becomes smaller and smaller. While this is true for an infinitely small point light, most of the lights that we deal with in the real world have width, depth, and height. What should really happen is the entire “reflection” should get blurrier as a material increases in “roughness”. When a material is perfectly smooth - that is, it has a roughness of zero - the “specular highlight” should be the same size and shape (relatively) as the lightsource itself. And, really, in a bright-ish environment, you should be able to see the rest of the environment crisply reflected as well.

The problems with cheating



Shadows.

Of course, we already talked about shadows. We can cheat them a little further by using area lights, but area lights only get us a little bit closer to reality.



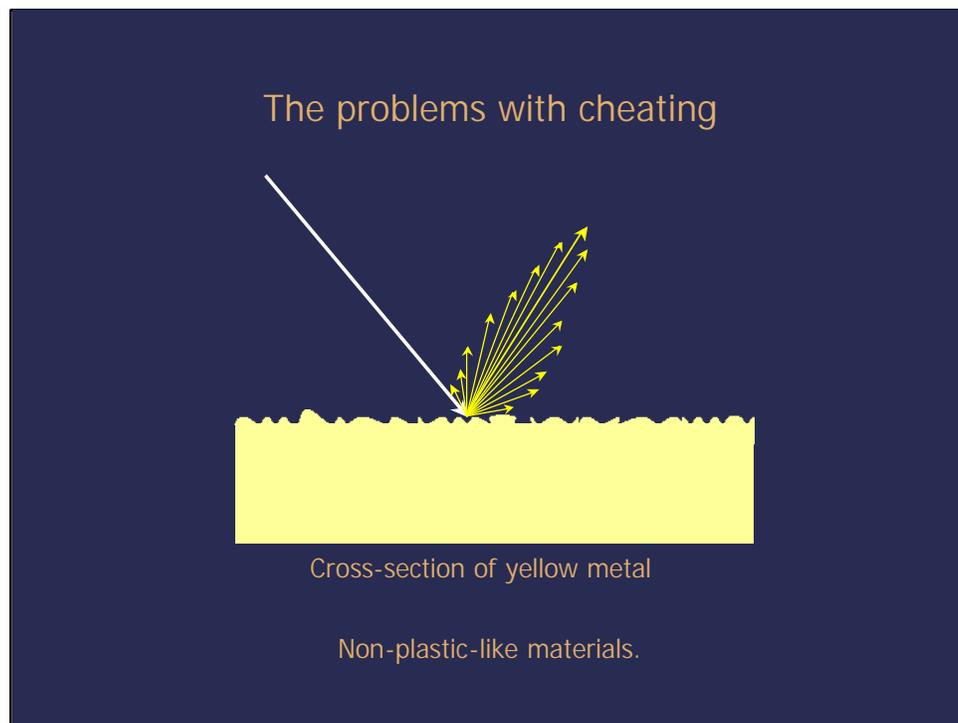
While our specular/diffuse separation strategy yields a reasonably good approximation of plastic-like materials, it doesn't really work for non-plastic-like materials. For example, we can try to treat brass, a yellow metal, in a plastic-like way. We might make it so that both the "specular" and the "diffuse" absorb the blue component of the light, casting the light yellow. We might turn the diffuse component way down and even add a reflection map.

The problems with cheating



Non-plastic-like materials.

The only problem is that metal doesn't really behave that way at all.



Really, metal doesn't have a "diffuse" component, nor anything that roughly approximates a "diffuse" component. The light doesn't penetrate the surface of the metal and there aren't little pieces of pigment suspended in a transparent substrate. Instead, light is scattered by scratches, pits and other surface irregularities we'll call micro-facets.

The problems with cheating



Non-plastic-like materials.

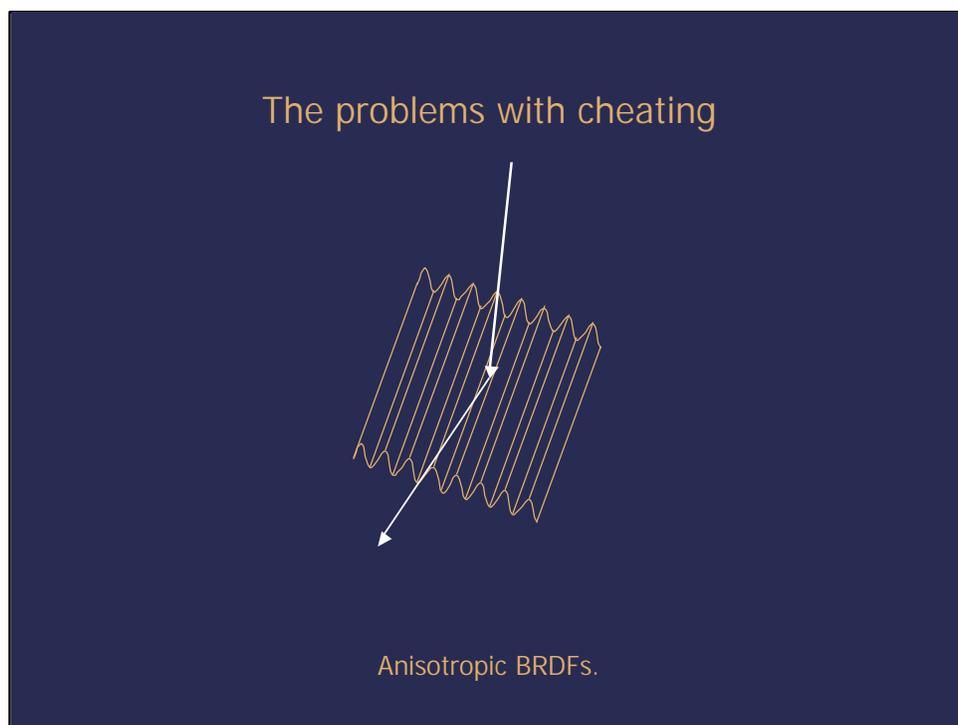
Depending on the frequency and amplitude of these micro-facets (how big they are, and how close together) the light may be scattered all over the place (e.g. machined aluminum) or may hardly be scattered at all and remain focused along the angle of reflection (e.g. polished chrome). Most metal falls somewhere in between, reflecting a slightly blurry image of the environment around it (e.g. the outside surface of a ring). And, of course, this isn't even considering factors like tarnish, rust, grime, fingerprints, dust and all the other factors that often accompany metal.

The problems with cheating

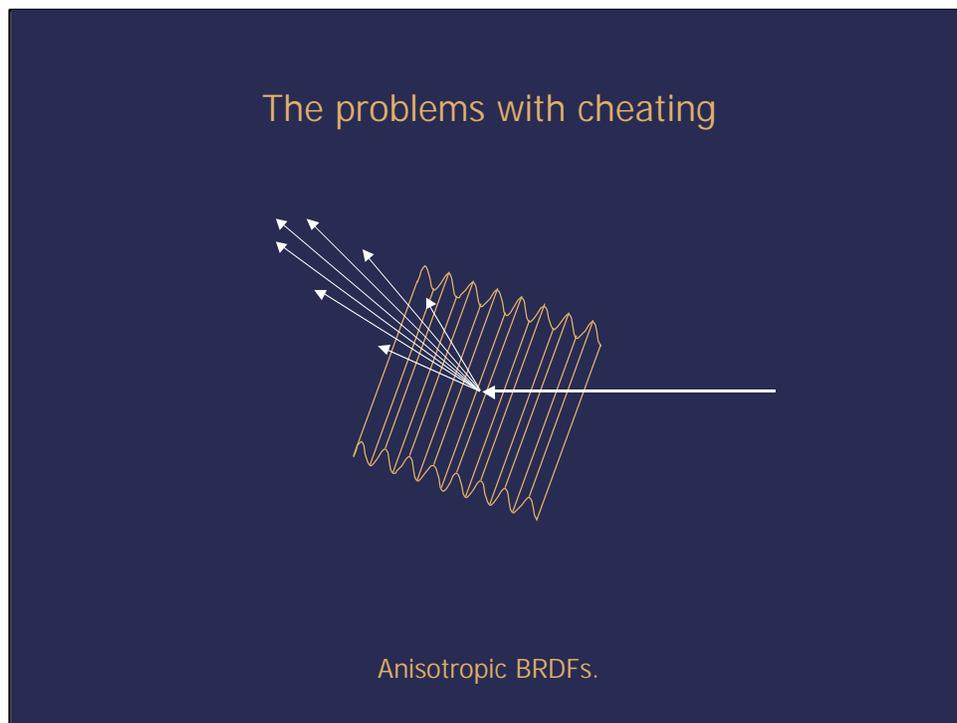


Anisotropic BRDFs.

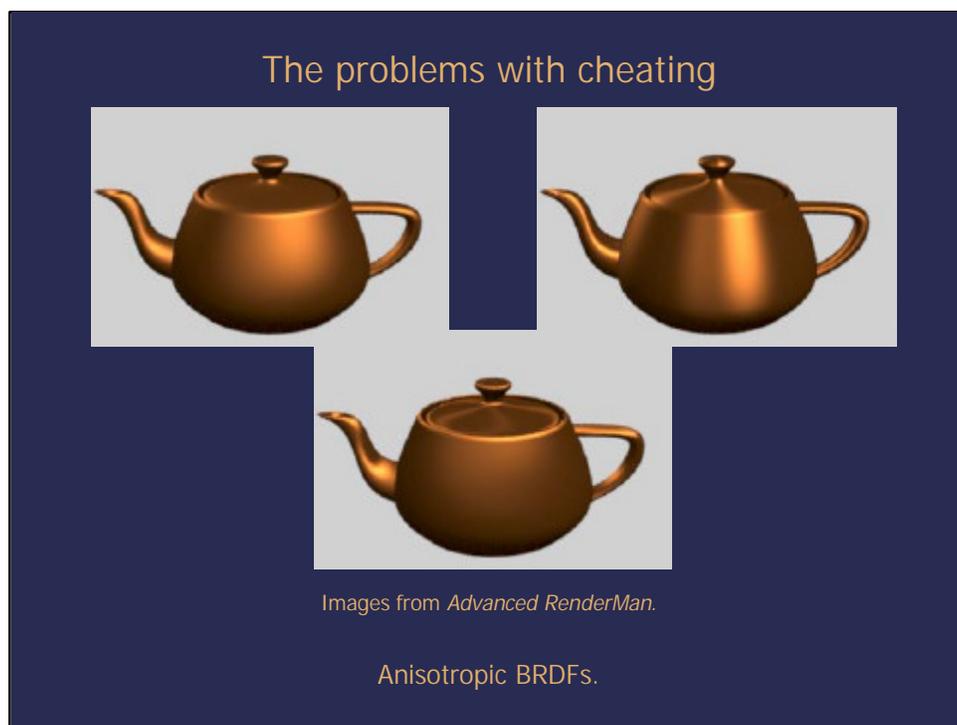
Also, because micro facets are often directional, light may be scattered more in one direction than another. Consider the case of brushed steel.



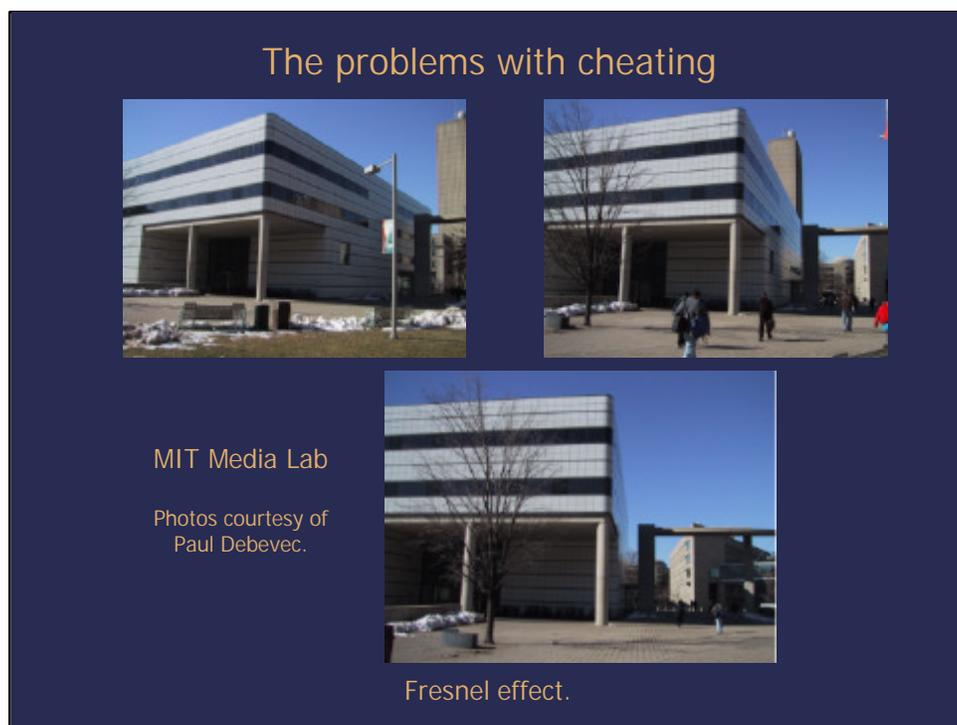
Consider the case of brushed steel. Imagine that the steel has a bunch of sub-pixel scratches that run roughly north to south. This makes it so that in a north-south direction the micro-facets are low-frequency, but in an east-west direction the micro-facets are high-frequency. If light hits these scratches from a northern direction, the light would stay reasonably focused as it bounced off of the surface and headed south.



However, if light hits these scratches from the east it would encounter a much higher frequency of micro facets, and so the light would be scattered as it headed west.



We can write shaders to mimic the effects of anisotropic materials; Greg Ward's algorithm is a popular one. This example from the "Advanced RenderMan" book shows a teapot made of an orange metal with different anisotropic properties. The teapot on the upper left has a rough, isotropic specular. Its highlight is distributed evenly in all directions. The teapot on the right has strong anisotropic properties in one direction. Notice how the highlight on the front of the teapot is much larger in the up-down direction than in the left-right direction. The teapot on the bottom has simulated micro-facets with two directions, as if the metal had one set of scratches running perpendicular to another set of scratches.



As the angle at which a light strikes a plastic-like material becomes shallower, more light is reflected. So, as you look at a surface at an increasingly shallow angle, the surface will appear to become more and more reflective. At a very sharp angle surfaces like a book, a wood table top, concrete and plastic almost become mirrors.

The problems with cheating



Fresnel effect.

At a very sharp angle surfaces like a book, a wood table top, concrete and plastic can almost become mirrors.

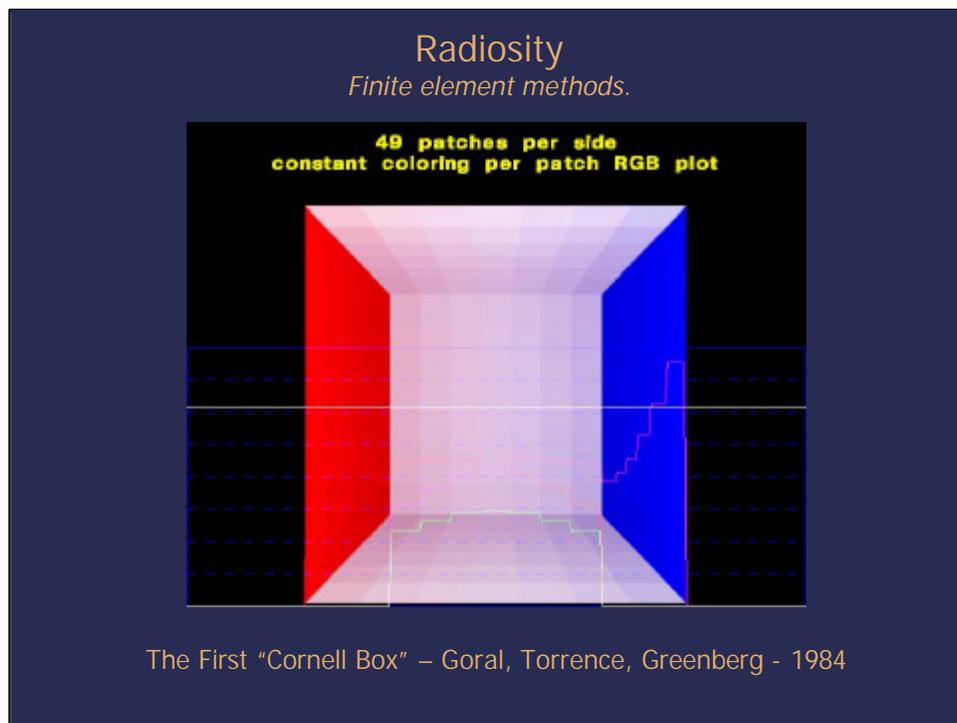
The problems with cheating



Image courtesy of Marcos.

Fresnel effect.

If we'd like, we can include algorithms to account for the fresnel effect. Notice the sky color dominating the pot color on the edges of the pots.



So rather than dealing with all the cheats, some researchers have tried to more realistically simulate the transport of light through a scene. One of the first successful approaches was *radiosity*, or “finite element methods.”

Radiosity

Finite element methods.

- Subdivide the geometry and calculate diffuse illumination for each patch through a variety of novel algorithms.
- Interpolate values across patches.

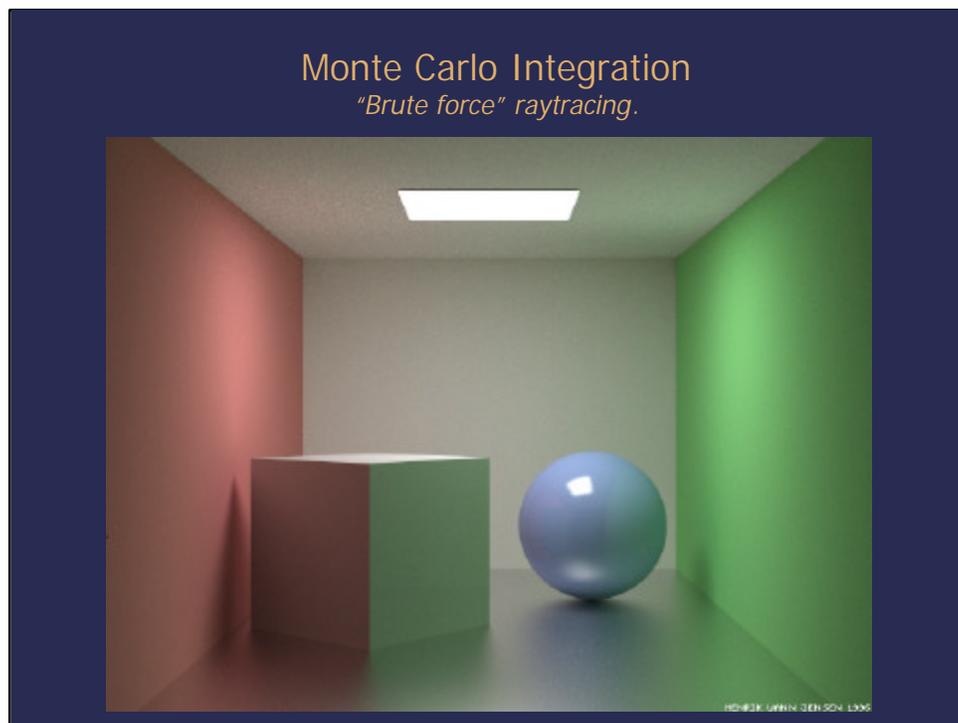
This technique involves subdividing geometry into several patches and then, for each patch, analyzing how much light reaches that point. In later research, the luminance values were interpolated across surfaces.

Radiosity

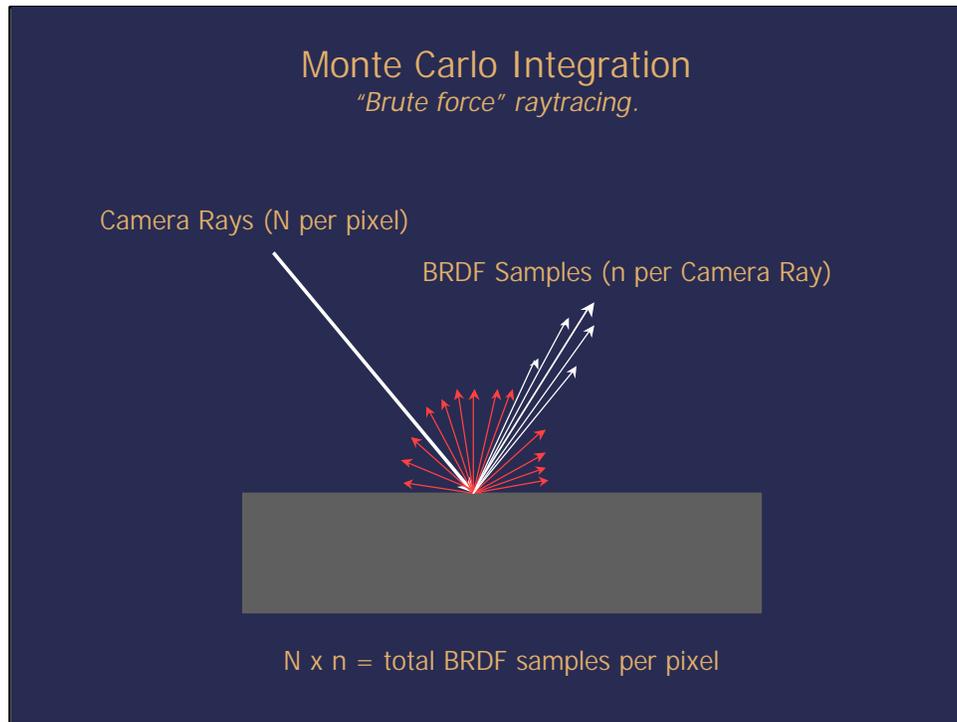
Finite element methods.

- Constrained by the resolution of your subdivisions.
- Had to calculate *all* of the geometry before you rendered an image.
- Shadows ("participating media") were problematic. (Note: no spheres or cubes in the first Cornell box.)
- No reflections or specular component.

However, the radiosity approach had a number of drawbacks.



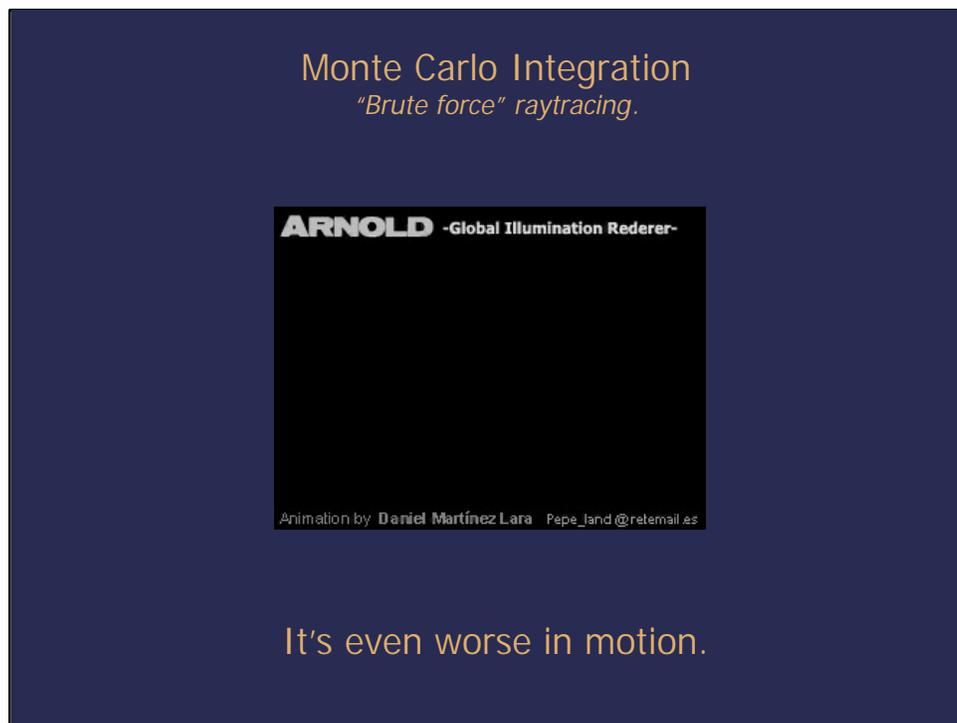
Another approach to solving the Global Illumination problem is through a technique that has become known as *Monte Carlo Integration*, or “brute force” raytracing. The Arnold renderer that has received so much recent attention falls into this category.



The approach involves sending out a whole lot of rays each time a camera ray strikes a surface. The rays are distributed according to the BRDF of the surface. (You can use BRDFs in the same way whether you are bouncing light rays or camera rays off of a surface. Hence the term “bidirectional”). The rays gather lighting information from the scene and are then summed to return a luminance value for that particular camera ray.



Unfortunately, this approach is prone to noise. The image above (a detail of the image a few slides back) shows some of that noise. And that was rendered with 16 subsamples per pixel. Each subsample spawned 1000 rays to sample the environment. So each pixel cast 16,000 rays from the surface; hence the term “brute force.” And there is still a lot of noise!



The noise gets worse when you animate! (Not to take away from this lovely example of the rich lighting effects that can be achieved with Global Illumination).

Monte Carlo Integration

"Brute force" raytracing.

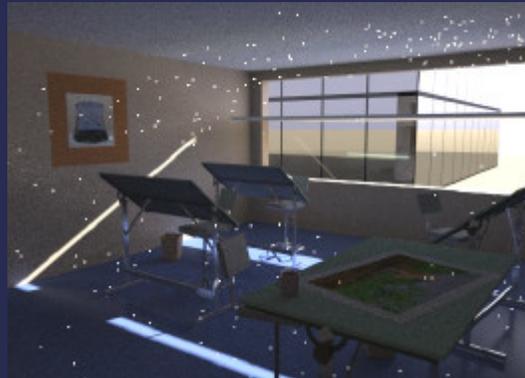


Image from Greg Ward's Siggraph ?? paper.

It's *much* worse when you have small light sources (e.g. the sun) or lots of light and dark variation (i.e. high-frequency) in your environment.

And the *Monte Carlo Integration* approach totally breaks down when you've got small light sources or high-frequency light maps.

Monte Carlo Integration

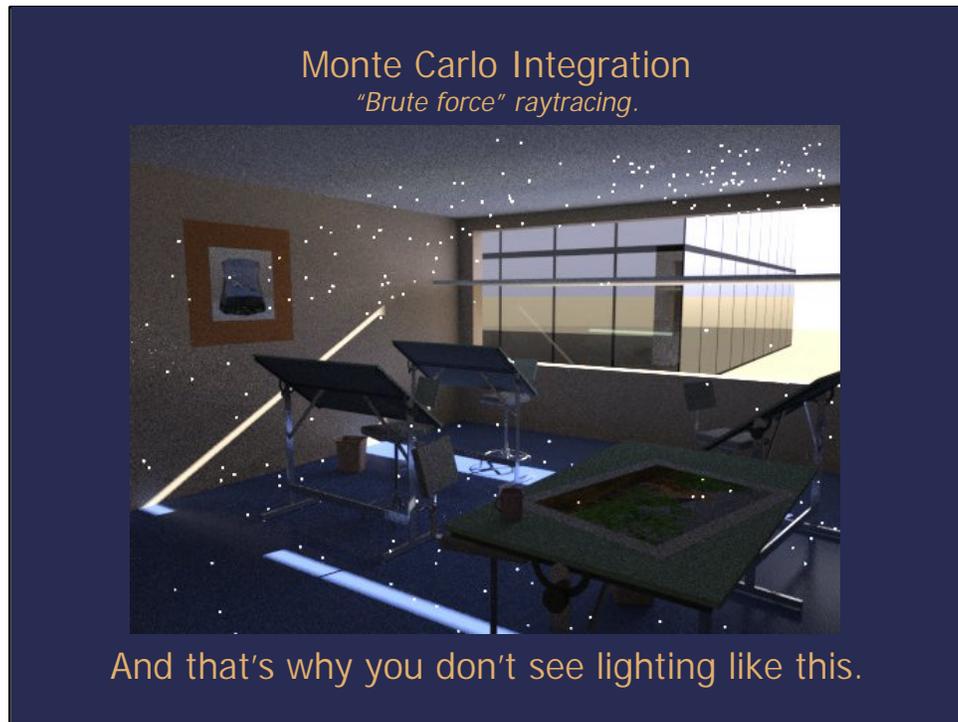
"Brute force" raytracing.



Image courtesy of Marcos.

That's why you always see "overcast" lighting like this.

For that reason, an "overcast" lighting situation is almost always used when demonstrating these kinds of renderers. (Nice image though!)



And that's why you don't see scenes with high-contrast lighting environments.

Monte Carlo Integration

"Brute force" raytracing.

Image unavailable at time
of publication.

Some people have been clever and combined the overcast skydome with some direct illumination (e.g. an area light) to yield nice results.

There are ways to deal with the limitations in some situations. The image above was rendered with an overcast lighting Monte Carlo approach combined with a large area light. It yielded some nice results, however there is still some noise in the bounced warm light on the shadow side of the hood near the grill. Nice contact shadows, though!

The Radiance Method

A two pass approach.



Image from Greg Ward's Siggraph ?? paper.

Other people have been even cleverer.

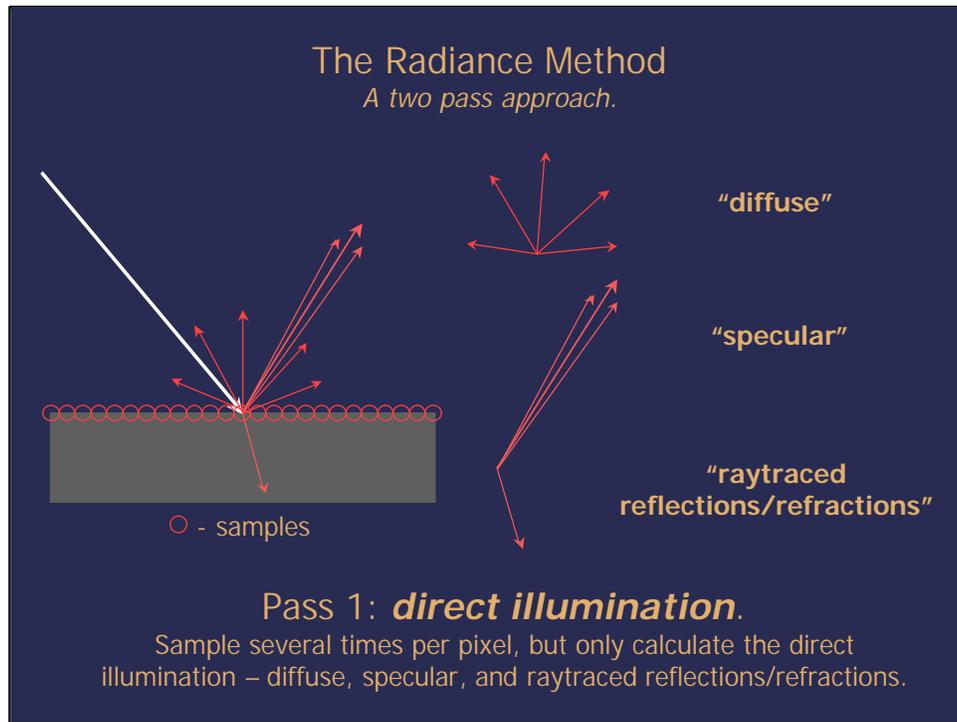
By modifying the Monte Carlo approach, some recent researchers have managed to nearly eliminate the noise and speed up render times. This is the approach used by Greg Ward's *Radiance* renderer and modified in MentalRay's "*Final Gathering*" technique. Most of Paul Debevec's research to this point (the Berkely balls, *Fiat Lux*, etc.) has been rendered using *Radiance*.

The Radiance Method

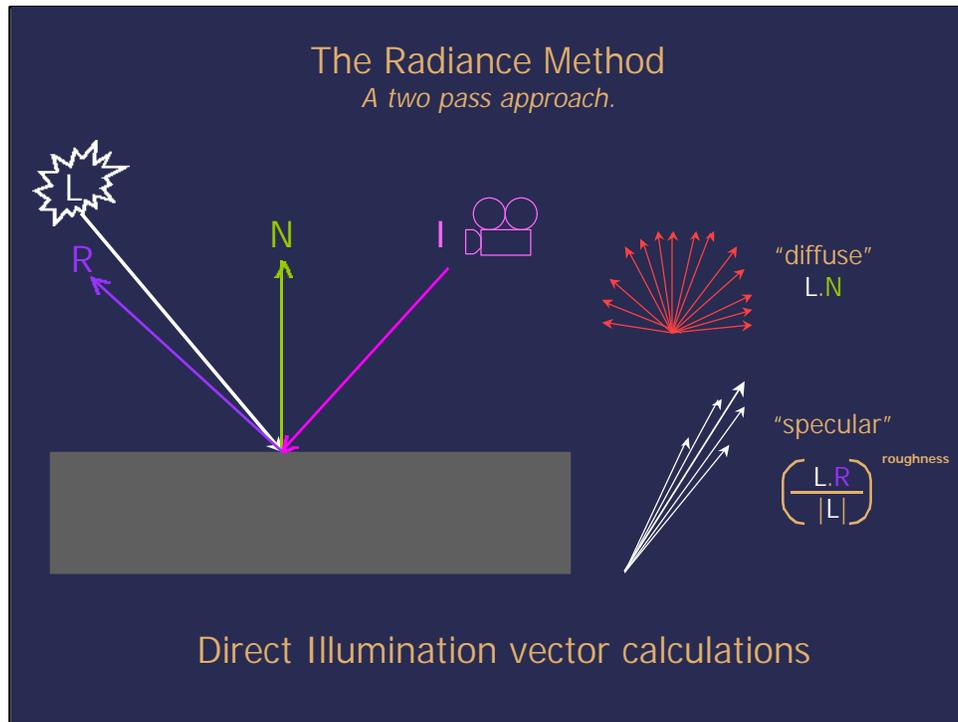
A two pass approach.

- Indirect diffuse illumination changes relatively slowly, especially over flat surfaces.
- So why not take fewer samples (one per several pixels), be more careful about taking them, and interpolate between them?
- Actually, this is not unlike *finite element methods* (radiosity) except that we're more clever about how we take our samples.

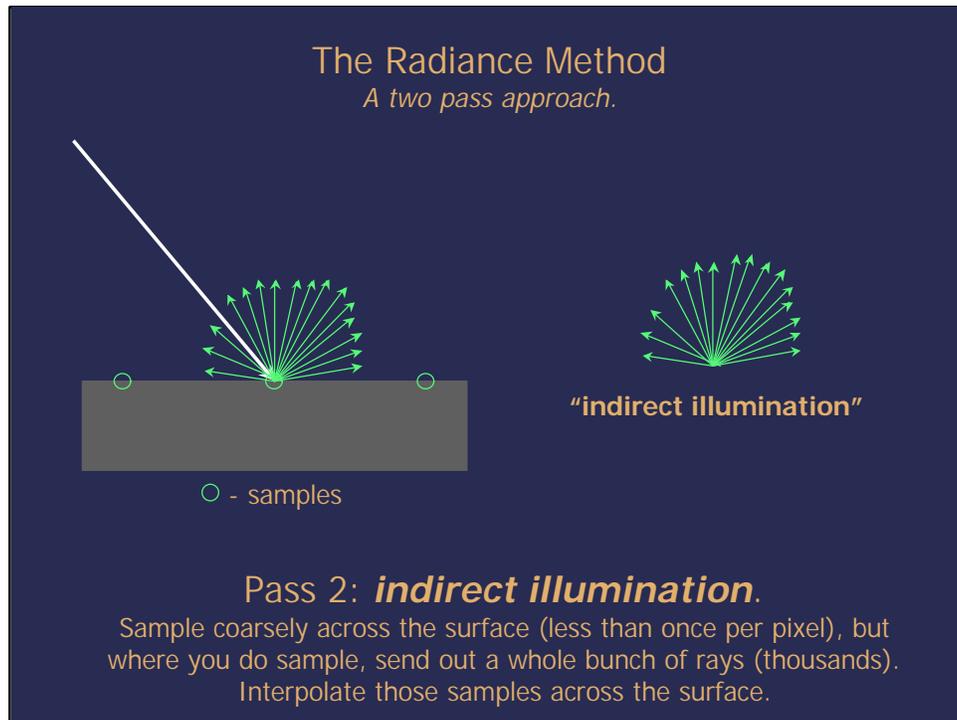
For the diffuse component, we take a low number of samples and interpolate between them.



First, we perform a direct illumination calculation just as we would in a traditional raytracer. Our diffuse, specular and reflection/refraction components are calculated based on discrete light sources. (See next slide).

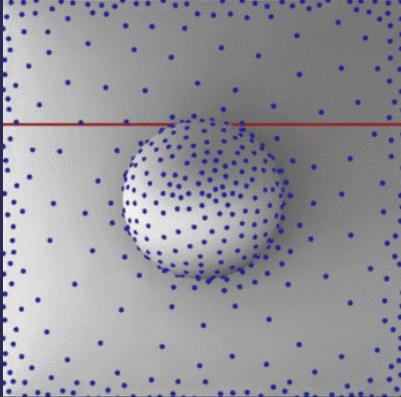


Fast vector-based algorithms for calculating direct illumination.



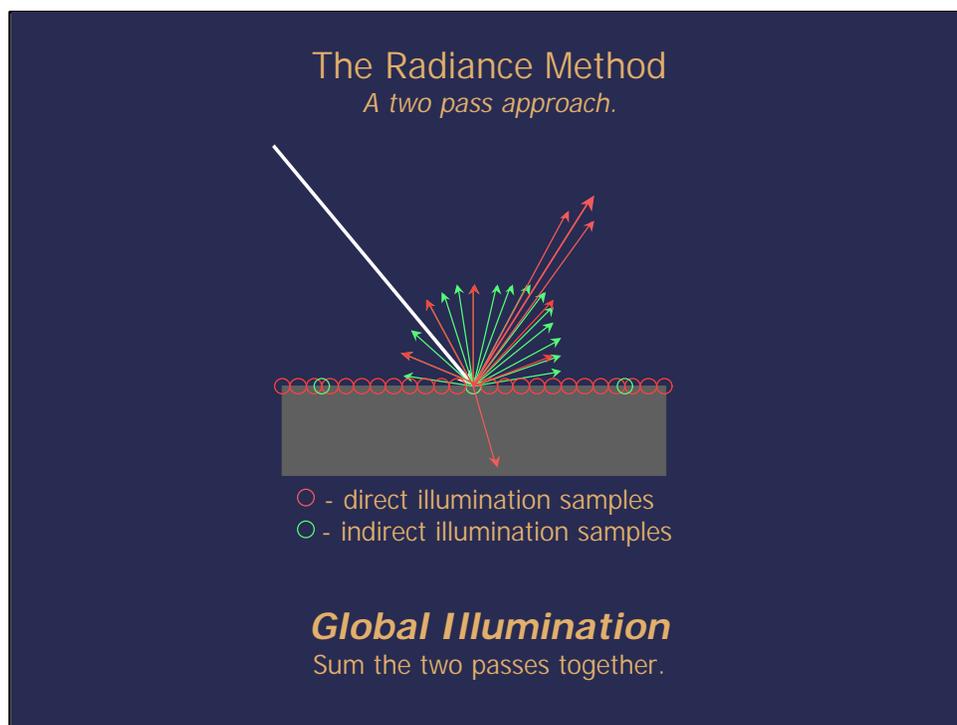
In the second pass, we send out thousands of rays based on the diffuse component of the BRDF of the material. When each ray intersects another object in the scene, we perform a direct illumination calculation. We average the results of all of these rays. This gives us a pretty good representation of the indirect light that is reaching the sampled point.

The Radiance Method
A two pass approach.



Adaptive sampling. Take more diffuse samples where the color changes quickly. (And take less where it doesn't).

We are clever about how we sample. We take more samples in areas of higher visual contrast.



By combining the two passes together, we achieve a fairly good simulation of all the light in the environment.

The Radiance Method

A two pass approach.

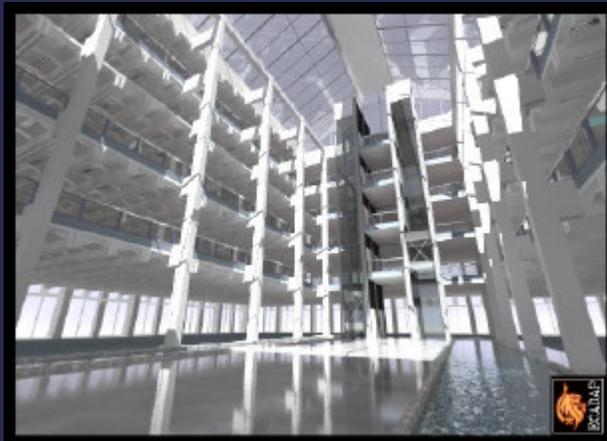


Image from Mental Images, rendered in Mental Ray.

For surfaces with a reflective (specular) component, raytrace a pass and then add it to the diffuse pass.

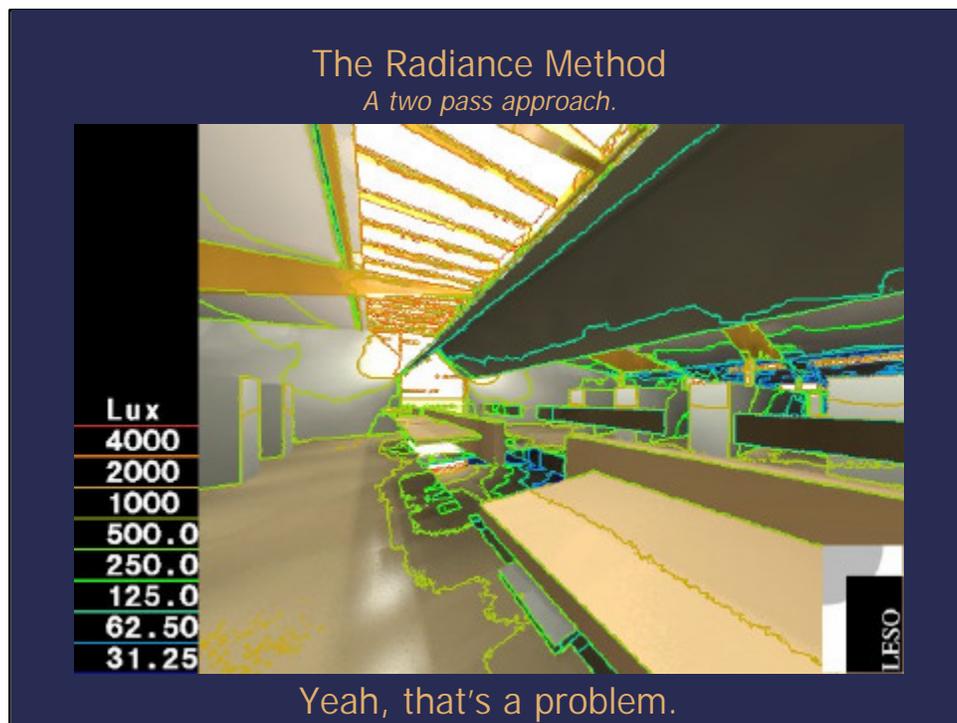
For the reflective component, we use the Monte Carlo approach, sampling multiple times per pixel. However, it isn't as prone to the Monte Carlo noise because we are only doing it for the reflective component of our lighting. So the sample rays we send out from the surface are much more focused and stay very near the angle of reflection.

The Radiance Method
A two pass approach.

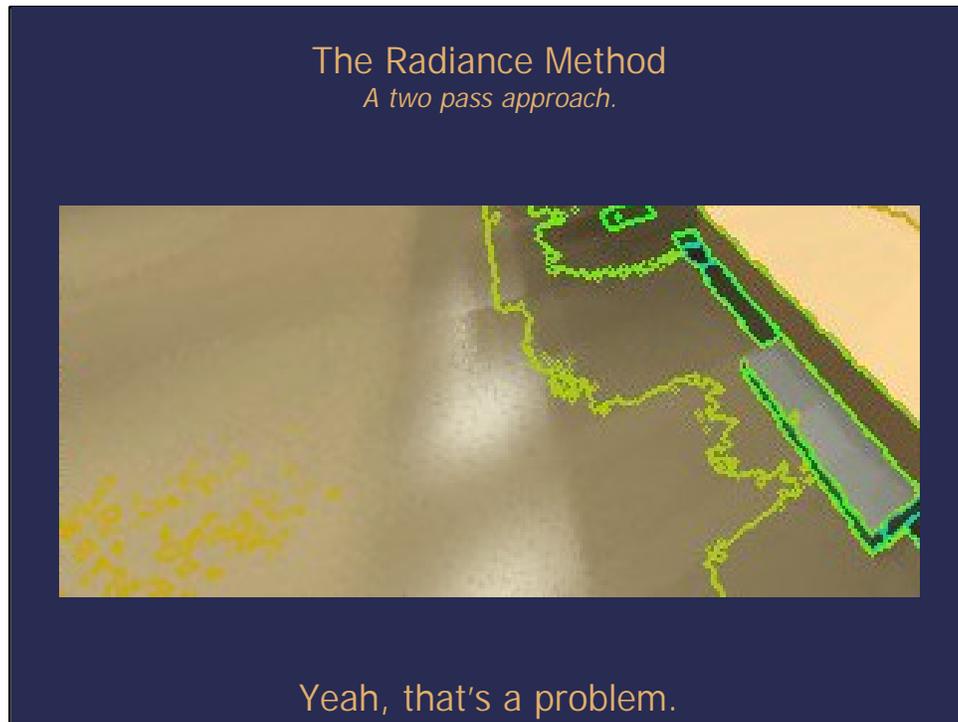


But wait, what if you have a kind of bumpy, kind of shiny surface?

We still might have some problems with bumpy, shiny surfaces.



We'd have to take our reflection samples up further in order to get around the noise. And that could get expensive.



Detail.

The Radiance Method

A two pass approach.



Image from Mental Images, rendered in Mental Ray.

Fortunately, many of the materials we are interested in have low and/or smooth reflective components.

Pretty picture.

The Radiance Method

A two pass approach.



Fortunately, many of the materials we are interested in have low and/or smooth reflective components.

Pretty picture. Note the reflections in the floor softening based on the distance of the reflected object from the surface of the floor. Nice!

The Radiance Method

A two pass approach.



Image rendered in Radiance.

Fortunately, many of the materials we are interested in have low and/or smooth reflective components.

Note the “directionality” of the light based on the angle of the blinds. Very nice!

The Radiance Method

A two pass approach.



Image from Mental Images, rendered in Mental Ray.

Fortunately, many of the materials we are interested in have low and/or smooth reflective components.

Pretty picture.

The Radiance Method

A two pass approach.



Image from Mental Images, rendered in Mental Ray.

Fortunately, many of the materials we are interested in have low and/or smooth reflective components.

Pretty picture. Gorgeous *De Stijl*.

