High Dynamic Range Imaging

Paul Debevec
USC Institute for Creative Technologies

Erik Reinhard
University of Central Florida

Greg Ward
Anyhere Software

Sumanta Pattanaik
University of Central Florida

SIGGRAPH 2004 Course 13 (Half Day)
Monday, August 9, 2004

Course Abstract

Current display devices can display only a limited range of contrast and colors, which is one of the main reasons that most image acquisition, processing, and display techniques use no more than eight bits per color channel. This course outlines recent advances in high-dynamic-range imaging, from capture to display, that remove this restriction, thereby enabling images to represent the color gamut and dynamic range of the original scene rather than the limited subspace imposed by current monitor technology. This hands-on course teaches how high-dynamic-range images can be captured, the file formats available to store them, and the algorithms required to prepare them for display on low-dynamic-range display devices. The trade-offs at each stage, from capture to display, are assessed, allowing attendees to make informed choices about data-capture techniques, file formats, and tone-reproduction operators. The course also covers recent advances in image-based lighting, in which HDR images can be used to illuminate CG objects and realistically integrate them into real-world scenes. Through practical examples taken from photography and the film industry, it shows the vast improvements in image fidelity afforded by high-dynamic-range imaging.

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 office / (310) 577-9140 fax
paul@debevec.org / http://www.debevec.org/

Paul Debevec received his Ph.D. from UC Berkeley in 1996 where he worked with C.J. Taylor and Jitendra Malik to produce Facade, an early image-based modeling and rendering system for creating photoreal architectural models from still photographs. His work with high dynamic range imagery (HDRI) and image-based lighting has been incorporated into commercial rendering systems such as LightWave and RenderMan and has helped influence recent advancements in dynamic range in graphics hardware. The technology used in Debevec's short films at the SIGGRAPH Electronic Theater including "The Campanile Movie", "Rendering with Natural Light", and "Fiat Lux" has contributed to the visual effects in films including "The Matrix", "X-Men", and "The Time Machine". In 2001 he received ACM SIGGRAPH's Significant New Researcher award and in 2002 was named one of the world's top 100 young innovators by MIT's Technology Review Magazine for his work to develop the Light Stage. Today Debevec leads the computer graphics laboratory at USC's Institute for Creative Technologies and is a Research Assistant Professor in USC's computer science department.

Erik Reinhard
Assistant Professor
Computer Science Department
University of Central Florida
Orlando, FL 32816-2362
telephone: 407 - 823 4289
fax: 407 - 823 5419
reinhard@cs.ucf.edu / http://www.cs.ucf.edu/~reinhard/

Erik Reinhard is assistant professor at the University of Central Florida and has an interest in the fields of visual perception and parallel graphics. He has a B.S. and a TWAIO diploma in computer science from Delft University of Technology and a Ph.D. in computer science from the University of Bristol. He was a post-doctoral researcher at the University of Utah. He is founder and coeditor-in-chief of the journal ACM Transactions on Applied Perception, and guest editor of a special issue on Parallel Graphics and Visualisation for the journal Parallel Computing (March 2003). He is also co-editor of Practical Parallel Rendering (A K Peters, 2002). His current interests are in visual perception and its application to computer graphics problems such as tone reproduction and color correction. He is also active in parallel rendering.

Greg Ward
Anyhere Software
1200 Dartmouth Street #C
Albany, CA 94706
gward@lmi.net

Greg Ward (a.k.a.Greg Ward Larson) graduated in Physics from UC Berkeley in 1983 and earned a Masters in Computer Science from SF State University in 1985. Since 1985, he has working in the field of light measurement, simulation, and rendering variously at the Berkeley National Lab, EPFL Switzerland, Silicon Graphics Inc., Shutterfly, and Exponent. He is the author of the widely used RADIANCE package for lighting simulation and rendering.

Sumanta Pattanaik
Associate Professor
Computer Science Building, Room 251
University of Central Florida
Orlando, FL 32816-2362
Telephone: (407) 823-2638
Fax: 407 - 823 5419
sumant@cs.ucf.edu / http://www.cs.ucf.edu/~sumant/

Sumanta Pattanaik is an Associate Professor in the Computer Science Department of University of Central Florida. From 1995 to 2001 he was a research associate in the Program of Computer Graphics of Cornell University. Prior to that he was a post-doctoral researcher in IRISA/INRIA (1993-1995), France. His main fields of research are: realistic image synthesis, Mixed Reality and digital imaging. His research focuses on real-time rendering for virtual and mixed reality environments, the application of visual perception for efficient lighting computation and accurate display. He is the Category Editor of ACM Computing Reviews for the computer graphics area.

Course Schedule and Syllabus

1. Introduction and Overview (Reinhard)
   (8:30 - 8:40)

2. Taking High Dynamic Range images (Debevec)
   (8:40 - 9:20)

   - Dynamic range in the real world
   - Taking images with varying exposure - shutter, f/stop, ND filters
   - Deriving the response curve
   - Producing the radiance map
   - Viewing and editing HDR imagery
   - Capturing omnidirectional HDR images
     - Acquiring a light probe image using a mirrored ball
     - Fisheye and Panoramic camera techniques
     - Indirect capture of HDR intensities
     - Direct HDR Capture of the Sun and Sky
   - HDR Image Post-Processing (Vignetting, glare and motion blur...)

3. HDR Is as Easy as 1-2-3 (Ward)
   (9:20 - 10:00)

   - Easy capture of HDR Images
     - Automatic exposure bracketing
     - Hand-held image alignment
     - Lens flare reduction and ghost removal
   - HDR Image File formats
     - Manufacturer's raw formats
     - RADIANCE .pic/.hdr
     - LogLuv Tiff .tif
     - ILM's OpenEXR .exr
     - Emerging formats
   - Emerging High Dynamic Range Display Technologies

10:00-10:15 Break

4. The Human Visual System and HDR Tone Mapping (Pattanaik)
   (10:15 - 10:50)

   - Human Visual System (HVS) Physiology
   - HVS Sensitivity
   - Models of Visual Adaptation
   - Background Luminance in Images
   - Tone Reproduction Operator Design

5. Tone Reproduction Operators (Reinhard)
   (10:50 - 11:25)

   - Global operators
   - Local operators
   - Perceptually-inspired operators
   - Engineering-based solutions
   - Current state-of-the-art

6. HDR Image Based Lighting (Debevec)
   (11:25 - 12:05)

   - Illuminating synthetic objects with real light
     - IBL using a global illumination renderer
     - Simulating HDR Lighting with point light sources
     - Frequency-Space Image-Based Lighting
     - Useful Approximations to image-based lighting
     - Real-Time IBL using Graphics Hardware
   - Rendering synthetic objects into real scenes
     - Choosing a model of the local scene
     - Solving for scene reflectance properties
     - Computing shadows and interreflected light
   - Image-Based Lighting Real Objects and Actors
     - Reflectance Field Relighting of Human Faces
     - A Lighting Reproduction Approach to Live-Action Compositing
     - Re-illuminating Live-Action in Post-Production

7. Discussion and Q&A (All)
   (12:05 - 12:15)

Course Notes Table of Contents

0. Prologue: Abstract, Speakers, Schedule and Syllabus, and Table of Contents

1. Expository Material

Overview: *High Dynamic Range Image Encodings*
Greg Ward, 2004

Monograph: *Tone Mapping Operators*
Erik Reinhard, 2004

Paper: *A Tutorial on Image-Based Lighting*
Paul Debevec, Computer Graphics and Applications, March/April 2002

2. Course Slides

Slides: *Introduction: High Dynamic Range Imaging (Reinhard)*

Slides: *High Dynamic Range Image Capture (Debevec)*

Slides: *HDR is as Easy as 1-2-3 (Ward)*

Slides: *The Human Visual System and HDR Tone Mapping (Pattanaik)*

Slides: *Tone Reproduction Operators (Reinhard)*

Slides: *HDR Image-Based Lighting for Synthetic and Real Scenes (Debevec)*

3. Supplemental Material

Sketch: *Image-Based Modeling, Rendering, and Lighting in Fiat Lux*
Paul Debevec, SIGGRAPH 99 Technical and Animation Sketch

Sketch: *HDR Shop*
Chris Tchou and Paul Debevec, SIGGRAPH 2001 Technical Sketch

Sketch: *Light Stage 2.0*
Tim Hawkins, Jonathan Cohen, Chris Tchou, and Paul Debevec, SIGGRAPH 2001 Technical Sketch

4. Paper Reprints

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

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

Paper: *Overcoming Gamut and Dynamic Range Limitations in Digital Images*
Larson, G.W, Proc. Sixth Color Imaging Conference, November 1998

Paper: *A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes*
Gregory Ward Larson, Holly Rushmeier, and Christine Piatko
IEEE Transactions on Visualization and Computer Graphics, 3:4 December 1997

Paper: *High Dynamic Range Imaging*
Greg Ward, Proc. Ninth Color Imaging Conference, November 2001

Paper: *Subband Encoding of High Dynamic Range Imagery*
Greg Ward and Maryann Simmons, 1st Symposium on Applied Perception in Graphics and Visualization, August 2004

5. DVD-ROM Material: HDR and IBL Animations
("Movies" directory)

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

Video: *Re-illuminating Live-Action in Post-Production*
USC ICT Graphics Lab

6. DVD-ROM Material: Light Probe Image Library
("Probes" directory)

Program: HDRView.exe Windows program for viewing 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/

7. DVD-ROM Material: Rendering with Natural Light Source Files
("RNL_Source" directory)

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

High Dynamic Range Image Encodings

Greg Ward, Anyhere Software

Introduction

We stand on the threshold of a new era in digital imaging, when image files will encode the color gamut and dynamic range of the original scene, rather than the limited subspace that can be conveniently displayed with 20 year-old monitor technology. In order to accomplish this goal, we need to agree upon a standard encoding for high dynamic range (HDR) image information. Paralleling conventional image formats, there are many HDR standards to choose from.

This article covers some of the history, capabilities, and future of existing and emerging standards for encoding HDR images. We focus here on the bit encodings for each pixel, as opposed to the file wrappers used to store entire images. This is to avoid confusing color space quantization and image compression, which are, to some extent, separable issues. We have plenty to talk about without getting into the details of discrete cosine transforms, wavelets, and entropy encoding. Specifically, we want to answer some basic questions about HDR color encodings and their uses.

What Is a Color Space, Exactly?

In very simple terms, the human visual system has three different types of color-sensing cells in the eye, each with different spectral sensitivities. (Actually, there are four types of retinal cells, but the "rods" do not seem to affect our sensation of color – only the "cones." The cells are named after their basic shapes.) Monochromatic light, as from a laser, will stimulate these three cell types in proportion to their sensitivities at the laser's wavelength. Light with a wider spectral distribution will stimulate the cells in proportion to the convolution of the cell sensitivities and the light's spectral distribution. However, since the eye only gets the integrated result, it cannot tell the difference between a continuous spectrum and one with monochromatic sources balanced to produce the same result. Because we have only three distinct spectral sensitivities, the eye can thus be fooled into thinking it is seeing any spectral distribution we wish to simulate by independently controlling only three color channels. This is called "color metamerism," and is the basis of all color theory.

Thanks to metamerism, we can choose any three color "primaries," and so long as our visual system sees them as distinct, we can stimulate the retina the same way it would be stimulated by any real spectrum simply by mixing these primaries in the appropriate amounts. This is the theory, but in practice, we must apply a negative amount of one or more primaries to reach some colors. To avoid this condition, the CIE designed the XYZ color space such that it could reach any visible color with strictly positive primary values. However, to accomplish this, they had to choose primaries that are more pure than the purest laser – these are called "imaginary primaries," meaning that they cannot be realized with any physical device. So, while it is true that the human eye can be fooled into seeing many colors with only three fixed primaries, as a practical matter, it cannot be fooled into thinking it sees *any* color we like – certain colors will always be out of reach of any three real primaries. To reach all possible colors, we would have to have tunable lasers as our emitting sources. One day, we may have such a device, but until then, we will be dealing with restricted gamut color devices.

Figure 1 shows the perceptually uniform CIE (u′,v′) color diagram, with the positions of the CCIR-709 primaries. These primaries are a reasonable approximation to most CRT computer monitors, and officially define the boundaries of the standard sRGB color space. This triangular region therefore denotes the range of colors that may be represented by these primaries, i.e., the colors your eyes can be fooled into seeing. The colors outside this region, continuing to the borders of our diagram, cannot be represented on a typical monitor. More to the point, these "out-of-gamut" colors cannot be stored in a

standard sRGB image file, so we are forced to show stand-in colors in this figure.

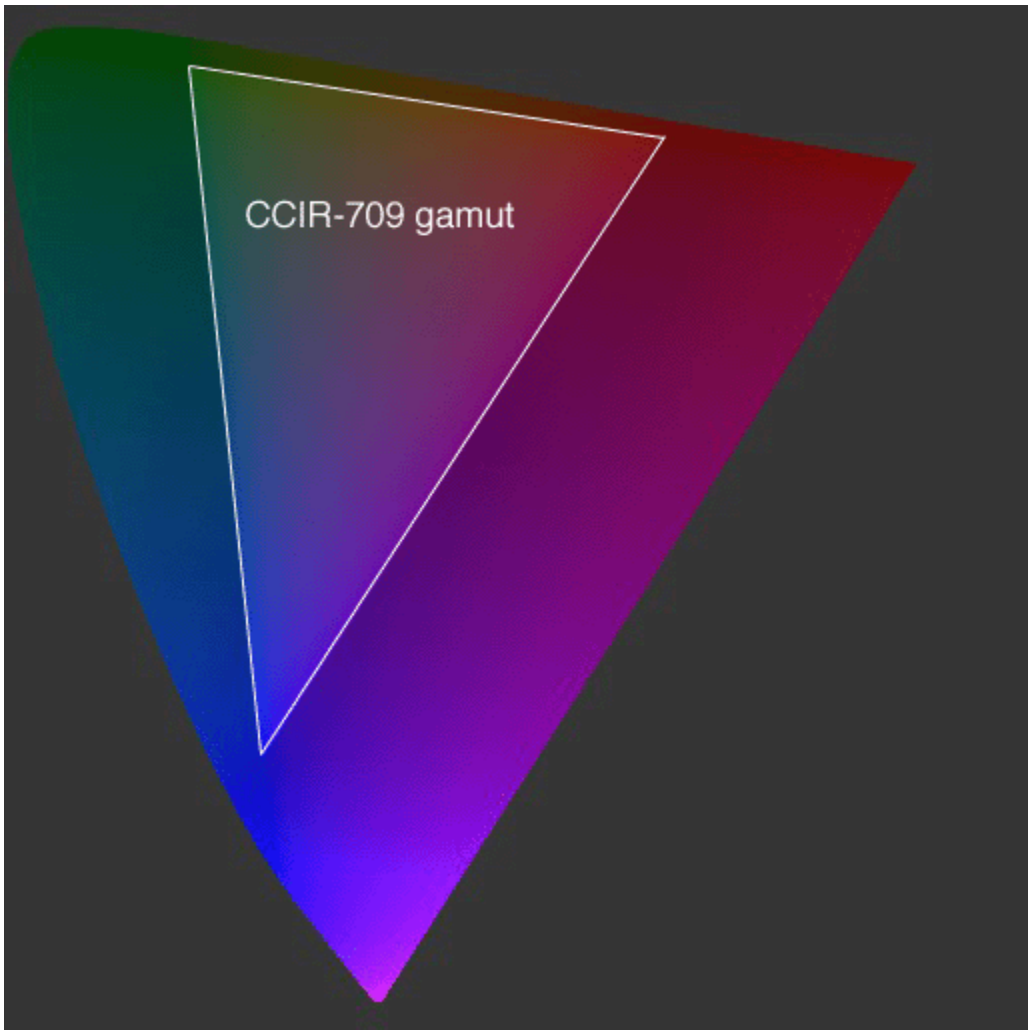


Figure 1. The CCIR-709 (sRGB) color gamut, shown within the CIE (u′,v′) color diagram.

The diagram in Figure 1 only shows two dimensions of what is a three-dimensional space. The third dimension, luminance, goes out of the page, and the *color space* is really a volume from which we have taken a slice. In the case of the sRGB color space, we have a six-sided polyhedron, often referred to as the "RGB color cube," which is misleading since the sides are only equal in the encoding (0-255 thrice), and not very equal perceptually.

A color space is really two things. First, it is a set of formulas that define a relationship between a color vector or triplet, and some standard color space, usually CIE XYZ. This is most often given in the form of a 3×3 *color transformation matrix*, though there may be additional formulas if the space is non-linear. Second, a color space is a two-dimensional boundary on the volume defined by this vector, usually determined by the minimum and maximum value of each primary, which is called the *color gamut*. Optionally, the color space may have an associated *quantization* if it has an explicit binary representation. In the case of sRGB, there is a color matrix with non-linear gamma, and quantized limits for all three primaries.

What Is a Gamma Encoding?

In the case of a color space defined on an integer domain, it is preferable for reasons of perceptual uniformity to establish a non-linear relationship between color values and intensity or luminance. By

coincidence, the output of a CRT monitor has a non-linear relationship between voltage and intensity that follows a *gamma law*, which is a power relation with a specific constant in the exponent:

$$I_{out} = K \cdot v^{\gamma}$$

In the above formula, output intensity is equal to some constant, K, times the input voltage or value, v, raised to a constant power, γ. (If v is normalized to a 0-1 range, then K simply becomes the maximum output intensity, I_{max}.) Typical CRT devices follow a power relation corresponding to a γ value between 2.4 and 2.8. The sRGB standard intentionally deviates from this value to something a target γ of 2.2, such that images get a slight contrast boost when displayed on a conventional CRT. Without going into any of the color appearance theory as to why this is desirable, let's just say that people seem to prefer it this way. The important thing to remember is that most CRTs do *not* have a gamma of 2.2 – even if most standard color encodings do. In fact, the color encoding and the display curve are two very separate things, which have become mixed in most programmers' minds because they follow the same basic formula, often mistakenly referred to as the "gamma correction curve." We are not "correcting for gamma" when we encode primary values with a power relation – we are attempting to minimize visual noise over the target intensity range. (See Charles Poynton's web pages on video encodings for a detailed explanation of gamma related issues.)

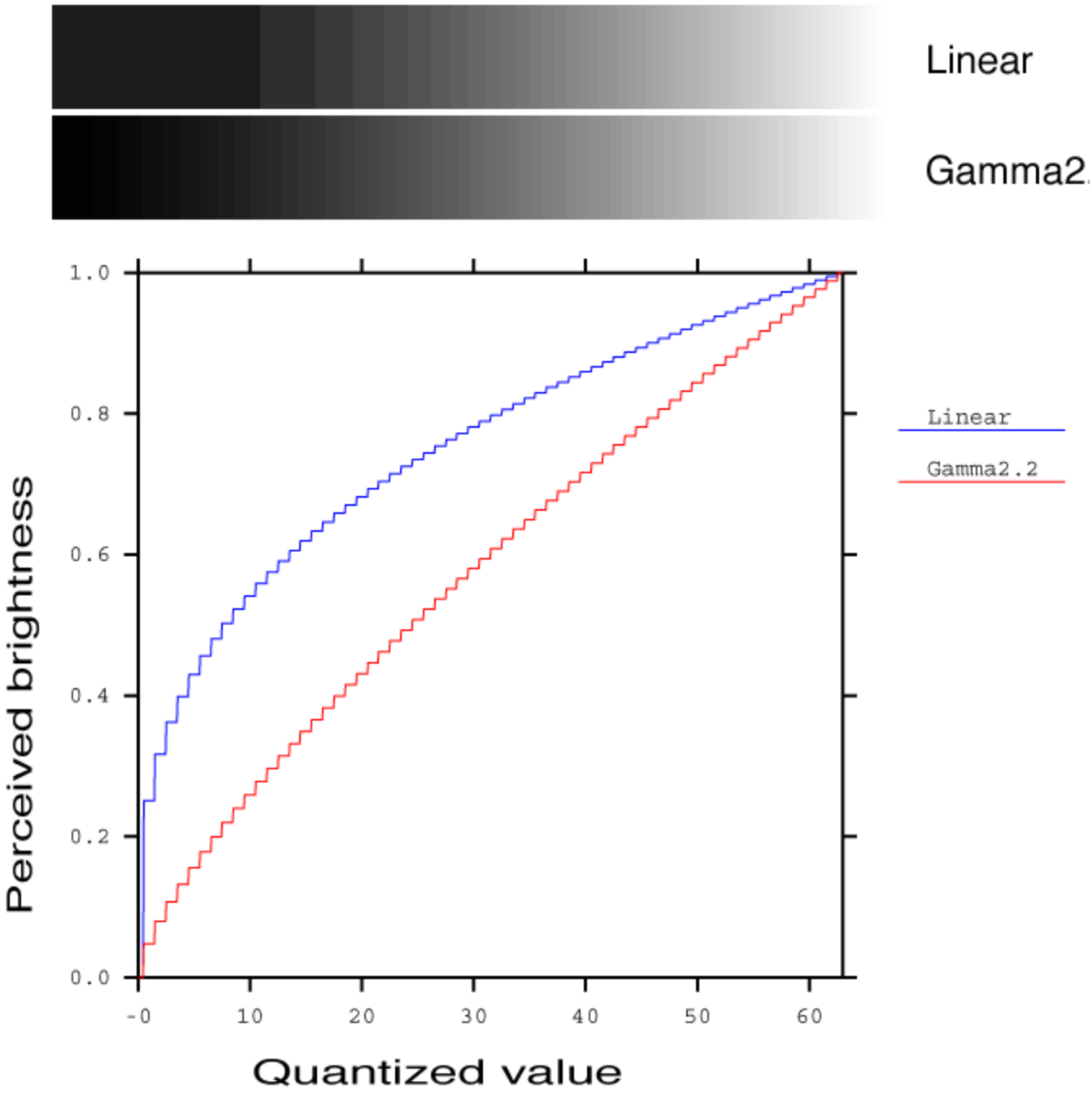


Figure 2. Perception of quantization steps using a linear and a gamma encoding. Only 6 bits are used in this example encoding to make the banding more apparent, but the same effect takes place in smaller steps using 8 bits per primary.

Errors arise from quantization in a digital color encoding. The eye has a non-linear response to intensity – at most adaptation levels, we perceive brightness roughly as the cube root of intensity. If we apply a linear quantization of color values, we see more steps in darker regions than we do in the brighter regions, as shown in Figure 2. Using a power law encoding with a γ value of 2.2, we see a much more even distribution of quantization steps, though the behavior near black is still not ideal. (For this reason, some encodings such as sRGB add a short linear range of values near zero.) However, we must ask what happens when luminance values range over several thousand or even a million to one. Simply adding bits to a gamma encoding does not result in a good distribution of steps, because we can no longer assume that the viewer is adapted to a particular luminance level, and the relative quantization error

tends towards infinity as the luminance tends towards zero.

What Is a Log Encoding?

To encompass a large range of values when the adaptation luminance is unknown, we really need an encoding with a constant or nearly constant relative error. A log encoding quantizes values using the following formula rather than the power law given earlier:

$$I_{out} = I_{min} \cdot \left[\frac{I_{max}}{I_{min}}\right]^{v}$$

This formula assumes that the encoded value v is normalized between 0 and 1, and is quantized in uniform steps over this range. Adjacent values in this encoding thus differ by a constant factor, equal to:

$$\left[\frac{I_{max}}{I_{min}}\right]^{(1/N)}$$

where N is the number of steps in the quantization. This is in contrast to a gamma encoding, whose relative step size varies over its range, tending towards infinity at zero. The price we pay for constant steps is a minimum representable value, I_{min}, in addition to the maximum intensity we had before.

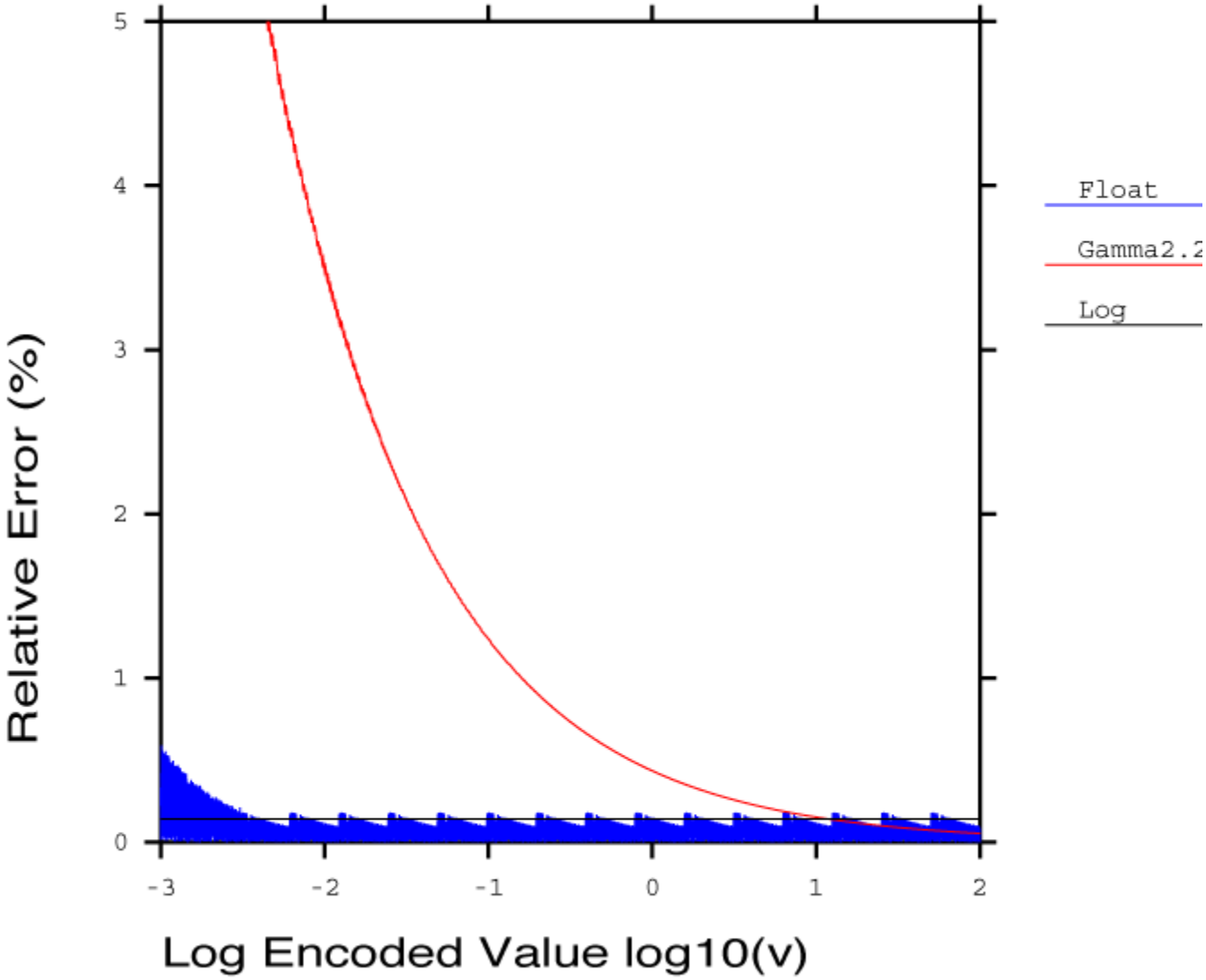


Figure 3. Relative error percentage plotted against $\log_{10}$ of image value for three encoding methods.

Another alternative closely related to the log encoding is a separate exponent and mantissa representation, better known as *floating point*. Floating point representations do not have perfectly equal step sizes, but follow a slight sawtooth pattern in their error envelope, as shown in Figure 3. To illustrate the quantization differences between gamma, log, and floating point encodings, we chose a bit size (12) and range (0.001 to 100) that could be reasonably covered by all three types. We chose a floating point representation with 4 bits in the exponent, 8 bits in the mantissa, and no sign bit since we are just looking at positive values. By *denormalizing* the mantissa at the bottom end of the range, we can also represent values between I_{min} and zero in a linear fashion, as we have done in this figure. By comparison, the error envelope of the log encoding is constant over the full range, while the gamma encoding error increases dramatically after just two orders of magnitude. Using a larger constant for γ helps this situation somewhat, but ultimately, gamma encodings are not really appropriate for HDR imagery.

What Is a Scene Referred Standard?

Most image encodings fall into a class we call *output referred standards*, meaning they employ a color space corresponding to a particular output device, rather than the original scene they are meant to

represent. The advantage of such a standard is that it does not require any manipulation prior to display, and it does not "waste" resources on colors that cannot be displayed on a particular output device. Conversely, the disadvantage of such a standard is that it cannot represent colors that may be displayable on some *other* output device, or may be useful in image processing operations along the way.

A *scene referred standard* follows a different philosophy, which is to represent the original, captured scene values as closely as possible. Display on a particular output device then requires some method for mapping the pixels to the device's gamut. This operation is referred to as *tone mapping*, and may be as simple as clamping RGB values to a 0-1 range, or something more sophisticated, like compressing the dynamic range or simulating human visual abilities and disabilities. The chief advantage gained by moving tone mapping to the image decoding and display stage is that we can produce correct output for *any* display device, now and in the future. Also, we have the freedom to apply complex image operations without suffering losses due to a presumed range of values.

The challenge of encoding a scene referred standard is finding an efficient representation that covers the full range of color values in which we are interested. This is precisely where HDR image encodings come into play.

What Are Some Applications of HDR Images?

Anywhere you would use a conventional image, you can use an HDR image instead by applying a tone mapping operation to get into the desired color space. The reverse is not true, because a conventional, output referred image cannot have its gamut extended to encompass a greater range – that information has been irretrievably lost. Therefore, HDR image applications are strictly a superset of conventional image applications, and suggest many new opportunities, which we have only begun to explore.

Below, we name just a few example applications that require HDR images as their input or as a critical part of their pipeline:

- Global illumination techniques (i.e., physically-based rendering)
- Mixed reality rendering (e.g., special effects for movies and commercials)
- Human vision simulation and psychophysics
- Reconnaissance and satellite imaging (i.e., remote sensing)
- Digital compositing for film
- Digital cinema

The list of applications is continuing to grow, and in the future, we expect digital photography to become almost exclusively HDR, as it was in the classic age of film and darkroom developing.

HDR Image Encoding Standards

The following list of HDR image standards is not meant to be comprehensive. Several other "deep pixel" standards that have been employed over the years, both in research and industry. However, most of these formats are not truly "high dynamic range," in the sense that they represent only an order of magnitude or so beyond the basic 24-bit RGB encoding, and most of them represent exactly the same color gamut. For this discussion, we are interested in pixel encodings that extend over 4 orders of magnitude, and prefer those which also encompass the visible color gamut, rather than a subset constrained by existing red, green, and blue monitor phosphors. If they meet these requirements, and have a luminance step size below 1% and good color resolution, they will be able to encode any image with fidelity as close to perfect as human vision is capable of discerning. We restrict our discussion to this class of encodings, with one or two exceptions.

Pixar Log Encoding (TIFF)

Computer graphics researchers and professionals have been aware of the limitations of standard 24-bit RGB representations for decades. One of the first groups to arrive at a standard for HDR image encoding was the Computer Graphics Division of Lucasfilm, which branched off in the mid-80's to become Pixar. Pixar's immediate need for HDR was to preserve their rendered images on the way out to their own custom film recorder. As most people know, film is capable of recording much more dynamic range than can be displayed on a typical CRT – close to 4 orders of magnitude instead of the usual 2, and generally with a log response rather than a gamma curve. Logically, Pixar settled on a logarithmic encoding, which Loren Carpenter implemented as a "codec" (compressor-decompressor) within Sam Leffler's TIFF library.

The format stored the usual three channels, one for each of red, green, and blue, but used an 11-bit log encoding for each, rather than the standard 8-bit gamma-based representation. Using this representation, Pixar was able to encode a dynamic range of roughly 3.6 orders of magnitude (3600:1) in 0.4% steps. A step of 1% in luminance is just on the threshold of being visible to the human eye.

This format is still employed internally at Pixar, but we are not aware of any regular applications using it outside the company. Its dynamic range of 3.8 orders of magnitude is marginal for HDR work, and it is not well-known to the computer graphics community, as there has never been a publication about it other than the source code sitting in Leffler's TIFF library. The lack of a negative range for the color primaries also means that the color gamut is restricted to lie within the triangle defined by the selected primary values. This limitation is shared by the next format we discuss, below.

Radiance RGBE Encoding (HDR)

In 1985, we began development of the *Radiance* physically-based rendering system at the Lawrence Berkeley National Laboratory. Because the system was designed to compute photometric quantities, it seemed unacceptable to throw away this information when writing out an image, so we settled on a 4-byte representation where three 8-bit mantissas shared a common 8-bit exponent. Had we spent a little more time examining the Utah Raster Toolkit, we might have noticed therein Rod Bogart had written an "experimental" addition that followed precisely the same logic to arrive at an almost identical representation as ours. (Rod later went on to develop the EXR format at Industrial Light and Magic, which we will discuss in a moment.) In any case, the RGBE encoding was written up in an article in Jim Arvo's Graphics Gems II, and distributed as part of the freely available *Radiance* system.

As the name implies, the *Radiance* RGBE format uses one byte for the red mantissa, one for the green, one for the blue, and one for a common exponent. The exponent is used as a scaling factor on the three linear mantissas, equal to 2 raised to the power of the exponent minus 128. The largest of the three components will have a mantissa value between 128 and 255, and the other two mantissas may be anywhere in the 0-255 range. The net result is a format that has an absolute accuracy of about 1%, covering a range of over 76 orders of magnitude.

Although RGBE is a big improvement over the standard RGB encoding, both in terms of precision and in terms of dynamic range, it has some important shortcomings. Firstly, the dynamic range is much more than anyone else could ever utilize as a color representation. The sun is about 10^8 cd/m^2, and the underside of a rock on a moonless night is probably around 10^{-6} or so, leaving about 62 orders of useless magnitude. It would have been much better if the format had less range but better precision in the same number of bits. This would require abandoning the byte-wise format for something like a log encoding. Another problem is that any RGB representation restricted to a positive range, such as RGBE and Pixar's Log format, cannot cover the visible gamut using any set of "real" primaries. Employing

"imaginary" primaries, as in the CIE XYZ color system, can encode the visible gamut with positive values, but often at the expense of coding efficiency since many unreal colors will also be represented – similar to the useless dynamic range issue. Finally, the distribution of error is not perceptually uniform with this encoding. In particular, step sizes may become visible in saturated blue and magenta regions where the green mantissa drops below 20. These shortcomings led us to develop a much better format, described next.

SGI LogLuv (TIFF)

Working at SGI in 1997, we set about correcting the mistakes made with RGBE, in hopes of providing an industry standard for HDR image encoding. This ultimately resulted in a LogLuv codec in Sam Leffler's TIFF library. This encoding is based on visual perception, and designed such that the quantization steps match human contrast and color detection thresholds (a.k.a. "just noticeable differences"). The key advantage is that quanta in the encoding are held below the level that might result in visible differences on a "perfect" display system. Its design is similar in spirit to a YCC encoding, but with the limitations on color gamut and dynamic range removed. By separating the luminance and chrominance channels, and applying a log encoding to luminance, we arrive at a very efficient quantization of what humans are able to see. This encoding and its variants were described in the 1998 Color Imaging Conference paper, "Overcoming Gamut and Dynamic Range Limitations in Digital Images."

There are actually three variants of this logarithmic encoding. The first squeezes a 10-bit log luminance value together with a 14-bit CIE (u′,v′) lookup to fit everything into a standard-length 24-bit pixel. This was mainly done to prove a point, which is that following a perceptual model allows you to make much better use of the same number of bits. In this case, we were able to extend to the full visible gamut and 4.8 orders of magnitude of luminance in *just* imperceptible steps. The second variation uses 16 bits for a pure luminance encoding, allowing negative values and covering a dynamic range of 38 orders of magnitude in 0.3% steps, which are comfortably below the perceptible level. The third variation uses the same 16 bits for signed luminance, then adds 8 bits each for CIE u′ and v′ coordinates to encompass all the visible colors in 32 bits/pixel.

Although the LogLuv format has been adopted by a number of computer graphics researchers, and its incorporation in Leffler's TIFF library means that a number of programs can read it even if they don't know they can, it has not found the widespread use we had hoped. Part of this is people's reluctance to deviate from their familiar RGB color space. Even something as simple as a 3×3 matrix to convert to and from the library's CIE XYZ interface is confounding to many programmers. Another issue is simply awareness – unless one has key industry partners, it is difficult to get word out on a new format, no matter what its benefits might be. The follow-up article, written for the ACM Journal of Graphics Tools (vol. 3, no. 1), "LogLuv encoding for full-gamut, high-dynamic range images," was an effort to stem these difficulties. We are still hopeful that this format will find wider use, as we have found it to work extremely well for HDR image encoding, ourselves. There is no more appropriate format for the archival storage of color images, at least until evolution provides an upgrade to human vision.

ILM OpenEXR (EXR)

More recently, Industrial Light and Magic published C++ source code for reading and writing their OpenEXR image format, which has been used internally by the company for special effects rendering and compositing for a number of years. This format is a general-purpose wrapper for the 16-bit Half data type, which has also been adopted by both NVidia and ATI in their floating-point frame buffers. Authors of the OpenEXR library include Florian Kainz, Rod Bogart (of Utah RLE fame), Drew Hess, Josh Pines, and Christian Rouet. The Half data type is a logical contraction of the IEEE-754 floating point representation to 16 bits. It has also been called the "S5E10" format for "Sign plus 5 exponent plus

10 mantissa," and this format has been floating around the computer graphics hardware developer network for some time. (OpenEXR also supports a standard IEEE 32-bit/component format, which we will not discuss here.)

Because it can represent negative primary values along with positive ones, the OpenEXR format covers the entire visible gamut and a range of about 10.7 orders of magnitude with a relative precision of 0.1%. Since humans can simultaneously see no more than 4 orders of magnitude, this makes OpenEXR a good candidate for archival image storage. Although the basic encoding is 48 bits/pixel as opposed to 32 for the LogLuv and RGBE formats, the additional precision is valuable when applying multiple blends and other operations where errors might accumulate. It would be nice if the encoding covered a larger dynamic range, but 10.7 orders is adequate for most purposes, so long as the exposure is not too extreme. The OpenEXR specification offers the additional benefit of extra channels, which may be used for alpha, depth, or spectral sampling. This sort of flexibility is critical for high-end compositing pipelines, and would have to be supported by some non-standard use of layers in a TIFF image.

We predict that the OpenEXR format will find widespread use in the computer graphics research community and special effects industry, as it has clear advantages for high quality image processing, and is supported directly by today's high-end graphics cards. ILM is to be commended for offering their excellent library for license-free use, and given the complexity of the most common variant, PIZ lossless wavelet compression, reimplementing EXR in a private library seems a formidable task.

Microsoft/HP scRGB Encoding

A new set of encodings for HDR image representation have been proposed by Microsoft and Hewlett-Packard, and accepted as an IEC standard (61966-2-2). The general format is called scRGB, formerly known as sRGB64. This standard grew out of the sRGB specification developed by Michael Stokes, Matthew Anderson, Srinivasan Chandrasekar, and Ricardo Motta, at HP and Microsoft. In essence, it is a logical extension of 24-bit sRGB to 16 linear bits per primary, or 12 bits per primary using a gamma encoding.

The scRGB standard is broken into two parts, one using 48 bits/pixel in an RGB encoding and the other employing 36 bits/pixel either as RGB or YCC. In the 48 bits/pixel substandard, scRGB specifies a linear ramp for each primary. Presumably, a linear ramp is employed to simplify graphics hardware and image-processing operations. However, a linear encoding spends most of its precision at the high end, where the eye can detect little difference in adjacent code values. Meanwhile, the low end is impoverished in such a way that the effective dynamic range of this format is only about 3.5 orders of magnitude – not really adequate from human perception standpoint, and too limited for light probes and HDR environment mapping. The standard does allow for negative primaries, which is an improvement over earlier RGB standards, but it makes wasteful use of this capability as much of the range is given to representing imaginary colors outside of the visible gamut, especially at the bottom end of the scale, where human color perception actually degrades. At the top end of its dynamic range, where people see colors more clearly, the gamut collapses in on itself again as primaries get clipped to the maximum representable value. This can be visualized on the colorspace animation page.

The second part of the standard offers a more sensible and compact 36 bits/pixel specification, which employs a standard gamma ramp with a linear subsection near zero. Although it uses 25% fewer bits, it has nearly as great a dynamic range as the 48-bit version, 3.2 orders as opposed to 3.5, and also allows negative primary values. It still suffers from a collapsing of its gamut at the top end and the unusable colors at the bottom, but it's a better encoding overall. Even better than this scRGB-nl substandard is the scYCC-nl encoding, which benefits from a separate luminance channel and a color space that does not collapse quite as quickly at the top end. However, there is some strange behavior at peak luminance where the brightest representable value is not white, but follows a ring of partially saturated colors

around white. This is because Cb and Cr contribute to the final luminance, so colored values (where Cb and Cr are non-zero) can be brighter than neutral ones. The net result is that colored light sources can become more saturated rather than less saturated during standard gamut clamping – fading to hot pink, for example. This situation is illustrated in Figure 4, below. This may also be seen on the colorspace animation page. In most situations, this would be an unexpected result, so care should be exercised when performing gamut mapping into scYCC-nl.

Figure 4. Gamut clamping anomalies using the scYCC-nl encoding. The first image shows a linear tone-mapping with clamping on the original HDR image of the sun. The second image shows the effect of simple clamping into an scYCC-nl gamut. Strange colorations result from the fact that the maximum representable luminances in scYCC-nl are not white.

It is unclear where the scRGB standard is headed at this point. Microsoft is promoting an undisclosed variant of this standard for use in digital cameras, graphics cards, and their Longhorn presentation engine, code named "Avalon" (October 2003 MS Developer's Conference). In the case of digital cameras, the only competing high resolution standard is no standard at all – the so-called "RAW" image formats that are not only manufacturer-specific, but camera model-specific as well. This is a disaster from a user and software maintenance standpoint, and almost any data standard would be an improvement over the current situation. Modern graphics cards, on the other hand, have already adopted the Half floating point data type used in OpenEXR and the two leading manufacturers, NVidia and ATI, have designed their chips around it. There would seem to be little benefit in switching to a more limited integer standard such as scRGB at this point. As for image processing software, the only benefit over existing 48-bit integer RGB formats is the extended gamut, which comes at the expense of dynamic range. A 48-bit RGB pixel using a standard 2.2 gamma as found in conventional TIFF images holds at least 5.4 orders of magnitude, though applications like PhotoshopCS are not presently designed to use this range to best advantage. Image processing would probably be done in memory using 32-bit IEEE floats, and the scRGB standard would only be useful for file storage. However, scRGB and scRGB-nl/scYCC-nl hold less dynamic range than any of the other formats we have discussed for this purpose. We hope that the secret variant that Microsoft is promoting is a substantial improvement over the written IEC standard.

HDR Encoding Comparison

The following table summarizes the information given in the previous section. The first row entry in the chart shows 24-bit sRGB, which of course is not a high dynamic-range standard, but offered here as a

baseline. The Bits/pixel value is for a tristimulus representation excluding alpha. The dynamic range is given as orders of magnitude, or the 10-based logarithm of the maximum representable value over the minimum value. The actual maximum and minimum are in parentheses. Dynamic range is difficult to pin down for non-logarithmic encodings such as scRGB, because the relative error is not constant throughout the range. We have selected 5% as the cut-off at the low end, when steps are considered too large to be included in the "useful" range. This is on the generous side, since viewers can detect luminance changes as small as 2%, but given that these errors usually occur in the darkest regions of the image, they may go unnoticed even at 5%. Non-logarithmic encodings are listed with "Variable" as their step size for this reason. Related formats with identical sizes and ranges are given together in the same row.

Encoding	Covers Gamut	Bits / pixel	Dynamic Range	Quant. Step
sRGB	No	24	1.6 (1.0:0.025)	Variable
Pixar Log	No	33	3.8 (25.0:0.004)	0.4%
RGBE XYZE	No Yes	32	76 (10^{38}:10^{-38})	1%
LogLuv 24	Yes	24	4.8 (15.9:0.00025)	1.1%
LogLuv 32	Yes	32	38 (10^{19}:10^{-20})	0.3%
EXR	Yes	48	10.7 (65000:0.0000012)	0.1%
scRGB	Yes	48	3.5 (7.5:0.0023)	Variable
scRGB-nl scYCC-nl	Yes Yes	36	3.2 (6.2:0.0039)	Variable

Table 1. HDR encoding comparison chart.

As we can see from Table 1, RGBE and XYZE are the winners in terms of having the most dynamic range in the fewest bits, and the XYZE encoding even covers the visible gamut. However, 76 orders of magnitude is so far beyond the useful range, that we have chosen to show XYZE as an outlier on our chart, below.

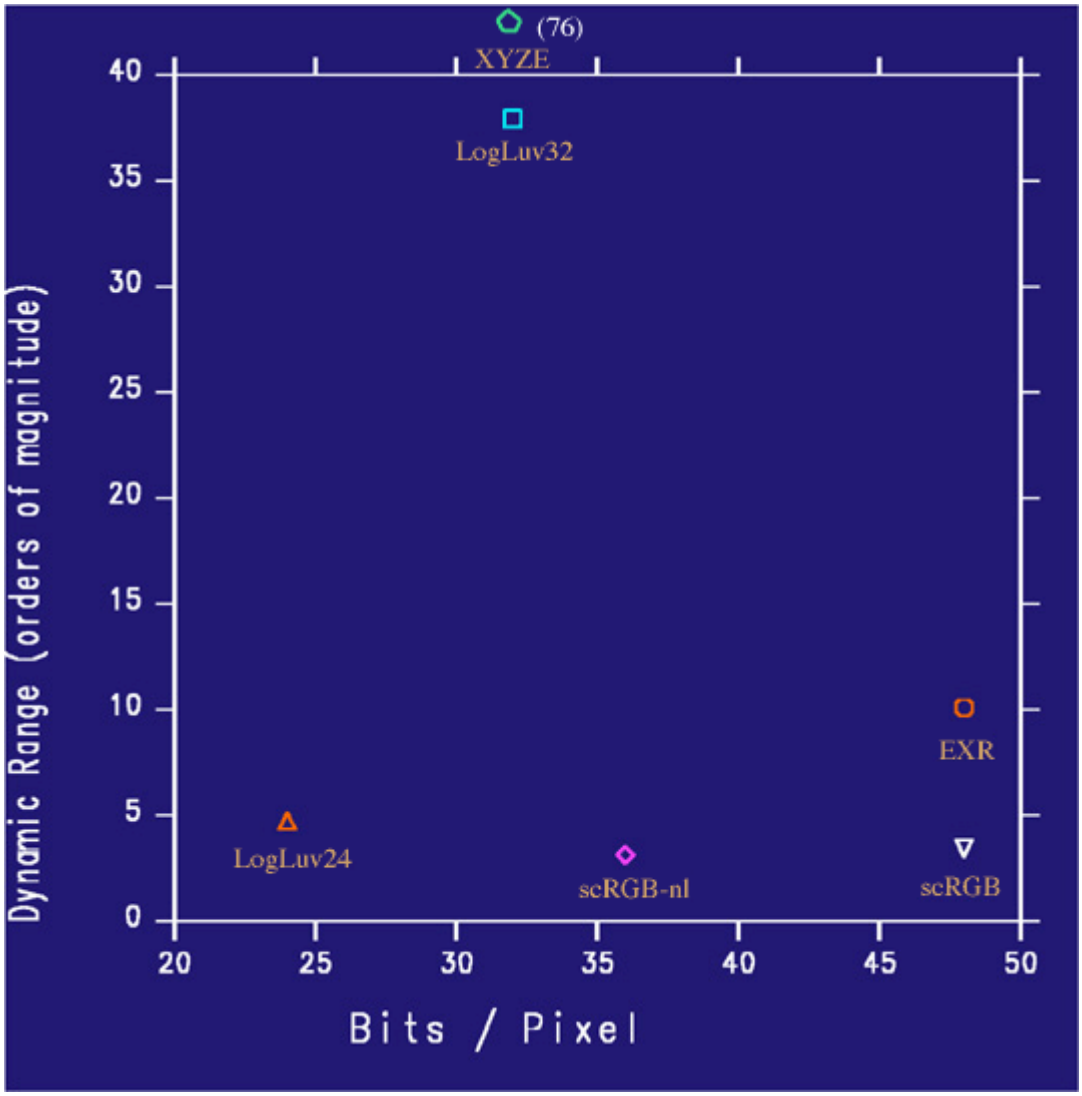


Figure 5. Cost (bits/pixel) vs. benefit (dynamic range) of full-gamut formats.

Figure 5 shows quite clearly the benefits of log and floating point representations over linear or gamma encodings. In 24 bits, the LogLuv format holds more dynamic range than the 36-bit scRGB-nl format, which uses a gamma encoding, and even the 48-bit scRGB linear encoding, which occupies twice the number of bits. In 32 bits, the LogLuv encoding holds 10 times the dynamic range of either of these formats. The EXR encoding holds 3 times the range of scRGB encoding in the same 48 bits, with much higher precision than any of the other formats.