The Radiance Method

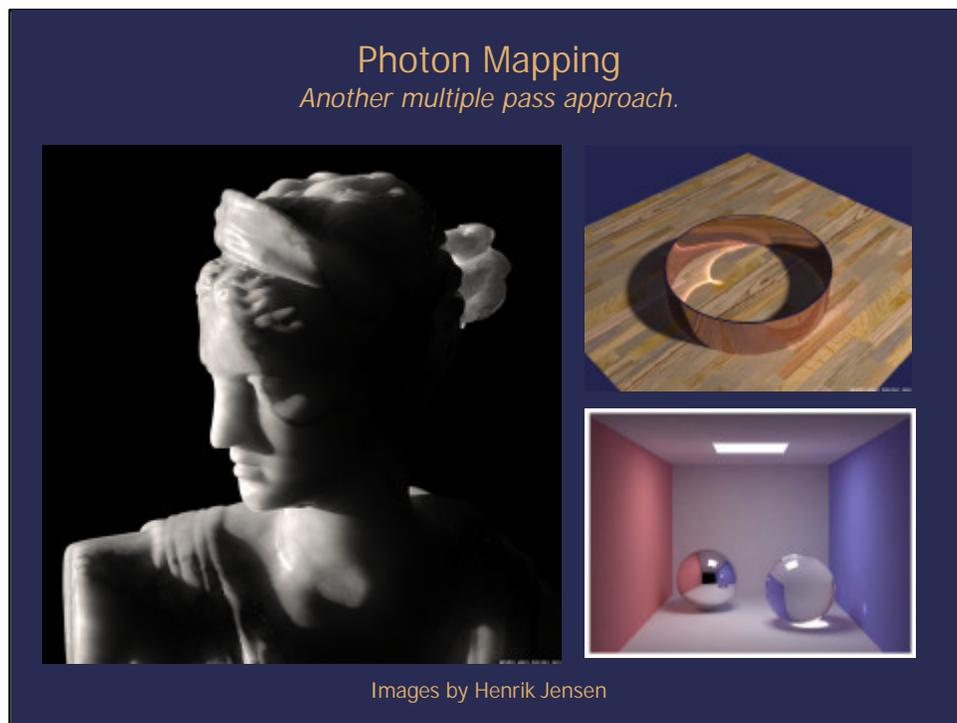
A two pass approach.

Further Optimizations

Baining Guo from Intel rendered some identical scenes in Radiance and his modified version of Radiance. He got better results by raytracing less pixels. By using his novel adaptive sampling technique, he ended up rendering only about 7% of the pixels.

Seems to work for architectural scenes.

More optimizations.



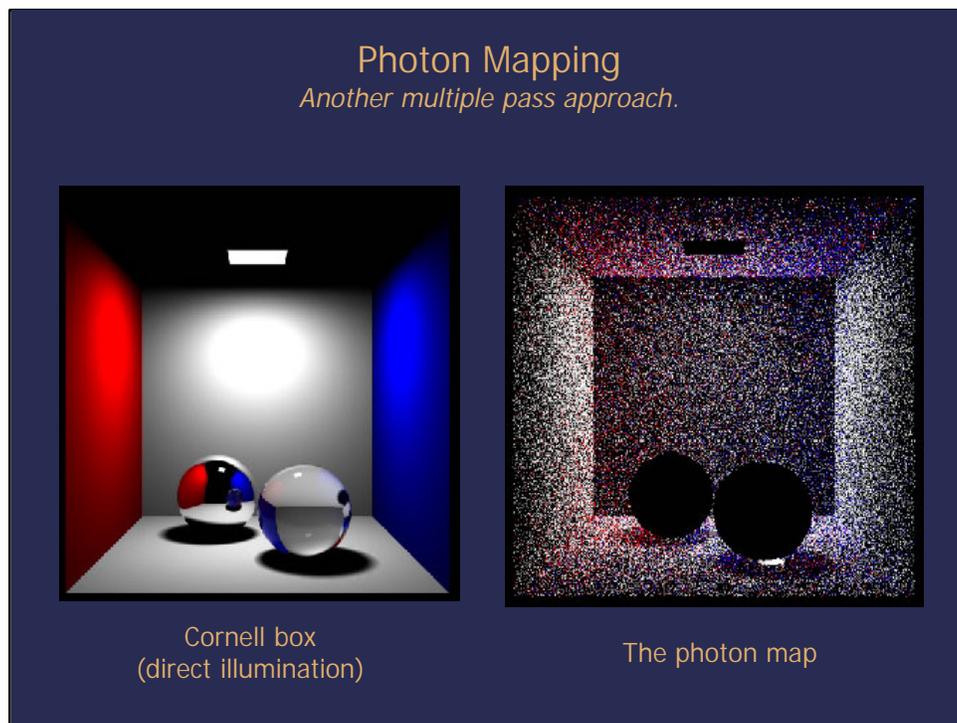
Another approach to Global Illumination is *Photon Mapping*. In addition to the various things that Monte Carlo Raytracing is good at, photon mapping is particularly successful in some areas where Monte Carlo Raytracing falls short: caustics and subsurface scattering of light.

Photon Mapping

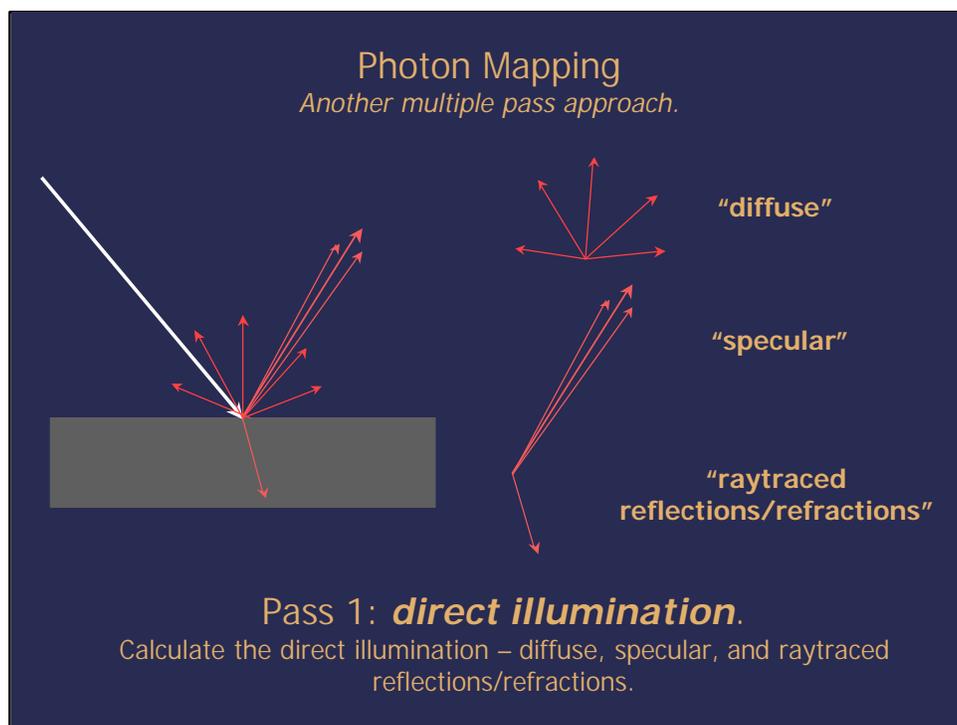
Another multiple pass approach.

- Monte Carlo raytracing relies on lots of camera rays to “find” the bright areas in a scene. Small bright areas can be a real problem. (Hence the typical “overcast” lighting).
- Why not start from the light sources themselves, scatter light into the environment, and keep track of where the light goes?

Rather than starting with camera rays and tracing backwards through the scene to find light, photon mapping starts with the light sources. We scatter photons from the light sources out into the scene and keep a record of where they go with a *photon map*.

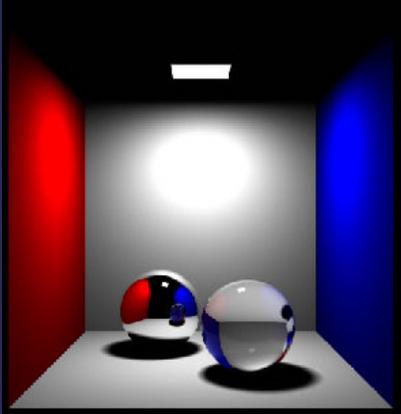


A raytraced Cornell box and an image of the *photon map* that might be generated to render its indirect illumination. The rendering of the *photon map* has no interpolation or statistical sampling.



Again, we separate the direct and indirect lighting components. We render direct illumination just as we would in a traditional raytracer. Our diffuse, specular and reflection/refraction components are calculated based on discrete light sources.

Photon Mapping
Another multiple pass approach.

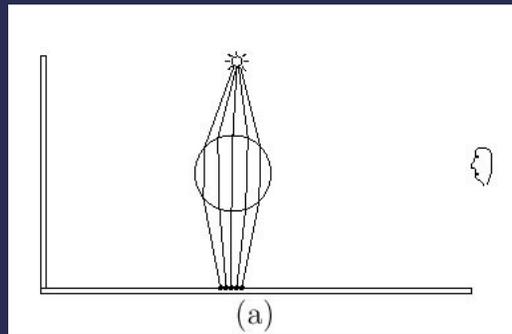


Pass 1: ***direct illumination.***
Calculate the direct illumination – diffuse, specular, and raytraced reflections/refractions.

Again, we separate the direct and indirect lighting components. We render direct illumination just as we would in a traditional raytracer. Our diffuse, specular and reflection/refraction components are calculated based on discrete light sources.

Photon Mapping

Another multiple pass approach.

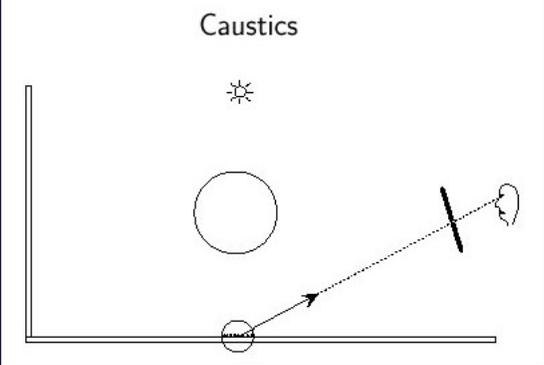


Pass 2a: *caustic illumination.*

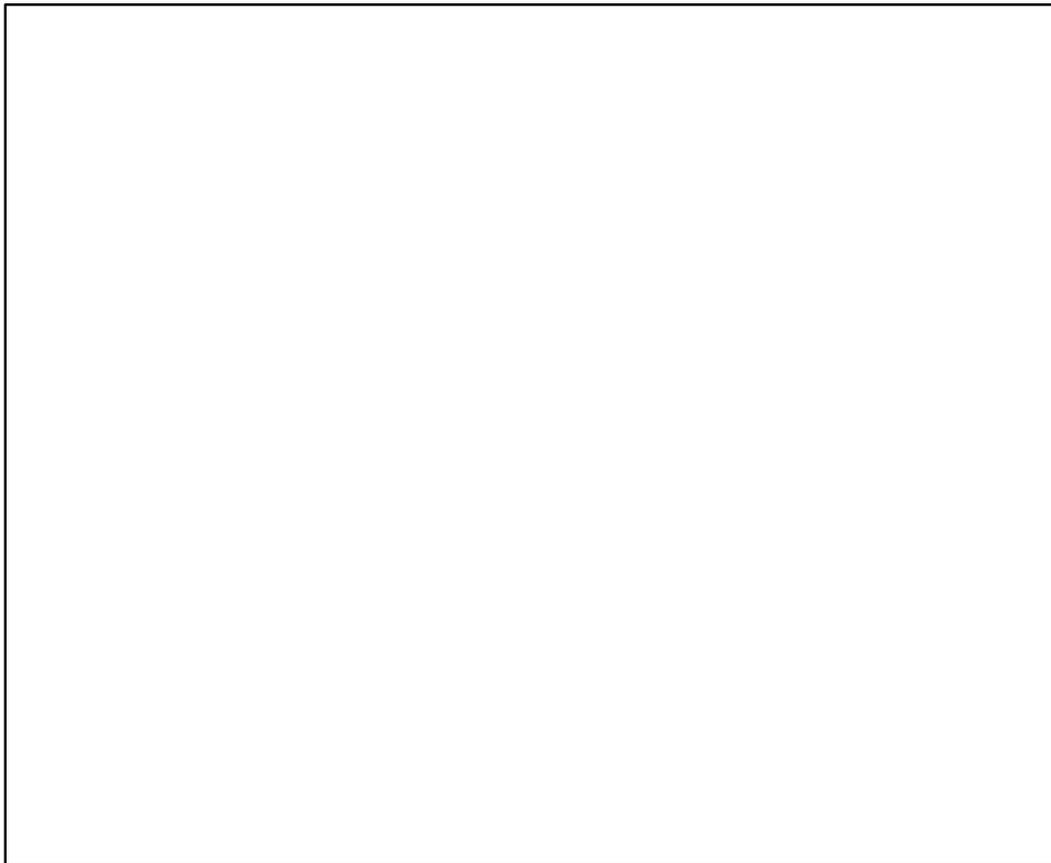
Scatter photons for caustic photon map.

Photon Mapping
Another multiple pass approach.

Caustics



Pass 2b: **caustic illumination.**
Render caustics using photon map.



Photon Mapping

Another multiple pass approach.

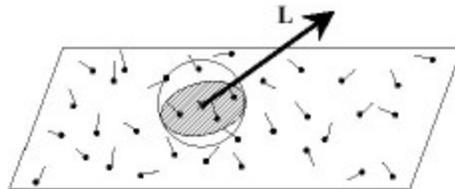
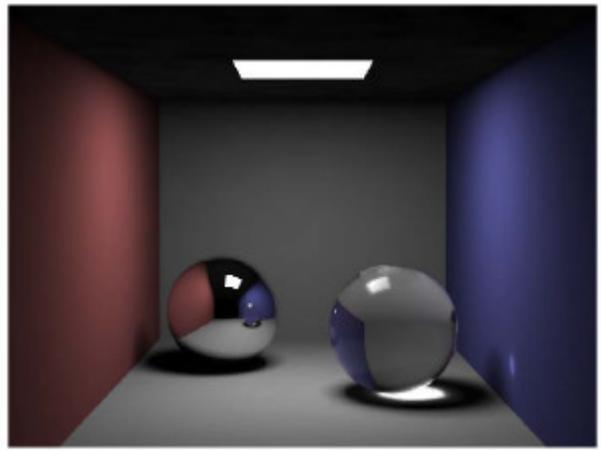


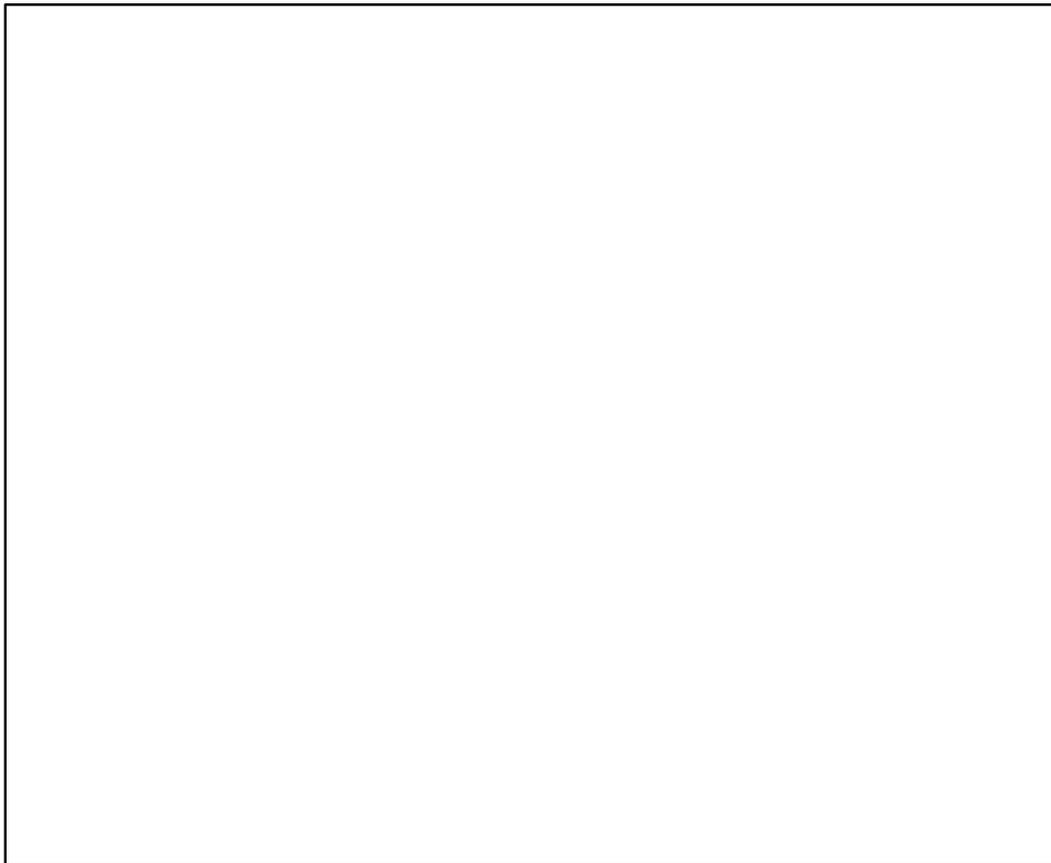
Figure 8: Radiance is estimated using the nearest photons in the photon map.

Using the photon map.

Photon Mapping
Another multiple pass approach.

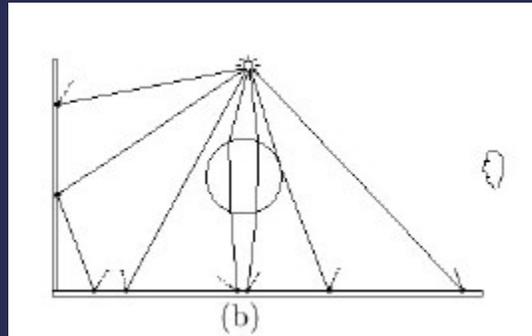


Pass 2b: **caustic illumination.**
Render caustics using photon map.



Photon Mapping

Another multiple pass approach.

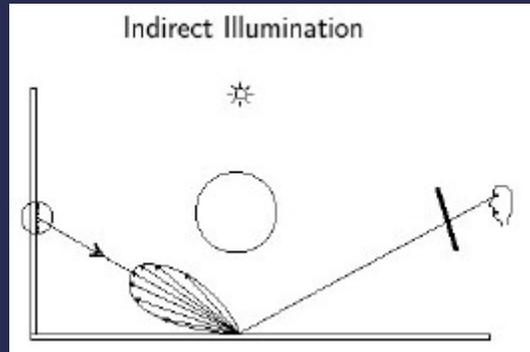


Pass 3a: *indirect diffuse illumination.*

Scatter photons to model indirect diffuse illumination.

Photon Mapping

Another multiple pass approach.

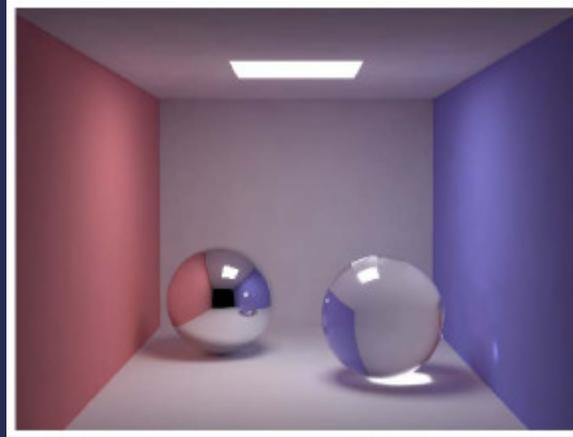


Pass 3b: ***indirect diffuse illumination.***

Render indirect diffuse illumination using photon map.

Photon Mapping

Another multiple pass approach.



Pass 3b: ***indirect diffuse illumination.***

Render indirect diffuse illumination using photon map.

Photon Mapping

Another multiple pass approach.

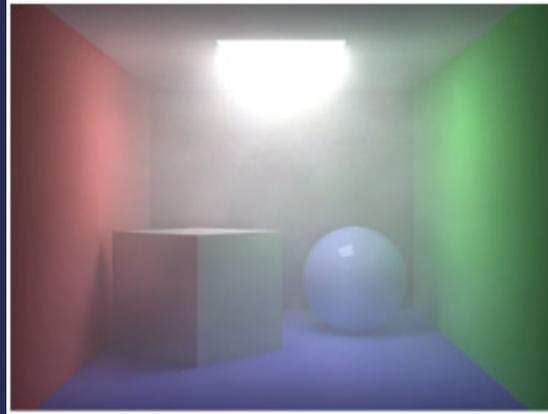


Pass 4, 5, etc.:

Subsurface scattering, participating media,
spectral separation.

Photon Mapping

Another multiple pass approach.



Pass 4, 5, etc.:
Subsurface scattering, participating media,
spectral separation.

Photon Mapping

Another multiple pass approach.



Pass 4, 5, etc.:
Subsurface scattering, participating media,
spectral separation.

Things that are *Not* (necessarily)
“Global Illumination”

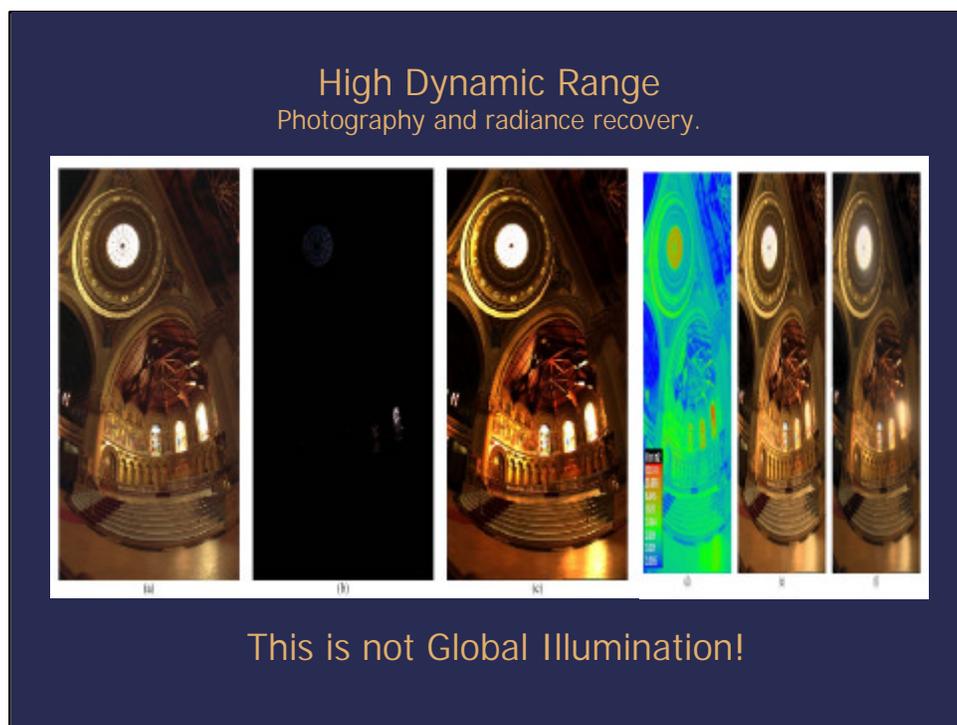
- High Dynamic Range anything (HDR)
- Glare simulation
- Image-Based rendering
- Photogrammetry
- Using an image of the environment (light probe) to light a scene

Because of some exciting recent research that has used Global Illumination in combination with other tools, some people have confused the term “Global Illumination” with the “other tools” that were used in conjunction. For this reason, I think it’s important to highlight a few neat and useful techniques that aren’t Global Illumination, but can be used with Global Illumination to create some pretty fantastic images.

High Dynamic Range

Photography and radiance recovery.

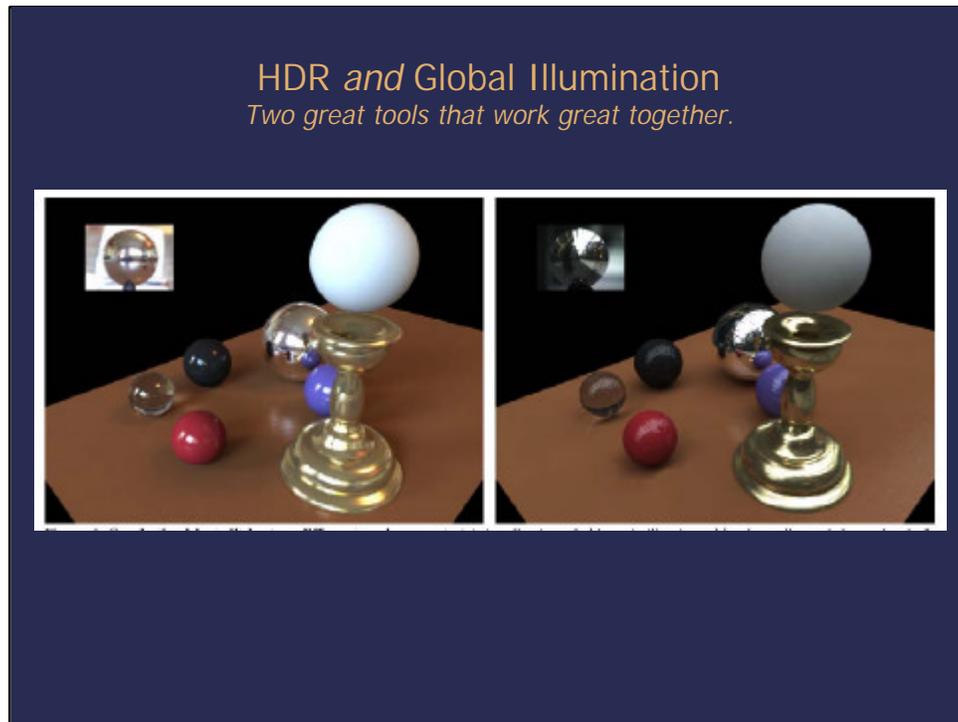
- HDR is **not** Global Illumination.
- HDR is a powerful tool for recovering accurate radiance information from a set of photographs with bracketed exposures.
- HDR is often used in conjunction with *Global Illumination* and *Image Based Rendering*, but they are *not* the same thing.
- HDR and Global Illumination **don't** mean the same thing.



The image on the left is one of the 12 images that was analyzed and combined to build the radiance map. The image in the middle shows what that radiance map would look like if it were mapped linearly into an eight-bit image. Wild, huh? The reason it is so dark is because brights are so much brighter than the darks. The false-color image on the right shows that the brightest spots are actually five-and-a-half orders of magnitude brighter than the darker areas. Debevec intentionally took his photos when the sun was shining right through the right window.

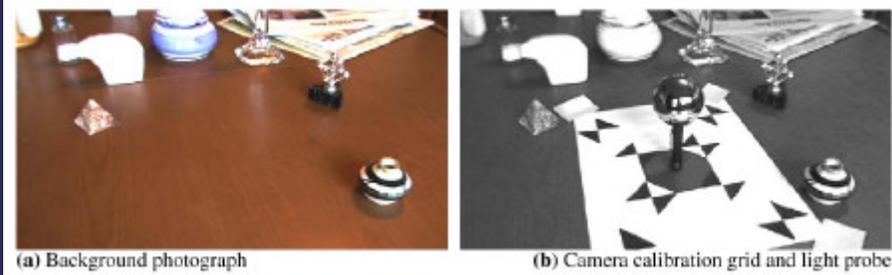


By taking HDR images of a reflective sphere in an environment, we can reconstruct the intensity and direction of all the light reaching that area of the environment. We sometimes call this a *light probe*. We can then use that light probe to render new, CG objects with lighting that very closely models the real light in the environment.

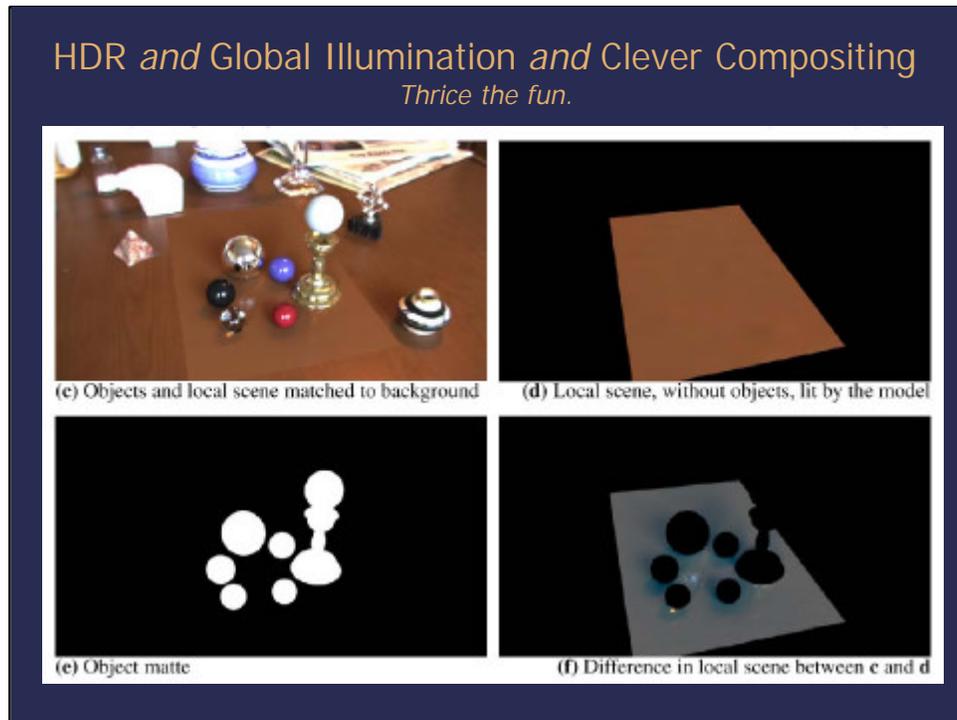


If we use a Global Illumination renderer, it isn't too difficult to get lovely shadows, caustics, reflections, and diffuse reflected light interactions.

HDR *and* Global Illumination *and* Clever Compositing
Thrice the fun.



And if we think ahead, we can do some clever compositing to drop the objects back into a real scene.



We estimate the material properties of any local objects that our synthetic objects might interact with. We render the estimated local objects with our synthetic objects (c). Then we render our estimated local scene alone (d). We render a matte for our local objects (e). And then, except for in the area of the matte, we subtract (c) from (d). This gives us (f). (f) is estimated effect of the synthetic objects on the real scene. [Note: (f) has been offset by +0.5 to show both the positive (caustics) and negative (shadow) interactions.] If we add (f) [without the offset] back into our plate, we get a fairly good approximation of the effect these objects would have on their local environment. Then we simply “over” our CG objects [using the matte (e)].

HDR *and* Global Illumination *and* Clever Compositing
Thrice the fun.



(g) Final result with differential rendering

The result is a fairly convincing composite.

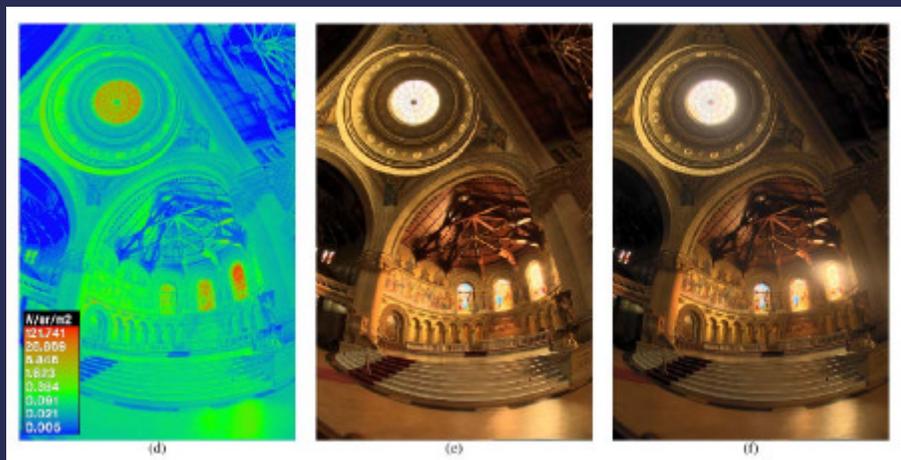
Radiance Preservation



Because in Global Illumination (and HDR) we are dealing with actual luminance values and not color values, we can do some additional things to add realism to our scenes. Because of the limited dynamic range of film, we often find that some of the highlights or shadows clip. What ends up as 1.0 in a digital image might really have been 14.0, but if your recording device only records 1.0, that's all you are going to get. However, if you used that clipped image in a texture map on an object that happened to be motion-blurred, you'd get some unrealistic results. Your highlights would become muddy as the clipped 1.0 value got sampled down to 0.5 (or whatever). What would happen in real life is the 14.0 value would get sampled down to 7.0 and, if you happened to take a picture, would again get clipped at 1.0. Your bright highlights would stay bright.

Radiance Preservation and Glare Simulation

Another level of realism missing from CG renders.



Additionally, if you kept that luminance information around, you could use it to simulate different camera exposure settings, different exposure curves, different film stocks, and even accurate glare, bloom, and flare effects.

Radiance Preservation and Glare Simulation

Another level of realism missing from CG renders.



Glare as a technique for rendering realistic images predates photographs.

Radiance Preservation and Glare Simulation

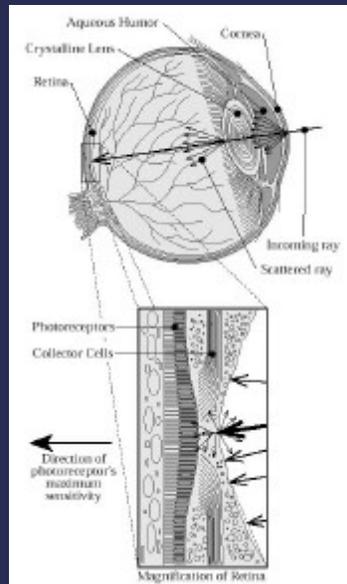
Another level of realism missing from CG renders.



It can make a significant difference in the realism of an image. This image is a composite of a tree over a sunny sky. The image on the left uses color values. The image on the right uses luminance values run through a color-render with a glare filter at the end of the comp.

Radiance Preservation and Glare Simulation

Another level of realism missing from CG renders.



The phenomenon of “glare” is based on the properties of the optical instruments and imaging devices you happen to be using. In the case of the human eye, some light is scattered as it passes through the cornea, the lens, and finally when it hits the retina.

Radiance Preservation and Glare Simulation

Another level of realism missing from CG renders.

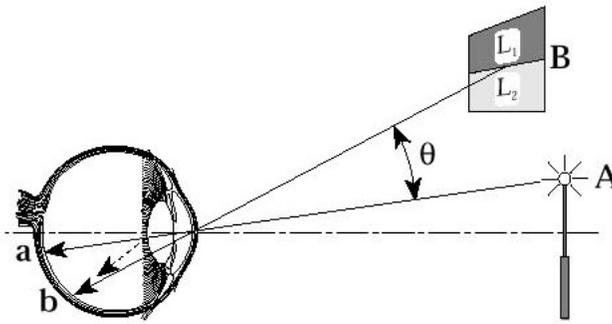
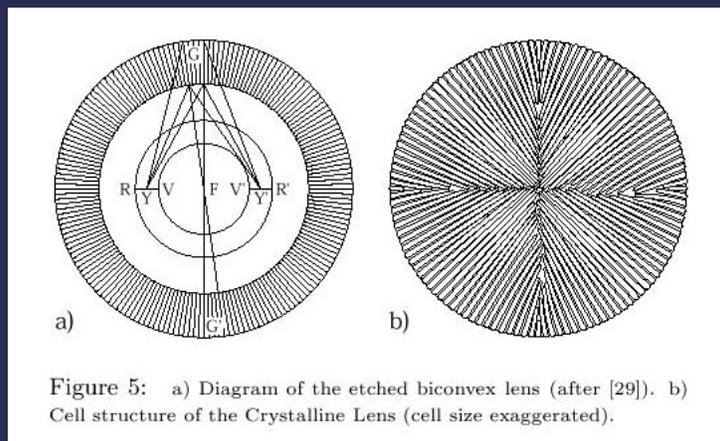


Figure 6: A reduction in contrast that results from scattered light in the eye causes a reduction in contrast that depends on θ , the angle of separation.

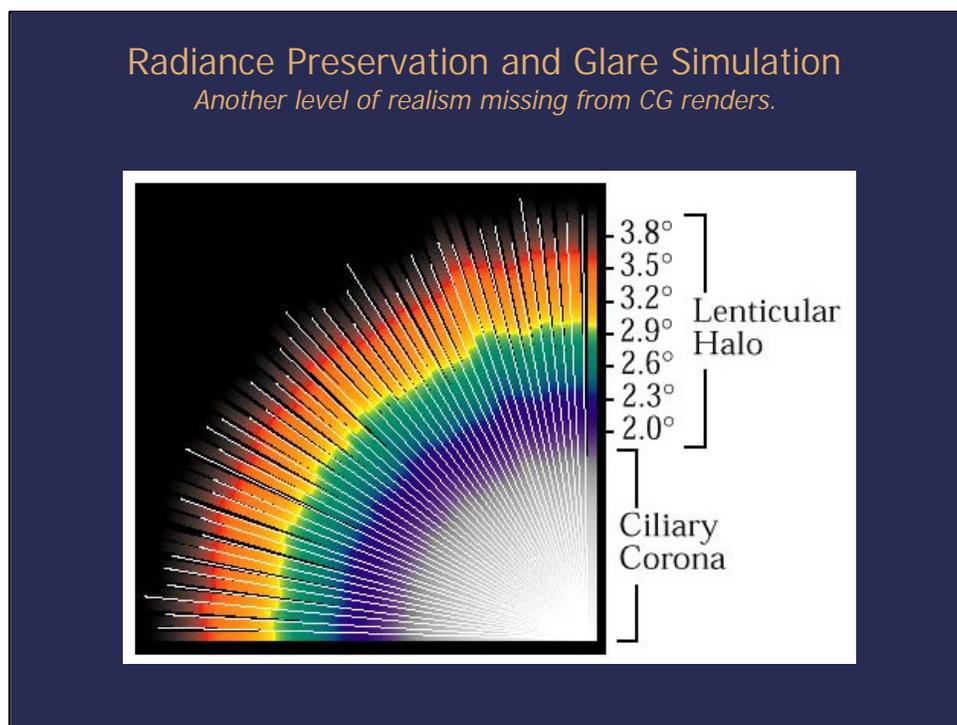
The scattering of light causes an apparent glow around the light and a reduction of contrast in surrounding dark areas.

Radiance Preservation and Glare Simulation

Another level of realism missing from CG renders.



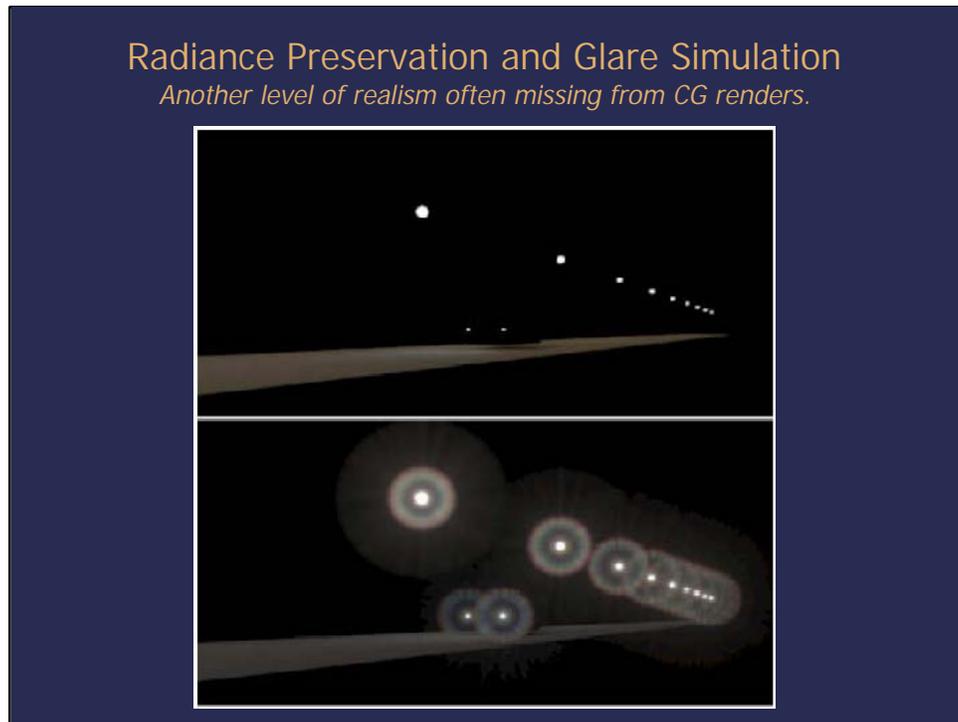
Additionally, small groove-like imperfections near the edge of the lens cause a rainbow halo to surround the glow. The rainbow halo is only visible in low-light conditions because only then is the iris open far enough for light to go through the grooves at the edge of the lens.



With small light sources, we also see bright lines radiating out from the center of the source. This is because of slight irregularities in the index of refraction of the lens. The effect isn't visible in larger light sources because the lines overlap one another and average out into a soft glow.



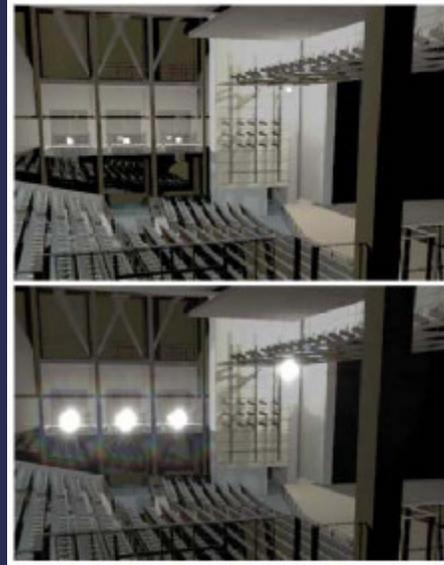
Go into a dark room and stare at a small, bright light. Neat, huh?



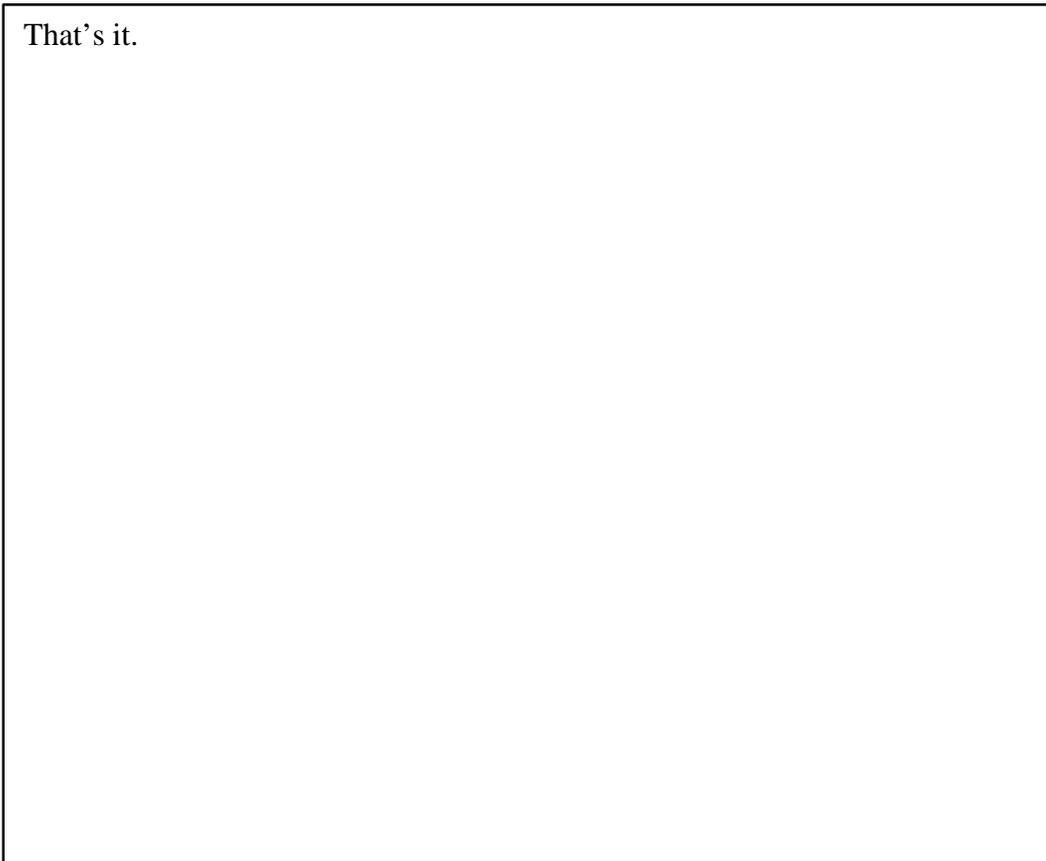
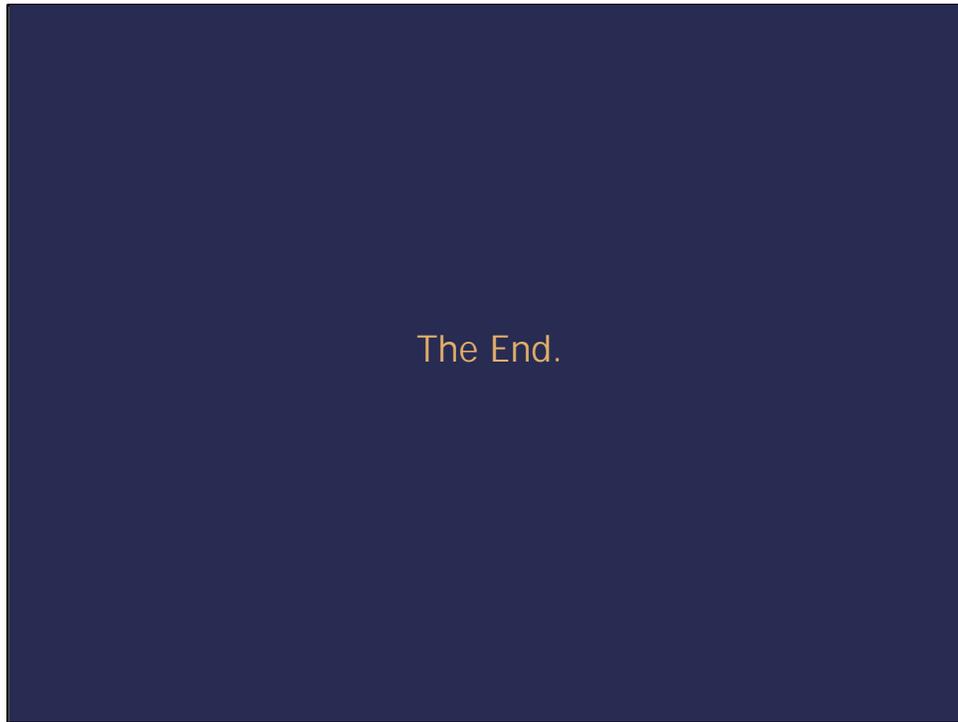
We can keep track of luminance information and simulate the glare effect with a filter. Adding the glare makes a dramatically different image.

Radiance Preservation and Glare Simulation

Another level of realism often missing from CG renders.



Again, before and after glare.



The Story of Reflection Mapping

(Title inspired by Frank Foster's "[The Story of Computer Graphics](#)")

The Quest Begins

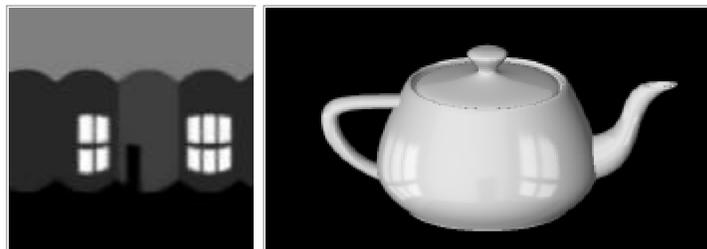
Some of the recent graphics research I've been working on builds on the techniques of reflection mapping and environment mapping developed in the late 70's and early 80's. I had a paper about the work at SIGGRAPH 98 ("Rendering Synthetic Objects into Real Scenes") which appears later in these notes.

Blinn and Newell 1976

In the paper I referenced reflection mapping using synthetically rendered environment maps as presented by Jim Blinn in 1976:

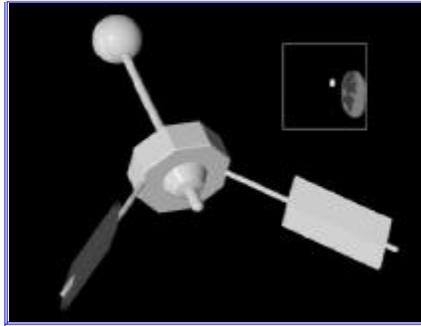
- Blinn, J. F. and Newell, M. E. Texture and reflection in computer generated images. Communications of the ACM Vol. 19, No. 10 (October 1976), 542-547.

I met with Jim Blinn in June 1999 during a visit to Microsoft Research, and by coincidence he was just in the process of resurrecting some old files, including the images from this paper. The first environment-mapped object, quite appropriately, was the Utah Teapot, with a room image made with a paint program (which Blinn wrote) as the environment map:



In the paper, Blinn also included an image of a satellite, environment-mapped with an image of the earth and the sun which he drew, shown below. Note that in both cases the objects are also being illuminated by a traditional light source to create their diffuse appearance.

Reflection Mapping History



More images from Blinn's early environment mapping work may be found [here](#).

What about Photographs?

I was surprised in writing the paper that there didn't seem to be a good reference for using real omnidirectional *photographs* as reflection maps. The seemed odd, since the technique is in common usage in the computer graphics industry, and was used in creating some of the more memorable movie effects in the 80's and 90's (e.g. the spaceship in *Flight of the Navigator* (1986), and the metal man in *Terminator II* (1991)). Furthermore, it can be regarded as one of the earliest forms of image-based rendering. So I've tried to go about figuring out where the technique came from.

The First Renderings

While at SIGGRAPH 98 in Orlando, I talked to Paul Heckbert, Ned Greene, Michael Chou, Lance Williams, and Ken Perlin to try to find out the origin of the technique. The story that took shape was that the technique was developed independently by Gene Miller working with Ken Perlin, and also by Michael Chou working with Lance Williams, around 1982 or 1983. I heard that the first two images in which reflection mapping was used to place objects into scenes were of a synthetic shiny robot standing next to Michael Chou in a garden, and of a reflective blobby dog floating over a parking lot.

A few months later, with the help of Gene Miller, Lance Williams, and Paul Heckbert, I was able to see both of these images side-by side:

Reflection Mapping History



A reflection-mapped blobby dog floating in the MAGI parking lot.
(Courtesy of Gene Miller)



A reflection-mapped robot standing next to Chou
(In hi-res courtesy of Lance Williams)

At NYIT, it was Michael Chou who carried out the very first experiments on using images as reflection maps. For obtaining the reflection image, Chou used a ten-inch "Gazing Ball", which is a shiny glass sphere with a metallic coating on the inside, usually sold as a lawn ornament. Gene Miller used a three-inch Christmas tree ornament, which was held in place by Christine Chang while he took a 35mm photograph of it:



In January 1999, [Gene Miller](#) sent over a wealth of information and images about his knowledge of the origin of the technique. Click here to [continue on](#) to [Gene Miller's](#) stories and images about the development of reflection mapping.

Williams 1983

The Chou and robot image appeared in Lance Williams's 1983 SIGGRAPH paper "Pyramidal

Parametrics". The paper introduced MIP-mapping, an elegant pre-filtering scheme for avoiding aliasing in texture-mapping algorithms. MIP-mapping has since been implemented on scores of graphics architectures and is used everywhere from video games to PC graphics to high-end flight simulators. The reflection-mapped robot image was just one example used to demonstrate the technique.

- Williams, Lance, "Pyramidal Parametrics," Computer Graphics (SIGGRAPH), vol. 17, No. 3, Jul. 1983 pp. 1-11.

Miller and Hoffman 1984

In talking to Gene Miller, I learned that in December 1982 he and Bob Hoffman submitted a paper on the technique to SIGGRAPH 83 but it was not accepted for publication. However, a revised version of this work appeared in the SIGGRAPH 84 course notes on advanced computer graphics animation:

- Gene S. Miller and C. Robert Hoffman. [Illumination and Reflection Maps: Simulated Objects in Simulated and Real Environments](#) Course Notes for Advanced Computer Graphics Animation, SIGGRAPH 84.

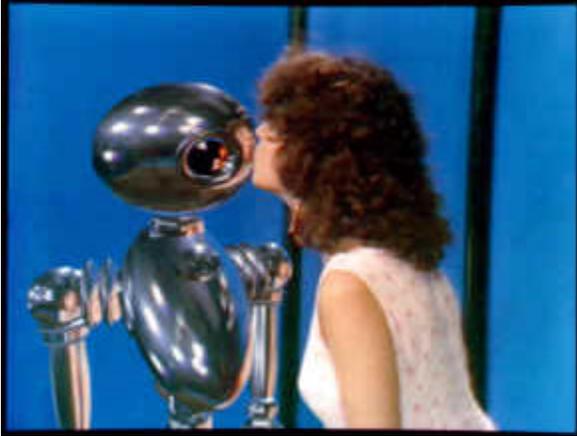
Thanks to Gene Miller, these notes can be viewed in [HTML](#) and [PDF](#) formats.

Noteworthy is that the notes suggest that reflection maps can be used to render diffuse as well as specular objects, and that issues arising from the limited dynamic range of film could be addressed by combining a series of photographs taken with different exposure levels. Both issues were problems I worked to address in my SIGGRAPH 98 paper.

Interface - 1985

In 1985, Lance Williams was part of a team at the New York Institute of Technology that used reflection mapping in a moving scene with an animated CG element. The piece "Interface" featured a young woman kissing a shiny robot. In reality, she was filmed kissing a 10-inch shiny ball, and the reflection map was taken from the reflection of the ball. To make the animation, the reflection map was applied to the robot, and the robot was composited into the scene to replace the ball.

Reflection Mapping History



"Interface", courtesy of Lance Williams.

	QuickTime - "Video" Compressor, 160 by 120, 15fps, 2.4MB
	QuickTime - "Sorenson" Compressor, 240 by 160, 15fps, 2.6MB

Interface is the first use of photo-based reflection mapping in an animation, and also its first use to help tell a story. The woman quickly kisses the robot and then heads out for the evening. As the silent robot waves goodbye, her reflected image recedes, and you can't help but think that he might have wanted to go along with her.

Interface was also worked on by Carter Burwell and Ned Greene, and the actress was Ginevra Walker. Carter Burwell later composed music for feature films such as *Raising Arizona*, *Miller's Crossing*, *The Hudsucker Proxy*, and *Barton Fink*.

Lance Williams shortly thereafter added reflection mapping (as well as texture, bump, and transparency mapping) to Pacific Data Images' renderer, which was used to create [Jose Dias' "Globo" reflection mapping images](#).

***Flight of the Navigator* - 1986**

The first feature film to use the technique was Randa Kleiser's *Flight of the Navigator* in 1986. Bob Hoffman was part of the effects team that rendered a CG shiny spaceship flying over and reflecting airports, fields, and oceans. The technique was recently revisited to render the reflective Naboo spacecraft in *Star Wars: Episode I*.

Reflection Mapping History



A still from Randal Kleiser's 1986 film Flight of the Navigator, demonstrating reflection mapping in a feature film. Image courtesy of Bob Hoffman.

Greene 1986

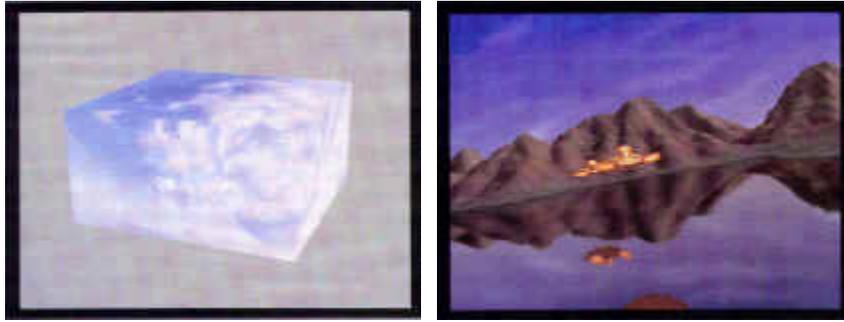
Also in 1986, Ned Greene published a paper further developing and formalizing the technique of reflection mapping. In particular, he showed that environment maps could be pre-filtered and indexed with summed-area tables in order to perform a good approximation to correct antialiasing. Greene combined a real 180-degree fisheye image of the sky with a computer-generated image of desert terrain to create a full-view environment cube.

- Ned Greene. Environment Mapping and Other Applications of World Projections. IEEE Computer Graphics and Applications, Vol 6. No. 11. Nov. 1986.



In this paper, Greene constructed an environment map using a photograph of the sky, and a rendering of the ground. In the rendering on the right, the map was re-warped to directly render the environment as well as to environment-map the ship.

Reflection Mapping History



Terminator II - 1991

Reflection Mapping made its most visible splash to date in 1991 in a film by James Cameron. Inspired by the use of reflection mapping (as well as shape morphing) in "Flight of the Navigator", Cameron used the technique to create the amazing look of the T1000 robot in "Terminator II".



Haeberli and Segal 1993

In 1993, Paul Haeberli and Mark Segal published a wonderful review of innovative uses of texture-mapping. Reflection mapping was one such application, and they demonstrated the technique by applying a mirrored ball image taken in a cafe to a torus shape.

Reflection Mapping History



A reflection mapping still from Haeberli and Segal, 1993.

- Paul Haeberli and Mark Segal. Texture Mapping as a Fundamental Drawing Primitive. Fourth Eurographics Workshop on Rendering. June 1993, pp. 259-266.

The [full paper](#) is available at [Paul Haeberli's](#) delightful [Graphica Obscura](#) website.

Paul Debevec / paul@debevec.org

<http://www.debevec.org/ReflectionMapping/>

Illumination and Reflection Maps:
Simulated Objects in
Simulated and Real Environments

Gene S. Miller
MAGI Synthavision
3 Westchester Plaza
Elmsford, New York 10523

C. Robert Hoffman 3
Digital Effects
321 West 44th Street
New York, New York 10036

23 Jul 1984
Course Notes for
Advances Computer Graphics Animation
SIGGRAPH 84

Abstract

Blinn and Newell introduced reflection maps for computer simulated mirror highlights. This paper extends their method to cover a wider class of reflectance models. Panoramic images of real, painted and simulated environments are used as illumination maps that are convolved (blurred) and transformed to create reflection maps. These tables of reflected light values are used to efficiently shade objects in an animation sequence. Shaders based on point illumination may be improved in a straightforward manner to use reflection maps. Shading is by table-lookup, and the number of calculations per pixel is constant regardless of the complexity of the reflected scene. Antialiased mapping further improves image quality. The resulting pictures have many of the reality cues associated with ray-tracing but at greatly reduced computational cost. The geometry of highlights is less exact than in ray-tracing, and multiple surface reflections are not explicitly handled. The color of diffuse reflections can be rendered more accurately than in ray-tracing.

CR Categories and Subject Descriptors:

I.3.7 [Computer Graphics]: Three-dimensional Graphics and Realism - Color, shading, shadowing and texture;

I.3.3 [Computer Graphics]: Picture/Image Generation: digitizing and scanning, display algorithms;

I.1 [Data]: Data Structures: tables

General Terms: Algorithms, Design

Additional key words and phrases: computer graphics, computer animation, shading, reflectance, illumination.

1. INTRODUCTION

The conventional point source lighting model [7] is used for efficient shading of simulated scenes, but it limits the kinds of scenes that can be rendered realistically by computer. This is because it does not adequately model the effects of area light sources (e.g. fluorescent lamps) and of objects in the environment, (i.e. the sky) that act as secondary light sources [16,19]. This is particularly noticeable on shiny surfaces.

A more realistic lighting model is also necessary when computer simulated objects are optically [17] or digitally matted [15] into photographs of real scenes. The lighting on a simulated object must be consistent with the real environment to create the illusion that it is actually in the scene.

A complete lighting model must account for all of the light that arrives at a surface which is directed towards the viewer. Blinn [2,5] obtained realistic mirror highlights by mapping the detailed reflection of complex environments. Whitted [25] provided a more accurate but slower solution for mirror highlights.

This paper presents an efficient and uniform framework for modeling light for realistic scene simulation. The idea of the illumination map is introduced -- a panoramic image of the environment that describes all the light incident at a point. Since the illumination map is an image, it can be created, manipulated and stored by conventional optical and digital image processing methods. Specific photographic and digital techniques are described.

The concept of reflection map is introduced. The shader looks up reflected light values in tables to model the appearance of specific materials under specific lighting conditions. These maps are obtained by blurring the illumination maps.

The resulting pictures have many of the reality cues associated with ray-tracing but at considerably less computational cost. The geometry of highlights is less exact than in ray-tracing, and multiple surface reflections are not explicitly handled. The color of diffuse reflections can be rendered more accurately than in ray-tracing.

2. HISTORICAL OVERVIEW

This section discusses three major illumination models. It concludes with a description of a real-time shader based on table-lookup.

2.1 Point Sources Illumination

The intensity calculation in Phong's shader [8] employs a finite number of point light sources at infinite distance. Intensity at a point on a surface is:

$$I = I_a + k_d \sum_{j=1}^{l_s} (N \cdot L_j) + \sum_{j=1}^{l_s} W(N \cdot L_j)(E \cdot M_j)^n$$

where

I = reflected intensity for the surface
 Ia = reflection due to ambient light
 kd = diffuse reflection constant
 ls = the number of point light sources
 N = unit surface normal
 Lj = the vector in the direction of the
 jth light source
 W = a function which gives the specular reflectance
 as a function of the angle of incidence
 E = the direction of the viewer
 Mj = the direction of the mirror reflection from the
 jth light source = $2(N \cdot L_j)N - L_j$
 n = an exponent that depends on the
 glossiness of the surface.

The number of calculations per point is proportional to the number of light sources. Phong shading is most realistic when the simulated environment is mostly one color (e.g. the black of outer space) or when surfaces are not too glossy. It does not deal with indirect lighting.

2.2 Ray Tracing

Whitted [25] presents a simple and general method that realistically simulates mirror highlights as well as shadows, transparency and multiple reflections. The intensity of reflected light is:

$$I = I_a + k_d \sum_{j=1}^{n_j} (N \cdot L_j) + k_s S$$

where

ks = the specular reflection coefficient
 S = the intensity of light incident
 from the reflected direction.

The cost of calculating S increase with the complexity of the environment, and ray-traced pictures are usually more costly than Phong shaded ones.

Since this method yields geometrically exact mirror highlights, the reflected environment for an object can be arbitrarily close to the object without introducing errors encountered with the illumination map method. Ray tracing is accurate for most kinds of scenes, and the results often look photographic. It does not deal, however with diffuse reflection of indirect light sources.

2.3 Mirror Highlights by Table Look-up

Blinn [2,5] simulates the mirror highlights on an object by using polar angles of the reflection direction as indices to a panoramic map of the environment. The intensity of reflected light at a point is given by a formula similar to the one used for ray tracing.

This map represents the light intensity for all indirect and direct area light sources as seen from a single point. For this reason, the results are approximate; features of the environment that are very close to the object are not correctly distorted at all points in the object. In addition, the object can not reflect parts of itself. However, for curved surfaces and restricted motion, these errors are not usually noticed, and the effect is realistic. The method is relatively fast but memory intensive.

Max [18] successfully applied a variation of this method that allows for self-reflection.

2.4 Video Lookup of Reflected Light Values

Blinn [6] and Sloan [23] present a method for real-time shading by using video lookup tables. Phong model intensity is computed for 256 different normal directions and the results are stored in the video lookup table. An image of the object is stored in a frame buffer as encoded normals, and the shaded picture is displayed on a video monitor. Lighting and reflectivity parameters can be changed quickly by editing the lookup table, however the resulting images are of low color resolution.

3. SHADING BY REFLECTION MAPS

The following algorithm illustrates shading from reflection tables. It creates a motion picture sequence of a simulated object in a real or simulated environment. For simplicity, the object is of a single surface type and the object is assumed not to move far through the environment, but may rotate about itself. The camera is free to move.

- (1) establish the environment illumination table I.
- (2) compute diffuse reflection table D.
- (3) compute specular reflection table S.
- (4) FOR every frame
- (5) FOR every pixel that is of the specified object:
- (6) determine unit vector N normal to surface.
- (7) determine unit vector E from surface to eye.
- (8) determine reflected direction $R = 2(E \cdot N)N - E$
- (9) output intensity =

$$W_d(E \cdot N) D[\text{polar } N] + W_s(E \cdot N) S[\text{polar } R]$$

In this algorithm, lines 1 to 3 initialize the tables, and lines 4 to 9 loop over animation frames.

At line 1, a panoramic image of the environment is created (see section 5). The viewpoint should be chosen so that the lighting at that location is representative of the object, e.g., the object's center. The image is transformed and stored in illumination table I which is indexed by the polar coordinates of direction (see section 4).

In lines 2 and 3 table I is convolved to produce tables D and S which are indexed by encoded directions. D, the diffuse reflection map, is the convolution or blurring of the illumination table with a diffuse reflection function. S, the specular reflection map, is the convolution of the illumination table with a specular reflection function (see section 6).

Lines 6 to 9 are executed for every pixel that represents the object of interest. Lines 6 to 8 are normally provided by conventional Phong shading.

In statement 9, vectors N and R are converted to polar coordinates which index tables D and S respectively. (Square brackets [] are used here to signify indexing.) Reflected light values are obtained by table lookup, with optional bi-linear interpolation and antialiasing. Functions Wd and Ws are used to scale the diffuse and specular reflectance as a function of viewing angle. (See Sections 7 and 8.)

4. DATA REPRESENTATION AND SPHERICAL MAPPINGS

This section considers several ways of representing environmental and reflected light.

The environmental light viewed from a point is characterized as a mapping of all view directions that are points on the unit sphere into color triplets that represent the red, green and blue light intensity from each direction. The diffuse component of reflected light is assumed to be a function of surface normals, and is a mapping of the unit sphere (normal directions) into color triplets that represent reflected intensity. Likewise, for the specular component of reflected light and the mirror reflection direction.

These mappings can then be stored as tables of color triplets that are indexed by discrete polar coordinates, each representing a small area on the sphere.

For simplicity and accuracy in interpolation and integration, it is desirable to have adjacent points on the sphere represented by adjacent indices: i.e. to have a continuous transformation. The higher derivatives should also be continuous. It is also desirable, though not possible, to have all indices represent equal areas on the sphere. This is the classic cartographic Problem: mapping a sphere onto a plane [10].

Described below are three kinds of spherical projection. Data can be transformed from one projection to another by image mapping [2,9,13].

4.1 Perspective Projection onto Cube Faces

This projection is used in the preparation of illumination maps of simulated environments and is the usual picture output of most scene-rendering systems. Place the eye at the center of the sphere and project the sphere onto the six faces of the unit cube. The front face is specified by $z > |x|$, $z > |y|$, and is mapped by

$$\begin{aligned} u &= y/Z \\ v &= x/Z \end{aligned}$$

The mapping is similar for the other five faces.

For the front face, the area of the sphere represented by each sample point is proportional to $1/z^2 = 1 + u^2 + v^2$. Thus, the resolution at the center of each face is effectively one-third the resolution at the corners.

4.2 Orthographic Projection

Each hemisphere is projected onto the xy plane. In this mapping,

$$\begin{aligned} u &= x \\ v &= y \end{aligned}$$

and there are two map arrays: one for the $z \geq 0$ hemisphere, one for the $z < 0$ hemisphere.

This is how the normal component of mirror reflection is projected onto a photographic image of a mirrored sphere (see section 5.1).

The area on the sphere represented by each sample point is proportional to $\sqrt{1 - u^2 - v^2} = 1/z$. Thus, the vicinity of the equator is severely under-sampled in the radial direction.

4.3 Polar Coordinates

Illumination data obtained in the two previous projections is transformed into polar coordinates for Illumination maps. This simplifies the interpolation and integration of light values. The north pole is identified with the y direction.

$$\begin{aligned} u &= \text{longitude } \arctan(x,z) \\ v &= \text{latitude } \arccos y \end{aligned}$$

The area on the sphere represented by each sample point is proportional to $\sin(v) = \sqrt{1 - z^2}$. Thus, the vicinity of the poles is over-sampled in the u direction.

5. CREATING THE ILLUMINATION MAP

The first step in our method is the creation of the illumination map. The illumination map may be of real, simulated, or painted environments.

About 24 bits of precision are required for the realistic simulation of natural scenes. This is because the ratio of the intensity of the sun to that of the darkest observable point is above 1,000,000:1 on a sunny day. Although this dynamic range is not reducible for film and video output, is important that it be available for accurate blurring and antialiasing.

5.1 Real Environments

When a simulated object is to be matted into a real environment, a representative location is selected and a conventional photograph made from that place. Two schemes are possible

1. Photograph the environment in six directions using a flat field lens with a 90-degree field of view to simulate projection onto a cube. The six pictures must be mosaiced into one seamless image.
2. Photograph the environment reflected in a mirrored sphere. This is relatively simple way to capture the whole environment. A silvered glass Christmas tree ball is nearly spherical and a good reflector. A moderately long focal length lens should be used to minimize the effects of perspective.

There may be small perturbations on the globe, and there can be much distortion near the edges. There is a blind spot, i.e. the areas directly behind the globe, which are not reflected in it. For these reasons, mirrored reflections should be confined to curved surfaces where these distortions would be less noticeable than for flat mirrored surfaces.

The photograph must then be digitized, registered, and color-corrected. If the matting is done optically, the color output of the film recorder should be consistent with the color of the film it is matted with. Color correction based on Newton iteration is described by Pratt [22].

The dynamic range of a real environment can greatly exceed the dynamic range of film and the film will saturate for high intensity regions (e.g. the sun). It is critical that high intensity regions are accurately recorded for computing the diffuse component; for the specular component, the accuracy of the low intensity regions is similarly crucial. Bracketing is the solution: i.e. photograph several exposures for each scene, varying the f-stop of the camera. The different exposures will then need to be registered and digitally combined into a single image.

5.2 Simulated Environments

Area light sources should be modeled as bright self-luminous patches. E.g. a fluorescent lamp would be a very bright cylinder. All reflecting objects will serve as indirect light sources. Render the scene 6 times from the point of view of the object that will be mapped, placing 6 viewing planes on the faces of a cube centered on the view point (see section 4.1).

The illumination table can also incorporate point light sources. For each point source, determine its direction and add the energy divided by distance squared into the illumination table.

5.3 Painted Environments

Environments may be created by the digitization of conventional paintings, or by digital painting.

6. CONVOLUTION - BUILDING THE REFLECTED LIGHT MAPS

For modeling each type of reflectance property, we postulate two components of reflection: diffuse and specular. Each component is represented by a reflection map or table. The diffuse map is indexed during shading by the direction of the surface normal, and the specular is indexed by the reflected direction. For mirror highlights, the specular reflection map is identical to the illumination map.

6.1 Diffuse Reflection Map

Map D contains the diffuse reflection component for each sample normal direction N. Map D is the convolution of the illumination map with a reflectance function f_d .

$$D[N] = \left(\sum_L I[L] \times \text{Area}[L] \times f_d(N \cdot L) \right) / (4 \pi)$$

where

L ranges over all sample directions
 indexing the illumination map
 I is the average light energy in direction L
 Area is the angular area re-presented by direction L
 N . L is the cosine between N and L
 Fd is the diffuse convolution function

Some examples of the diffuse convolution function are:

1. Lambert reflection, k is the diffuse reflection constant:

$$fd(x) = \begin{cases} kd * x & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

2. Self-luminous material:

$$fd(x) = kd \quad \text{for all } x.$$

Tables of size 36x72 pixels are adequate to store the Lambert reflection map. This is equivalent to a Gouraud shaded sphere with facets at every five degrees.

6.2 Specular Reflection Map

Map S contains the specular reflection component for each sample reflected direction R. Map S is the convolution of the illumination map with a reflectance function fs.

$$S[R] = \left(\sum_L I[L] \times \text{Area}[L] \times fs(R \cdot L) \right) / (4 \pi)$$

where

fs = the specular convolution function of R . L

Examples of the specular convolution function are:

1. Perfect mirror:

$$fs(x) = \begin{cases} 1 & \text{for } x = 1 \\ 0 & \text{for } x < 1 \end{cases}$$

2. Conventional Phong specular model,
 n is the glossiness parameter,
 and ks is the specular reflection constant:

$$fs(x) = \begin{cases} ks * x^{**n} & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

3. Conventional Phong specular with a clear varnish coat:

$$fs(x) = \begin{cases} ks * x^{**n} + .5 & \text{for } x = 1 \\ ks * x^{**n} & \text{for } 0 < x < 1 \\ 0 & \text{for } x \leq 0 \end{cases}$$

7. INTENSITY CALCULATION

For every pixel in the scene occupied by the object, the rendering algorithm should provide:

1. Material type
2. Surface normal N
3. Direction E from the point to the eye

The intensity at that point is then given by line 9 of the algorithm in section 3. This expression has no separate ambient light term since the environment itself is the ambience. The W_d and W_s weighting functions are used to scale the reflected light as a function of viewing angle because many surfaces are more specular at low viewing angles [3,11].

7.1 Interpolation between Tabulated Values

Line 9 quantizes direction. For surfaces of low curvature this can be quite noticeable, especially if the table is of low resolution. This can be alleviated with bilinear interpolation [13] of the table values.

8. ANTIALIASING

If a surface has high curvature, then adjacent pixels in the images will index non-adjacent pixels in the reflection maps. For glossy surfaces, small bright light sources (e.g. the sun) will get lost between the pixels, leading to highlights that break up and scintillate.

High curvature occurs with simulated wrinkles [4], at the edges of polyhedra, and when objects move far away from the eye. A pixel of a highly curved glossy surface can reflect a sizeable portion of the environment and in a number of different ways depending on its curvature [23]. For these reasons, antialiasing for reflectance maps is more critical than it is for ordinary mapped images.

8.1 Recursive Subdivision

This method is inspired by Whitted [25] and is effectively antialiasing with a box filter.

1. Compute the reflected direction at each corner of the pixel.
2. If they index into the same or adjacent table entries, then use the average of the four intensities.
3. If not, subdivide horizontally or vertically.
4. Linearly interpolate the reflected direction and renormalize. Repeat until condition 2 is met or until the regions are inconsequentially small.

8.2 Integrate the Area Spanned by Reflected Corners of Pixel

Here we integrate the area within the quadrilateral defined by the reflected corners of the pixel. We divide by the area to obtain an average value. This method is faster but less accurate than recursive subdivision. Errors however, will average out over neighboring pixels: i.e. highlights may be shifted a fraction of a pixel.

This and the previous method can be extremely costly for very curved shiny surfaces. This is because each pixel maps into many pixels in the reflection map.

8.3 Pre-Integrating the Reflection Map

A significant speedup can be obtained by pre-integrating the reflection map, whereby the number of computations per pixel is constant. This was suggested to us by a fast filtering technique devised by K. Perlin [21]. The quadrilateral spanned by a pixel is approximated by a rectangle in the integrated reflection table. Values are looked-up at the four corners and the integrated value within the rectangle is obtained by differencing. The pre-integrated table must have enough precision to accurately store the highest value times the number of pixels in the map.

When the quadrilateral is not well approximated by the rectangle, then a blurring of the highlight may occur. This blurring is less than the blur produced by pre-filtering the reflection map [12,26].

9. GENERALIZING THE REFLECTION TABLE

The Bidirectional Reflectance-Distribution Function [11,14] specifies non-isotropic reflectance as a function of four dimensions and isotropic reflectance can be described as a function of three dimensions. Convolution with the 2-dimensional illumination map results in a 3-dimensional reflectance map that completely describes isotropically reflected light as a function of eye point and surface normal. The reflection map algorithm described in section 1.4 is deficient since it gives intensity as a weighted sum of two 2-dimensional reflection maps. Additional reflection tables may prove valuable in extending the class of materials that can be accurately simulated.

For example, certain materials reflect more light back in the viewing direction: e.g. lunar dust, reflective signs, cat's eyes. Use the viewing direction to index a specular reflection table to simulate these materials.

Another example is that the refraction of light at a single surface. This is easily simulated by computing the refraction direction as a function of surface normal and direction to the eye. Use this direction to index a mirror reflection table.

10. CONCLUSIONS

Vast amounts of computer resources are used to accurately simulate light's interaction with matter. Illumination and reflection maps provide an efficient means for realistically, simulating many types of reflections. The illumination map works because it reduces a 3-D data structure (luminance at all points in space) to a 2-D one (luminance as seen from a point). This simplification reduces the calculation time with a small sacrifice in accuracy.

Other interactions -- scattering, translucency, and diffuse shadows -- are now prohibitively expensive. Perhaps the illumination map can be applied to these areas also.

ACKNOWLEDGEMENTS

We would like to thank the production staffs and management of MAGI SynthaVision and Digital Effects for providing the motivation and, support for this work. Thanks to Ken Perlin and Joshua Pines for creating the advanced SynthaVision IMAGE program which now contains "chrome". Thanks to James Blinn and the 1983 reviewers who provided helpful comments.

REFERENCES

1. Bass, Daniel H. Using the Video Lookup Table for Reflectivity Calculations: Specific Techniques and Graphics Results. Computer Graphics and Image Processing, V.17 (1981), pp. 249-261.
2. Blinn, James F. and Newell, Martin E. Texture and Reflection in Computer Generated Images. Communications of the ACM, V.19 #10 (1976), pp. 542-547.
3. Blinn, James F. Models of Light Reflection for Computer Synthesized Pictures. SIGGRAPH 1977 Proceedings, Computer Graphics, V.11 #2 (1977), pp. 192-198.
4. Blinn, James F. Simulation of Wrinkled Surfaces. SIGGRAPH 1978 Proceedings, Computer Graphics, V.12 #3 (1978), pp. 286-292.
5. Blinn, James F. Computer Display of Curved Surfaces. PhD dissertation, University of Utah, Salt Lake City (1978).
6. Blinn, James F. Raster Graphics. Tutorial: Computer Graphics, KelloG Booth, ed., (1979), pp. 150-156. (IEEE Cat. No. EHO 147-9)
7. Blinn, James F. Light Reflection Functions for Simulation of Clouds and Dusty Surfaces. SIGGRAPH 1982 Proceedings, Computer Graphics, V.16 #3 (1982), pp. 21-29.
8. Bui-Tuong Phong. Illumination for computer generated pictures. Communications of the ACM, V.18 #6 (June 1975), pp. 311-317.
9. Catmull, Edwin A. Computer display of curved surfaces. Proc. Conf. on Computer Graphics, Pattern Recognition and Data Structures (May 1975), pp. 11-17. (IEEE Cat. No. 7SCH 0981-IC)
10. Central Intelligence Agency, Cartographic Automatic Mapping Program Documentation - 5th Edition, GC 77-10126 (June 1977).
11. Cook, Robert L. and Torrance, Kenneth E. A reflectance model for computer graphics. SIGGRAPH 1981 Proceedings, Computer Graphics, V.15 #3 (1981), pp. 307-316.

12. Dungan, William. Texture tile considerations for raster graphics. SIGGRAPH 1978 Proceedings, Computer Graphics, V.12 #3 (1978), pp. 130-134.
13. Feibush, Eliot A., Levoy, Marc and Cook, Robert L. Synthetic texturing using digital filters. SIGGRAPH 1980 Proceedings, Computer Graphics, V.14 #3 (July 1980), pp. 294-301.
14. Horn, B. K. P. and Sjoberg, R. W. Calculating the reflectance map. Applied Optics, v-18 #11 (June 1979), pp. 1170-1179.
15. Kay, Douglas S. and Greenberg, Donald. Transparency for computer synthesized images. SIGGRAPH 1979 Proceedings, Computer Graphics V.13 2 (Aug. 1979), pp. 158-164.
16. Minnaert, M. The Nature of Light and Color in the Open Air. Dover Publishing Inc., New York, 1954
17. Max, Nelson and Blunden, John. Optical printing in computer animation. SIGGRAPH 1980 Proceedings, Computer Graphics, V.14 #3 (July 1980), pp. 171-177.
18. Max, N. L. Vectorized procedural models for natural terrains: waves and island in the sunset. SIGGRAPH 1981 Proceedings, Computer Graphics, V.15 #3 (.1981), pp. 317-324.
19. Newell, Martin E. and Blinn, James F. The progression of realism in computer generated images. Proceedings of ACM Annual Conf. (1977), pp. 444-448.
20. Norton, Alan, Rockwood, Alyn and Skolmoski, Philip T. Clamping: a method of antialiasing-textured surfaces by bandwidth limiting in object space. SIGGRAPH 1982 Proceedings, Computer Graphics, V.16 #3 (July 1982), pp. 1-8.
21. Perlin, Kenneth. Unpublished personal communication.
22. Pratt, William K. Digital image processing. John Wiley and Sons, New York, 1978.
23. Sloan, K. R. and Brown, C. M. Color map techniques. Computer Graphics and Image Processing. V.10 (1979), pp. 297-317.
24. Thomas, David E. Mirror images. Scientific American (Dec. 1980), pp. 206-228.
25. Whitted, Turner. An improved illumination model for shaded display. Communications of the ACM, V.23 #6 (June 1980), pp. 343-349.
26. Williams, Lance. Pyramidal Parametrics. SIGGRAPH 1983 Proceedings, Computer Graphics, V.17 #3 (1983), pp. 1-11.

Image-based Lighting in Lightwave 3D

Arnie Cachelin 2001

Lightwave 3D Background

The Lightwave 3D renderer is one of the most widely used in Film and Broadcast production because of its quality, workflow, and speed. It can now do image-based lighting due to a number of architecture enhancements in the 6.0 and 6.5 versions. One of these critical new architectures is a full precision rendering pipeline from top to bottom. Another critical ingredient is a global illumination model.

Global illumination simulations produce more accurate images of lighting in scenes, by considering more light sources than simple illumination models. Light sources include actual lights as well as their diffuse reflection from surfaces, their specular reflection, luminous surfaces, surfaces textured with HDR (high dynamic-range) images, volumetrics, and environments. These simulations produce 'pixel' values which vary from millions to millionths. Such results require floating point storage to hold both the bright values over 100% (255), and the subtle shading in the dark areas, which have values less than 1/255, and would become black in a 24 bit image. Preserving very high values as well as subtle gradations is important not only for output, but also for image-based lighting methods which require HDR images to reproduce natural lighting environments.

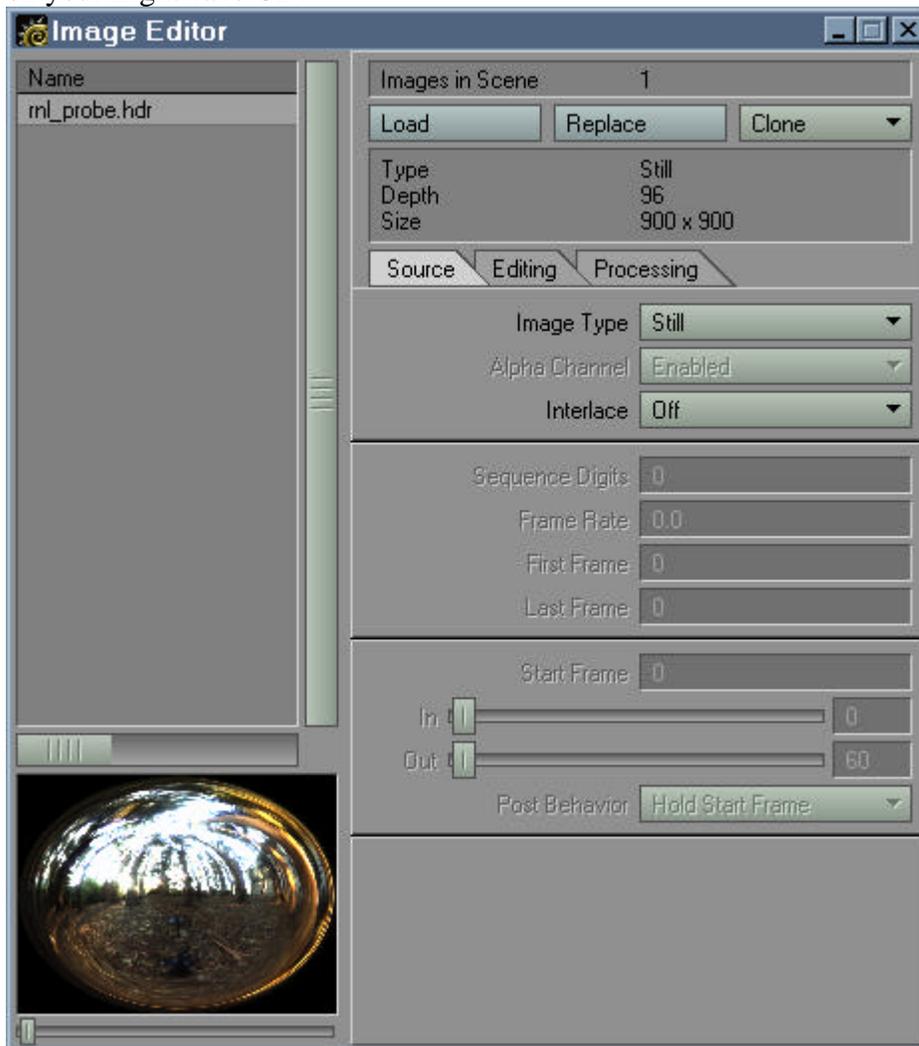
The Lightwave renderer maintains full floating point (fp) color precision upon loading HDR image formats like Tiff Log-LUV, Radiance RGBE, and the native "Flexible Format". Any image pre- or post- processing is performed with fp precision by plug-ins or basic color correction. This precision is preserved through texturing and compositing operations as well, as it must be through ALL phases of rendering. Finally these full-precision, possibly HDR images are saved in an HDR format, or exposed onto a lower precision 'print' (a 24 bit image). The extra image precision has a cost in terms of memory, and it wouldn't necessarily be worthwhile for standard ray tracing or scanline rendering. It is, however, very useful for global illumination rendering, which models the subtle effects of indirect lighting.

Setting Up a Simple Image-Based Lighting Scene

The example below uses the ImageWorld environment plugin, which uses the light-probe type of spherical image to place the entire Lightwave scene inside of the environment in which the picture was taken. This environment will illuminate the scene, be visible as a backdrop, and show up in reflective surfaces even if they use the background for reflections rather than ray tracing.

1. Load and position some objects for a pleasing composition in front of the camera. I tend to use the standard cow object, which was referred to as the teapot of the 90's.
-

2. Load an HDR light-probe image with the Load button on the Image Editor. These images are available on the web (look for the light-probe image page or the Virtual light-probe page), or on your Lightwave CD.

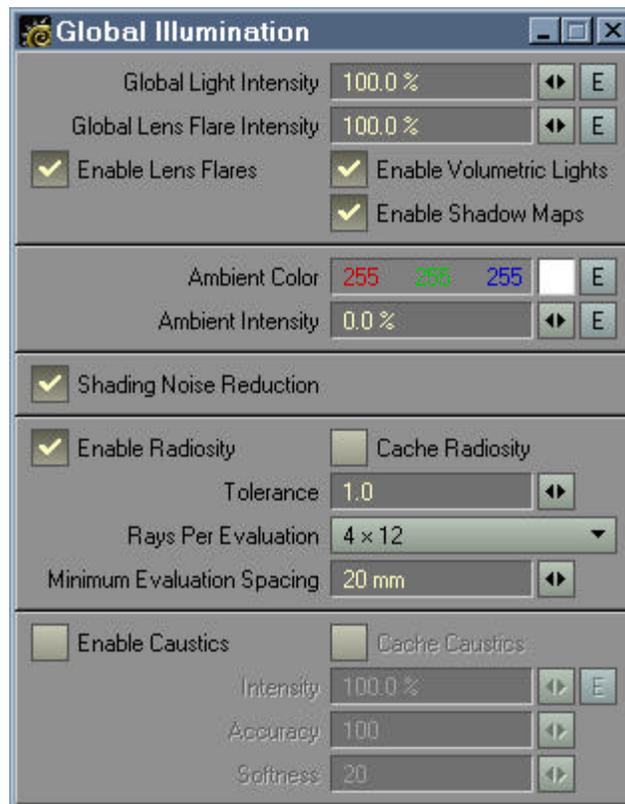


3. In the Scene Backdrop panel, select the Image World Environment plug-in and double-click it to edit its properties. Set the light-probe image for Image World to use the image just loaded.



4. Edit your light properties to turn off "Affect Diffuse" and "Affect Specular" for any lights in the scene. All lighting will come solely from the HDR environment image. This environment image will also show up in reflective surfaces.
5. In the Global Illumination panel, enable Radiosity and Shading Noise Reduction. The radiosity settings will have to be optimized for time vs. quality tradeoffs on a scene-by-scene

basis.



6. Set the Render Display to "Image Viewer FP" in the Render options panel. This will preserve the full precision image in the display buffer, so exposure adjustments can be tested rapidly.
7. Hit F9 for a preview render. When it is complete, select Image Controls from the ImageViewer's File menu and enable the exposure adjustment. Tweak the white and black point values to taste.
8. To begin the process of tweaking the scene's lighting, some handy parameters to adjust are the global illumination brightness, the brightness in ImageWorld, and the global illumination parameters.

Image Illumination Tips

Lightwave's global illumination rendering is based on the photon mapping algorithm for computing lighting in space, which samples the lighting contributions from different directions at every point on a surface. This time consuming procedure will find hot spots from lights in HDR images used for environments, but if the spots are small, or the samples too few, the results will be grainy or speckled. Increasing the sampling to improve the accuracy of image illumination increases render times. Adding real lights to the scene to match/replace images of lights will improve time and quality, since real lights are found without sampling.

Because the sampling of an environment image is sparse, details and small hot spots are likely to be missed. This means that high resolution images are not necessary for illumination, and may cause uneven lighting intensities over surfaces.

Because the environment plug-in displays fractions of the image spread over the field of view, low resolution images will appear blurry. To approximately match a light-probe image to output resolution, one needs $360/HFOV$ (~10) times greater resolution in the light-probe image.

The scene's Ambient light intensity encapsulates the strength of higher order diffuse bounces. It should be above 0%.

Rendering Light Probe Images

Real light probe images are made by photographing a mirrored sphere, and using multiple exposures to recover an HDR image. Virtual light probe images can be rendered from any Lightwave environment simply by rendering a mirrored sphere. Interesting HDR environments can be generated using the SkyTracer, or Texture Environment plug-ins. Some examples of this can be found on the Virtual Light Probe gallery web site.

Image Exposure Tips

The variations in lighting on global illumination renders can be very subtle, and a process of 'exposing' these HDR images can bring details out of dark renders. To take advantage the full precision rendered, save renders in an HDR format, so alternate exposures can be applied and tested using the HDRExpose filter, or Virtual Darkroom. You can also test exposures with the FP Image Viewer's exposure controls; try reducing the black point for dark images, or raising the white point for hot images. Settings determined interactively this way can be used in the HDRExpose filter for final batch rendering. Some examples of the exposure correction for HDR images can be found in the supporting materials.

LightWave [6] Notebook:

High Dynamic Range Images Exposed

Arnie Cachelin

HDR Exposure Filter

LightWave [6] can render high dynamic range (HDR) Images, which may have very subtle shading differences as well as very bright areas. Because normal displays and printing technology have a more limited dynamic range than LightWave's Global Illumination engine, some colors in such images will be too bright or too dark to display. They will become either white or black. on screen or in print. The HDRExpose filter can select how to spend the limited output dynamic range (typically 255:1) to best display the HDR data. Like gamma correction, this process can bring up detail in shaded areas. HDRExpose rescales the image colors based on a "black point", which is the highest level which will be black in the output. This is expressed as a percentage of the standard 24 bit black point (1/255). The "white point" is the lowest level that will be white. The default value, 1.0, usually maps to 255 in 24 bit imaging. Raising the white point brings detail out of the bright areas at the expense of the darker, while lowering the black point uses more colors in the darker areas.

The standard HDR image below has hot spots with values in excess of 5000. Adjusting the black point from 50% to 1% shows how much detail was hidden in the shadows. The [original image](#), and details on its creation are available.