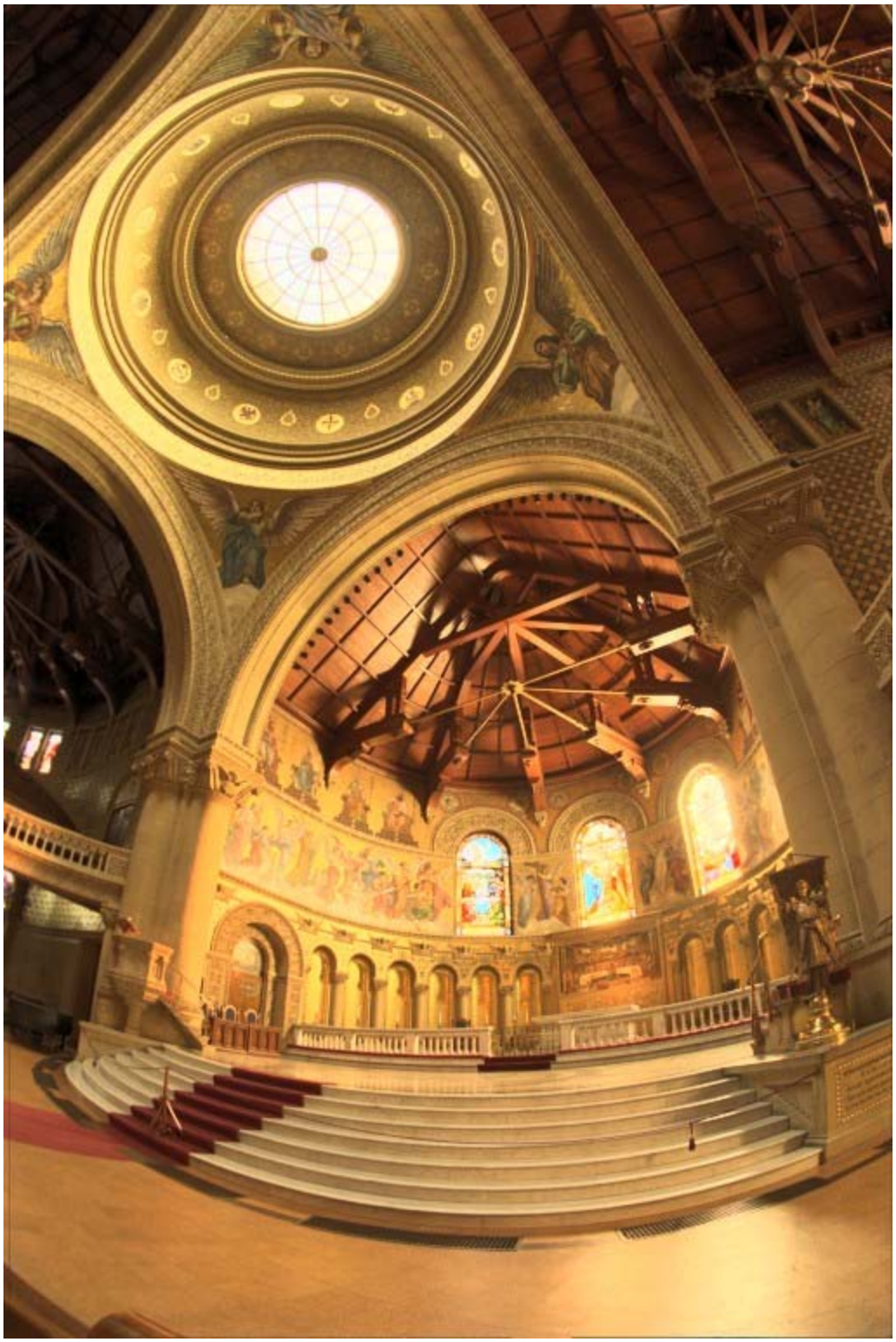
Image Results

What difference does any of this make to actual images? To find out, we collected a set of HDR images from various sources in different formats and set about converting from one format to another to see what sorts of errors were introduced by the encodings. We took images from three sources: captured exposure sequences combined directly into 96-bit/pixel floating-point TIFF, *Radiance*-rendered images in RGBE format, other people's captures in RGBE, and ILM-supplied captures in EXR format. In each case, we wrote out the image in a competing format and read it back, then did a pixel-by-pixel comparison. Comparing images visually is problematic, since high dynamic range displays are not currently available. The best we can do is run up and down locally with a linear tone-mapping, comparing the original and re-encoded images.

To be objective, we would also prefer a numerical comparison to a visual one. CIE ΔE* is a popular metric for quantifying color differences, but here again we run into difficulties when confronted with HDR input. Specifically, the CIE metric assumes a global white adaptation value, which makes sense for a paper task, but not for a scene where the brightest to the darkest regions may span a dynamic range of 10000:1. Relative to the brightest region in many of our images, the rest of the pixels would register as "a similar shade of black" as far as CIE 1994 is concerned. To cope with this problem, we decided to apply a "local reference white" in our comparisons, defined as the brightest pixel within 50 pixels of our current test position. The luminance of this brightest pixel, together with a global white point (chromaticity) for our image, is employed as the "reference white" in the CIE 1994 formula. This is not a perfect solution, and one could argue about the appropriate distance for choosing such a maximum, or if this is a valid approach at all, but it gave us what we were looking for, which was a measure of color differences between HDR images that roughly correlates with what we can see when we tone-map the regions locally.

Another problem we ran up against was optimizing the dynamic range for the desired comparison format. Especially for the LogLuv24 and scRGB-nl formats, the limited dynamic range available meant we needed to scale the original image to minimize losses at the top and bottom ends. Rather than coming up with a different scale factor optimized to the range of each format, we settled on one scale factor for each original image based on the most constrained destination, scRGB-nl. To find each factor, we examined the original image histogram and determined the scale factor that would deliver the largest population of pixels within the 6.2:0.0039 range of scRGB-nl. This factor was then applied to every format conversion for that image, simplifying our difference measurements. We expect this policy made the LogLuv24 results look worse than they should have, but did not adversely affect other encodings.

To demonstrate how these comparisons work and why they are necessary, let's start with an example. Figure 6a shows the familiar Memorial Church, tone-mapped to fit within a standard sRGB range. Figure 6b shows the same image passed through the 24-bit LogLuv encoding, which is not able to contain all of the original image's dynamic range. However, the degradation is difficult to see here, because we had to compress the whole range into a much smaller destination space, 24-bit sRGB. To actually observe the effect of the LogLuv24 encoding, we must zoom into the brightest region of the image, the stained glass windows, and remap locally using a linear operator. This comparison is shown in Figure 7.



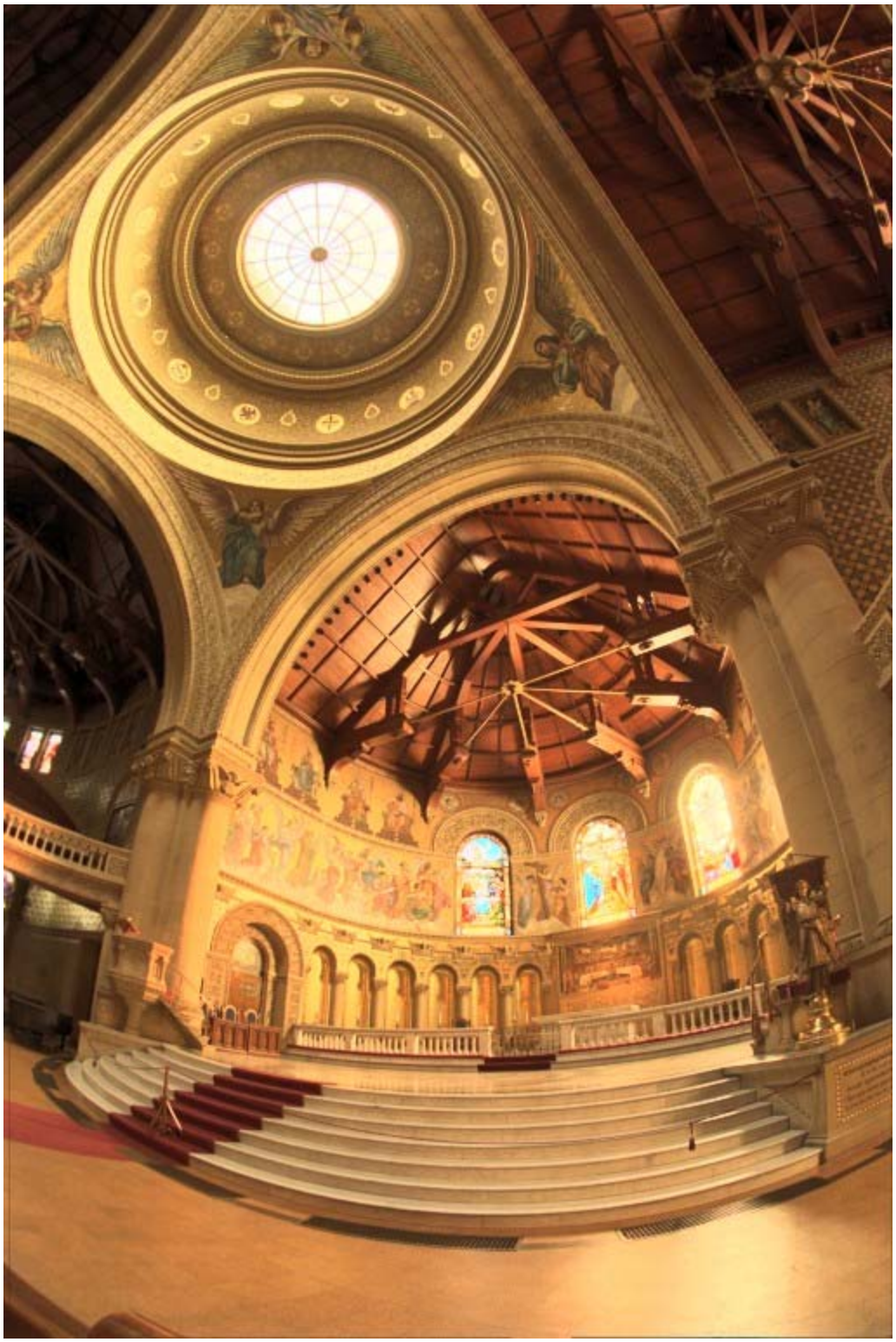


Figure 6. The Stanford Memorial Church. (Image courtesy Paul Debevec.) The first image is the original, tone-mapped into a printable range. The second image has been passed through the 24-bit LogLuv encoding prior to tone-mapping.

Figure 7. A close-up of the brightest region of the image with a linear tone-mapping. The original is shown on the left, and the subimage on the right has been passed through the 24-bit LogLuv encoding. Here we can see the losses incurred by exceeding the dynamic range of this encoding.

Through consistent application of our numerical difference metric, we can more easily identify problem areas. Figure 8 shows the difference metric comparing the original image with the version that has been passed through the 24-bit LogLuv encoding. This false color image highlights the errors in the window quite clearly, without the need for multiple scales and visual comparisons. The values corresponding to the color scale of the image are shown in the legend to the left. A ΔE^* value of 1 corresponds to the human detection threshold for colors that are immediately adjacent to each other. A ΔE^* value of 2 is generally considered visible for adjacent color patches, and ΔE^* values of 5 or greater may be discerned without difficulty in side-by-side images. As we can see, the brightest window in this image contains ΔE^* values greater than 12, which are highly visible.

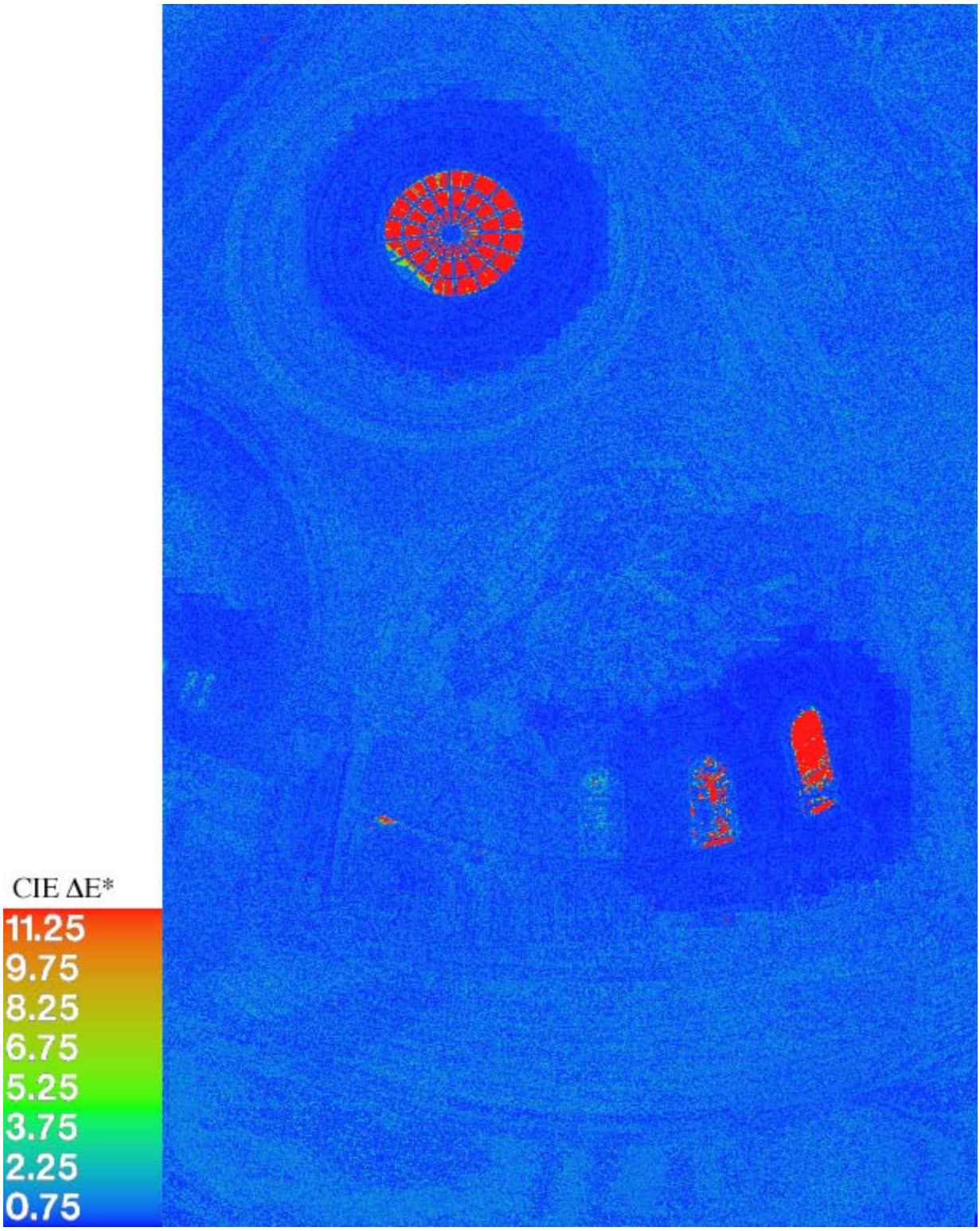


Figure 8. The modified 1994 CIE ΔE* metric showing the expected visible differences due to the 24-bit LogLuv encoding.

Figures 9 and 10 show the difference metrics and corresponding close-ups for the 32-bit LogLuv and 48-bit scRGB encodings, respectively.

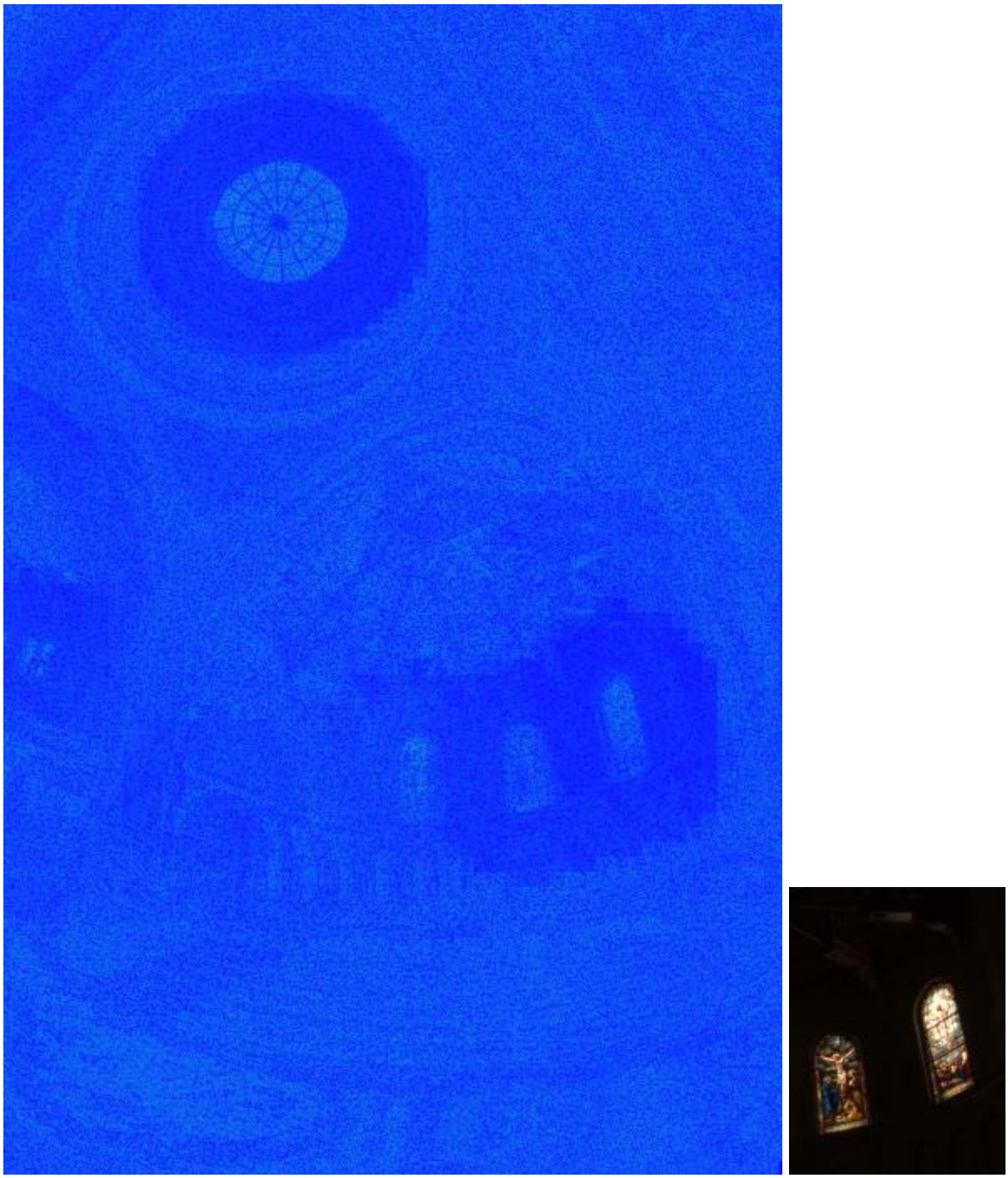


Figure 9. The Memorial Church image loses very little passing through the 32-bit LogLuv encoding, as can be seen in the color difference analysis and in the detail on the right. (The false color scale is the same in all images.)

Figure 10. As expected from its limited dynamic range, the 48-bit scRGB pixel format suffers significant losses in its attempt to encode the Memorial Church image. The detail on the right shows just how little of the original dynamic range has been preserved by this encoding.

We have performed similar analyses on a total of 33 original images using 8 different HDR encodings. These images are linked to the originals page, which is in turn linked with the encoding difference pages. Although it might be instructive, it would be tedious to go through all of these images and compare the results visually, running up and down the range of possible exposures in each encoding. Even relying on our false color representations of color differences, we still have 264 before/after comparison images to examine. One useful way of condensing this information for a particular image is to plot the percentile of pixels above a particular CIE ΔE* value for each encoding. This set of curves is shown in Figure 11 for the Memorial Church image.

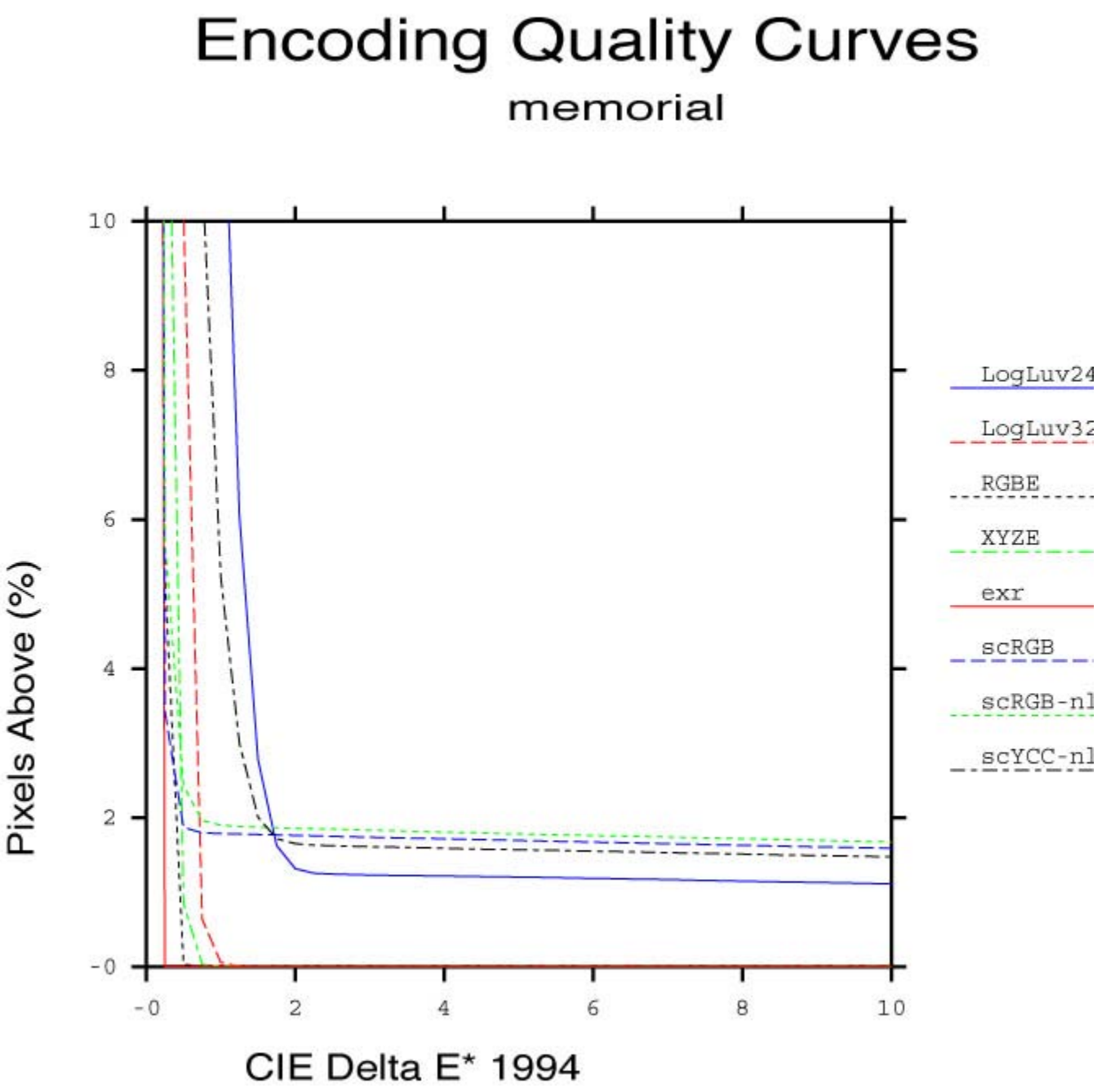


Figure 11. A plot showing the percentage of pixels above a particular ΔE* for each encoding of the Memorial Church image.

The ideal encoding behavior would be a steep descent that reached a very small percentile for anything above 2 on the CIE ΔE* axis. Indeed, this behavior is demonstrated by the EXR format in all of our example images, making it the clear winner in terms of color fidelity. This is not surprising, since the EXR format uses 48 bits/pixel and has smaller step sizes over its usable range than all the rest. The same cannot be said for the 48-bit scRGB format, which performed poorly in our comparisons – worse even than the lightweight 24-bit/pixel LogLuv encoding in terms of perceivable differences.

Results Summary

For our results summary in Table 2, we report a pair of values derived from our color difference metrics that we feel give a fair measure of how well a particular encoding performed on a particular HDR image. These are the percentile quantities taken from our plots for the percentage of pixels above two selected ΔE* values: ΔE*=2 and ΔE*=5. These correspond to the levels at which differences are detectable under ideal conditions and noticeable in side-by-side images, respectively.

Let's look at the first row of our results summary in Table 2. For the 24-bit LogLuv encoding, this says that 6.92% of the Apartment image pixels have a ΔE* of 2 or greater, meaning that one could (under ideal circumstances) distinguish these pixels from their original values. This may or may not be significant, depending on how critical your application is. If small color differences are acceptable, we may not care about the lower threshold, because one would not be able to tell the pixels apart unless they were flashed one on top of the other. However, the table also indicates that 0.31% of the pixels have a ΔE* of 5 or greater, which means they would be visibly different in side-by-side images. This is a more important value; however, unless these pixels are all bunched together in some important part of the image, they are likely to escape notice because they represent such a small fraction of the total. In general, it is time to worry when more than 2% of the pixels have a ΔE* of 5 or greater, meaning that a noticeable fraction of the image has colors that are visibly different from the original. These entries are highlighted in magenta in our table. For critical work, we might also be concerned when the ΔE*is above 2 over more than 5% of the image, and we have highlighted these entries in yellow.

Image	LogLuv24	LogLuv32	RGBE	XYZE	EXR	scRGB	scRGB-nl	scYCC-nl
Apartment	6.92% 0.31%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	1.21% 0.49%	5.14% 3.31%	2.63% 0.96%
AtriumNight	3.35% 0.18%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.27% 0.24%	0.31% 0.28%	0.30% 0.26%
Desk	5.95% 5.30%	1.05% 0.97%	1.56% 1.39%	0.81% 0.70%	0.00% 0.00%	8.68% 8.05%	9.92% 9.01%	8.26% 7.39%
Display1000	0.74% 0.05%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.72% 0.63%	0.99% 0.88%	0.73% 0.64%
Montreal	1.56% 0.03%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.08% 0.07%	0.09% 0.09%	0.06% 0.06%
MtTamWest	2.54% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.12% 0.08%	0.26% 0.20%	0.22% 0.12%
Spheron3	4.19% 0.02%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.11% 0.03%	0.15% 0.04%	0.19% 0.02%
Spheron NapaValley	4.47% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.01% 0.00%	0.09% 0.00%
Spheron Nice	11.14% 0.32%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.32% 0.32%	0.33% 0.32%	0.37% 0.32%
Spheron PriceWestern	2.32% 0.05%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.27% 0.20%	0.49% 0.36%	0.39% 0.32%
Spheron Siggraph2001	9.97% 0.66%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.97% 0.93%	1.06% 0.99%	1.00% 0.94%
StillLife	2.16% 1.39%	0.00% 0.00%	0.05% 0.05%	0.02% 0.02%	0.00% 0.00%	4.89% 4.12%	5.79% 4.84%	16.30% 3.71%
Tree	0.74% 0.27%	0.01% 0.01%	0.01% 0.01%	0.01% 0.01%	0.00% 0.00%	6.42% 5.48%	9.21% 7.89%	3.03% 0.95%
bigFogMap	2.02% 0.02%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.09% 0.08%	0.11% 0.09%	0.10% 0.08%
Dani belgium	2.61% 0.62%	0.00% 0.00%	0.01% 0.01%	0.01% 0.01%	0.00% 0.00%	1.51% 1.41%	1.72% 1.59%	1.58% 1.46%
Dani cathedral	0.64% 0.03%	0.00% 0.00%	0.02% 0.02%	0.02% 0.02%	0.00% 0.00%	2.64% 2.10%	3.80% 3.07%	2.40% 1.98%
Dani synagogue	5.06% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	2.70% 0.45%	0.00% 0.00%
memorial	1.31% 1.20%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	1.76% 1.69%	1.85% 1.78%	1.65% 1.57%
nave	1.66% 1.42%	0.00% 0.00%	0.01% 0.01%	0.00% 0.00%	0.00% 0.00%	2.10% 1.85%	2.35% 1.97%	1.95% 1.73%
rend01	13.76%	0.00%	0.00%	0.00%	0.00%	0.01%	0.02%	0.05%

	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.01% 0.00%	0.02% 0.00%	0.02% 0.00%
rend02	6.53% 0.05%	0.00% 0.00%	0.00% 0.00%	0.02% 0.02%	0.00% 0.00%	0.19% 0.17%	0.72% 0.70%	0.72% 0.69%
rend03	1.51% 0.11%	0.02% 0.02%	0.26% 0.26%	0.17% 0.17%	0.00% 0.00%	0.23% 0.22%	0.25% 0.24%	0.08% 0.08%
rend04	3.85% 1.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	1.71% 1.56%	1.93% 1.72%	1.91% 1.68%
rend05	3.18% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.05% 0.00%	0.57% 0.38%	0.00% 0.00%
rend06	1.72% 0.02%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.08% 0.07%	0.09% 0.08%	0.15% 0.06%
rend07	4.89% 0.56%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.76% 0.63%	0.80% 0.68%	0.81% 0.69%
rend08	12.05% 0.01%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.60% 0.59%	0.62% 0.61%	0.59% 0.58%
rend09	5.44% 1.31%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	1.36% 1.36%	1.37% 1.36%	1.37% 1.37%
rend10	3.54% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%
rend11	3.14% 0.01%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.06% 0.01%	0.16% 0.02%	0.84% 0.04%
rend12	6.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	2.05% 1.73%	3.55% 2.88%	3.69% 2.28%
rend13	9.47% 0.01%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	1.30% 1.28%	1.42% 1.34%	1.47% 1.34%
rosette	1.92% 1.60%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	0.00% 0.00%	4.92% 4.56%	5.35% 4.97%	3.66% 3.34%

Table 2. Image encoding difference results summary. The top value in each cell is the percentage of pixels with a CIE 1994 ΔE* above 2. The bottom value in each cell is the percentage of pixels with a ΔE* above 5. Highlighted entries correspond to images that are notably different from the originals.

The best overall summary we can offer is given in Figure 12. In this plot, we have combined results from all our examples for each encoding, weighing each of our 33 images equally. This plot shows the average behavior of each encoding over the whole data set. As hinted in the Memorial Church plot from Figure 11 and the results in Table 2, the 48-bit/pixel OpenEXR format is the clear winner, demonstrating very high accuracy over all of our example images. Surprisingly, second place goes to the 32-bit RGBE format, which performed consistently well despite its inability to record out-of-gamut colors. This may not be a fair test, however, since we presume none of our source images contained colors outside the standard CCIR-709 gamut. However, very close in performance to RGBE was XYZE, which *is* able to record out-of-gamut colors, removing this objection. As expected, the 32-bit LogLuv encoding performed very well in our tests, rarely exceeding a ΔE* of 2, and then only in the very darkest regions. This encoding did not keep errors quite as small as the others, but it always had a steep and rapid descent that reached 0.00% by a ΔE* of 1.5 in almost all of our examples. In comparison, the other 32-bit formats, RGBE and XYZE, occasionally left some small percentage of pixels with visible errors. This is due to the fact that the RGBE and XYZE do not employ a perceptual color space in their encodings, so we end up with perceptually significant step sizes over certain color ranges.

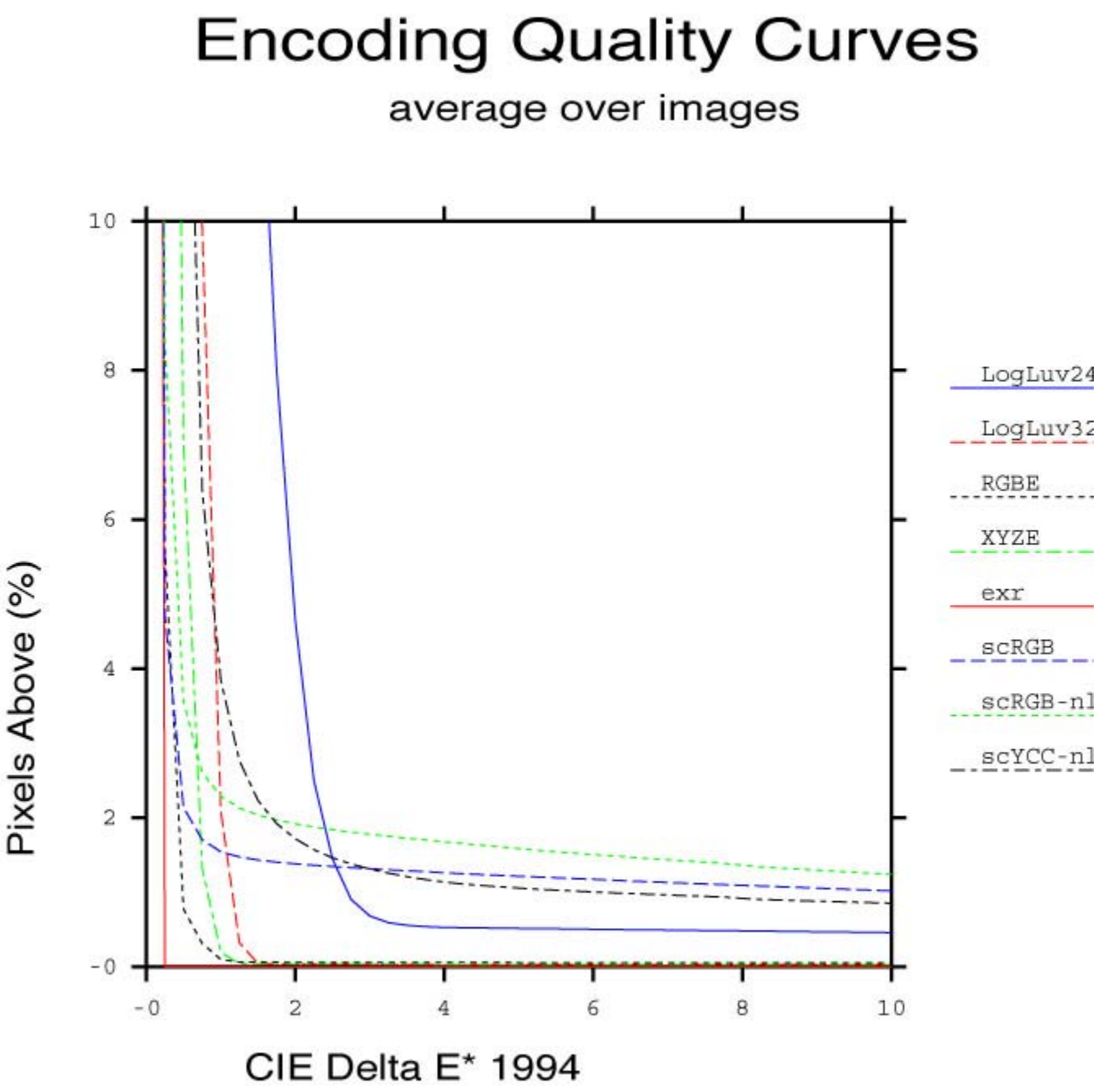


Figure 12. Summary performance of the different HDR encodings, summed over all our examples.

The biggest disappointments were the 48-bit scRGB, and the related 36-bit scRGB-nl and scYCC-nl encodings, which showed worse performance overall than their more compact predecessors, 32-bit/pixel RGBE, XYZE, and LogLuv. We should note that the scRGB format actually showed very good accuracy over the colors it was able to represent within its limited dynamic range, as demonstrated by its steep drop at the beginning. However, because many of our images would not fit within this encoding's 3.5 orders of magnitude, the tail of the curve continues out past ΔE* values where differences are visible. Since the express purpose of an HDR format is to represent these pixels accurately, in our opinion, the scRGB, scRGB-nl, and scYCC-nl formats might be better classified as "medium dynamic

range" encodings, along with the 24-bit variant of LogLuv TIFF.

Conclusion

There are many applications for high dynamic range images, and suitable formats exist in sufficient variety to support most work. The original *Radiance* RGBE format and its XYZE variant cover a vast dynamic range (76 orders of magnitude) with moderate accuracy (1%). The TIFF 32-bit LogLuv format covers half this range with over twice the accuracy. By comparison, the 48-bit ILM EXR format covers one quarter the LogLuv range with four times the accuracy. Each of these formats has been deployed and in use for a number of years, and they are all considered stable and reliable. The appropriate choice depends on the particular application's need for range, accuracy, and color space flexibility.

In contrast, the new IEC standard (61966-2-2) for extended gamut color seems a poor choice for HDR image encoding. For the number of bits consumed, this format offers poor resolution and range compared to other HDR standards, and is not recommended. We hope that Microsoft will offer a revised standard for the scRGB encoding, or abandon it for an existing solution. Since there are no royalties or burdens associated with any of the established encodings, there is no particular reason to reinvent them.

There is much work to be done in the area of lossy compression formats for HDR images. This is needed especially for digital photography and digital cinema, where image bit rates are a very important practical consideration. We simply cannot afford to use 48 bits for every pixel value, or even 24 – we need to compress the data to store it and transmit it without exhausting our resources, and we are willing to suffer some data loss along the way, so long as it is kept (mostly) below the visible threshold.

We said near the beginning that pixel encodings and image wrappers are largely separable problems, but they are not completely separate. Some color encodings work better than others for compression. Many compression algorithms benefit by splitting each pixel color into a luminance channel and an *opponent* chroma pair. This is the idea behind the YCC encoding used in JPEG, as well as the NTSC video standard, which also downsample the chrominance channels prior to transmission. ILM has experimented with an opponent color space with downsampling, which may be included in future versions of their OpenEXR library. The LogLuv encoding is perfectly matched for this type of compression, since the CIE (u′,v′) coordinates need only an offset to be an opponent chroma pair. We hope to see such an encoding in a future version of JPEG 2000, which offers a choice of original bit lengths for its color channels.

Tone Reproduction Operators

Erik Reinhard

Abstract

In this short survey several solutions to the problem of dynamic range reduction are classified into two groups: operators inspired by image formation, and operators based on knowledge of the human visual system.

1 Introduction

Visual simulations routinely produce images with a high dynamic range [Ward Larson and Shakespeare 1998]. Recent developments in image capturing techniques allow multiple exposures to be aligned and recombined into a single high dynamic range image [Debevec and Malik 1997]. Multiple exposure techniques are also available for video. In addition we expect future hardware to be able to photograph or film high dynamic range scenes directly.

As it is becoming easier to create high dynamic range imagery, the need to display such data is rapidly increasing. There are two strategies available to display high dynamic range images. First, we may develop display devices which can directly accommodate a high dynamic range [Seetzen et al. 2003]. This approach is detailed elsewhere in this course.

Second, we may prepare high dynamic range images for display on low dynamic range display devices. This is currently the more common approach, and the topic of this chapter. Although we foresee that high dynamic range display devices will become widely used in the (near) future, the need to compress the dynamic range of an image may diminish, but will not disappear. In particular, printed media are by their very nature low dynamic range.

Compressing the range of values of an image for the purpose of display on a low dynamic range display device is called tonemapping or tone reproduction. A simple compressive function would be to normalize an image. This constitutes a linear scaling which tends to be sufficient only if the dynamic range of the image is only marginally higher than the dynamic range of the display device. For images with a higher dynamic range, small intensity differences will be quantized to the same display value such that visible details are lost.

In general linear scaling will not be appropriate for tone reproduction. The key issue in tone reproduction is then to compress an image while at the same time preserving some attributes of the image. Different tone reproduction algorithms focus on different attributes with the result that we have a large set of algorithms which aim to preserve contrast, visible detail, brightness or even appearance.

Although it would be possible to classify tone reproduction operators by which attribute they try to preserve, we prefer to follow a different approach and classify algorithms according to their general technique. This will enable us to show the differences and similarities between a significant number of different operators, and so hopefully contribute to the meaningful selection of specific operators for given tone reproduction tasks.

The main classification scheme we follow hinges upon the realization that tone reproduction operators are based on insights gained from various different disciplines and fields. In particular, a large number of operators are based on knowledge of human visual perception. Since this constitutes the largest group of operators, we have dedicated a separate module of this course to an outline of human visual perception.

A second class of operators is grounded in physics. We know that images are formed by light being reflected off surfaces. In computer graphics this is generally modeled by the rendering equation. However, for purely diffuse surfaces this equation may be simplified to be a simple product between light incident upon a surface, and this surface's ability to reflect light [Oppenheim et al. 1968]. The former is called illuminance, and the latter is called reflectance.

Since reflectance is a passive property of surfaces, it is by definition low dynamic range - typically between 0.005 and 1 [Stockham 1972]. The reflectance of a surface cannot be larger than 1, since then it would reflect more light than was incident upon the surface. Illuminance on the other hand can produce arbitrarily large values and is limited only by the intensity and proximity of the light sources.

The dynamic range of an image is thus predominantly governed by the illuminance component. A viable approach to tone reproduction may therefore be to separate reflectance from illuminance, compress the illuminance component, and then recombine the image.

A third and much smaller class of tone reproduction operators is not based on insights of image formation or human visual perception, but uses principles from engineering or mathematics.

In the following sections, we describe each class of operator in more detail.

2 Operators based on image formation

A reasonable assumption is that an image is formed as the result of light being reflected off surfaces. The luminance L of each pixel is then given by the following product:

$$L(x, y) = r(x, y)\, i(x, y)$$

Here, r denotes the reflectance of a surface, and i denotes the illuminance. Alternatively, we may write this expression in the logarithmic domain [Oppenheim et al. 1968]:

$$D(x, y) = log(r(x, y)\, i(x, y)) = log(r(x, y)) + log(i(x, y))$$

In analogy with common practice in photography, we will use the term 'density representation' for log luminance. We see that when represented in the log domain, reflectance and illuminance become additive. This is a feature exploited for instance in homomorphic filtering.

2.1 Frequency-based operators

An observation already made is that the dynamic range of the reflectance component is low, whereas the illuminance may have an almost arbitrarily high dynamic range. A second observation is that for typical scenes, the reflectance component tends to exhibit high spatial frequencies due to textured surfaces as well as the presence of surface edges. On the other hand, illuminance tends to be a slowly varying function over space. The latter is of course only true if light sources are not directly visible.

Since reflectance is low dynamic range and illuminance is high dynamic range, we may try to separate the two components. The second observation we made provides a possible way to achieve this. We could for instance compute the Fourier transform of an image and attenuate only the low frequencies. This would compress the illuminance component while leaving the reflectance component largely unaffected. As it happens, the very first tone reproduction operator known to us takes this approach [Oppenheim et al. 1968].

More recently, other operators have also followed this line of reasoning. In particular, the bilateral and trilateral filters were used to separate an image into 'base' and 'detail' layers [Durand and Dorsey 2002; Choudhury and Tumblin 2003]. Both filters are edge-preserving smoothing operators which may be used in a variety of different ways. The result of applying an edge-preserving smoothing operator to a density image is a blurred image in which sharp edges remain present. We may view such an image as a 'base' layer. If we then pixel-wise divide the high dynamic range image by the base layer, we obtain a 'detail' layer which contains all the high frequency detail.

It is thus not a big stretch of the imagination to view the base layer as the illuminance component and the detail layer as the reflectance component. By compressing the base layer before recombining into a compressed density image followed by exponentiation, a low dynamic range image may be obtained.

We note that edge-preserving smoothing operators may also be used to compute a local adaptation level for each pixel, which may be used in a spatially varying or local tone reproduction operator. We describe this use of the bilateral and trilateral filters in the Section 3.

2.2 Gradient-domain operators

In image areas that exhibit a high dynamic range, we may also expect the spatial derivative or gradient of the image to exhibit large jumps. The derivative of a discrete image may be computed using a differencing scheme.

The arguments made for the frequency-based operators in the preceding section also hold for the gradient field. Assuming that no light sources are directly visible, the reflectance component will show up as a constant function with sharp spikes in the gradient field. Similarly, the illuminance component will cause small gradients everywhere.

To assess the lightness of an image, which is defined as perceived reflectance, Horn was the first to separate reflectance and illuminance using a gradient field [Horn 1974]. He used simple thresholding to remove all small gradients and then integrated the image, which is currently done by solving a Poisson equation using the Full Multigrid Method [Press et al. 1992]. The result is remarkably similar to an edge-preserving smoothing filter, which is no big surprise since the method is based on the same principle of image formation and uses the same assumptions. In particular, his work was directly aimed at 'mini-worlds of Mondriaans', which are simplified versions of diffuse scenes which to some extend resemble the abstract paintings by the famous Dutch painter Pieter Mondriaan.

Horn's work cannot be employed directly as a tone reproduction operator, since most high dynamic range images depict light sources. However, a relatively small variation will turn this work into a suitable tone reproduction operator. If light sources are present in the image, then large gradients will be associated with the edges of the light sources and thus cause the image to have a high dynamic range. We could therefore compress a high dynamic range image by attenuating large gradients, rather than thresholding the gradient field. This approach was taken by Fattal et al who showed that high dynamic range imagery may be successfully compressed by integrating a compressed gradient field [Fattal et al. 2002].

3 Operators based on human vision

Human brain cells carry information through discrete electrical signals called spike-trains. By their very nature spike trains have an upper and a lower frequency. Similarly the first few layers of cells in the human visual system carry analog information over short distances using graded potentials which also have an upper and a lower limit. Thus, the information carrying capacity of neural pathways in the human visual system is limited.

The human visual system takes as input light that enters the eye and converts it first to voltages and then spike trains before analyzing the captured light with a large variety of modules located in the visual cortex. Since the amount of light entering the eye may have an almost arbitrary dynamic range, which is subsequently represented with signals that are limited in dynamic range, the human visual system appears to be solving the tone reproduction problem in a natural and effortless way.

It is thus reasonable to assume that we may learn from the human visual system and develop tone reproduction operators based on our knowledge of the human visual system. This approach has gained sufficient popularity to warrant devoting a separate section of these course notes to describe the human visual system. In this chapter we consider this knowledge known, and limit ourselves to describing specific compression algorithms which are based on aspects of the human visual system.

3.1 Global operators

Global tone reproduction operators rescale pixel intensities according to a particular functional shape which may be adjusted to the image using information computed from all pixels together. For instance, we may compute the minimum, the maximum or the average luminance, and use these quantities to steer the compression function. Often, the log average is used as an adaptation value. The log average is equivalent to the geometric mean of an image.

The functional shape of global operators tends to fall in one of four categories, namely linear factors, logarithmic functions, histogram-based techniques, and sigmoidal response curves.

Linear scaling is sometimes used if the amount of compression required is reasonably small. Linear scale factors based on the human visual system attempt to map the average image intensity to a sensibly chosen display luminance [Ward 1994; Ferwerda et al. 1996].

Logarithmic compression is a straightforward non-linear functional shape which may be used for dynamic range reduction:

$$L_d(x,y) = \frac{\log(1 + L_w(x,y))}{\log(1 + L_{w,\max})}$$

Since the base of the logarithm is not specified in the above equation, it is possible to freely choose the base. In addition, the base may be made a function of the pixel intensity itself [Drago et al. 2003].

The cumulative histogram of an image may be directly used to remap image intensities. This is normally called histogram equalization. However, for the purpose of tone reproduction it is better to reshape the cumulative histogram to avoid gradients larger than one so that contrasts in highly populated areas of the histogram are not expanded [Ward et al. 1997].

Electro-physiological measurements of photoreceptors have shown that the visual system of many different species compresses input intensities according to an S-shapes curves (when plotted on a log-linear scale) [Naka and Rushton 1966; Kleinschmidt and Dowling 1975; Hood et al. 1979]. The same type of response curves are used in photography. The advantages of this shape is that it bounds the output in both the dark and the light regions. Moreover, the mid-region shows logarithmic behavior. Logarithmic compression

with asymptotes at the low and high ends are desirable features of tone reproduction operators, which accounts for their success. Sigmoidal response curves are given by the following general functional form:

$$L_d(x,y) = \frac{L_w(x,y)^n}{L_w(x,y)^n + \sigma^n}$$

The exponent n may either be fixed or be made image dependent, in which case it will most likely depend on some globally computed quantities such as the minimum, maximum or mean image intensity. The constant σ is generally known as the semi-saturation constant. This constant may either be the same value for all images [Schlick 1994; Reinhard et al. 2002], or may be made dependent on either a global or local adaptation value [Reinhard and Devlin 2004]. It may also be used to steer time-dependent behavior [Pattanaik et al. 2000].

Global operators are effective for medium dynamic range images. They may also produce reasonable output for high dynamic range images, although the visual quality of the results tends to depend on the composition of the image.

3.2 Local operators

For high dynamic range images, compression above the level afforded by global operators may be obtained by using a spatially variant, or local, operator. This class of operators reduces the range of values by considering for each pixel its intensity as well as the intensities of a set of neighboring pixels. The general observation is that a bright pixel which is surrounded by a collection of dark pixels will be perceived differently from a bright pixels that is surrounded by other bright pixels, and therefore would have to be compressed differently.

There are two general classes of local operators to be distinguished. The first class divides each pixel's intensity by a local adaptation value. This local adaptation value is then computed as a weighted average of a local neighborhood of pixels. In particular, a low-pass filtered (blurred) image is often used. The general shape of this class of operator is given by [Chiu et al. 1993]:

$$L_d(xx,y) = \frac{L_w(x,y)}{k\, L_w^{\mathrm{LPF}}(x,y)}$$

The scale factor k weights the relative contribution of the low-pass filtered pixel. Dark pixels that are near very bright pixels are darkened because the local average will have a higher value than the pixel itself. This effect is known as a contrast reversal, a halo or a ringing artifact. Artists use contrast reversals to make certain areas of their artworks look brighter than they really are [Chiu et al. 1993]. However, contrast reversals produced by the above approach are rarely as pleasing as those present in artworks. We thus conclude that some contrast reversals are pleasing to the eye, whereas others are distracting.

In the above approach, the magnitude and spatial extent of contrast reversals may be controlled by choosing the size of the low-pass filter kernel as well as the scale factor k. In general, artifacts are minimized if the size of the filter kernel is chosen to be large. Small values of k will also help minimize artifacts. Unfortunately, when artifacts are minimized to the extent that they no longer distract, the amount of compression afforded by the above technique is also reduced such that certain image features are likely to start burning out. Any tone reproduction operator which divides the image by a low-pass filtered version of itself effectively makes the same trade-off [Chiu et al. 1993; Jobson et al. 1995; Rahman et al. 1996; Rahman et al. 1997; Fairchild and Johnson 2002; Moroney and Tastl 2004].

The second class of local tone reproduction operator may be thought of as taking a global tone reproduction operator while replacing the global adaptation level with a spatially localized adaptation level.

A local adaptation level may be computed in several different ways, although one critical observation needs to be made for this approach to become successful. The local adaptation level should be computed over as large a spatial area as possible. However, if this area includes sharp contrasts, this area should be reduced to exclude large gradients. The scale selection mechanism employed in the photographic tone reproduction operator is for instance able to avoid filtering across sharp edges in the image [Reinhard et al. 2002]. Scale selection is also successfully employed in Ashikhmin's operator [Ashikhmin 2002].

Another viable way to compute a useful local adaptation level for each pixel is by applying an edge-preserving smoothing operator and using its output as a local adaptation level. Several edge-preserving smoothing operators may be effectively employed, including the bilateral filter [Tomasi and Manduchi 1998], the adaptive gain control operator [Pattanaik and Yee 2002], the mean shift algorithm [Comaniciu and Meer 2002], or LCIS [Tumblin and Turk 1999].

Sigmoidal tone reproduction curves are good candidates for augmentation for a local adaptation level [Reinhard et al. 2002; Reinhard and Devlin 2004]. Since filtering across large contrast gradients is undesirable, in practice the filter kernel size tends to be rather small — in the order of a couple of pixels wide. This is in stark contrast to the first class of local operators where the image was divided by a low-pass filtered version of itself.

4 discussion

While a large number of tone reproduction operators is currently available, only a small number of fundamentally different approaches exist. In this short survey, the four different approaches are briefly outlined and pointers to additional literature are given. Fourier domain and gradient domain operators are both rooted in knowledge about image formation. Spatial domain operators are either spatially variant (local) or global in nature. These operators are based on insights gained from studying the human visual system (and the visual system of many other species).