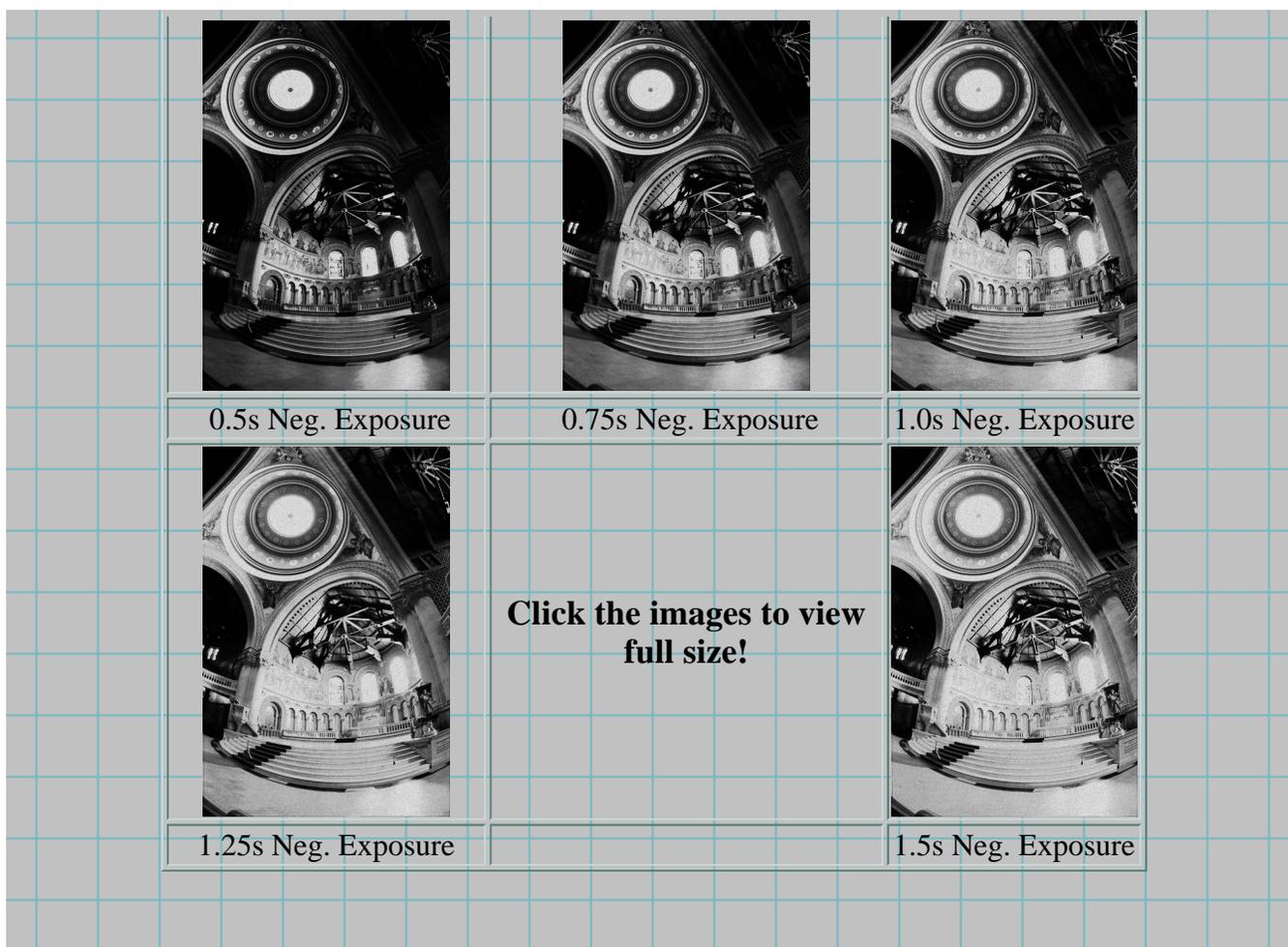


Base Image: 100%	Black Point: 50%	Black Point: 20%	Black Point: 10%
			
Base Image: 100%	Black Point: 5%	Black Point: 2%	Black Point: 1%

Virtual Darkroom Filter

Making photographic prints from film also requires a restriction of an image's dynamic range. The original image, captured in the chemical emulsion on film goes through two exposure processes that re-map the dynamic range. The first to creates a negative, which is then used in the second pass to make a positive print. The Virtual Darkroom image filter simulates these two transformations using light response parameters from actual black and white or color film to match the results of photographic processing. This complex plug-in can be used to control the exposure of HDR images, while adding some film artifacts like grain and halo which may enhance the image's apparent naturalism. The following images were exposed using the default B&W film stock, and varying only the negative exposure time:





References and Resources:

[Recovering High Dynamic Range Radiance Maps from Photographs](http://graphics3.isi.edu/~debevec/Research/HDR/)

<http://graphics3.isi.edu/~debevec/Research/HDR/>

[Light Probe Image Gallery](http://graphics3.isi.edu/~debevec/Probes/)

<http://graphics3.isi.edu/~debevec/Probes/>

[HDRView](http://graphics3.isi.edu/~debevec/FiatLux/hdrview/)

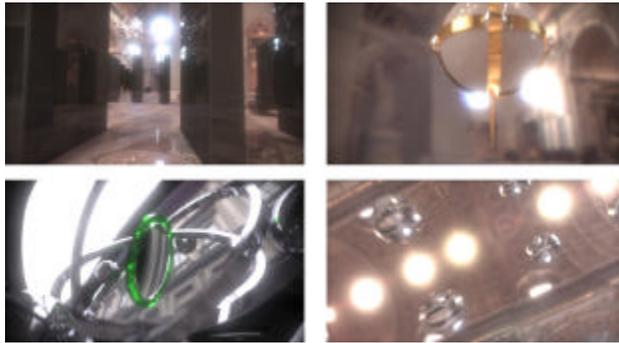
<http://graphics3.isi.edu/~debevec/FiatLux/hdrview/>

Image-Based Modeling, Rendering, and Lighting in *Fiat Lux*

Paul Debevec
University of California at Berkeley
387 Soda Hall #1776
Berkeley, CA 94720-1776
debevec@cs.berkeley.edu

Introduction

This animation sketch presents how image-based modeling, rendering, and lighting were used to create the animation *Fiat Lux* from the SIGGRAPH 99 Electronic Theater. The film features a variety of dynamic objects realistically rendered into real environments, including St. Peter's Basilica in Rome. The geometry, appearance, and illumination of the environments were acquired through digital photography and augmented with the synthetic objects to create the animation. The film builds on the techniques of *The Campanile Movie* and *Rendering with Natural Light* from SIGGRAPH 97 and 98.



The Imagery and Story

Fiat Lux draws its imagery from the life of Galileo Galilei (1564-1642) and his conflict with the church. When he was twenty, Galileo discovered the principle of the pendulum by observing a swinging chandelier while attending mass. This useful timing device quickly set into motion a series of other important scientific discoveries. As the first to observe the sky with a telescope, Galileo made a number of discoveries supporting the Copernican theory of the solar system. As this conflicted with church doctrine, an elderly Galileo was summoned to Rome where he was tried, convicted, forced to recant, and sentenced to house arrest for life. Though honorably buried in Florence, Galileo was not formally exonerated by the church until 1992. *Fiat Lux* presents an abstract interpretation of this story using artifacts and environments from science and religion.

The Technology

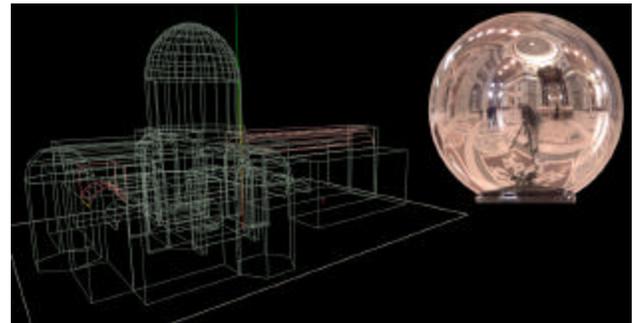
The objects in *Fiat Lux* are synthetic, but the environments and the lighting are real. The renderings are a computed simulation of what the scenes would actually look like with the synthetic objects added to the real environments. The techniques we used represent an alternative to traditional compositing methods, in which the lighting on the objects is specified manually.

The environments were acquired in Florence and Rome; the images in St. Peter's were taken within the span of an hour in accordance with our permissions. To record the full range of illumination, we used high dynamic range photography², in which a series of exposures with varying shutter speeds is combined into a single linear-response

radiance image. Several scenes exhibited a dynamic range of over 100,000:1.

The appearance and illumination of each environment was recorded with a set of panoramic images and light probe measurements³. Each light probe measurement was made by taking one or two telephoto radiance images of a 2-inch mirrored ball placed on a tripod; each provided an omnidirectional illumination measurement at a particular point in space. Several radiance images were retouched using a special high dynamic range editing procedure and specially processed to diminish glare.

We constructed a basic 3D model of each environment using the Façade photogrammetric modeling system¹. The models allowed us to create virtual 3D camera moves using projective texture-mapping, as well as to fix the origin of the captured illumination. The light probe images were used to create light sources of the correct intensity and location, thus replicating the illumination for each environment. The illumination was used to "un-light" the ground in each scene, allowing the synthetic objects to cast shadows and appear in reflections³. The dynamic objects were animated either procedurally or by using the dynamic simulator in Maya 1.0. Renderings were created on a cluster of workstations using Greg Larson's RADIANCE global illumination system to simulate the photometric interaction of the objects and the environments. The final look of the film was achieved using a combination of blur, flare, and vignetting filters applied to the high dynamic range renderings.



Contributors: Christine Cheng, H.P. Duiker, Tal Garfinkel, Tim Hawkins, Jenny Huang, and Westley Sarokin. Supported by Interval Research Corporation, the Digital Media Innovation program, and the Berkeley Millennium project.

See also: <http://www.cs.berkeley.edu/~debevec/FiatLux>

References

- 1 Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. *Modeling and Rendering Architecture from Photographs*. In SIGGRAPH 96, August 1996.
- 2 Paul Debevec and Jitendra Malik. *Recovering High Dynamic Range Radiance Maps from Photographs*. In SIGGRAPH 97, August 1997.
- 3 Paul Debevec. *Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and Dynamic Range Photography*. In SIGGRAPH 98, July 1998.

Light Probe Image Gallery

Updated 2

New! SIGGRAPH 2001 Course

Paul Debevec will teach a half-day course on Image-Based Lighting at SIGGRAPH 2001. The evolving web page for the course is [here](#). The course will cover in detail both the theory and practice of using light probe images to illuminate computer-generated objects.

Introduction

A light probe image is an omnidirectional, [high dynamic range](#) image that records the incident illumination conditions at a particular point in space. Such images were used in [Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography](#) at SIGGRAPH 98 to illuminate synthetic objects with measurements of real light, and in a more recent [SIGGRAPH 2000 paper](#) to illuminate human faces. Two animations [Rendering with Natural Light](#) and [Fiat Lux](#) have been made using the image-based lighting technique. The recently released product LightWave 3D 6.0 implements image-based lighting; see [below](#) for links to some sample renderings made using our probe images. Some of these images were assembled from high dynamic-range panoramas, others were acquired by taking one or two high dynamic range images of a mirrored ball (see [Reflection Mapping](#)).

RADIANCE Image Format and Viewing on Unix

Light Probe images are measurements of light in the real world, and thus are high dynamic range. As a result, these images are provided in the [RADIANCE](#) Synthetic Imaging System **.hdr** image format (Described in Greg Ward Larson's "Real Pixels" article in *Graphics Gems II*.) They can be viewed on X11 display using RADIANCE's **ximage** program and/or converted to floating point data using the "pvalue -df -H -h" program from the RADIANCE package. RADIANCE images also use the more traditional but somewhat ambiguous **.pic** filename extension, so feel free to rename any of the **.hdr** images to **.pic** if your software requires this.

Viewing on Windows 95/98/2000/NT

The images may also be viewed on Windows 95/98/2000/NT using our [HDRView](#) program which is downloadable [here](#). Press the + and - keys to re-expose the image and use the mouse to pan, zoom, and bring up HDRView's menu.

Raw Floating-Point Image Data

The raw floating-point format images are 4-byte single precision, big-endian, with interleaved color planes. There is now also a directory with [little-endian](#) versions of these float images (which are useful if you're using a PC rather than an SGI, for example). Take note of the image dimensions so that you can load them correctly. We have also developed a very simple floating-point version of the Portable Pixmap **.ppm** format (**.pfm** for Portable Floatmap) which we will support shortly, along with the floating point versions of the TIFF format.

Making Your Own Light Probes

If you're interested in creating your own high dynamic range images, the **mkhdr** package written by H. R. Duiker, Tim Hawkins, and Paul Debevec is available here. To create a light probe image, acquire a high dynamic range image of a mirrored ball. There will be more information and software to do this on this site in the near future.

Light Probe Images

The following light probe images were created by taking two pictures of a mirrored ball at ninety degrees of separation and assembling the two radiance maps into this registered dataset. The coordinate mapping is such that the center of the image is straight forward, the circumference of the image is straight backwards, and the horizontal line through the center linearly maps azimuthal angle to pixel coordinates.

Thus, if we consider the images to be normalized to have coordinates $\mathbf{u}=[-1,1]$, $\mathbf{v}=[-1,1]$, we have $\theta = \text{atan2}(\mathbf{v}, \mathbf{u})$, $\phi = \pi * \sqrt{(\mathbf{u} * \mathbf{u} + \mathbf{v} * \mathbf{v})}$. The unit vector pointing in the corresponding direction is obtained by rotating $(0,0,-1)$ by ϕ degrees around the y (up) axis and then θ degrees around the x (forward) axis. If for a direction vector in the world (D_x, D_y, D_z) , the corresponding (\mathbf{u}, \mathbf{v}) coordinate for the light probe image is $(D_x * r, D_y * r)$ where $r = (1/\pi) * \text{acos}(D_z) / \sqrt{D_x^2 + D_y^2}$.

Note that each light probe image represents a full 360×360 degrees, or 4π steradians. Also note that the mapping being used is different than the mapping one observes in a mirrored ball -- the mapping we're using provides better sampling near the backwards-facing directions.

Note on downloading:

The light probes need to be transferred as binary files. However, since the .hdr format begins with a text header some browsers (e.g. Netscape) interpret them as text and perform character conversion, corrupting the files. Internet Explorer does not appear to have this problem. To avoid this problem, all of the probes are available as one gzipped tar file [all_probes.tar.gz](#); it's 18,557,917 bytes. If you're using a PC rather than a Unix-based machine, you can download all of the images as [all_probes.zip](#) which is 20,686,564 bytes (reportedly PC's sometimes don't untar Unix .tar.gz files correctly).





Grace Cathedral, San Francisco
1000 × 1000
Dynamic range: 200,000:1

Angular map: [.hdr](#) | [.float.gz](#)

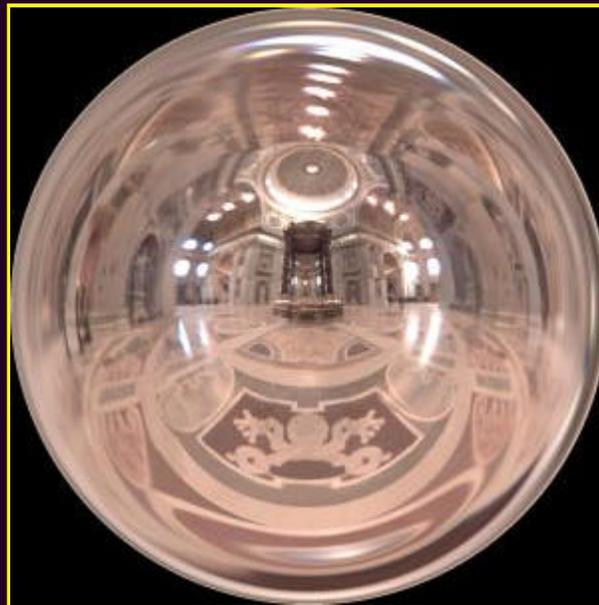
Used as the illumination environment for [Figure 6](#) of the [SIGGRAPH 98 paper](#). Assembled from two radiance images of a mirrored sphere taken with a Sony VX1000 digital video camera (approx. ten image per sample.)



Eucalyptus Grove, UC Berkeley
900 × 900
Dynamic range: 5000:1

Angular map: [.hdr](#) | [.float.gz](#)

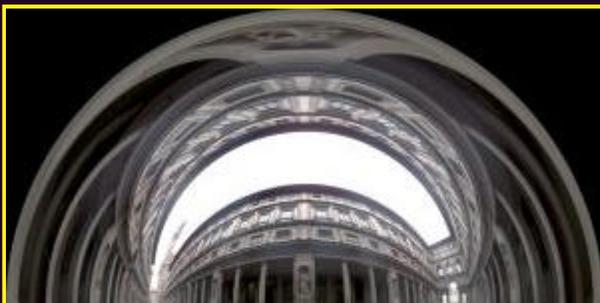
Used as the illumination environment for [Render with Natural Light](#); acquired similarly to the Grace Cathedral image..

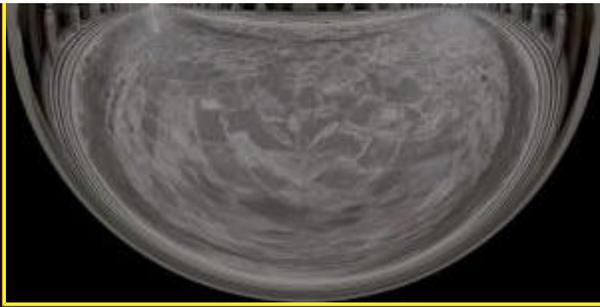


St. Peter's Basilica, Rome
1500 × 1500
Dynamic range: 200,000:1

Angular map: [.hdr](#) | [.float.gz](#)

Used as some of the illumination data for the main sequences of *Fiat Lux*. Assembled from two radiance images of a mirrored ball and then touched up at the bottom using Photoshop. Each Radiance image was assembled from six exposures, three stops apart, from a Kodak DCS520 digital camera.





The Uffizi Gallery, Florence
 1500 × 1500
 Dynamic range: 500:1

Angular map: [.hdr](#) | [.float.gz](#)

Used as the illumination environment for the middle sequence of *Fiat Lux*. Assembled from a set of eighteen images forming a full-view panorama; it has the greatest detail of any of the images on this site.



Galileo's Tomb, Santa Croce, Florence
 1000 × 1000
 Dynamic range: 7000:1

Angular map: [.hdr](#) | [.float.gz](#)

Used as the illumination environment for the final sequence of *Fiat Lux*, taken similarly to the St. Peter's Basilica probe but not retouched.



[Skull](#) lit with this image in LW3D by Terrence Walker

Single Image Light Probes

The following light probe images were created from single high dynamic range images of a mirrored sphere. Since just a single image was used, the images show the camera and the photographer and they are not well sampled in the area that is opposite the camera.





Kitchen at [2213 Vine St](#)
640 x 640
Dynamic range: 2000:1

Angular map: [.hdr](#) | [.float.gz](#)

Used as the illumination environment for Figures 2, 3, [4 \(a\)](#), 5, and 8 of the [SIGGRAPH 98 paper](#).



Overcast Breezeway, Soda Hall
640 x 640
Dynamic range: 1000:1

Angular map: [.hdr](#) | [.float.gz](#)

Used as the illumination environment for Figure [4\(b\)](#) of the [SIGGRAPH 98 paper](#).



Campus at Sunset
640 x 640
Dynamic range: 2000:1

Angular map: [.hdr](#) | [.float.gz](#)



Funston Beach at Sunset
640 x 640
Dynamic range: 800:1

Angular map: [.hdr](#) | [.float.gz](#)



[Skull](#) lit with this image in LW3D by Terrence Walker

More Images

Arnie Cachelin of NewTek, Inc. has generated some synthetic Light Probe images in the high dynamic range TIFF log-luv format, available on his [Virtual Light Probe](#) page.

There are some additional high dynamic range images on the [main high dynamic range images page](#).

Illuminating Synthetic Objects with Real Light

Using RADIANCE

We're currently putting together some sample RADIANCE model files that use these images to illuminate synthetic objects, and we'll put them online when they're together.

Using LightWave 3D 6.0

Image-Based Lighting has been implemented in [NewTek's LightWave 3D 6.0](#) package, and the Light Probe Image Gallery has been included on their distribution CD. Some sample images of [skulls rendered using our probe images](#) have been rendered in LW3D by Terrence Walker. Eliza Ra has prepared a [brief web page](#) about using image-based lighting in both LightWave 6.0 and in Radiance, complete with some sample files. We're putting a more comprehensive tutorial in connection with the [SIGGRAPH 2001 course on Image-Based Lighting](#).

Images Copyright © 1998, 1999 Paul Debevec. If you'd like to use them, [let me know](#).

Paul Debevec / paul@debevec.org

HDR Shop

SIGGRAPH 2001 Technical Sketch

HDR Shop is a computer application, currently under development, designed to view and edit High Dynamic Range images. High dynamic range images [1] are pictures that can capture a much greater range of light intensities than standard photographs or computer images, making them useful for image based lighting and post-render processing.

Photographs from traditional cameras do not record the amount of light over a certain level. All the bright points in a photo are white, making it impossible to tell any difference in intensity. The standard technique to acquire high dynamic range images that capture this missing information is to take several photographs at different exposures (making each photo progressively darker, without moving the camera), until the bright lights no longer saturate. The sequence of photographs can then be analyzed to derive the light intensity of each point in the scene.

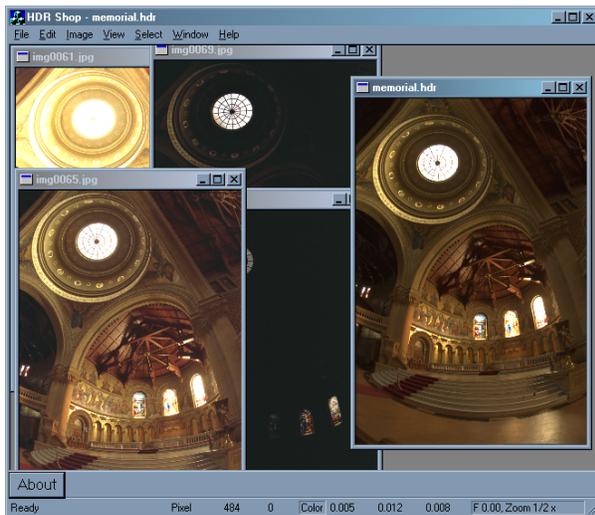


Figure 1: In HDR Shop, a sequence of low dynamic range images (left) can be compiled into a single high dynamic range image (right).

Whereas traditional image editors work with 8 or 16 bit images, HDR Shop is built from the ground up to work correctly with HDR images. All operations are done with linear floating-point numbers. In many cases, this simplifies the code, as well as providing more correct output.

For the purpose of real-time display of the images, however, it is important to quickly convert linear floating-point images to 8-bit RGB with the appropriate gamma curve. The standard gamma formula involves an exponentiation, which is slow. In the interest of speed, we have found it useful to approximate this calculation by constructing a lookup-table, indexed by the most significant bits of the floating-point values. For common gamma values of 1.4 ~ 2.2, it suffices to use 16 bits (8

Chris Tchou and Paul Debevec
USC Institute for Creative Technologies
tchou@ict.usc.edu, paul@debevec.org

exponent bits and 8 mantissa bits) to reduce the error below rounding error.

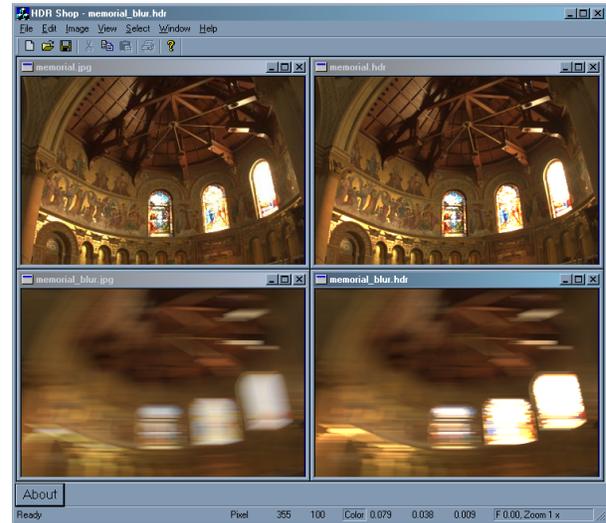


Figure 2: Comparison of HDR Shop's horizontal motion blur on a low dynamic range image (left) vs. a high dynamic range image (right).

Aside from resampling, cropping, and mathematical operations, HDR Shop also supports transformations between most common panoramic formats, facilitating the use of HDR panoramas in image based lighting [2].

HDR Shop can also automatically export a low dynamic range copy of any image to an external image editor. Changes to the LDR image are then incorporated into the HDR image, allowing the use of existing tools to modify HDR images.



Figure 3: St. Paul's Cathedral panorama, originally in cube-map format (left), converted in HDR Shop to latitude-longitude (upper right), mirrored ball and light probe formats (lower right).

References

1. Paul Debevec and Jitendra Malik. [Recovering High Dynamic Range Radiance Maps from Photographs](#), In SIGGRAPH 97, July 1997.
2. Paul Debevec. [Rendering Synthetic Objects Into Real Scenes: Bridging Traditional and Image-Based Graphics With Global Illumination and High Dynamic Range Photography](#), In SIGGRAPH 98, August 1998.

A Real Time High Dynamic Range Light Probe

SIGGRAPH 2001 Technical Sketch

In order to successfully composite CG elements into live action scenes it is important that the lighting of the CG objects match the lighting of the scenes they are being composited into. One technique that has been used to reproduce the incident illumination in a real scene is to acquire a high dynamic range photograph of a mirrored ball and then use this light probe image as a source of illumination for image based lighting [1].

Previous Work

Currently, in order to create a high dynamic range image of a mirrored ball one must take an iterative series of photographs with the exposure value of each image being stopped down by a given increment from the exposure value of the one before. Later, each of the images are assembled into a single high dynamic range image using a program such as *HDR Shop* [2]. If an artist wished to accurately illuminate a CG object traveling through a complex lighting environment, it would be necessary to capture these iterative photographs at numerous locations (ideally at every frame) along the object's path. Clearly, this would be an ambitious task.

Technique

Our solution for creating a real time high dynamic range light probe is to capture multiple exposures of the same image within a single video frame. We did this by modifying a five point Multi-Image Filter (a faceted lens commonly used to create photographic kaleidoscope effects), and applying successively increasing values of neutral density gel to four of the five facets of the filter (3?, 6?, 10 and 13? stops). This modified filter effectively produces a single image that is divided into five identical regions, with the center region capturing a "direct" view and the four outer regions stopped down to their respective exposure values. This modified filter is placed on a video camera that is mounted along with a mirrored ball on a span of angle iron (see Figure 1).

Assuming the relation between the camera and the ball never changes, the light probe only needs to be geometrically calibrated once. To compensate for the angle shift introduced by parallax effects from the facets of the multi-image filter, one can compute the arctangent of the distance between facets divided by the distance between the lens and the silver ball. By determining the number of degrees each facet is offset from the center, we are able to warp each region of the filter according to the direction space of its view of the ball. In our case, each facet's view of the ball was computed to be 2.7 degrees off from center.

$$\arctan \frac{\text{distance between facets}}{\text{distance from lens to ball}} \approx \text{angular offset between facets}$$

More accurate calibration can be done with the help of a light stage [3], which provides a "master key" for factoring out lens distortion and imperfections in the mirrored ball. In practice, it can be sufficiently accurate to simply

Jamie Waese and Paul Debevec
 USC Institute for Creative Technologies
 13274 Fiji Way, 5th Floor
 Marina del Rey, CA 90292
waese@ict.usc.edu, paul@debevec.org

measure the image translation between the five images and overlap them directly to assemble the high dynamic range image.

In order to capture high dynamic range light probe data at every frame along a path, one presses record and carries the light probe along the desired path. A computer program then imports each recorded frame and assembles the five images in each frame into a high dynamic range omnidirectional measurement of incident illumination.

Results

Figure 2 shows a raw image from the light probe camera:



Figure 1. A real time high dynamic range light probe.



Figure 2. Five exposures of a mirrored ball in a single image.

Figure 3 shows the five exposures of the ball derived from the light probe frame:



Figure 3. Five exposures of the mirrored ball captured in a single frame.

Figure 4 shows a CG object, illuminated by light captured by the real time high dynamic range light probe:



Figure 4. A CG model that is synthetically illuminated with light captured with the real time light probe.

Conclusion

This new technique will permit artists to composite CG objects into dynamic complex lighting environments, accurately reproducing high dynamic range lighting parameters for each frame. In the future, this technique would benefit from greater precision in applying the neutral density gels to the multi-image filter, a smaller camera rig, and higher resolution video cameras.

References

1. Paul Debevec. *Rendering Synthetic Objects Into Real Scenes: Bridging Traditional and Image-Based Graphics With Global Illumination and High Dynamic Range Photography*. In SIGGRAPH 98, August 1998.
2. Chris Tchou and Paul Debevec. *HDR Shop*. SIGGRAPH 2001 Technical Sketch.
3. Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duijker, Westley Sarokin and Mark Sagar. *Acquiring the Reflectance Field of a Human Face*. In SIGGRAPH 2000, August 2000.

Light Stage 2.0

SIGGRAPH 2001 Technical Sketch

Tim Hawkins, Jonathan Cohen, Chris Tchou, and Paul Debevec

At SIGGRAPH 2000, we presented an apparatus for capturing the appearance of a person's face under all possible directions of illumination. The captured data can be directly used to render the person into any imaginable lighting environment, and can also be used to build photoreal computer graphics models of the person that capture the unique texture and reflectance of their face. We have recently been developing the next generation of this lighting apparatus, which we call Light Stage 2.0.

Light Stage 2.0 is a much faster and more precise version of its predecessor. The original device allowed a single light to be spun around on a spherical path so that a subject could be illuminated from all directions, and regular video cameras were used to record the subject's appearance as the light moved. This system had two major problems. First, since the light was moved around by pulling on various ropes, it was hard to be sure what the precise location of the light was at any given time. Second, because the device could not be spun very fast, and because of the limit of 30 frames per second imposed by the video cameras, it took over a minute to do a data capture. Since the subject must remain still during the data capture, this meant we could only capture people in very passive expressions, and even then multiple trials were often needed.



Figure 1. Light Stage 2.0

With Light Stage 2.0, seen in Figure 1, we can capture all of the different lighting directions much more rapidly, with only a single rotation of a semicircular arm, and with greater accuracy. Thirty strobe lights are arrayed along the length of the arm, and flash in rapid sequence repeatedly as the arm rotates. High-speed

digital cameras capture the subject's appearance. This allows all directions of illumination to be provided in about four seconds, a period of time for which a person can easily remain still. It is also much easier to capture facial expressions that would be very difficult to maintain for an extended period of time (smiling, frowning, wincing, etc.).

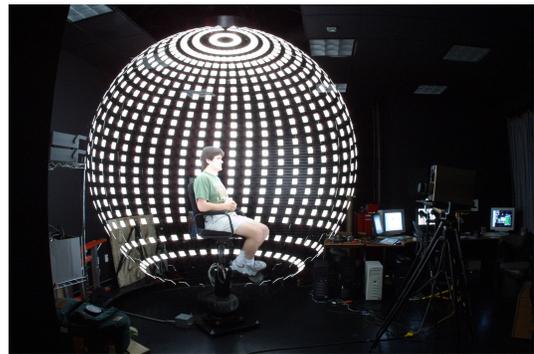


Figure 2. A long exposure (about 8 seconds) of Light Stage 2.0 in action.

We are currently working on integrating geometry capture to provide a truly complete model of the subject. For this, we use digital LCD projectors to project different structured patterns onto the subject, recording the appearance of the subject under each of the patterns quickly using our high-speed cameras. From this structured-light data, the geometry of the subject is easily recovered. This data together with the reflectance data may provide more complete and photoreal models of faces than ever before.

In the next few months, we will be researching new ways of analyzing the large amount of information captured in a Light Stage 2.0 scan and adapting the datasets for use in facial animation. We would also like to make our capture process even faster, with the goal of being able to capture both geometry and reflectance information in about five seconds. Our future plans include new prototype lighting devices that will allow similar datasets to be captured many times a second. This will allow an actor's performance to be recorded and then rendered photorealistically into virtual environments with arbitrary lighting and viewed from arbitrary angles.



Recovering High Dynamic Range Radiance Maps from Photographs

Paul E. Debevec

Jitendra Malik

University of California at Berkeley¹

ABSTRACT

We present a method of recovering high dynamic range radiance maps from photographs taken with conventional imaging equipment. In our method, multiple photographs of the scene are taken with different amounts of exposure. Our algorithm uses these differently exposed photographs to recover the response function of the imaging process, up to factor of scale, using the assumption of reciprocity. With the known response function, the algorithm can fuse the multiple photographs into a single, high dynamic range radiance map whose pixel values are proportional to the true radiance values in the scene. We demonstrate our method on images acquired with both photochemical and digital imaging processes. We discuss how this work is applicable in many areas of computer graphics involving digitized photographs, including image-based modeling, image compositing, and image processing. Lastly, we demonstrate a few applications of having high dynamic range radiance maps, such as synthesizing realistic motion blur and simulating the response of the human visual system.

CR Descriptors: I.2.10 [Artificial Intelligence]: Vision and Scene Understanding - *Intensity, color, photometry and thresholding*; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - *Color, shading, shadowing, and texture*; I.4.1 [Image Processing]: Digitization - *Scanning*; I.4.8 [Image Processing]: Scene Analysis - *Photometry, Sensor Fusion*.

1 Introduction

Digitized photographs are becoming increasingly important in computer graphics. More than ever, scanned images are used as texture maps for geometric models, and recent work in image-based modeling and rendering uses images as the fundamental modeling primitive. Furthermore, many of today's graphics applications require computer-generated images to mesh seamlessly with real photographic imagery. Properly using photographically acquired imagery in these applications can greatly benefit from an accurate model of the photographic process.

When we photograph a scene, either with film or an electronic imaging array, and digitize the photograph to obtain a two-dimensional array of "brightness" values, these values are rarely

true measurements of relative radiance in the scene. For example, if one pixel has twice the value of another, it is unlikely that it observed twice the radiance. Instead, there is usually an unknown, nonlinear mapping that determines how radiance in the scene becomes pixel values in the image.

This nonlinear mapping is hard to know beforehand because it is actually the composition of several nonlinear mappings that occur in the photographic process. In a conventional camera (see Fig. 1), the film is first exposed to light to form a latent image. The film is then developed to change this latent image into variations in transparency, or *density*, on the film. The film can then be digitized using a film scanner, which projects light through the film onto an electronic light-sensitive array, converting the image to electrical voltages. These voltages are digitized, and then manipulated before finally being written to the storage medium. If prints of the film are scanned rather than the film itself, then the printing process can also introduce nonlinear mappings.

In the first stage of the process, the film response to variations in exposure X (which is $E\Delta t$, the product of the irradiance E the film receives and the exposure time Δt) is a non-linear function, called the "characteristic curve" of the film. Noteworthy in the typical characteristic curve is the presence of a small response with no exposure and saturation at high exposures. The development, scanning and digitization processes usually introduce their own nonlinearities which compose to give the aggregate nonlinear relationship between the image pixel exposures X and their values Z .

Digital cameras, which use charge coupled device (CCD) arrays to image the scene, are prone to the same difficulties. Although the charge collected by a CCD element is proportional to its irradiance, most digital cameras apply a nonlinear mapping to the CCD outputs before they are written to the storage medium. This nonlinear mapping is used in various ways to mimic the response characteristics of film, anticipate nonlinear responses in the display device, and often to convert 12-bit output from the CCD's analog-to-digital converters to 8-bit values commonly used to store images. As with film, the most significant nonlinearity in the response curve is at its saturation point, where any pixel with a radiance above a certain level is mapped to the same maximum image value.

Why is this any problem at all? The most obvious difficulty, as any amateur or professional photographer knows, is that of limited dynamic range—one has to choose the range of radiance values that are of interest and determine the exposure time suitably. Sunlit scenes, and scenes with shiny materials and artificial light sources, often have extreme differences in radiance values that are impossible to capture without either under-exposing or saturating the film. To cover the full dynamic range in such a scene, one can take a series of photographs with different exposures. This then poses a problem: how can we combine these separate images into a composite radiance map? Here the fact that the mapping from scene radiance to pixel values is unknown and nonlinear begins to haunt us. The purpose of this paper is to present a simple technique for recovering this response function, up to a scale factor, using nothing more than a set of photographs taken with varying, known exposure durations. With this mapping, we then use the pixel values from all available photographs to construct an accurate map of the radiance in the scene, up to a factor of scale. This radiance map will cover

¹Computer Science Division, University of California at Berkeley, Berkeley, CA 94720-1776. Email: debevec@cs.berkeley.edu, malik@cs.berkeley.edu. More information and additional results may be found at: <http://www.cs.berkeley.edu/~debevec/Research>

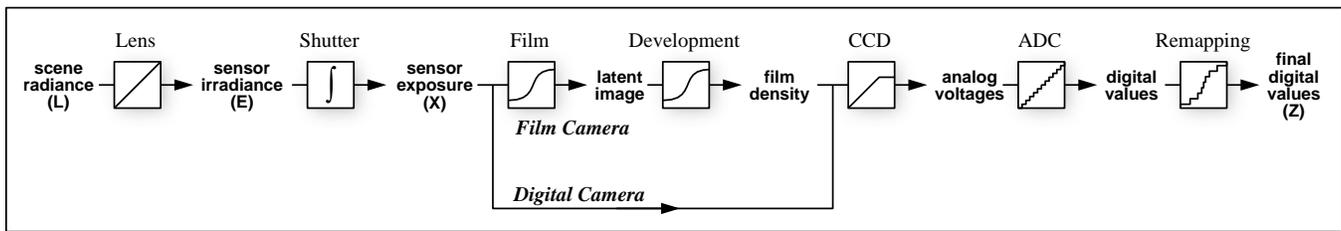


Figure 1: **Image Acquisition Pipeline** shows how scene radiance becomes pixel values for both film and digital cameras. Unknown nonlinear mappings can occur during exposure, development, scanning, digitization, and remapping. The algorithm in this paper determines the aggregate mapping from scene radiance L to pixel values Z from a set of differently exposed images.

the entire dynamic range captured by the original photographs.

1.1 Applications

Our technique of deriving imaging response functions and recovering high dynamic range radiance maps has many possible applications in computer graphics:

Image-based modeling and rendering

Image-based modeling and rendering systems to date (e.g. [11, 15, 2, 3, 12, 6, 17]) make the assumption that all the images are taken with the same exposure settings and film response functions. However, almost any large-scale environment will have some areas that are much brighter than others, making it impossible to adequately photograph the scene using a single exposure setting. In indoor scenes with windows, this situation often arises within the field of view of a single photograph, since the areas visible through the windows can be far brighter than the areas inside the building.

By determining the response functions of the imaging device, the method presented here allows one to correctly fuse pixel data from photographs taken at different exposure settings. As a result, one can properly photograph outdoor areas with short exposures, and indoor areas with longer exposures, without creating inconsistencies in the data set. Furthermore, knowing the response functions can be helpful in merging photographs taken with different imaging systems, such as video cameras, digital cameras, and film cameras with various film stocks and digitization processes.

The area of image-based modeling and rendering is working toward recovering more advanced reflection models (up to complete BRDF's) of the surfaces in the scene (e.g. [21]). These methods, which involve observing surface radiance in various directions under various lighting conditions, require absolute radiance values rather than the nonlinearly mapped pixel values found in conventional images. Just as important, the recovery of high dynamic range images will allow these methods to obtain accurate radiance values from surface specularities and from incident light sources. Such higher radiance values usually become clamped in conventional images.

Image processing

Most image processing operations, such as blurring, edge detection, color correction, and image correspondence, expect pixel values to be proportional to the scene radiance. Because of nonlinear image response, especially at the point of saturation, these operations can produce incorrect results for conventional images.

In computer graphics, one common image processing operation is the application of synthetic motion blur to images. In our results (Section 3), we will show that using true radiance maps produces significantly more realistic motion blur effects for high dynamic range scenes.

Image compositing

Many applications in computer graphics involve compositing image data from images obtained by different processes. For example, a background matte might be shot with a still camera, live action might be shot with a different film stock or scanning process, and CG elements would be produced by rendering algorithms. When there are significant differences in the response curves of these imaging processes, the composite image can be visually unconvincing. The technique presented in this paper provides a convenient and robust method of determining the overall response curve of any imaging process, allowing images from different processes to be used consistently as radiance maps. Furthermore, the recovered response curves can be inverted to render the composite radiance map as if it had been photographed with any of the original imaging processes, or a different imaging process entirely.

A research tool

One goal of computer graphics is to simulate the image formation process in a way that produces results that are consistent with what happens in the real world. Recovering radiance maps of real-world scenes should allow more quantitative evaluations of rendering algorithms to be made in addition to the qualitative scrutiny they traditionally receive. In particular, the method should be useful for developing reflectance and illumination models, and comparing global illumination solutions against ground truth data.

Rendering high dynamic range scenes on conventional display devices is the subject of considerable previous work, including [20, 16, 5, 23]. The work presented in this paper will allow such methods to be tested on real radiance maps in addition to synthetically computed radiance solutions.

1.2 Background

The photochemical processes involved in silver halide photography have been the subject of continued innovation and research ever since the invention of the daguerretype in 1839. [18] and [8] provide a comprehensive treatment of the theory and mechanisms involved. For the newer technology of solid-state imaging with charge coupled devices, [19] is an excellent reference. The technical and artistic problem of representing the dynamic range of a natural scene on the limited range of film has concerned photographers from the early days – [1] presents one of the best known systems to choose shutter speeds, lens apertures, and developing conditions to best coerce the dynamic range of a scene to fit into what is possible on a print. In scientific applications of photography, such as in astronomy, the nonlinear film response has been addressed by suitable calibration procedures. It is our objective instead to develop a simple self-calibrating procedure not requiring calibration charts or photometric measuring devices.

In previous work, [13] used multiple flux integration times of a CCD array to acquire extended dynamic range images. Since direct CCD outputs were available, the work did not need to deal with the

problem of nonlinear pixel value response. [14] addressed the problem of nonlinear response but provide a rather limited method of recovering the response curve. Specifically, a parametric form of the response curve is arbitrarily assumed, there is no satisfactory treatment of image noise, and the recovery process makes only partial use of the available data.

2 The Algorithm

This section presents our algorithm for recovering the film response function, and then presents our method of reconstructing the high dynamic range radiance image from the multiple photographs. We describe the algorithm assuming a grayscale imaging device. We discuss how to deal with color in Section 2.6.

2.1 Film Response Recovery

Our algorithm is based on exploiting a physical property of imaging systems, both photochemical and electronic, known as *reciprocity*.

Let us consider photographic film first. The response of a film to variations in exposure is summarized by the characteristic curve (or Hurter-Driffield curve). This is a graph of the optical density D of the processed film against the logarithm of the exposure X to which it has been subjected. The exposure X is defined as the product of the irradiance E at the film and exposure time, Δt , so that its units are Jm^{-2} . Key to the very concept of the characteristic curve is the assumption that only the product $E\Delta t$ is important, that halving E and doubling Δt will not change the resulting optical density D . Under extreme conditions (very large or very low Δt), the reciprocity assumption can break down, a situation described as reciprocity failure. In typical print films, reciprocity holds to within $\frac{1}{3}$ stop¹ for exposure times of 10 seconds to 1/10,000 of a second.² In the case of charge coupled arrays, reciprocity holds under the assumption that each site measures the total number of photons it absorbs during the integration time.

After the development, scanning and digitization processes, we obtain a digital number Z , which is a nonlinear function of the original exposure X at the pixel. Let us call this function f , which is the composition of the characteristic curve of the film as well as all the nonlinearities introduced by the later processing steps. Our first goal will be to recover this function f . Once we have that, we can compute the exposure X at each pixel, as $X = f^{-1}(Z)$. We make the reasonable assumption that the function f is monotonically increasing, so its inverse f^{-1} is well defined. Knowing the exposure X and the exposure time Δt , the irradiance E is recovered as $E = X/\Delta t$, which we will take to be proportional to the radiance L in the scene.³

Before proceeding further, we should discuss the consequences of the spectral response of the sensor. The exposure X should be thought of as a function of wavelength $X(\lambda)$, and the abscissa on the characteristic curve should be the integral $\int X(\lambda)R(\lambda)d\lambda$ where $R(\lambda)$ is the spectral response of the sensing element at the pixel location. Strictly speaking, our use of irradiance, a radiometric quantity, is not justified. However, the spectral response of the sensor site may not be the photopic luminosity function V_λ , so the photometric term *illuminance* is not justified either. In what follows, we will use the term irradiance, while urging the reader to remember that the

quantities we will be dealing with are weighted by the spectral response at the sensor site. For color photography, the color channels may be treated separately.

The input to our algorithm is a number of digitized photographs taken from the same vantage point with different known exposure durations Δt_j .⁴ We will assume that the scene is static and that this process is completed quickly enough that lighting changes can be safely ignored. It can then be assumed that the film irradiance values E_i for each pixel i are constant. We will denote pixel values by Z_{ij} where i is a spatial index over pixels and j indexes over exposure times Δt_j . We may now write down the film reciprocity equation as:

$$Z_{ij} = f(E_i \Delta t_j) \quad (1)$$

Since we assume f is monotonic, it is invertible, and we can rewrite (1) as:

$$f^{-1}(Z_{ij}) = E_i \Delta t_j$$

Taking the natural logarithm of both sides, we have:

$$\ln f^{-1}(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

To simplify notation, let us define function $g = \ln f^{-1}$. We then have the set of equations:

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j \quad (2)$$

where i ranges over pixels and j ranges over exposure durations. In this set of equations, the Z_{ij} are known, as are the Δt_j . The unknowns are the irradiances E_i , as well as the function g , although we assume that g is smooth and monotonic.

We wish to recover the function g and the irradiances E_i that best satisfy the set of equations arising from Equation 2 in a least-squared error sense. We note that recovering g only requires recovering the *finite* number of values that $g(z)$ can take since the domain of Z , pixel brightness values, is finite. Letting Z_{min} and Z_{max} be the least and greatest pixel values (integers), N be the number of pixel locations and P be the number of photographs, we formulate the problem as one of finding the $(Z_{max} - Z_{min} + 1)$ values of $g(Z)$ and the N values of $\ln E_i$ that minimize the following quadratic objective function:

$$\mathcal{O} = \sum_{i=1}^N \sum_{j=1}^P [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} g''(z)^2 \quad (3)$$

The first term ensures that the solution satisfies the set of equations arising from Equation 2 in a least squares sense. The second term is a smoothness term on the sum of squared values of the second derivative of g to ensure that the function g is smooth; in this discrete setting we use $g''(z) = g(z-1) - 2g(z) + g(z+1)$. This smoothness term is essential to the formulation in that it provides coupling between the values $g(z)$ in the minimization. The scalar λ weights the smoothness term relative to the data fitting term, and should be chosen appropriately for the amount of noise expected in the Z_{ij} measurements.

Because it is quadratic in the E_i 's and $g(z)$'s, minimizing \mathcal{O} is a straightforward linear least squares problem. The overdetermined

¹ 1 stop is a photographic term for a factor of two; $\frac{1}{3}$ stop is thus $2^{\frac{1}{3}}$

² An even larger dynamic range can be covered by using neutral density filters to lessen to amount of light reaching the film for a given exposure time. A discussion of the modes of reciprocity failure may be found in [18], ch. 4.

³ L is proportional E for any particular pixel, but it is possible for the proportionality factor to be different at different places on the sensor. One formula for this variance, given in [7], is $E = L \frac{\pi}{4} \left(\frac{d}{f}\right)^2 \cos^4 \alpha$, where α measures the pixel's angle from the lens' optical axis. However, most modern camera lenses are designed to compensate for this effect, and provide a nearly constant mapping between radiance and irradiance at f/8 and smaller apertures. See also [10].

⁴ Most modern SLR cameras have electronically controlled shutters which give extremely accurate and reproducible exposure times. We tested our Canon EOS Elan camera by using a Macintosh to make digital audio recordings of the shutter. By analyzing these recordings we were able to verify the accuracy of the exposure times to within a thousandth of a second. Conveniently, we determined that the actual exposure times varied by powers of two between stops ($\frac{1}{64}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8, 16, 32$), rather than the rounded numbers displayed on the camera readout ($\frac{1}{60}, \frac{1}{30}, \frac{1}{15}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8, 15, 30$). Because of problems associated with vignetting, varying the aperture is not recommended.

system of linear equations is robustly solved using the singular value decomposition (SVD) method. An intuitive explanation of the procedure may be found in Fig. 2.

We need to make three additional points to complete our description of the algorithm:

First, the solution for the $g(z)$ and E_i values can only be up to a single scale factor α . If each log irradiance value $\ln E_i$ were replaced by $\ln E_i + \alpha$, and the function g replaced by $g + \alpha$, the system of equations 2 and also the objective function \mathcal{O} would remain unchanged. To establish a scale factor, we introduce the additional constraint $g(Z_{mid}) = 0$, where $Z_{mid} = \frac{1}{2}(Z_{min} + Z_{max})$, simply by adding this as an equation in the linear system. The meaning of this constraint is that a pixel with value midway between Z_{min} and Z_{max} will be assumed to have unit exposure.

Second, the solution can be made to have a much better fit by anticipating the basic shape of the response function. Since $g(z)$ will typically have a steep slope near Z_{min} and Z_{max} , we should expect that $g(z)$ will be less smooth and will fit the data more poorly near these extremes. To recognize this, we can introduce a weighting function $w(z)$ to emphasize the smoothness and fitting terms toward the middle of the curve. A sensible choice of w is a simple hat function:

$$w(z) = \begin{cases} z - Z_{min} & \text{for } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases} \quad (4)$$

Equation 3 now becomes:

$$\mathcal{O} = \sum_{i=1}^N \sum_{j=1}^P \{w(Z_{ij}) [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z)g''(z)]^2$$

Finally, we need not use every available pixel site in this solution procedure. Given measurements of N pixels in P photographs, we have to solve for N values of $\ln E_i$ and $(Z_{max} - Z_{min})$ samples of g . To ensure a sufficiently overdetermined system, we want $N(P - 1) > (Z_{max} - Z_{min})$. For the pixel value range $(Z_{max} - Z_{min}) = 255$, $P = 11$ photographs, a choice of N on the order of 50 pixels is more than adequate. Since the size of the system of linear equations arising from Equation 3 is on the order of $N \times P + Z_{max} - Z_{min}$, computational complexity considerations make it impractical to use every pixel location in this algorithm. Clearly, the pixel locations should be chosen so that they have a reasonably even distribution of pixel values from Z_{min} to Z_{max} , and so that they are spatially well distributed in the image. Furthermore, the pixels are best sampled from regions of the image with low intensity variance so that radiance can be assumed to be constant across the area of the pixel, and the effect of optical blur of the imaging system is minimized. So far we have performed this task by hand, though it could easily be automated.