Although high dynamic range display devices are just around the corner [Seetzen et al. 2003], we believe that tone reproduction operators will remain important for the foreseeable future. Even if every single monitor is replaced with a high dynamic range display device, hardcopy will always be low dynamic range. With high dynamic range image capture techniques becoming more widespread, high dynamic range imagery will see use in a large selection of different tasks.

A good conclusion to an overview paper such as this one would be to recommend a specific operator. We are fortunate to have a relatively large number of operators at our disposal. We are unfortunate in the sense that validation studies are still in their infancy. Such validation studies are vital to be able to determine which operator is most suited for any particular task. Since validation studies are task-specific, we should first determine which tasks will be most common. We know of only one validation study to date [Drago et al. 2002] which compares tone reproduction operators with respect to similarity and preference. More work is required to determine which operators are good allround performers and which are candidates for specialized tasks. As comparison of tone reproduction operators is a complex issue, we have refrained from providing visual comparison in this survey. We believe that it is too early to draw specific conclusions regarding individual tone reproduction operators.

References

ASHIKHMIN, M. 2002. A tone mapping algorithm for high contrast images. In *Proceedings of 13th Eurographics Workshop on Rendering*, 145–155.

CHIU, K., HERF, M., SHIRLEY, P., SWAMY, S., WANG, C., AND ZIMMERMAN, K. 1993. Spatially nonuniform scaling functions for high contrast images. In *Proceedings of Graphics Interface '93*, 245–253.

CHOUDHURY, P., AND TUMBLIN, J. 2003. The trilateral filter for high contrast images and meshes. In *Proceedings of the Eurographics Symposium on Rendering*, 186–196.

COMANICIU, D., AND MEER, P. 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5, 603–619.

DEBEVEC, P. E., AND MALIK, J. 1997. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, 369–378.

DRAGO, F., MARTENS, W. L., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2002. Perceptual evaluation of tone mapping operators with regard to similarity and preference. Tech. Rep. MPI-I-2002-4-002, Max Plank Institut für Informatik.

DRAGO, F., MYSZKOWSKI, K., ANNEN, T., AND CHIBA, N. 2003. Adaptive logarithmic mapping for displaying high contrast scenes. *Computer Graphics Forum 22*, 3.

DURAND, F., AND DORSEY, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics 21*, 3, 257–266.

FAIRCHILD, M. D., AND JOHNSON, G. M. 2002. Meet iCAM: an image color appearance model. In *IS&T/SID* 10^{th} *Color Imaging Conference*, 33–38.

FATTAL, R., LISCHINSKI, D., AND WERMAN, M. 2002. Gradient domain high dynamic range compression. *ACM Transactions on Graphics 21*, 3, 249–256.

FERWERDA, J. A., PATTANAIK, S., SHIRLEY, P., AND GREENBERG, D. P. 1996. A model of visual adaptation for realistic image synthesis. In *SIGGRAPH 96 Conference Proceedings*, 249–258.

HOOD, D. C., FINKELSTEIN, M. A., AND BUCKINGHAM, E. 1979. Psychophysical tests of models of the response function. *Vision Res 19*, 401–406.

HORN, B. K. P. 1974. Determining lightness from an image. *CVGIP 3*, 277–299.

JOBSON, D. J., RAHMAN, Z., AND WOODELL, G. A. 1995. Retinex image processing: Improved fidelity to direct visual observation. In *Proceedings of the IS&T Fourth Color Imaging Conference: Color Science, Systems, and Applications*, vol. 4, 124–125.

KLEINSCHMIDT, J., AND DOWLING, J. E. 1975. Intracellular recordings from gecko photoreceptors during light and dark adaptation. *J gen Physiol 66*, 617–648.

MORONEY, N., AND TASTL, I. 2004. A comparison of retinex and iCAM for scene rendering. *Journal of Electronic Imaging 13*, 1.

NAKA, K. I., AND RUSHTON, W. A. H. 1966. S-potentials from luminosity units in the retina of fish (cyprinidae). *J Physiol 185*, 587–599.

OPPENHEIM, A. V., SCHAFER, R., AND STOCKHAM, T. 1968. Nonlinear filtering of multiplied and convolved signals. *Proceedings of the IEEE 56*, 8, 1264–1291.

PATTANAIK, S. N., AND YEE, H. 2002. Adaptive gain control for high dynamic range image display. In *Proceedings of Spring Conference in Computer Graphics (SCCG2002)*.

PATTANAIK, S. N., TUMBLIN, J., YEE, H., AND GREENBERG, D. P. 2000. Time-dependent visual adaptation for fast realistic display. In *SIGGRAPH 2000 Conference Proceedings*, 47–54.

PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. 1992. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press.

RAHMAN, Z., JOBSON, D. J., AND WOODELL, G. A. 1996. A multiscale retinex for color rendition and dynamic range compression. In *SPIE Proceedings: Applications of Digital Image Processing XIX*, vol. 2847.

RAHMAN, Z., WOODELL, G. A., AND JOBSON, D. J. 1997. A comparison of the multiscale retinex with other image enhancement techniques. In *IS&T's 50th Annual Conference: A Celebration of All Imaging*, vol. 50, 426–431.

REINHARD, E., AND DEVLIN, K. 2004. Dynamic range reduction inspired by photoreceptor physiology. *Accepted for IEEE Transactions on Visualization and Computer Graphics*.

REINHARD, E., STARK, M., SHIRLEY, P., AND FERWERDA, J. 2002. Photographic tone reproduction for digital images. *ACM Transactions on Graphics 21*, 3, 267–276.

SCHLICK, C. 1994. Quantization techniques for the visualization of high dynamic range pictures. In *Photorealistic Rendering Techniques*, Springer-Verlag Berlin Heidelberg New York, P. Shirley, G. Sakas, and S. Müller, Eds., 7–20.

SEETZEN, H., WHITEHEAD, L. A., AND WARD, G. 2003. A high dynamic range display using low and high resolution modulators. In *The Society for Information Display International Symposium*.

STOCKHAM, T. 1972. Image processing in the context of a visual model. *Proceedings of the IEEE 60*, 7, 828–842.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *Proc. IEEE International Conference on Computer Vision*, 836–846.

TUMBLIN, J., AND TURK, G. 1999. LCIS: A boundary hierarchy for detail-preserving contrast reduction. In *Siggraph 1999, Computer Graphics Proceedings*, Addison Wesley Longman, Los Angeles, A. Rockwood, Ed., Annual Conference Series, 83–90.

WARD, G., RUSHMEIER, H., AND PIATKO, C. 1997. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics 3*, 4.

WARD LARSON, G., AND SHAKESPEARE, R. A. 1998. *Rendering with Radiance*. Morgan Kaufmann Publishers.

WARD, G. J. 1994. The RADIANCE lighting simulation and rendering system. In *Proceedings of SIGGRAPH '94*, A. Glassner, Ed., 459–472.

Image-Based Lighting

Paul Debevec
USC Institute for Creative Technologies

This tutorial shows how image-based lighting can illuminate synthetic objects with measurements of real light, making objects appear as if they're actually in a real-world scene.

Image-based lighting (IBL) is the process of illuminating scenes and objects (real or synthetic) with images of light from the real world. It evolved from the reflection-mapping technique[1,2] in which we use panoramic images as texture maps on computer graphics models to show shiny objects reflecting real and synthetic environments. IBL is analogous to image-based modeling, in which we derive a 3D scene's geometric structure from images, and to image-based rendering, in which we produce the rendered appearance of a scene from its appearance in images. When used effectively, IBL can produce realistic rendered appearances of objects and can be an effective tool for integrating computer graphics objects into real scenes.

The basic steps in IBL are

1. capturing real-world illumination as an omnidirectional, high dynamic range image;
2. mapping the illumination onto a representation of the environment;
3. placing the 3D object inside the environment; and
4. simulating the light from the environment illuminating the computer graphics object.

1 A microscope, modeled by Gary Butcher in 3D Studio Max, rendered using Marcos Fajardo's Arnold rendering system as illuminated by light captured in a kitchen.

Figure 1 shows an example of an object illuminated entirely using IBL. Gary Butcher created the models in 3D Studio Max, and the renderer used was the Arnold global illumination system written by Marcos Fajardo. I captured the light in a kitchen so it includes light from a ceiling fixture; the blue sky from the windows; and the indirect light from the room's walls, ceiling, and cabinets. Gary mapped the light from this room onto a large sphere and placed the model of the microscope on the table in the middle of the sphere. Then, he used Arnold to simulate the object's appearance as illuminated by the light coming from the sphere of incident illumination.

In theory, the image in Figure 1 should look about how a real microscope would appear in that environment. It simulates not just the direct illumination from the ceiling light and windows but also the indirect illumination from the rest of the room's surfaces. The reflections in the smooth curved bottles reveal the kitchen's appearance, and the shadows on the table reveal the colors and spread of the area light sources. The objects also successfully reflect each other, owing to the ray-tracing-based global-illumination techniques we used.

This tutorial gives a basic IBL example using the freely available Radiance global illumination renderer to illuminate a simple scene with several different lighting environments.

Capturing light

The first step in IBL is obtaining a measurement of real-world illumination, also called a light probe image.[3] The easiest way to do this is to download one. There are several available in the Light Probe Image Gallery at http://www.debevec.org/Probes. The Web site includes the kitchen environment Gary used to render the microscope as well as lighting captured in various other interior and outdoor environments. Figure 2 shows a few of these environments.

Light probe images are photographically acquired images of the real world, with two important properties. First, they're omnidirectional—for every direction in the world, there's a pixel in the image that corresponds to that direction. Second, their pixel values are

0272-1716/02/$17.00 © 2002 IEEE

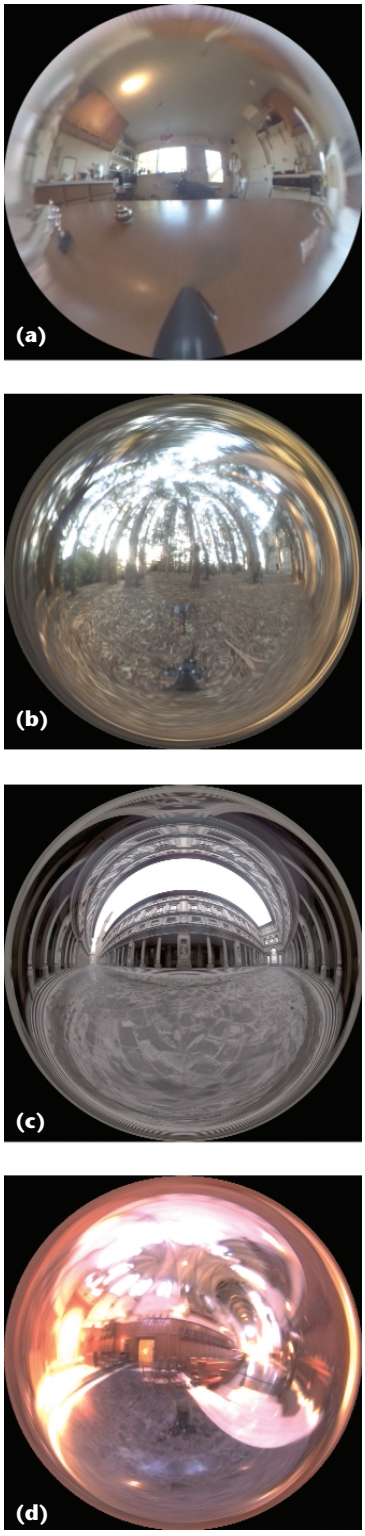
linearly proportional to the amount of light in the real world. In the rest of this section, we'll see how to take images satisfying both of these properties.

We can take omnidirectional images in a number of ways. The simplest way is to use a regular camera to take a photograph of a mirrored ball placed in a scene. A mirrored ball actually reflects the entire world around it, not just the hemisphere toward the camera. Light rays reflecting off the outer circumference of the ball glance toward the camera from the back half of the environment. Another method of obtaining omnidirectional images using a regular camera is to shoot a mosaic of many pictures looking in different directions and combine them using an image stitching program such as RealViz's Stitcher. A good way to cover a particularly large area in each shot is to use a fisheye lens,[4] which lets us cover the full field in as few as two images. A final technique is to use a special scanning panoramic camera (such as the ones Panoscan and Sphereon make), which uses a vertical row of image sensors on a rotating camera head to scan across a 360-degree field of view.

In most digital images, pixel values aren't proportional to the light levels in the scene. Usually, light levels are encoded nonlinearly so they appear either more correctly or more pleasingly on nonlinear display devices such as cathode ray tubes. Furthermore, standard digital images typically represent only a small fraction of the dynamic range—the ratio between the dimmest and brightest regions accurately represented—present in most real-world lighting environments. When part of a scene is too bright, the pixels saturate to their maximum value (usually 255) no matter how bright they really are.

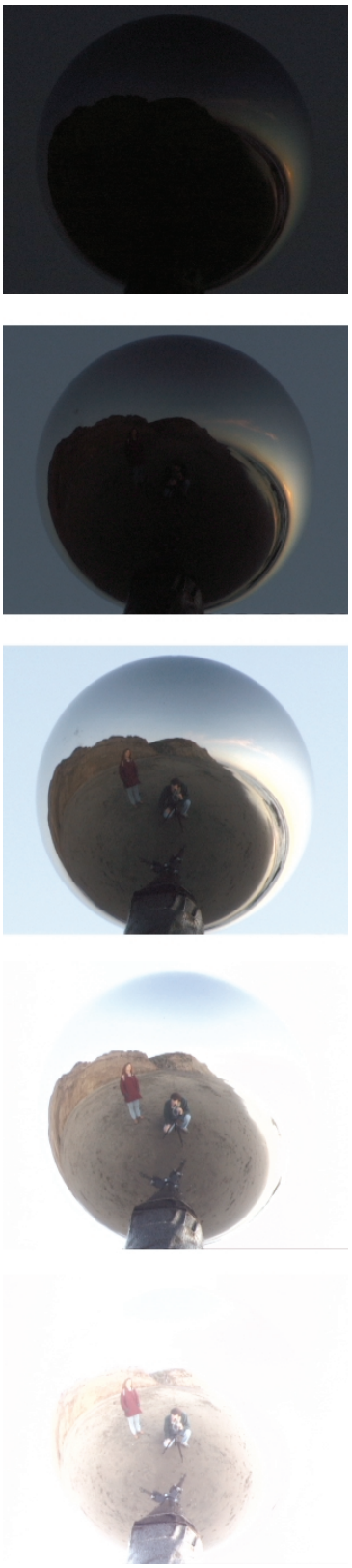
We can ensure that the pixel values in the omnidirectional images are truly proportional to quantities of light using high dynamic range (HDR) photography techniques.[5] The process typically involves taking a series of pictures of the scene with varying exposure levels and then using these images to solve for the imaging system's response curve and to form a linear-response composite image covering the entire range of illumination values in the scene. Software for assembling images in this way includes the command-line mkhdr program at http://www.debevec.org/Research/HDR and the Windows-based HDR Shop program at http://www.debevec.org/HDRShop.

HDR images typically use a single-precision floating-point number for red, green, and blue, allowing the full range of light from thousandths to billions to be represented. We can store HDR image data in a various file formats, including the floating-point version of the TIFF file format or the Portable Floatmap variant of Jef Postsanzer's Portable Pixmap format. Several other representations that use less storage are available, including Greg Ward's Red-Green-Blue Exponent (RGBE) format[6] (which uses one byte each for red, green, blue and a common 8-bit exponent) and his new 24-bit and 32-bit LogLuv formats recently included in the TIFF standard. The light probe images in the light probe image gallery are in the RGBE format, which lets us easily use them in Ward's Radiance global illumination renderer. (We'll see how to do precisely that in the next section.)



2 Several light probe images from the Light Probe Image Gallery at http://www.debevec.org/Probes. The light is from (a) a residential kitchen, (b) the eucalyptus grove at UC Berkeley, (c) the Uffizi gallery in Florence, Italy, and (d) Grace Cathedral in San Francisco.

Figure 3 (next page) shows a series of images used in creating a light probe image. To acquire these images, we placed a three-inch polished ball bearing on top of a tripod at Funston Beach near San Francisco and used a digital camera with a telephoto zoom lens to take a series of exposures of the ball. Being careful not to disturb the camera, we took pictures at shutter speeds ranging from 1/4 second to 1/10000 second, allowing



3 A series of differently exposed images of a mirrored ball photographed at Funston Beach near San Francisco. I merged the exposures, ranging in shutter speed from 1/4 second to 1/1000 second, into a high dynamic range image so we can use it as an IBL environment.

the camera to properly image everything from the dark cliffs to the bright sky and the setting sun. We assembled the images using code similar to that now found in mkhdr and HDR Shop, yielding a high dynamic range, linear-response image.

Illuminating synthetic objects with real light

IBL is now supported by several commercial renderers, including LightWave 3D, Entropy, and Blender. For this tutorial, we'll use the freely downloadable Radiance lighting simulation package written by Greg Ward at Lawrence Berkeley Laboratories. Radiance is a Unix package, which means that to use it you'll need to use a computer running Linux or an SGI or Sun workstation. In this tutorial, we'll show how to perform IBL to illuminate synthetic objects in Radiance in just seven steps.

1. Download and install Radiance

First, test to see if you already have Radiance installed by typing `which rpict` at a Unix command prompt. If the shell returns "Command not found," you'll need to install Radiance. To do this, visit the Radiance Web site at http://radsite.lbl.gov/radiance and click on the download option. As of this writing, the current version is 3.1.8, and it's precompiled for SGI and Sun workstations. For other operating systems, such as Linux, you can download the source files and then compile the executable programs using the `makeall` script. Once installed, make sure that the Radiance binary directory is in your `$PATH` and that your `$RAYPATH` environment variable includes the Radiance library directory. Your system administrator should be able to help you if you're not familiar with installing software packages on Unix.

2. Create the scene

The first thing we'll do is create a Radiance scene file. Radiance scene files contain the specifications for your scene's geometry, reflectance properties, and lighting. We'll create a simple scene with a few spheres sitting on a platform. First, let's specify the material properties we'll use for the spheres. Create a new directory and then call up your favorite text editor to type in the following material specifications to the file scene.rad:

```
# Materials

void plastic red_plastic
0
0
5 .7 .1 .1 .06 .1

void metal steel
0
0
5 0.6 0.62 0.68 1 0

void metal gold
0
0
5 0.75 0.55 0.25 0.85 0.2

void plastic white_matte
0
0
5 .8 .8 .8 0 0
```

```
rview -vtv -vp 8 2.5 -1.5 -vd -8 -2.5 1.5 -vu 0 1 0 -vh 60 -vv 40
```

4 Use your text editor to create the file camera.vp with the camera parameters as the file's first and only line.

```
void dielectric crystal
0
0
5 .5 .5 .5 1.5 0

void plastic black_matte
0
0
5 .02 .02 .02 .00 .00

void plastic gray_plastic
0
0
5 0.25 0.25 0.25 0.06 0.0
```

These lines specify the diffuse and specular characteristics of the materials we'll use in our scene, including crystal, steel, and red plastic. In the case of the red plastic, the diffuse RGB color is (.7, .1, .1), the proportion of light reflected specularly is .06, and the specular roughness is .1. The two zeros and the five on the second through fourth lines are there to tell Radiance how many alphanumeric, integer, and floating-point parameters to expect.

Now let's add some objects with these material properties to our scene. The objects we'll choose will be some spheres sitting on a pedestal. Add the following lines to the end of scene.rad:

```
# Objects

red_plastic sphere ball0
0
0
4 2 0.5 2 0.5

steel sphere ball1
0
0
4 2 0.5 -2 0.5

gold sphere ball2
0
0
4 -2 0.5 -2 0.5

white_matte sphere ball3
0
0
4 -2 0.5 2 0.5

crystal sphere ball4
0
0
4 0 1 0 1

!genworm black_matte twist \
 "cos(2*PI*t)*(1+0.1*cos(30*PI*t))" \
 "0.06+0.1+0.1*sin(30*PI*t)" \
 "sin(2*PI*t)*(1+0.1*cos(30*PI*t))" \
 "0.06" 200 | xform -s 1.1 -t 2 0 2 \
 -a 4 -ry 90 -i 1

!genbox gray_plastic pedestal_top 8 \
 0.5 8 -r 0.08 | xform -t -4 -0.5 \
 -4
!genbox gray_plastic pedestal_shaft \
 6 16 6 | xform -t -3 -16.5 -3
```

These lines specify five spheres made from various materials sitting in an arrangement on the pedestal. The first sphere, **ball0**, is made of the **red_plastic** material and located in the scene at (2,0.5,2) with a radius of 0.5. The pedestal itself is composed of two beveled boxes made with the Radiance **genbox** generator program. In addition, we invoke the **genworm** program to create some curly iron rings around the spheres. (You can leave the **genworm** line out if you want to skip some typing; also, the backslashes indicate line continuations which you can omit if you type everything on one line.)

3. Add a traditional light source

Next, let's add a traditional light source to the scene to get our first illuminated glimpse —without IBL—of what the scene looks like. Add the following lines to scene.rad to specify a traditional light source:

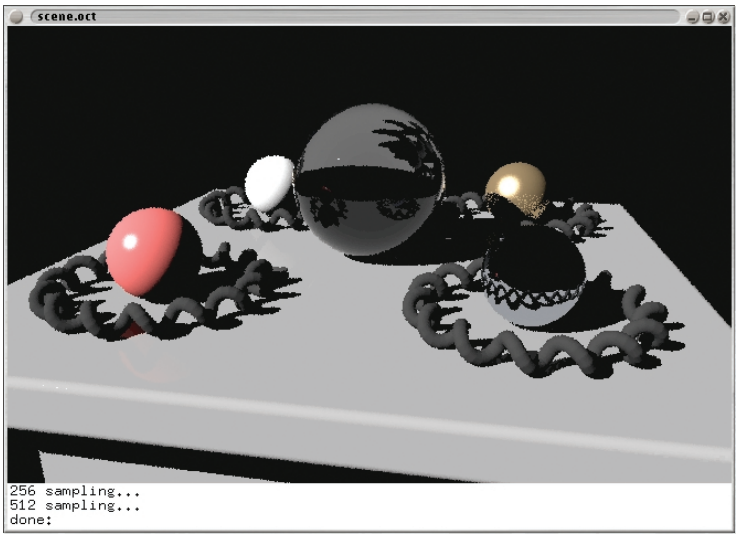
```
# Traditional Light Source

void light lightcolor
0
0
3 10000 10000 10000

lightcolor source lightsource
0
0
4 1 1 1 2
```

4. Render the scene with traditional lighting

In this step, we'll create an image of the scene. First, we need to use the **oconv** program to process the scene file into an octree file for Radiance to render. Type the following command at the Unix command prompt:

```
# oconv scene.rad > scene.oct
```

The **#** indicates the prompt, so you don't need to type it. This will create an octree file scene.oct that can be rendered in Radiance's interactive renderer **rview**. Next, we need to specify a camera position. This can be done as command arguments to **rview**, but to make things



5 The Radiance rview interactive renderer viewing the scene as illuminated by a traditional light source.

simpler, let's store our camera parameters in a file. Use your text editor to create the file camera.vp with the camera parameters as the file's first and only line (see Figure 4). In the file, this should be typed as a single line.

These parameters specify a perspective camera (**-vtv**) with a given viewing position (**-vp**), direction (**-vd**), and up vector (**-vu**) and with horizontal (**-vh**) and vertical (**-vv**) fields of view of 60 and 40 degrees, respectively. (The **rview** text at the beginning of the line is a standard placeholder in Radiance camera files, not an invocation of the **rview** executable.)

Now let's render the scene in **rview**. Type:

```
# rview -vf camera.vp scene.oct
```

In a few seconds, you should get an image window similar to the one in Figure 5. The image shows the spheres on the platform, surrounded by the curly rings, and illuminated by the traditional light source. The image might or might not be pleasing, but it certainly looks computer generated. Now let's see if we can make it more realistic by lighting the scene with IBL.

5. Download a light probe image

Visit the Light Probe Image Gallery at http://www.debevec.org/Probes and choose a light probe image to download. The light probe images without concentrated light sources tend to produce good-quality renders more quickly, so I'd recommend starting with the beach, uffizi, or kitchen probes. Here we'll choose the beach probe for the first example. Download the beach_probe.hdr file by shift-clicking or right-clicking "Save Target As..." or "Save Link As..." and then view it using the Radiance image viewer **ximage**:

```
# ximage beach_probe.hdr
```

If the probe downloaded properly, a window should pop up displaying the beach probe image. While the window is up, you can click and drag the mouse pointer over a region of the image and then press "=" to re-expose the image to properly display the region of interest. If the image didn't download properly, try downloading and expanding the all_probes.zip or all_probes.tar.gz archive from the same Web page, which will download all the light probe images and preserve their binary format. When you're done examining the light probe image, press the "q" key in the **ximage** window to dismiss the window.

6. Map the light probe image onto the environment

Let's now add the light probe image to our scene by mapping it onto an environment surrounding our objects. First, we need to create a new file that will specify the mathematical formula for mapping the light probe image onto the environment. Use your text editor to create the file angmap.cal with the following content (the text between the curly braces is a comment that you can skip typing if you wish):

```
{
angmap.cal

Convert from directions in the world \
 (Dx, Dy, Dz) into (u,v) \
 coordinates on the light probe \
 image

-z is forward (outer edge of sphere)
+z is backward (center of sphere)
+y is up (toward top of sphere)
}

d = sqrt(Dx*Dx + Dy*Dy);

r = if(d, 0.159154943*acos(Dz)/d,0);

u = 0.5 + Dx * r;
v = 0.5 + Dy * r;
```

This file will tell Radiance how to map direction vectors in the world (Dx, Dy, Dz) into light probe image coordinates (u, v). Fortunately, Radiance accepts these coordinates in the range of zero to one (for square images) no matter the image size, making it easy to try out light probe images of different resolutions. The formula converts from the angular map version of the light probe images in the light probe image gallery, which differs from the mapping a mirrored ball produces. If you need to convert a mirrored-ball image to this format, HDR Shop has a Panoramic Transformations function for this purpose.

Next, comment out (by adding **#**'s at the beginning of the lines) the traditional light source in scene.rad that we added in step 3:

```
#lightcolor source lightsource
#0
#0
#4 1 1 1 2
```

Note that these aren't new lines to add to the file but lines to modify from what you've already entered. Now, add the following to the end of scene.rad to include the IBL environment:

```
# Image-Based Lighting Environment

void colorpict hdr_probe_image
7 red green blue beach_probe.hdr
 angmap.cal u v
0
0

hdr_probe_image glow light_probe
0
0
4 1 1 1 0

light_probe source ibl_environment
0
0
4 0 1 0 360
```

The **colorpict** sequence indicates the name of the light probe image and the calculations file to use to map directions to image coordinates. The **glow** sequence specifies a material property comprising the light probe image treated as an emissive glow. Finally, the **source** specifies the geometry of an infinite sphere mapped with the emissive glow of the light probe. When Radiance's rays hit this surface, their illumination contribution will be taken to be the light specified for the corresponding direction in the light probe image.

Finally, because we changed the scene file, we need to update the octree file. Run **oconv** once more to do this:

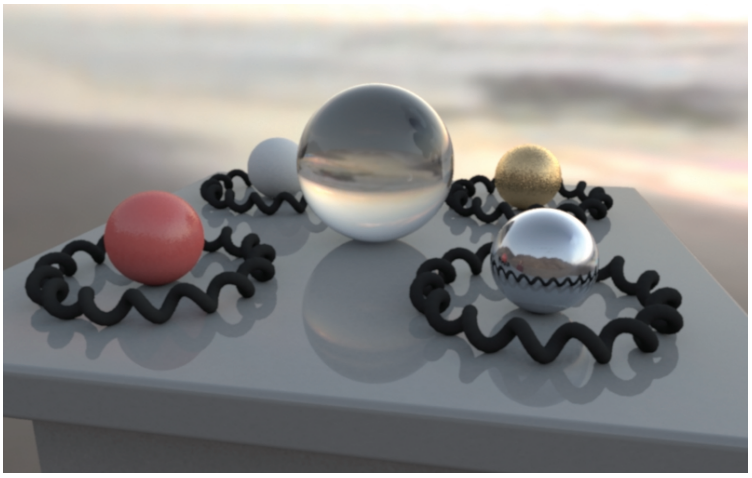
```
# oconv scene.rad > scene.oct
```

7. Render the scene with IBL

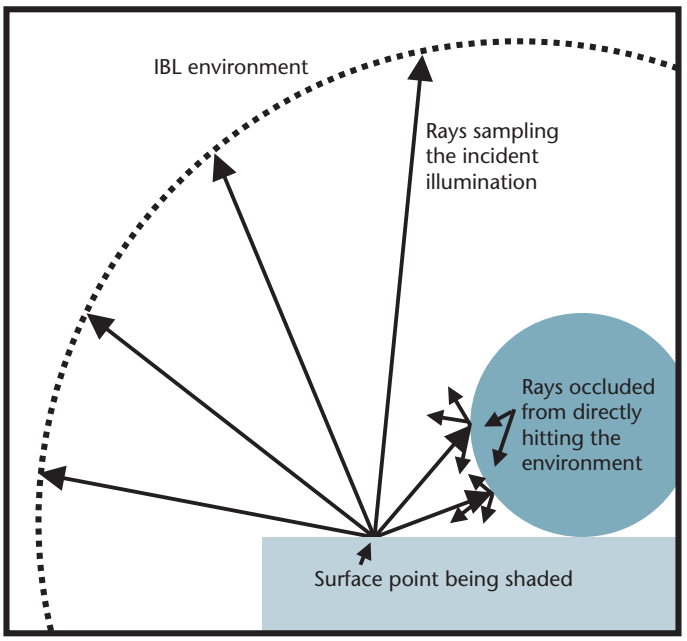
Let's now render the scene using IBL. Enter the following command to bring up a rendering in **rview**:

```
# rview -ab 1 -ar 5000 -aa 0.08 -ad \
 128 -as 64 -st 0 -sj 1 -lw 0 -lr \
 8 -vf camera.vp scene.oct
```

Again, you can omit the backslashes if you type the whole command as one line. In a few moments, you should see the image in Figure 6 begin to take shape. Radiance is tracing rays from the camera into the scene, as Figure 7 illustrates. When a ray hits the environment, it takes as its pixel value the corresponding value in the light probe image. When a ray hits a particular point on an object, Radiance calculates the color and intensity of the incident illumination (also known as irradiance) on that point by sending out a multitude of rays (in this case 192 of them) in random directions to quantify the light arriving at that point on the object. Some of these rays will hit the environment, and others will hit other parts of the object, causing Radiance to recurse into computing the light coming from this new part of the object. After Radiance computes the illumination on the object point, it calculates the light reflected toward the camera based on the object's material properties and this becomes the pixel value of that point of the object. Images calculated in this way can take a while to render, but they produce a faithful rendition of how the captured illumination would illuminate the objects.



6 The spheres on the pedestal illuminated by the beach light probe image from Figure 3.



7 How Radiance traces rays to determine the incident illumination on a surface from an IBL environment.

The command-line arguments to **rview** tell Radiance how to perform the lighting calculations. The **-ab 1** indicates that Radiance should produce only one ambient-bounce recursion in computing the object's illumination—more accurate simulations could be produced with a value of 2 or higher. The **-ar** and **-aa** set the resolution and accuracy of the surface illumination calculations, and the **-ad** and **-as** set the number of rays traced out from a surface point to compute its illumina-

8 The objects illuminated by the kitchen, eucalyptus grove, Uffizi Gallery, and Grace Cathedral light probe images in Figure 2.

tion. The `-st`, `-sj`, `-lw`, and `-lr` specify how the rays should be traced for glossy and shiny reflections. For more information on these and more Radiance parameters, see the reference guide on the Radiance Web site.

When the render completes, you should see an image of the objects as illuminated by the beach lighting environment. The synthetic steel ball reflects the environment and the other objects directly. The glass ball both reflects and refracts the environment, and the diffuse white ball shows subtle shading, which is lighter toward the sunset and darkest where the ball contacts the pedestal. The rough specular reflections in the red and gold balls appear somewhat speckled in this medium-resolution rendering; the reason is that Radiance sends out just one ray for each specular sample (regardless of surface roughness) rather than the much greater number it sends out to compute the diffuse illumination. Rendering at a higher resolution and filtering the image down can alleviate this effect.

We might want to create a particularly high-quality rendering using the command-line `rpict` renderer, which outputs the rendered image to a file. Run the following `rpict` command:

```
# rpict -x 800 -y 800 -t 30 -ab 1 - \
 ar 5000 -aa 0.08 -ad 128 -as 64 - \
 st 0 -sj 1 -lw 0 -lr 8 -vf \
 camera.vp scene.oct > render.hdr
```

The command-line arguments to `rpict` are identical to `rview` except that one also specifies the maximum x and y resolutions for the image (here, 800×800 pixels) as well as how often to report back on the rendering progress (here, every 30 seconds.) On an 800-MHz computer, this should take approximately 10 minutes. When it completes, we can only view the rendered output image with the ximage program. To produce high-quality renderings, you can increase the x and y resolutions to high numbers, such as $3{,}000 \times 3{,}000$ pixels and then filter the image down to produce an antialiased rendering. We can perform this filtering down by using either Radiance's `pfilt` command or the HDR Shop. To filter a $3{,}000 \times 3{,}000$ pixel image down to $1{,}000 \times 1{,}000$ pixels using `pfilt`, enter:

```
# pfilt -1 -x /3 -y /3 -r 1 \
 render.hdr > filtered.hdr
```

I used this method for the high-quality renderings in this article. To render the scene with different lighting environments, download a new probe image, change the beach_probe.hdr reference in the scene.rad file, and call `rview` or `rpict` once again. Light probe images with concentrated light sources such as grace and stpeters will require increasing the `-ad` and `-as` sampling parameters to the renderer to avoid mottled renderings. Figure 8 shows renderings of the objects illuminated by the light probes in Figure 2. Each rendering shows different effects of the lighting, from the particularly soft shadows under the spheres in the overcast Uffizi environment to the focused pools of light from the stained glass windows under the glass ball in the Grace Cathedral environment.

Advanced IBL

This tutorial has shown how to illuminate synthetic objects with measurements of real light, which can help the objects appear as if they're actually in a real-world scene. We can also use the technique to light large-scale environments with captured illumination from real-world skies. Figure 9 shows a computer

9 A computer model of the ruins of the Parthenon as illuminated just after sunset by a sky captured in Marina del Rey, California. Modeled by Brian Emerson and Yikuong Chen and rendered using the Arnold global illumination system.

10 A rendering from the Siggraph 99 film *Fiat Lux*, which combined image-based modeling, rendering, and lighting to place monoliths and spheres into a photorealistic reconstruction of St. Peter's Basilica.

model of a virtual environment of the Parthenon illuminated by a real-world sky captured with high dynamic range photography.

We can use extensions of the basic IBL technique in this article to model illumination emanating from a geometric model of the environment rather than from an infinite sphere of illumination and to have the objects cast shadows and appear in reflections in the environment. We used these techniques[3] to render various animated synthetic objects into an image-based model of St. Peter's Basilica for the Siggraph 99 film *Fiat Lux*, (see Figure 10). (You can view the full animation at http://www.debevec.org/FiatLux/.)

Some more recent work[7] has shown how to use IBL to illuminate real-world objects with captured illumination. The key to doing this is to acquire a large set of images of the object as illuminated by all possible lighting directions. Then, by taking linear combinations of the color channels of these images, images can be produced showing the objects under arbitrary colors and intensities of illumination coming simultaneously from all possible directions. By choosing the colors and intensities of the incident illumination to correspond to those in a light probe image, we can show the objects as they would be illuminated by the captured lighting environment, with no need to model the objects' geometry or reflectance properties. Figure 11 shows a collection of real objects illuminated by two of the light probe images from Figure 2. In these renderings, we used the additional image-based technique of environment matting[8] to compute high-resolution refractions and reflections of the background image through the objects.

11 Real objects illuminated by the Eucalyptus grove and Grace Cathedral lighting environments from Figure 2.

Conclusion

IBL lets us integrate computer-generated models into real-world environments according to the principles of global illumination. It requires a few special practices for us to apply it, including taking omnidirectional photographs, recording images in high dynamic range, and including measurements of incident illumination as sources of illumination in computer-generated scenes. After some experimentation and consulting the Radiance reference manual, you should be able to adapt these examples to your own scenes and applications. With a mirrored ball and a digital camera, you should be able to acquire your own lighting environments as well. For more information, please explore the course notes for the Siggraph 2001 IBL course at http://www.debevec.org/IBL2001. Source files and more image-based lighting examples are available at http://www.debevec.org/CGAIBL. ■

References

1. J.F. Blinn, "Texture and Reflection in Computer Generated Images," *Comm. ACM*, vol. 19, no. 10, Oct. 1976, pp. 542-547.
2. G.S. Miller and C.R. Hoffman, "Illumination and Reflection Maps: Simulated Objects in Simulated and Real Environments," *Proc. Siggraph 84*, Course Notes for Advanced Computer Graphics Animation, ACM Press, New York, 1984.
3. P. Debevec, "Rendering Synthetic Objects Into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography," *Computer Graphics* (Proc. Siggraph 98), ACM Press, New York, 1998, pp. 189-198.
4. N. Greene, "Environment Mapping and Other Applications of World Projections," *IEEE Computer Graphics and Applications*, vol. 6, no. 11, Nov. 1986, pp. 21-29.
5. P.E. Debevec and J. Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," *Computer Graphics* (Proc. Siggraph 97), ACM Press, New York, 1997, pp. 369-378.
6. G. Ward, "Real Pixels," *Graphics Gems II*, J. Arvo, ed., Academic Press, Boston, 1991, pp. 80-83.
7. P. Debevec et. al, "Acquiring the Reflectance Field of a Human Face," *Computer Graphics* (Proc. Siggraph 2000), ACM Press, New York, 2000, pp. 145-156.
8. D.E. Zongker et. al, "Environment Matting and Compositing," *Computer Graphics* (Proc. Siggraph 99), ACM Press, New York, 1999, pp. 205-214.

Paul Debevec *is an executive producer 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. For the past five years, he 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. He has a BS in math and a BSE in computer engineering from the University of Michigan and a PhD in computer science from University of California, Berkeley. In August 2001, he received the Significant New Researcher Award from ACM Siggraph for his innovative work in the image-based modeling and rendering field. He is a member of ACM Siggraph and the Visual Effects Society, and is a program cochair for the 2002 Eurographics Workshop on Rendering.*

Readers may contact Paul Debevec at USC/ICT, 13274 Fiji Way, 5th Floor, Marina Del Rey, CA 90292, email paul@debevec.org.

High Dynamic Range Imaging

Paul Debevec
USC ICT

Erik Reinhard
University of Central Florida

Greg Ward
Anyhere Software

Sumanta Pattanaik
University of Central Florida

Images

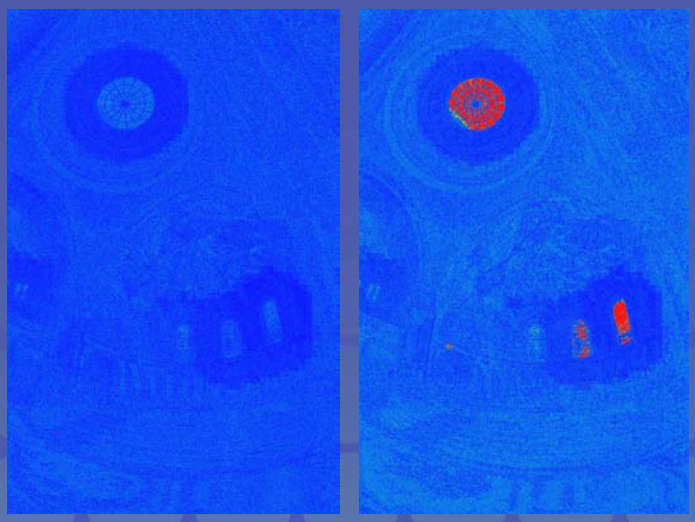
SIGGRAPH2004

- Traditionally we use one byte per pixel per color channel
- What of we had floating point numbers to represent colored pixels?
- This may seem a small issue, but really isn't…

HDR Image capture

HDR Image capture

HDR File formats



HDR Display devices

Tone reproduction

Image-based lighting

Overview

Course schedule

– Introduction – Erik Reinhard
(almost finished)

– HDR Image capture – Paul Debevec
8:40 – 9:20

– HDR is as easy as 1-2-3 – Greg Ward
9:20 – 10:00

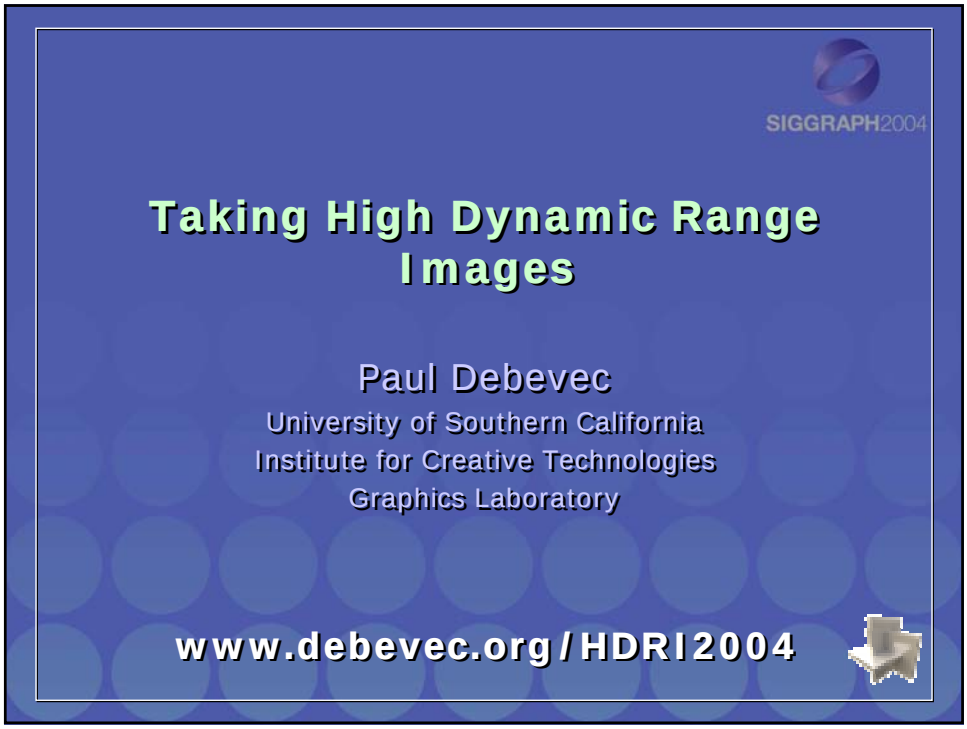
– Dynamic range – Sumanta Pattanaik
10:15 – 10:50

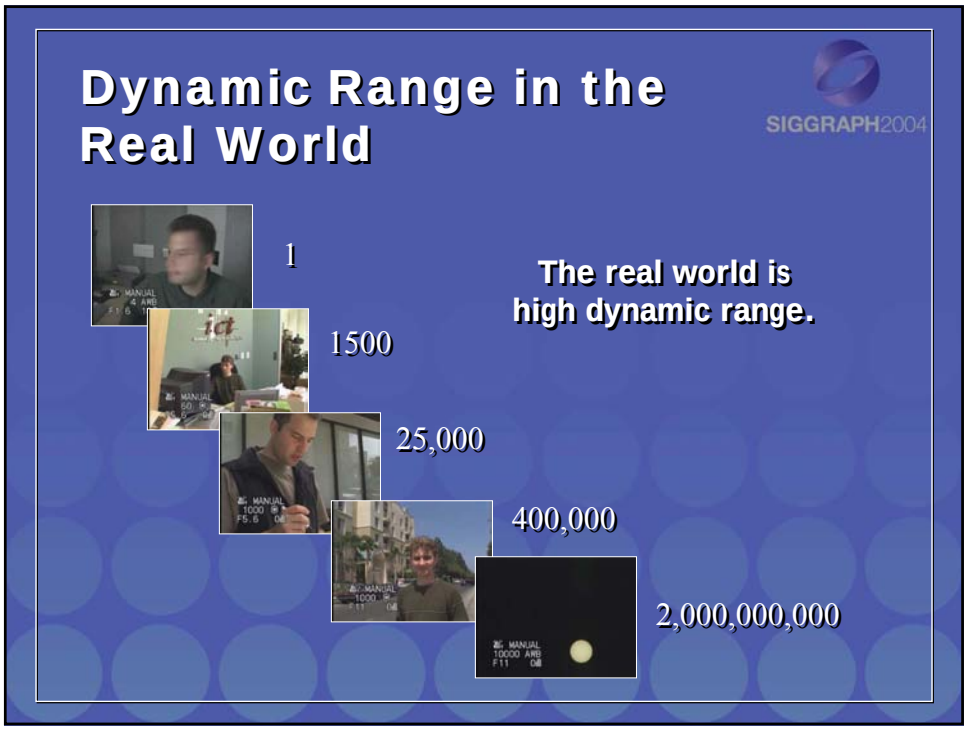
Course schedule

– Tone reproduction operators – Erik Reinhard
10:50 – 11:25

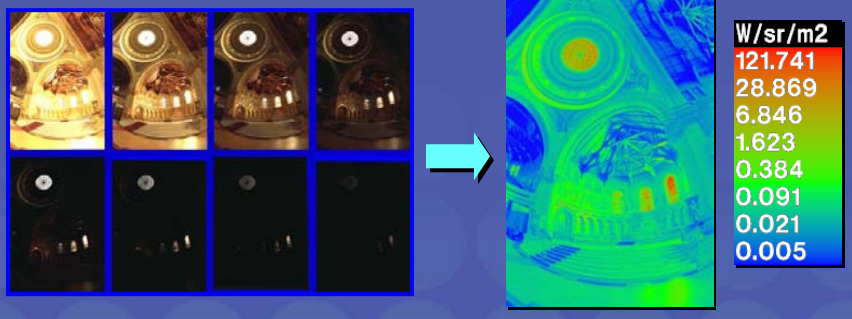
– HDR Image-based lighting – Paul Debevec
11:25 – 12:05

– Discussion – All
12:05 – 12:15


SIGGRAPH2004
Taking High Dynamic Range Images
Paul Debevec
University of Southern California
Institute for Creative Technologies
Graphics Laboratory
www.debevec.org/HDRI2004


Dynamic Range in the Real World
SIGGRAPH2004
1
1500
25,000
400,000
2,000,000,000
The real world is high dynamic range.

High-Dynamic Range Photography



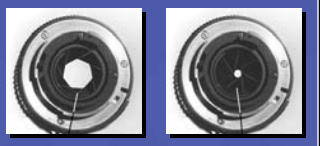
300,000 : 1

Visualization: Greg Ward

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

Ways to vary exposure

- Shutter Speed (*)
- F/stop (aperture, iris)
- Neutral Density (ND) Filters



Shutter Speed

Ranges: Canon D30: 30 to 1/4,000 sec.
Sony VX2000: ¼ to 1/10,000 sec.

Pros:
- Directly varies the exposure
- Usually accurate and repeatable

Issues:
- Digital: Noise in long exposures
- Film: Reciprocity failure at > ~ 5 sec.

Shutter Speed

Note: shutter times usually obey a power series – each "stop" is a factor of 2

¼, 1/8, 1/15, 1/30, 1/60, 1/125, 1/250, 1/500, 1/1000 sec

Usually really is:

¼, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 sec

Measuring Shutter Speed

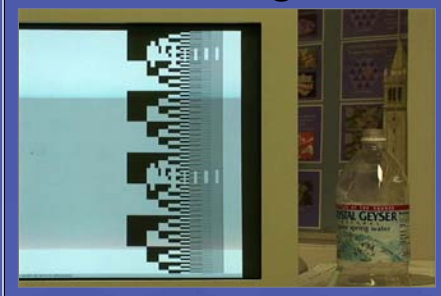
SIGGRAPH2004

For SIGGRAPH 97 HDR paper, made digital recordings of the shutter clicking

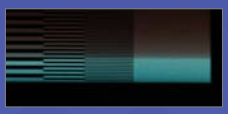
Can measure shorter speeds using a CRT…

Measuring Shutter Speed

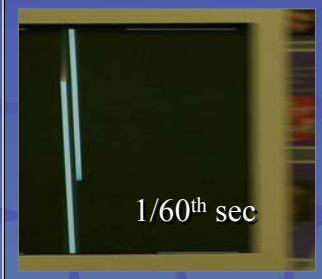
SIGGRAPH2004



1/500th sec

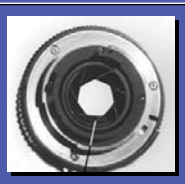


1/4000th sec



1/8th sec

F/stop (aperture)




Ranges: Canon D30: f/2.8 to f/22
Sony VX2000: f/1.6 to f/11

Standard f-numbers: 2, 2.8, 4, 5.6, 8, 11, 16, 22

Exposure is proportional to the inverse square of the f-number:

f/22 is 1/64 the light of f/2.8 (6 stops)

Pros:

Can use aperture when you run out of shutter speed variation

F/stop (aperture)

SIGGRAPH2004

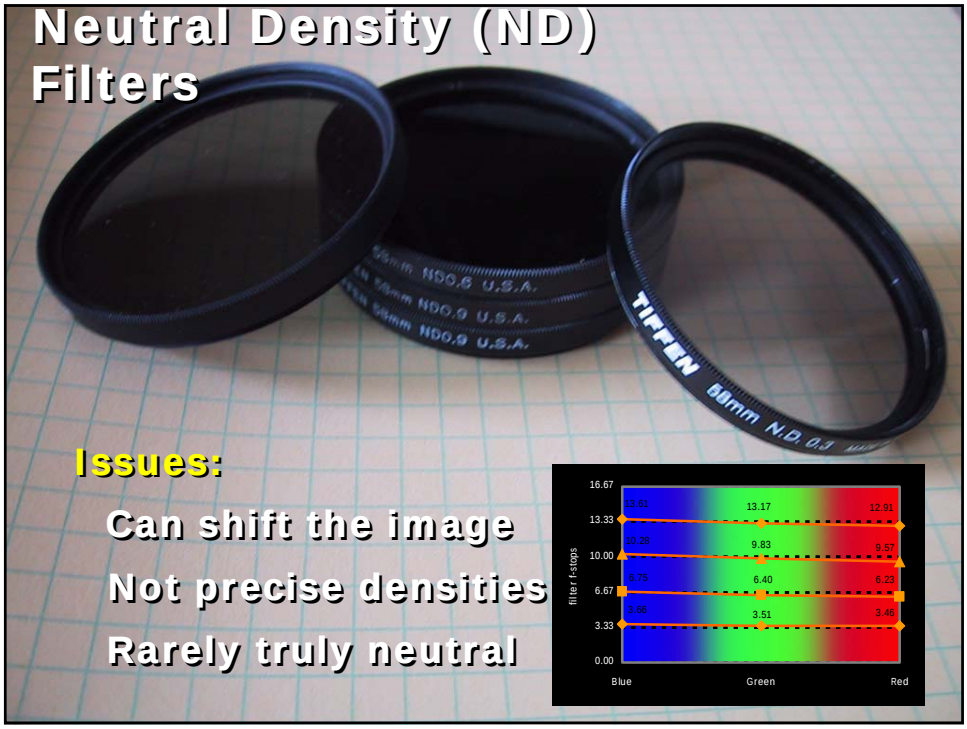

Issues:

Changes depth of field
Not very repeatable
Limited range of exposure variation
=> Not recommended for HDR


Neutral Density (ND) Filters
Ranges: 0.1 to 4.0 density
(0.3, 0.6, 0.9 density = 1, 2, 3 stops common)
Log base 10 scale:
Density of 0.3 = ½ the light (1 stop)
Density of 1.0 = 1/10 the light
Density of 4.0 = 1/10,000 the light
Drawback: Not perfectly neutral


Neutral Density (ND) Filters
Pros:
Can be stacked => very wide range
Work for strobe lighting

Neutral Density (ND) Filters



Issues:

- Can shift the image
- Not precise densities
- Rarely truly neutral

Gain / ISO / Film Speed

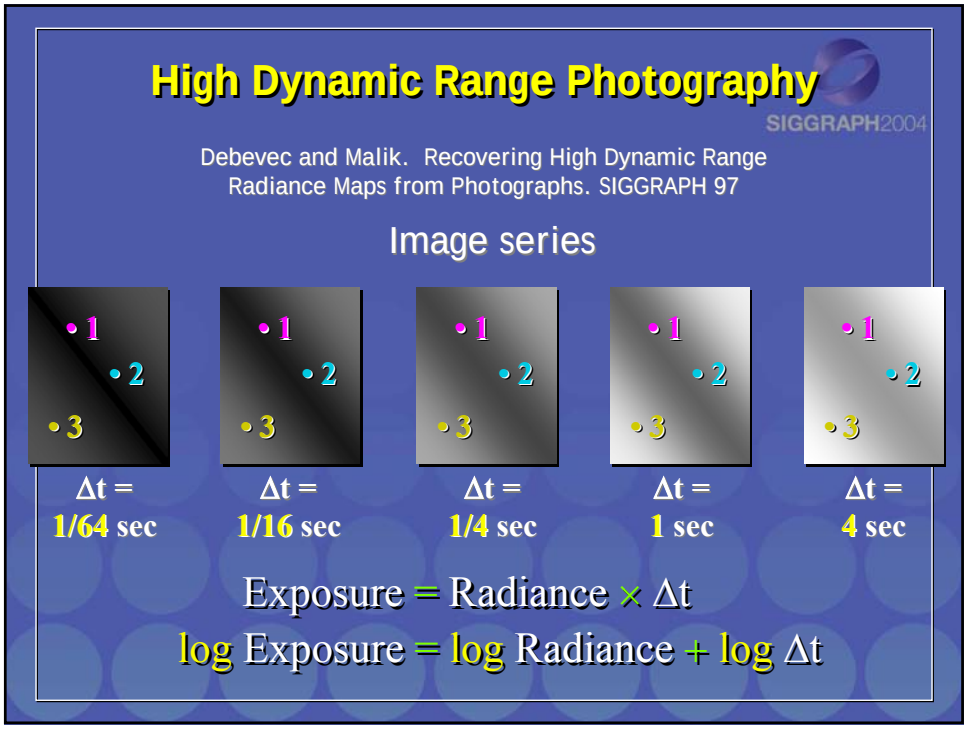
SIGGRAPH2004

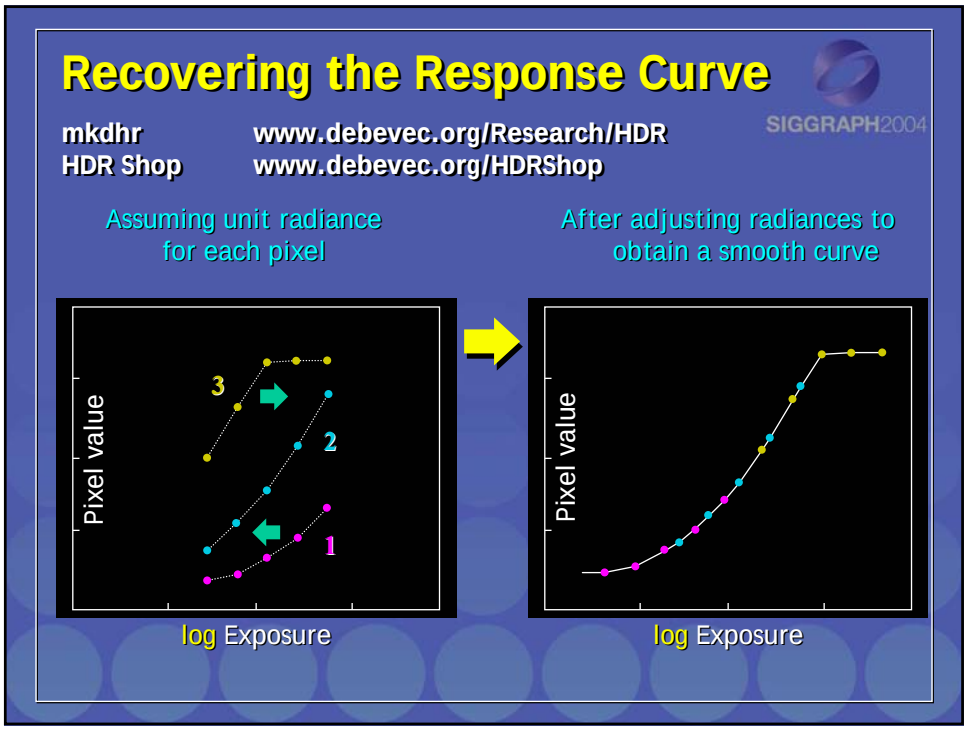
Range: ISO 100 to 1600
0dB to 18dB (3dB = factor of 2)

Issue:

...adds noise to your

+18dB Gain on Sony VX2000


High Dynamic Range Photography
SIGGRAPH2004
Debevec and Malik. Recovering High Dynamic Range Radiance Maps from Photographs. SIGGRAPH 97
Image series
• 1
• 2
• 3
Δt = 1/64 sec
Δt = 1/16 sec
Δt = 1/4 sec
Δt = 1 sec
Δt = 4 sec
Exposure = Radiance × Δt
log Exposure = log Radiance + log Δt


Recovering the Response Curve
SIGGRAPH2004
mkdhr www.debevec.org/Research/HDR
HDR Shop www.debevec.org/HDRShop
Assuming unit radiance for each pixel
After adjusting radiances to obtain a smooth curve
Pixel value
3
2
1
log Exposure
Pixel value
log Exposure

H D R S h o p

High Dynamic Range Image Processing and Manipulation

www.debevec.org/HDRShop

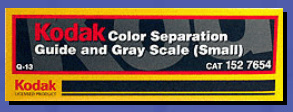
Introduction | Tutorials | Reference | Plugins | FAQ | Download/Licensing | WWW Links | Mailing List

Chris Tchou et al. *HDR Shop*. S2001 Technical Sketch

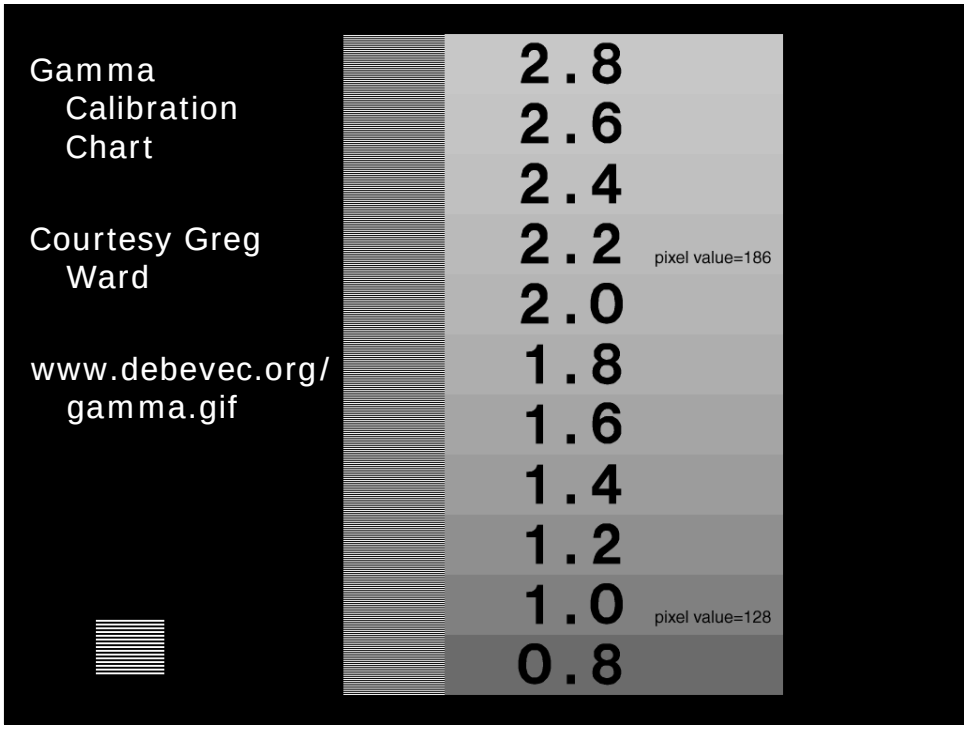
Gray Scale Calibration Charts

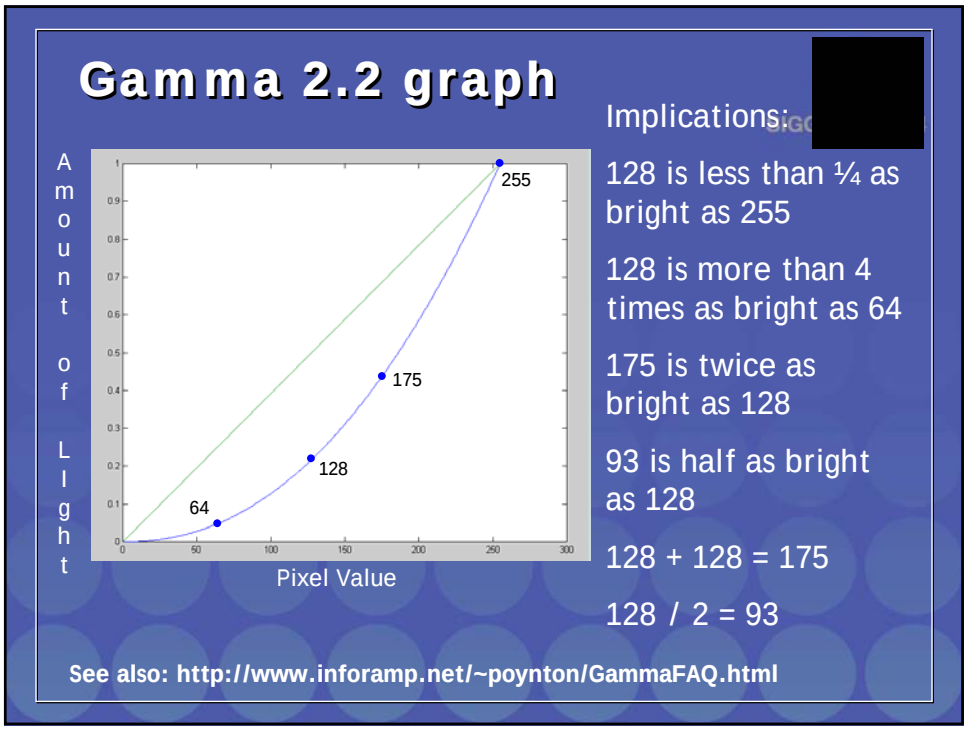
SIGGRAPH2004

Twenty 0.1 density increments =
26% more reflective each step = 1/3 stop



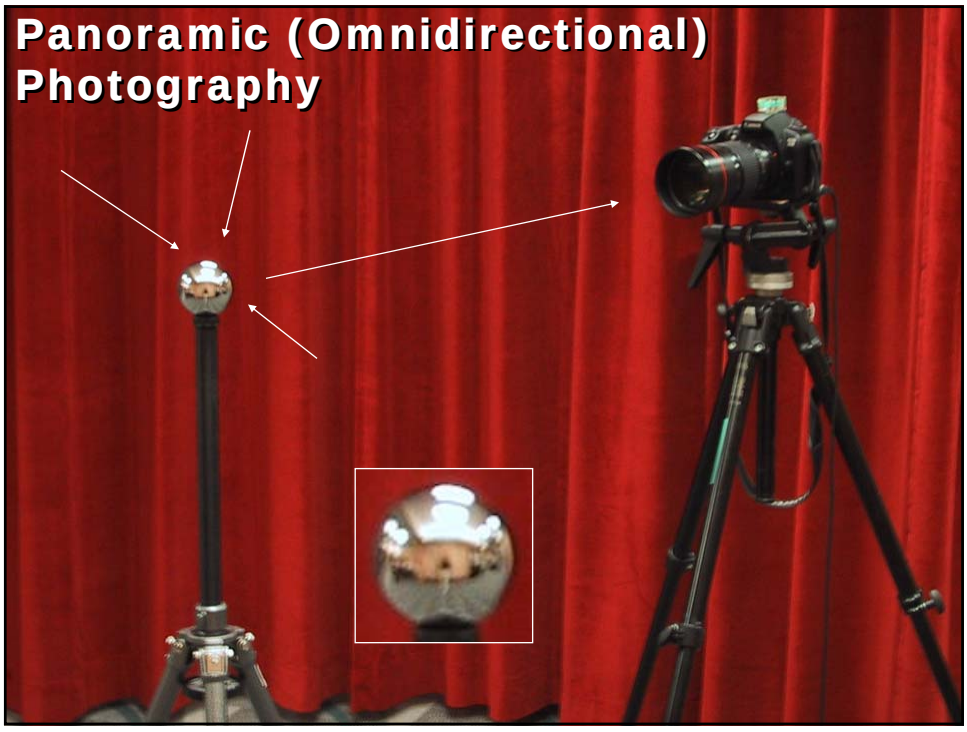
www.bhphotovideo.com


Gamma Calibration Chart
Courtesy Greg Ward
www.debevec.org/gamma.gif
2.8
2.6
2.4
2.2
pixel value=186
2.0
1.8
1.6
1.4
1.2
1.0
pixel value=128
0.8


Gamma 2.2 graph
Amount of Light
Pixel Value
255
175
128
64
Implications:
128 is less than ¼ as bright as 255
128 is more than 4 times as bright as 64
175 is twice as bright as 128
93 is half as bright as 128
128 + 128 = 175
128 / 2 = 93
See also: http://www.inforamp.net/~poynton/GammaFAQ.html

Methods for taking omnidirectional HDR images

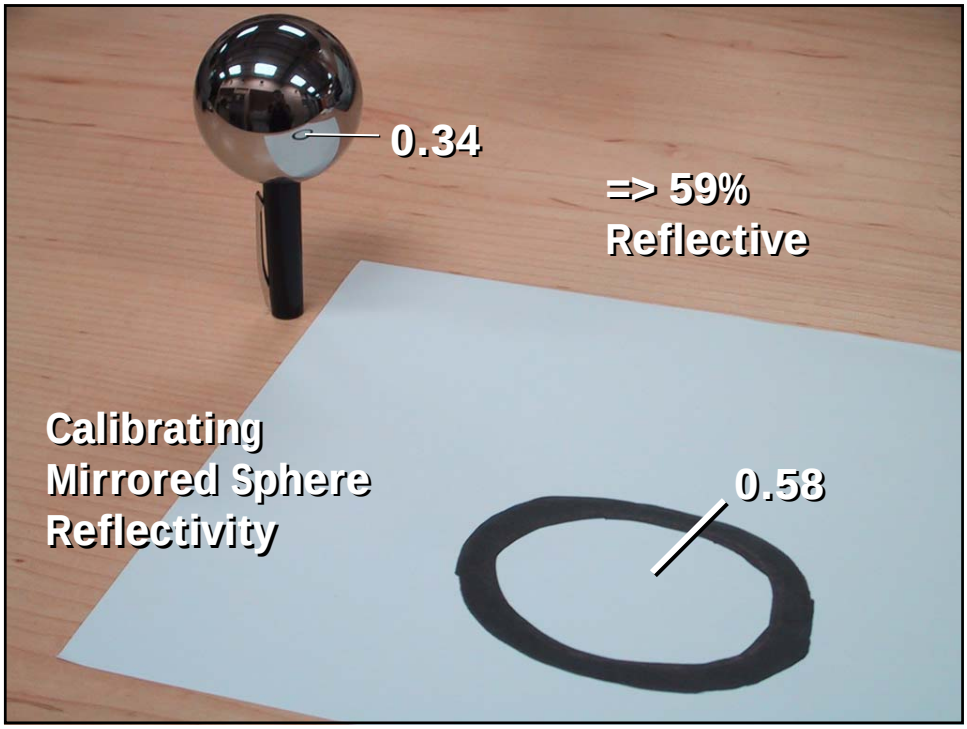
- Mirrored ball + camera
- Fisheye lens images
- Panoramic camera
- Stitching images together



Sources of Mirrored Balls

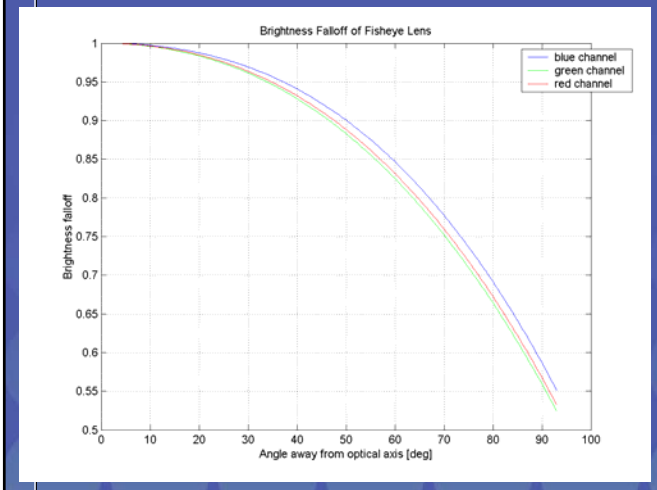
SIGGRAPH2004

- 2-inch chrome balls ~ $20 ea.
  - McMaster-Carr Supply Company
    www.mcmaster.com
- 6-12 inch large gazing balls
  - Baker's Lawn Ornaments
    www.bakerslawnorn.com
- Hollow Spheres, 2in – 4in
  - Dube Juggling Equipment
    www.dube.com
- FAQ on www.debevec.org/HDRShop


0.34
=> 59%
Reflective
Calibrating
Mirrored Sphere
Reflectivity
0.58


Fisheye Images

Fisheye Lens Radial Falloff



Scanning Panoramic Cameras

Pros:

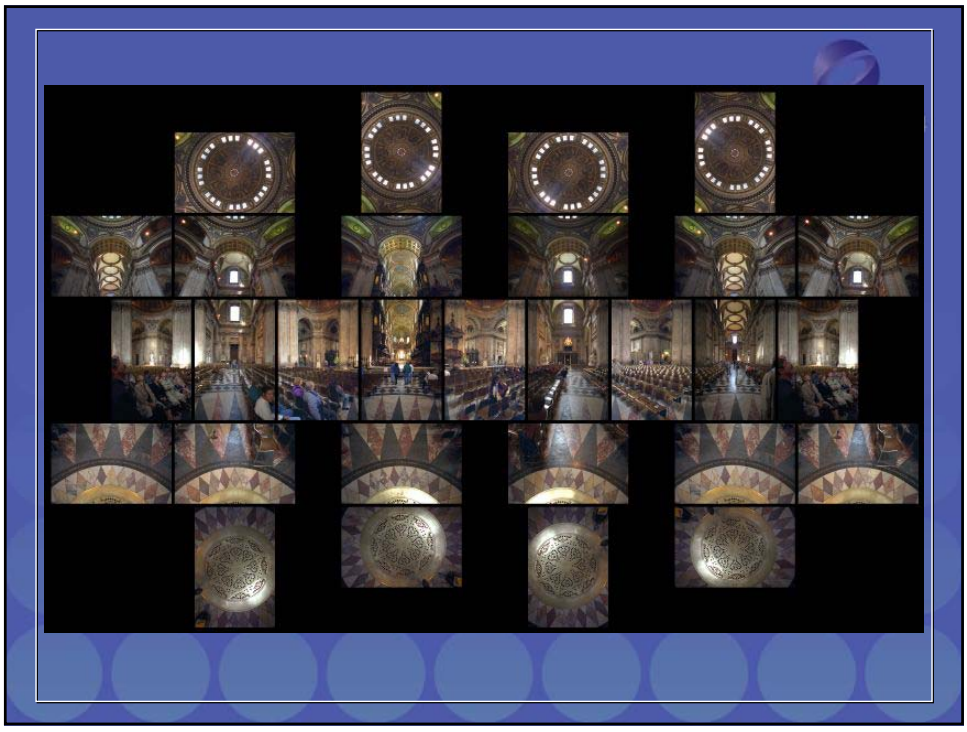
- *very* high res (10K x 7K+)
- Full sphere in one scan – no stitching
- Good dynamic range, some are HDR

Issues:

- More expensive
- Scans take a while

Companies: Panoscan, Sphereon
(SIGGRAPH 2003 booth #3340)

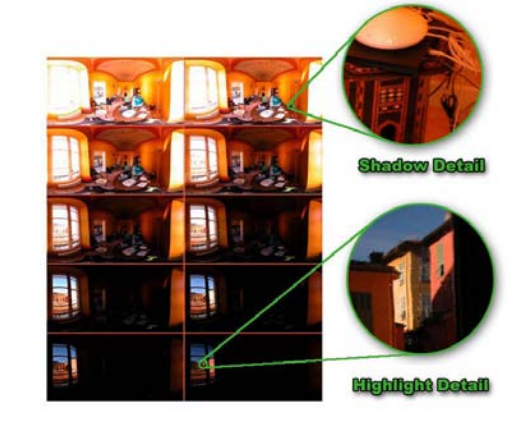

SIGGRAPH2004
See also www.kaidan.com

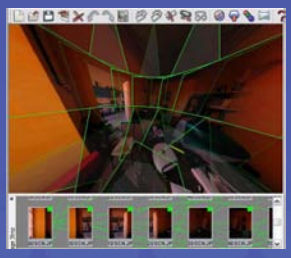


Stitching HDRI with Realviz Stitcher

SIGGRAPH2004

http://www.gregdowning.com/HDRI/stitched/





SIGGRAPH2004

HDR Image Post-Processing

Image Processing: Motion Blur

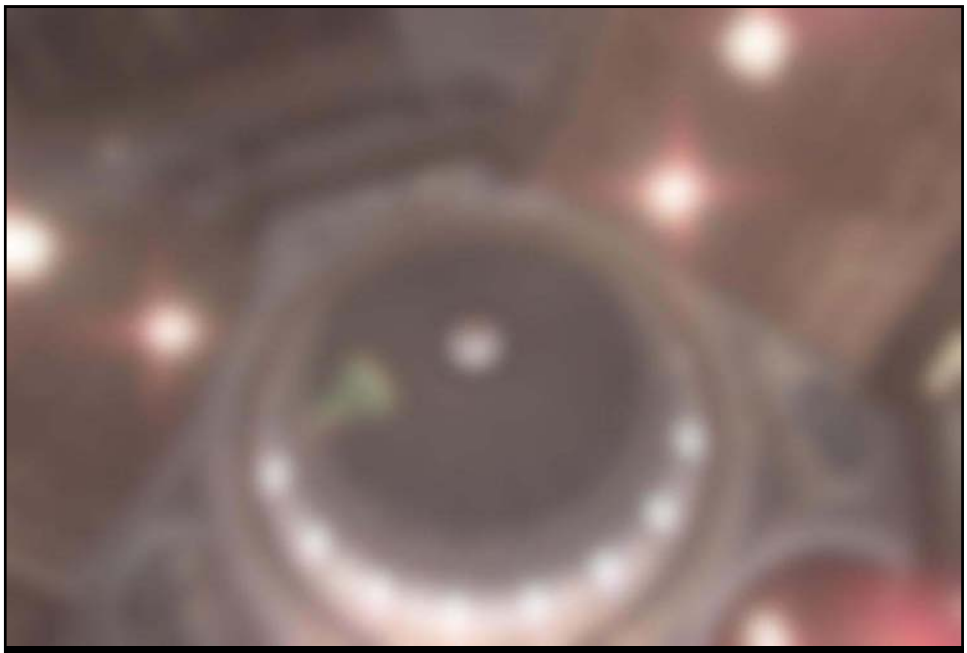
Normal digitized photo

Synthetic blur added

Blurred radiance map, virtually rephotographed

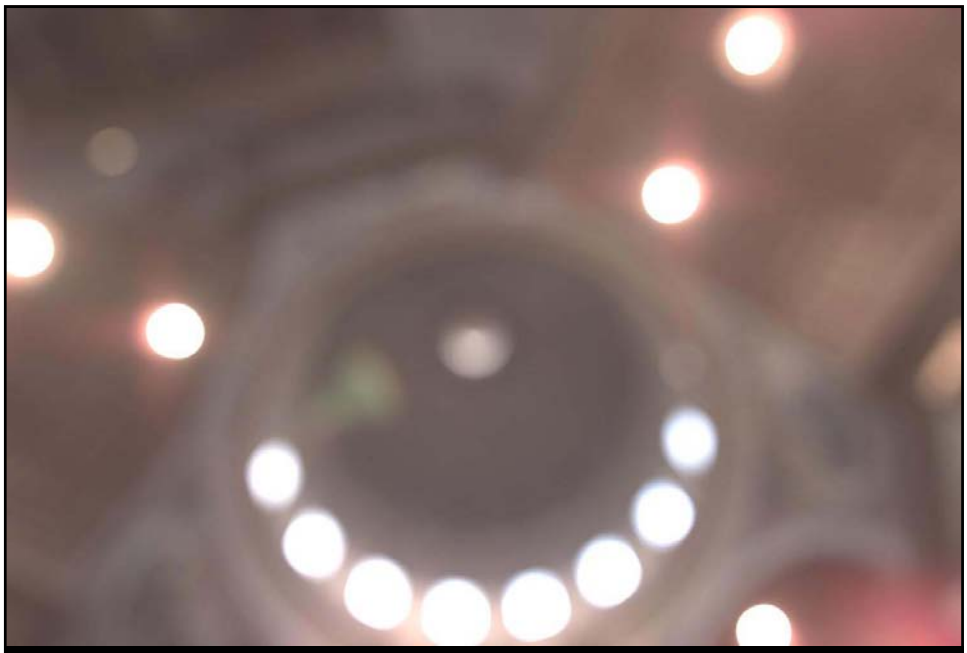
Actual blurred photograph

HDR Image

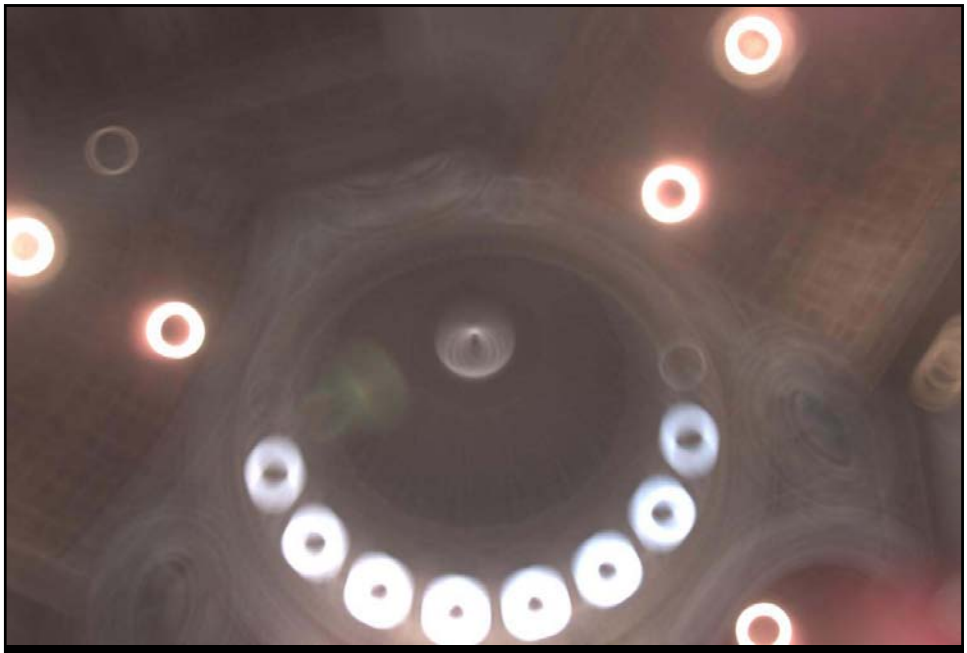


Gaussian Blur, LDR information Only

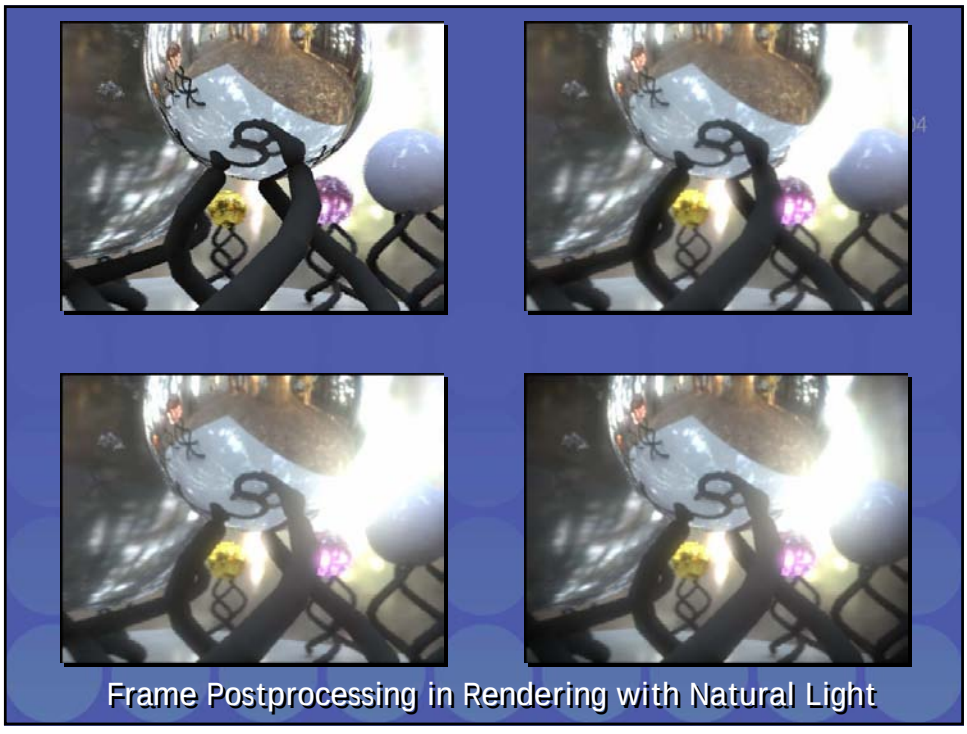
Gaussian Blur, Full HDR Information



Full HDR Disc Blur



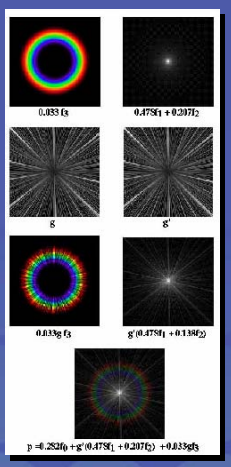
Full HDR Annular Blur



Simulating the Glare in the Human Eye

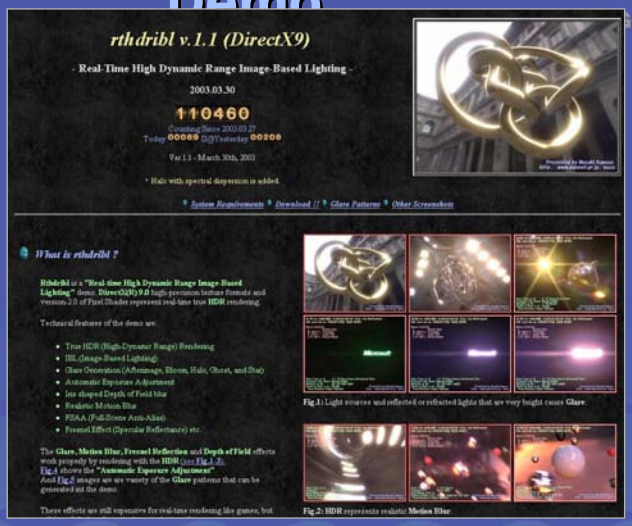
SIGGRAPH2004

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



Masaki Kawase RT HDR/IBL Demo

SIGGRAPH2004



http://www.daionet.gr.jp/~masa/

Types of Omnidirectional Images

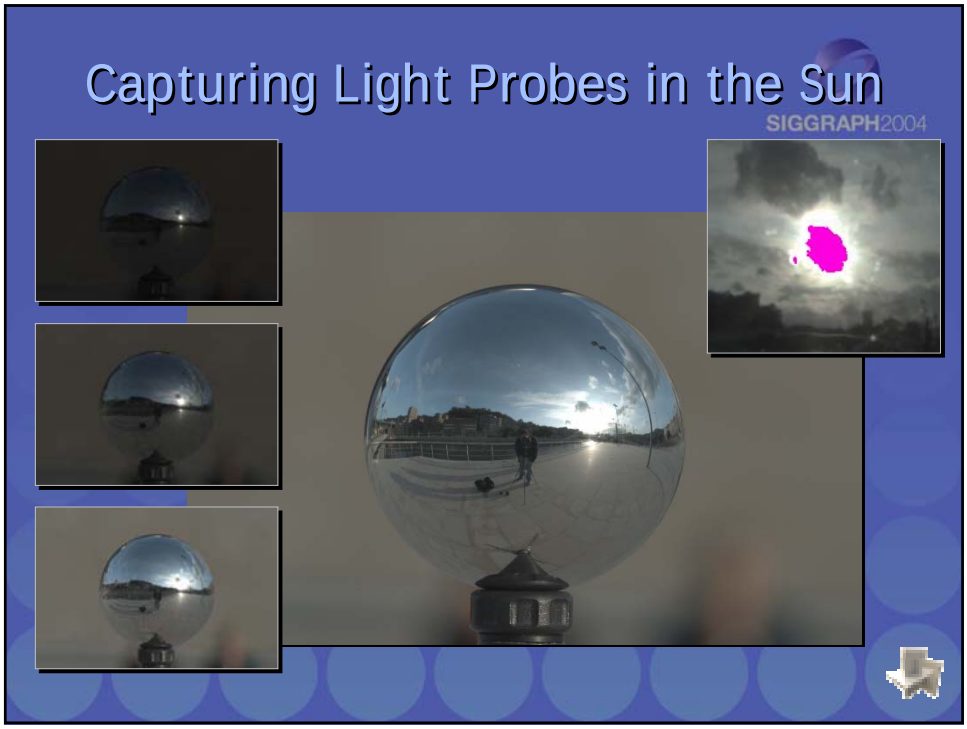
Cube Map

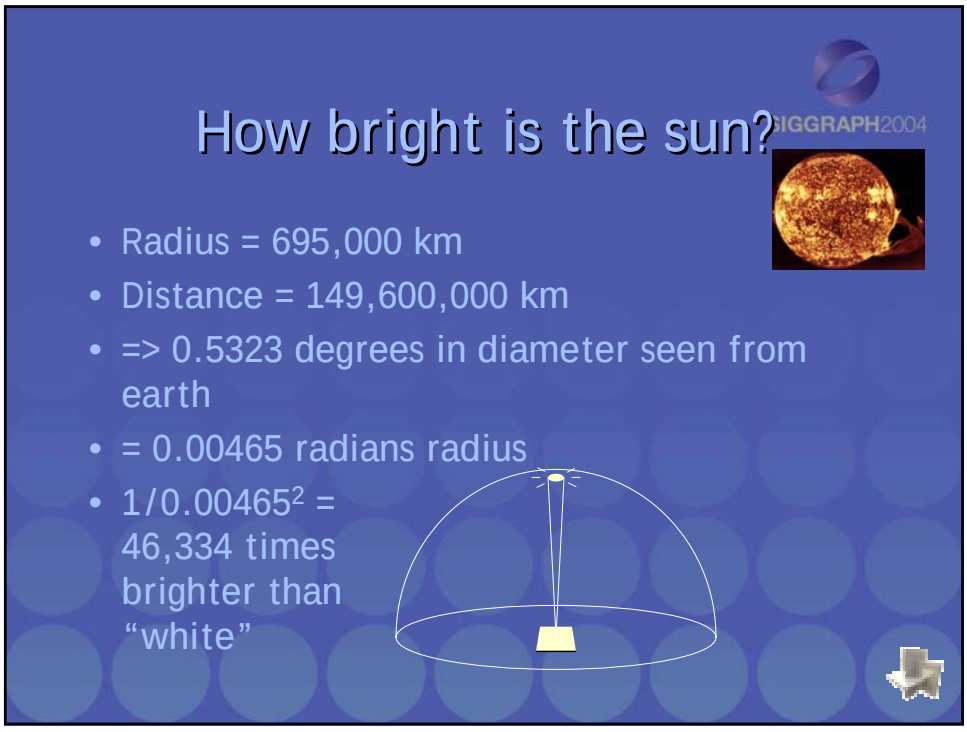
Latitude/Longitude

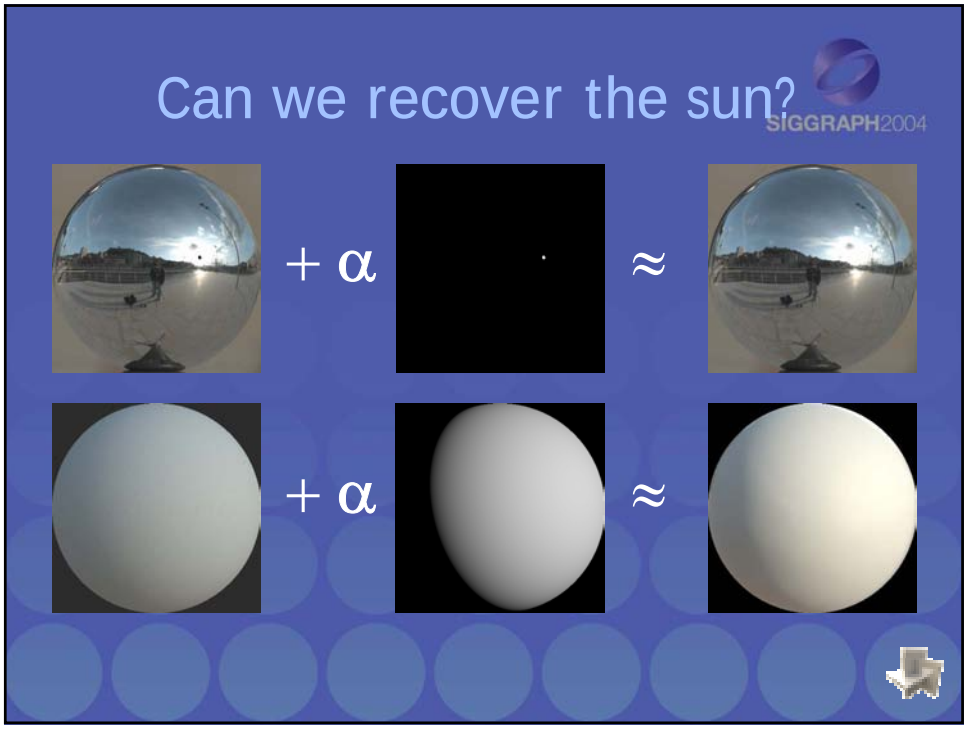
Types of Omnidirectional Images

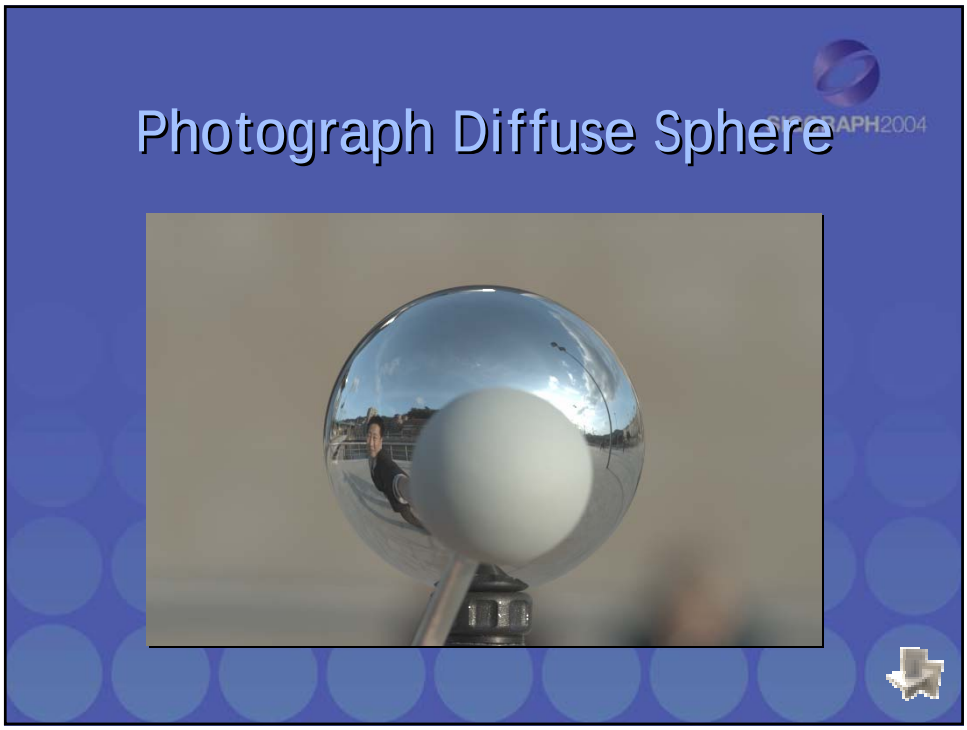
Mirrored Ball

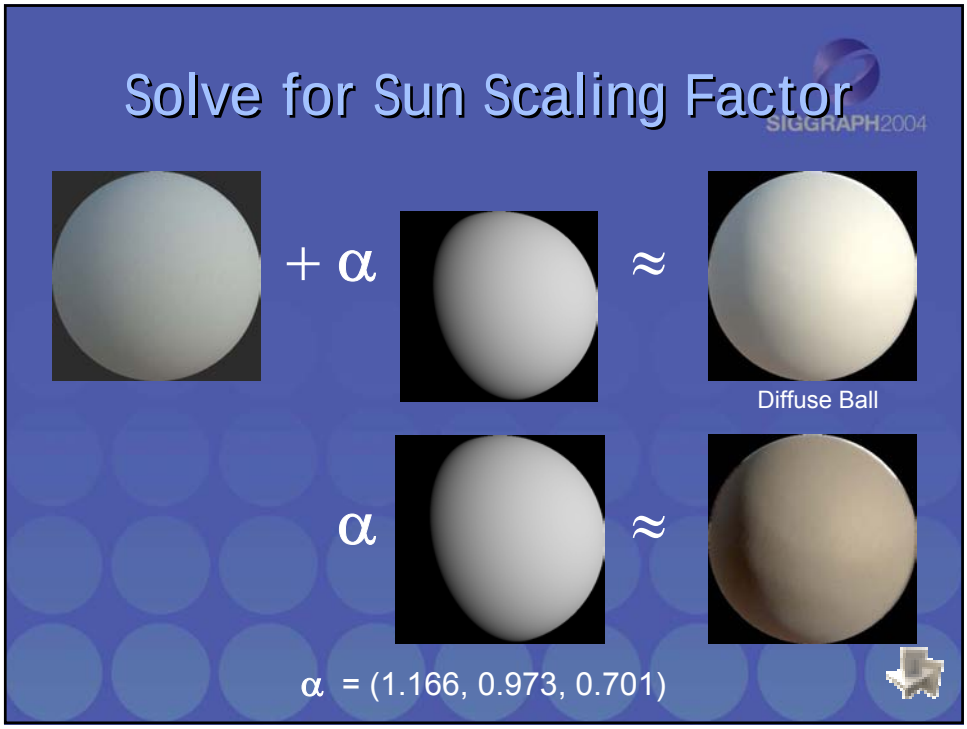
Angular Map

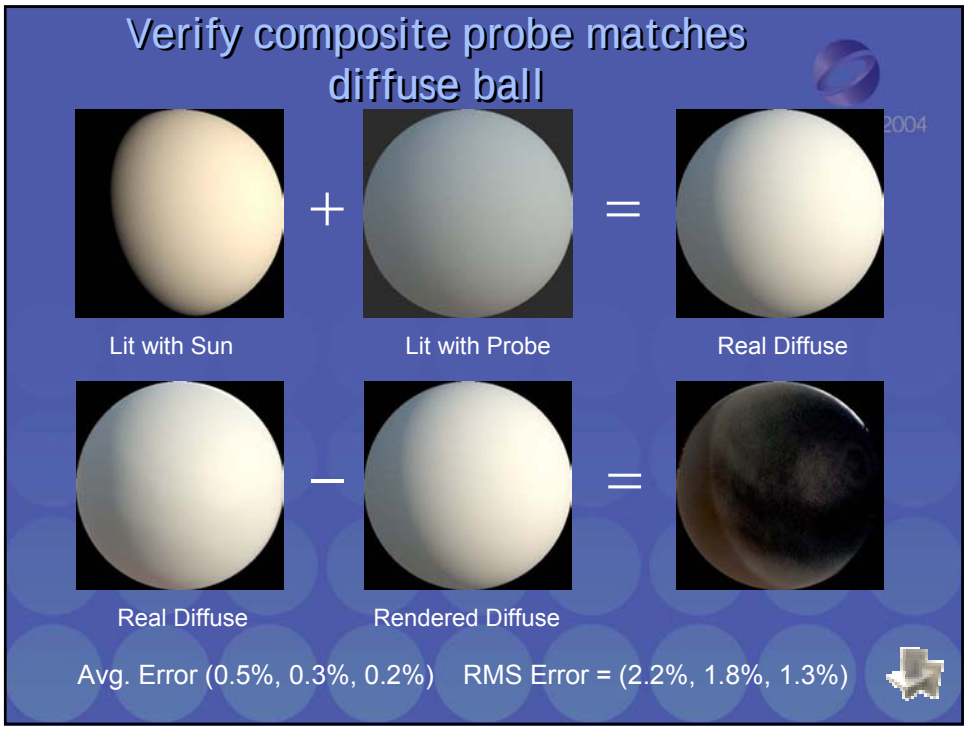

Capturing Light Probes in the Sun
SIGGRAPH2004

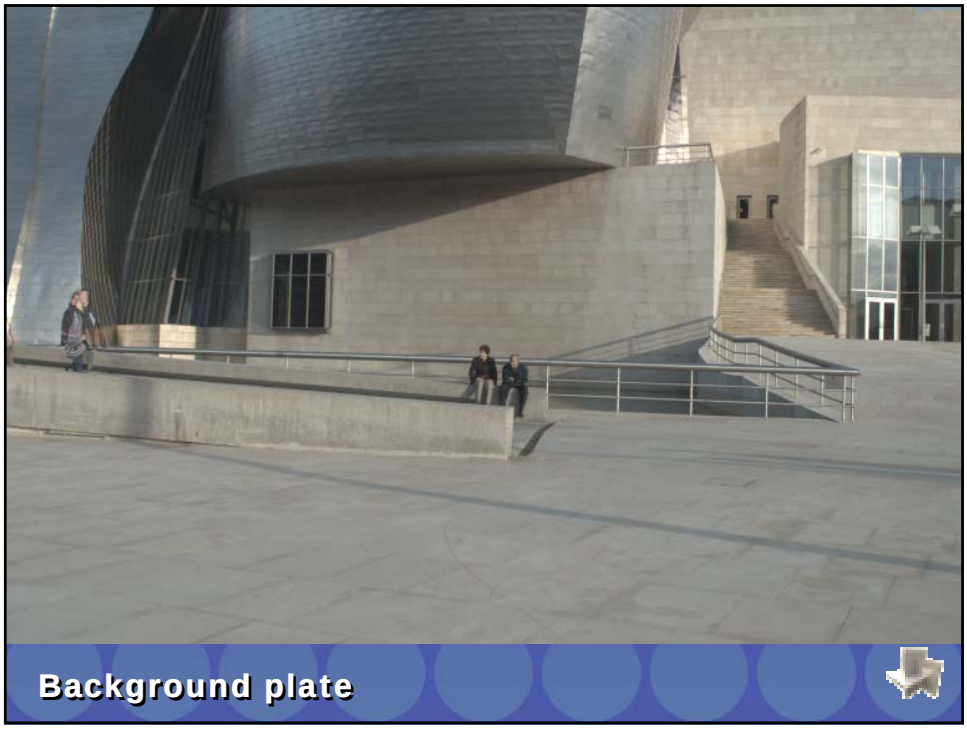

How bright is the sun?
SIGGRAPH2004
• Radius = 695,000 km
• Distance = 149,600,000 km
• => 0.5323 degrees in diameter seen from earth
• = 0.00465 radians radius
• 1/0.00465² = 46,334 times brighter than "white"