Note that we have not explicitly enforced the constraint that g must be a monotonic function. If desired, this can be done by transforming the problem to a non-negative least squares problem. We have not found it necessary because, in our experience, the smoothness penalty term is enough to make the estimated g monotonic in addition to being smooth.

To show its simplicity, the MATLAB routine we used to minimize Equation 5 is included in the Appendix. Running times are on the order of a few seconds.

2.2 Constructing the High Dynamic Range Radiance Map

Once the response curve g is recovered, it can be used to quickly convert pixel values to relative radiance values, assuming the exposure Δt_j is known. Note that the curve can be used to determine radiance values in any image(s) acquired by the imaging process associated with g , not just the images used to recover the response function.

From Equation 2, we obtain:

$$\ln E_i = g(Z_{ij}) - \ln \Delta t_j \quad (5)$$

For robustness, and to recover high dynamic range radiance values, we should use all the available exposures for a particular pixel to compute its radiance. For this, we reuse the weighting function in Equation 4 to give higher weight to exposures in which the pixel's value is closer to the middle of the response function:

$$\ln E_i = \frac{\sum_{j=1}^P w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^P w(Z_{ij})} \quad (6)$$

Combining the multiple exposures has the effect of reducing noise in the recovered radiance values. It also reduces the effects of imaging artifacts such as film grain. Since the weighting function ignores saturated pixel values, "blooming" artifacts⁵ have little impact on the reconstructed radiance values.

2.2.1 Storage

In our implementation the recovered radiance map is computed as an array of single-precision floating point values. For efficiency, the map can be converted to the image format used in the RADIANCE [22] simulation and rendering system, which uses just eight bits for each of the mantissa and exponent. This format is particularly compact for color radiance maps, since it stores just one exponent value for all three color values at each pixel. Thus, in this format, a high dynamic range radiance map requires just one third more storage than a conventional RGB image.

2.3 How many images are necessary?

To decide on the number of images needed for the technique, it is convenient to consider the two aspects of the process:

1. *Recovering the film response curve:* This requires a minimum of two photographs. Whether two photographs are enough can be understood in terms of the heuristic explanation of the process of film response curve recovery shown in Fig. 2. If the scene has sufficiently many different radiance values, the entire curve can, in principle, be assembled by sliding together the sampled curve segments, each with only two samples. Note that the photos must be similar enough in their exposure amounts that some pixels fall into the working range⁶ of the film in both images; otherwise, there is no information to relate the exposures to each other. Obviously, using more than two images with differing exposure times improves performance with respect to noise sensitivity.
2. *Recovering a radiance map given the film response curve:* The number of photographs needed here is a function of the dynamic range of radiance values in the scene. Suppose the range of maximum to minimum radiance values that we are

⁵Blooming occurs when charge or light at highly saturated sites on the imaging surface spills over and affects values at neighboring sites.

⁶The *working range* of the film corresponds to the middle section of the response curve. The ends of the curve, in which large changes in exposure cause only small changes in density (or pixel value), are called the *toe* and the *shoulder*.

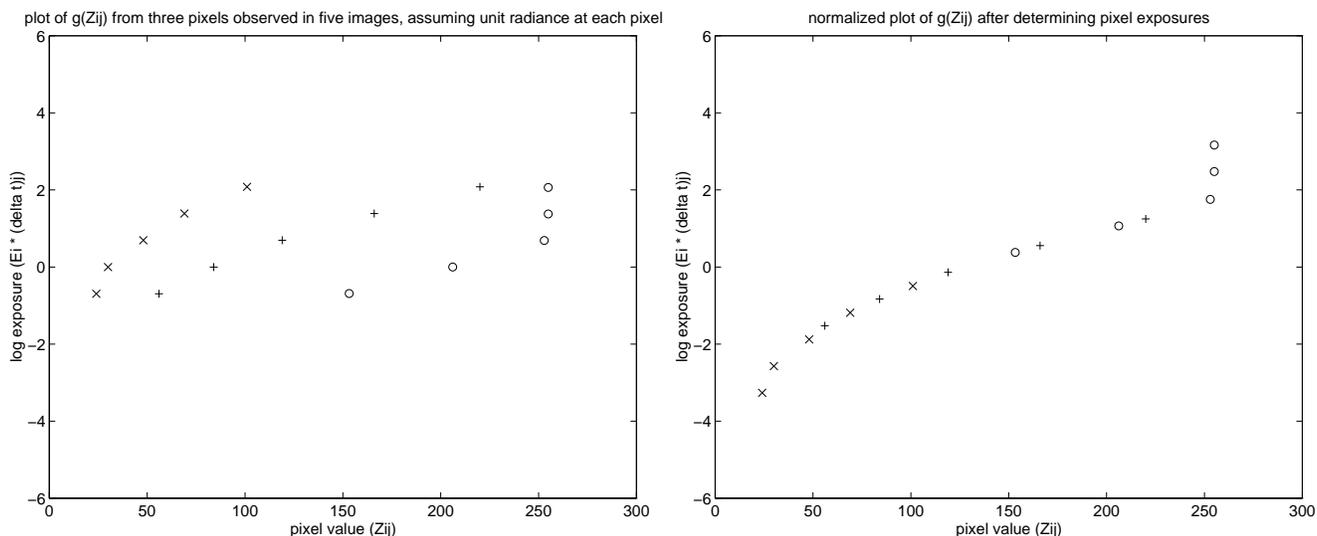


Figure 2: In the figure on the left, the \times symbols represent samples of the g curve derived from the digital values at one pixel for 5 different known exposures using Equation 2. The unknown log irradiance $\ln E_i$ has been arbitrarily assumed to be 0. Note that the shape of the g curve is correct, though its position on the vertical scale is arbitrary corresponding to the unknown $\ln E_i$. The $+$ and o symbols show samples of g curve segments derived by consideration of two other pixels; again the vertical position of each segment is arbitrary. Essentially, what we want to achieve in the optimization process is to slide the 3 sampled curve segments up and down (by adjusting their $\ln E_i$'s) until they "line up" into a single smooth, monotonic curve, as shown in the right figure. The vertical position of the composite curve will remain arbitrary.

interested in recovering accurately is R , and the film is capable of representing in its working range a dynamic range of F . Then the minimum number of photographs needed is $\lceil \frac{R}{F} \rceil$ to ensure that every part of the scene is imaged in at least one photograph at an exposure duration that puts it in the working range of the film response curve. As in recovering the response curve, using more photographs than strictly necessary will result in better noise sensitivity.

If one wanted to use as few photographs as possible, one might first recover the response curve of the imaging process by photographing a scene containing a diverse range of radiance values at three or four different exposures, differing by perhaps one or two stops. This response curve could be used to determine the working range of the imaging process, which for the processes we have seen would be as many as five or six stops. For the remainder of the shoot, the photographer could decide for any particular scene the number of shots necessary to cover its entire dynamic range. For diffuse indoor scenes, only one exposure might be necessary; for scenes with high dynamic range, several would be necessary. By recording the exposure amount for each shot, the images could then be converted to radiance maps using the pre-computed response curve.

2.4 Recovering extended dynamic range from single exposures

Most commercially available film scanners can detect reasonably close to the full range of useful densities present in film. However, many of these scanners (as well as the Kodak PhotoCD process) produce 8-bit-per-channel images designed to be viewed on a screen or printed on paper. Print film, however, records a significantly greater dynamic range than can be displayed with either of these media. As a result, such scanners deliver only a portion of the detected dynamic range of print film in a single scan, discarding information in either high or low density regions. The portion of the detected dynamic range that is delivered can usually be influenced by "brightness" or "density adjustment" controls.

The method presented in this paper enables two methods for recovering the full dynamic range of print film which we will briefly

outline⁷. In the first method, the print negative is scanned with the scanner set to scan slide film. Most scanners will then record the entire detectable dynamic range of the film in the resulting image. As before, a series of differently exposed images of the same scene can be used to recover the response function of the imaging system with each of these scanner settings. This response function can then be used to convert individual exposures to radiance maps. Unfortunately, since the resulting image is still 8-bits-per-channel, this results in increased quantization.

In the second method, the film can be scanned twice with the scanner set to different density adjustment settings. A series of differently exposed images of the same scene can then be used to recover the response function of the imaging system at each of these density adjustment settings. These two response functions can then be used to combine two scans of any single negative using a similar technique as in Section 2.2.

2.5 Obtaining Absolute Radiance

For many applications, such as image processing and image compositing, the relative radiance values computed by our method are all that are necessary. If needed, an approximation to the scaling term necessary to convert to absolute radiance can be derived using the ASA of the film⁸ and the shutter speeds and exposure amounts in the photographs. With these numbers, formulas that give an approximate prediction of film response can be found in [9]. Such an approximation can be adequate for simulating visual artifacts such as glare, and predicting areas of scotopic retinal response. If desired, one could recover the scaling factor precisely by photographing a calibration luminaire of known radiance, and scaling the radiance values to agree with the known radiance of the luminaire.

2.6 Color

Color images, consisting of red, green, and blue channels, can be processed by reconstructing the imaging system response curve for

⁷This work was done in collaboration with Gregory Ward Larson

⁸Conveniently, most digital cameras also specify their sensitivity in terms of ASA.

each channel independently. Unfortunately, there will be three unknown scaling factors relating relative radiance to absolute radiance, one for each channel. As a result, different choices of these scaling factors will change the color balance of the radiance map.

By default, the algorithm chooses the scaling factor such that a pixel with value Z_{mid} will have unit exposure. Thus, any pixel with the RGB value $(Z_{mid}, Z_{mid}, Z_{mid})$ will have equal radiance values for R, G, and B, meaning that the pixel is achromatic. If the three channels of the imaging system actually do respond equally to achromatic light in the neighborhood of Z_{mid} , then our procedure correctly reconstructs the relative radiances.

However, films are usually calibrated to respond achromatically to a particular color of light C , such as sunlight or fluorescent light. In this case, the radiance values of the three channels should be scaled so that the pixel value $(Z_{mid}, Z_{mid}, Z_{mid})$ maps to a radiance with the same color ratios as C . To properly model the color response of the entire imaging process rather than just the film response, the scaling terms can be adjusted by photographing a calibration luminaire of known color.

2.7 Taking virtual photographs

The recovered response functions can also be used to map radiance values back to pixel values for a given exposure Δt using Equation 1. This process can be thought of as taking a virtual photograph of the radiance map, in that the resulting image will exhibit the response qualities of the modeled imaging system. Note that the response functions used need not be the same response functions used to construct the original radiance map, which allows photographs acquired with one imaging process to be rendered as if they were acquired with another.⁹

3 Results

Figures 3-5 show the results of using our algorithm to determine the response curve of a DCS460 digital camera. Eleven grayscale photographs filtered down to 765×509 resolution (Fig. 3) were taken at $f/8$ with exposure times ranging from $\frac{1}{30}$ of a second to 30 seconds, with each image receiving twice the exposure of the previous one. The film curve recovered by our algorithm from 45 pixel locations observed across the image sequence is shown in Fig. 4. Note that although CCD image arrays naturally produce linear output, from the curve it is evident that the camera nonlinearly remaps the data, presumably to mimic the response curves found in film. The underlying registered $(E_i \Delta t_j, Z_{ij})$ data are shown as light circles underneath the curve; some outliers are due to sensor artifacts (light horizontal bands across some of the darker images.)

Fig. 5 shows the reconstructed high dynamic range radiance map. To display this map, we have taken the logarithm of the radiance values and mapped the range of these values into the range of the display. In this representation, the pixels at the light regions do not saturate, and detail in the shadow regions can be made out, indicating that all of the information from the original image sequence is present in the radiance map. The large range of values present in the radiance map (over four orders of magnitude of useful dynamic range) is shown by the values at the marked pixel locations.

Figure 6 shows sixteen photographs taken inside a church with a Canon 35mm SLR camera on Fuji 100 ASA color print film. A fish-eye 15mm lens set at $f/8$ was used, with exposure times ranging from 30 seconds to $\frac{1}{1000}$ of a second in 1-stop increments. The film was developed professionally and scanned in using a Kodak PhotoCD film scanner. The scanner was set so that it would not individually



Figure 3: (a) Eleven grayscale photographs of an indoor scene acquired with a Kodak DCS460 digital camera, with shutter speeds progressing in 1-stop increments from $\frac{1}{30}$ of a second to 30 seconds.

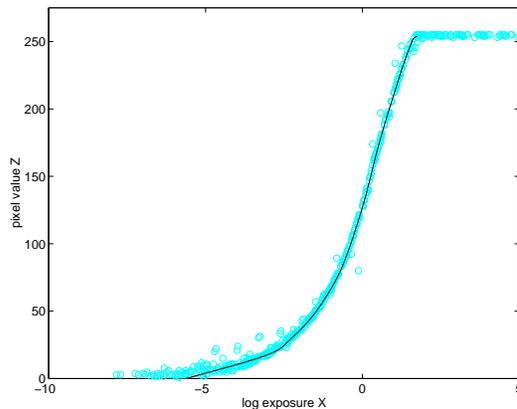


Figure 4: The response function of the DCS460 recovered by our algorithm, with the underlying $(E_i \Delta t_j, Z_{ij})$ data shown as light circles. The logarithm is base e .



Figure 5: The reconstructed high dynamic range radiance map, mapped into a grayscale image by taking the logarithm of the radiance values. The relative radiance values of the marked pixel locations, clockwise from lower left: 1.0, 46.2, 1907.1, 15116.0, and 18.0.

⁹Note that here we are assuming that the spectral response functions for each channel of the two imaging processes is the same. Also, this technique does not model many significant qualities of an imaging system such as film grain, chromatic aberration, blooming, and the modulation transfer function.

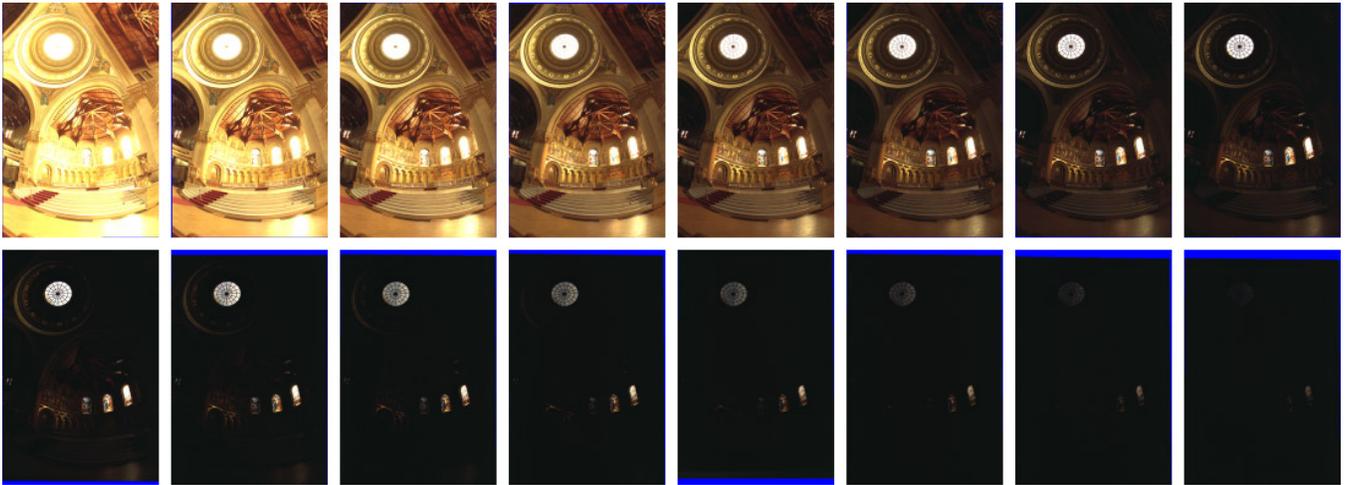


Figure 6: Sixteen photographs of a church taken at 1-stop increments from 30 sec to $\frac{1}{1000}$ sec. The sun is directly behind the rightmost stained glass window, making it especially bright. The blue borders seen in some of the image margins are induced by the image registration process.

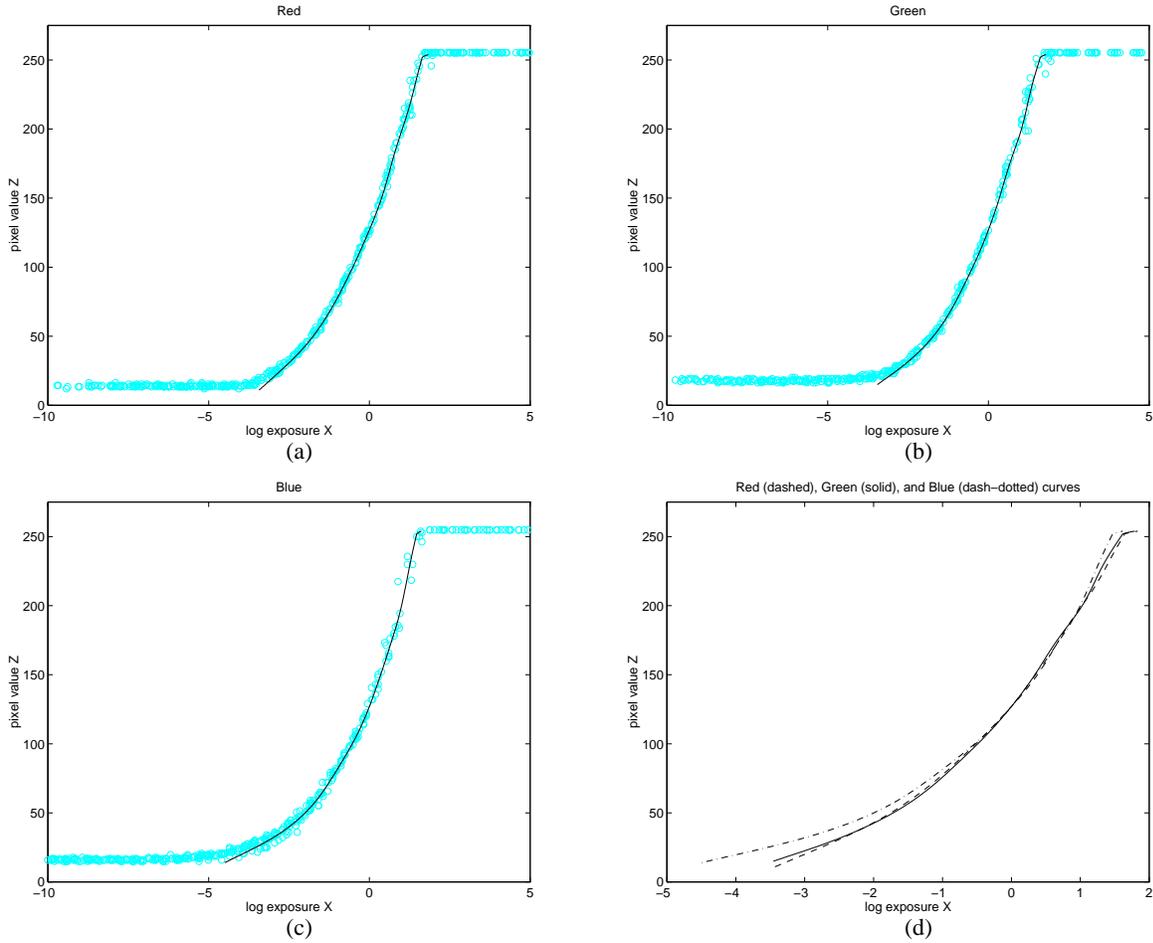


Figure 7: Recovered response curves for the imaging system used in the church photographs in Fig. 8. (a-c) Response functions for the red, green, and blue channels, plotted with the underlying $(E_i \Delta t_j, Z_{ij})$ data shown as light circles. (d) The response functions for red, green, and blue plotted on the same axes. Note that while the red and green curves are very consistent, the blue curve rises significantly above the others for low exposure values. This indicates that dark regions in the images exhibit a slight blue cast. Since this artifact is recovered by the response curves, it does not affect the relative radiance values.

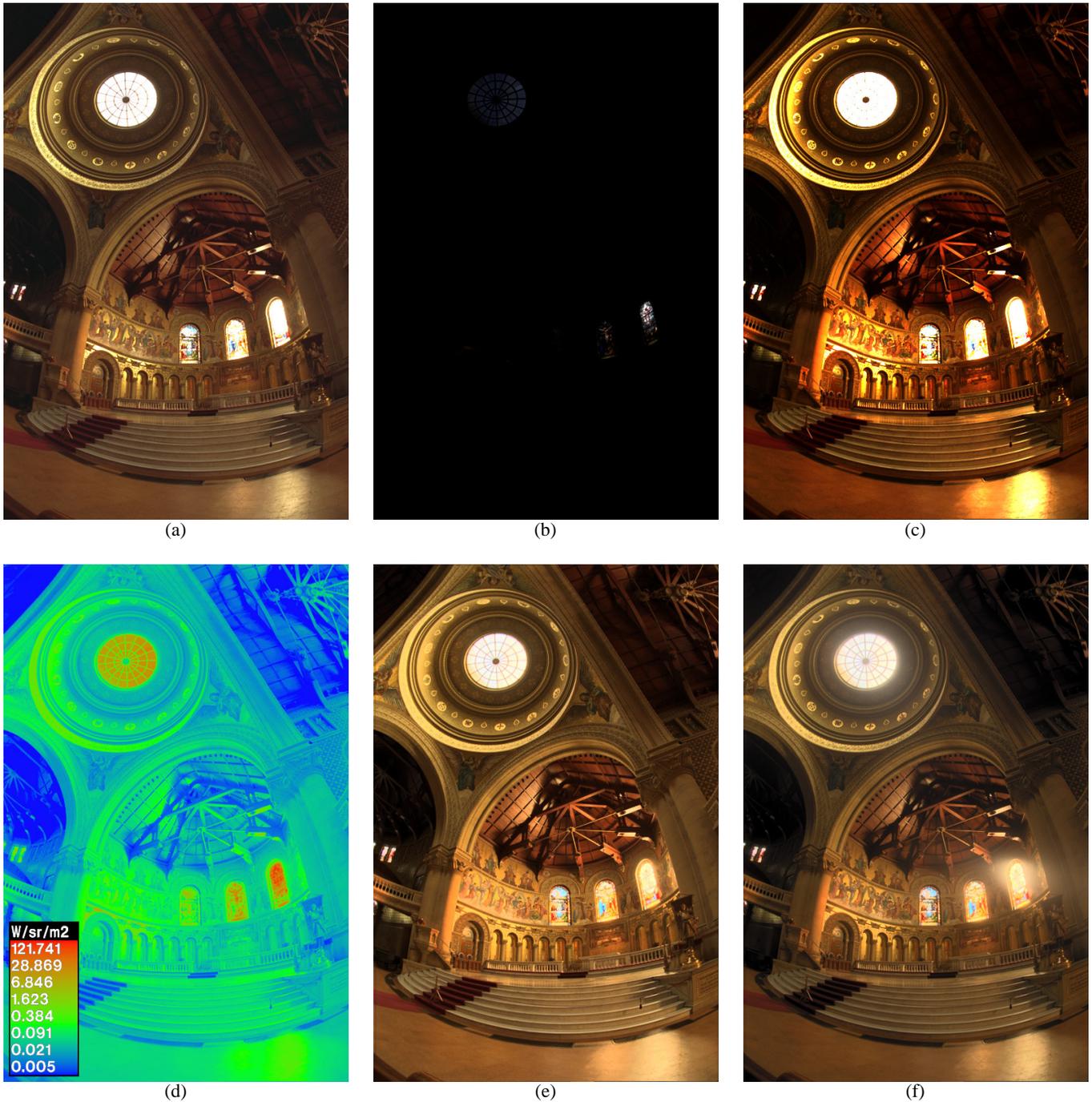


Figure 8: **(a)** An actual photograph, taken with conventional print film at two seconds and scanned to PhotoCD. **(b)** The high dynamic range radiance map, displayed by linearly mapping its entire dynamic range into the dynamic range of the display device. **(c)** The radiance map, displayed by linearly mapping the lower 0.1% of its dynamic range to the display device. **(d)** A false-color image showing relative radiance values for a grayscale version of the radiance map, indicating that the map contains over five orders of magnitude of useful dynamic range. **(e)** A rendering of the radiance map using adaptive histogram compression. **(f)** A rendering of the radiance map using histogram compression and also simulating various properties of the human visual system, such as glare, contrast sensitivity, and scotopic retinal response. Images (e) and (f) were generated by a method described in [23]. Images (d-f) courtesy of Gregory Ward Larson.

adjust the brightness and contrast of the images¹⁰ to guarantee that each image would be digitized using the same response function.

An unfortunate aspect of the PhotoCD process is that it does not scan precisely the same area of each negative relative to the extents of the image.¹¹ To counteract this effect, we geometrically registered the images to each other using a using normalized correlation (see [4]) to determine, with sub-pixel accuracy, corresponding pixels between pairs of images.

Fig. 7(a-c) shows the response functions for the red, green, and blue channels of the church sequence recovered from 28 pixel locations. Fig. 7(d) shows the recovered red, green, and blue response curves plotted on the same set of axes. From this plot, we can see that while the red and green curves are very consistent, the blue curve rises significantly above the others for low exposure values. This indicates that dark regions in the images exhibit a slight blue cast. Since this artifact is modeled by the response curves, it will not affect the relative radiance values.

Fig. 8 interprets the recovered high dynamic range radiance map in a variety of ways. Fig. 8(a) is one of the actual photographs, which lacks detail in its darker regions at the same time that many values within the two rightmost stained glass windows are saturated. Figs. 8(b,c) show the radiance map, linearly scaled to the display device using two different scaling factors. Although one scaling factor is one thousand times the other, there is useful detail in both images. Fig. 8(d) is a false-color image showing radiance values for a grayscale version of the radiance map; the highest listed radiance value is nearly 250,000 times that of the lowest. Figs. 8(e,f) show two renderings of the radiance map using a new tone reproduction algorithm [23]. Although the rightmost stained glass window has radiance values over a thousand times higher than the darker areas in the rafters, these renderings exhibit detail in both areas.

Figure 9 demonstrates two applications of the techniques presented in this paper: accurate signal processing and virtual photography. The task is to simulate the effects of motion blur caused by moving the camera during the exposure. Fig. 9(a) shows the results of convolving an actual, low-dynamic range photograph with a 37×1 pixel box filter to simulate horizontal motion blur. Fig. 9(b) shows the results of applying this same filter to the high dynamic range radiance map, and then sending this filtered radiance map back through the recovered film response functions using the same exposure time Δt as in the actual photograph. Because we are seeing this image through the actual image response curves, the two left images are tonally consistent with each other. However, there is a large difference between these two images near the bright spots. In the photograph, the bright radiance values have been clamped to the maximum pixel values by the response function. As a result, these clamped values blur with lower neighboring values and fail to saturate the image in the final result, giving a muddy appearance.

In Fig. 9(b), the extremely high pixel values were represented properly in the radiance map and thus remained at values above the level of the response function's saturation point within most of the blurred region. As a result, the resulting virtual photograph exhibits several crisply-defined saturated regions.

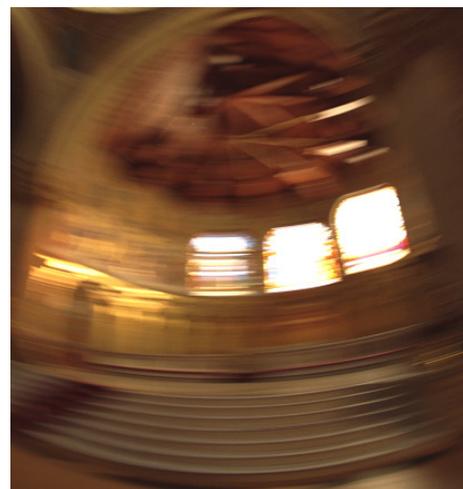
Fig. 9(c) is an actual photograph with real motion blur induced by spinning the camera on the tripod during the exposure, which is equal in duration to Fig. 9(a) and the exposure simulated in Fig. 9(b). Clearly, in the bright regions, the blurring effect is qualitatively similar to the synthetic blur in 9(b) but not 9(a). The precise shape of the real motion blur is curved and was not modeled for this demonstration.



(a) Synthetically blurred digital image



(b) Synthetically blurred radiance map



(c) Actual blurred photograph

Figure 9: (a) Synthetic motion blur applied to one of the original digitized photographs. The bright values in the windows are clamped before the processing, producing mostly unsaturated values in the blurred regions. (b) Synthetic motion blur applied to a recovered high-dynamic range radiance map, then virtually re-photographed through the recovered film response curves. The radiance values are clamped to the display device after the processing, allowing pixels to remain saturated in the window regions. (c) Real motion blur created by rotating the camera on the tripod during the exposure, which is much more consistent with (b) than (a).

¹⁰This feature of the PhotoCD process is called "Scene Balance Adjustment", or SBA.

¹¹This is far less of a problem for cinematic applications, in which the film sprocket holes are used to expose and scan precisely the same area of each frame.

4 Conclusion

We have presented a simple, practical, robust and accurate method of recovering high dynamic range radiance maps from ordinary photographs. Our method uses the constraint of sensor reciprocity to derive the response function and relative radiance values directly from a set of images taken with different exposures. This work has a wide variety of applications in the areas of image-based modeling and rendering, image processing, and image compositing, a few of which we have demonstrated. It is our hope that this work will be able to help both researchers and practitioners of computer graphics make much more effective use of digitized photographs.

Acknowledgments

The authors wish to thank Tim Hawkins, Carlo Séquin, David Forsyth, Steve Chenney, Chris Healey, and our reviewers for their valuable help in revising this paper. This research was supported by a Multidisciplinary University Research Initiative on three dimensional direct visualization from ONR and BMDO, grant FDN00014-96-1-1200.

References

- [1] ADAMS, A. *Basic Photo*, 1st ed. Morgan & Morgan, Hastings-on-Hudson, New York, 1970.
- [2] CHEN, E. QuickTime VR - an image-based approach to virtual environment navigation. In *SIGGRAPH '95* (1995).
- [3] DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH '96* (August 1996), pp. 11–20.
- [4] FAUGERAS, O. *Three-Dimensional Computer Vision*. MIT Press, 1993.
- [5] FERWERDA, J. A., PATTANAİK, S. N., SHIRLEY, P., AND GREENBERG, D. P. A model of visual adaptation for realistic image synthesis. In *SIGGRAPH '96* (1996), pp. 249–258.
- [6] GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. F. The Lumigraph. In *SIGGRAPH '96* (1996), pp. 43–54.
- [7] HORN, B. K. P. *Robot Vision*. MIT Press, Cambridge, Mass., 1986, ch. 10, pp. 206–208.
- [8] JAMES, T., Ed. *The Theory of the Photographic Process*. Macmillan, New York, 1977.
- [9] KAUFMAN, J. E., Ed. *IES Lighting Handbook; the standard lighting guide*, 7th ed. Illuminating Engineering Society, New York, 1987, p. 24.
- [10] KOLB, C., MITCHELL, D., AND HANRAHAN, P. A realistic camera model for computer graphics. In *SIGGRAPH '95* (1995).
- [11] LAVEAU, S., AND FAUGERAS, O. 3-D scene representation as a collection of images. In *Proceedings of 12th International Conference on Pattern Recognition* (1994), vol. 1, pp. 689–691.
- [12] LEVOY, M., AND HANRAHAN, P. Light field rendering. In *SIGGRAPH '96* (1996), pp. 31–42.
- [13] MADDEN, B. C. Extended intensity range imaging. Tech. rep., GRASP Laboratory, University of Pennsylvania, 1993.
- [14] MANN, S., AND PICARD, R. W. Being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures. In *Proceedings of IS&T 46th annual conference* (May 1995), pp. 422–428.
- [15] MCMILLAN, L., AND BISHOP, G. Plenoptic Modeling: An image-based rendering system. In *SIGGRAPH '95* (1995).

- [16] SCHLICK, C. Quantization techniques for visualization of high dynamic range pictures. In *Fifth Eurographics Workshop on Rendering (Darmstadt, Germany)* (June 1994), pp. 7–18.
- [17] SZELISKI, R. Image mosaicing for tele-reality applications. In *IEEE Computer Graphics and Applications* (1996).
- [18] TANI, T. *Photographic sensitivity: theory and mechanisms*. Oxford University Press, New York, 1995.
- [19] THEUWISSEN, A. J. P. *Solid-state imaging with charge-coupled devices*. Kluwer Academic Publishers, Dordrecht; Boston, 1995.
- [20] TUMBLIN, J., AND RUSHMEIER, H. Tone reproduction for realistic images. *IEEE Computer Graphics and Applications* 13, 6 (1993), 42–48.
- [21] WARD, G. J. Measuring and modeling anisotropic reflection. In *SIGGRAPH '92* (July 1992), pp. 265–272.
- [22] WARD, G. J. The radiance lighting simulation and rendering system. In *SIGGRAPH '94* (July 1994), pp. 459–472.
- [23] WARD, G. J., RUSHMEIER, H., AND PIATKO, C. A visibility matching tone reproduction operator for high dynamic range scenes. Tech. Rep. LBNL-39882, Lawrence Berkeley National Laboratory, March 1997.

A Matlab Code

Here is the MATLAB code used to solve the linear system that minimizes the objective function \mathcal{O} in Equation 3. Given a set of observed pixel values in a set of images with known exposures, this routine reconstructs the imaging response curve and the radiance values for the given pixels. The weighting function $w(z)$ is found in Equation 4.

```
% gsolve.m - Solve for imaging system response function
%
% Given a set of pixel values observed for several pixels in several
% images with different exposure times, this function returns the
% imaging system's response function g as well as the log film irradiance
% values for the observed pixels.
%
% Assumes:
% Zmin = 0
% Zmax = 255
%
% Arguments:
% Z(i,j) is the pixel values of pixel location number i in image j
% E(j) is the log delta t, or log shutter speed, for image j
% lambda is lambda, the constant that determines the amount of smoothness
% w(z) is the weighting function value for pixel value z
%
% Returns:
% g(z) is the log exposure corresponding to pixel value z
% IE(i) is the log film irradiance at pixel location i
%
function [g,IE]=gsolve(Z,B,lambda,w)

n = 256;

A = zeros(size(Z,1)*size(Z,2)+n+1,n+size(Z,1));
b = zeros(size(A,1),1);

%% Include the data-fitting equations

k = 1;
for i=1:size(Z,1)
    for j=1:size(Z,2)
        wij = w(Z(i,j)+1);
        A(k,Z(i,j)+1) = wij; A(k,n+1) = -wij; b(k,1) = wij * B(i,j);
        k=k+1;
    end
end

%% Fix the curve by setting its middle value to 0

A(k,129) = 1;
k=k+1;

%% Include the smoothness equations

for i=1:n-2
    A(k,i)=1*w(i+1); A(k,i+1)=-2*1*w(i+1); A(k,i+2)=1*w(i+1);
    k=k+1;
end

%% Solve the system using SVD

x = A\b;

g = x(1:n);
IE = x(n+1:size(x,1));
```

Overcoming Gamut and Dynamic Range

Limitations in Digital Images

*Gregory Ward Larson
Silicon Graphics, Inc.
Mountain View, California*

Abstract

The human eye can accommodate luminance in a single view over a range of about 10,000:1 and is capable of distinguishing about 10,000 colors at a given brightness. By comparison, typical CRT displays have a luminance range less than 100:1 and cover about half of the visible color gamut. Despite this difference, most digital image formats are geared to the capabilities of conventional displays, rather than the characteristics of human vision. In this paper, we propose two compact encodings suitable for the transfer, manipulation, and storage of full range color images. The first format is a replacement for conventional RGB images, and encodes color pixels as log luminance values and CIE (u',v') chromaticity coordinates. We have implemented and distributed this encoding as part of the standard TIFF I/O library on the net. The second format is proposed as an adjunct to conventional RGB data, and encodes out-of-gamut (and out-of-range) pixels in a supplemental image, suitable as a layer extension to the Flashpix standard. This data can then be recombined with the original RGB layer to obtain a high dynamic range image covering the full gamut of perceivable colors. Finally, we demonstrate the power and utility of full gamut imagery with example images and applications.

Introduction

What is the ultimate use of a digital image? How will it be presented? Will it be modified or adjusted? What kind of monitor will it be displayed on? What type of printer will it be sent to? How accurate do the colors need to be? More often than not, we don't know the answers to these questions a priori. More important, we don't know how these questions will be answered 10 or 100 years from now, when everything we know about digital imaging will have changed, but someone may still want to use our image. We should therefore endeavor to

record image data that will be valuable under a broad range of foreseeable and postulated circumstances. Although this seems problematic, there is a simple solution. We may not be able to predict the technology, but we can predict that people will still be the primary consumers.

Most commonly used image standards based on current display technology, i.e., CRT monitors, rather than something less apt to change, i.e., human vision. All RGB standards are limited to a fraction of the visible gamut, since this gamut cannot be contained between any three *real* colors. Even Kodak's PhotoYCC encoding is ultimately geared for CRT display, and doesn't encompass the full gamut of colors or cover more than two orders of magnitude in brightness. The human eye is capable of perceiving at least four orders of magnitude in a daylight scene, and adapting more gradually over seven *additional* orders of magnitude, which means that most digital images encode only a small fraction of what a human observer can see.

In this sense, negative photography is superior to digital imaging in its ability to capture the dynamic range of a scene. A typical, consumer-grade color negative film has about 5-8 f-stops of *exposure latitude*, meaning that it can capture regions of a scene that are 2^5 to 2^8 times brighter than the camera's exposure setting (or dimmer if the image is overexposed), and still have enough range left over to reproduce each region*. Of course, most prints do not make use of the full range, unless a photographer picks up a wand or a cutout in the darkroom, but its presence permits re-exposure during the printing process to optimize the appearance of salient features, such as a person's face.

* To compute the latitude of a film or recording medium, take the log to the base 2 of the total usable dynamic range, from darkest unique value to brightest, and subtract 5 f-stops, which is the approximate range required for a usable image. There are about 3.3 f-stops per order of magnitude.

The question to ask is this: in 10 years or 100 years, what medium will be preferred for old photographs, a digital image, or a negative? Unless we change the way digital images are encoded, the answer in most cases will be a negative. Even considering aging and degradation (processes that can be partially compensated), a negative has both superior resolution and greater dynamic range than an RGB or YCC image. This needn't be the case.

In this paper, we present a compact pixel encoding using a log representation of luminance and a CIE (u',v') representation of color. We call this a *LogLuv* encoding. A log luminance representation means that at any exposure level, there will be equal brightness steps between values. This corresponds well with human visual response, whose contrast threshold is constant over a wide range of adaptation luminances (Weber's law). For color, the use of an approximately uniform perceptual space enables us to record the full gamut of visible colors using step sizes that are imperceptible to the eye. The combination of these two techniques permits us to make nearly optimal use of the bits available to record a given pixel, so that it may be reproduced over a broad range of viewing conditions. Also, since we are recording the full visible gamut and dynamic range, the output or display device can be *anything* and we won't be able to detect any errors or artifacts from our representation, simply because they will be reproduced below the visible threshold.

In this paper, we describe our LogLuv pixel encoding method, followed by a description of our extension to Sam Leffler's free TIFF library. We then put forth a proposal for extending the Flashpix format, and follow this with an example image to demonstrate the value of this encoding, ending with a brief conclusion.

Encoding Method

We have implemented two LogLuv pixel encodings, a 24-bit encoding and a 32-bit encoding. The 24-bit encoding breaks down into a 10-bit log luminance portion and a 14-bit, indexed uv coordinate mapping. Color indexing minimizes waste, allowing us to cover the irregular shape of the visible gamut in imperceptible steps. The 32-bit encoding uses 16 bits for luminance and 8 bits each for u' and v' . Compared to the 24-bit encoding, the 32-bit version provides greater dynamic range and precision at the cost of an extra byte per pixel. The exact interpretations of these two encodings are described below.

24-bit Encoding

In 24 bits, we can pack much more visible information than is commonly stored in three gamma-compressed 8-bit color primary values. By separating luminance and using a log encoding, we can use 10 bits to record nearly 5 orders of magnitude in 1.1% relative steps that will be imperceptible under most conditions. The remaining 14 bits will be used to store a color index

corresponding to the smallest distinguishable patch size on a uv color chart. The bit allocation is shown graphically in Fig. 1.



Figure 1. 24-bit encoding. L_e is the encoded log luminance, and C_e is the encoded uv color index.

To compute the integer encoding L_e from real luminance, L , we use the formula given in Eq. 1a. To compute real luminance from L_e , we use the inverse formula given in Eq. 1b.

$$L_e = \lfloor 64(\log_2 L + 12) \rfloor \quad (1a)$$

$$L = \exp_2 \left[\frac{(L_e + 0.5)}{64} - 12 \right] \quad (1b)$$

In addition, an L_e value of 0 is taken to equal 0.0 exactly. An L_e value of 1 corresponds to a real luminance value of 0.000248 on an arbitrary scale, and the maximum L_e value of 1023 corresponds to a real value of 15.9 for a dynamic range of 65,000:1, or 4.8 orders of magnitude. It is difficult to compare this range to an 8-bit gamma-compressed encoding, because 1.1% accuracy is possible only near the very top of the 8-bit range. Allowing the luminance error to go as high as 5%, the dynamic range of an 8-bit encoding with a nominal gamma of 2.2 is 47:1, or 1.7 orders of magnitude. This leaves less than one f-stop of exposure latitude, compared to 11 f-stops for our 10-bit log encoding.

To capture full-gamut chrominance using only 14 bits, we cannot afford to waste codes on imaginary colors. We therefore divide our "perceptually uniform" (u',v') color space [8] into equal area regions using a scanline traversal over the visible gamut. This encoding concept is shown graphically in Fig. 2. The actual encoding has many more scanlines of course (163 to be exact), but the figure shows roughly how they are laid out. The minimum code value (0) is at the lower left, and codes are assigned left to right along each scanline until the maximum value (just less than 2^{14}) is assigned to the rightmost value on the top scanline.

$$u' = \frac{4x}{-2x + 12y + 3} \quad (2a)$$

$$v' = \frac{9y}{-2x + 12y + 3} \quad (2b)$$

To encode a given color, we start with the standard conversion from CIE (x,y) chromaticity to (u',v') shown in Eq. 2. We then look up the appropriate scanline for our v' value based on a uniform scanline height, and compute the position within the scanline using our uniform cell width. The index C_e is equal to the total of the scanlines below us plus the cells to the left in this

scanline. Cell width and height are both set to 0.0035 in our implementation, which corresponds to slightly less than the minimum perceptible step in this color space and uses up nearly all of the codes available in 14 bits.

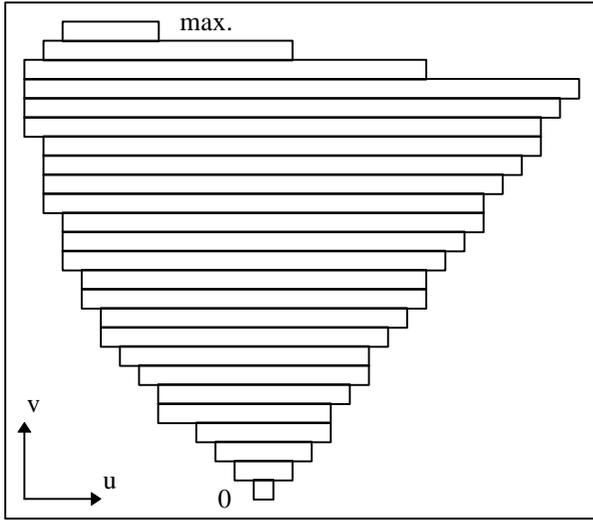


Figure 2. Scanline traversal of (u,v) coordinate space for 14-bit chromaticity encoding.

To get back the (x,y) chromaticity corresponding to a specific color index, we may either use a 16 Kentry look-up table, or apply a binary search to find the scanline containing corresponding to our C_e index. Once we have our original (u',v') coordinates back, we can apply the inverse conversion given in Eq. 3 to get the CIE chromaticity coordinates. (Note that this final computation may also be avoided using the same look-up table.)

$$x = \frac{9u'}{6u' - 16v' + 12} \quad (3a)$$

$$y = \frac{4v'}{6u' - 16v' + 12} \quad (3b)$$

32-bit Encoding

The 32-bit encoding is actually simpler, since we have 16 bits for (u',v') , which is more than enough that we can dispense with the complex color indexing scheme. The encoding of luminance is similar, with the addition of a sign bit so that negative luminances may also be encoded. In the remaining 15 bits, we can record over 38 orders of magnitude in 0.27% relative steps, covering the full range of perceivable world luminances in imperceptible steps. The bit breakdown is shown in Fig. 3.

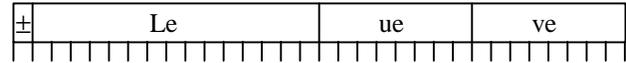


Figure 3. Bit allocation for 32-bit pixel encoding. MSB is a sign bit, and the next 15 bits are used for a log luminance encoding. The uv coordinates are separate 8-bit quantities.

The conversion to and from our log luminance encoding is given in Eq. 4. The maximum luminance using this encoding is 1.84×10^{19} , and the smallest magnitude is 5.44×10^{-20} . As in the 10-bit encoding, an L_e value of 0 is taken to be exactly 0.0. The sign bit is extracted before encoding and reapplied after the conversion back to real luminance.

$$L_e = \lfloor 256(\log_2 L + 64) \rfloor \quad (4a)$$

$$L = \exp_2 \left[\left(\frac{L_e + 0.5}{256 - 64} \right) \right] \quad (4b)$$

As we mentioned, the encoding of chrominance is simplified because we have enough bits to record u_e and v_e separately. Since the gamut of u and v values is between 0 and 0.62, we chose a scale factor of 410 to go between our $[0,255]$ integer range and real coordinates, as given in Eq. 5.

$$u_e = \lfloor 410u' \rfloor \quad (5a)$$

$$v_e = \lfloor 410v' \rfloor \quad (5b)$$

$$u' = (u_e + 0.5) / 410 \quad (5c)$$

$$v' = (v_e + 0.5) / 410 \quad (5d)$$

This encoding captures the full color gamut in 8 bits each for u_e and v_e . There will be some unused codes outside the visible gamut, but the tolerance this gives us of 0.0017 units in uv space is already well below the visible threshold. Conversions to and from CIE (x,y) chromaticities are the same as given earlier in Eqs. 2 and 3.

TIFF Input/Output Library

The LogLuv encodings described have been embedded as a new SGILOG compression type in Sam Leffler's popular TIFF I/O library. This library is freely distributed by anonymous ftp on ftp.sgi.com in the "/graphics/tiff/" directory.

When writing a high dynamic range (HDR) TIFF image, the LogLuv *codec* (compression/decompression module) takes floating point CIE XYZ scanlines and writes out 24-bit or 32-bit compressed LogLuv-encoded values. When reading an HDR TIFF, the reverse conversion is performed to get back floating point XYZ values. (We also provide a simple conversion to 24-bit gamma-compressed RGB for the convenience of readers that do not know how to handle HDR pixels.)

An additional tag is provided for absolute luminance calibration, named `TIFFTAG_STONITS`. This is a single floating point value that may be used to convert Y values returned by the reader to absolute luminance in candelas per square meter. This tag may also be set by the application that writes out a HDR TIFF to permit calibrated scaling of values to a reasonable brightness range, where values of 1.0 will be displayed at the maximum output of the destination device. This scale factor may also be necessary for calibration of the 24-bit format due to its more limited dynamic range.

Run-length Compression

Although at first it may appear that the 24-bit code is a more compact representation, the 32-bit encoding offers some advantages when it comes to applying nondestructive techniques to reduce storage requirements. By separating the bytes into four streams on each scanline, the 32-bit encoding can be efficiently compressed using an adaptive run-length encoding [3]. Since the top byte containing the sign bit and upper 7 log luminance bits changes very slowly, this byte-stream submits very well to run-length encoding. Likewise, the encoded u_e and v_e byte-streams compress well over areas of constant color. In contrast, the 24-bit encoding does not have a nice byte-stream breakup, so we do not attempt to run-length encode it, and the resulting files are quite often larger than the same data stored in the 32-bit format.

Grayscale Images

For maximum flexibility, a pure luminance mode is also provided by the codec, which stores and retrieves run-length encoded 16-bit log luminance values using the same scheme as applied in the 32-bit LogLuv encoding. There is no real space savings over a straight 32-bit encoding, since the u_e and v_e byte-streams compress to practically nothing for grayscale data, but this option provides an explicit way to specify floating point luminance images for TIFF readers that care.

Raw I/O

It is also possible to decode the raw 24-bit and 32-bit LogLuv data retrieved from an HDR TIFF directly, and this has some advantages for implementing fast tone mapping and display algorithms. In the case of the 24-bit format, one can simply multiply the output of a 1 Kentry L_e table and a 16 Kentry C_e table to get a tone-mapped and gamma-compressed RGB result. The 32-bit encoding requires a little more work, since its precomputed tables are 32 and 64 Kentries, but the same logic applies.

We have implemented this type of integer-math tone-mapping algorithm in an HDR image viewer, and it takes about a second to load and display a 512 by 512 picture on a 180 MHz processor.

Example TIFF Code and Images

Use of this encoding is demonstrated and sample images are provided on the following web site:

<http://www.sgi.com/Technology/pixformat/>

A converter has been written to and from the *Radiance* floating point picture format [6][7], and serves as an example of LogLuv codec usage. The web site itself also offers programming tips and example code segments.

Example TIFF images using the 32-bit LogLuv and 16-bit LogL encoding are provided on the web site. These images are either scanned from photographic negatives or rendered using *Radiance* and converted to the new TIFF format. Some images are rendered as 360° QuickTime VR panoramas suitable for experiments in HDR virtual reality.

Proposed Extension to Flashpix

The *Flashpix* format was originally developed by Kodak in collaboration with Hewlett-Packard, Live Picture and Microsoft. Its definition and maintenance has since been taken over by the Digital Imaging Group, a consortium of these and other companies. Flashpix is basically a multiresolution JPEG encoding, optimized for quick loading and editing at arbitrary pixel densities. It supports standard RGB as well as YCC color spaces with 8 bits/primary maximum resolution. For further information, see the DIG web site:

<http://www.digitalimaging.org>

Because Flashpix starts with 8-bit gamma-compressed color primaries, the dynamic range is limited to the same 1.7 orders of magnitude provided by other 24-bit RGB encodings. Furthermore, since JPEG encoding is applied, there will be additional losses and artifacts depending on the source image and the compression quality setting.

We cannot directly replace the JPEG-encoded Flashpix image with our own, alternate format, since this would violate standard compatibility as put forth by Kodak and enforced by the DIG. We must therefore provide any enhancement to the format as an optional extension, which results in a certain amount of redundancy in our case since the same pixels may be represented by two encodings. This is unavoidable.

For our extension, we need a second layer of “deeper” image data be provided for Flashpix users and applications that demand it. There are two ways we might go about this. The simplest method is to completely duplicate the source image in a 24 or 32-bit/pixel LogLuv encoding. On average, this will take roughly four to sixteen times as much space as the original JPEG encoding. A more sophisticated method is to replace only those pixels that are out of gamut or otherwise inadequate in the original encoding. We discuss this method below.

High Dynamic Range Extension Layer

Our proposed extension consists of a layer added to the standard Flashpix format. This layer contains two logical elements, a *presence map* of which pixels are included in the layer, and the list of corresponding 24-bit LogLuv pixels. The presence map may be represented by an entropy-encoded bitmap, which will typically take up 5% to 15% as much space as the JPEG layer. The extended pixels themselves will take between one half and four times as much space as the original JPEG layer, depending on the proportion of out-of-gamut pixels in the original image.

For an image that is entirely within gamut in the JPEG encoding, the presence map will compress to almost nothing, and there will be no LogLuv pixels, so the total overhead will be less than 1% of the original image. If the image is mostly out of the JPEG gamut, then the presence map might take half a bit per pixel, and the additional data will be the same size as a 24-bit RGB image. A typical high dynamic range image with 15% out-of-gamut pixels will take roughly the same space for the extension layer as the multiresolution JPEG layer, so the total image size will be about twice what it was originally. If the information is being accessed over the internet, the HDR layer may be loaded as an option, so it does not cost extra unless and until it is needed.

Example Results

Fig. 4a shows a scanned photograph as it might appear on a PhotoCD using a YCC encoding. Since YCC can capture up to “200% reflectance,” we can apply a tone mapping operator to bring this extra dynamic range into our print, as shown in Fig. 5a. However, since many parts of the image were brighter than this 200% value, we still lose much of the sky and circumsolar region, and even the lighter asphalt in the foreground. In Fig. 4b, we see where 35% of the original pixels are outside the gamut of a YCC encoding.

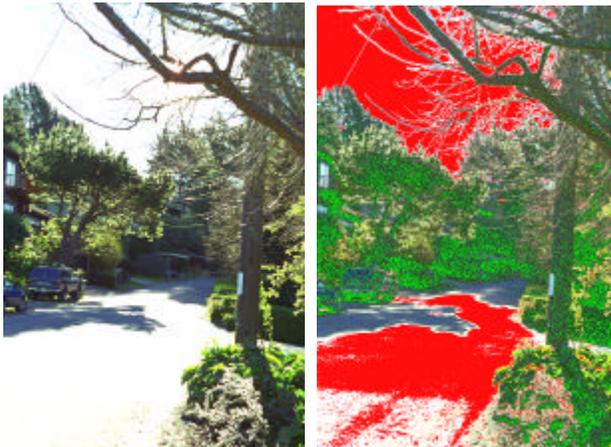


Figure 4. The left image (a) shows a PhotoYCC encoding of a color photograph tone-mapped with a linear operator. The right image (b) shows the out-of-gamut regions. Red areas are too bright or too dim, and green areas have inaccurate color.

Fig. 5b shows the same color negative scanned into our 32-bit/pixel high dynamic range TIFF format and tone mapped using a histogram compression technique [4]. Fig. 6c shows the same HDR TIFF remapped using the perceptual model of Pattanaik et al [5]. Figs. 6a and 6b show details of light and dark areas of the HDR image whose exposure has been adjusted to show the detail captured in the original negative. Without an HDR encoding, this information is either lost or unusable.



Figure 5. The left image (a) shows the YCC encoding after remapping with a high dynamic range tone operator [4]. Unfortunately, since YCC has so little dynamic range, most of the bright areas are lost. The right image (b) shows the same operator applied to a 32-bit HDR TIFF encoding, showing the full dynamic range of the negative.



Figure 6. The upper-left image (a) shows the circumsolar region reduced by 4 f-stops to show the image detail recorded on the negative. The lower-left image (b) shows house details boosted by 3 f-stops. The right image (c) shows our HDR TIFF mapped with the Pattanaik-Ferwerda tone operator [5].

Discussion

It is clear from looking at these images that current methods for tone-mapping HDR imagery, although better than a simple S-curve, are less than perfect. It would therefore be a mistake to store an image that has been irreversibly tone mapped in this fashion, as some scanner

software attempts to do. Storing an HDR image allows us to take full advantage of future improvements in tone mapping and display algorithms, at a nominal cost.

Besides professional photography, there are a number of application areas where HDR images are key. One is lighting simulation, where designers need to see an interior or exterior space as it would really appear, plus they need to evaluate things in terms absolute luminance and illuminance levels. Since an HDR image can store the real luminance in its full-gamut coverage, this information is readily accessible to the designer. Another application is image-based rendering, where a user is allowed to move about in a scene by warping captured or rendered images [1]. If these images have limited dynamic range, it is next to impossible to adapt the exposure based on the current view, and quality is compromised. Using HDR pixels, a natural view can be provided for any portion of the scene, no matter how bright or how dim. A fourth application area is digital archiving, where we are making a high-quality facsimile of a work of art for posterity. In this case, the pixels we record are precious, so we want to make sure they contain as much information as possible. At the same time, we have concerns about storage space and transmission costs, so keeping this data as compact as possible is important. Since our HDR format requires little more space than a standard 24-bit encoding to capture the full visible gamut, it is a clear winner for archiving applications.

Our essential argument is that we can make better use of the bits in each pixel by adopting a perceptual encoding of color and brightness. Although we don't know how a given image might be used or displayed in the future, we do know something about what a human can observe in a given scene. By faithfully recording this information, we ensure that our image will take full advantage of any future improvements in imaging technology, and our basic format will continue to find new uses.

Conclusion

We have presented a new method for encoding high dynamic range digital images using log luminance and uv chromaticity to capture the entire visible range of color and brightness. The proposed format requires little additional storage per pixel, while providing significant benefits to suppliers, caretakers and consumers of digital imagery.

Through the use of re-exposure and dynamic range compression, we have been able to show some of the benefits of HDR imagery. However, it is more difficult to illustrate the benefits of a larger color gamut without carefully comparing hard copy output of various multi-ink printers. Also, since we currently lack the ability to

capture highly saturated scenes, our examples would have to be contrived from individual spectral measurements and hypothetical scenes. We therefore leave this as a future exercise.

Future work on the format itself should focus on the application of lossy compression methods (such as JPEG and fractal image encoding) for HDR images. Without such methods, the storage cost for a given resolution may hinder broad acceptance of this representation. Another extension we should look at is multispectral data, which is needed for remote imaging and some types of lighting simulation.

References

1. Paul Debevec, "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography," *Computer Graphics (Proceedings of ACM Siggraph 98)*.
2. Paul Debevec, Jitendra Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," *Computer Graphics (Proceedings of ACM Siggraph 97)*.
3. Andrew Glassner, "Adaptive Run-Length Encoding," in *Graphics Gems II*, edited by James Arvo, Academic Press, (1991).
4. Greg Larson, Holly Rushmeier, Christine Piatko, "A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes," *IEEE Transactions on Visualization and Computer Graphics*, 3, 4, (1997).
5. Sumant Pattanaik, James Ferwerda, Mark Fairchild, Don Greenberg, "A Multiscale Model of Adaptation and Spatial Vision for Realistic Image Display," *Computer Graphics (Proceedings of Siggraph 98)*.
6. Greg Ward, "The RADIANCE Lighting Simulation and Rendering System," *Computer Graphics (Proceedings of Siggraph 94)*.
7. Greg Ward, "Real Pixels," in *Graphics Gems II*, edited by James Arvo, Academic Press, (1991).
8. Gunter Wyszecki, W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, Second Edition, Wiley, (1982).

Biography

Gregory Ward Larson is a member of the technical staff in the engineering division of SGI. He graduated with an AB in Physics in 1983 from the UC Berkeley, and earned his Master's in CS from San Francisco State in 1985. Greg has done work in physically-based rendering, surface reflectance measurements, and electronic data standards. He is the developer of the widely-used *Radiance* synthetic imaging system and the MGF exchange standard for scene data.

Greg may be reached by e-mail at gregl@sgi.com.

Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography

Paul Debevec

University of California at Berkeley¹

ABSTRACT

We present a method that uses measured scene radiance and global illumination in order to add new objects to light-based models with correct lighting. The method uses a high dynamic range image-based model of the scene, rather than synthetic light sources, to illuminate the new objects. To compute the illumination, the scene is considered as three components: the distant scene, the local scene, and the synthetic objects. The distant scene is assumed to be photometrically unaffected by the objects, obviating the need for reflectance model information. The local scene is endowed with estimated reflectance model information so that it can catch shadows and receive reflected light from the new objects. Renderings are created with a standard global illumination method by simulating the interaction of light amongst the three components. A differential rendering technique allows for good results to be obtained when only an estimate of the local scene reflectance properties is known.

We apply the general method to the problem of rendering synthetic objects into real scenes. The light-based model is constructed from an approximate geometric model of the scene and by using a light probe to measure the incident illumination at the location of the synthetic objects. The global illumination solution is then composited into a photograph of the scene using the differential rendering technique. We conclude by discussing the relevance of the technique to recovering surface reflectance properties in uncontrolled lighting situations. Applications of the method include visual effects, interior design, and architectural visualization.

CR Descriptors: I.2.10 [Artificial Intelligence]: Vision and Scene Understanding - *Intensity, color, photometry and thresholding*; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - *Color, shading, shadowing, and texture*; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - *Radiosity*; I.4.1 [Image Processing]: Digitization - *Scanning*; I.4.8 [Image Processing]: Scene Analysis - *Photometry, Sensor Fusion*.

¹Computer Science Division, University of California at Berkeley, Berkeley, CA 94720-1776. Email: debevec@cs.berkeley.edu. More information and additional results may be found at: <http://www.cs.berkeley.edu/~debevec/Research>

1 Introduction

Rendering synthetic objects into real-world scenes is an important application of computer graphics, particularly in architectural and visual effects domains. Oftentimes, a piece of furniture, a prop, or a digital creature or actor needs to be rendered seamlessly into a real scene. This difficult task requires that the objects be lit consistently with the surfaces in their vicinity, and that the interplay of light between the objects and their surroundings be properly simulated. Specifically, the objects should cast shadows, appear in reflections, and refract, focus, and emit light just as real objects would.

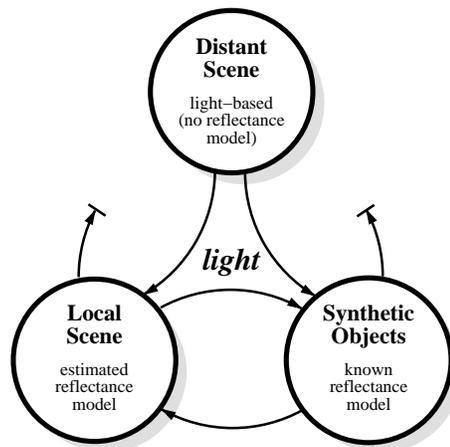


Figure 1: **The General Method** In our method for adding synthetic objects into light-based scenes, the scene is partitioned into three components: the distant scene, the local scene, and the synthetic objects. Global illumination is used to simulate the interplay of light amongst all three components, except that light reflected back at the distant scene is ignored. As a result, BRDF information for the distant scene is unnecessary. Estimates of the geometry and material properties of the local scene are used to simulate the interaction of light between it and the synthetic objects.

Currently available techniques for realistically rendering synthetic objects into scenes are labor intensive and not always successful. A common technique is to manually survey the positions of the light sources, and to instantiate a virtual light of equal color and intensity for each real light to illuminate the synthetic objects. Another technique is to photograph a reference object (such as a gray sphere) in the scene where the new object is to be rendered, and use its appearance as a qualitative guide in manually configuring the lighting environment. Lastly, the technique of reflection mapping is useful for mirror-like reflections. These methods typically require considerable hand-refinement and none of them easily simulates the effects of indirect illumination from the environment.

Accurately simulating the effects of both direct and indirect lighting has been the subject of research in global illumination. With a global illumination algorithm, if the entire scene were modeled with its full geometric and reflectance (BRDF) characteristics, one could correctly render a synthetic object into the scene simply by adding it to the model and recomputing the global illumination solution. Unfortunately, obtaining a full geometric and reflectance model of a large environment is extremely difficult. Furthermore, global illumination solutions for large complex environments are extremely computationally intensive.

Moreover, it seems that having a full reflectance model of the large-scale scene should be unnecessary: under most circumstances, a new object will have no significant effect on the appearance of most of the of the distant scene. Thus, for such distant areas, knowing just its radiance (under the desired lighting conditions) should suffice.

Recently, [9] introduced a high dynamic range photographic technique that allows accurate measurements of scene radiance to be derived from a set of differently exposed photographs. This technique allows both low levels of indirect radiance from surfaces and high levels of direct radiance from light sources to be accurately recorded. When combined with image-based modeling techniques (e.g. [22, 24, 4, 10, 23, 17, 29]), and possibly active techniques for measuring geometry (e.g. [35, 30, 7, 27]) these derived radiance maps can be used to construct spatial representations of scene radiance.

We will use the term **light-based model** to refer to a representation of a scene that consists of radiance information, possibly with specific reference to light leaving surfaces, but not necessarily containing material property (BRDF) information. A light-based model can be used to evaluate the 5D plenoptic function [1] $P(\theta, \phi, V_x, V_y, V_z)$ for a given virtual or real subset of space¹. A material-based model is converted to a light-based model by computing an illumination solution for it. A light-based model is differentiated from an image-based model in that its light values are actual measures of radiance², whereas image-based models may contain pixel values already transformed and truncated by the response function of an image acquisition or synthesis process.

In this paper, we present a general method for using accurate measurements of scene radiance in conjunction with global illumination to realistically add new objects to light-based models. The synthetic objects may have arbitrary material properties and can be rendered with appropriate illumination in arbitrary lighting environments. Furthermore, the objects can correctly interact with the environment around them: they cast the appropriate shadows, they are properly reflected, they can reflect and focus light, and they exhibit appropriate diffuse interreflection. The method can be carried out with commonly available equipment and software.

In this method (see Fig. 1), the scene is partitioned into three components. The first is the distant scene, which is the visible part of the environment too remote to be perceptibly affected by the synthetic object. The second is the local scene, which is the part of the environment which will be significantly affected by the presence of the objects. The third component is the synthetic objects. Our approach uses global illumination to correctly simulate the interaction of light amongst these three elements, with the exception that light radiated toward the distant environment will not be considered in the calculation. As a result, the BRDF of the distant environment need not be known — the technique uses BRDF information only for the local scene and the synthetic objects. We discuss the challenges in estimating the BRDF of the local scene, and methods for obtaining usable approximations. We also present a differential rendering

¹Time and wavelength dependence can be included to represent the general 7D plenoptic function as appropriate.

²In practice, the measures of radiance are with respect to a discrete set of spectral distributions such as the standard tristimulus model.

technique that produces perceptually accurate results even when the estimated BRDF is somewhat inaccurate.

We demonstrate the general method for the specific case of rendering synthetic objects into particular views of a scene (such as background plates) rather than into a general image-based model. In this method, a light probe is used to acquire a high dynamic range panoramic radiance map near the location where the object will be rendered. A simple example of a light probe is a camera aimed at a mirrored sphere, a configuration commonly used for acquiring environment maps. An approximate geometric model of the scene is created (via surveying, photogrammetry, or 3D scanning) and mapped with radiance values measured with the light probe. The distant scene, local scene, and synthetic objects are rendered with global illumination from the same point of view as the background plate, and the results are composited into the background plate with a differential rendering technique.

1.1 Overview

The rest of this paper is organized as follows. In the next section we discuss work related to this paper. Section 3 introduces the basic technique of using acquired maps of scene radiance to illuminate synthetic objects. Section 4 presents the general method we will use to render synthetic objects into real scenes. Section 5 describes a practical technique based on this method using a *light probe* to measure incident illumination. Section 6 presents a differential rendering technique for rendering the local environment with only an approximate description of its reflectance. Section 7 presents a simple method to approximately recover the diffuse reflectance characteristics of the local environment. Section 8 presents results obtained with the technique. Section 9 discusses future directions for this work, and we conclude in Section 10.

2 Background and Related Work

The practice of adding new objects to photographs dates to the early days of photography in the simple form of pasting a cut-out from one picture onto another. While the technique conveys the idea of the new object being in the scene, it usually fails to produce an image that as a whole is a believable photograph. Attaining such realism requires a number of aspects of the two images to match. First, the camera projections should be consistent, otherwise the object may seem too foreshortened or skewed relative to the rest of the picture. Second, the patterns of film grain and film response should match. Third, the lighting on the object needs to be consistent with other objects in the environment. Lastly, the object needs to cast realistic shadows and reflections on the scene. Skilled artists found that by giving these considerations due attention, synthetic objects could be painted into still photographs convincingly.

In optical film compositing, the use of object mattes to prevent particular sections of film from being exposed made the same sort of cut-and-paste compositing possible for moving images. However, the increased demands of realism imposed by the dynamic nature of film made matching camera positions and lighting even more critical. As a result, care was taken to light the objects appropriately for the scene into which they were to be composited. This would still not account for the objects casting shadows onto the scene, so often these were painted in by an artist frame by frame [13, 2, 28]. Digital film scanning and compositing [26] helped make this process far more efficient.

Work in global illumination [16, 19] has recently produced algorithms (e.g. [31]) and software (e.g. [33]) to realistically simulate lighting in synthetic scenes, including indirect lighting with both specular and diffuse reflections. We leverage this work in order to create realistic renderings.

Some work has been done on the specific problem of compositing objects into photography. [25] presented a procedure for ren-

dering architecture into background photographs using knowledge of the sun position and measurements or approximations of the local ambient light. For diffuse buildings in diffuse scenes, the technique is effective. The technique of *reflection mapping* (also called *environment mapping*) [3, 18] produces realistic results for mirror-like objects. In reflection mapping, a panoramic image is rendered or photographed from the location of the object. Then, the surface normals of the object are used to index into the panoramic image by reflecting rays from the desired viewpoint. As a result, the shiny object appears to properly reflect the desired environment³. However, the technique is limited to mirror-like reflection and does not account for objects casting light or shadows on the environment.

A common visual effects technique for having synthetic objects cast shadows on an existing environment is to create an approximate geometric model of the environment local to the object, and then compute the shadows from the various light sources. The shadows can then be subtracted from the background image. In the hands of professional artists this technique can produce excellent results, but it requires knowing the position, size, shape, color, and intensity of each of the scene's light sources. Furthermore, it does not account for diffuse reflection from the scene, and light reflected by the objects onto the scene must be handled specially.

To properly model the interaction of light between the objects and the local scene, we pose the compositing problem as a global illumination computation as in [14] and [12]. As in this work, we apply the effect of the synthetic objects in the lighting solution as a differential update to the original appearance of the scene. In the previous work an approximate model of the entire scene and its original light sources is constructed; the positions and sizes of the light sources are measured manually. Rough methods are used to estimate diffuse-only reflectance characteristics of the scene, which are then used to estimate the intensities of the light sources. [12] additionally presents a method for performing fast updates of the illumination solution in the case of moving objects. As in the previous work, we leverage the basic result from incremental radiosity [6, 5] that making a small change to a scene does not require recomputing the entire solution.

3 Illuminating synthetic objects with real light

In this section we propose that computer-generated objects be lit by actual recordings of light from the scene, using global illumination. Performing the lighting in this manner provides a unified and physically accurate alternative to manually attempting to replicate incident illumination conditions.

Accurately recording light in a scene is difficult because of the high dynamic range that scenes typically exhibit; this wide range of brightness is the result of light sources being relatively concentrated. As a result, the intensity of a source is often two to six orders of magnitude larger than the intensity of the non-emissive parts of an environment. However, it is necessary to accurately record both the large areas of indirect light from the environment and the concentrated areas of direct light from the sources since both are significant parts of the illumination solution.

Using the technique introduced in [9], we can acquire correct measures of scene radiance using conventional imaging equipment. The images, called *radiance maps*, are derived from a series of images with different sensor integration times and a technique for computing and accounting for the imaging system response function f . We can use these measures to illuminate synthetic objects exhibiting arbitrary material properties.

Fig. 2 shows a high-dynamic range lighting environment with electric, natural, and indirect lighting. This environment was

³Using the surface normal indexing method, the object will not reflect itself. Correct self-reflection can be obtained through ray tracing.

recorded by taking a full dynamic range photograph of a mirrored ball on a table (see Section 5). A digital camera was used to acquire a series of images in one-stop exposure increments from $\frac{1}{4}$ to $\frac{1}{10000}$ second. The images were fused using the technique in [9].

The environment is displayed at three exposure levels (-0, -3.5, and -7.0 stops) to show its full dynamic range. Recovered RGB radiance values for several points in the scene and on the two major light sources are indicated; the color difference between the tungsten lamp and the sky is evident. A single low-dynamic range photograph would be unable to record the correct colors and intensities over the entire scene.

Fig. 3(a-e) shows the results of using this panoramic radiance map to synthetically light a variety of materials using the RADIANCE global illumination algorithm [33]. The materials are: (a) perfectly reflective, (b) rough gold, (c) perfectly diffuse gray material, (d) shiny green plastic, and (e) dull orange plastic. Since we are computing a full illumination solution, the objects exhibit self-reflection and shadows from the light sources as appropriate. Note that in (c) the protrusions produce two noticeable shadows of slightly different colors, one corresponding to the ceiling light and a softer shadow corresponding to the window.

The shiny plastic object in (d) has a 4 percent specular component with a Gaussian roughness of 0.04 [32]. Since the object's surface both blurs and attenuates the light with its rough specular component, the reflections fall within the dynamic range of our display device and the different colors of the light sources can be seen. In (e) the rough plastic diffuses the incident light over a much larger area.

To illustrate the importance of using high dynamic range radiance maps, the same renderings were produced using just one of the original photographs as the lighting environment. In this single image, similar in appearance to Fig. 2(a), the brightest regions had been truncated to approximately 2 percent of their true values. The rendering of the mirrored surface (f) appears similar to (a) since it is displayed in low-dynamic range printed form. Significant errors are noticeable in (g-j) since these materials blur the incident light. In (g), the blurring of the rough material darkens the light sources, whereas in (b) they remain saturated. Renderings (h-j) are very dark due to the missed light; thus we have brightened by a factor of eight on the right in order to make qualitative comparisons to (c-e) possible. In each it can be seen that the low-dynamic range image of the lighting environment fails to capture the information necessary to simulate correct color balance, shadows, and highlights.

Fig. 4 shows a collection of objects with different material properties illuminated by two different environments. A wide variety of light interaction between the objects and the environment can be seen. The (synthetic) mirrored ball reflects both the synthetic objects as well as the environment. The floating diffuse ball shows a subtle color shift along its right edge as it shadows itself from the windows and is lit primarily by the incandescent lamp in Fig. 4(a). The reflection of the environment in the black ball (which has a specular intensity of 0.04) shows the colors of the light sources, which are too bright to be seen in the mirrored ball. A variety of shadows, reflections, and focused light can be observed on the resting surface.

The next section describes how the technique of using radiance maps to illuminate synthetic objects can be extended to compute the proper photometric interaction of the objects with the scene. It also describes how high dynamic range photography and image-based modeling combine in a natural manner to allow the simulation of arbitrary (non-infinite) lighting environments.

4 The General Method

This section explains our method for adding new objects to light-based scene representations. As in Fig. 1, we partition our scene into three parts: the distant scene, the local scene, and the synthetic

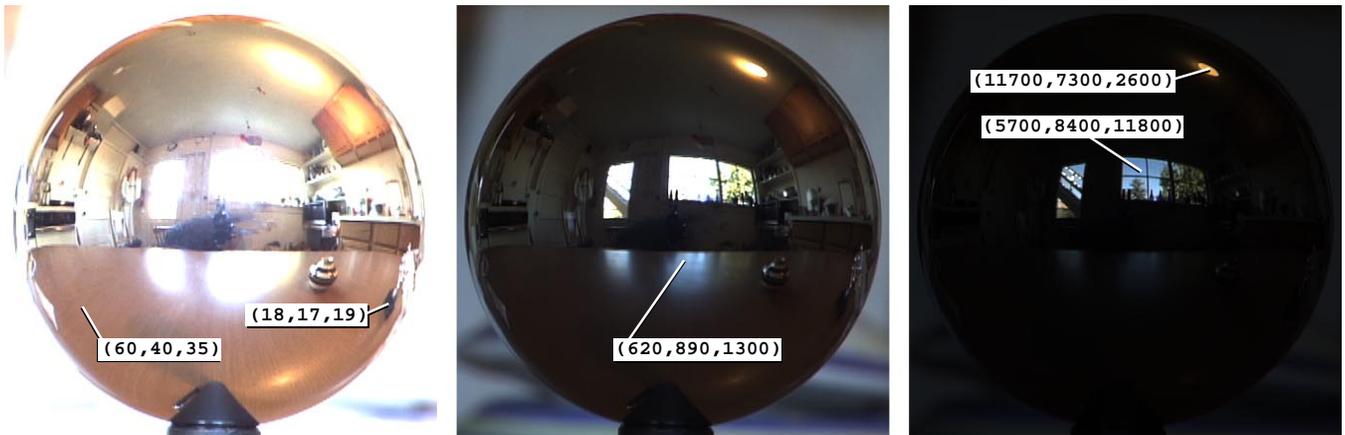


Figure 2: **An omnidirectional radiance map** This full dynamic range lighting environment was acquired by photographing a mirrored ball balanced on the cap of a pen sitting on a table. The environment contains natural, electric, and indirect light. The three views of this image adjusted to (a) +0 stops, (b) -3.5 stops, and (c) -7.0 stops show that the full dynamic range of the scene has been captured without saturation. As a result, the image usefully records the direction, color, and intensity of all forms of incident light.

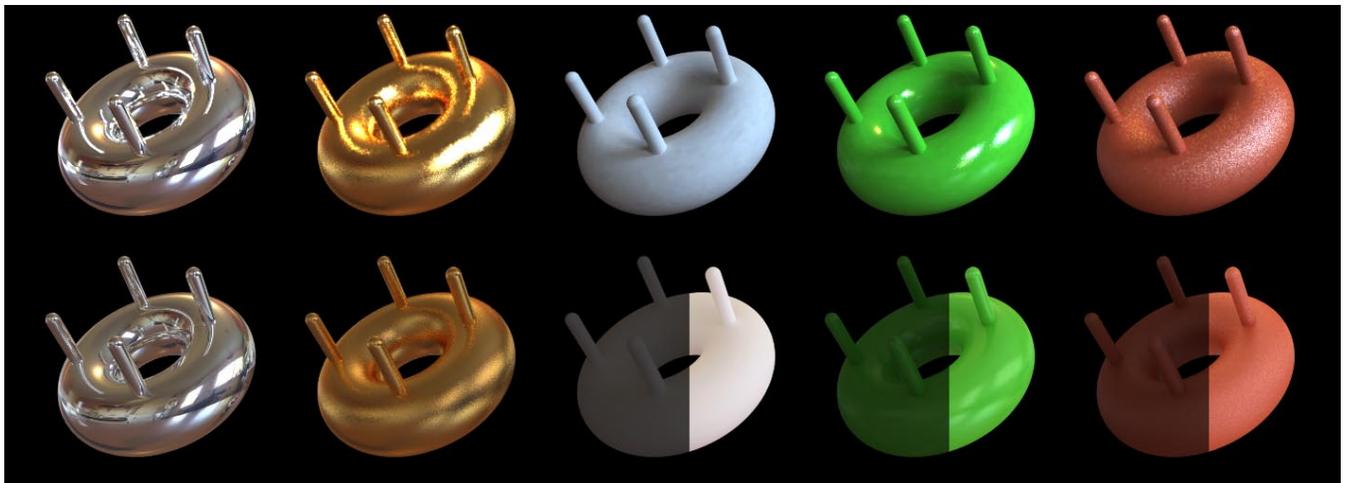


Figure 3: **Illuminating synthetic objects with real light** (Top row: a,b,c,d,e) With full dynamic range measurements of scene radiance from Fig. 2. (Bottom row: f,g,h,i,j) With low dynamic range information from a single photograph of the ball. The right sides of images (h,i,j) have been brightened by a factor of six to allow qualitative comparison to (c,d,e). The high dynamic range measurements of scene radiance are necessary to produce proper lighting on the objects.

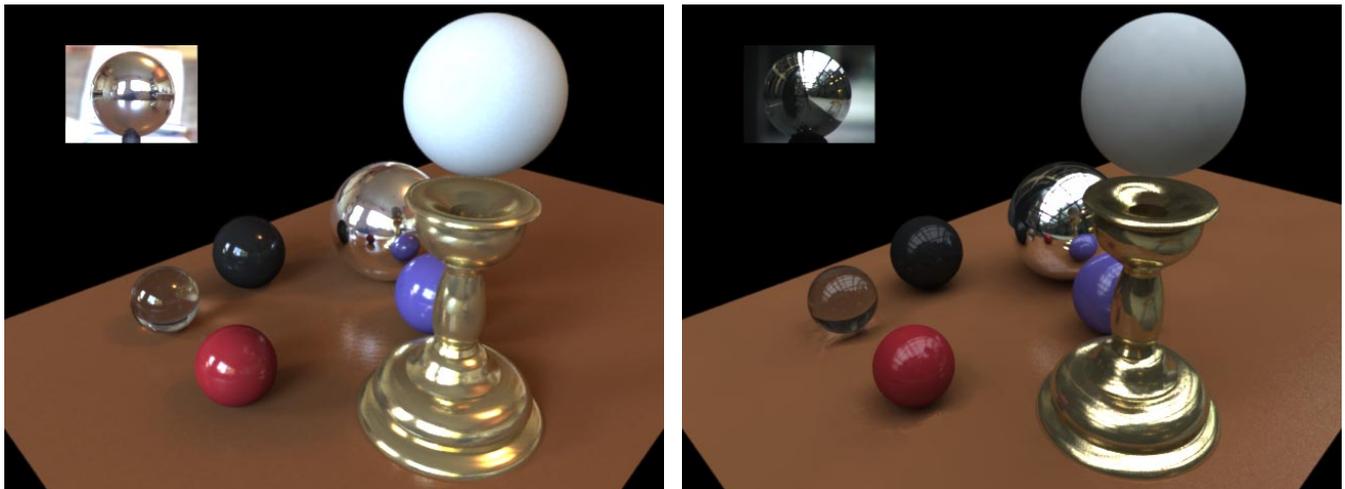


Figure 4: **Synthetic objects lit by two different environments** (a) A collection of objects is illuminated by the radiance information in 2. The objects exhibit appropriate interreflection. (b) The same objects are illuminated by different radiance information obtained in an outdoor urban environment on an overcast day. The radiance map used for the illumination is shown in the upper left of each image. Candle holder model courtesy of Gregory Ward Larson.

objects. We describe the geometric and photometric requirements for each of these components.

1. A light-based model of the distant scene

The distant scene is constructed as a light-based model. The synthetic objects will receive light from this model, so it is necessary that the model store true measures of radiance rather than low dynamic range pixel values from conventional images. The light-based model can take on any form, using very little explicit geometry [23, 17], some geometry [24], moderate geometry [10], or be a full 3D scan of an environment with view-dependent texture-mapped [11] radiance. What is important is for the model to provide accurate measures of incident illumination in the vicinity of the objects, as well as from the desired viewpoint. In the next section we will present a convenient procedure for constructing a minimal model that meets these requirements.

In the global illumination computation, the distant scene radiates light toward the local scene and the synthetic objects, but ignores light reflected back to it. We assume that no area of the distant scene will be significantly affected by light reflecting from the synthetic objects; if that were the case, the area should instead belong to the local scene, which contains the BRDF information necessary to interact with light. In the RADIANCE [33] system, this exclusively emissive behavior can be specified with the "glow" material property.

2. An approximate material-based model of the local scene

The local scene consists of the surfaces that will photometrically interact with the synthetic objects. It is this geometry onto which the objects will cast shadows and reflect light. Since the local scene needs to fully participate in the illumination solution, both its geometry and reflectance characteristics should be known, at least approximately. If the geometry of the local scene is not readily available with sufficient accuracy from the light-based model of the distant scene, there are various techniques available for determining its geometry through active or passive methods. In the common case where the local scene is a flat surface that supports the synthetic objects, its geometry is determined easily from the camera pose. Methods for estimating the BRDF of the local scene are discussed in Section 7.

Usually, the local scene will be the part of the scene that is geometrically close to the synthetic objects. When the local scene is mostly diffuse, the rendering equation shows that the visible effect of the objects on the local scene decreases as the inverse square of the distance between the two. Nonetheless, there is a variety of circumstances in which synthetic objects can significantly affect areas of the scene not in the immediate vicinity. Some common circumstances are:

- If there are concentrated light sources illuminating the object, then the object can cast a significant shadow on a distant surface collinear with it and the light source.
- If there are concentrated light sources and the object is flat and specular, it can focus a significant amount of light onto a distant part of the scene.
- If a part of the distant scene is flat and specular (e.g. a mirror on a wall), its appearance can be significantly affected by a synthetic object.
- If the synthetic object emits light (e.g. a synthetic laser), it can affect the appearance of the distant scene significantly.

These situations should be considered in choosing which parts of the scene should be considered local and which parts distant. Any part of the scene that will be significantly affected in

its appearance from the desired viewpoint should be included as part of the local scene.

Since the local scene is a full BRDF model, it can be added to the global illumination problem as would any other object. The local scene may consist of any number of surfaces and objects with different material properties. For example, the local scene could consist of a patch of floor beneath the synthetic object to catch shadows as well as a mirror surface hanging on the opposite wall to catch a reflection. The local scene replaces the corresponding part of the light-based model of the distant scene.

Since it can be difficult to determine the precise BRDF characteristics of the local scene, it is often desirable to have only the *change* in the local scene's appearance be computed with the BRDF estimate; its appearance due to illumination from the distant scene is taken from the original light-based model. This differential rendering method is presented in Section 6.

3. Complete material-based models of the objects

The synthetic objects themselves may consist of any variety of shapes and materials supported by the global illumination software, including plastics, metals, emitters, and dielectrics such as glass and water. They should be placed in their desired geometric correspondence to the local scene.

Once the distant scene, local scene, and synthetic objects are properly modeled and positioned, the global illumination software can be used in the normal fashion to produce renderings from the desired viewpoints.

5 Compositing using a light probe

This section presents a particular technique for constructing a light-based model of a real scene suitable for adding synthetic objects at a particular location. This technique is useful for compositing objects into actual photography of a scene.

In Section 4, we mentioned that the light-based model of the distant scene needs to appear correctly in the vicinity of the synthetic objects as well as from the desired viewpoints. This latter requirement can be satisfied if it is possible to directly acquire radiance maps of the scene from the desired viewpoints. The former requirement, that the appear photometrically correct in all directions in the vicinity of the synthetic objects, arises because this information comprises the incident light which will illuminate the objects.

To obtain this part of the light-based model, we acquire a full dynamic range omnidirectional radiance map near the location of the synthetic object or objects. One technique for acquiring this radiance map is to photograph a spherical first-surface mirror, such as a polished steel ball, placed at or near the desired location of the synthetic object⁴. This procedure is illustrated in Fig. 7(a). An actual radiance map obtained using this method is shown in Fig. 2.

The radiance measurements observed in the ball are mapped onto the geometry of the distant scene. In many circumstances this model can be very simple. In particular, if the objects are small and resting on a flat surface, one can model the scene as a horizontal plane for the resting surface and a large dome for the rest of the environment. Fig. 7(c) illustrates the ball image being mapped onto a table surface and the walls and ceiling of a finite room; 5 shows the resulting light-based model.

5.1 Mapping from the probe to the scene model

To precisely determine the mapping between coordinates on the ball and rays in the world, one needs to record the position of the ball

⁴Parabolic mirrors combined with telecentric lenses [34] can be used to obtain hemispherical fields of view with a consistent principal point, if so desired.

relative to the camera, the size of the ball, and the camera parameters such as its location in the scene and focal length. With this information, it is straightforward to trace rays from the camera center through the pixels of the image, and reflect rays off the ball into the environment. Often a good approximation results from assuming the ball is small relative to the environment and that the camera's view is orthographic.

The data acquired from a single ball image will exhibit a number of artifacts. First, the camera (and possibly the photographer) will be visible. The ball, in observing the scene, interacts with it: the ball (and its support) can appear in reflections, cast shadows, and can reflect light back onto surfaces. Lastly, the ball will not reflect the scene directly behind it, and will poorly sample the area nearby. If care is taken in positioning the ball and camera, these effects can be minimized and will have a negligible effect on the final renderings. If the artifacts are significant, the images can be fixed manually in image editing program or by selectively combining images of the ball taken from different directions; Fig. 6 shows a relatively artifact-free environment constructed using the latter method. We have found that combining two images of the ball taken ninety degrees apart from each other allows us to eliminate the camera's appearance and to avoid poor sampling.

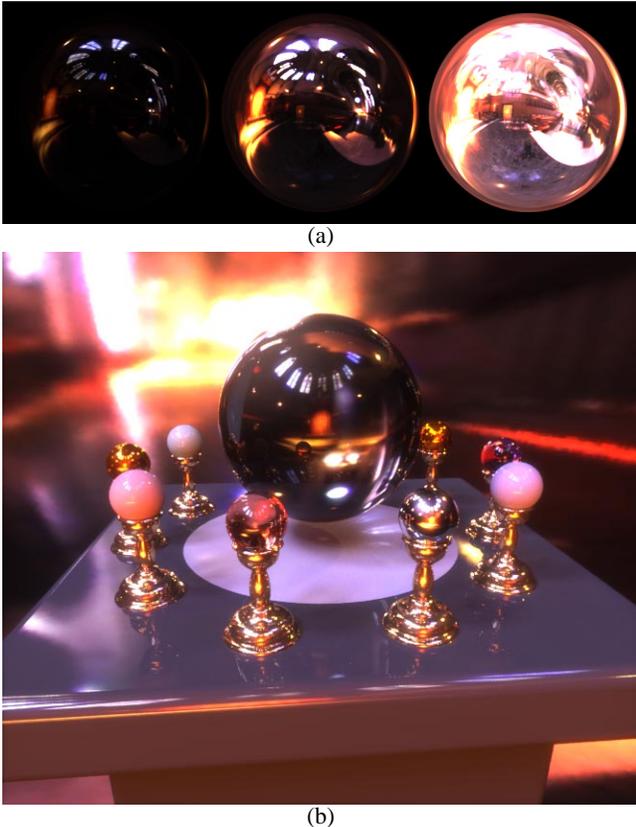


Figure 6: Rendering with a Combined Probe Image The full dynamic range environment map shown at the top was assembled from two light probe images taken ninety degrees apart from each other. As a result, the only visible artifact is small amount of the probe support visible on the floor. The map is shown at -4.5 , 0 , and $+4.5$ stops. The bottom rendering was produced using this lighting information, and exhibits diffuse and specular reflections, shadows from different sources of light, reflections, and caustics.

5.2 Creating renderings

To render the objects into the scene, a synthetic local scene model is created as described in Section 4. Images of the scene from the desired viewpoint(s) are taken (Fig. 7(a)), and their position relative to the scene is recorded through pose-instrumented cameras or (as in our work) photogrammetry. The location of the ball in the scene is also recorded at this time. The global illumination software is then run to render the objects, local scene, and distant scene from the desired viewpoint (Fig. 7(d)).

The objects and local scene are then composited onto the background image. To perform this compositing, a mask is created by rendering the objects and local scene in white and the distant scene in black. If objects in the distant scene (which may appear in front of the objects or local scene from certain viewpoints) are geometrically modeled, they will properly obscure the local scene and the objects as necessary. This compositing can be considered as a subset of the general method (Section 4) wherein the light-based model of the distant scene acts as follows: if (V_x, V_y, V_z) corresponds to an actual view of the scene, return the radiance value looking in direction (θ, ϕ) . Otherwise, return the radiance value obtained by casting the ray $(\theta, \phi, V_x, V_y, V_z)$ onto the radiance-mapped distant scene model.

In the next section we describe a more robust method of compositing the local scene into the background image.

6 Improving quality with differential rendering

The method we have presented so far requires that the local scene be modeled accurately in both its geometry and its spatially varying material properties. If the model is inaccurate, the appearance of the local scene will not be consistent with the appearance of adjacent distant scene. Such a border is readily apparent in Fig. 8(c), since the local scene was modeled with a homogeneous BRDF when in reality it exhibits a patterned albedo (see [21]). In this section we describe a method for greatly reducing such effects.

Suppose that we compute a global illumination solution for the local and distant scene models without including the synthetic objects. If the BRDF and geometry of the local scene model were perfectly accurate, then one would expect the appearance of the rendered local scene to be consistent with its appearance in the light-based model of the entire scene. Let us call the appearance of the local scene from the desired viewpoint in the light-based model LS_b . In the context of the method described in Section 5, LS_b is simply the background image. We will let LS_{nobj} denote the appearance of the local scene, without the synthetic objects, as calculated by the global illumination solution. The error in the rendered local scene (without the objects) is thus: $Err_{ls} = LS_{nobj} - LS_b$. This error results from the difference between the BRDF characteristics of the actual local scene as compared to the modeled local scene.

Let LS_{obj} denote the appearance of the local environment as calculated by the global illumination solution with the synthetic objects in place. We can compensate for the error if we compute our final rendering LS_{final} as:

$$LS_{final} = LS_{obj} - Err_{ls}$$

Equivalently, we can write:

$$LS_{final} = LS_b + (LS_{obj} - LS_{nobj})$$

In this form, we see that whenever LS_{obj} and LS_{nobj} are the same (i.e. the addition of the objects to the scene had no effect on the local scene) the final rendering of the local scene is equivalent to LS_b (e.g. the background plate). When LS_{obj} is darker than LS_{nobj} , light is subtracted from the background to form shadows,

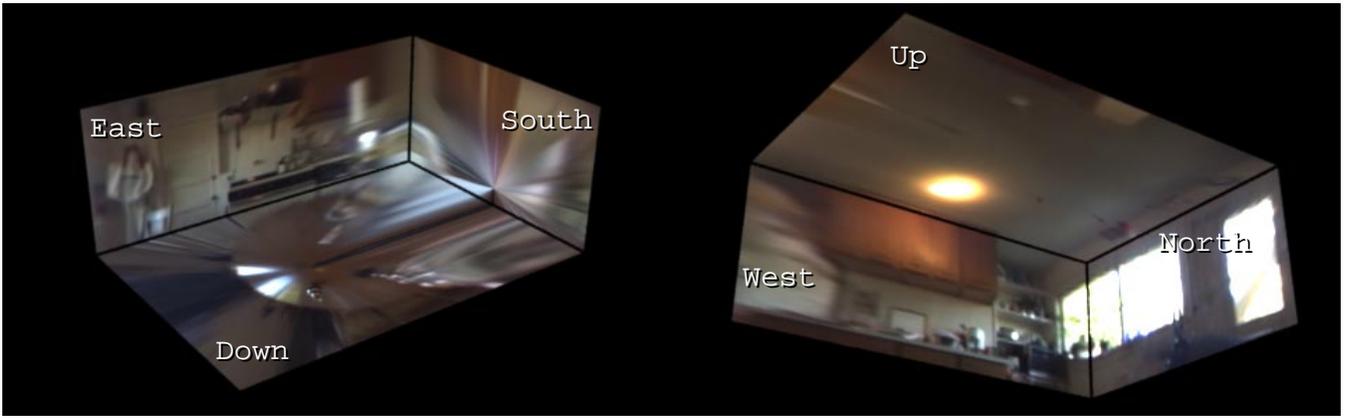


Figure 5: **A Light-Based Model** A simple light-based model of a room is constructed by mapping the image from a light probe onto a box. The box corresponds to the upper half of the room, with the bottom face of the box being coincident with the top of the table. The model contains the full dynamic range of the original scene, which is not reproduced in its entirety in this figure.

and when LS_{obj} is lighter than LS_{nobj} light is added to the background to produce reflections and caustics.