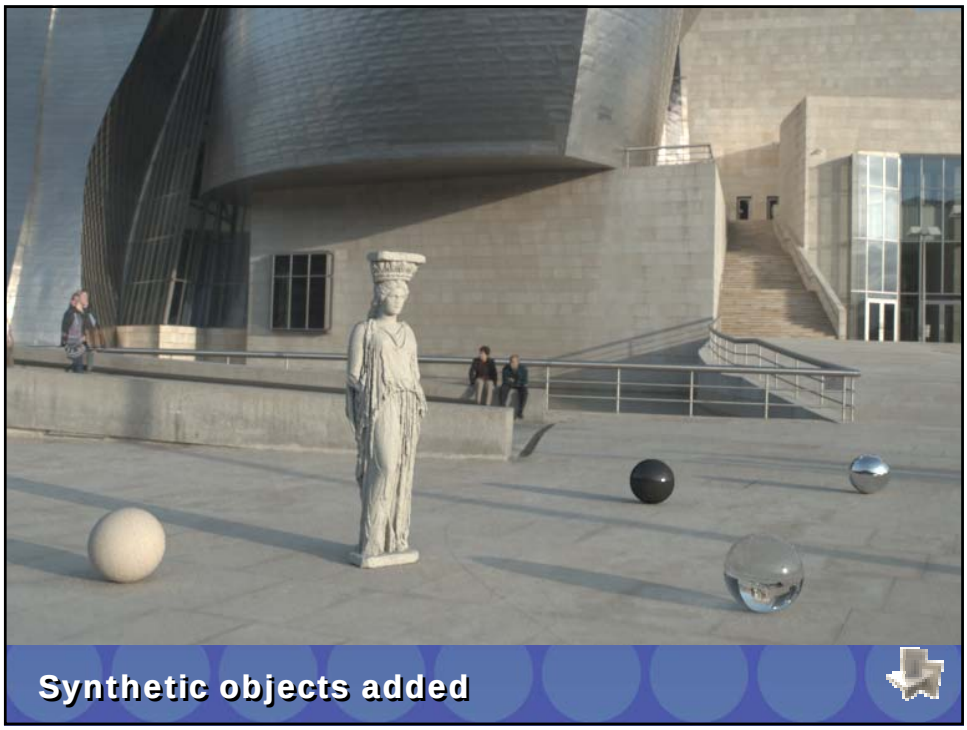

Can we recover the sun?
SIGGRAPH2004
+ α
≈
+ α
≈

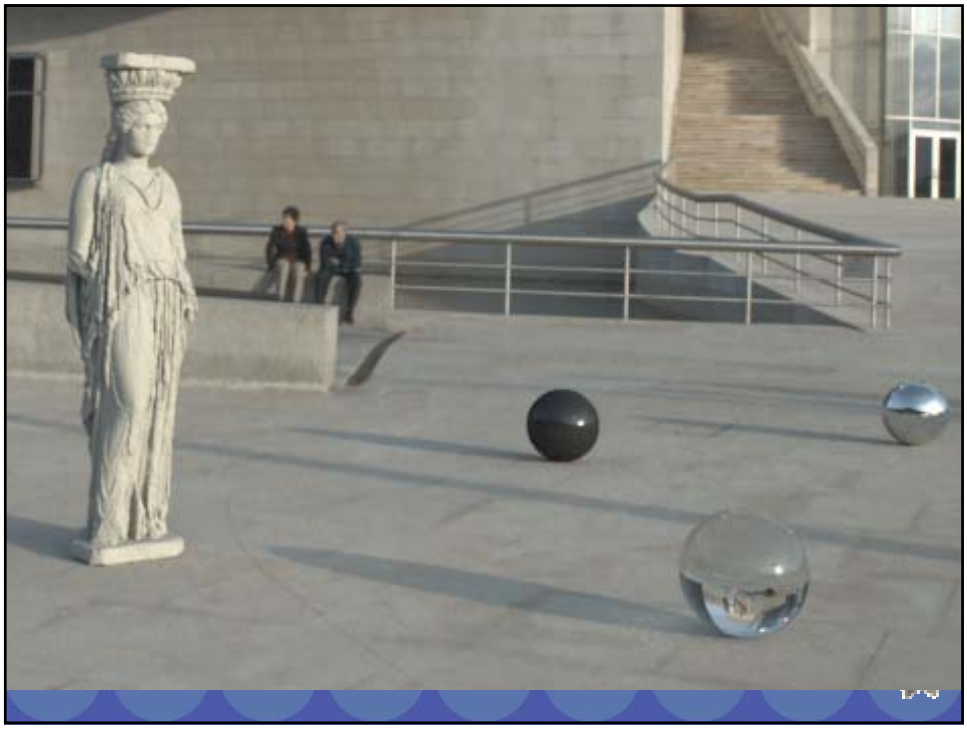

Photograph Diffuse Sphere


Solve for Sun Scaling Factor
SIGGRAPH2004
+ α
≈
Diffuse Ball
α
≈
α = (1.166, 0.973, 0.701)


Verify composite probe matches diffuse ball
+
=
Lit with Sun
Lit with Probe
Real Diffuse
−
=
Real Diffuse
Rendered Diffuse
Avg. Error (0.5%, 0.3%, 0.2%)
RMS Error = (2.2%, 1.8%, 1.3%)


Background plate


Synthetic objects added

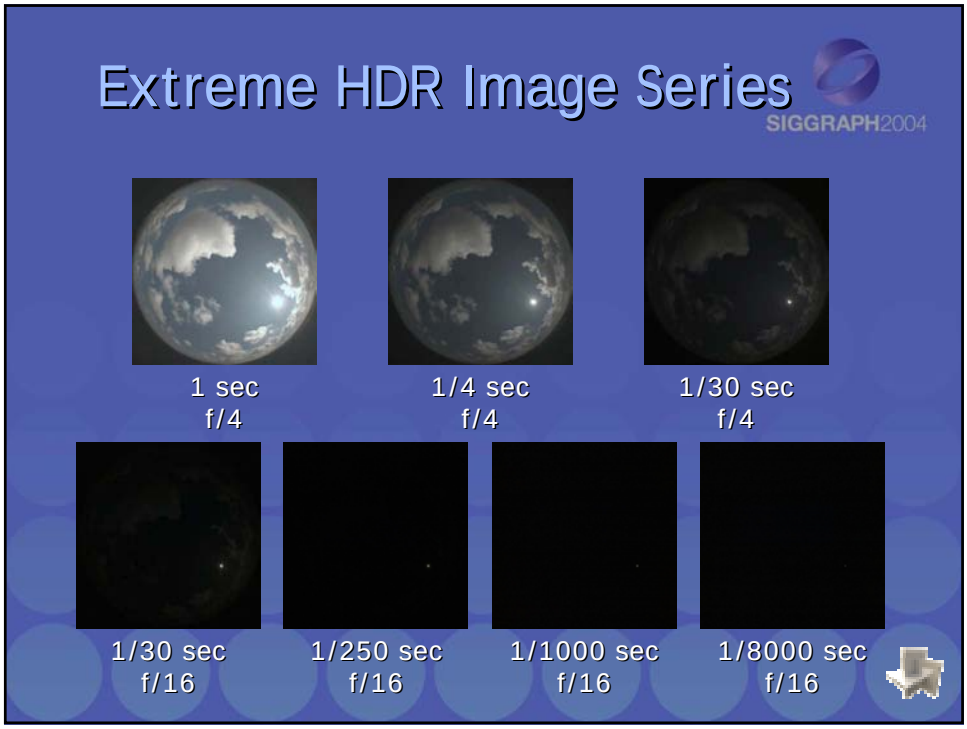


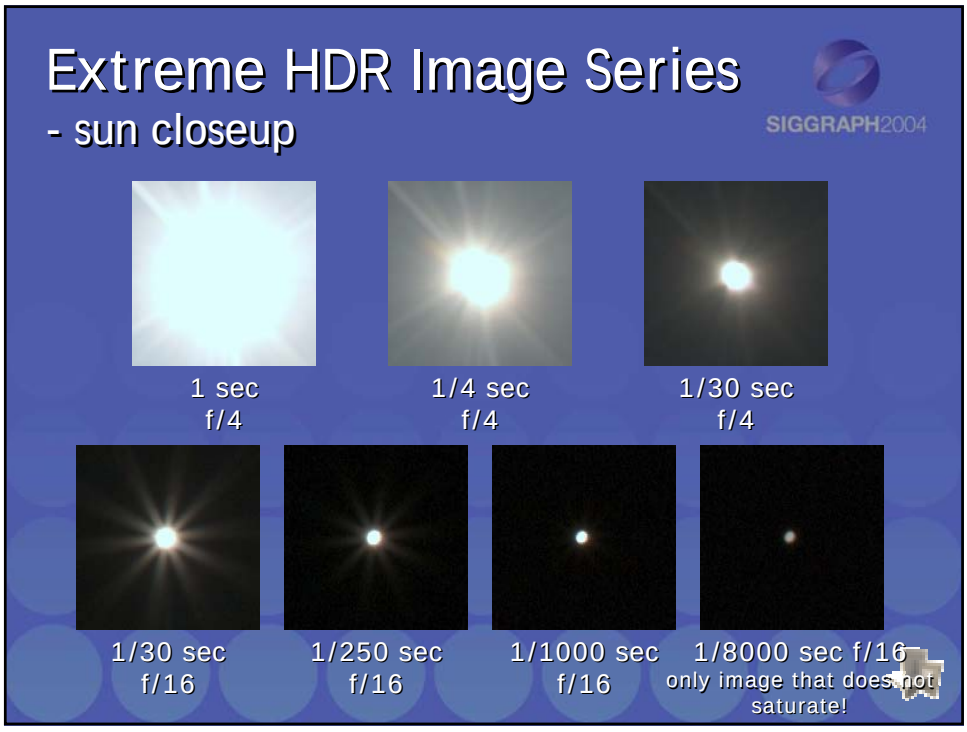
Direct HDR Capture of the Sun and Sky

SIGGRAPH2004

- Use Sigma 8mm fisheye lens and Canon EOS 1Ds to cover entire sky
- Use 3.0 ND filter on lens back to cover full range of light

Stumpfel, Jones, Wenger, Tchou, Hawkins, and Debevec. "Direct HDR Capture of the Sun and Sky". Submitted to EGSR 2004.


Extreme HDR Image Series
SIGGRAPH2004
1 sec
f/4
1/4 sec
f/4
1/30 sec
f/4
1/30 sec
f/16
1/250 sec
f/16
1/1000 sec
f/16
1/8000 sec
f/16

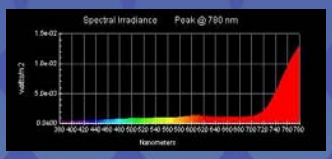

Extreme HDR Image Series
- sun closeup
SIGGRAPH2004
1 sec
f/4
1/4 sec
f/4
1/30 sec
f/4
1/30 sec
f/16
1/250 sec
f/16
1/1000 sec
f/16
1/8000 sec f/16
only image that does not saturate!

Spectral Calibration – ND filters are NOT Necessarily Neutral!

SIGGRAPH2004

Before correction

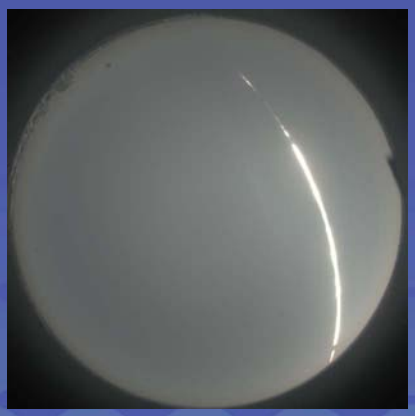
After correction based on MacBeth ColorChecker chart appearance



Must also verify that camera is not sensitive to infrared light, since this is passed by ND filter

Two Complete days of HDR Lighting

SIGGRAPH2004




IBL Results
SIGGRAPH2004
Virtual Parthenon model lit by a full day of light captured in Marina del Rey, CA

HDR Imaging Made Easy

SIGGRAPH2004

1. Capture in High Dynamic Range
2. Store in High Dynamic Range
3. Display in High Dynamic Range

Now, was that so difficult?

What Is Holding HDR Back in the Marketplace?

- Inertia
  - Output-referred RGB is de facto standard
  - Scene-referred color is new & unfamiliar
- Perceived Cost/Benefit
  - Demands on image capture
  - Demands on image manipulation
  - Demands on image display
  - Potential benefits are poorly understood

No Reason Not to Use HDR

- HDRI Formats & Libraries Freely Available
- Graphics Hardware Now in Floating Point
- HDR Displays Are Around the Corner
- Commercial Software Is Already Here
  - *E.g., Photogenics* by idruna
- HDRI Capture Possible within Digital Cameras by Reprogramming Firmware

1. HDR Capture Made Easy

SIGGRAPH2004

- Automatic Exposure Alignment
- Lens Flare Removal
- "Ghost" Removal
- Reprogramming Digital Cameras for HDR

LDR Exposure Registration

[Ward 2003, *Journal of Graphics Tools*, 8(2)]

SIGGRAPH2004

The *median threshold bitmap* (MTB) allows us to quickly compare and align different images, because it is constant with respect to exposure for any camera with a monotonic response function

The same is not true for an edge map, which changes with exposure even with careful normalization and approximate response curves

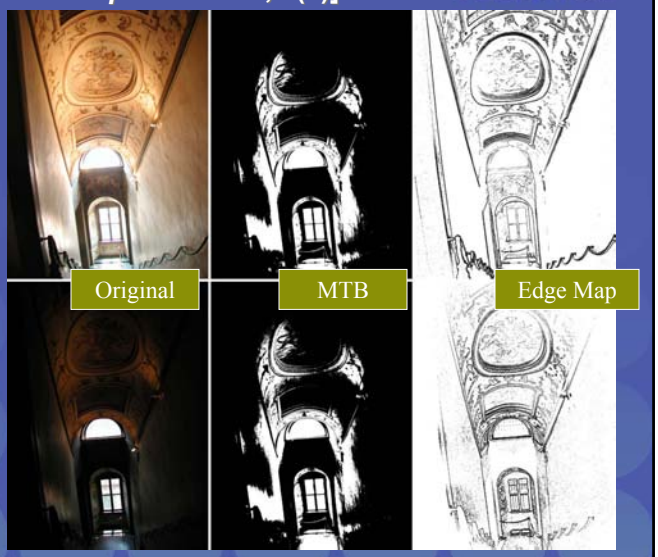


Image Pyramid Alignment

Grayscale images are scaled down repeatedly to create an image pyramid, which is then converted into MTBs for comparison

The smallest images are aligned first within a ±1 pixel distance, which corresponds to a ±32 pixel distance in the original

This becomes the MSB in the offset, which is shifted and used as the starting point for the next higher resolution alignment, and so on to the top

Alignment Results

5 unaligned exposures | Close-up detail | MTB alignment

Time: About .2 second/exposure for 3 MPixel image

Lens Flare Reduction

Before After

"Ghost" Removal

Before After

Photosphere

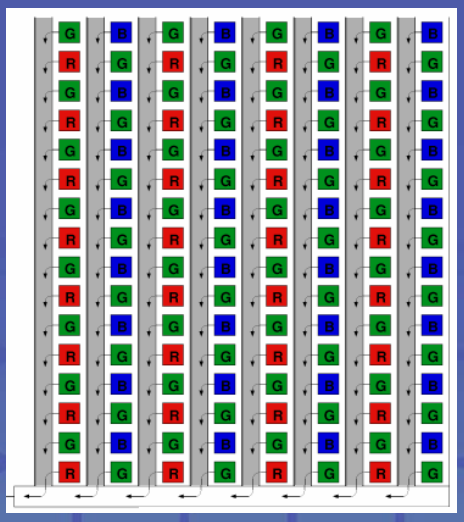
- HDRI Browsing & Cataloging Application
  - Also builds HDRI's from bracketed exposures
- Available from www.anyhere.com
  - Mac OS X app., Linux command-line tool

Photosphere Demo

Shortcuts to HDR for "Prosumer" Cameras

- Leverage CMOS Sensor Technology
  - Fuji has sensor with two-part pixel
  - SMaL Camera has log sensor
  - Other proprietary sensors in the works
- Reprogram CCD Camera Exposure
  - No hardware changes necessary

Progressive Scan CCD



Old Program:

Electronic shutter holds entire frame during scanout

Preview/movie uses electronic shutter, while still capture relies on mechanical shutter

New Program:

Instead, shift pixels under electronic shutter with 1/16th of mechanical exposure still remaining

After scanning out long exposure, shift and scan out short exposure

Result: two exposures separated by 4 f-stops

Two-Exposure HDR

- Compared Results to 5 exp. on 12 Scenes
  - Two exposures was usually sufficient
  - Less noise averaging but otherwise good
- Camera Implementation Reduces Artifacts
  - No alignment issues on short exposures
  - Longer exposure akin to “slow flash”
- Marginal Manufacturing Cost: $0.00

Sample Results

2. HDR Storage Made Easy

- Good HDRI Standards Already Exist
  - And yet, vendors create new ones
    - E.g., *scRGB*
  - Camera RAW is the *opposite* of a standard
- What about Backwards Compatibility?
  - May be necessary to bridge consumer gap
  - Naïve app's get tone-mapped *sRGB*

HDR Image Formats

- *Radiance* RGBE & XYZE
  - RGBE usually referred to as "HDR" format
- TIFF RGB IEEE Float & LogLuv Subtypes
- ILM's OpenEXR RGB (16-bit *Half* type)
- Proposed & Emerging Standards
  - Microsoft's *scRGB* encodings
  - Subband encoding within JPEG

Radiance HDR 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

TIFF 32-bit LogLuv Format

Implemented in Leffler's TIFF library, available at remotesensing.org/libtiff

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

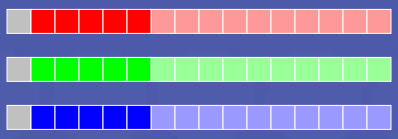
$$u_e = \lfloor 410u' \rfloor \qquad u' = \frac{4x}{-2x+12y+3}$$

$$v_e = \lfloor 410v' \rfloor \qquad v' = \frac{9y}{-2x+12y+3}$$

From Larson, JGT '98

ILM's OpenEXR Format

- 48 bits/pixel, 16 for each channel



sign exponent mantissa

- Several lossless compression options, 2:1 typical
- Compatible with the "half" datatype in NVidia's Cg
- Supported natively on GeForce FX and Quadro FX

- Available at http://www.openexr.org/

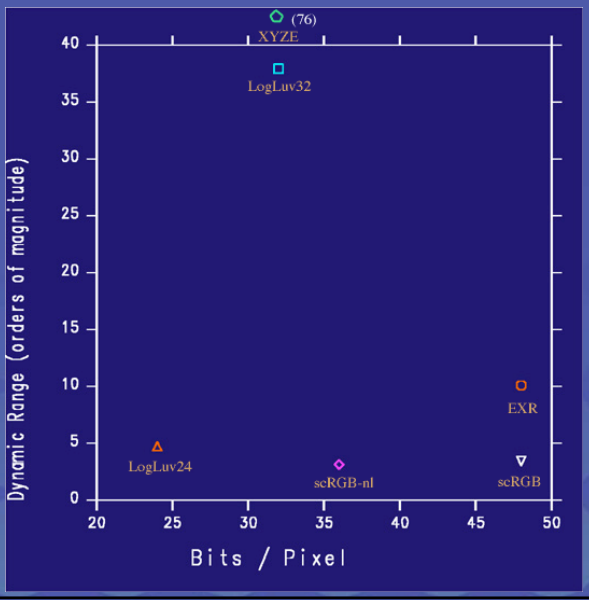
Emerging HDR Standards

- Microsoft's *scRGB* standard
  - 3 versions, *scRGB48*, *scRGB-nl*, *scYCC-nl*
  - None of these may be implemented as written
  - But something else probably will
- Subband Encoding within JPEG
  - Front image contains tone-mapped version
  - Subband contains log-encoded ratio image

Encoding Comparison Chart

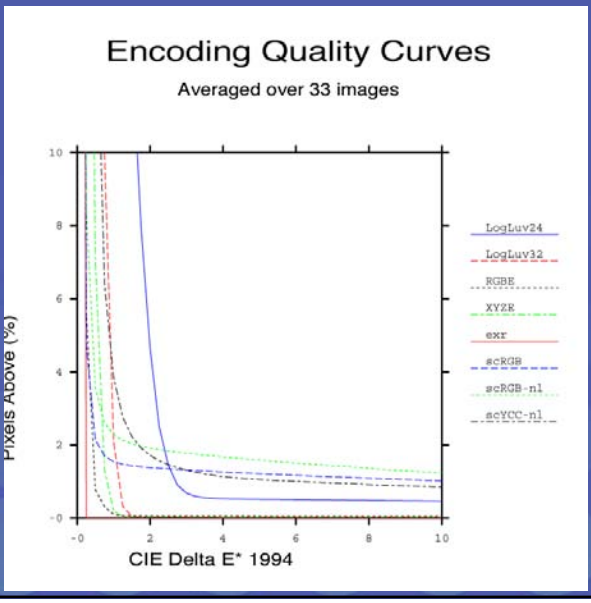
Encoding	Bits / pixel	Dynamic Range	Quant. Step	Covers Gamut
sRGB	24	1:10^1.6	Variable	No
Pixar Log	33	1:10^3.8	0.4%	No
Radiance RGBE	32	1:10^76	1%	No
Radiance XYZE	"	"	"	Yes
LogLuv 24	24	1:10^4.8	1.1%	Yes
LogLuv 32	32	1:10^38	0.3%	Yes
ILM EXR	48	1:10^10.7	0.1%	Yes
scRGB	48	1:10^3.5	Variable	Yes
scRGB-nl	36	1:10^3.2	Variable	Yes

Full Gamut Encodings



- *Radiance* XYZE has most bang for the bits
- LogLuv also fairly efficient
- OpenEXR favors accuracy over dynamic range
- LogLuv24 is marginal
- *scRGB* variants inefficient

Encoding Error Analysis



- OpenEXR sets standard
- XYZE surprisingly good
- LogLuv24 & *scRGB* feel inadequate
- LogLuv32 has higher average error than RGBE, but fewer errors over threshold ($\Delta E^* = 2$)

Backwards-Compatible HDRI Format Proposal

Debevec's Memorial Church Image

Reinhard's global photographic TMO

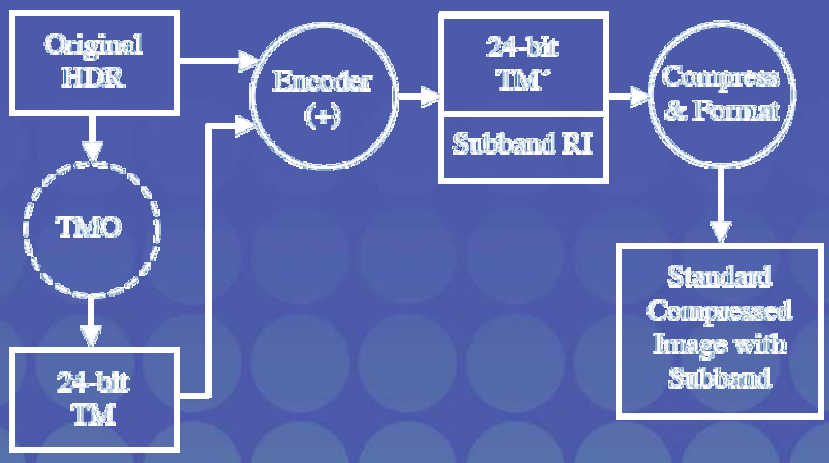
Log of original over tone-mapped

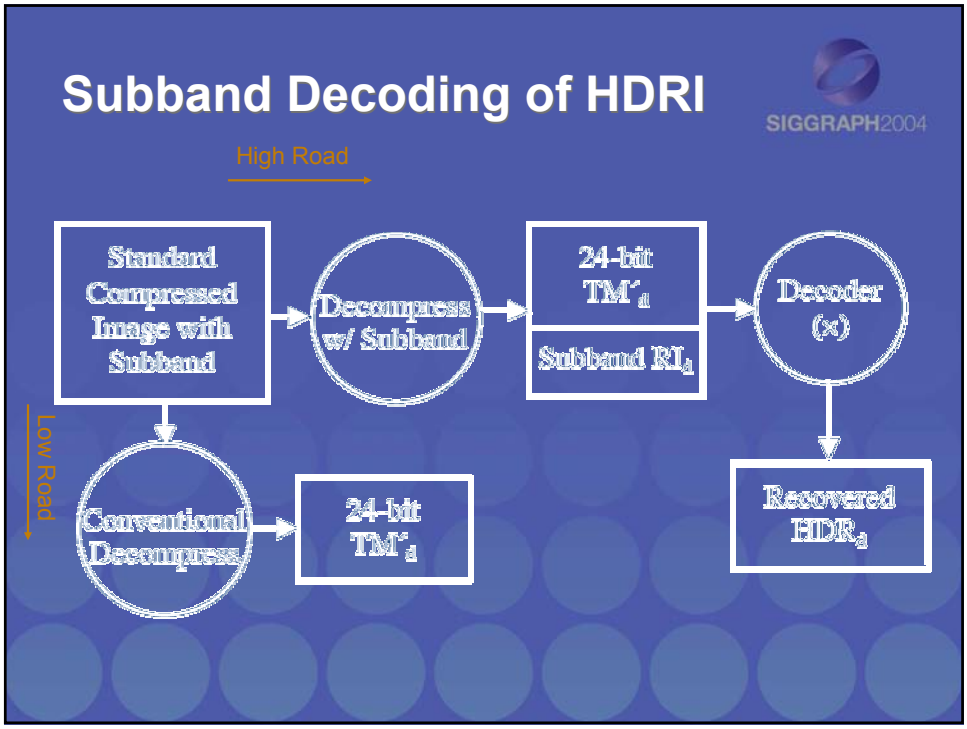
Problem: Not everyone has or wants HDR image editing software

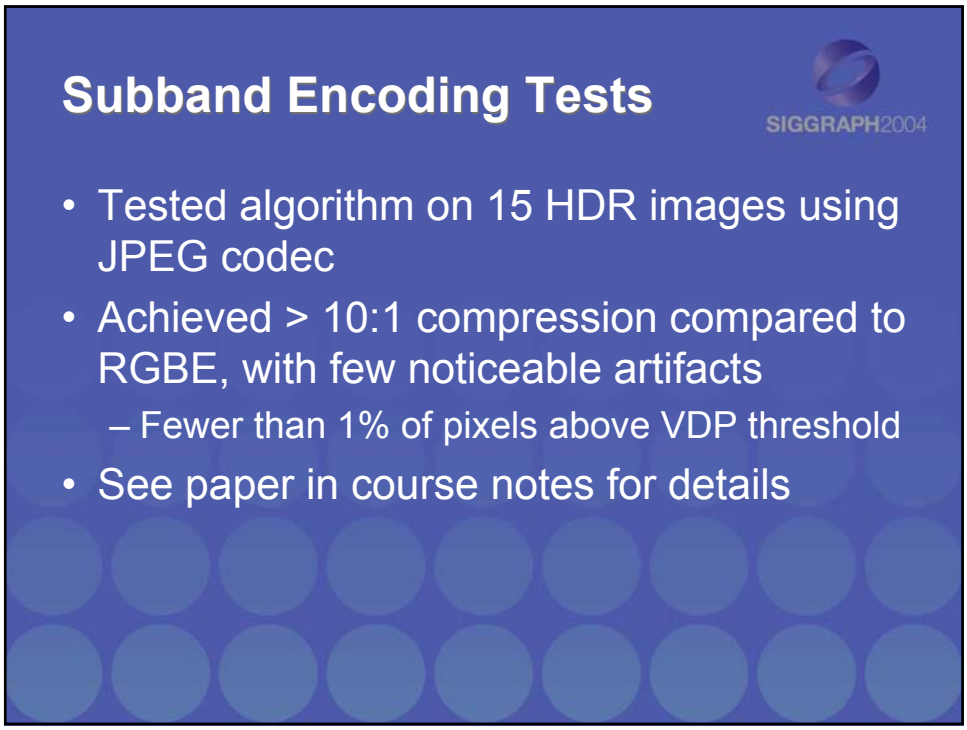
Solution: Store tone-mapped version + subband for HDR recovery

- Naïve consumers still benefit from increased dynamic range

Subband Encoding of HDRI




Subband Decoding of HDRI
SIGGRAPH2004
High Road
Standard Compressed Image with Subband
Decompress w/ Subband
24-bit TM'd
Subband RId
Decoder (×)
Low Road
Conventional Decompress
24-bit TM'd
Recovered HDRd


Subband Encoding Tests
SIGGRAPH2004
• Tested algorithm on 15 HDR images using JPEG codec
• Achieved > 10:1 compression compared to RGBE, with few noticeable artifacts
– Fewer than 1% of pixels above VDP threshold
• See paper in course notes for details

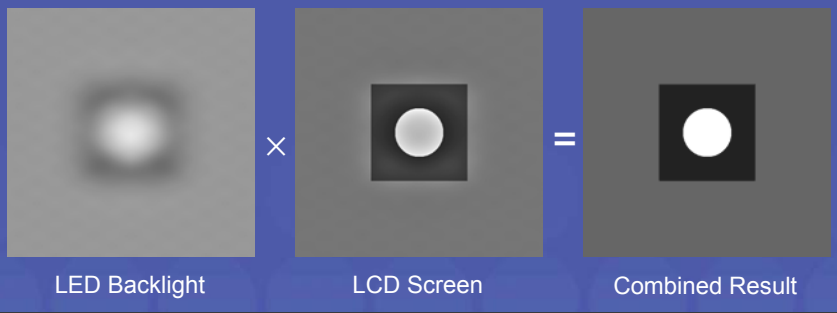
3. HDR Display Made Easy

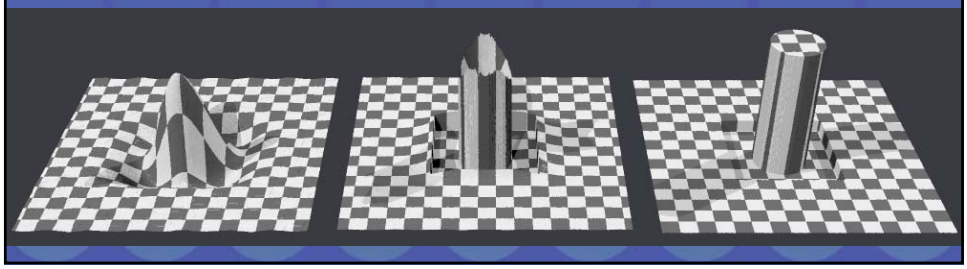
SIGGRAPH2004

- Use Bright Source + Two 8-bit Modulators
  - Transmission multiplies together
  - Over 10,000:1 dynamic range possible

How It Works

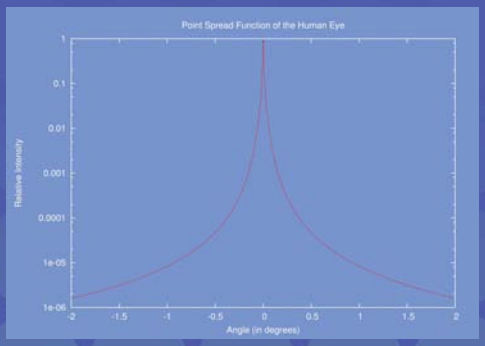
SIGGRAPH2004

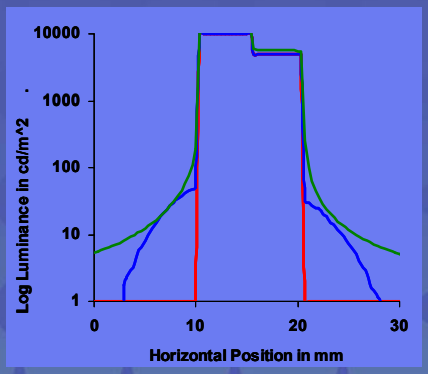




What If Edge Contrast Exceeds LCD Range?

Observers cannot tell when this happens because the eye has limited local contrast capacity due to scattering





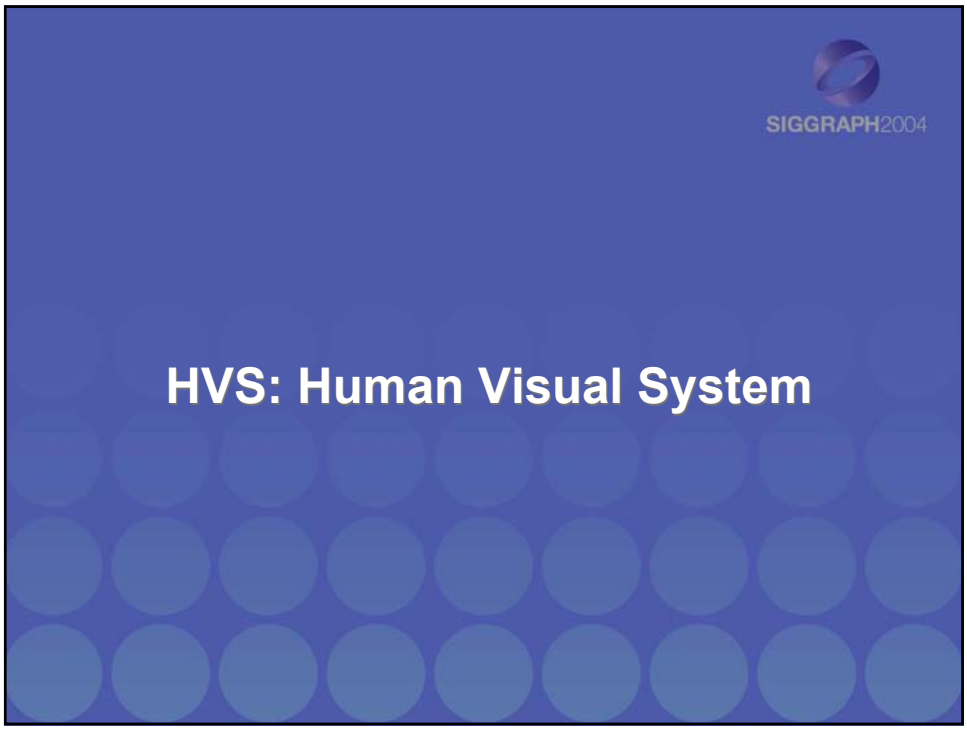
See Seetzen et al., SIGGRAPH 2004

Conclusions

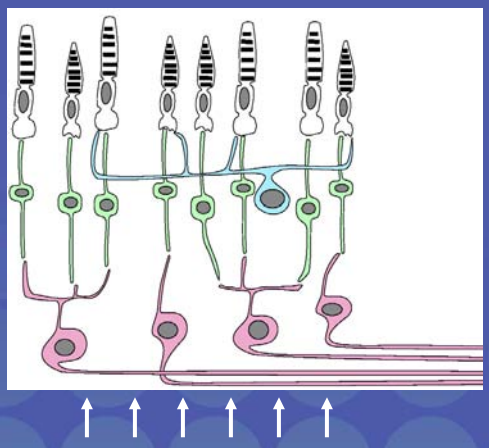
- There Is Nothing Difficult about HDR
- Benefits Beyond Special Effects
- HDR Displays Are Coming
  - Tone-mapping always needed for hard copy

It's Your Job to Demand Hi-Fi for the Eye!


SIGGRAPH2004
HVS and HDR Tone mapping
Sumanta Pattanaik


SIGGRAPH2004
HVS: Human Visual System

Retina

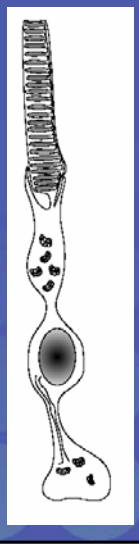


- Converts light in to signals
- Transmits signal to brain

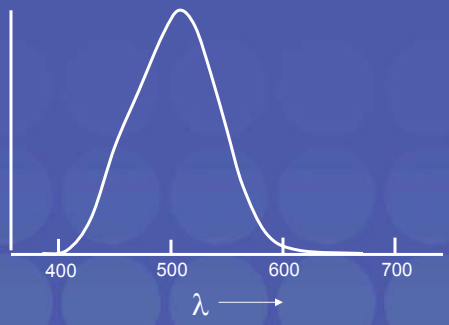
Rods

Number of rods in each retina: 10^8

Photoreceptors which
- generate achromatic response
- Sensitive only in low light levels
  (scotopic range)



Rod Sensitivity

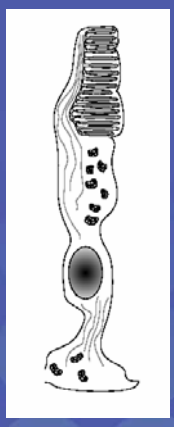


Rod Peak Sensitivity @ 507 nm

Cones

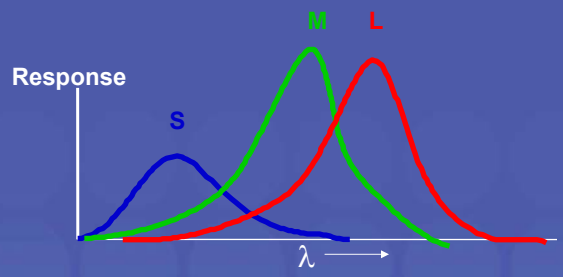
Number of cones in each retina: 5 x 10^6

Photoreceptors
- responsible for color vision
- 3 Types of cones: S, M, L
  In Fovea (64%L, 32%M, 2%S)
- Sensitive only in high light levels



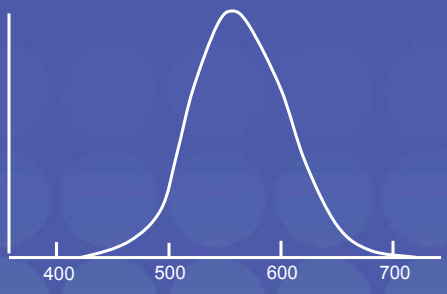
(photopic range)

Cones : S, M and L



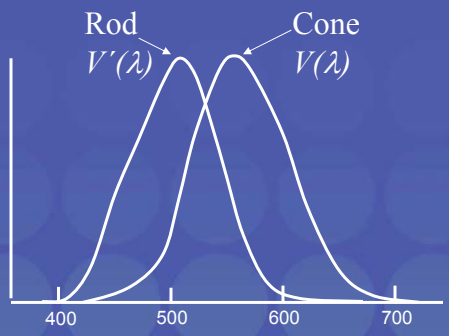
- S : Blue response (peak: 445nm)
- M : Green or yellow response (peak 535nm)
- L : Red response (peak: 575nm)
- deficiency of one or more cone type results in colorblindness
  (~ 8% caucasian males are color blind)
- Not uniformly sensitive
  -more sensitive to green than red or blue

Luminous Efficiency Function: Cone



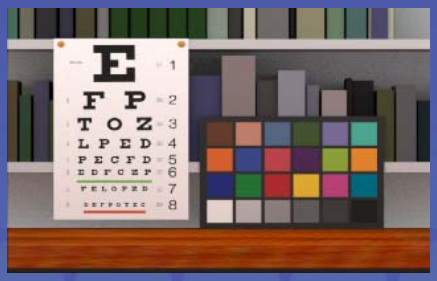
Cone peak efficiency @ 555 nm

Luminous Efficiency Function: Cone and Rod



Normalized efficiency functions.

Cone Vision vs. Rod Vision



Day Light

Twilight

Photometry

Matching of Brightness is fundamental in Photometry.

$$P_v = K \int_{\lambda} P_e(\lambda)\beta(\lambda)d\lambda$$

where

$P_e(\lambda)$: radiant quantity

$\beta(\lambda)$: Luminous efficiency function

for photopic : $V(\lambda)$ or for scotopic : $V'(\lambda)$

K: 683 for photopic

1700 for scotopic

Photometry

- Luminous Flux
  - unit: *Lumen* (*lm*)
  - 1 lm (photopic) is the luminous flux corresponding to 1/683 W of monochrmatic radiant flux of 555 nm
  - 1 lm (scotopic) is the luminous flux corresponding to 1/1700 W of monochrmatic radiant flux of 507nm
- Luminous Intensity: Luminous Flux per unit solid angle

  unit: *Candela* (*cd*) *or Lumen per steradian* (*lm.sr*$^{-1}$)

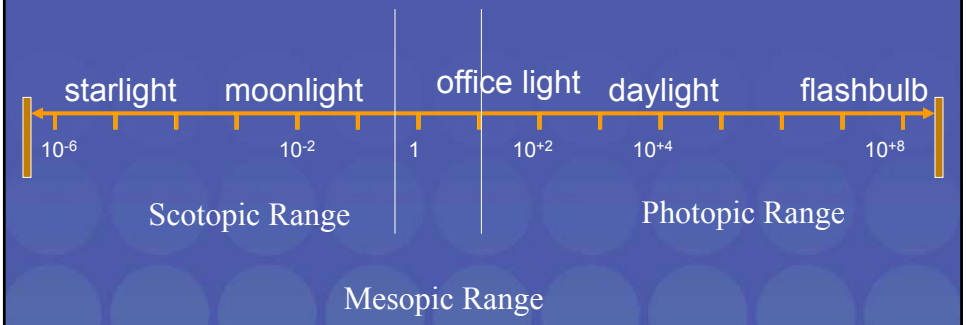
Adopted in the International Lighting Vocabulary of the Commission Internationale de l'Ecalairage (CIE)

Photometry

- Luminance : Luminous Flux per unit projected area per unit solid angle
  - unit *Candela per meter²* (*cd.m⁻²*)
    - *Lumen per steradian per meter²* (*lm. m⁻².sr⁻¹*)
  - unit *Lambert (L)*
    - *1 Lambert = (1/π) Candela per meter²*
- Retinal Illuminance:
  - unit *Troland (td)*
  - *1 troland = 1 candela per meter² x mm²*

Adopted in the International Lighting Vocabulary of the Commission Internationale de l'Ecalairage (CIE)

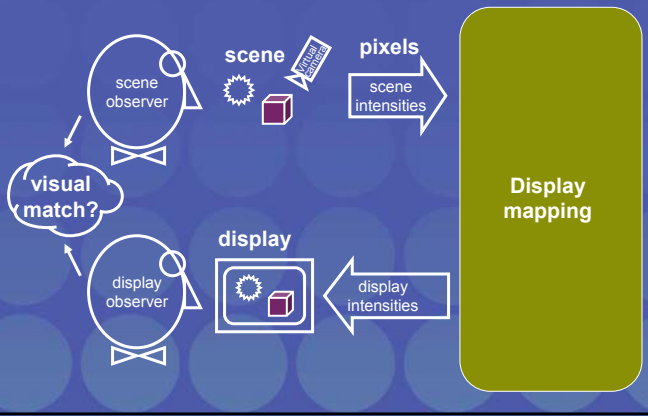
Range of Illumination



Problem Statement

SIGGRAPH2004

- How do we realistically display scene images on a display device ?



SIGGRAPH2004

Visual Adaptation and Models

Visual Adaptation

SIGGRAPH2004

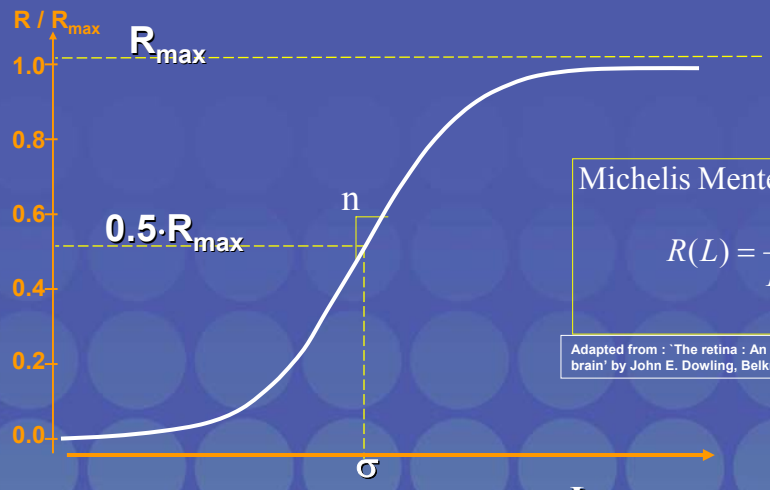
A Complex gain control mechanism.

- the pupil
- photo-pigment bleaching
- rod-cone system
- photo-receptor adaptation

Photoreceptor Response

SIGGRAPH2004

•Cells of the visual system have limited response range.

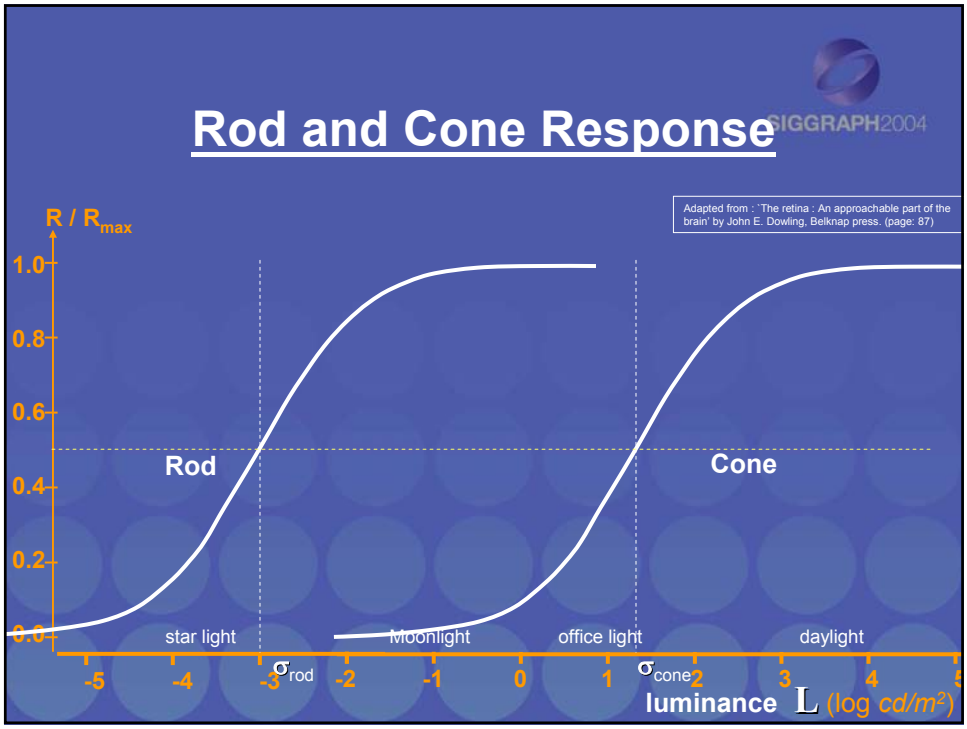


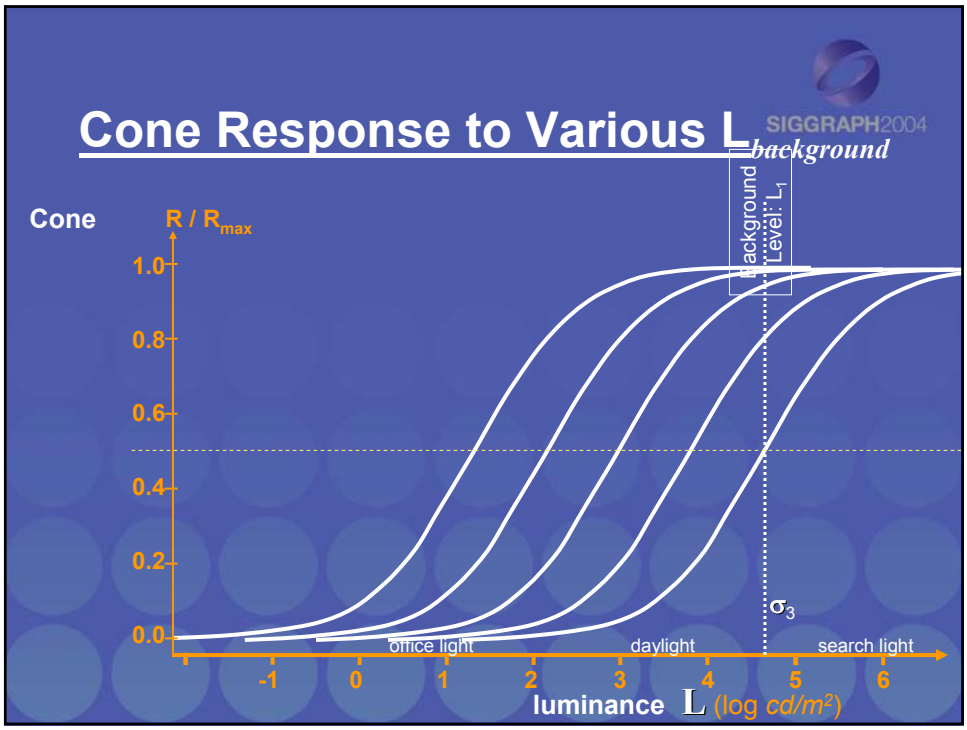
Michelis Menten Equation :

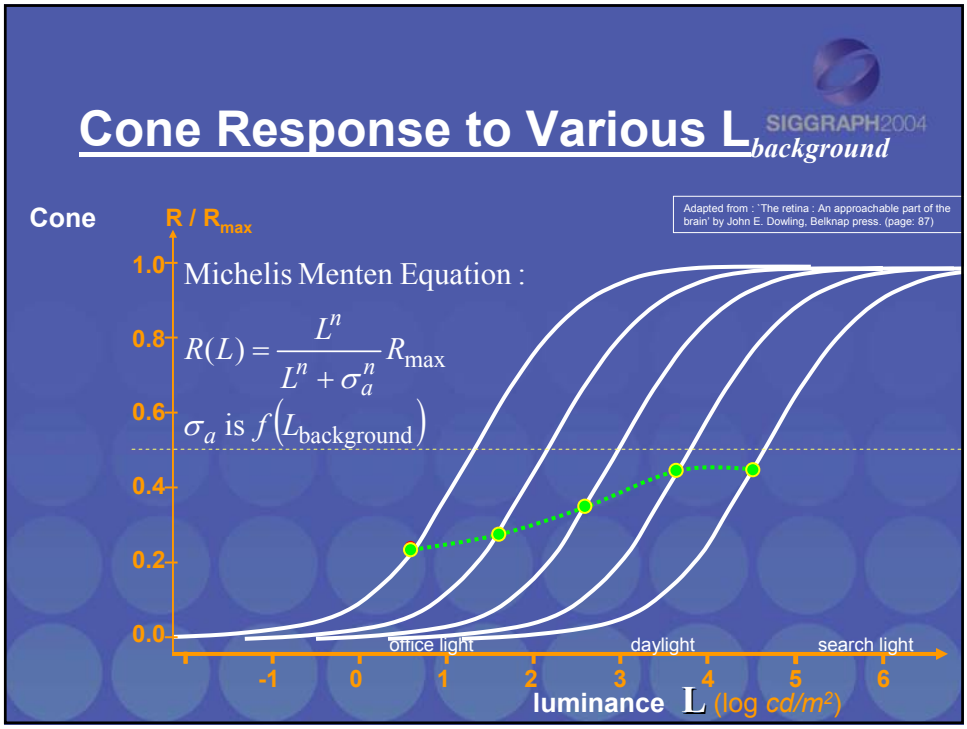
$$R(L) = \frac{L^n}{L^n + \sigma^n} \cdot R_{\max}$$

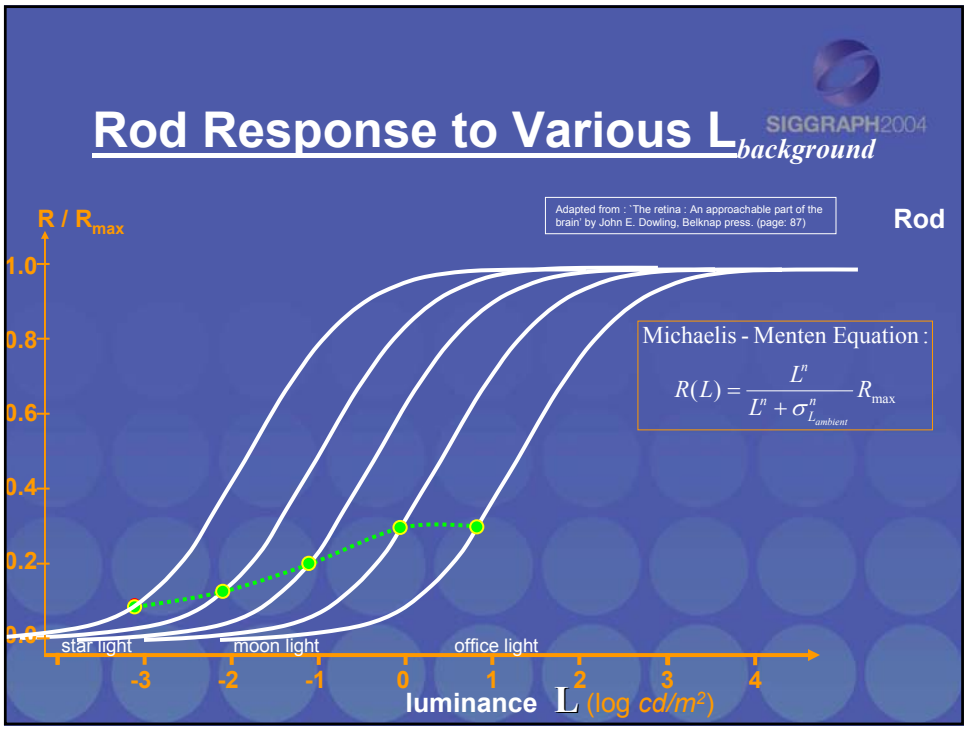
$$n \approx 0.74$$

Adapted from : `The retina : An approachable part of the brain' by John E. Dowling, Belknap press. (page: 87)


Rod and Cone Response
SIGGRAPH2004
Adapted from : 'The retina : An approachable part of the brain' by John E. Dowling, Belknap press. (page: 87)
R / Rmax
1.0
0.8
0.6
0.4
0.2
0.0
Rod
Cone
star light
Moonlight
office light
daylight
σrod
σcone
-5
-4
-3
-2
-1
0
1
2
3
4
luminance L (log cd/m²)

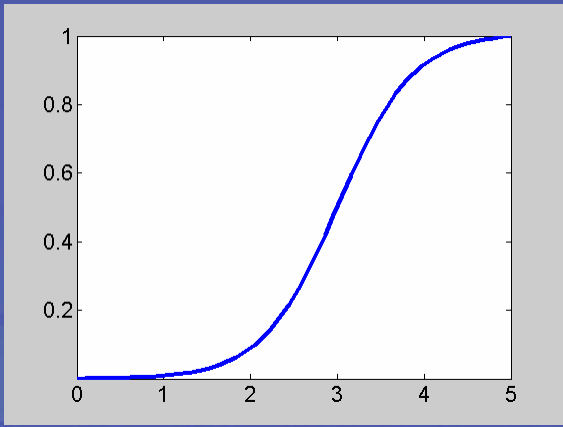

Cone Response to Various L background
SIGGRAPH2004
Cone
R / Rmax
1.0
0.8
0.6
0.4
0.2
0.0
Background Level: L1
σ3
office light
daylight
search light
-1
0
1
2
3
4
5
6
luminance L (log cd/m²)


Cone Response to Various L background
SIGGRAPH2004
Cone
R / Rmax
Adapted from : 'The retina : An approachable part of the brain' by John E. Dowling, Belknap press. (page: 87)
Michelis Menten Equation :
R(L) = L^n / (L^n + σ_a^n) R_max
σ_a is f(L_background)
1.0
0.8
0.6
0.4
0.2
0.0
office light
daylight
search light
-1
0
1
2
3
4
5
6
luminance L (log cd/m²)


Rod Response to Various L background
SIGGRAPH2004
R / Rmax
Adapted from : 'The retina : An approachable part of the brain' by John E. Dowling, Belknap press. (page: 87)
Rod
Michaelis - Menten Equation :
R(L) = L^n / (L^n + σ_Lambient^n) R_max
1.0
0.8
0.6
0.4
0.2
0.0
star light
moon light
office light
-3
-2
-1
0
1
2
3
4
luminance L (log cd/m²)

Functions similar to Photoreceptor Response Functions

$$F(L) = \frac{L}{L + \frac{L_{\max} - L}{p}}$$

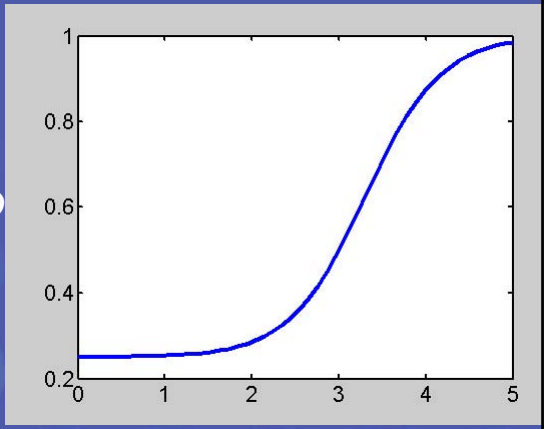


[Schlick 1994: Rational Quantization function]

p = 100

Functions similar to Photoreceptor Response Functions

$$F(L) = \left(\frac{L^n}{L^n + kL_a^n} + \frac{L_a^n}{k\left(L^n + kL_a^n\right)} \right) D$$

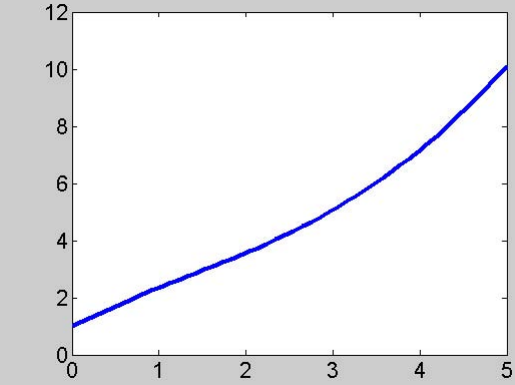


[Tumblin 98: S-shaped function]

L_a = 1000, k = 2, n = 1, D = 1

Functions similar to Photoreceptor Response Functions

$$F_{cone}(L) = \frac{L}{c_1(L_a + c_2)^n}$$

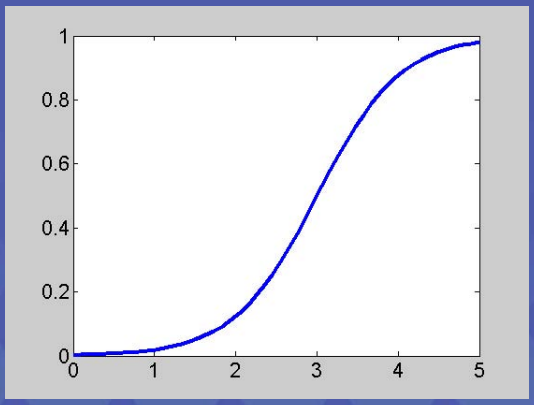


[Pattanaik 98: Gain control function]

$L_a = L$, $n = 0.85$, $c_1 = 0.555$, $c_2 = 1$

Functions similar to Photoreceptor Response Functions

$$F_{cone}(I) = B\frac{L^n}{L^n + \sigma^n}$$

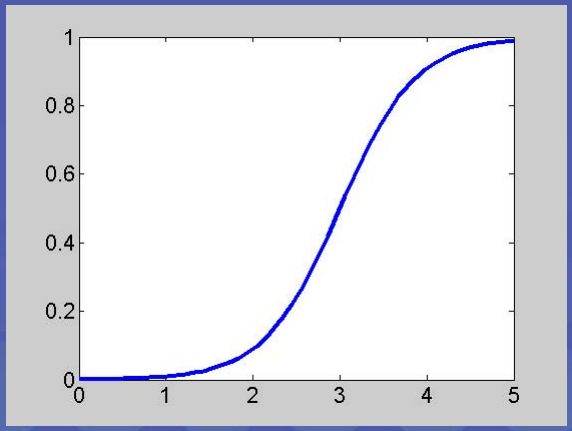


[Pattanaik 2000: Adaptation model]

$\sigma = 1000$, $n = 0.85$, $B = 1$

Functions similar to Photoreceptor Response Functions

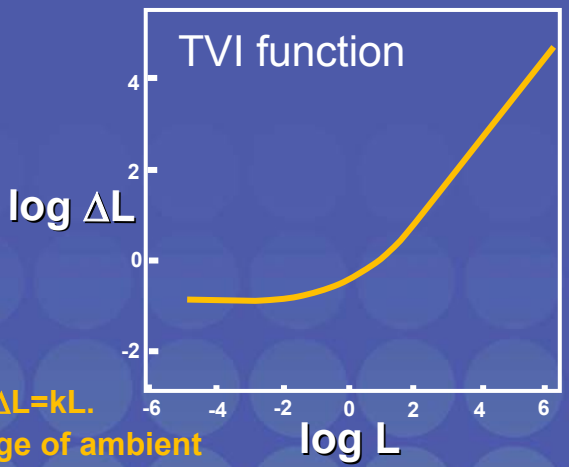
$$F(I) = \frac{L}{L + \bar{L}}$$



[Reinhard 2002: Photographic Tone repro. operator]

$\bar{L}$= 1000

Threshold Model of Adaptation



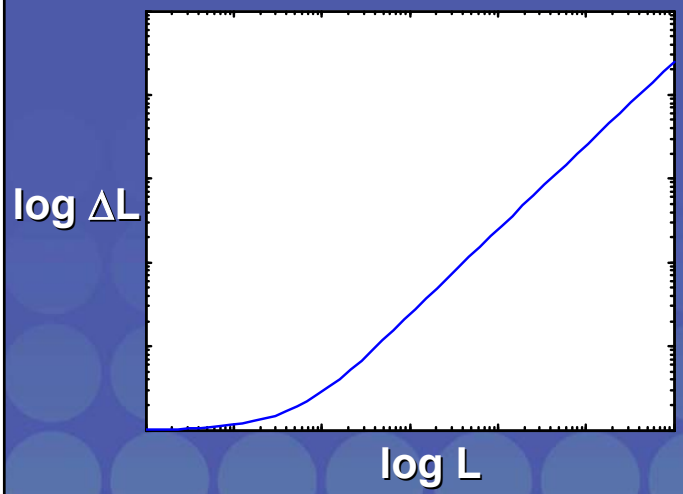
•Weber's Law : ΔL=kL. Over a wide range of ambient light, L, ΔL is directly proportional to L.

Photoreceptor Adaptation *vs* Threshold Based Adaptation

- Threshold is the incremental light (ΔL) required to create an increase in cellular response by a small criterion amount (δ).

$$\Delta L = \frac{\sigma_a}{\sigma_{\text{dark}}}\left(\frac{\delta}{R}\right)^{\frac{n-1}{n}}\left(\frac{R_{\max}}{R_{\max}-R}\right)$$

ΔL from Response Function



Example Parameters:

$n = 1$

$\sigma_{\text{dark}} = 100$

$\sigma = \frac{0.4L + \sigma_{\text{dark}}}{0.6}$

Just Noticeable Difference (JND)

SIGGRAPH2004

$\Delta L = 1$ JND

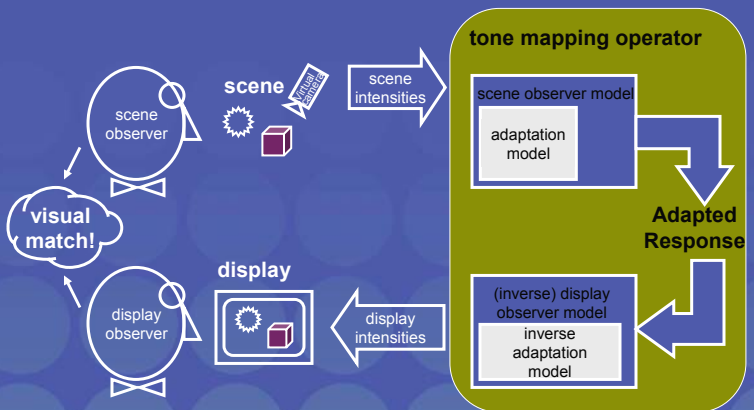
$$\#\text{JNDs for a pixel value } L = \frac{L - L_{background}}{\Delta L_{background}}$$

Display pixel value = $\#\text{JNDs} \times \Delta L_d + L_d$

where L_d is the display background Luminance.

Tone Reproduction Operator

SIGGRAPH2004




SIGGRAPH2004
Background Luminance in Images.

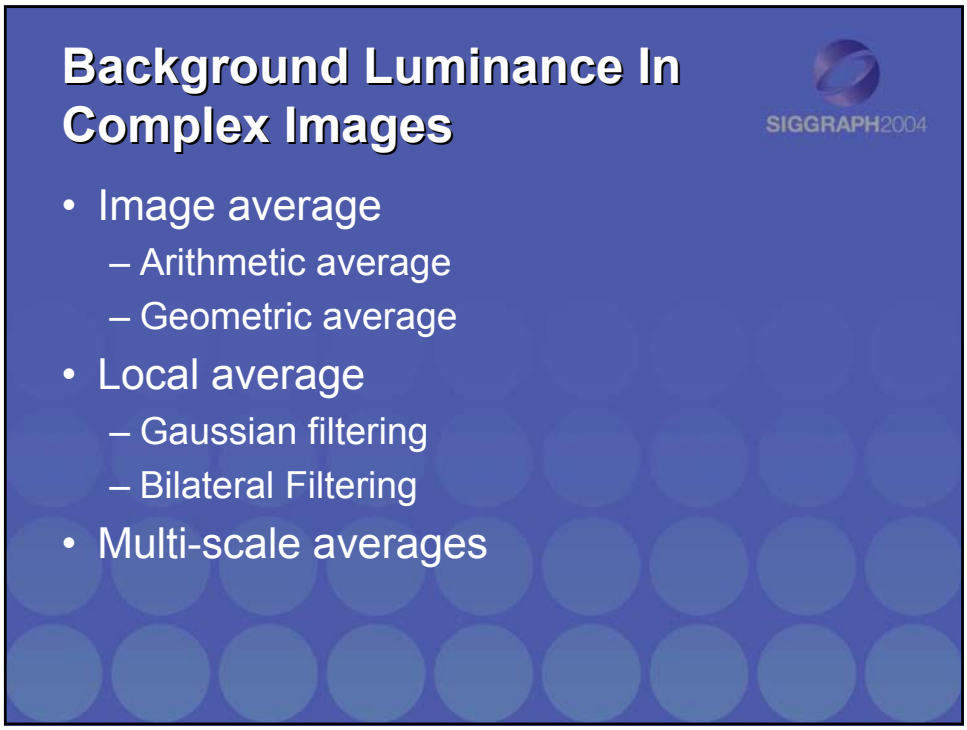

Background Luminance In Complex Images
SIGGRAPH2004
• Image average
– Arithmetic average
– Geometric average
• Local average
– Gaussian filtering
– Bilateral Filtering
• Multi-scale averages

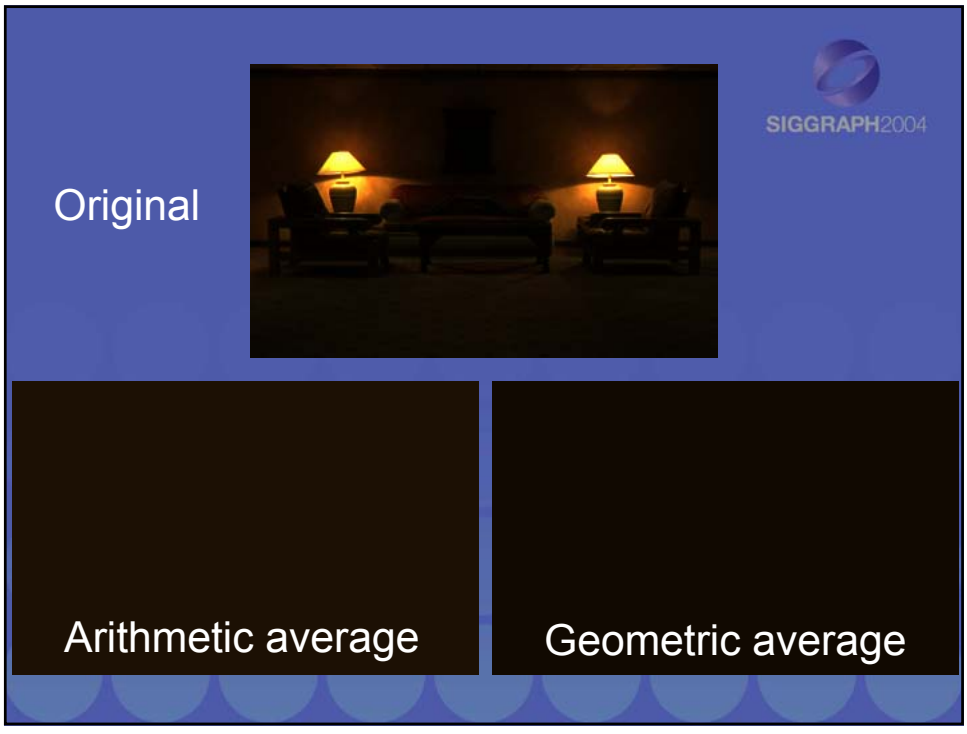
Image Average

- Arithmetic average: $L_{background} = \frac{1}{n} \sum_{pixels} L_i$
- Geometric average:

$$L_{background} = \left(\prod_{pixels} (\delta + L_i) \right)^{\frac{1}{n}}$$

or

$$L_{background} = \exp\left(\frac{1}{n} \sum_{pixels} \log(\delta + L_i) \right)$$



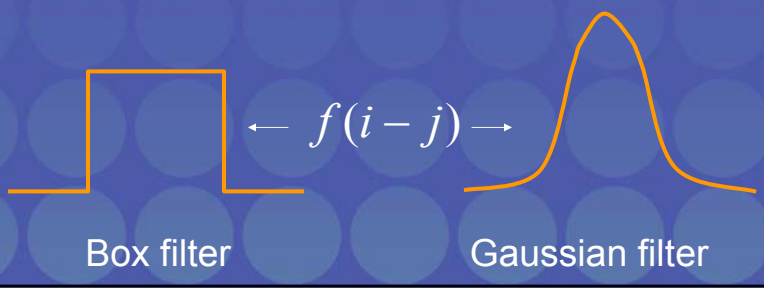
Local Average

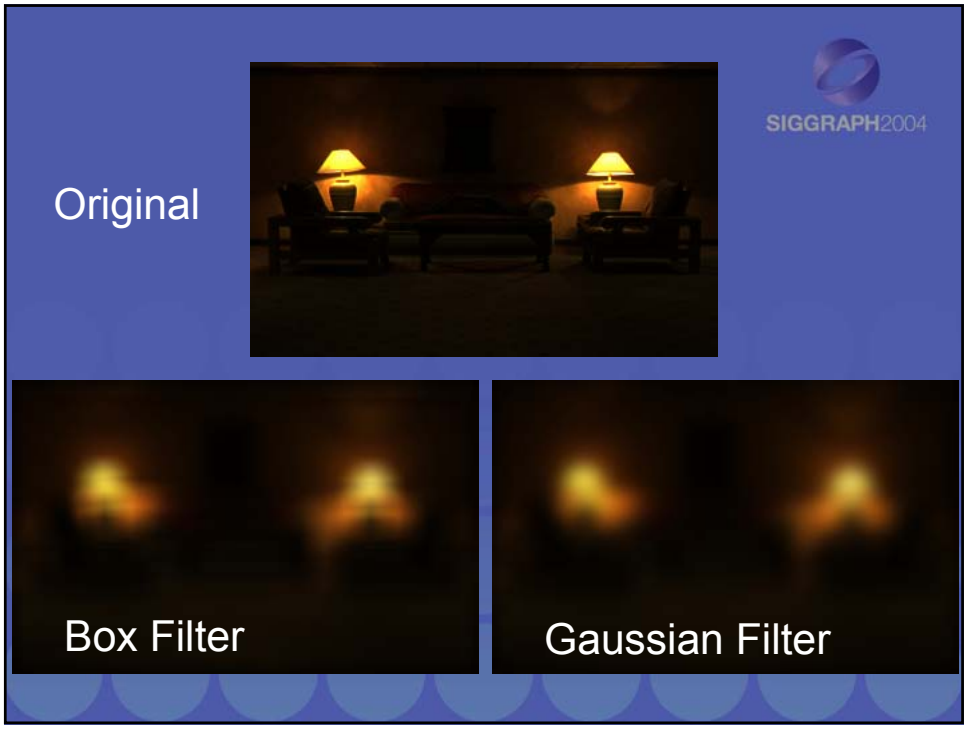
- our eyes adapt as we look from place to place in a scene
- local adaptation enhances our ability to see in high dynamic range scenes
- our visual impression of the scene is constructed from what we are able to see as we look around

Local Average

- **Filtering.**

$$L_{background}(i) = w(i)\sum_{j\in\Omega} f(i-j)L_j$$



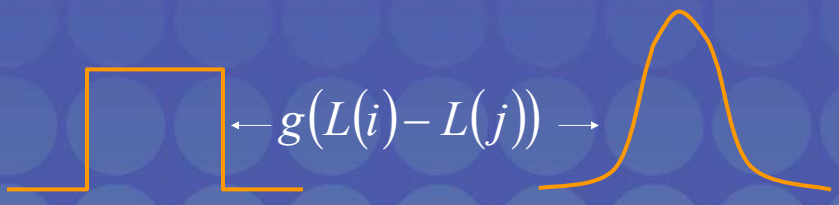


Local Average

SIGGRAPH2004

- **Bilateral Filtering.**

$$L_{background}(i) = w(i)\sum_{j\in\Omega} f(i-j)\,g\big(L(i)-L(j)\big)L_j$$

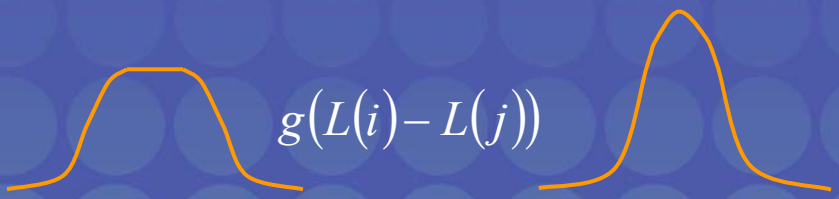


Local Average

SIGGRAPH2004

- **Bilateral Filtering.**

$$L_{background}(i) = w(i)\sum_{j\in\Omega} f(i-j)\,g\big(L(i)-L(j)\big)L_j$$

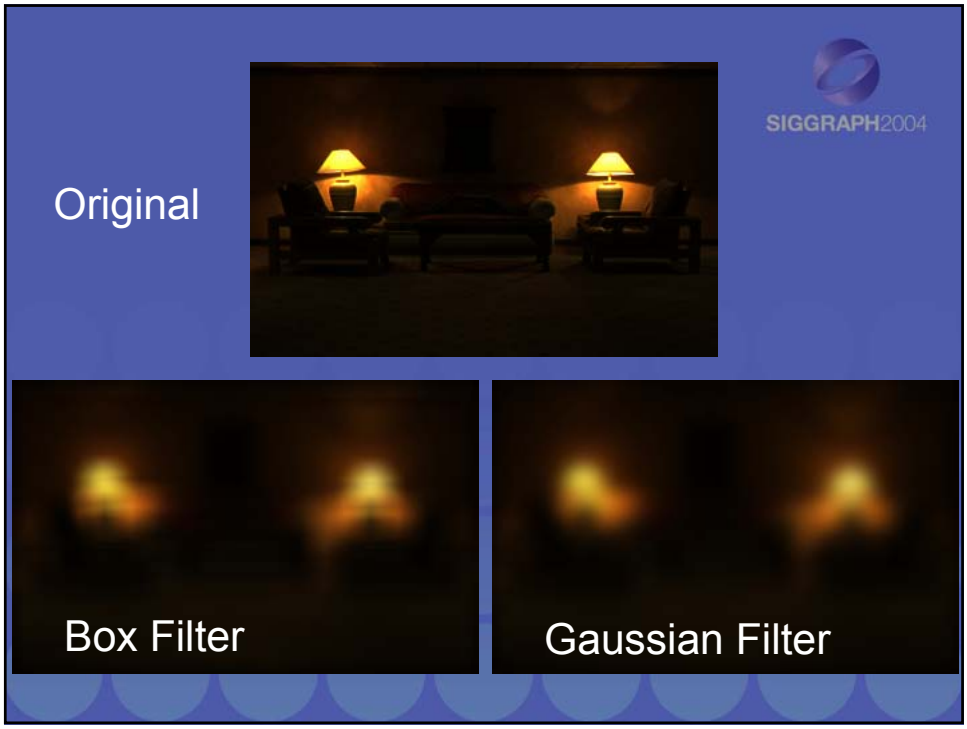


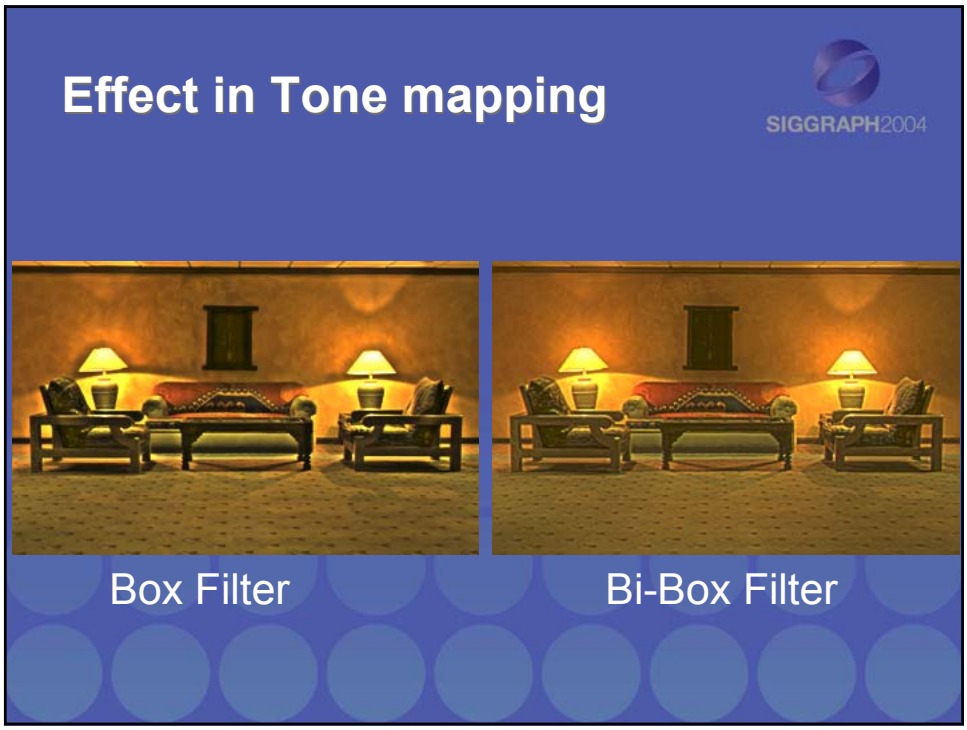
Tapered Box filter
[Pattanaik 2002: Susan Filter]

Gaussian filter
[Durand 2002: Bilateral Filter]

Original

SIGGRAPH2004

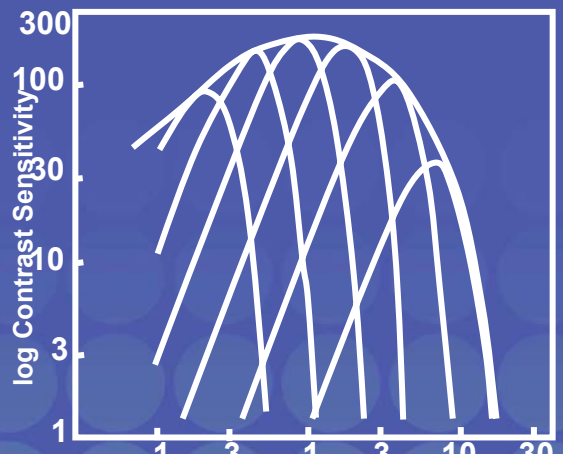

Original
SIGGRAPH2004
Box Filter
Gaussian Filter


Effect in Tone mapping
SIGGRAPH2004
Box Filter
Bi-Box Filter

Multi-scale Average

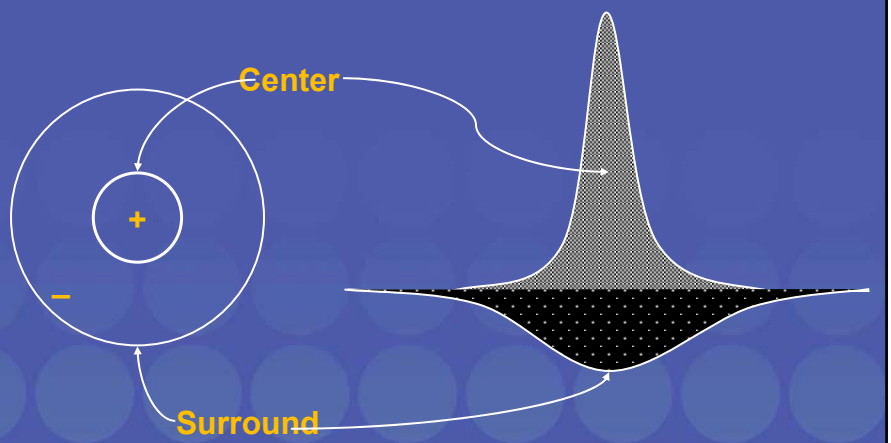
SIGGRAPH2004

Multi-Scale Model of Spatial Vision



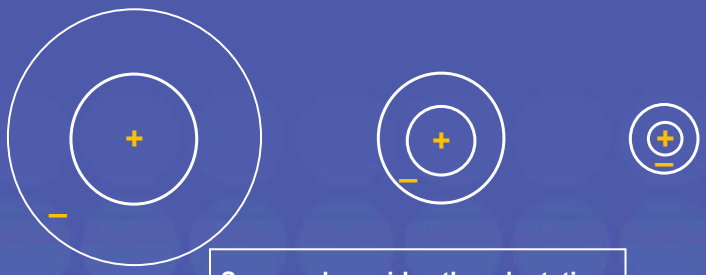
Center-Surround mechanisms

SIGGRAPH2004



Surround provides the adaptation.

Multiple Center-Surround mechanisms



Surround provides the adaptation.

Used by:

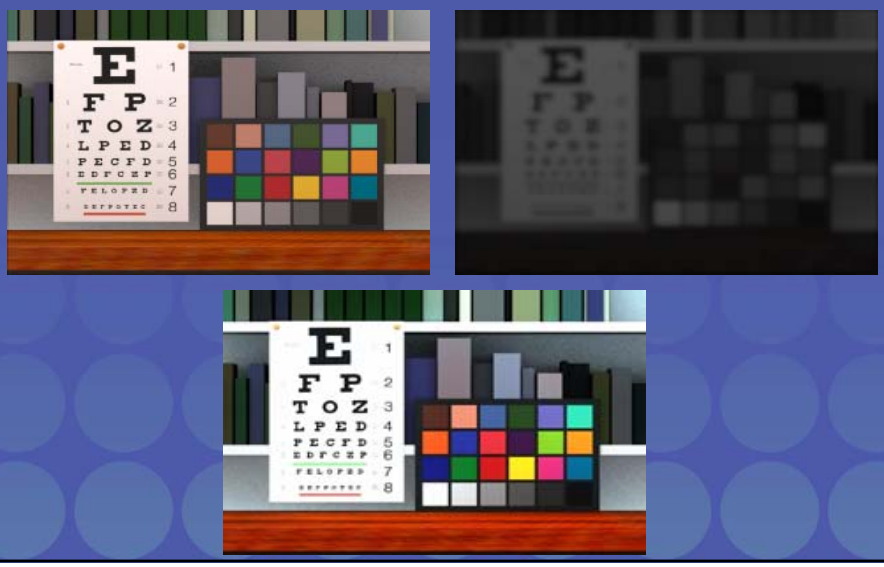
- Jobson 1996
- Pattanaik 1998
- Reinhard 2002
- Ashikman 2002

Color Appearance

Colorimetry *vs.* color appearance

- colorimetry
  - predicts visual matches of simple patches under identical viewing conditions
- color appearance
  - attempts to predict visual matches between complex scenes under different viewing conditions

Color Appearance



Chromatic adaptation

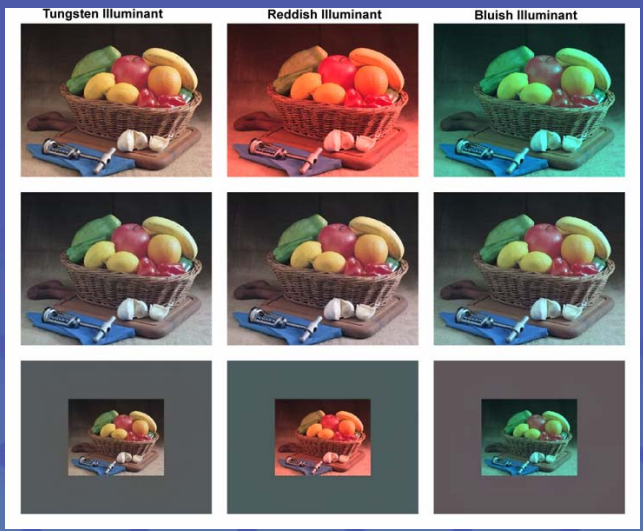
SIGGRAPH2004

Raw “Radiance” Images

Adapted “Perceptual” Images

Spatial aspects of chromatic adaptation

SIGGRAPH2004



Color Appearance Correlates

Brightness : Q

Colorfulness : M

Hue : h

Lightness : J

Chroma : C

Saturation : S

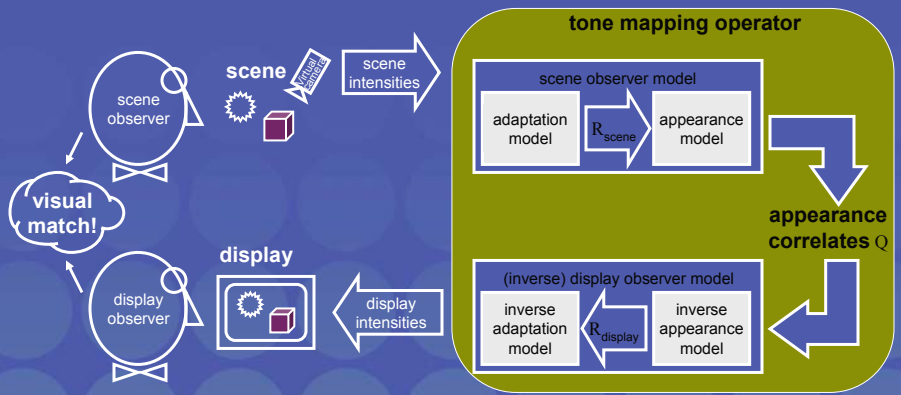
Colorimetry vs Color Appearance

- Colorimetry:
  - Predicts visual matches under identical conditions
- Color Appearance Model:
  - Predict color appearance in complex scenes
  - Determine adaptation in a complex scene
  - Define surround in a complex scene
  - Take into account adaptation and spatial vision in a unified context

A new model for Tone Reproduction

- Integrated model of
  - adaptation
  - spatial vision
  - color appearance

Desirable Tone Reproduction Operator



Tone reproduction

Match range of image intensities to those of the display device.

Possible approaches:
- Linear scaling
- Image formation
- Human visual system

Caveats

I will show results for several tone reproduction operators

They are all based on my own implementation/ interpretation

I will avoid direct comparison between operators, i.e. no side-by-side comparisons

Linear scaling

Linear | Linear + cropping

Image formation

Assumption: pixel intensities are the result of light reflecting off surfaces and hitting an image sensor.

A much simplified model: $L = i * r$

- L: pixel intensity
- i: illuminance
- r: surface reflectance

Surface reflectance

Surfaces reflect a fraction of between 0.005 and 1.0 of incident light (Stockham 1972)

Surface reflectance does not by itself create high dynamic range images.

Illuminance

The amount of light incident upon a surface could have any range of values (as long as these values are positive)

$L = i * r$: If L has a high dynamic range, then this is because i has a high dynamic range

Intensity vs. Density

In photography images are often represented as densities, i.e. in the logarithmic domain:

$$D = \log(L) = \log(i) + \log(r)$$

Note that D could be negative

The notation given here will be used throughout this section of the course. D is a 'density image'

Frequency domain analysis

One more observation about reflectance and illuminance:

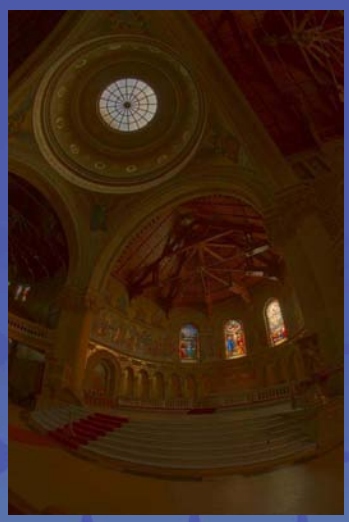
- Reflectance tends to exhibit high spatial frequencies
- Illuminance is slowly varying over surfaces

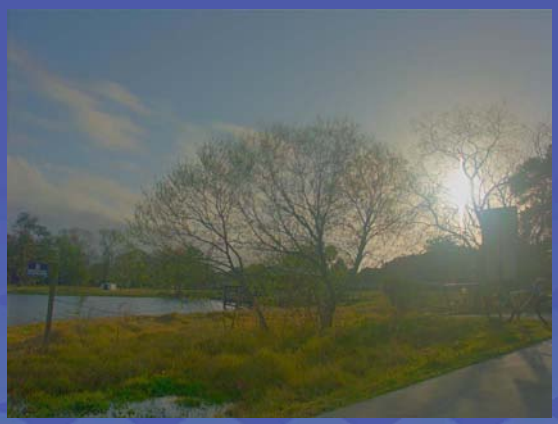
Oppenheim 1968

Given a set of pixel intensities, we wish to attenuate the illuminance component while keeping the reflectance component

Solution: Take the Fourier transform of a density image and attenuate the low frequencies. Then exponentiate to form a displayable image

Oppenheim 1968





Durand and Dorsey 2002

Split image into base- and detail layers:

- Use edge-preserving smoothing operator to filter density image and call result ‘base layer’
- Divide density image by base layer and call result ‘detail layer’

Durand and Dorsey 2002

- Note that base layer is high dynamic range and could be viewed as the illuminance component
- The detail layer is low dynamic range and could be viewed as the reflectance component

Durand and Dorsey 2002

Input image

Smoothed result

Durand and Dorsey 2002

Related techniques

- Choudhury and Tumblin (2003): Trilateral filter
  - A refinement of the edge-preserving smoothing operator used by Durand and Dorsey
- Pattanaik and Yee (2002): Gain control operator
  - Different kernel shapes

Alternatively…

Illuminance is slowly varying over a surface, whereas reflectance produces sharp discontinuities

In other words: large image gradients are caused by reflectance edges

Horn 1974

Model of lightness (i.e. a model of the perception of surface reflectance):

- Differentiate a density image
- Threshold density gradients (keep large gradients)
- Integrate image (numerical process)
- Exponentiate

Horn 1974

Original Processed

Image formation

This model of image formation breaks down

- if specular highlights are visible in the image
- if light sources are directly visible
- if image depicts fluorescent materials

In fact, we might want to attenuate large gradients rather than preserve them!

Fattal et al 2002

Gradient domain compression:

- Differentiate density image
- Attenuate large gradients
- Integrate
- Exponentiate

Fattal et al 2002

Human visual system

Sumant already explained the principles by which the HVS deals with high dynamic range scenery

Here, we build on those insights to classify HVS-based tone reproduction operators

Global vs. Local

We may apply compressive functions directly in image space

- Global: use the same function for each image pixel (use a global adaptation value)
- Local: modulate compressive function by pixel neighborhood (use a local adaptation value)

Global operators

Since each pixel is treated independently:

- Efficient and fast
- Suitable for medium dynamic range imagery

Global operators

Only a small number of different functional shapes are known:

- Linear scaling
- Logarithmic scaling
- Histogram-based compression
- Sigmoidal functions (discussed by Sumant)

Linear scaling

- Already shown an ad-hoc linear scaling function at the start of this section
- Ward 1994 and Ferwerda et al 1996: Linear scaling based on TVI functions
  - Bring the log average luminance to a sensible display level
  - Better than ad-hoc scaling, not suitable for very high dynamic range images

Ward 1994 (linear scaling)

Ferwerda et al 1996

Applied different prescaling factors

Logarithmic scaling

Simple version:

$$L_d = \frac{\log(L_w + 1)}{\log(L_{w,\max} + 1)}$$

Drago et al 2003:

$$L_d = \frac{\log_b(L_w + 1)}{\log_b(L_{w,\max} + 1)}$$

b is a function of the pixel intensity and is modulated by a user parameter to steer overall appearance

Logarithmic compression

Drago et al 2003

Histogram adjustment

Ward et al (1997) use the shape of the image's histogram

- Compute histogram
- Compute cumulate histogram
- Result is a monotonically increasing function
- Reshape this function to avoid slopes greater than 1
- Remap luminances according to this function

Ward et al 1997

Sigmoids

S-shaped compression function used in a large number of tone reproduction operators.

Discussed in detail by Sumant in previous section. Will omit discussion here.

Global operators

Recap:

- Easy to implement
- Fast
- Some provide very reasonable amounts of compression and plausible results are frequently obtained
- Useful for medium to high dynamic range images

Local operators

Sometimes the mismatch between HDR image and display device is very large

More compression afforded by local operators

Often at the cost of artifacts

Local operators

Two approaches:

- $L_d(x,y) = s(x,y)L_w(x,y)$
- Replace global adaptation value with a per-pixel local adaptation value

Local operators

- Chiu et al, Rahman et al, iCAM model:

$$L_d(x,y) = s(x,y)L_w(x,y)$$

$$s(x,y) = f\left(\frac{1}{L_w^{\mathrm{LPF}}(x,y)}\right)$$

i.e. divide the image by a low-pass filtered version of itself

Chiu et al 1993

Rahman et al

Johnson and Fairchild (iCAM)

Local operators

When dividing by a low-pass filtered image, we need two things to minimize artifacts:

- A very large kernel size
- Reduce the weight of the LPF image

Global to local

- Replace global average with local average within global operators
- Sigmoids! General shape:

$$L_d(x,y) = \frac{L_w(x,y)^n}{L_w^{\mathrm{LPF}}(x,y)^n + L_w(x,y)^n}$$

Local adaptation

There is a catch, though:

- The size of the low-pass filter kernel is important.
- Too large and haloing artifacts will occur
- Too small and compression will not be better than global operators

Low-pass filter kernel size

- For each pixel the size of the LPF kernel should be such that it does not overlap with sharp discontinuities
- But we still want to average over the largest spatial area for which the above is true (which may be different for each pixel)
- In practice often a small kernel size!

How to compute?

- Multi-scale techniques using difference of Gaussians and scale selection (Reinhard 2002, Ashikhmin 2002)
- Edge-preserving smoothing operator (bilateral filter, mean shift algorithm, LCIS…)

Reinhard et al 2002

Ashikhmin 2002

Global vs. local

Sigmoid with global adaptation

Sigmoid with local adaptation

Conclusions

Only a few fundamentally different approaches to tone reproduction exist

- Based on image formation
  - Frequency domain
  - Gradient domain
- Based on the human visual system
  - Global operators
  - Local operators

Conclusions

Trade-offs exist between:

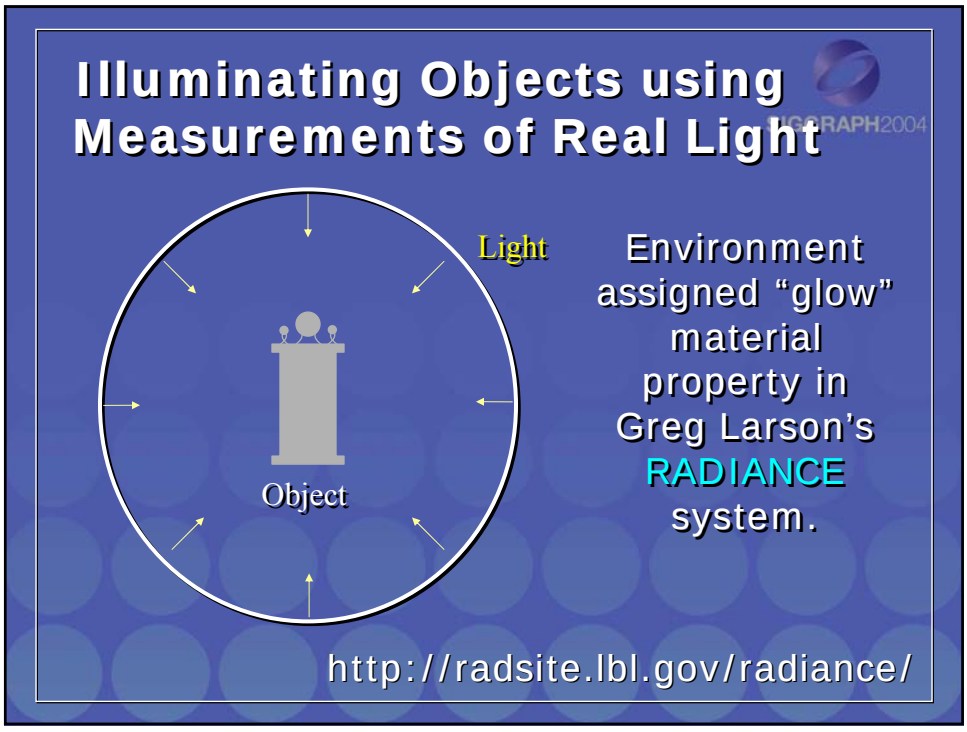
- Amount of compression
- Presence of artifacts
- Computation time

We have no good algorithms for deciding which operator is most suited for any particular task!


SIGGRAPH2004
HDR Image-Based Lighting for Synthetic and Real Scenes
Paul Debevec
University of Southern California
Institute for Creative Technologies
www.debevec.org/HDRI2004


scene.oct
256 sampling...
512 sampling...
done:
CG Objects Illuminated by a Traditional CG Light Source


Real-World HDR Lighting Environments
Funston Beach
Eucalyptus Grove
Uffizi Gallery
Grace Cathedral
Lighting Environments from the Light Probe Image Gallery:
http://www.debevec.org/Probes/

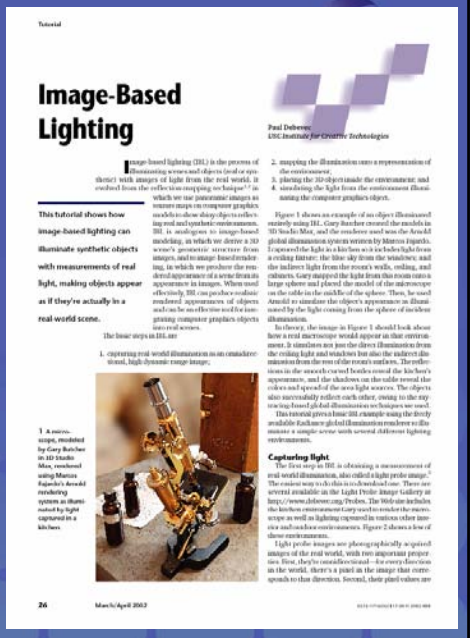

Illuminating Objects using Measurements of Real Light
SIGGRAPH2004
Light
Object
Environment assigned "glow" material property in Greg Larson's RADIANCE system.
http://radsite.lbl.gov/radiance/

Paul Debevec. A Tutorial on Image-Based Lighting. IEEE Computer Graphics and Applications, Jan/Feb 2002.

IBL in Radiance Tutorial

- In Jan/Feb 2002 Computer Graphics and Applications and the IBL Course Notes

  and www.debevec.org under “Publications”



Tutorial

Image-Based Lighting

Paul Debevec
USC Institute for Creative Technologies

This tutorial shows how image-based lighting can illuminate synthetic objects with measurements of real light, making objects appear as if they're actually in a real-world scene.