Stated more generally, the appearance of the local scene without the objects is computed with the correct reflectance characteristics lit by the correct environment, and the change in appearance due to the presence of the synthetic objects is computed with the modeled reflectance characteristics as lit by the modeled environment. While the realism of LS_{final} still benefits from having a good model of the reflectance characteristics of the local scene, the perceptual effect of small errors in albedo or specular properties is considerably reduced. Fig. 8(g) shows a final rendering in which the local environment is computed using this differential rendering technique. The objects are composited into the image directly from the LS_{obj} solution shown in Fig. 8(c).

It is important to stress that this technique can still produce arbitrarily wrong results depending on the amount of error in the estimated local scene BRDF and the inaccuracies in the light-based model of the distance scene. In fact, Err_{ls} may be larger than LS_{obj} , causing LS_{final} to be negative. An alternate approach is to compensate for the *relative* error in the appearance of the local scene: $LS_{final} = LS_b(LS_{obj}/LS_{nobj})$. Inaccuracies in the local scene BRDF will also be reflected in the objects.

In the next section we discuss techniques for estimating the BRDF of the local scene.

7 Estimating the local scene BRDF

Simulating the interaction of light between the local scene and the synthetic objects requires a model of the reflectance characteristics of the local scene. Considerable recent work [32, 20, 8, 27] has presented methods for measuring the reflectance properties of materials through observation under controlled lighting configurations. Furthermore, reflectance characteristics can also be measured with commercial radiometric devices.

It would be more convenient if the local scene reflectance could be estimated directly from observation. Since the light-based model contains information about the radiance of the local scene as well as its irradiance, it actually contains information about the local scene reflectance. If we hypothesize reflectance characteristics for the local scene, we can illuminate the local scene with its known irradiance from the light-based model. If our hypothesis is correct, then the appearance should be consistent with the measured appearance. This suggests the following iterative method for recovering the reflectance properties of the local scene:

1. Assume a reflectance model for the local scene (e.g. diffuse only, diffuse + specular, metallic, or arbitrary BRDF, including

spatial variation)

2. Choose approximate initial values for the parameters of the reflectance model
3. Compute a global illumination solution for the local scene with the current parameters using the observed lighting configuration or configurations.
4. Compare the appearance of the rendered local scene to its actual appearance in one or more views.
5. If the renderings are not consistent, adjust the parameters of the reflectance model and return to step 3.

Efficient methods of performing the adjustment in step 5 that exploit the properties of particular reflectance models are left as future work. However, assuming a diffuse-only model of the local scene in step 1 makes the adjustment in step 5 straightforward. We have:

$$L_{r1}(\theta_r, \phi_r) = \int_0^{2\pi} \int_0^{\pi/2} \rho_d L_i(\theta_i, \phi_i) \cos \theta_i \sin \theta_i d\theta_i d\phi_i = \rho_d \int_0^{2\pi} \int_0^{\pi/2} L_i(\theta_i, \phi_i) \cos \theta_i \sin \theta_i d\theta_i d\phi_i$$

If we initialize the local scene to be perfectly diffuse ($\rho_d = 1$) everywhere, we have:

$$L_{r2}(\theta_r, \phi_r) = \int_0^{2\pi} \int_0^{\pi/2} L_i(\theta_i, \phi_i) \cos \theta_i \sin \theta_i d\theta_i d\phi_i$$

The updated diffuse reflectance coefficient for each part of the local scene can be computed as:

$$\rho'_d = \frac{L_{r1}(\theta_r, \phi_r)}{L_{r2}(\theta_r, \phi_r)}$$

In this manner, we use the global illumination calculation to render each patch as a perfectly diffuse reflector, and compare the resulting radiance to the observed value. Dividing the two quantities yields the next estimate of the diffuse reflection coefficient ρ'_d . If there is no interreflection within the local scene, then the ρ'_d estimates will make the renderings consistent. If there is interreflection, then the algorithm should be iterated until there is convergence.

For a trichromatic image, the red, green, and blue diffuse reflectance values are computed independently. The diffuse characteristics of the background material used to produce Fig. 8(c) were

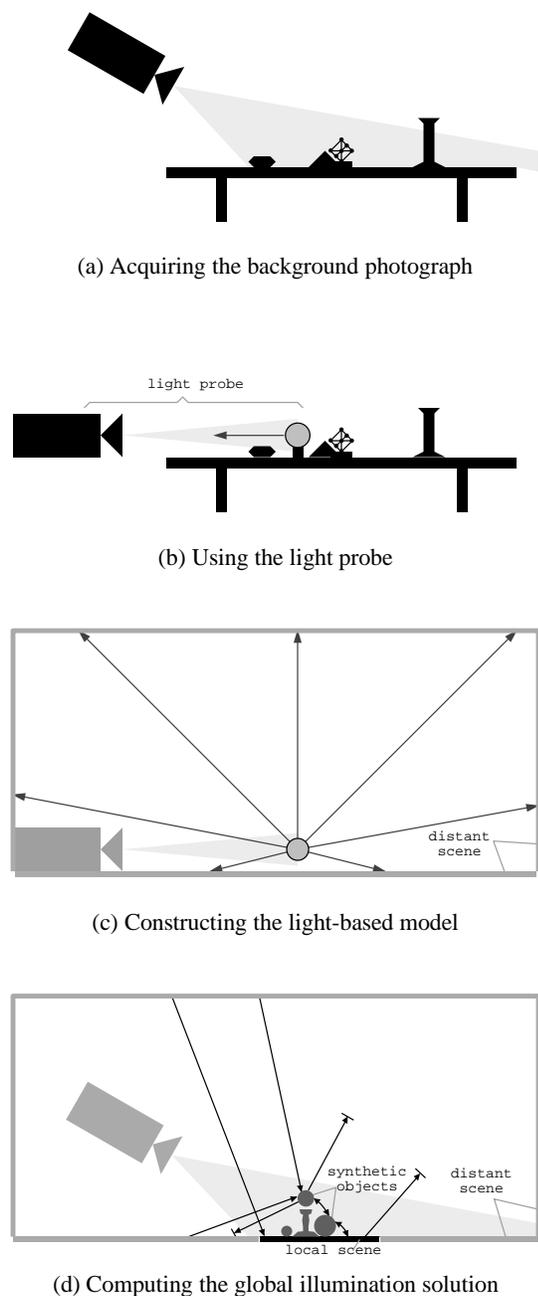


Figure 7: Using a light probe (a) The background plate of the scene (some objects on a table) is taken. (b) A light probe (in this case, the camera photographing a steel ball) records the incident radiance near the location of where the synthetic objects are to be placed. (c) A simplified light-based model of the distant scene is created as a planar surface for the table and a finite box to represent the rest of the room. The scene is texture-mapped in high dynamic range with the radiance map from the light probe. The objects on the table, which were not explicitly modeled, become projected onto the table. (d) Synthetic objects and a BRDF model of the local scene are added to the light-based model of the distant scene. A global illumination solution of this configuration is computed with light coming from the distant scene and interacting with the local scene and synthetic objects. Light reflected back to the distant scene is ignored. The results of this rendering are composited (possibly with differential rendering) into the background plate from (a) to achieve the final result.

computed using this method, although it was assumed that the entire local scene had the same diffuse reflectance.

In the standard “plastic” illumination model, just two more coefficients – those for specular intensity and roughness – need to be specified. In Fig. 8, the specular coefficients for the local scene were estimated manually based on the specular reflection of the window in the table in Fig. 2.

8 Compositing Results

Fig. 5 shows a simple light-based model of a room constructed using the panoramic radiance map from Fig. 2. The room model begins at the height of the table and continues to the ceiling; its measurements and the position of the ball within it were measured manually. The table surface is visible on the bottom face. Since the room model is finite in size, the light sources are effectively local rather than infinite. The stretching on the south wall is due to the poor sampling toward the silhouette edge of the ball.

Figs. 4 and 6 show complex arrangements of synthetic objects lit entirely by a variety of light-based models. The selection and composition of the objects in the scene was chosen to exhibit a wide variety of light interactions, including diffuse and specular reflectance, multiple soft shadows, and reflected and focused light. Each rendering was produced using the RADIANCE system with two diffuse light bounces and a relatively high density of ambient sample points.

Fig. 8(a) is a background plate image into which the synthetic objects will be rendered. In 8(b) a calibration grid was placed on the table in order to determine the camera pose relative to the scene and to the mirrored ball, which can also be seen. The poses were determined using the photogrammetric method in [10]. In 8(c), a model of the local scene as well as the synthetic objects is geometrically matched and composited onto the background image. Note that the local scene, while the same average color as the table, is readily distinguishable at its edges and because it lacks the correct variations in albedo.

Fig. 8(d) shows the results of lighting the local scene model with the light-based model of the room, without the objects. This image will be compared to 8(c) in order to determine the effect the synthetic objects have on the local scene. Fig. 8(e) is a mask image in which the white areas indicate the location of the synthetic objects. If the distant or local scene were to occlude the objects, such regions would be dark in this image.

Fig. 8(f) shows the difference between the appearance of the local scene rendered with (8(c)) and without (8(d)) the objects. For illustration purposes, the difference in radiance values have been offset so that zero difference is shown in gray. The objects have been masked out using image 8(e). This difference image encodes both the shadowing (dark areas) and reflected and focussed light (light areas) imposed on the local scene by the addition of the synthetic objects.

Fig. 8(g) shows the final result using the differential rendering method described in Section 6. The synthetic objects are copied directly from the global illumination solution 8(c) using the object mask 8(e). The effects the objects have on the local scene are included by adding the difference image 8(f) (without offset) to the background image. The remainder of the scene is copied directly from the background image 8(a). Note that in the mirror ball’s reflection, the modeled local scene can be observed without the effects of differential rendering — a limitation of the compositing technique.

In this final rendering, the synthetic objects exhibit a consistent appearance with the real objects present in the background image 8(a) in both their diffuse and specular shading, as well as the direction and coloration of their shadows. The somewhat speckled nature of the object reflections seen in the table surface is due to



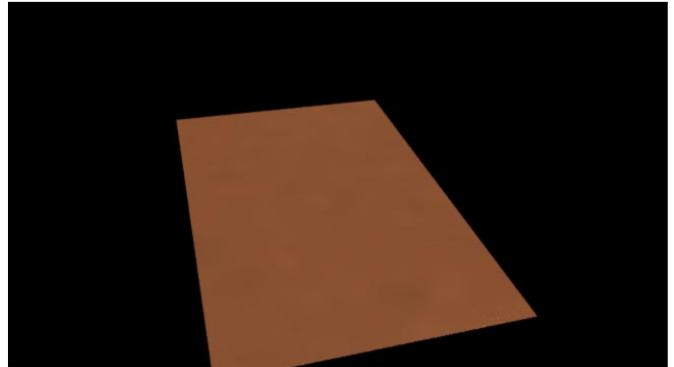
(a) Background photograph



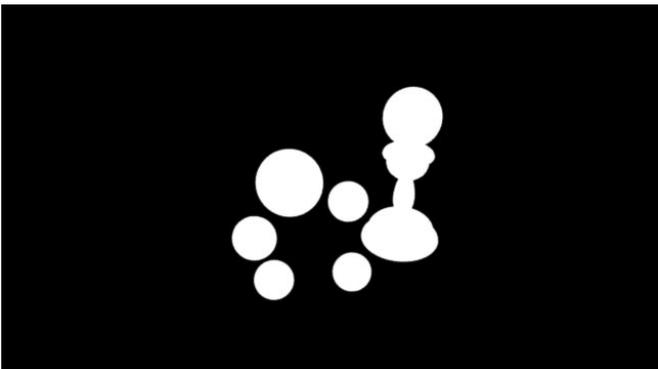
(b) Camera calibration grid and light probe



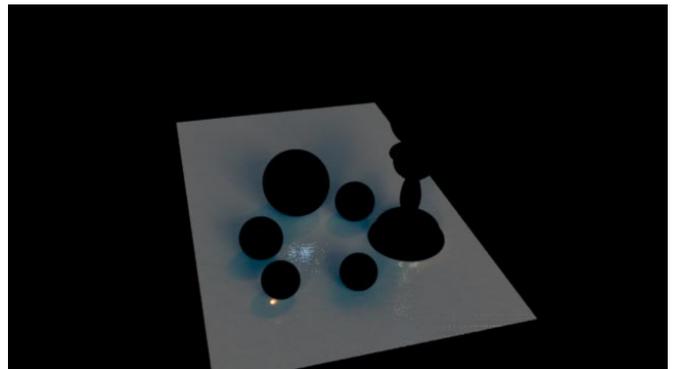
(c) Objects and local scene matched to background



(d) Local scene, without objects, lit by the model



(e) Object matte



(f) Difference in local scene between c and d



(g) Final result with differential rendering

Figure 8: Compositing synthetic objects into a real scene using a light probe and differential rendering

the stochastic nature of the particular global illumination algorithm used.

The differential rendering technique successfully eliminates the border between the local scene and the background image seen in 8(c). Note that the albedo texture of the table in the local scene area is preserved, and that a specular reflection of a background object on the table (appearing just to the left of the floating sphere) is correctly preserved in the final rendering. The local scene also exhibits reflections from the synthetic objects. A caustic from the glass ball focusing the light of the ceiling lamp onto the table is evident.

9 Future work

The method proposed here suggests a number of areas for future work. One area is to investigate methods of automatically recovering more general reflectance models for the local scene geometry, as proposed in Section 7. With such information available, the program might also be able to suggest which areas of the scene should be considered as part of the local scene and which can safely be considered distant, given the position and reflectance characteristics of the desired synthetic objects.

Some additional work could be done to allow the global illumination algorithm to compute the illumination solution more efficiently. One technique would be to have an algorithm automatically locate and identify concentrated light sources in the light-based model of the scene. With such knowledge, the algorithm could compute most of the direct illumination in a forward manner, which could dramatically increase the efficiency with which an accurate solution could be calculated. To the same end, use of the method presented in [15] to expedite the solution could be investigated. For the case of compositing moving objects into scenes, greatly increased efficiency could be obtained by adapting incremental radiosity methods to the current framework.

10 Conclusion

We have presented a general framework for adding new objects to light-based models with correct illumination. The method leverages a technique of using high dynamic range images of real scene radiance to synthetically illuminate new objects with arbitrary reflectance characteristics. We leverage this technique in a general method to simulate interplay of light between synthetic objects and the light-based environment, including shadows, reflections, and caustics. The method can be implemented with standard global illumination techniques.

For the particular case of rendering synthetic objects into real scenes (rather than general light-based models), we have presented a practical instance of the method that uses a light probe to record incident illumination in the vicinity of the synthetic objects. In addition, we have described a differential rendering technique that can convincingly render the interplay of light between objects and the local scene when only approximate reflectance information for the local scene is available. Lastly, we presented an iterative approach for determining reflectance characteristics of the local scene based on measured geometry and observed radiance in uncontrolled lighting conditions. It is our hope that the techniques presented here will be useful in practice as well as comprise a useful framework for combining material-based and light-based graphics.

Acknowledgments

The author wishes to thank Chris Bregler, David Forsyth, Jianbo Shi, Charles Ying, Steve Chenney, and Andrean Kalemis for the various forms of help and advice they provided. Special gratitude is also due to Jitendra Malik for helping make this work possible. Discussions with Michael Naimark and Steve Saunders helped motivate this work. Tim Hawkins provided extensive assistance on improving and revising this paper and provided invaluable assistance with image acquisition. Gregory Ward Larson deserves great thanks for the RADIANCE lighting simulation system and his invaluable assistance and advice in using RADIANCE in this research, for assisting with reflectance measurements, and for very helpful comments and suggestions on the paper. This research was supported by a Multidisciplinary University Research Initiative on

three dimensional direct visualization from ONR and BMDO, grant FDN00014-96-1-1200.

References

- [1] ADELSON, E. H., AND BERGEN, J. R. *Computational Models of Visual Processing*. MIT Press, Cambridge, Mass., 1991, ch. 1. The Plenoptic Function and the Elements of Early Vision.
- [2] AZARMI, M. *Optical Effects Cinematography: Its Development, Methods, and Techniques*. University Microfilms International, Ann Arbor, Michigan, 1973.
- [3] BLINN, J. F. Texture and reflection in computer generated images. *Communications of the ACM* 19, 10 (October 1976), 542–547.
- [4] CHEN, E. QuickTime VR - an image-based approach to virtual environment navigation. In *SIGGRAPH '95* (1995).
- [5] CHEN, S. E. Incremental radiosity: An extension of progressive radiosity to an interactive synthesis system. In *SIGGRAPH '90* (1990), pp. 135–144.
- [6] COHEN, M. F., CHEN, S. E., WALLACE, J. R., AND GREENBERG, D. P. A progressive refinement approach to fast radiosity image generation. In *SIGGRAPH '88* (1988), pp. 75–84.
- [7] CURLLESS, B., AND LEVOY, M. A volumetric method for building complex models from range images. In *SIGGRAPH '96* (1996), pp. 303–312.
- [8] DANA, K. J., GINNEKEN, B., NAYAR, S. K., AND KOENDERINK, J. J. Reflectance and texture of real-world surfaces. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.* (1997), pp. 151–157.
- [9] DEBEVEC, P. E., AND MALIK, J. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '97* (August 1997), pp. 369–378.
- [10] DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH '96* (August 1996), pp. 11–20.
- [11] DEBEVEC, P. E., YU, Y., AND BORSHUKOV, G. D. Efficient view-dependent image-based rendering with projective texture-mapping. Tech. Rep. UCB/CSD-98-1003, University of California at Berkeley, 1998.
- [12] DRETTAKIS, G., ROBERT, L., AND BOUGNOUX, S. Interactive common illumination for computer augmented reality. In *8th Eurographics workshop on Rendering, St. Etienne, France* (May 1997), J. Dorsey and P. Slusallek, Eds., pp. 45–57.
- [13] FIELDING, R. *The Technique of Special Effects Cinematography*. Hastings House, New York, 1968.
- [14] FOURNIER, A., GUNAWAN, A., AND ROMANZIN, C. Common illumination between real and computer generated scenes. In *Graphics Interface* (May 1993), pp. 254–262.
- [15] GERSHBEIN, R., SCHRODER, P., AND HANRAHAN, P. Textures and radiosity: Controlling emission and reflection with texture maps. In *SIGGRAPH '94* (1994).
- [16] GORAL, C. M., TORRANCE, K. E., GREENBERG, D. P., AND BATTAILE, B. Modeling the interaction of light between diffuse surfaces. In *SIGGRAPH '84* (1984), pp. 213–222.
- [17] GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. F. The Lumigraph. In *SIGGRAPH '96* (1996), pp. 43–54.
- [18] HECKBERT, P. S. Survey of texture mapping. *IEEE Computer Graphics and Applications* 6, 11 (November 1986), 56–67.
- [19] KAJIYA, J. The rendering equation. In *SIGGRAPH '86* (1986), pp. 143–150.
- [20] KARNER, K. F., MAYER, H., AND GERVAUTZ, M. An image based measurement system for anisotropic reflection. In *EUROGRAPHICS Annual Conference Proceedings* (1996).
- [21] KOENDERINK, J. J., AND VAN DOORN, A. J. Illuminance texture due to surface mesostructure. *J. Opt. Soc. Am.* 13, 3 (1996).
- [22] LAVEAU, S., AND FAUGERAS, O. 3-D scene representation as a collection of images. In *Proceedings of 12th International Conference on Pattern Recognition* (1994), vol. 1, pp. 689–691.
- [23] LEVOY, M., AND HANRAHAN, P. Light field rendering. In *SIGGRAPH '96* (1996), pp. 31–42.
- [24] McMILLAN, L., AND BISHOP, G. Plenoptic Modeling: An image-based rendering system. In *SIGGRAPH '95* (1995).
- [25] NAKAMAE, E., HARADA, K., AND ISHIZAKI, T. A montage method: The overlaying of the computer generated images onto a background photograph. In *SIGGRAPH '86* (1986), pp. 207–214.
- [26] PORTER, T., AND DUFF, T. Compositing digital images. In *SIGGRAPH 84* (July 1984), pp. 253–259.
- [27] SATO, Y., WHEELER, M. D., AND IKEUCHI, K. Object shape and reflectance modeling from observation. In *SIGGRAPH '97* (1997), pp. 379–387.
- [28] SMITH, T. G. *Industrial Light and Magic: The Art of Special Effects*. Ballantine Books, New York, 1986.
- [29] SZELISKI, R. Image mosaicing for tele-reality applications. In *IEEE Computer Graphics and Applications* (1996).
- [30] TURK, G., AND LEVOY, M. Zipped polygon meshes from range images. In *SIGGRAPH '94* (1994), pp. 311–318.
- [31] VEACH, E., AND GUIBAS, L. J. Metropolis light transport. In *SIGGRAPH '97* (August 1997), pp. 65–76.
- [32] WARD, G. J. Measuring and modeling anisotropic reflection. In *SIGGRAPH '92* (July 1992), pp. 265–272.
- [33] WARD, G. J. The radiance lighting simulation and rendering system. In *SIGGRAPH '94* (July 1994), pp. 459–472.
- [34] WATANABE, M., AND NAYAR, S. K. Telecentric optics for computational vision. In *Proceedings of Image Understanding Workshop (IUW 96)* (February 1996).
- [35] Y.CHEN, AND MEDIONI, G. Object modeling from multiple range images. *Image and Vision Computing* 10, 3 (April 1992), 145–155.

Acquiring the Reflectance Field of a Human Face

Paul Debevec[†] Tim Hawkins[†] Chris Tchou[†] Haarm-Pieter Duiker[†] Westley Sarokin[†]
and Mark Sagar[‡]

[†]University of California at Berkeley¹ [‡]LifeF/X, Inc.

ABSTRACT

We present a method to acquire the reflectance field of a human face and use these measurements to render the face under arbitrary changes in lighting and viewpoint. We first acquire images of the face from a small set of viewpoints under a dense sampling of incident illumination directions using a light stage. We then construct a reflectance function image for each observed image pixel from its values over the space of illumination directions. From the reflectance functions, we can directly generate images of the face from the original viewpoints in any form of sampled or computed illumination. To change the viewpoint, we use a model of skin reflectance to estimate the appearance of the reflectance functions for novel viewpoints. We demonstrate the technique with synthetic renderings of a person's face under novel illumination and viewpoints.

Categories and subject descriptors: I.2.10 [Artificial Intelligence]: Vision and Scene Understanding - *intensity, color, photometry and thresholding*; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - *color, shading, shadowing, and texture*; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - *radiosity*; I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture - *radiometry, reflectance, scanning*; I.4.8 [Image Processing]: Scene Analysis - *photometry, range data, sensor fusion*. **Additional Key Words and Phrases:** facial animation; image-based modeling, rendering, and lighting.

1 Introduction

Creating realistic renderings of human faces has been an endeavor in computer graphics for nearly three decades [28] and remains a subject of current interest. It is a challenging problem due to the complex and individual shape of the face, the subtle and spatially varying reflectance properties of skin, and the complex deformations of the face during movement. Compounding the problem, viewers are extremely sensitive to the appearance of other people's faces.

Recent work has provided solutions to the problems of geometrically modeling and animating faces. 3D photography techniques, such as the Cyberware scanner, can acquire accurate geometric

models of individual faces. Work to animate facial expressions through morphing [2, 4, 29], performance-driven animation [38], motion capture [14], and physics-based simulation [34, 20, 30] has produced examples of realistic facial motion.

An outstanding problem is the lack of a method for capturing the spatially varying reflectance characteristics of the human face. The traditional approach of texture-mapping a photograph of a face onto a geometric model usually fails to appear realistic under changes in lighting, viewpoint, and expression. The problem is that the reflectance properties of the face are complex: skin reflects light both diffusely and specularly, and both of these reflection components are spatially varying. Recently, skin reflectance has been modeled using Monte Carlo simulation [16], and several aggregate reflectance descriptions have been recorded from real people [22], but there has not yet been a method of accurately rendering the complexities of an individual's facial reflectance under arbitrary changes of lighting and viewpoint.

In this paper we develop a method to render faces under arbitrary changes in lighting and viewing direction based on recorded imagery. The central device in our technique is a *light stage* (Fig. 2) which illuminates the subject from a dense sampling of directions of incident illumination. During this time the subject's appearance is recorded from different angles by stationary video cameras.

From this illumination data, we can immediately render the subject's face from the original viewpoints under any incident field of illumination by computing linear combinations of the original images. Because of the additive nature of light [5], this correctly reproduces all of the effects of diffuse and specular reflection as well as interreflections between parts of the face. We demonstrate this technique by rendering faces in various forms of natural illumination captured in real-world environments, and discuss how this process can be performed directly from compressed images.

In the second part of this paper we present a technique to extrapolate a complete reflectance field from the acquired data which allows us to render the face from novel viewpoints. For this acquire a geometric model of the face through structured lighting, which allows us to project the appearance from the original viewpoints onto the geometry to render from novel viewpoints. However, re-rendering directly from such projected images does not reproduce view-dependent reflection from the face; most notably, the specular components need to shift position according to the rendered viewpoint.

To reproduce these view-dependent effects, we use a skin reflectance model to extrapolate the reflectance observed by the cameras to that which would be observed from novel viewpoints. The model is motivated by a set of in-plane reflectance measurements of a patch of skin using polarizers on the light and the camera to separate the reflection components. This model allows us to separate the specular and sub-surface reflection components of the light stage data using chromaticity analysis, and then to transform each reflectance component into how it would appear from a novel viewpoint. Using this technique, we can realistically render the face from arbitrary viewpoints and in arbitrary lighting.

The rest of this paper is organized as follows. In the next section we review related work and discuss the reflectance field. In Sec-

¹Computer Science Division, University of California at Berkeley. Email: {debevec,tsh,ctchou,duiker,wsarokin}@cs.berkeley.edu, msagar@lifefx.com. For more information see <http://www.debevec.org/>

tion 3 we describe the light stage and how we synthesize physically correct images of the subject under arbitrary illumination. In Section 4 we develop a model of skin reflectance and use it to render the face from novel viewpoints under arbitrary illumination. We discuss future work in Section 5 and conclude in Section 6.

2 Background and Related Work

In this section we give an overview of related work in the areas of facial modeling and animation, reflectometry, and image-based modeling and rendering. We conclude with a description of the reflectance field.

Facial Modeling and Animation Since the earliest work in facial modeling and animation [28], generating realistic faces has been a central goal. 3D photography techniques for acquiring facial geometry, such as the laser-triangulation based scanners made by Cyberware, have been a helpful development. Such techniques often also photograph a texture map for the face at the time of the scan, which can be projected onto the face to produce renderings. However, using such texture maps usually falls short of producing photorealistic renderings since the map is illumination-dependent and does not capture directionally varying reflectance properties. Other work estimates facial models directly from images: [11, 29, 3] recover geometry by fitting morphable facial models; [11, 3] use the models to estimate albedo maps but do not consider specular properties. [29] produces view-dependent reflectance under the original illumination conditions through view-dependent texture mapping [10].

Several techniques have been used to animate facial models; [2, 4, 29] blend between images in different expressions to produce intermediate expressions. [38, 14] use the captured facial motion of a real actor to drive the performance of a synthetic one. Physics-based simulation techniques [34, 30, 40, 20] have helped animate the complex deformations of a face in its different expressions.

Reflectometry Reflectometry is the measurement of how materials reflect light, or, more specifically, how they transform incident illumination into radiant illumination. This transformation can be described by the four-dimensional bi-directional reflectance distribution function, or BRDF, of the material measured [25]. Several efforts have been made to represent common BRDFs as parameterized functions called *reflectance models* [35, 6, 37, 27, 19].

Hanrahan and Krueger [16] developed a parameterized model for reflection from layered surfaces due to subsurface scattering, with human skin as a specific case of their model. Their model of skin reflectance was motivated by the optical properties of its surface, epidermal, and dermal layers [36]. Each layer was given several parameters according to its scattering properties and pigmentation, and a Monte Carlo simulation of the paths light might take through the skin surfaces produced renderings exhibiting a variety of qualitatively skin-like reflectance properties. The authors selected the reflectance properties manually, rather than acquiring them from a particular individual. The authors also simulated a uniform layer of oil over the face to produce specular reflection; in our work we acquire a reflectance model that reproduces the varying diffuse and specular properties over the skin.

Much work has been done to estimate reflectance properties of surfaces based on images taken under known lighting. [37] and [17] presented techniques and apparatus for measuring anisotropic reflectance of material samples; [7] applied reflectometry techniques to the domain of textured objects. In our work, we leverage being able to separate reflection into diffuse and specular components. This separation can be done through colorspace analysis [31] as well as a combined analysis of the color and polarization of the reflected light [24]; in our work we make use of both color and polarization. [32] used object geometry and varying light directions to derive diffuse and specular parameters for a coffee mug; [41]

used an inverse radiosity method to account for mutual illumination in estimating spatially varying diffuse and piecewise constant specular properties within a room.

Marschner, Westin, Lafortune, Torrance, and Greenberg [22] recently acquired the first experimental measurements of living human facial reflectance in the visible spectrum. The authors photographed the forehead of their subjects under constant point-source illumination and twenty viewing directions, and used the curvature of the forehead to obtain a dense set of BRDF samples. From these measurements, they derived a non-parametric isotropic BRDF representing the average reflectance properties of the surface of the forehead. In our work, we have chosen the goal of reproducing the spatially varying reflectance properties across the surface of the face; as a result, we sacrifice the generality of measuring a full BRDF at each surface point and use models of specular and diffuse reflectance to extrapolate the appearance to novel viewpoints.

Image-Based Modeling and Rendering In our work we leverage several principles explored in recent work in image-based modeling and rendering. [26, 15] showed how correct views of a scene under different lighting conditions can be created by summing images of the scene under a set of basis lighting conditions; [39] applied such a technique to create light fields [21, 13] with controllable illumination. [42] showed that by illuminating a shiny or refractive object with a set of coded lighting patterns, it could be correctly composited over an arbitrary background by determining the direction and spread of the reflected and refracted rays. [8] presented a technique for capturing images of real-world illumination and using this lighting to illuminate synthetic objects; in this paper we use such image-based lighting to illuminate real faces.

2.1 Definition of the Reflectance Field

The light field [12, 21], plenoptic function [1], and lumigraph [13] all describe the presence of light within space. Ignoring wavelength and fixing time, this is a five dimensional function of the form $P = P(x, y, z, \theta, \phi)$. The function represents the radiance leaving point (x, y, z) in the direction (θ, ϕ) .

[21, 13] observed that when the viewer is moving within unoccluded space, the light field can be described by a four-dimensional function. We can characterize this function as $P' = P'(u, v, \theta, \phi)$, where (u, v) is a point on a closed surface A and (θ, ϕ) is a direction as before. A light field parameterized in this form induces a five-dimensional light field in the space outside of A : if we follow the ray beginning at (x, y, z) in the direction of (θ, ϕ) until it intersects A at (u, v) , we have $P(x, y, z, \theta, \phi) = P'(u, v, \theta, \phi)$. In an example from [21] A was chosen to be a cube surrounding the object; in an example from [13] A was chosen to be the visual hull of the object. We can also consider the viewer to be inside of A observing illumination arriving from outside of A as shown in [21].

Images generated from a light field can have any viewing position and direction, but they always show the scene under the same lighting. In general, each field of incident illumination on A will induce a different field of radiant illumination from A . We can represent the radiant light field from A under every possible incident field of illumination as an eight-dimensional *reflectance field*:

$$R = R(R_i; R_r) = R(u_i, v_i, \theta_i, \phi_i; u_r, v_r, \theta_r, \phi_r) \quad (1)$$

Here, $R_i(u_i, v_i, \theta_i, \phi_i)$ represents the incident light field arriving at A and $R_r(u_r, v_r, \theta_r, \phi_r)$ represents the radiant light field leaving A (see Figure 1(a)). Except that we do not presume A to be coincident with a physical surface, the reflectance field is equivalent to the bidirectional scattering-surface reflectance distribution function S , or BSSRDF, described in Nicodemus et al. [25]. Paraphrasing [25], this function “provides a way of quantitatively expressing the connection between reflected flux leaving (u_r, v_r) in

a given direction and the flux incident at (u_i, v_i) in another given direction.”

In this work we are interested in acquiring reflectance fields of real objects, in particular human faces. A direct method to acquire the reflectance field of a real object would be to acquire a set of light fields of an object $R_r(u_r, v_r, \theta_r, \phi_r)$ for a dense sampling of incident beams of illumination from direction (θ_i, ϕ_i) arriving at the surface A at (u_i, v_i) . However, recording a four dimensional light field for every possible incident ray of light would require a ponderous amount of acquisition time and storage. Instead, in this work we acquire only *non-local reflectance fields* where the incident illumination field originates far away from A so that $R_i(u_i, v_i, \theta_i, \phi_i) = R_i(u'_i, v'_i, \theta_i, \phi_i)$ for all u_i, v_i, u'_i, v'_i . Thus a non-local reflectance field can be represented as $R' = R'(\theta_i, \phi_i; u_r, v_r, \theta_r, \phi_r)$. This reduces the representation to six dimensions, and is useful for representing objects which are some distance from the rest of the scene. In Section 3.4 we discuss using a non-local reflectance field to produce local illumination effects.

In this work we extrapolate the complete field of radiant illumination from data acquired from a sparse set of camera positions (Section 3) and choose the surface A to be a scanned model of the face (Figure 1(b)), yielding a surface reflectance field analogous to a surface light field [23]. A model of skin reflectance properties is used to synthesize views from arbitrary viewpoints (Section 4).

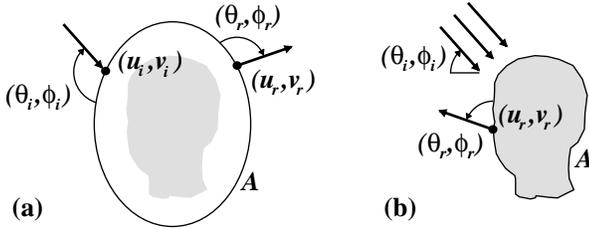


Figure 1: **The Reflectance Field** (a) describes how a volume of space enclosed by a surface A transforms an incident field of illumination $R_i(u_i, v_i, \theta_i, \phi_i)$ into a radiant field of illumination $R_r(u_r, v_r, \theta_r, \phi_r)$. In this paper, we acquire a non-local reflectance field (b) in which the incident illumination consists solely of directional illumination (θ_i, ϕ_i) . We choose A to be coincident with the surface of the face, yielding a surface reflectance field which allows us to extrapolate the radiant light field $R_r(u_r, v_r, \theta_r, \phi_r)$ from a sparse set of viewpoints.

3 Re-illuminating Faces

The goal of our work is to capture models of faces that can be rendered realistically under any illumination, from any angle, and, eventually, with any sort of animated expression. The data that we use to derive our models is a sparse set of viewpoints taken under a dense set of lighting directions. In this section, we describe the acquisition process, how we transform each facial pixel location into a reflectance function, and how we use this representation to render the face from the original viewpoints under any novel form of illumination. In the following section we will describe how to render the face from new viewpoints.

3.1 The Light Stage

The light stage used to acquire the set of images is shown in Fig. 2. The subject sits in a chair which has a headrest to help keep his or her head still during the capture process. Two digital video cameras view the head from a distance of approximately three meters; each captures a view of the left or the right side of the face. A spotlight, calibrated to produce an even field of illumination across the



Figure 2: **The Light Stage** consists of a two-axis rotation system and a directional light source. The outer black bar θ is rotated about the central vertical axis and the inner bar ϕ is lowered one step for each θ rotation. Video cameras placed outside the stage record the face’s appearance from the left and right under this complete set of illumination directions, taking slightly over a minute to record. The axes are operated manually by cords and an electronic audio signal triggered by the θ axis registers the video to the illumination directions. The inset shows a long-exposure photograph of the light stage in operation.

subject’s head, is affixed at a radius of 1.5 meters on a two-axis rotation mechanism that positions the light at any azimuth θ and any inclination ϕ . In operation, the light is spun about the θ axis continuously at approximately 25 rpm and lowered along the ϕ axis by $\frac{180}{32}$ degrees per revolution of θ (the cord controlling the ϕ axis is marked at these increments). The cameras, which are calibrated for their flat-field response and intensity response curve, capture frames continuously at 30 frames per second which yields 64 divisions of θ and 32 divisions of ϕ in approximately one minute, during which our subjects are usually capable of remaining still. A future version could employ high-speed cameras running at 250 to 1000 frames per second to lower the capture time to a few seconds. Some source images acquired with the apparatus are shown in Fig. 5.

3.2 Constructing Reflectance Functions

For each pixel location (x, y) in each camera, we observe that location on the face illuminated from 64×32 directions of θ and ϕ . From each pixel we form a slice of the reflectance field called

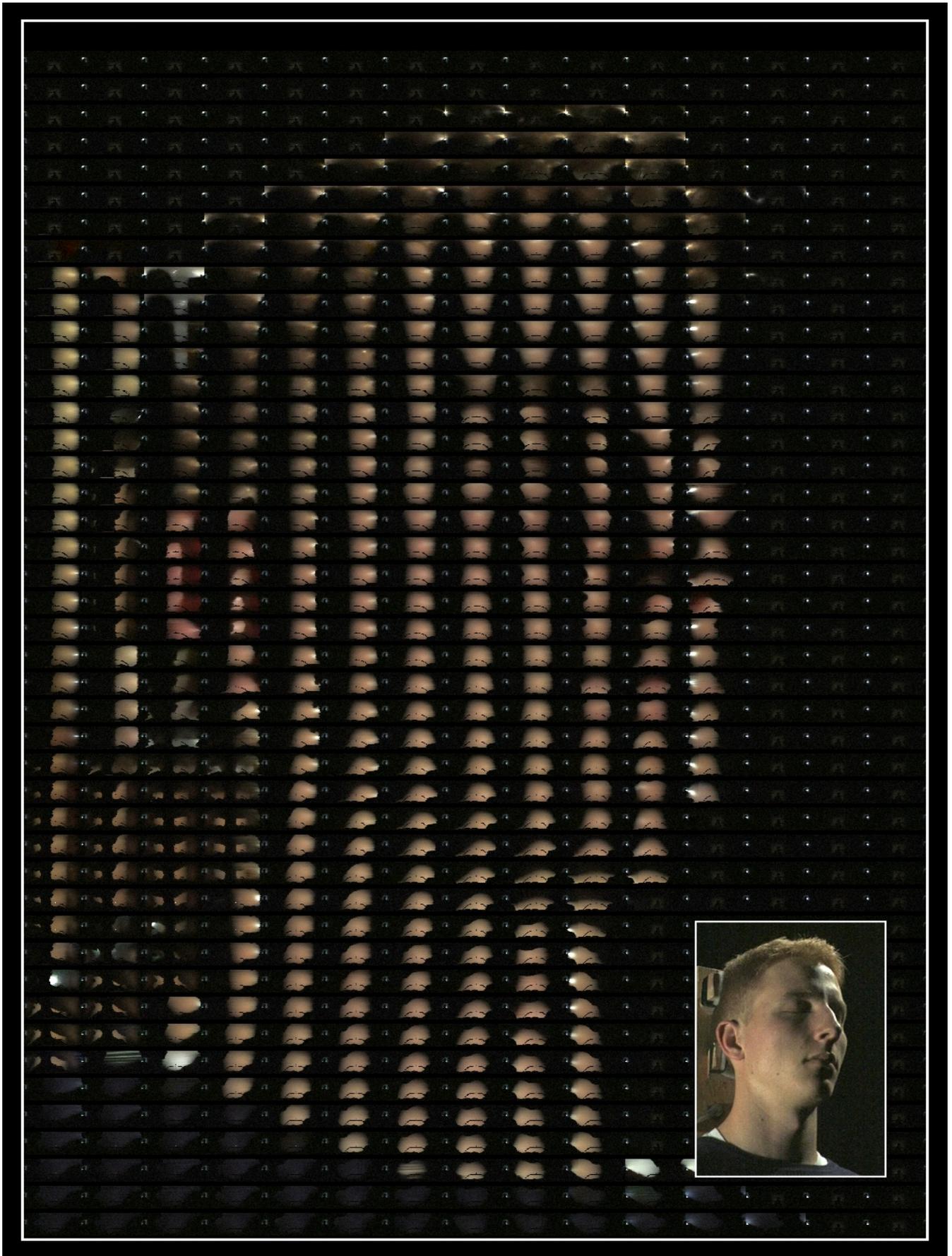


Figure 3: **Reflectance Functions for a Face** This mosaic is formed from the reflectance functions of a 15×44 sampling of pixels from the original 480×720 image data. Each 64×32 reflectance function consists of the corresponding pixel location's appearance under two thousand lighting directions distributed throughout the sphere. The inset shows the same view of the face under a combination of three lighting directions. The functions have been brightened by a factor of four from the original data.

a reflectance function $R_{xy}(\theta, \phi)$ corresponding to the ray through that pixel. Note that we are using the term “reflectance” loosely as true reflectance divides out the effect of the foreshortening of incident light. However, since the surface normal is unknown, we do not make this correction. If we let the pixel value at location (x, y) in the image with illumination direction (θ, ϕ) be represented as $L_{\theta, \phi}(x, y)$, then we have simply:

$$R_{xy}(\theta, \phi) = L_{\theta, \phi}(x, y) \quad (2)$$

Fig. 3 shows a mosaic of reflectance functions for a particular viewpoint of the face. Four of these mosaics are examined in detail in Fig. 4. The reflectance functions exhibit and encode the effects of diffuse reflection, specular reflection, self-shadowing, translucency, mutual illumination, and subsurface scattering.

3.3 Re-illuminating the Face

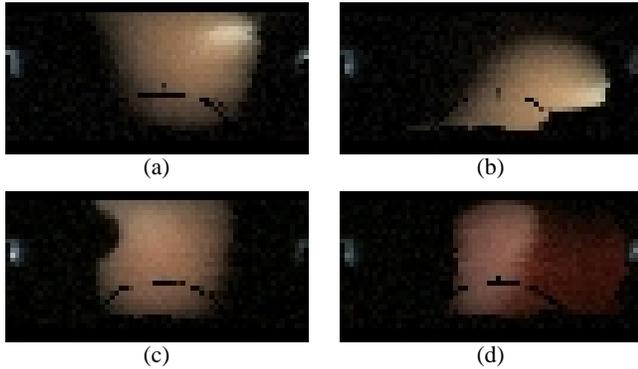


Figure 4: **A Sampling of Facial Reflectance Functions** The above reflectance functions appear in the mosaic of Fig. 3. The middle of each function corresponds to the pixel being illuminated from the direction of the camera; as one moves within the reflectance function the light direction moves in the same manner. Reflectance function (a) is taken from the forehead toward the right of the image, and exhibits a noticeable specular lobe as well as an unoccluded diffuse lobe. (b) from the right of the underside of the jaw exhibits a weaker specular component and some self-shadowing at lower lighting angles caused by the shoulder blocking the light source. (c) from the subject’s cheek to the right and below the nose exhibits a mild specular reflection and shadowing due to the nose in the upper left. (d) sampled from a pixel inside the pinna of the ear exhibits illumination from diffuse reflection and from light scattering through the tissue when illuminated from behind. Each function exhibits a thin black curve in its lower half where the phi axis bar occasionally obscures the view of the face, and a bright spot due to lens flare where the light points into the camera. These regions appear in the same places across images and are ignored in the lighting analysis.

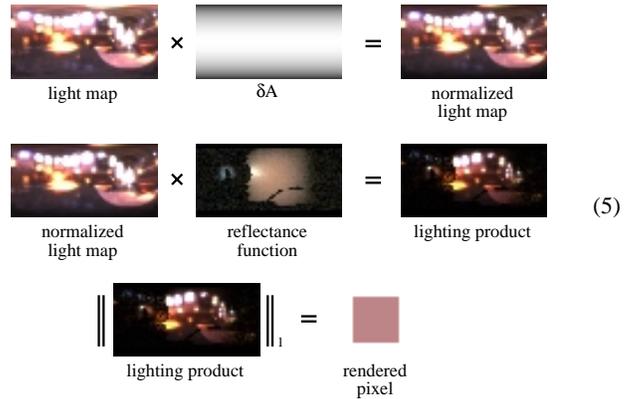
Suppose that we wish to generate an image of the face in a novel form of illumination. Since each $R_{xy}(\theta, \phi)$ represents how much light is reflected toward the camera by pixel (x, y) as a result of illumination from direction (θ, ϕ) , and since light is additive, we can compute an image of the face $\hat{L}(x, y)$ under any combination of the original light sources $L_i(\theta, \phi)$ as follows:

$$\hat{L}(x, y) = \sum_{\theta, \phi} R_{xy}(\theta, \phi) L_i(\theta, \phi) \quad (3)$$

Each color channel is computed separately using the above equation. Since the light sources densely sample the viewing sphere, we can represent any form of sampled incident illumination using this basis. In this case, it is necessary to consider the solid angle δA covered by each of the original illumination directions:

$$\hat{L}(x, y) = \sum_{\theta, \phi} R_{xy}(\theta, \phi) L_i(\theta, \phi) \delta A(\theta, \phi) \quad (4)$$

For our data, $\delta A(\theta, \phi) = \sin \phi$; the light stage records more samples per solid angle near the poles than at the equator. Equation 5 shows the computation of Equation 4 graphically. First, the map of incident illumination (filtered down to the 64×32 (θ, ϕ) space) is normalized by the map of $\delta A(\theta, \phi)$. Then, the resulting map is multiplied by the pixel’s reflectance function. Finally, the pixel values of this product are summed to compute the re-illuminated pixel value. These equations assume the light stage’s light source is white and has unit radiance; in practice we normalize the reflectance functions based on the light source color. Figure 6 shows a face synthetically illuminated with several forms of sampled and synthetic illumination using this technique.



Writing the re-illumination equation of Equation 4 as the sum of the product of two 64×32 images allows us to gain efficiency in both storage and computation using the techniques presented by Smith and Rowe [33] by computing the product directly on JPEG-compressed versions of the images. This can reduce both storage and computation by a factor of twenty while maintaining good image quality.

3.4 Discussion

Since each rendered image can also be represented as a linear combination of the original images, all of the proper effects of non-diffuse reflectance, mutual illumination, translucency, and subsurface scattering are preserved, as noted in [26]. The 64×32 set of illumination directions used is somewhat coarse; however, the reflectance functions are generally not aliased at this resolution, which implies that when the light maps are also properly filtered down to 64×32 there will be no aliasing in the resulting renderings. The place where the reflectance functions do become aliased is where there is self-shadowing; the expected result of this is that one would see somewhat stairstepped shadows in harsh lighting situations. Such effects could be smoothed by using an area light source to illuminate the subject.

Since this technique captures slices of a non-local reflectance field, it does not tell us how to render a person under dappled light or in partial shadow. A technique that will in many cases produce reasonable results is to illuminate different pixels of the face using different models of incident illumination; however, this will no longer produce physically valid images because changes to the indirect illumination are not considered. As an example, consider rendering a face with a shaft of light hitting just below the eye. In reality, the light below the eye would throw indirect illumination on the underside of the brow and the side of the nose; this technique would not capture this effect.