Putting the probe onto the sphere


```
# Lighting Environment
# specify the probe image and how it is mapped onto
# geometry

void colorpict hdr_env
7 red green blue rnl_probe.hdr angmap.cal u v
0
0

# specify a "glow" material that will use this image hdr_env glow env_glow 0 0
4 1 1 1 0

# specify a large inward-pointing box for the HDR envir.
!genbox env_glow box 500 500 500 -i | xform -t -250 -18 -250
```

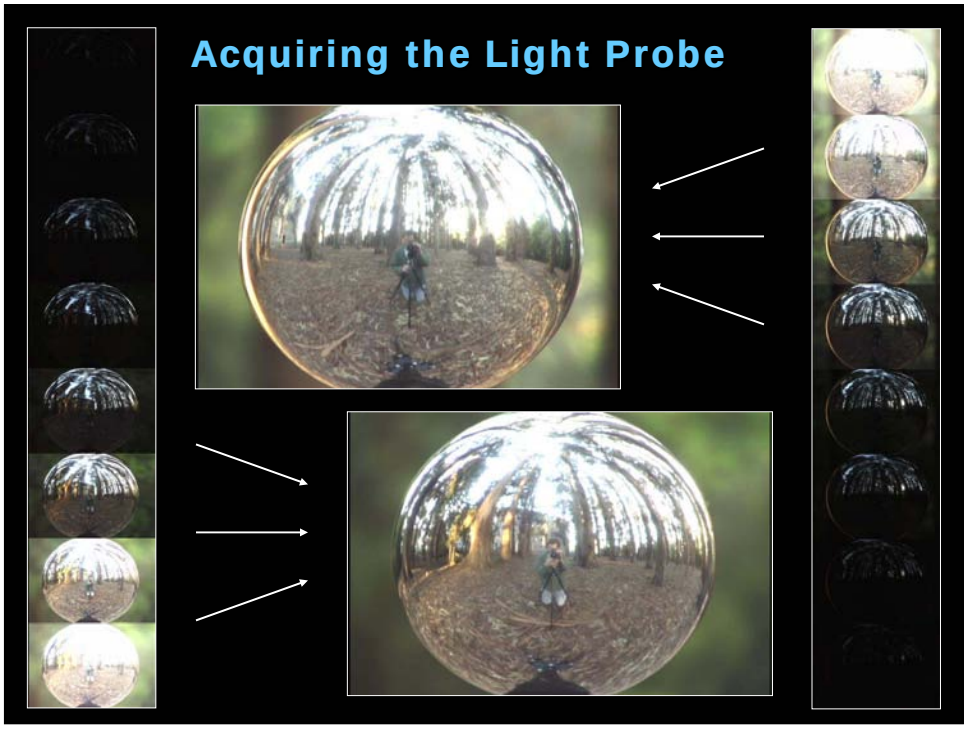
Light Probe Coordinate Mapping

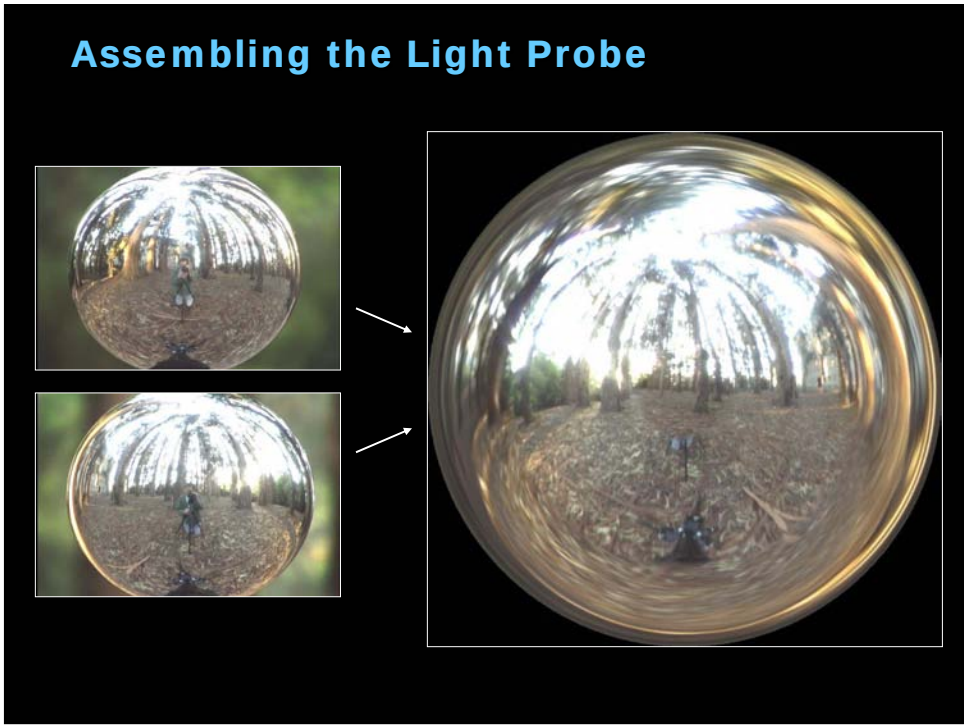
{angmap.cal Convert from directions in the world to coordinates on the angular sphere image

-z is forward (outer edge of sphere)
+z is backward (center of sphere)
+y is up (toward top of sphere)}

```
norm = 1/sqrt(Py*Py + Px*Px + Pz*Pz);
DDy = Py*norm;
DDx = Px*norm;
DDz = Pz*norm;
r = 0.159154943*acos(DDz)/sqrt(DDx*DDx + DDy*DDy);

u = 0.5 + DDx * r;
v = 0.5 + DDy * r;
```


Making
Rendering with Natural Light
SIGGRAPH 98 Electronic Theater


Acquiring the Light Probe


Assembling the Light Probe


RNL Environment mapped onto interior of large cube

Rendering with Natural Light Source Files

This directory contains the original scene files for Paul Debevec's animation "Rendering with Natural Light" shown at the SIGGRAPH 1998 Electronic Theater in Orlando, Florida.

angmap.cal	Angular map equation for mapping light probe to the environment
anim.vf	Animation camera path viewfile
credits.tif	RNL credits card TIF image
gensupl.sh	Sphere support stand generator script
gensupla.sh	Central sphere support stand generator script
marble.hdr	Texture map for the pedestal
optrad	RADIANCE options for rendering the animation
pflare.c	C program for pflare, the HDR image blurring program
pflare	Intel Linux binary for pflare, the HDR image blurring program
rnl_probe.hdr	UC Berkeley Eucalyptus Grove light probe image
rnl_scene.rad	Main scene file for the spheres on the pedestal
texmap0.cal	Texture map equation for mapping marble.hdr onto pedestal
rnl_source.tgz	gzip'ped tar archive of all these files (2,721,019 bytes)

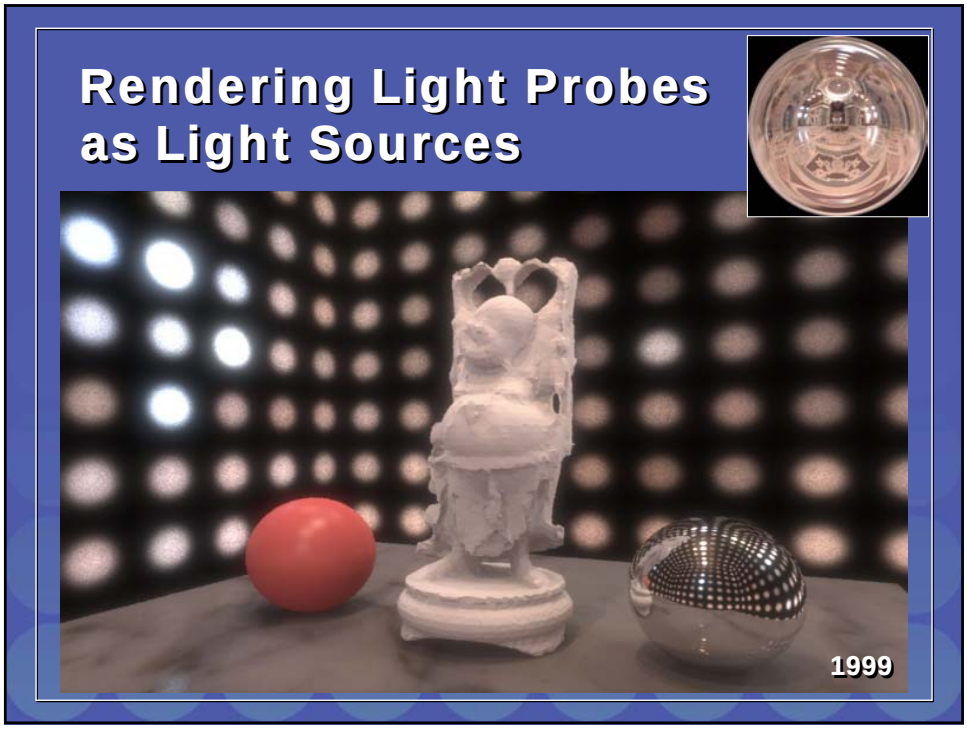
"Rendering with Natural Light" was rendered entirely with Image-Based Lighting captured through High-Dynamic Range Photography in the UC Berkeley Eucalyptus Grove.

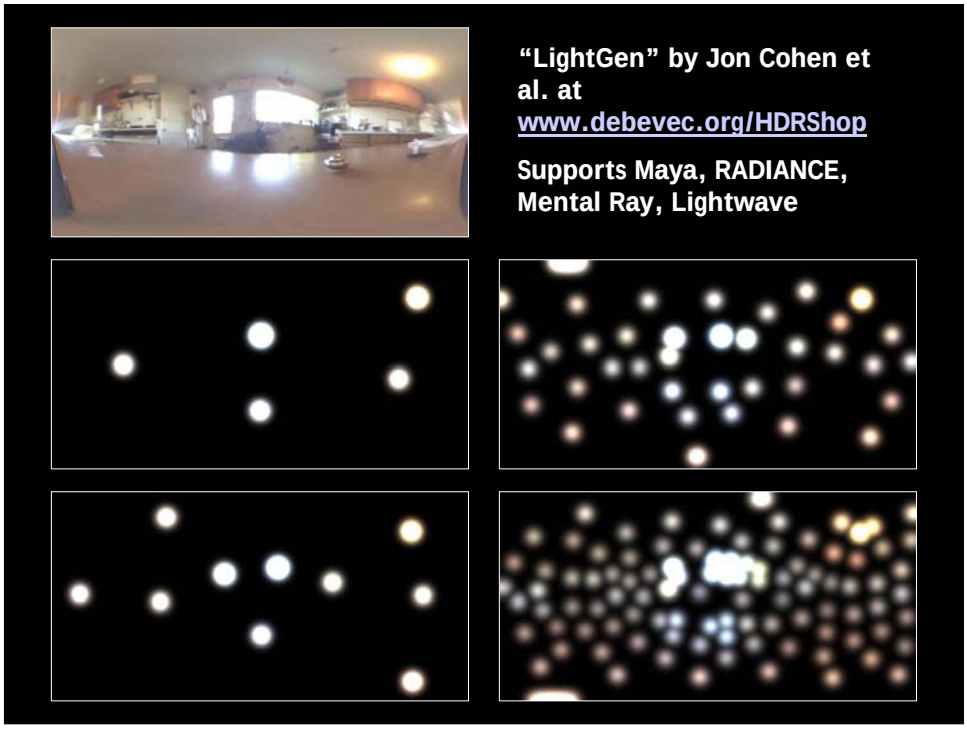
To render the animation yourself, follow the following procedure:

www.debevec.org/RNL

RNL in Real Time!

SIGGRAPH2004


Rendering Light Probes as Light Sources
1999

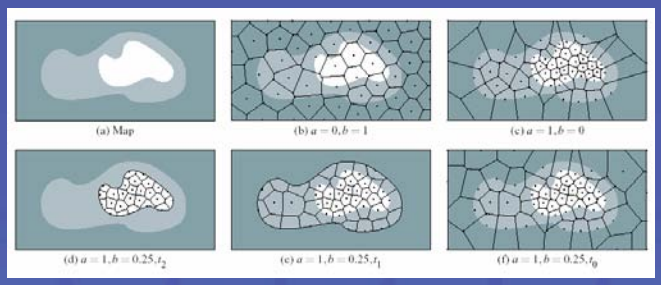

"LightGen" by Jon Cohen et al. at
www.debevec.org/HDRShop
Supports Maya, RADIANCE, Mental Ray, Lightwave

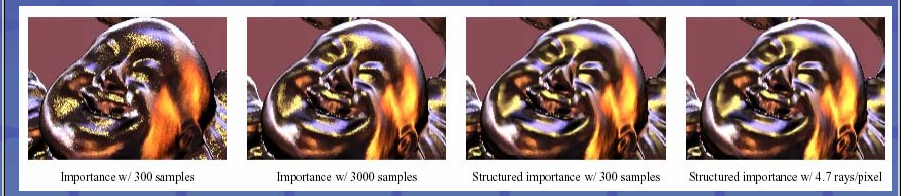
"Structured Importance Sampling of Environment Maps"

Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen

Proceedings of SIGGRAPH 2003

SIGGRAPH2004

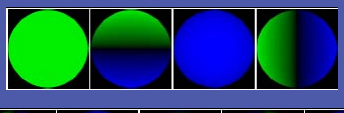


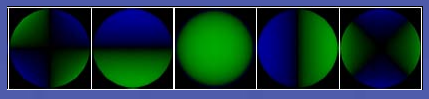


Importance w/ 300 samples | Importance w/ 3000 samples | Structured importance w/ 300 samples | Structured importance w/ 4.7 rays/pixel

Real-Time IBL with Spherical Harmonics

SIGGRAPH2004





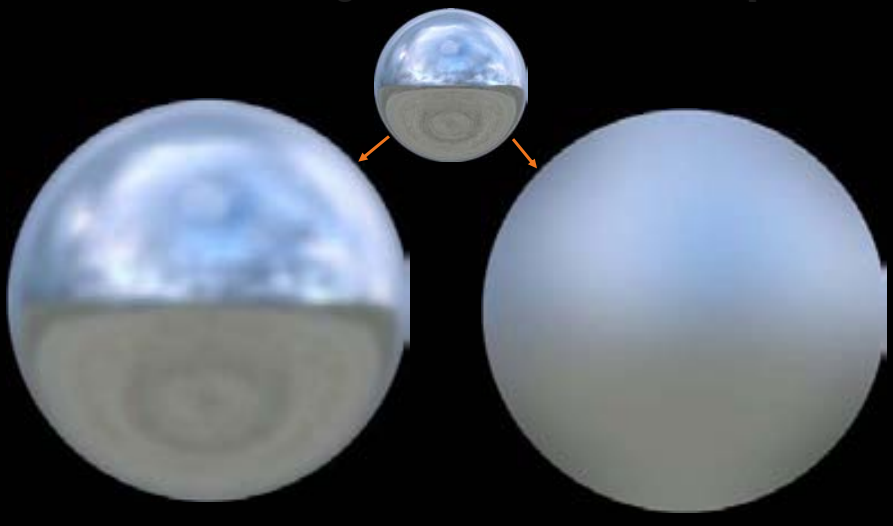
Frequency Space Environment Map Rendering
Ravi Ramamoorthi, Pat Hanrahan, SIGGRAPH2002

Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments
Peter-Pike Sloan, Jan Kautz, John Snyder, SIGGRAPH2002

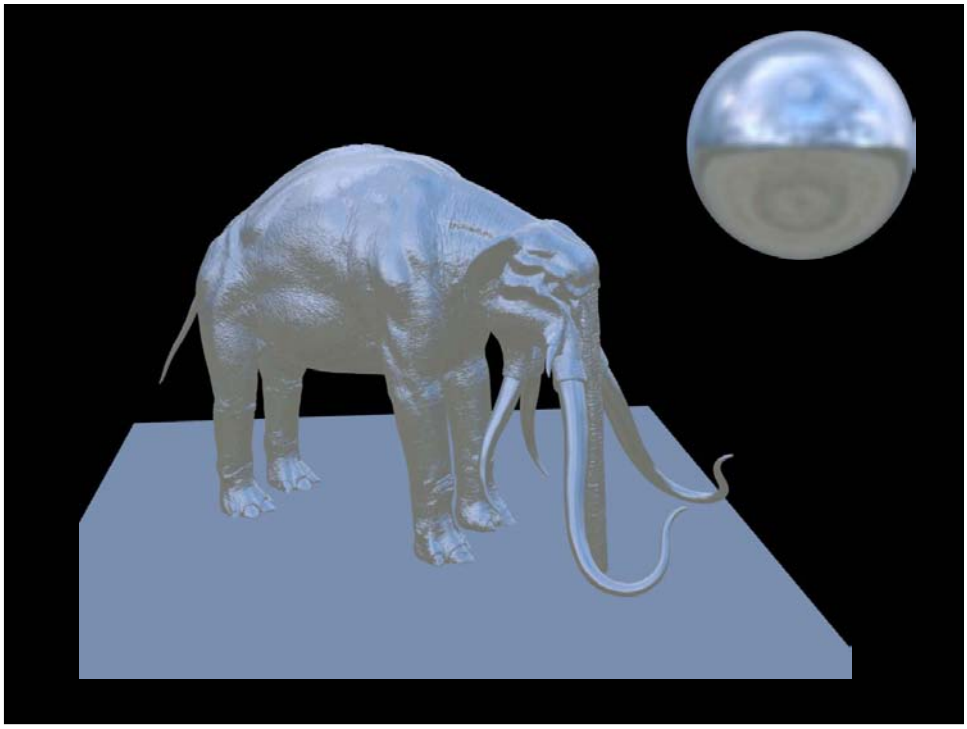
Using Image-Based Lighting in a Production Environment

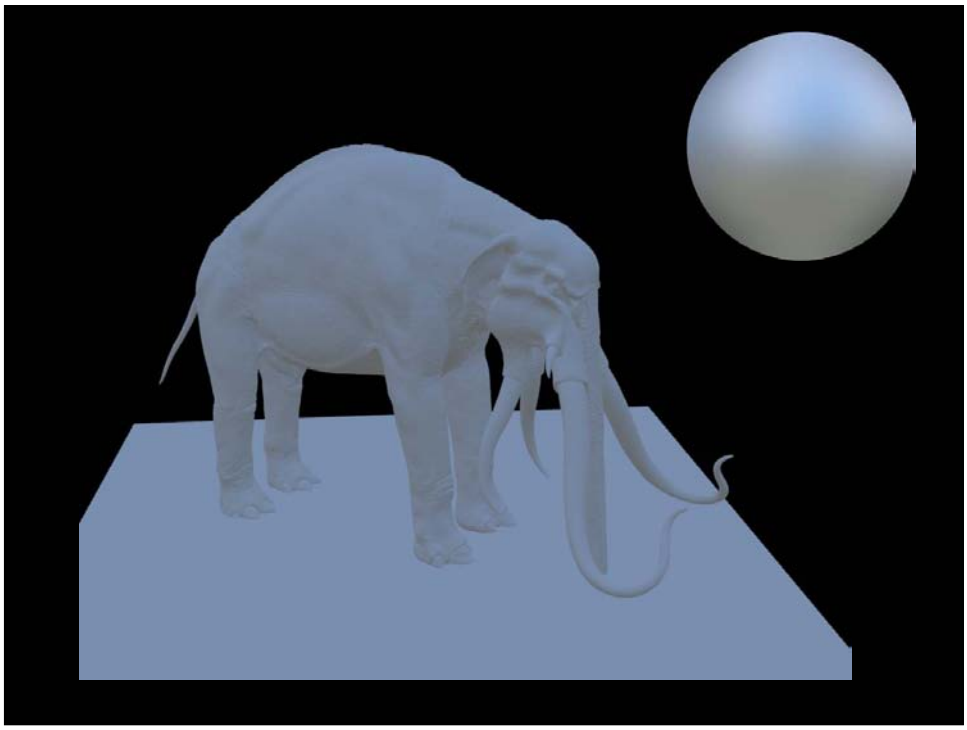
Dan Lemmon, WETA Digital

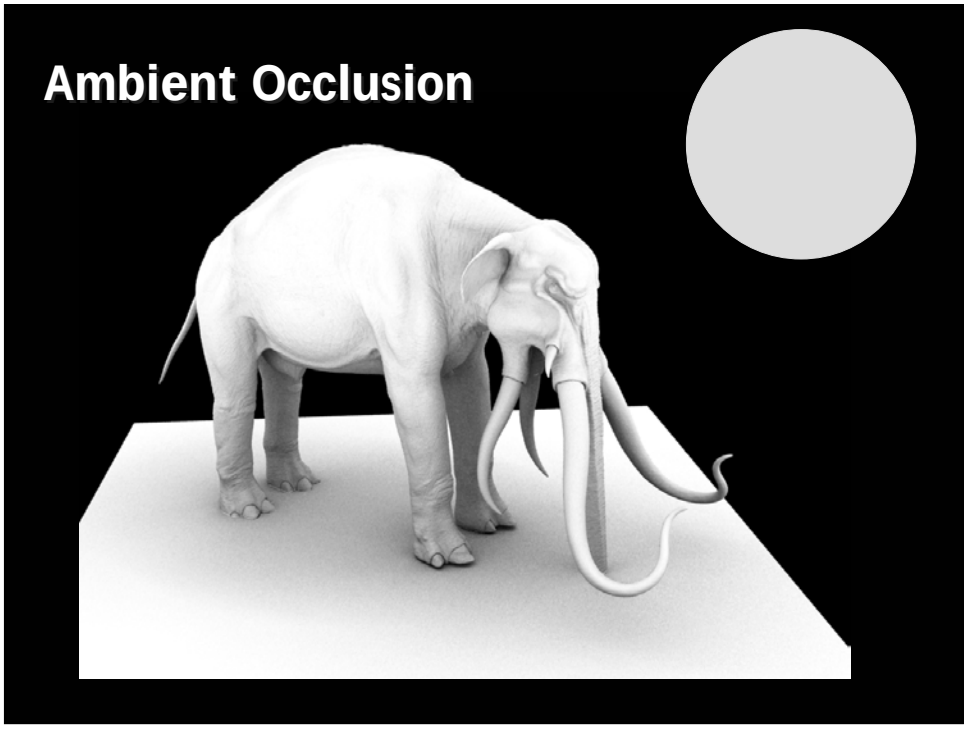
Pre-Convolving Environment Maps

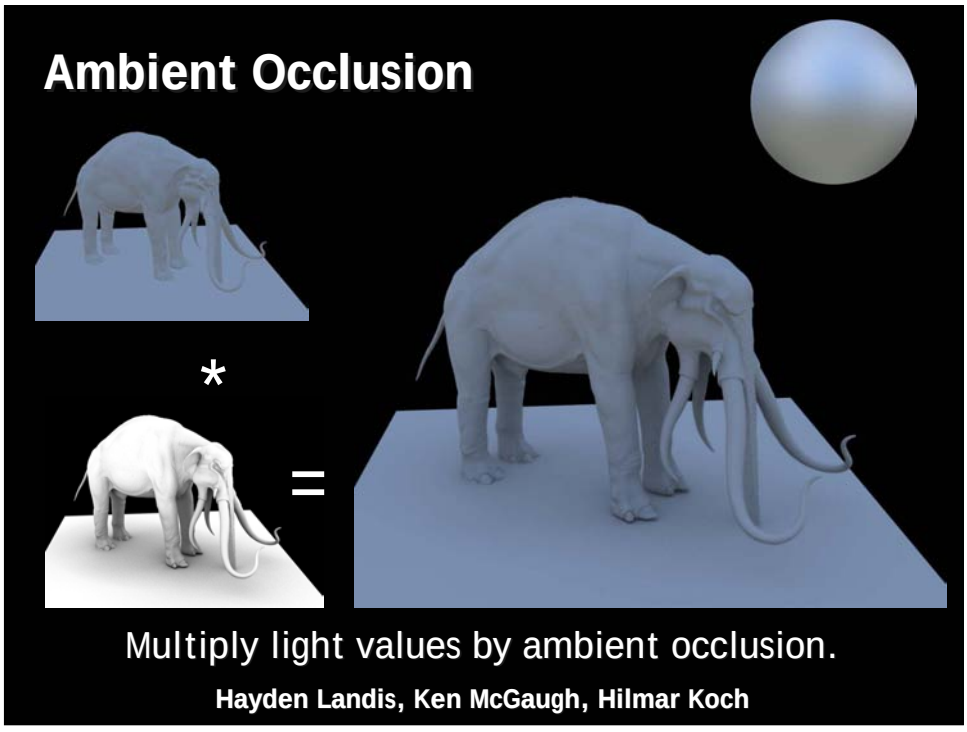


Specular Environment Map Diffuse Environment Map

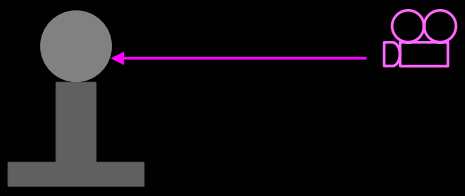





Ambient Occlusion

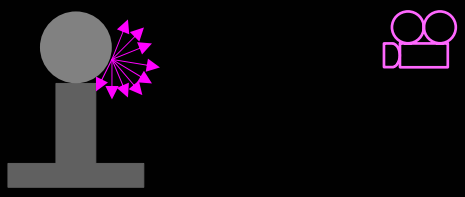

Ambient Occlusion
*
=
Multiply light values by ambient occlusion.
Hayden Landis, Ken McGaugh, Hilmar Koch

Raytracing occlusion maps



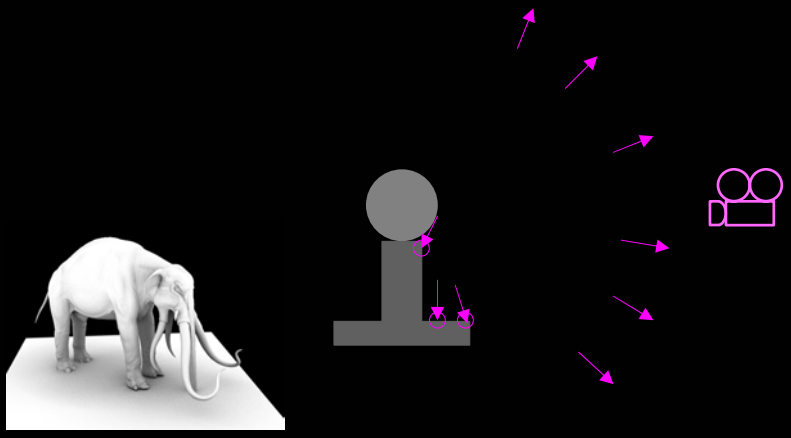
- Using a raytracer, fire rays from the camera. (Brazil, LightWave, Mental Ray, etc.).

Raytracing occlusion maps



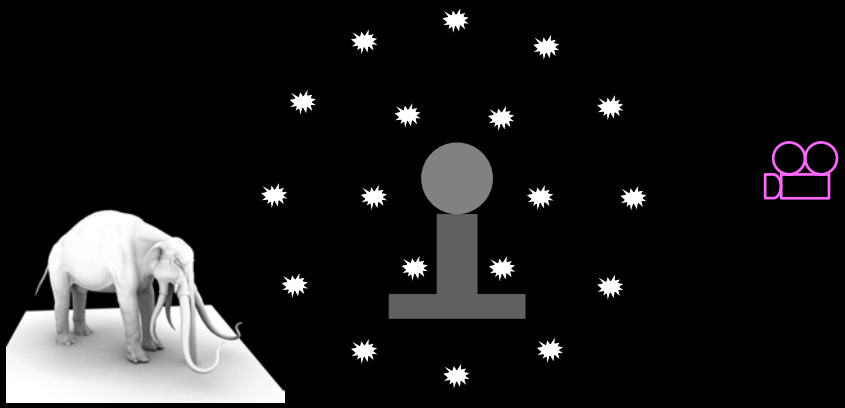
- Upon contact with geometry, spray hemispherical samples (Poisson distribution).

Raytracing occlusion maps



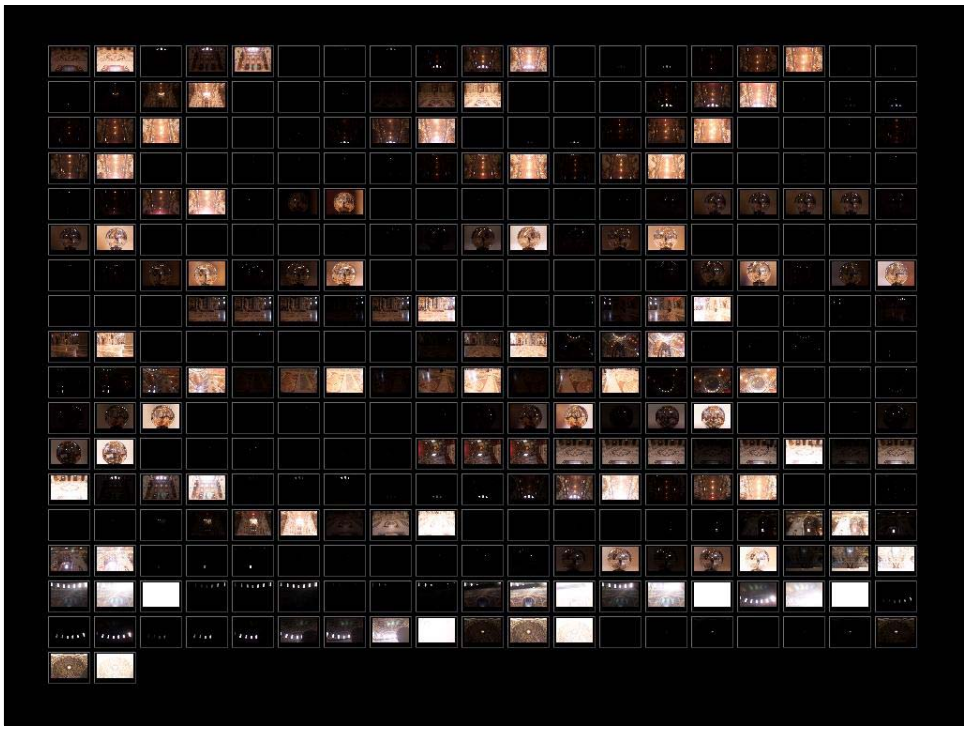
- If sample intersection distance > threshold, i++. Occlusion map value = i ÷ number of samples.

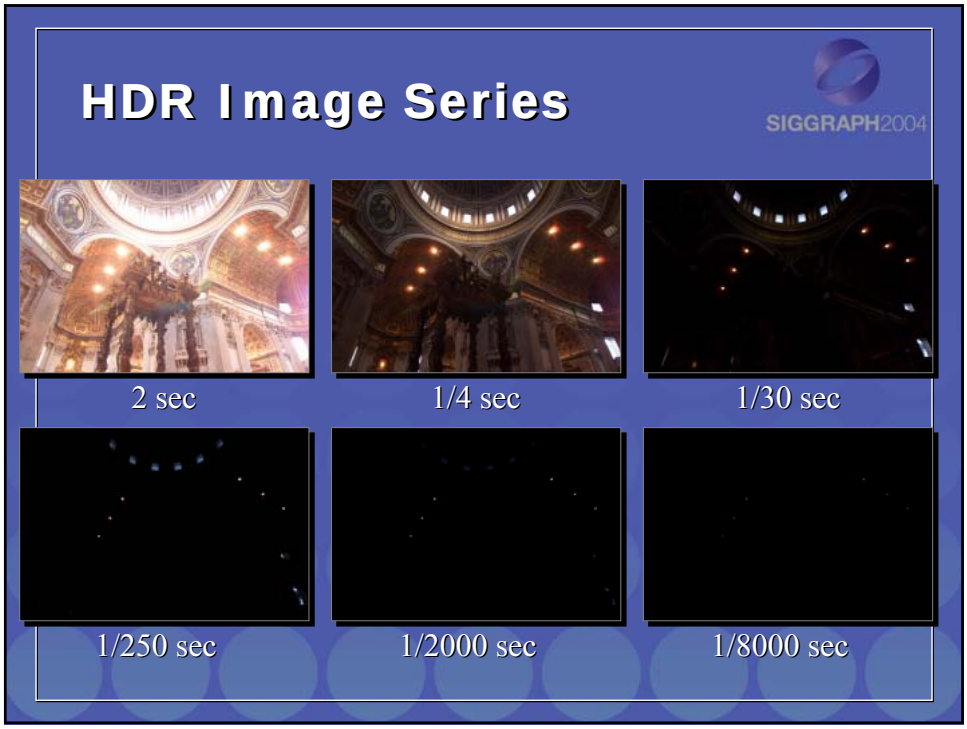
Light dome occlusion maps

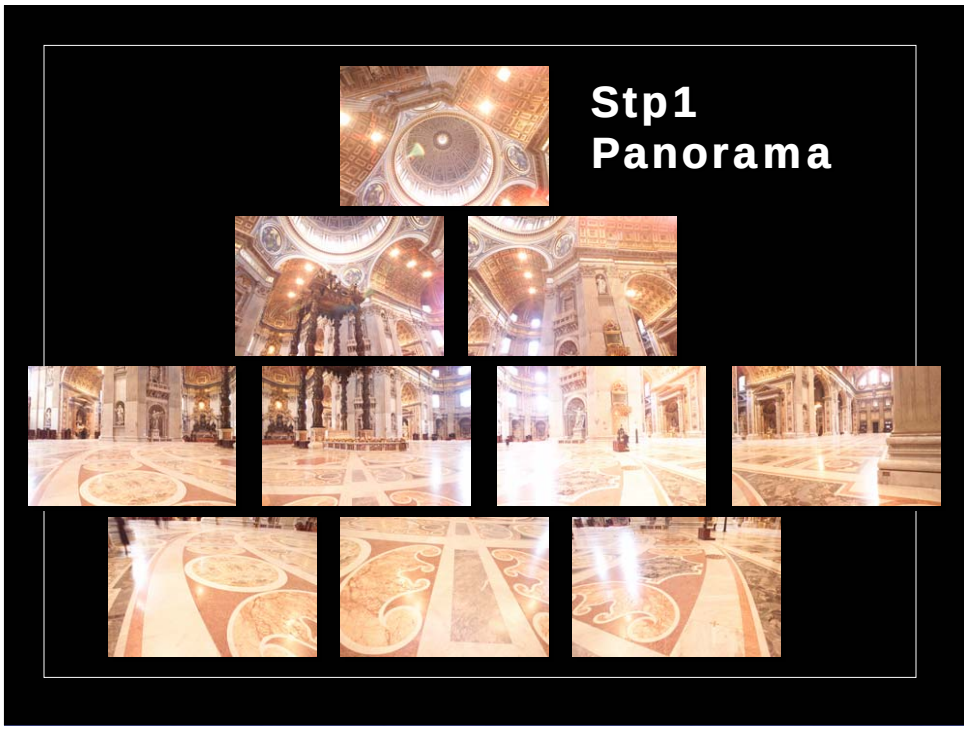


- Place lights around object in geodesic configuration. Blur z-depth shadows.


IMAGE-BASED LIGHTING IN *FIAT LUX*
Paul Debevec, Tim Hawkins, Westley Sarokin, H. P. Duiker, Christine Cheng, Tal Garfinkel, Jenny Huang
SIGGRAPH 99 Electronic Theater

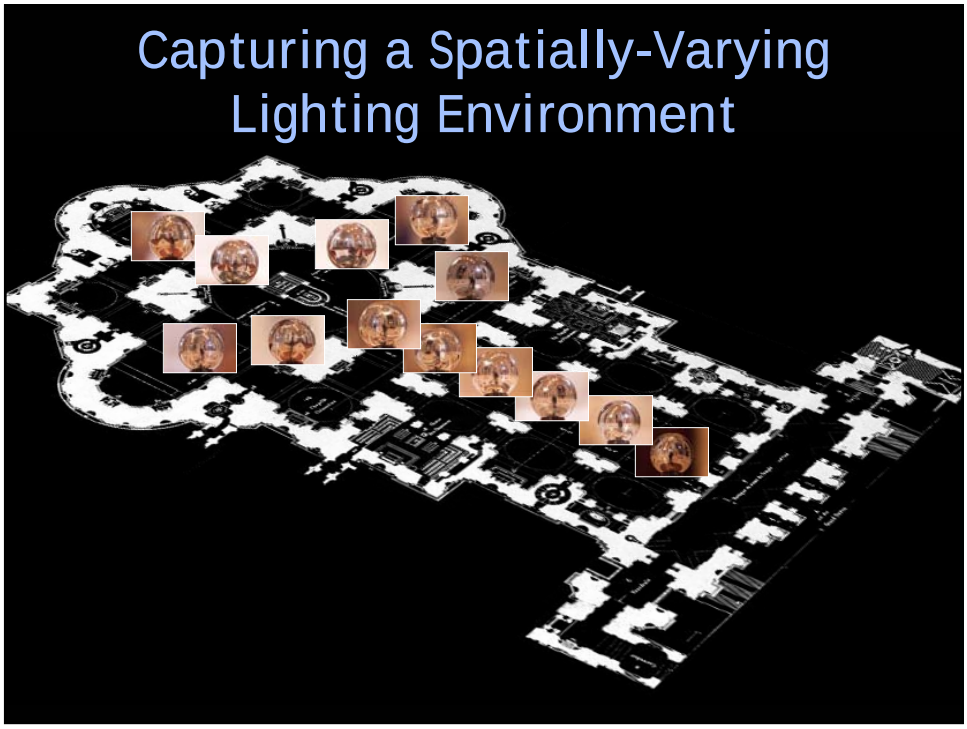



HDR Image Series
SIGGRAPH2004
2 sec
1/4 sec
1/30 sec
1/250 sec
1/2000 sec
1/8000 sec


Stp1
Panorama

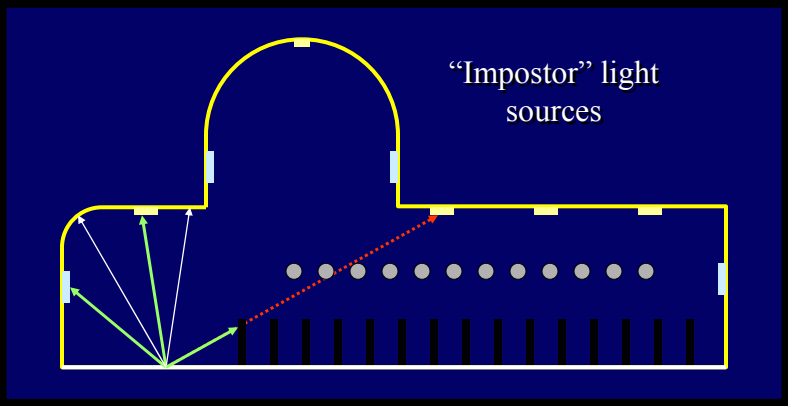
Assembled Panorama

Light Probe Images


Capturing a Spatially-Varying
Lighting Environment


Identified Light Sources
SIGGRAPH2004

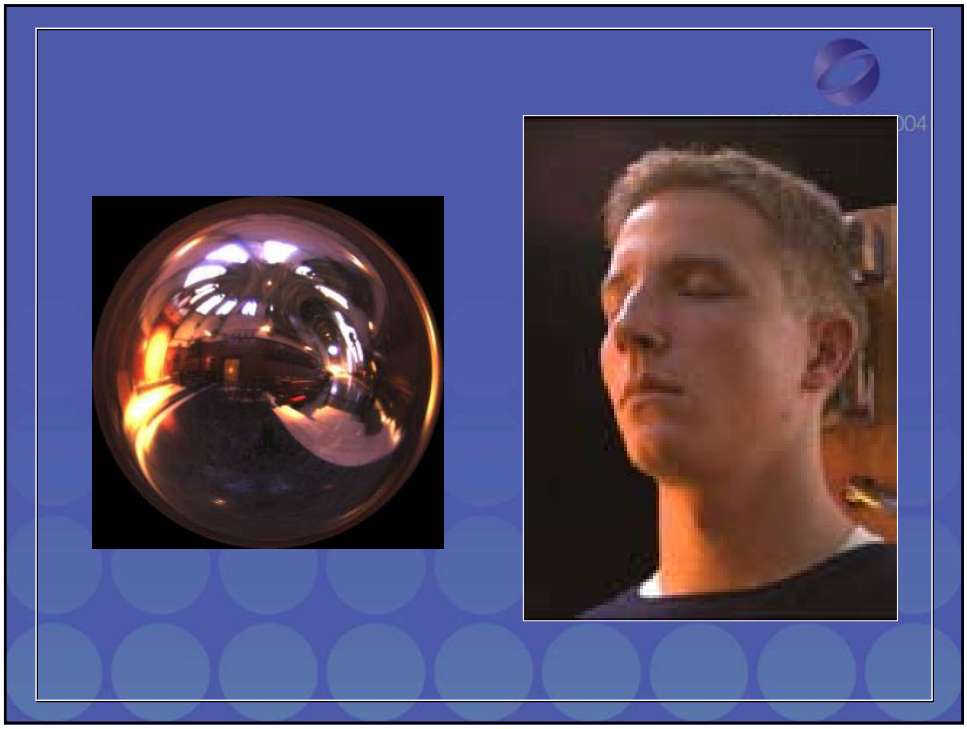
Lighting Calculation



Renderings made with Radiance: http://radsite.lbl.gov/radiance/

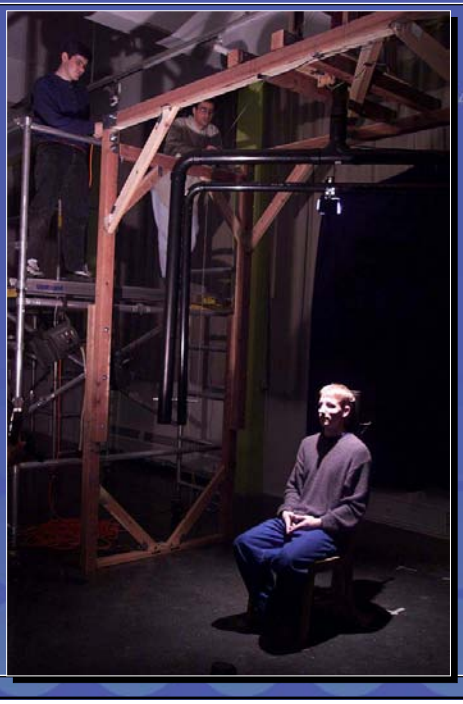
SIGGRAPH2004

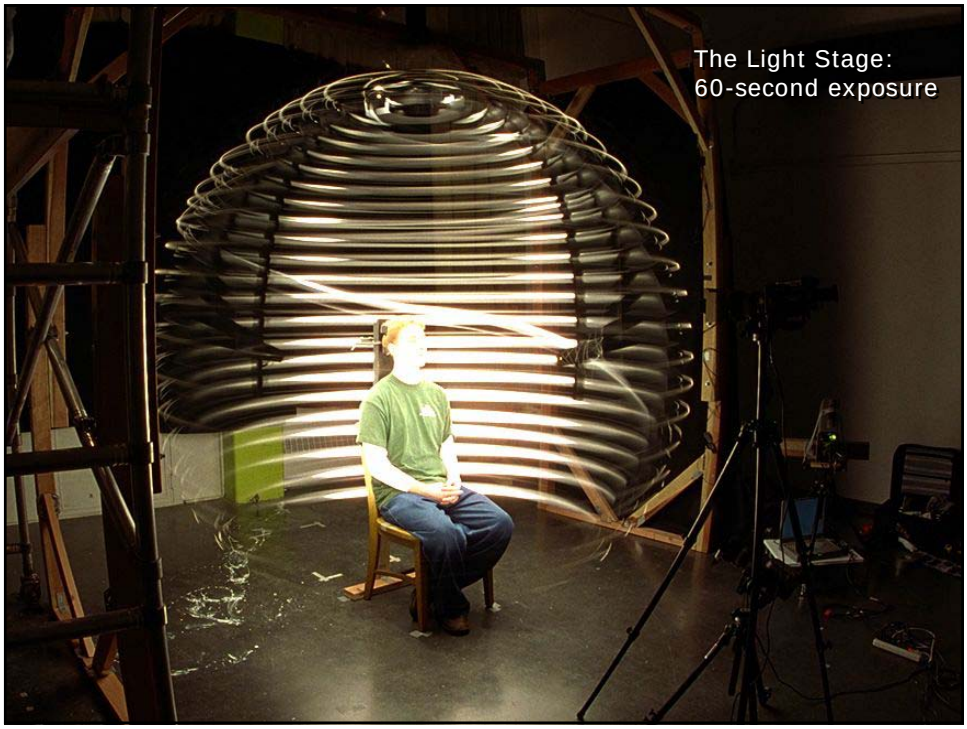
Image-Based Lighting Real Objects and Actors

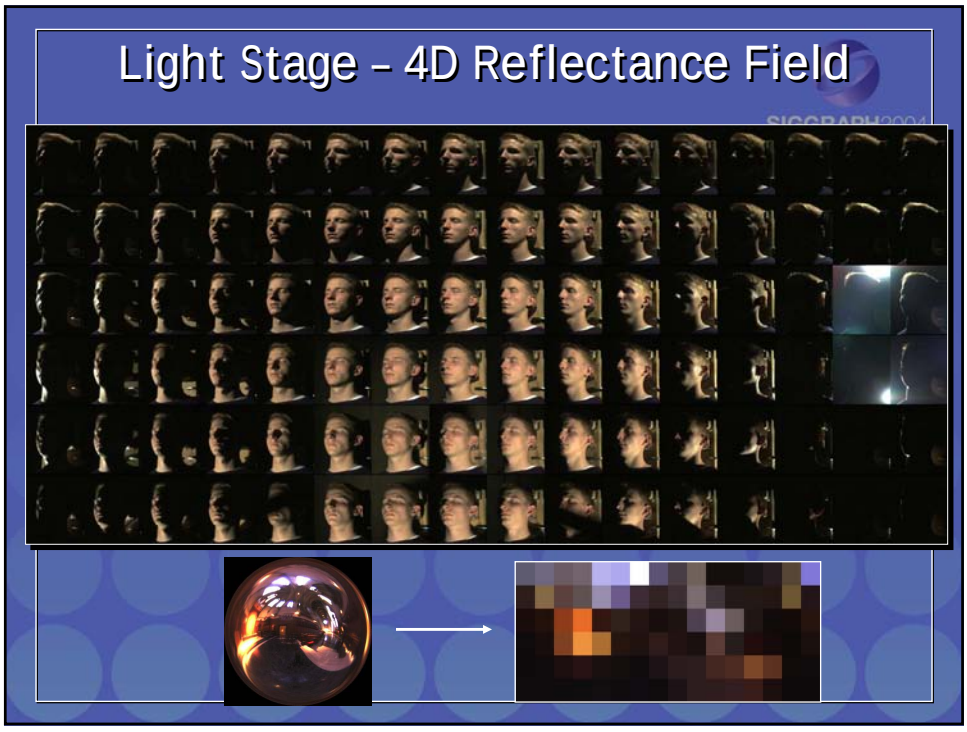


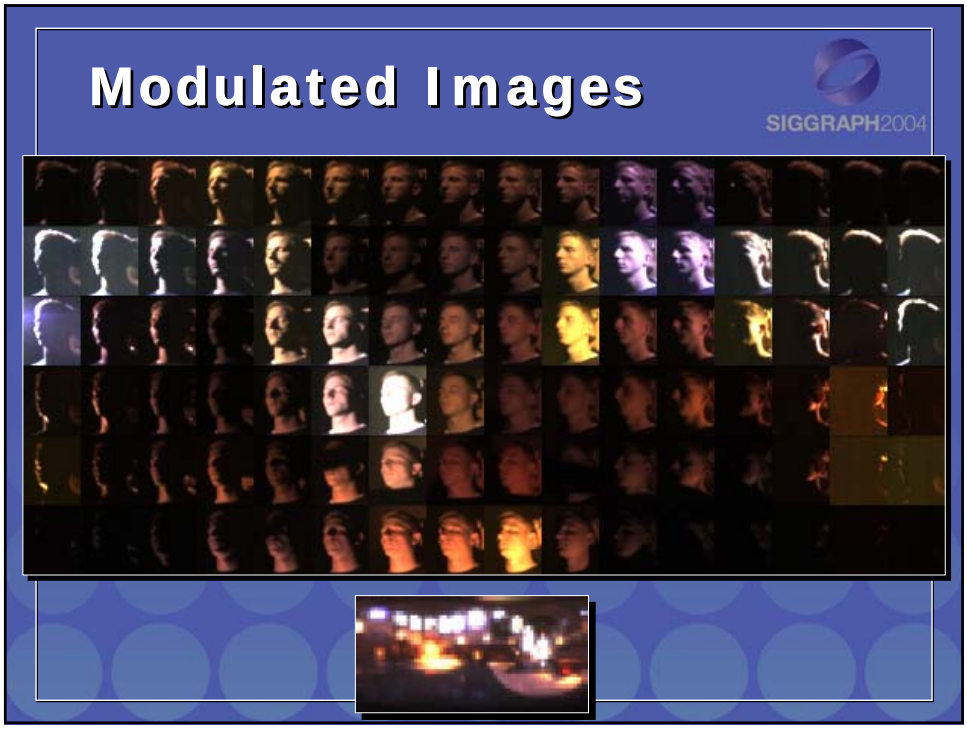
Light Stage 1.0

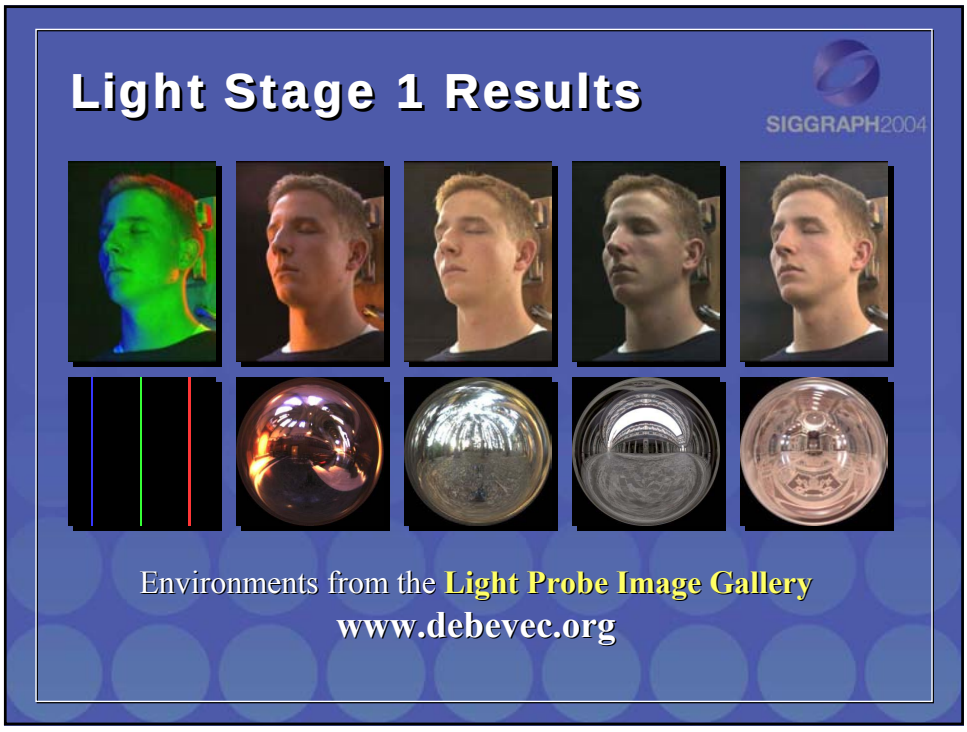
Debevec, Hawkins, Tchou, Duiker, Sarokin, and Sagar. *Acquiring the Reflectance Field of a Human Face*. SIGGRAPH 2000.