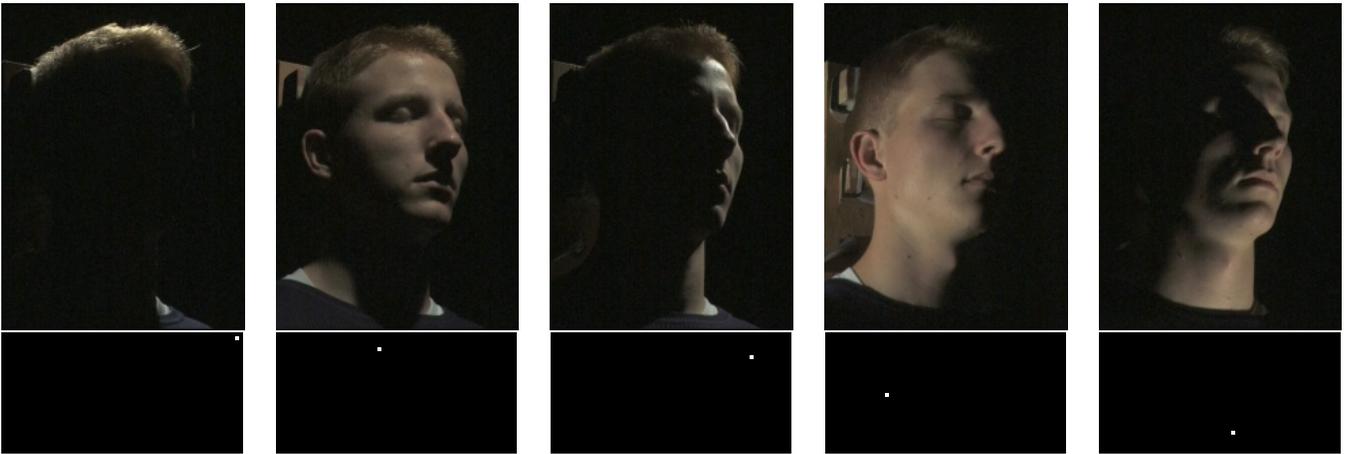


Figure 5: **Light Stage Images** Above are five of the 2048 images taken by one camera during a run of the light stage. The pixel values of each location on the face under the 2048 illumination directions are combined to produce the mosaic images in Fig. 3. Below each image is the impulse light map that would generate it.

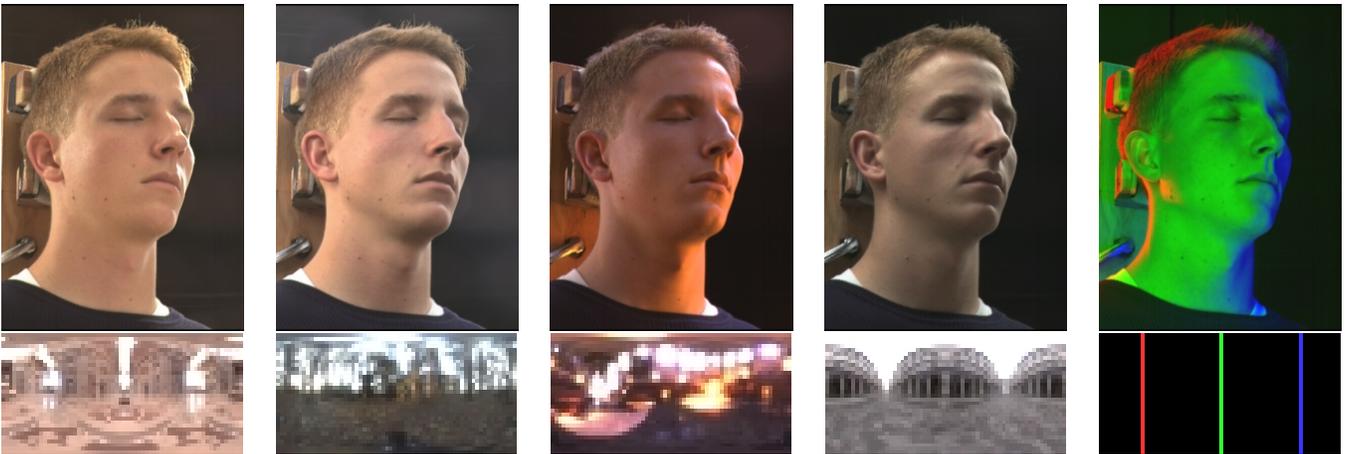


Figure 6: **Face Rendered under Sampled Illumination** Each of the above images shows the face synthetically illuminated with novel lighting, with the corresponding light map shown below. Each image is created by taking the dot product of each pixel's reflectance function with the light map. The first four illumination environments are light probe measurements acquired from real-world illumination (see [8]) recorded as omnidirectional high dynamic range images; the rightmost lighting environment is a synthetic test case.

A person's clothing reflects indirect light back onto the face, and our capture technique reproduces the person's appearance in whatever clothing they were wearing during the capture session. If we need to change the color of the person's clothing (for example, to place a costume on a virtual actor), we can record the subject twice, once wearing white clothing and once with black clothing. Subtracting the second image from the first yields an image of the indirect illumination from the clothing, which can then be tinted to any desired color and added back in to the image taken with the black clothing; this process is illustrated in Figure 7.

By recording the light stage images in high dynamic range [9] and using the process of environment matting [42], we can apply this technique to translucent and refractive objects and reproduce the appearance of the environment in the background; this process is described in the Appendix.

4 Changing the Viewpoint

In this section we describe our technique to extrapolate complete reflectance fields from the reflectance field slices acquired in Section 3, allowing us to render the face from arbitrary viewpoints as well as under arbitrary illumination. In our capture technique, we

observe the face under a dense set of illumination conditions but from only a small set of viewpoints. To render the face from a novel viewpoint, we must resynthesize the reflectance functions to appear as they would from the new viewpoint.

To accomplish this, we make use of a skin reflectance model which we introduce in Section 4.1. This model is used to guide the shifting and scaling of measured reflectance function values as the viewpoint changes. As such, our technique guarantees that the resynthesized reflectance function will agree exactly with the measured data if the novel viewpoint is the same as the viewpoint for data capture.

The resynthesis technique requires that our reflectance functions be decomposed into specular and diffuse (subsurface) components. Section 4.2 describes this separation process. Section 4.3 describes the re-synthesis of a reflectance function for a new viewpoint. Section 4.4 discusses the technique in the context of shadowing and mutual illumination effects. Section 4.5 explains the method used to produce renderings of the entire face using resynthesized reflectance functions.

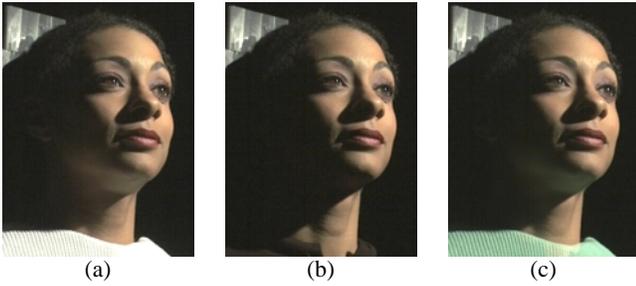


Figure 7: **Modeling indirect light from clothing** Indirect reflectance from the subject's clothing can be modeled by recording the subject wearing both white (a) and black (b) clothing (we drape the white clothing on the subject and pull it away to reveal the black clothing.) (a) exhibits indirect lighting on the neck and beneath the chin and nose. Correct renderings of the person wearing any color clothing can be created by adding a tinted version of (a) minus (b) to (b). Using this method, (c) shows the subject with the indirect light she would receive from green clothing.

4.1 Investigating Skin Reflectance

In this section we consider the reflectance properties of skin, and describe our data-driven skin reflectance model. The model is intended to capture the behavior of skin, but could be useful for a wider class of surfaces.

Following [16], we note that the light reflected from the skin can be decomposed into two components: a specular component consisting of light immediately reflected at the index of refraction transition at the air-oil interface (see Figure 8), and a non-Lambertian diffuse component consisting of light transmitted through the air-oil interface that, after some number of subsurface scattering interactions, is transmitted from the oil layer to air.

We first investigated the general behavior of these two components. As shown in Figure 8, light which reflects specularly off the skin will maintain the polarization of the incident light; however, light which emerges from below the surface will have been depolarized by scattering interactions. Taking advantage of this fact, we can separate the reflection components by placing linear polarizers on both the light source and the camera¹. Figure 9 shows separated specular and diffuse reflection components of a face using this technique.

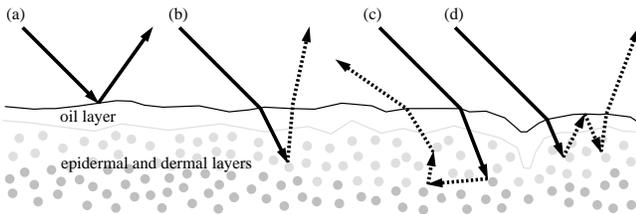


Figure 8: **Skin Reflectance** Light reflecting from skin must have reflected specularly off the surface (a) or at some point entered one or more of the scattering layers (b, c, d). If the incident light is polarized, the specularly reflected light will maintain this polarization; however, light which scatters within the surface becomes depolarized. This allows reflection components to be separated as in Figures 9 and 10.

Using this effect, we carried out an in-plane experiment to measure the specular and diffuse reflectance properties of a small patch

¹In these tests we polarize the light source vertically with respect to the plane of incidence so that the specular reflection does not become attenuated near the Brewster angle.

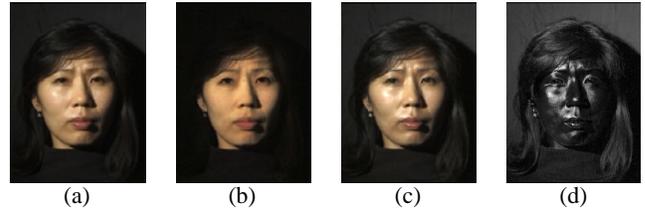


Figure 9: **Separating diffuse and specular components** can be performed by placing a linear polarizer on both the light source and the camera. (a) Normal image under point-source illumination. (b) Image of diffuse reflectance obtained by placing a vertical polarizer on the light source and a horizontal polarizer on the camera, blocking specularly reflected light. (c) Image of accentuated specular reflectance obtained by placing both polarizers vertically (half the diffusely reflected light is blocked relative to the specularly reflected light). (d) Difference of (c) and (b) yielding the specular component. The images have been scaled to appear consistent in brightness.

of skin on a person's forehead. Figure 10 shows how we adapted the light stage of Figure 2 for this purpose by placing the ϕ axis in the horizontal position and placing a vertical polarizer on the light source. We rotated the horizontal θ axis continuously while we placed a video camera aimed at our subject's vertically aligned forehead at a sampling of reflected illumination angles. The camera angles we used were $\pm(0, 22.5, 45, 60, 75, 82.5, 86.25, 89)$ degrees relative to the forehead's surface normal in order to more densely sample the illumination at grazing angles. At 89 degrees the skin area was very foreshortened so we were not able to say with certainty that the measurement we took originated only from the target area. We performed the experiment twice: once with the camera polarizer placed horizontally to block specular reflection, and once with the camera polarizer placed vertically to accentuate it. The average intensity and color of the reflected light from a 2×5 pixel area on the forehead was recorded in this set of configurations.

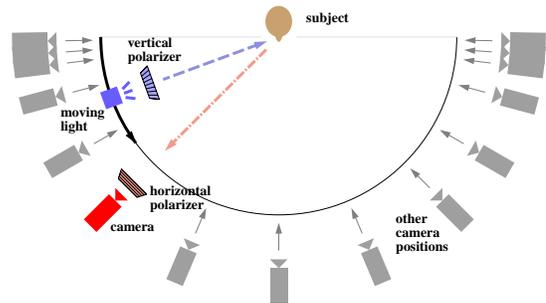


Figure 10: **Reflectometry Experiment** In this experiment, the diffuse and specular reflectance of an area of skin on the subject's forehead was recorded from sixty-four illumination directions for each of fifteen camera positions. Polarizers on the light and camera were used to separate the reflection components.

We noted two trends in the acquired reflectance data (Figure 11). First, the specular component becomes much stronger for large values of θ_i or θ_r , and exhibits off-specular reflection. To accommodate this behavior in our model, we use the microfacet-based framework introduced by Torrance and Sparrow [35]. This framework assumes geometric optics and models specular lobes as surface (Fresnel) reflection from microfacets having a Gaussian distribution of surface normals. Shadowing and masking effects between the microfacets are computed under the assumption that the microfacets form V-shaped grooves. Our model differs only in that we do not

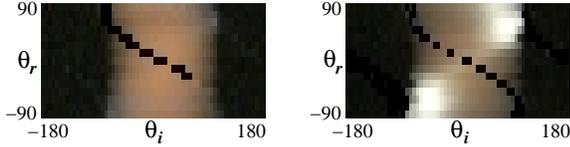


Figure 11: **Reflectometry Results** The left image shows the measured diffuse (sub-surface) component of the skin patch obtained from the experiment in Fig. 10 for incident illumination angle θ_i and viewing direction θ_r . θ_r is nonuniformly spaced at angles of $\pm(0, 22.5, 45, 60, 75, 82.5, 86.25, 89)$ degrees. Invalid measurements from the light source blocking the camera’s view are set to black. The right image shows the corresponding data for accentuated specular reflectance.

assume that the microfacet normal distribution is Gaussian; since we have measurements of the specular component for dense incident directions, we simply take the microfacet normal distribution directly from the observed data. This allows the measured specular lobe to be reproduced exactly if the viewpoint is unchanged.

The second trend in the data is a desaturation of the diffuse component for large values of θ_i and θ_r . To accommodate this, we make a minor deviation from pure Lambertian behavior, allowing the saturation of the diffuse chromaticity to ramp between two values as θ_i and θ_r vary.

Representing chromaticities as unit RGB vectors, we model the diffuse chromaticity as:

$$\text{normalize}(\vec{d}_0 + f(\theta_i, \theta_r)(\vec{d}_0 - \vec{s})) \quad (6)$$

where \vec{d}_0 is a representative diffuse chromaticity, \vec{s} is the light source chromaticity, and $f(\theta_i, \theta_r)$ is given by:

$$f(\theta_i, \theta_r) = \alpha_0(\cos \theta_i \cos \theta_r) + \alpha_1(1 - \cos \theta_i \cos \theta_r) \quad (7)$$

We recover the parameters α_0 and α_1 directly from our data for each reflectance function. This correction to the diffuse chromaticity is used for the color space separation of diffuse and specular components described in Section 4.2, and also in our reflectance function resynthesis technique described in Section 4.3.

In addition to this experiment, we also performed Monte Carlo simulations of subsurface scattering similar to those in [16]. We used two scattering layers, both with strong forward scattering, and with the lower layer having significant absorption of shorter wavelengths to simulate the presence of blood in the dermis. These simulations yielded a variation in the chromaticity of the diffuse component similar to that observed in our data.

4.2 Separating Specular and Subsurface Components

We begin by separating the specular and subsurface (diffuse) components for each pixel’s reflectance function. While we could perform this step using the polarization approach of Section 4.1, this would require two passes of the lighting rig (one for diffuse only and one that includes specular) or additional cameras. Furthermore, one of the polarizers would have to rotate in a non-trivial pattern to maintain the proper relative orientations of the polarizers when ϕ is non-horizontal. Instead, we use a color space analysis technique related to [31].

For a reflectance function RGB value $R_x y(\theta, \phi)$, we can write R as a linear combination of its diffuse color \vec{d} and its specular color \vec{s} . In reality, due to noise, interreflections, and translucency, there will also be an error component \vec{e} :

$$R = \mu_d \vec{d} + \mu_s \vec{s} + \mu_e \vec{e}$$

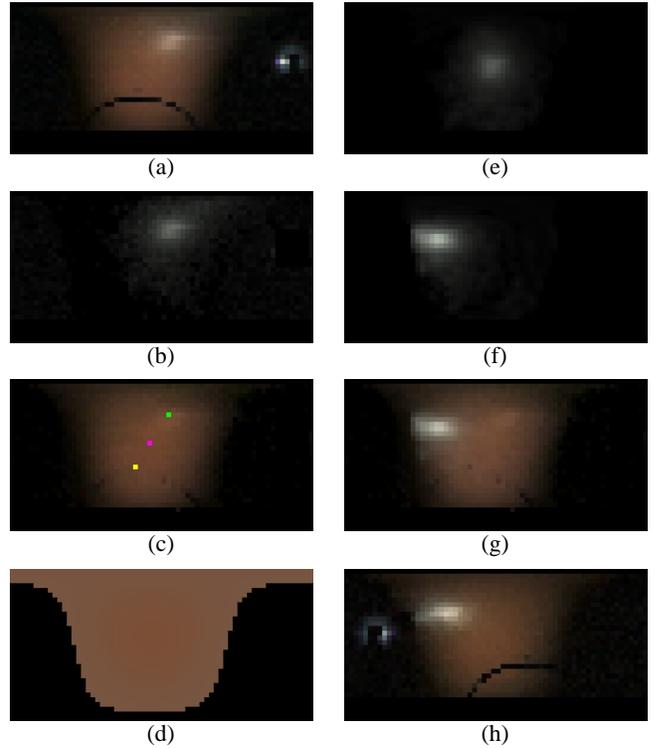


Figure 12: **Analyzing and Resynthesizing Reflectance Functions** Reflectance functions (a) can be decomposed into specular (b) and diffuse (c) components using colorspace analysis based on a model of the variation in diffuse chromaticity (d). We compute a surface normal \vec{n} based on the diffuse component (magenta dot in (c)), and a normal \vec{n}_s (coincident with \vec{n} in this case) based on the maximum (green dot) of the specular component and the known viewing direction (yellow dot). We demonstrate the resynthesis of reflectance functions for new viewpoints by resynthesizing (a), which was captured by the left camera, from the viewpoint of the right camera. We first transform the specular component to a representation independent of the original viewpoint (essentially a microfacet normal distribution) as shown in (e), then transform (e) in accordance with the new viewpoint to produce (f). The diffuse component is chrominance-shifted for the new viewpoint and added to the transformed specular component to produce the new reflectance function (g). For comparison, (h) shows the actual reflectance function (with lens flare spot and ϕ -bar shadow) from the second camera.

We choose $\vec{e} = \vec{d} \times \vec{s}$ and determine values for μ_d , μ_s , and μ_e by inverting the resulting 3×3 matrix. To form the final separation, we compute $S = \max(\mu_s, 0)\vec{s}$ and $D = R - S$ so that the sum of D and S yields the original reflectance function R .

This analysis assumes that the specular and diffuse colors are known. While we can assume that the specular component is the same color as the incident light, the diffuse color presents a more difficult problem, because it changes not only from pixel to pixel, but also within each reflectance function, as described in Section 4.1. To achieve an accurate separation, we must first estimate the diffuse chromaticity ramp.

Since we assume the diffuse chromaticity is a function f of θ_i and θ_r , we must first estimate the surface normal. For this we perform an initial rough color space separation based on a uniform diffuse chromaticity \vec{d}_0 . We derive this diffuse chromaticity by computing the median of the red-green and green-blue ratios over reflectance function values falling in a certain brightness range. We then perform a diffuse-specular separation and fit a Lambertian lobe to the diffuse component, using a coarse-to-fine direct search. This fitting yields an estimate of the surface normal.

We then find the parameters α_0 and α_1 which give the best fit to the observed chromaticities in the original unseparated reflectance function, again using a coarse-to-fine direct search. Knowing the viewpoint and the surface normal, we downweight values near the mirror angle to prevent the color ramp from being biased by strong specularities. The final separation into diffuse and specular components is computed using the fitted model of diffuse chromaticity as shown in Fig. 12.

We use the final separated diffuse component to recompute the surface normal \vec{n} , as seen in Fig. 14(b). For visualization purposes, we can also compute an estimate of the diffuse albedo $\vec{\rho}_d$ and total specular energy ρ_s , which are shown in Fig. 14(c) and (d).

4.3 Transforming Reflectance Functions to Novel Viewpoints

The process of resynthesizing a reflectance function for a novel viewpoint is illustrated in Fig. 12. The resynthesis algorithm takes the following input:

1. The diffuse reflectance function $D(\theta, \phi)$
2. The specular reflectance function $S(\theta, \phi)$
3. The surface normal \vec{n}
4. The index of refraction for surface (specular) reflection
5. The diffuse chromaticity ramp parameters α_0 and α_1
6. The original and novel view direction vectors \vec{v}_0 and \vec{v}_n

The diffuse and specular reflectance functions may optionally be transformed to a representation that does not depend on the original viewing direction, for example by transforming the functions to the form they would have if $\vec{v} = \vec{n}$. In this case, the resynthesis no longer requires the original view direction. An example of this for the specular component is shown in Fig. 12(e).

To synthesize a reflectance function from a novel viewpoint, we separately synthesize the diffuse and specular components. A sample in a specular reflectance function represents a specular response to a light source in the corresponding direction. If the view direction is known, we may consider this specular response to be a measure of the proportion of microfacets with normals oriented within some solid angle of the halfway vector between the view direction and the sample's light source direction sample. To compute a specular reflectance function from a new view direction \vec{v}_n , we compute for each light source direction \vec{l}_p the halfway vector:

$$\vec{H} = \text{normalize}(\vec{v}_n + \vec{l}_p)$$

We then find the light source direction \vec{l}_q that would have responded to microfacets near \vec{H} from the original view direction \vec{v}_0 :

$$\vec{l}_q = 2(\vec{H} \cdot \vec{v}_0)\vec{H} - \vec{v}_0$$

Letting ω_i specify a direction of incoming radiance, the Torrance-Sparrow model relates the observed radiance L to the microfacet normal distribution P as follows:

$$L_{\vec{v}} = \int \frac{PL\omega_i GF}{4 \cos \theta_r} d\omega_i \quad (8)$$

where G is a geometric attenuation factor and F is the Fresnel reflectivity. G depends on \vec{v} , \vec{l} , and \vec{n} . The expression for G is somewhat complicated, and we refer the interested reader to [35]. F is given by the Fresnel equation for unpolarized light, which can be computed from \vec{v} and \vec{l} .

Considering all quantities in (8) to be constant over the small solid angle Ω subtended by our light source, we have:

$$L_{\vec{v}} = \frac{PL_{-\vec{l}}\Omega GF}{4(\vec{v} \cdot \vec{n})}$$

Assuming the light source presents a constant $L_{-\vec{l}}\Omega$ as it moves, and recalling that the light direction \vec{l}_q is chosen to sample the same point in the microfacet normal distribution as \vec{l}_p , we can compute the new sample radiance $L_{\vec{v}_n}$ due to a light at \vec{l}_p as a function of the original radiance sample $L_{\vec{v}_0}$ due to a light at \vec{l}_q :

$$L_{\vec{v}_n} = L_{\vec{v}_0} \frac{G(\vec{v}_n, \vec{l}_p, \vec{n})F(\vec{v}_n, \vec{l}_p)(\vec{v}_0 \cdot \vec{n})}{G(\vec{v}_0, \vec{l}_q, \vec{n})F(\vec{v}_0, \vec{l}_q)(\vec{v}_n \cdot \vec{n})} \quad (9)$$

Fig. 12(f) shows a specular reflectance function synthesized using (9) for a view direction 80 degrees from the original view.

For the diffuse component we apply our diffuse chrominance ramp correction to each value in the diffuse reflectance function, first inverting the chrominance shift due to the original view direction and then applying the chrominance shift for the new view direction. The chrominance shift is computed with the recovered parameters α_0 and α_1 as in (6), using the actual sample chromaticity in place of \vec{d}_0 .

A final synthesized reflectance function consisting of the resynthesized diffuse and specular components is shown in Fig. 12(g), and is consistent with an actual reflectance function acquired from the novel viewpoint in Fig. 12(h).

4.4 Considering Shadowing and Interreflection

Since our geometry is presumed to be non-convex, we expect reflectance functions in areas not on the convex hull to exhibit global illumination effects such as shadows and interreflections. To deal with such areas, we compute a shadow map for each reflectance function. This could be done using our geometric model, but since the geometry is incomplete we instead compute the shadow map using brightness thresholding on the original reflectance function. This is demonstrated in Figure 13. We then do the analysis of Section 4.2 on the reflectance function modulated by the shadow map. This will give good results when the direct light dominates the indirect light over the non-shadowed portion of the reflectance function, a good assumption for most areas of the face.

When synthesizing a new specular reflectance function, the shadow map is used to prevent a specular lobe from appearing in shadowed directions. The converse of this effect is that when a specularity is shadowed in our original data, we are unable to recover the specular lobe. This problem could be reduced by using more cameras.

An advantage of our synthesis technique is that diffuse interreflections, and in fact all light paths terminating with a diffuse reflection, are left intact in the diffuse reflectance function and are thus reproduced without the necessity of performing the difficult steps of inverse and forward global illumination.

4.5 Creating Renderings

With the ability to resynthesize reflectance functions for new view directions, it is straightforward to render the face in arbitrary illumination from arbitrary viewpoints. We first use the technique of Section 3 to render a view of the face in the novel lighting using the modified reflectance functions. Although geometrically from the original point of view, the face is shaded as if it were viewed from the novel point of view. We then project this image onto a geometric model of the face (see Fig. 14(e)) and view the model from the novel viewpoint, yielding a rendering in which the illumination and viewpoint are consistent. In our work we use two original viewpoints, one for the left and one for the right of the face, and blend the results over the narrow region of overlap (with more cameras, view-dependent texture mapping could be used to blend between viewpoints as in [10, 29]). Renderings made with this technique are shown in Figs. 14(f),(g) and (h), and comparisons with actual photographs are shown in Fig. 15.

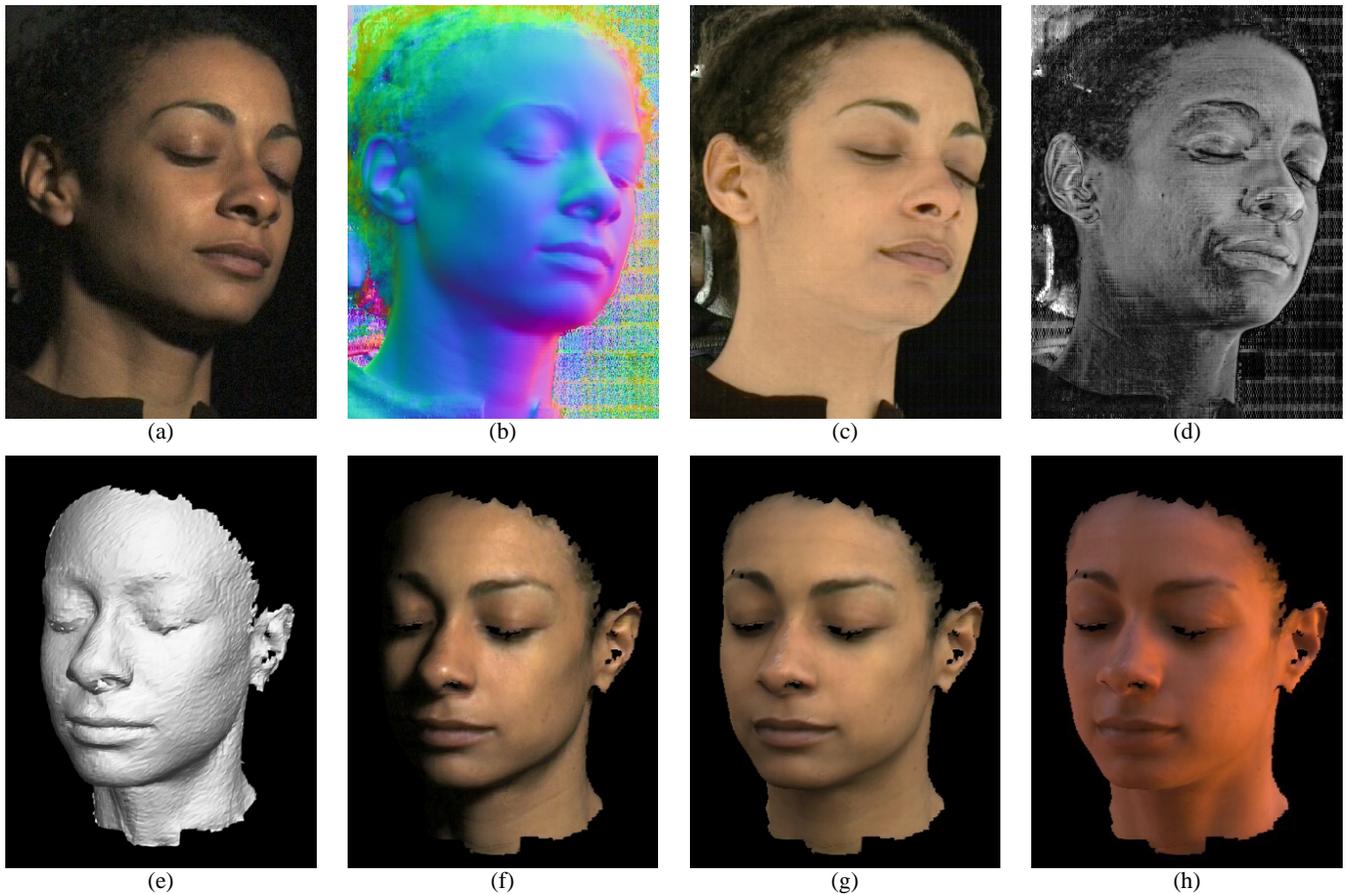


Figure 14: **Analyzing Reflectance and Changing the Viewpoint** (a) An original light stage image taken by the left camera. (b) Recovered surface normals n_d derived from the fitted diffuse reflectance lobe for each pixel; the RGB value for each pixel encodes the X, Y, and Z direction of each normal. (c) Estimated diffuse albedo ρ_d . Although not used by our rendering algorithm, such data could be used in a traditional rendering system. (d) Estimated specular energy ρ_s , also of potential use in a traditional rendering system. (e) Face geometry recovered using structured lighting. (f) Face rendered from a novel viewpoint under synthetic directional illumination. (g,h) Face rendered from a novel viewpoint under the two sampled lighting environments used in the second two renderings of Fig. 6.

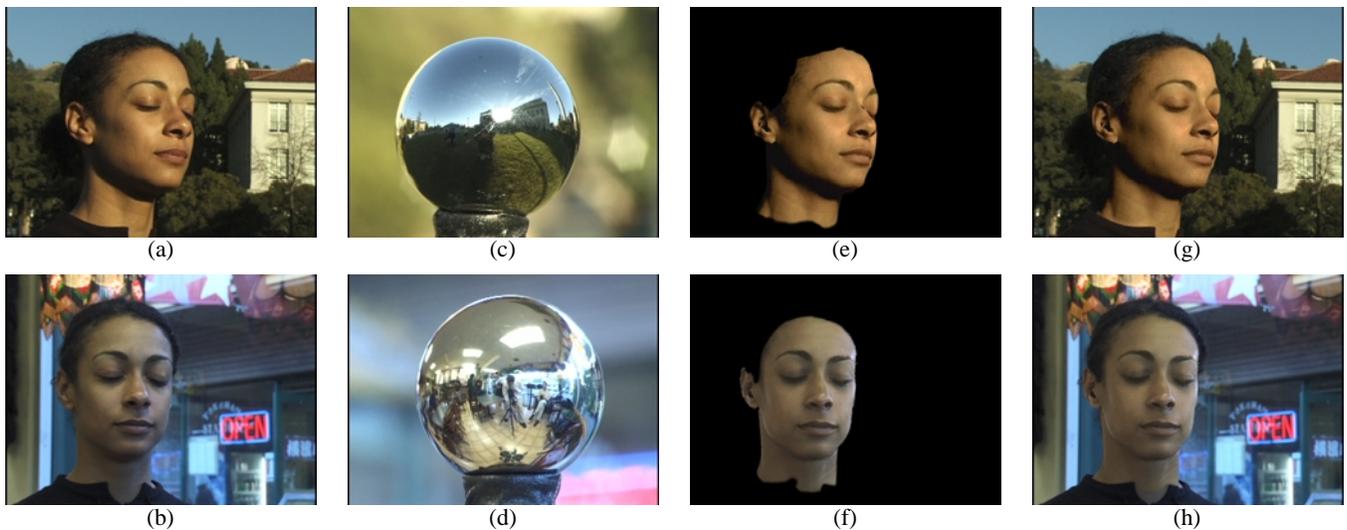


Figure 15: **Matching to Real-World Illumination** (a,b) Actual photographs of the subject in two different environments. (c,d) Images of a light probe placed in the position of the subject's head in the same environments. (e,f) Synthetic renderings of the face matched to the photographed viewpoints and illuminated by the captured lighting. (g,h) Renderings of the synthetic faces (e,f) composited over the original faces (a,b); the hair and shoulders come from the original photographs and are not produced using our techniques. The first environment is outdoors in sunlight; the second is indoors with mixed lighting coming from windows, incandescent lamps, and fluorescent ceiling fixtures.

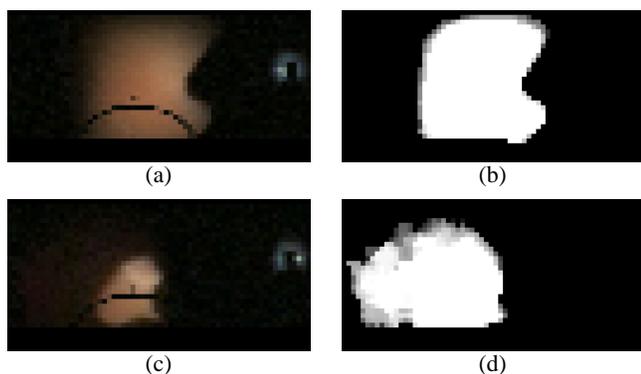


Figure 13: **Reflectance Function Shadow Maps** The reflectance function of a point near the nose (a) and the corresponding shadow map (b) computed using brightness thresholding. (c) shows a point in the ear which receives strong indirect illumination, causing the non-shadowed region in (d) to be overestimated. This causes some error in the diffuse-specular separation and the diffuse albedo to be underestimated in the ear as seen in Fig. 14(c).

5 Discussion and Future work

The work we have done suggests a number of avenues for improvements and extensions. First, we currently extrapolate reflectance functions using data from single viewpoints. Employing additional cameras to record reflectance functions for each location on the face would improve the results since less extrapolation of the data would be required. Using the polarization technique of Fig. 9 to directly record specular and subsurface reflectance functions could also improve the renderings, especially for subjects with pale skin.

A second avenue of future work is to animate our recovered facial models. For this, there already exist effective methods for animating geometrically detailed facial models such as [29], [14], and [34]. For these purposes, it will also be necessary to model and animate the eyes, hair, and inner mouth; reflectometry methods for obtaining models of such structures would need to be substantially different from our current techniques.

We would also like to investigate real-time rendering methods for our facial models. While the fixed-viewpoint re-illumination presented in Section 3 can be done interactively, synthesizing new viewpoints takes several minutes on current workstations. Some recent work has presented methods of using graphics hardware to render complex reflectance properties [18]; we would like to investigate employing such methods to create renderings at interactive rates. We also note that the storage required for a reflectance field could be substantially reduced by compressing the source data both in (u, v) space as well as (θ, ϕ) space to exploit similarities amongst neighboring reflectance functions.

Real skin has temporally varying reflectance properties depending on temperature, humidity, mood, health, and age. The surface blood content can change significantly as the face contorts and contracts, which alters its coloration. Future work could characterize these effects and integrate them into a facial animation system; part the acquisition process could be to capture the reflectance field of a person in a variety different expressions.

Lastly, the data capture techniques could be improved in a number of ways. High-definition television cameras would acquire nearly eight times as many pixels of the face, allowing the pixel size to be small enough to detect illumination variations from individual skin pores, which would increase the skin-like quality of the renderings. One could also pursue faster capture by using high-speed video cameras running at 250 or 1000 frames per second, allowing full reflectance capture in just a few seconds and perhaps, with more advanced techniques, in real time.

6 Conclusion

In this paper we have presented a practical technique for acquiring the reflectance field of a human face using standard video equipment and a relatively simple lighting apparatus. The method allows the face to be rendered under arbitrary illumination conditions including image-based illumination. The general technique of modeling facial reflectance from dense illumination directions, sparse viewpoints, and recovered geometry suggests several areas for future work, such as fitting to more general reflectance models and combining this work with facial animation techniques. It is our hope that the work we have presented in this paper will help encourage continued investigations into realistic facial rendering.

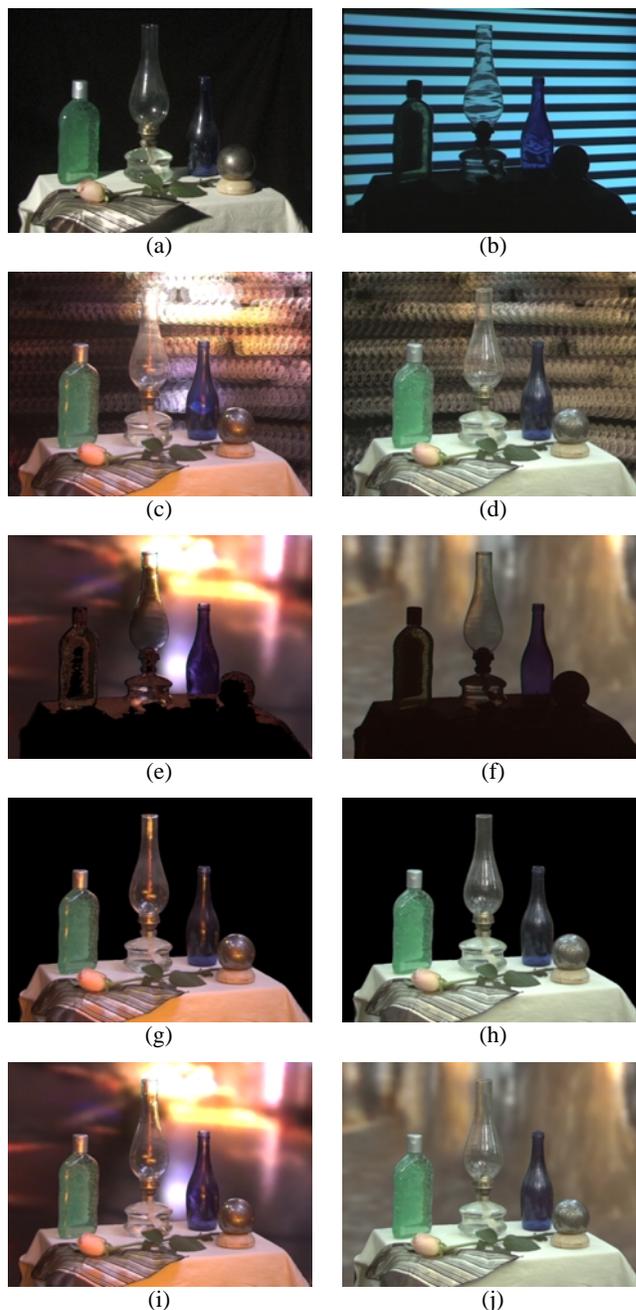
Acknowledgements

We would like to thank Shawn Brixey and UC Berkeley's Digital Media/New Genre program for use of their laboratory space, as well as Bill Buxton and Alias Wavefront for use of the Maya modeling software, and Larry Rowe, Jessica Vallot (seen in Fig. 14), Patrick Wilson, Melanie Levine, Eric Paulos, Christine Waggoner, Holly Cim, Eliza Ra, Bryan Musson, David Altenau, Marc Levoy, Maryann Simmons, Henrik Wann Jensen, Don Greenberg, Pat Hanrahan, Chris Bregler, Michael Naimark, Steve Marschner, Kevin Binkert, and the Berkeley Millennium Project for helping make this work possible. We would also like to acknowledge the Cornell's 1999 Workshop on Rendering, Perception, and Measurement for helping encourage this line of research. Thanks also to the anonymous reviewers for their insightful suggestions for this work. This work was sponsored by grants from Interactive Pictures Corporation, the Digital Media Innovation Program, and ONR/BMDO 3DDI MURI grant FDN00014-96-1-1200.

References

- [1] ADELSON, E. H., AND BERGEN, J. R. *Computational Models of Visual Processing*. MIT Press, Cambridge, Mass., 1991, ch. 1. The Plenoptic Function and the Elements of Early Vision.
- [2] BEIER, T., AND NEELY, S. Feature-based image metamorphosis. *Computer Graphics (Proceedings of SIGGRAPH 92)* 26, 2 (July 1992), 35–42.
- [3] BLANZ, V., AND VETTER, T. A morphable model for the synthesis of 3d faces. *Proceedings of SIGGRAPH 99* (August 1999), 187–194.
- [4] BREGLER, C., COVELL, M., AND SLANEY, M. Video rewrite: Driving visual speech with audio. *Proceedings of SIGGRAPH 97* (August 1997), 353–360.
- [5] BUSBRIDGE, I. W. *The Mathematics of Radiative Transfer*. Cambridge University Press, Bristol, UK, 1960.
- [6] COOK, R. L., AND TORRANCE, K. E. A reflectance model for computer graphics. *Computer Graphics (Proceedings of SIGGRAPH 81)* 15, 3 (August 1981), 307–316.
- [7] DANA, K. J., GINNEKEN, B., NAYAR, S. K., AND KOENDERINK, J. J. Reflectance and texture of real-world surfaces. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.* (1997), pp. 151–157.
- [8] DEBEVEC, P. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH 98* (July 1998).
- [9] DEBEVEC, P. E., AND MALIK, J. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH 97* (August 1997), pp. 369–378.
- [10] DEBEVEC, P. E., YU, Y., AND BORSHUKOV, G. D. Efficient view-dependent image-based rendering with projective texture-mapping. In *9th Eurographics workshop on Rendering* (June 1998), pp. 105–116.
- [11] FUA, P., AND MICCIO, C. From regular images to animated heads: A least squares approach. In *ECCV98* (1998).
- [12] GERSHUN, A. Svetovoe Pole (the Light Field, in English). *Journal of Mathematics and Physics XVIII* (1939), 51–151.
- [13] GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. F. The Lumigraph. In *SIGGRAPH 96* (1996), pp. 43–54.
- [14] GUENTER, B., GRIMM, C., WOOD, D., MALVAR, H., AND PIGHIN, F. Making faces. *Proceedings of SIGGRAPH 98* (July 1998), 55–66.
- [15] HABERLI, P. Synthetic lighting for photography. Available at <http://www.sgi.com/grafica/synth/index.html>, January 1992.

- [16] HANRAHAN, P., AND KRUEGER, W. Reflection from layered surfaces due to subsurface scattering. *Proceedings of SIGGRAPH 93* (August 1993), 165–174.
- [17] KARNER, K. F., MAYER, H., AND GERVAUTZ, M. An image based measurement system for anisotropic reflection. In *EUROGRAPHICS Annual Conference Proceedings* (1996).
- [18] KAUTZ, J., AND MCCOOL, M. D. Interactive rendering with arbitrary BRDFs using separable approximations. *Eurographics Rendering Workshop 1999* (June 1999).
- [19] LAFORTUNE, E. P. F., FOO, S.-C., TORRANCE, K. E., AND GREENBERG, D. P. Non-linear approximation of reflectance functions. *Proceedings of SIGGRAPH 97* (August 1997), 117–126.
- [20] LEE, Y., TERZOPOULOS, D., AND WATERS, K. Realistic modeling for facial animation. *Proceedings of SIGGRAPH 95* (August 1995), 55–62.
- [21] LEVOY, M., AND HANRAHAN, P. Light field rendering. In *SIGGRAPH 96* (1996), pp. 31–42.
- [22] MARSCHNER, S. R., WESTIN, S. H., LAFORTUNE, E. P. F., TORRANCE, K. E., AND GREENBERG, D. P. Image-based BRDF measurement including human skin. *Eurographics Rendering Workshop 1999* (June 1999).
- [23] MILLER, G. S. P., RUBIN, S., AND PONCELEON, D. Lazy decomposition of surface light fields for precomputed global illumination. *Eurographics Rendering Workshop 1998* (June 1998), 281–292.
- [24] NAYAR, S., FANG, X., AND BOULT, T. Separation of reflection components using color and polarization. *IJCV 21*, 3 (February 1997), 163–186.
- [25] NICODEMUS, F. E., RICHMOND, J. C., HSIA, J. J., GINSBERG, I. W., AND LIMPERIS, T. Geometric considerations and nomenclature for reflectance.
- [26] NIMEROFF, J. S., SIMONCELLI, E., AND DORSEY, J. Efficient re-rendering of naturally illuminated environments. *Fifth Eurographics Workshop on Rendering* (June 1994), 359–373.
- [27] OREN, M., AND NAYAR, S. K. Generalization of Lambert’s reflectance model. *Proceedings of SIGGRAPH 94* (July 1994), 239–246.
- [28] PARKE, F. I. Computer generated animation of faces. *Proc. ACM annual conf.* (August 1972).
- [29] PIGHIN, F., HECKER, J., LISCHINSKI, D., SZELISKI, R., AND SALESIN, D. H. Synthesizing realistic facial expressions from photographs. *Proceedings of SIGGRAPH 98* (July 1998), 75–84.
- [30] SAGAR, M. A., BULLIVANT, D., MALLINSON, G. D., HUNTER, P. J., AND HUNTER, I. W. A virtual environment and model of the eye for surgical simulation. *Proceedings of SIGGRAPH 94* (July 1994), 205–213.
- [31] SATO, Y., AND IKEUCHI, K. Temporal-color space analysis of reflection. *JOSA-A 11*, 11 (November 1994), 2990–3002.
- [32] SATO, Y., WHEELER, M. D., AND IKEUCHI, K. Object shape and reflectance modeling from observation. In *SIGGRAPH 97* (1997), pp. 379–387.
- [33] SMITH, B., AND ROWE, L. Compressed domain processing of JPEG-encoded images. *Real-Time Imaging 2*, 2 (1996), 3–17.
- [34] TERZOPOULOS, D., AND WATERS, K. Physically-based facial modelling, analysis, and animation. *Journal of Visualization and Computer Animation 1*, 2 (August 1990), 73–80.
- [35] TORRANCE, K. E., AND SPARROW, E. M. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America 57*, 9 (1967).
- [36] VAN GEMERT, M. F. C., JACQUES, S. L., STERENBERG, H. J. C. M., AND STAR, W. M. Skin optics. *IEEE Transactions on Biomedical Engineering 36*, 12 (December 1989), 1146–1154.
- [37] WARD, G. J. Measuring and modeling anisotropic reflection. In *SIGGRAPH 92* (July 1992), pp. 265–272.
- [38] WILLIAMS, L. Performance-driven facial animation. *Computer Graphics (Proceedings of SIGGRAPH 90)* 24, 4 (August 1990), 235–242.
- [39] WONG, T.-T., HENG, P.-A., OR, S.-H., AND NG, W.-Y. Image-based rendering with controllable illumination. *Eurographics Rendering Workshop 1997* (June 1997), 13–22.
- [40] WU, Y., THALMANN, N. M., AND THALMANN, D. A dynamic wrinkle model in facial animation and skin aging. *Journal of Visualization and Computer Animation 6*, 4 (October 1995), 195–206.
- [41] YU, Y., DEBEVEC, P., MALIK, J., AND HAWKINS, T. Inverse global illumination: Recovering reflectance models of real scenes from photographs. *Proceedings of SIGGRAPH 99* (August 1999), 215–224.
- [42] ZONGKER, D. E., WERNER, D. M., CURLESS, B., AND SALESIN, D. H. Environment matting and compositing. *Proceedings of SIGGRAPH 99* (August 1999), 205–214.



Appendix: Combining with Environment Matting

The light stage can be used to relight objects as well as faces. In this experiment we created a scene with diffuse, shiny, refractive, and transmissive objects seen in (a). Because of the sharp specularities, we recorded the scene with a finer angular resolution of 128×64 directions of θ and ϕ and in high dynamic range [9] using five passes of the light stage at different exposure settings. Renderings of the scene in two environments are shown in (c,d). Because high dynamic range imagery was used, the direct appearance of the light source was captured properly, which allows the renderings to reproduce a low-resolution version of the lighting environment in the background. To replace this with a high resolution version of the environment, we captured an environment matte [42] of the scene (b) and computed the contribution of the reflected, refracted, and transmitted light from the background (e,f). We then summed all but the contribution from the background lighting directions to produce (g,h) and added in the light from the environment matte (e,f) to produce a complete rendering of the scene and background (i,j).