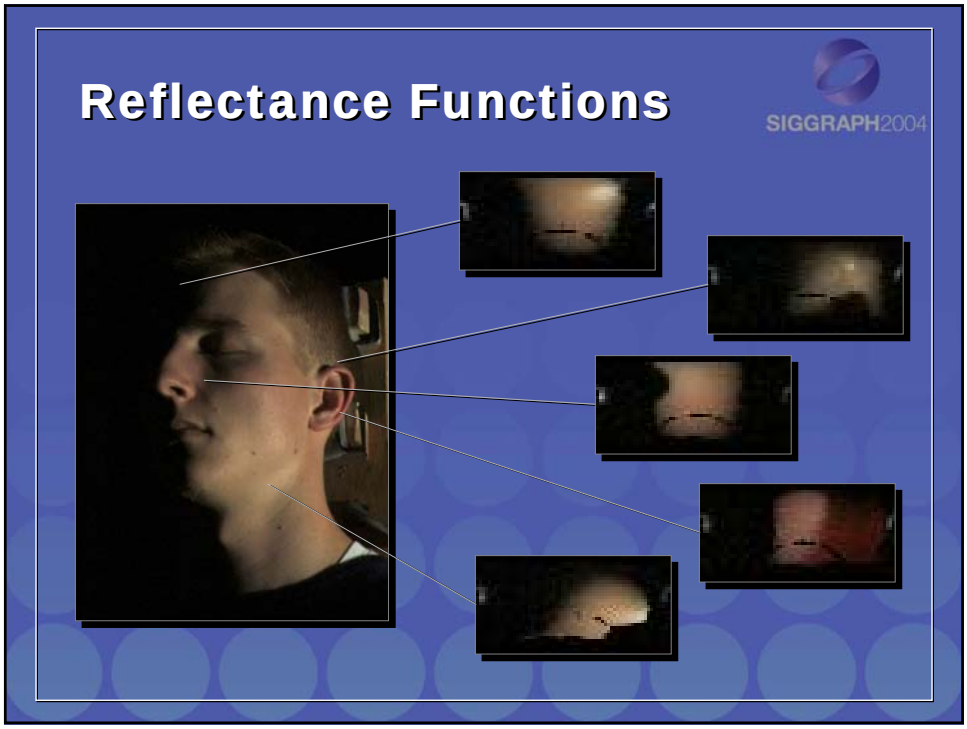


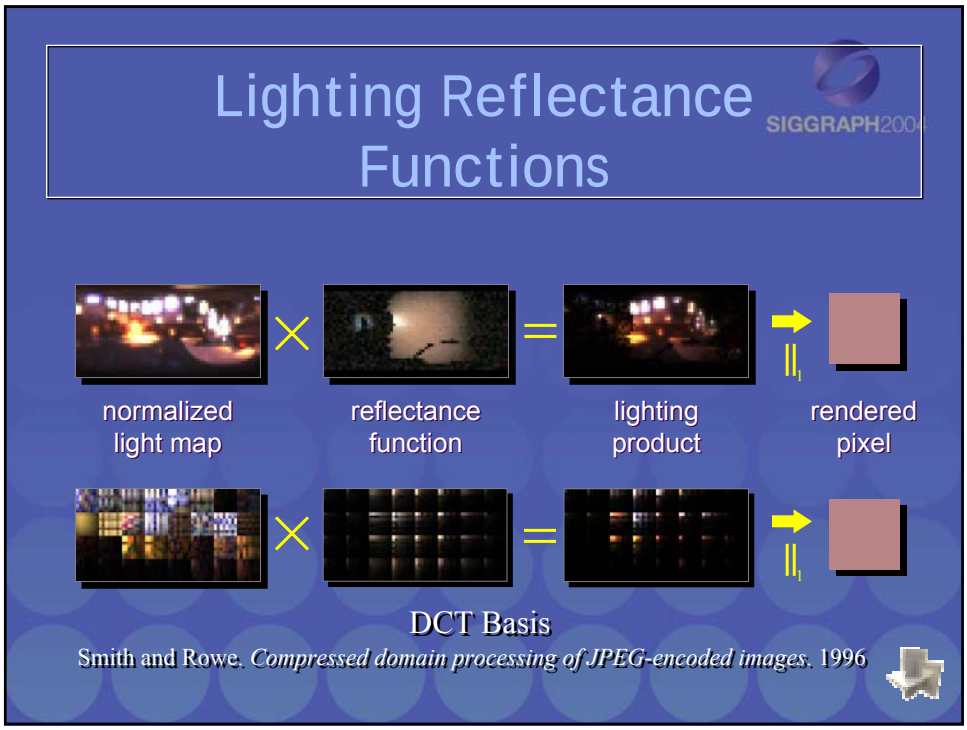

The Light Stage:
60-second exposure

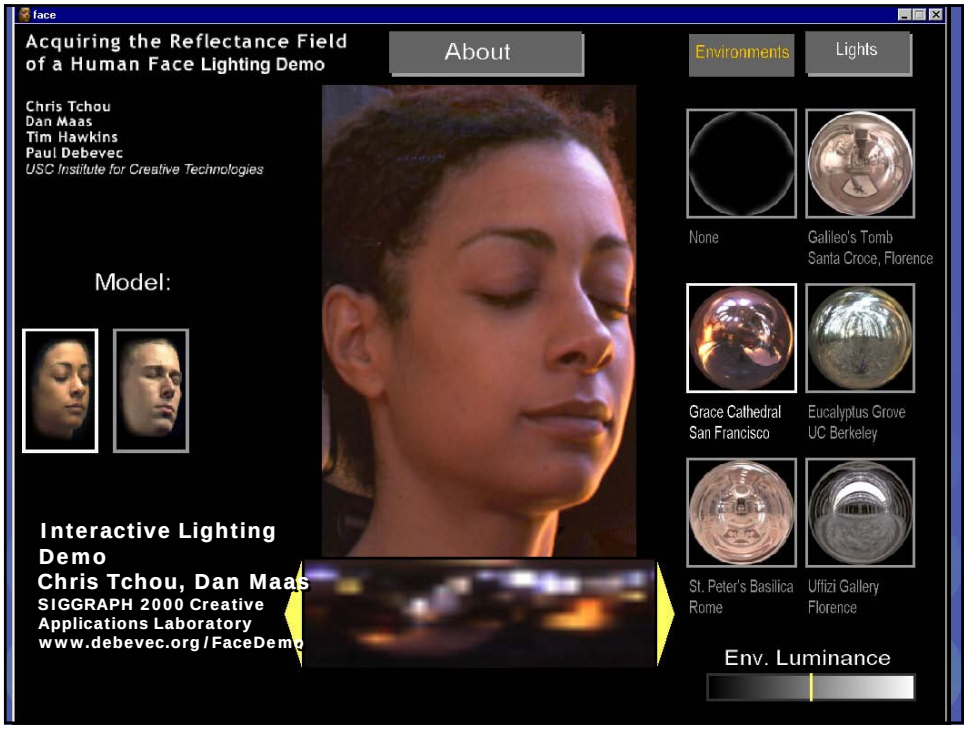

Light Stage – 4D Reflectance Field


Modulated Images
SIGGRAPH2004

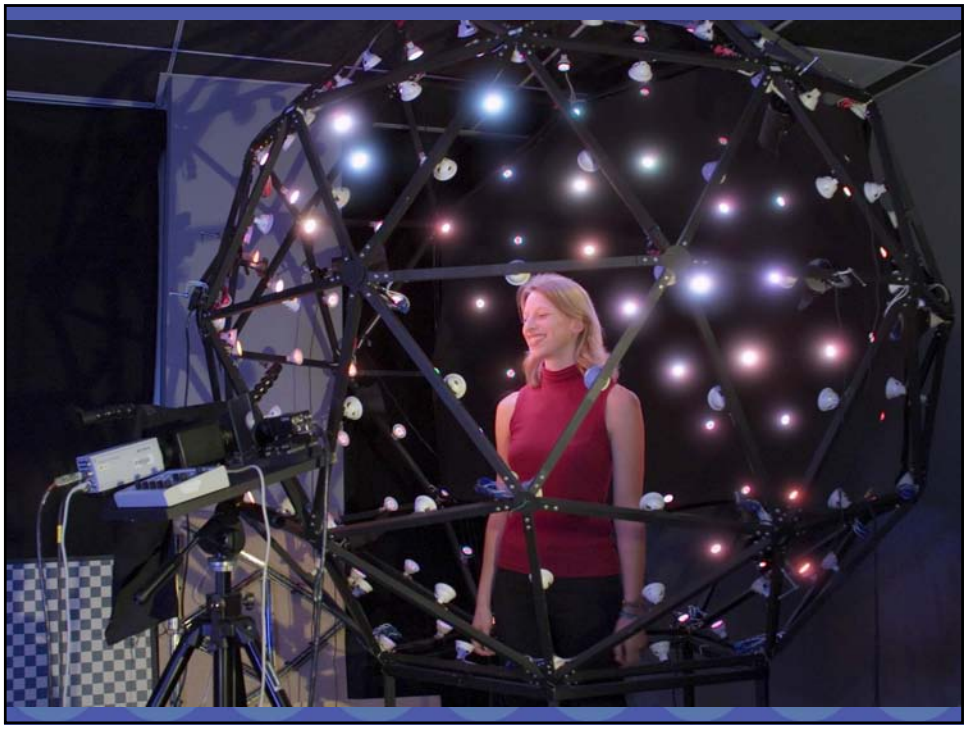

Light Stage 1 Results
SIGGRAPH2004
Environments from the Light Probe Image Gallery
www.debevec.org

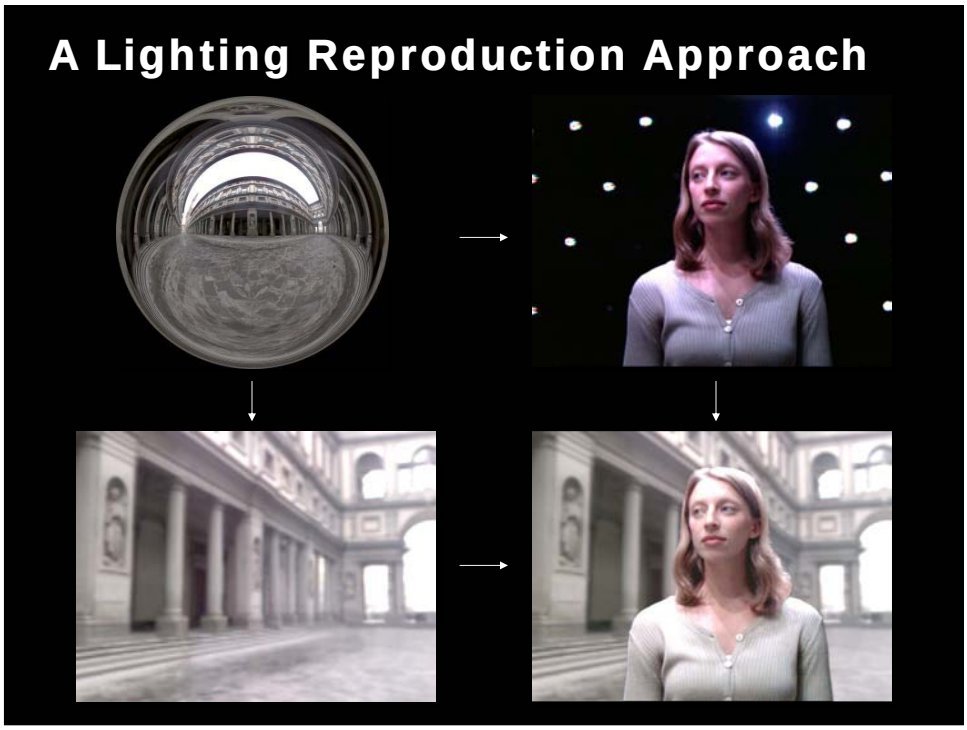

Reflectance Functions
SIGGRAPH2004


Lighting Reflectance Functions
SIGGRAPH2004
normalized light map
reflectance function
lighting product
rendered pixel
DCT Basis
Smith and Rowe. Compressed domain processing of JPEG-encoded images. 1996

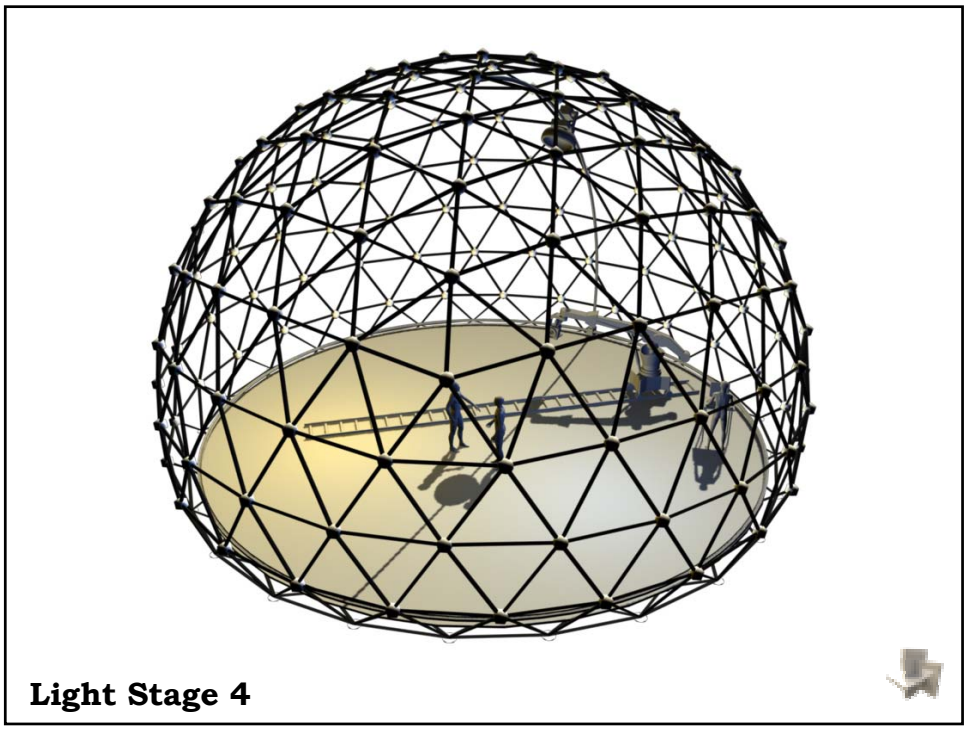

face
Acquiring the Reflectance Field of a Human Face Lighting Demo
About
Environments
Lights
Chris Tchou
Dan Maas
Tim Hawkins
Paul Debevec
USC Institute for Creative Technologies
Model:
None
Galileo's Tomb
Santa Croce, Florence
Grace Cathedral
San Francisco
Eucalyptus Grove
UC Berkeley
St. Peter's Basilica
Rome
Uffizi Gallery
Florence
Interactive Lighting Demo
Chris Tchou, Dan Maas
SIGGRAPH 2000 Creative Applications Laboratory
www.debevec.org/FaceDemo
Env. Luminance

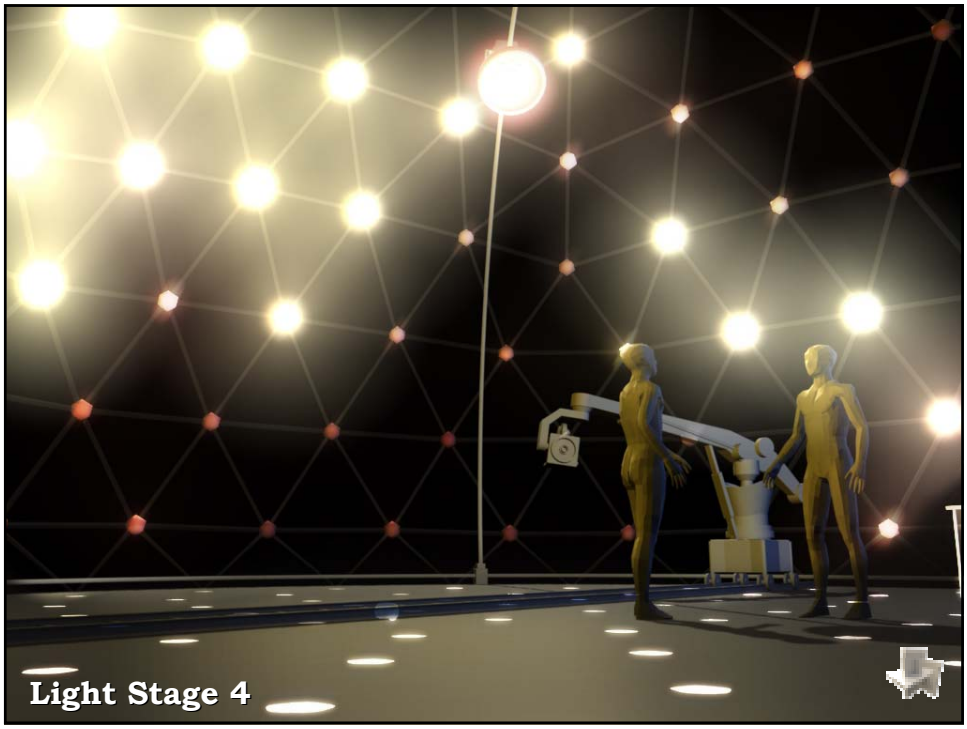

A Lighting Reproduction Approach to Live-Action Compositing
SIGGRAPH2004
Paul Debevec Andreas Wenger Chris Tchou
Andrew Gardner Jamie Waese Tim Hawkins
USC Institute for Creative Technologies




A Lighting Reproduction Approach


Composited Results

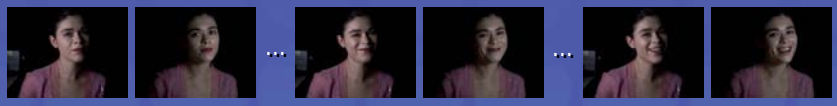

Light Stage 4



Reilluminating Live-Action in Post-Production

Quickly interleave ***n*** different lighting configurations for each frame of a performance to be shown at ***f*** frames per second using a camera recording ***nf*** frames per second.

Example: use two interlaced TV monitors to illuminate a performance alternately from the left and right at 60fps.



Then take linear combinations of the lighting configurations to produce a novel illumination of the performance at 30fps.

Light Stage 5: Relighting Live-Action in Postproduction

SIGGRAPH2004

Capture each frame of performance in several basis lighting configurations at high frame rates, then relight the moving image sequence, possibly using motion compensation.

Basis Lighting Elements =>

(Spherical harmonics, directional lighting, backlighting for mattes, etc.)

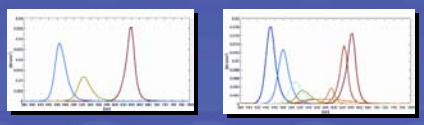
Postproduction Reillumination of a Live-Action Performance

SIGGRAPH2004

Uffizi Gallery Environment

Grace Cathedral Environment

Optimizing Multispectral Lighting Reproduction



RGB and custom 9-channel LED lights

Original | RGB Match | 9-Channel Match

Wenger, Hawkins and Debevec. *Optimizing Color Matching in a Lighting Reproduction System for Complex Subject and Illuminant Spectra.* 14th Eurographics Symposium on Rendering, June 2003.

Rendering Synthetic Objects into Real Scenes

Compositing Objects into a Scene

Paul Debevec. *Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography. SIGGRAPH 98.*

Rendering into the Scene

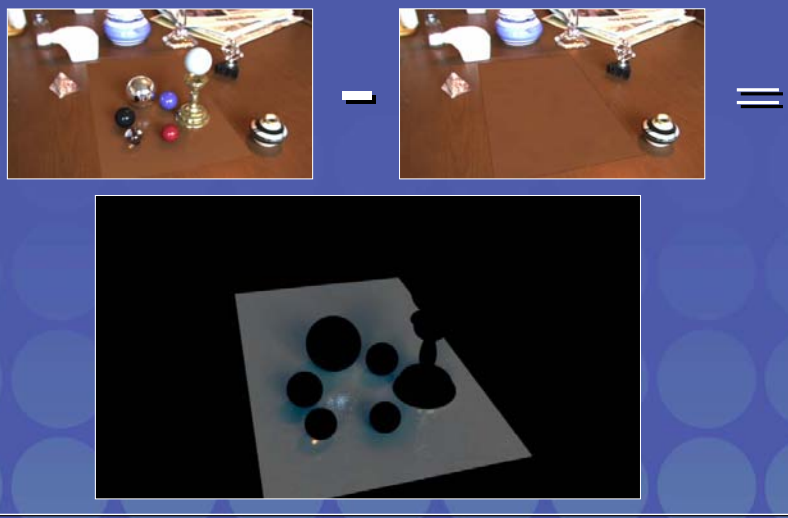
SIGGRAPH2004

Objects and Local Scene matched to Scene

Differential Rendering

Local scene w/o objects, illuminated by model

Differential Rendering 2



Differential Rendering 3

Final Result

Thanks!

Light Stage 1
Chris Tchou
Tim Hawkins
Westley Sarokin
HP Duiker
Mark Sagar

Light Stage 2
Tim Hawkins
Andrew Gardner
John Biondo
Jonathan Cohen
George Randal

Light Stage 3
Andreas Wenger
Andrew Gardner
Chris Tchou
Tim Hawkins
Maya Martinez

ILF / LLS
Jonas Unger
Andrew Gardner
Chris Tchou
Tim Hawkins

Modeling and Animation
Brian Emerson
Mark Brownlow
Westley Sarokin
HP Duiker

"Arnold" Rendering Software: Marcos Fajardo
RADIANCE Rendering Software: Greg Ward

Sponsored by Interval Research Corporation, Interactive Pictures Corporation, the US Army, TOPPAN Printing Co., Ltd., The Digital Media Innovation Program, and the University of Southern California

Plus 4-8 students and interns from USC and around the world

www.debevec.org

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 Galielo 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.

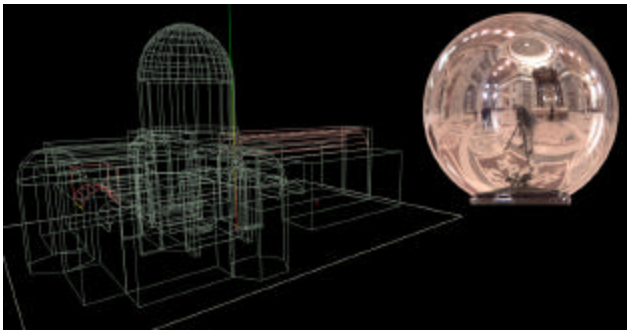
The Technology

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 photography[2], 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 measurements[3]. 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 system[1]. 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 reflections[3]. 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.

HDR Shop

Contact
Chris Tchou
USC Institute for
Creative Technologies
tchou@ict.usc.edu

Paul Debevec
USC Institute for
Creative Technologies

HDR Shop is a computer application (currently under development) designed to view and edit high-dynamic-range (HDR)[1] images: pictures that can capture a much greater range of light intensities than standard photographs or computer images. This approach is very 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, which makes it impossible to detect any difference in intensity. The standard technique to acquire HDR 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.

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, 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 (eight exponent bits and eight mantissa bits) to reduce the error below rounding error.

In addition to resampling, cropping, and mathematical operations, HDR Shop also supports transformations among 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 (LDR) copy of any image to an external image editor. Changes to the LDR image are then incorporated into the HDR image, so existing tools can be used to modify HDR images.

See also: www.debevec.org/HDRShop

References

1. Debevec, P. & Malik, J. (1997). Recovering high dynamic range radiance maps from photographs. *Proceedings of SIGGRAPH 97.*
2. Debevec, P. (1998). Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. *Proceedings of SIGGRAPH 98.*

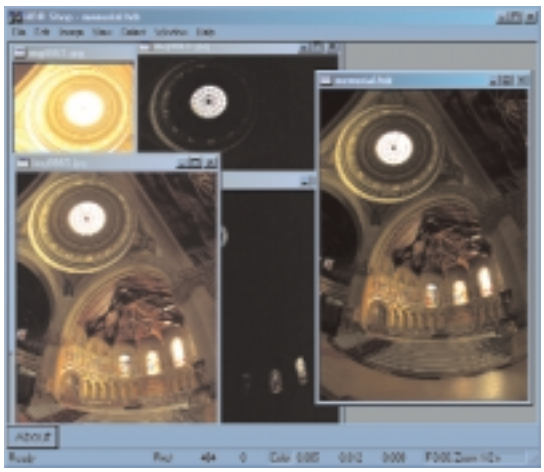


Figure 1. In HDR Shop, a sequence of low-dynamic-range images (left) can be compiled into a single high-dynamic-range image (right).

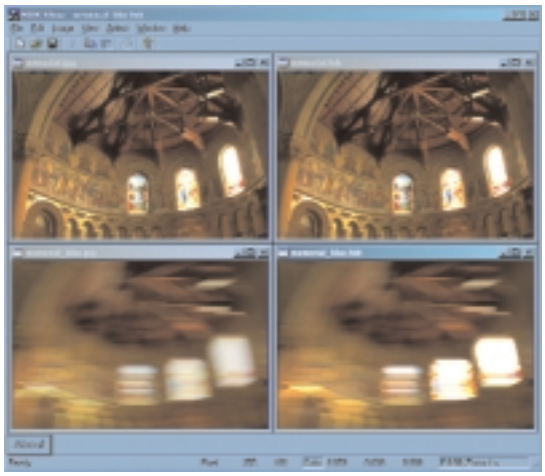


Figure 2. Comparison of HDR Shop's horizontal motion blur on a low-dynamic-range image (left) vs. a high-dynamic-range image (right).

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).

Light Stage 2.0

Contact
Tim Hawkins
Institute for Creative Technologies
University of Southern California
timh@ict.usc.edu

Jonathan Cohen
Chris Tchou
Paul Debevec
Institute for Creative Technologies
University of Southern California

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 photo-real computer graphics models that capture the unique texture and reflectance of the 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[1]. 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.

With Light Stage 2.0 (shown 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 arrayed along the length of the arm flash repeatedly in rapid sequence 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.).

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

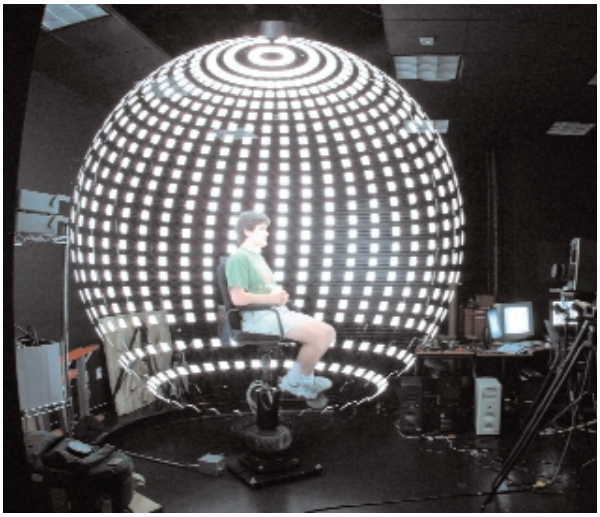
In the next few months, we will be researching new ways of analyzing the large amount of reflectance field information captured in a Light Stage 2.0 scan and adapting the datasets for use in facial animation. We would also like to make our capture process even faster, with the goal of being able to capture 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 photo-realistically into virtual environments with arbitrary lighting, where the performance can be viewed from arbitrary angles.

Another approach is to directly illuminate an actor with light sources aimed from all directions whose intensity and color is controlled by the computer. In this case, if the incident illumination necessary to realistically composite the actor into a particular scene is known in advance, the actor can be filmed directly under this illumination.

Reference

1. Debevec, P., Hawkins, T., Tchou, C., Duiker,H.-P., Sarokin, W., & Sagar, M. (2000). Acquiring the reflectance field of a human face. In *Proceedings of SIGGRAPH 2000.*

Light Stage 2.0, with seated subject.



A 10-second-exposure photograph of Light Stage 2.0 acquiring a 4D reflectance field dataset of the subject's face.

Recovering High Dynamic Range Radiance Maps from Photographs

Paul E. Debevec Jitendra Malik

University of California at Berkeley[1]

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

[1]Computer 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

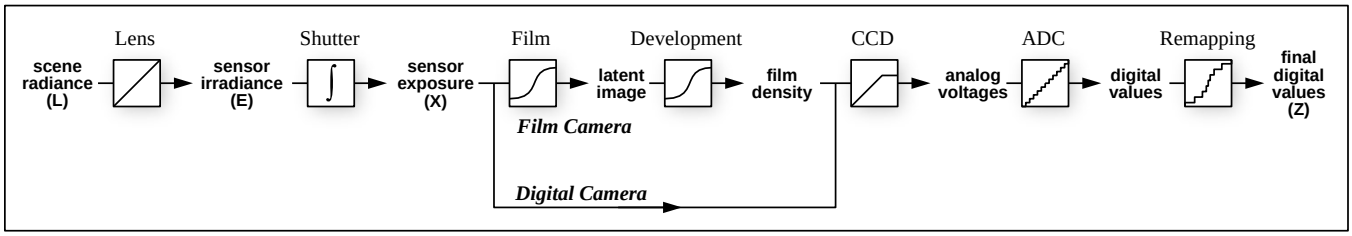


Figure 1: **Image Acquisition Pipeline** *shows how scene radiance becomes pixel values for both film and digital cameras. Unknown nonlinear mappings can occur during exposure, development, scanning, digitization, and remapping. The algorithm in this paper determines the aggregate mapping from scene 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 $\mathrm{J m}^{-2}$. 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 $\frac{1}{3}$ stop[1] 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:

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

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:

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

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:

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

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:

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

Equation 3 now becomes:

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

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:

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

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:

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

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" artifacts[5] 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 range[6] 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

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

[6] The *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*.

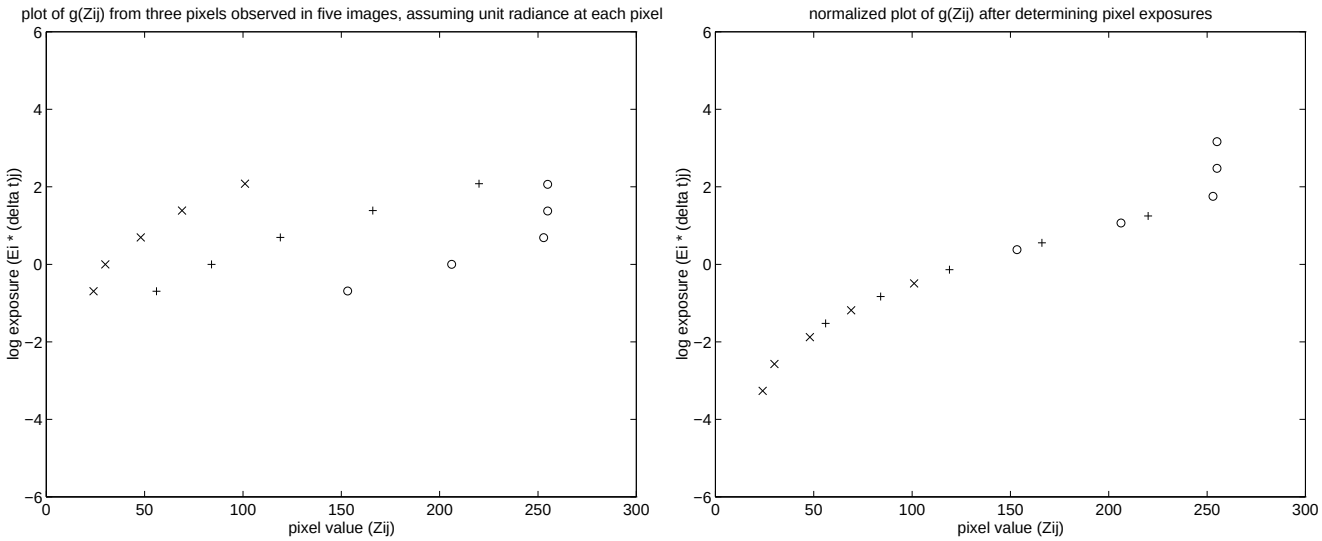


Figure 2: *In the figure on the left, the $\times$ symbols represent samples of the g curve derived from the digital values at one pixel for 5 different known exposures using Equation 2. The unknown log irradiance $\ln E_i$ has been arbitrarily assumed to be 0. Note that the shape of the g curve is correct, though its position on the vertical scale is arbitrary corresponding to the unknown $\ln E_i$. The $+$ and $\circ$ 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 commericially 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 outline[7]. 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 film[8] 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

[7]This work was done in collaboration with Gregory Ward Larson

[8]Conveniently, 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 fisheye 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

[9] Note 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 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.*

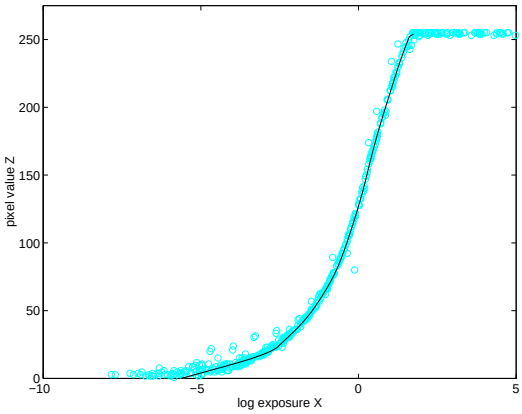


Figure 4: *The response function of the DCS460 recovered by our algorithm, with the underlying $(E_i \Delta t_j, Z_{ij})$ data shown as light circles. The logarithm is base e.*

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.*

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.*

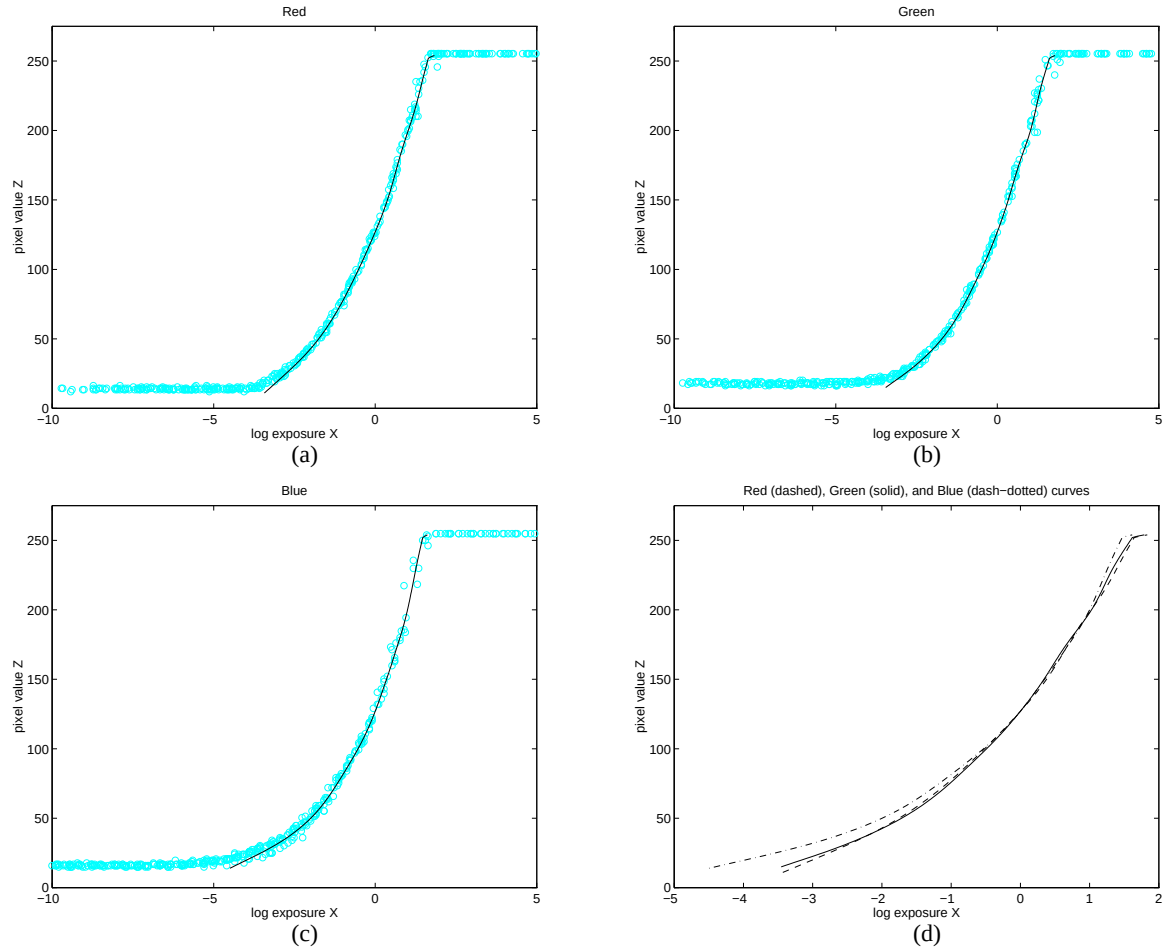
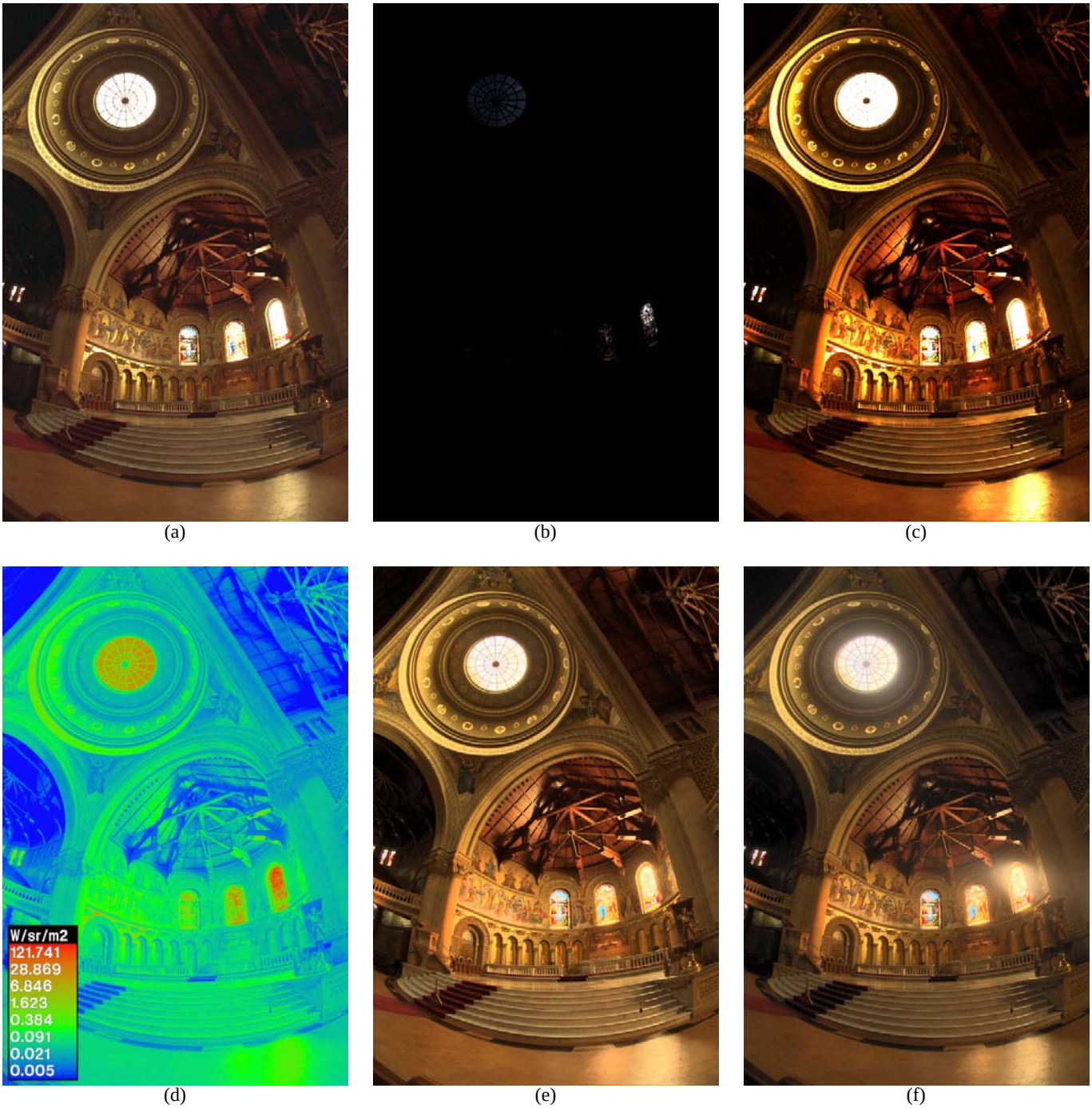


Figure 7: *Recovered response curves for the imaging system used in the church photographs in Fig. 8.* **(a-c)** *Response functions for the red, green, and blue channels, plotted with the underlying* $(E_i \Delta t_j, Z_{ij})$ *data shown as light circles.* **(d)** *The response functions for red, green, 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.*



(a) (b) (c)

(d) (e) (f)

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 images[10] 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.

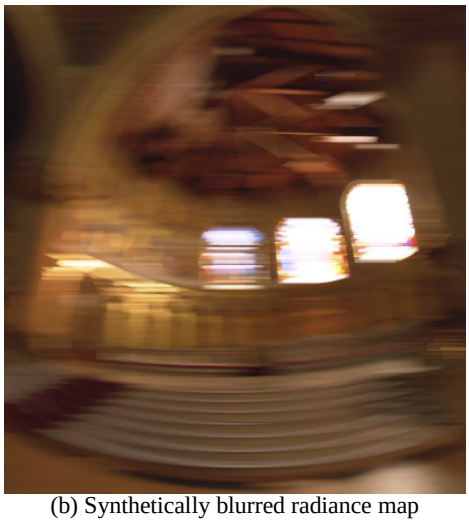
In Fig. 9(b), the extremely high pixel values were represented properly in the radiance map and thus remained at values above the level of the response function's saturation point within most of the blurred region. As a result, the resulting virtual photograph exhibits several crisply-defined saturated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

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

[11] This 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.

(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 rephotographed 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).*

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., PATTANAIK, 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
%  B(j)   is the log delta t, or log shutter speed, for image j
%  l      is lamdba, 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
%  lE(i)  is the log film irradiance at pixel location i
%

function [g,lE]=gsolve(Z,B,l,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+i) = -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)=l*w(i+1);        A(k,i+1)=-2*l*w(i+1);  A(k,i+2)=l*w(i+1);
  k=k+1;
end

%% Solve the system using SVD

x = A\b;

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

To appear in the SIGGRAPH 98 conference proceedings

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 Berkeley[1]

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*.

[1]Computer 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.

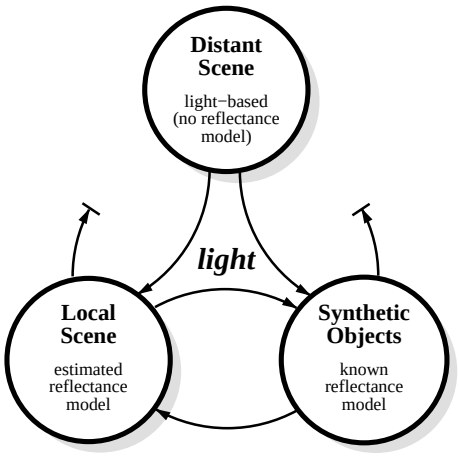


Figure 1: **The General Method** *In our method for adding synthetic objects into light-based scenes, the scene is partitioned into three components: the distant scene, the local scene, and the synthetic objects. Global illumination is used to simulate the interplay of light amongst all three components, 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 extremeley 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 space[1]. 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 radiance[2], 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 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-

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

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

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 environment[3]. 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 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 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

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

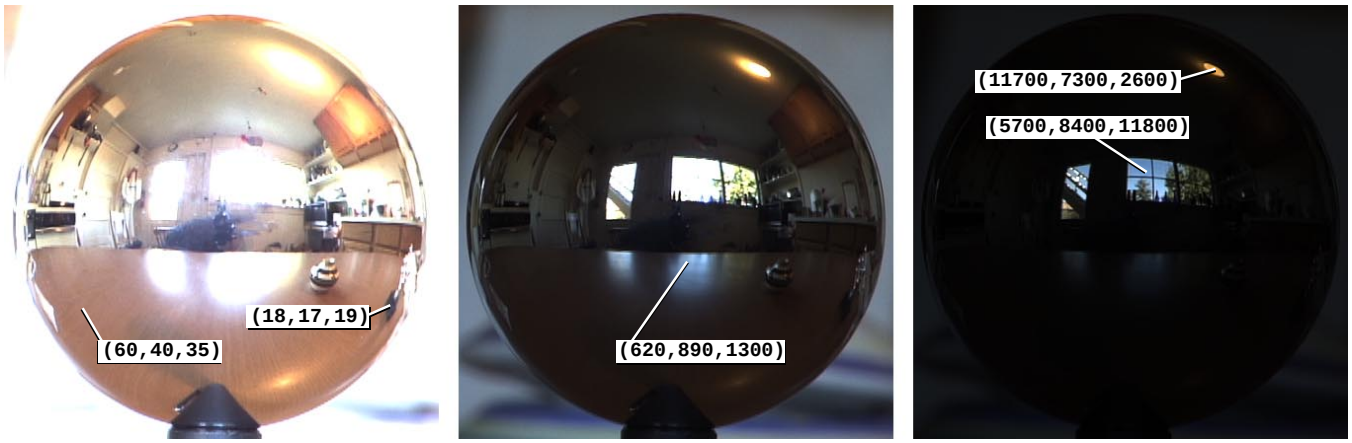


Figure 2: **An omnidirectional radiance map** This full dynamic range lighting environment was acquired by photographing a mirrored ball balanced on the cap of a pen sitting on a table. The environment contains natural, electric, and indirect light. The three views of this image adjusted to (a) +0 stops, (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.

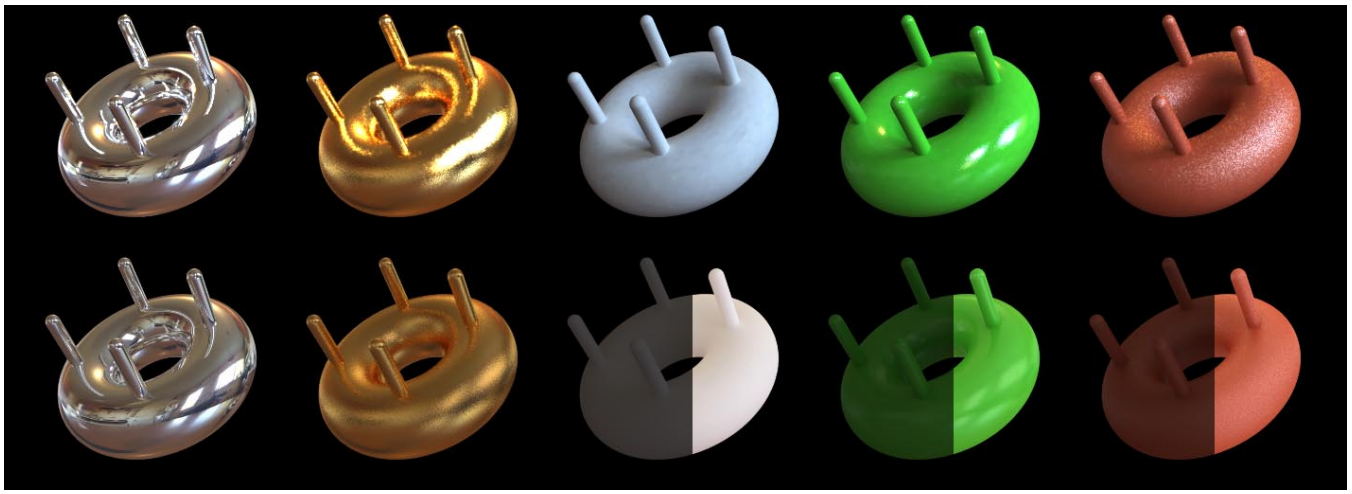


Figure 3: **Illuminating synthetic objects with real light (Top row: a,b,c,d,e)** *With full dynamic range measurements of scene radiance from Fig. 2.* **(Bottom row: f,g,h,i,j)** *With low dynamic range information from a single photograph of the ball. The right sides of images (h,i,j) have been brightened 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.*

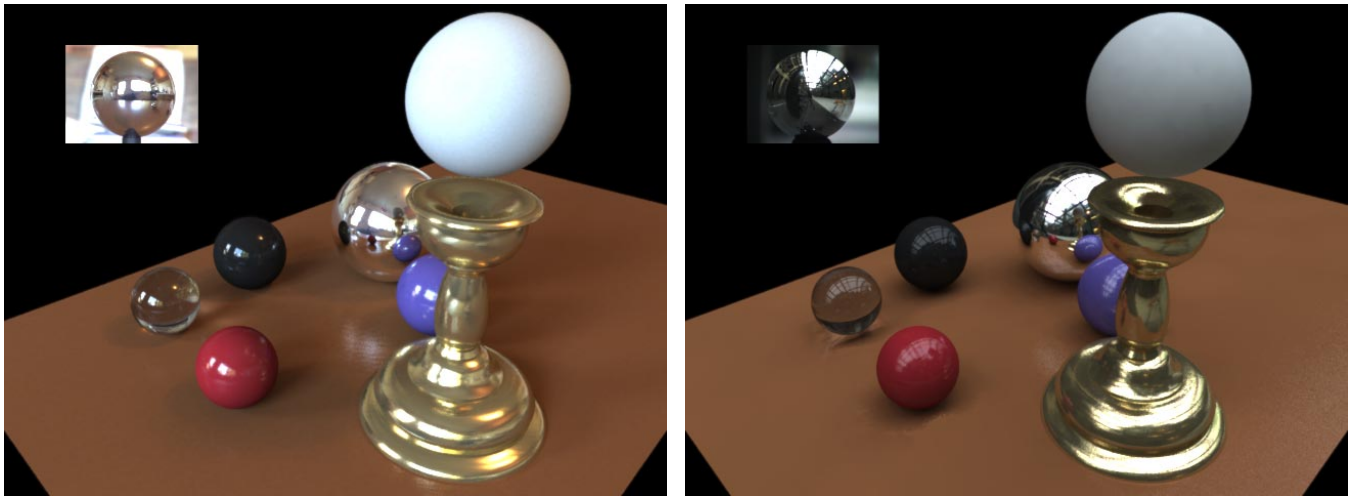


Figure 4: **Synthetic objects lit by two different environments (a)** *A collection of objects is illuminated by the radiance information in 2. The objects exhibit appropriate interreflection.* **(b)** *The same objects are illuminated by different radiance information obtained in an outdoor urban environment 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 object[4]. 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.

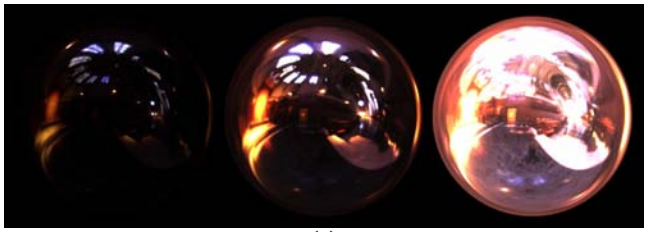
5.1 Mapping from the probe to the scene model

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

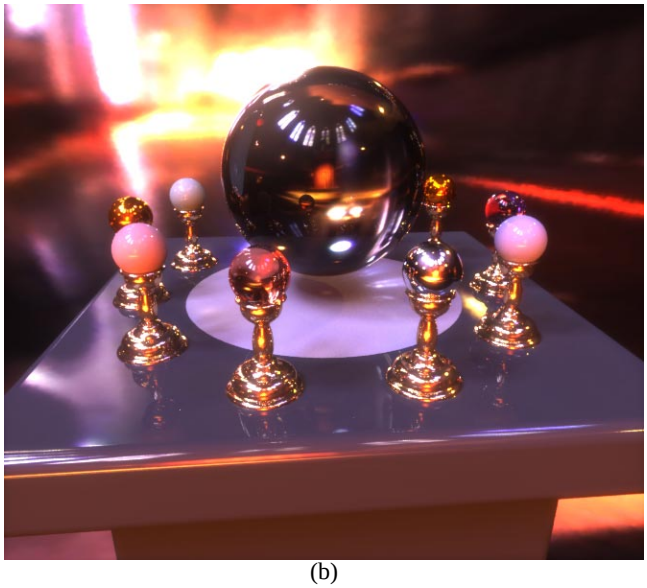
[4]Parabolic mirrors combined with telecentric lenses [34] can be used to obtain hemispherical fields of view with a consistent principal point, if so desired.

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 enviroment 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.



(a)



(b)

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_{noobj} 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: $Err_{ls} = LS_{noobj} - LS_b$. 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:

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

In this form, we see that whenever LS_{obj} and LS_{noobj} 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_{noobj}, light is subtracted from the background to form shadows,

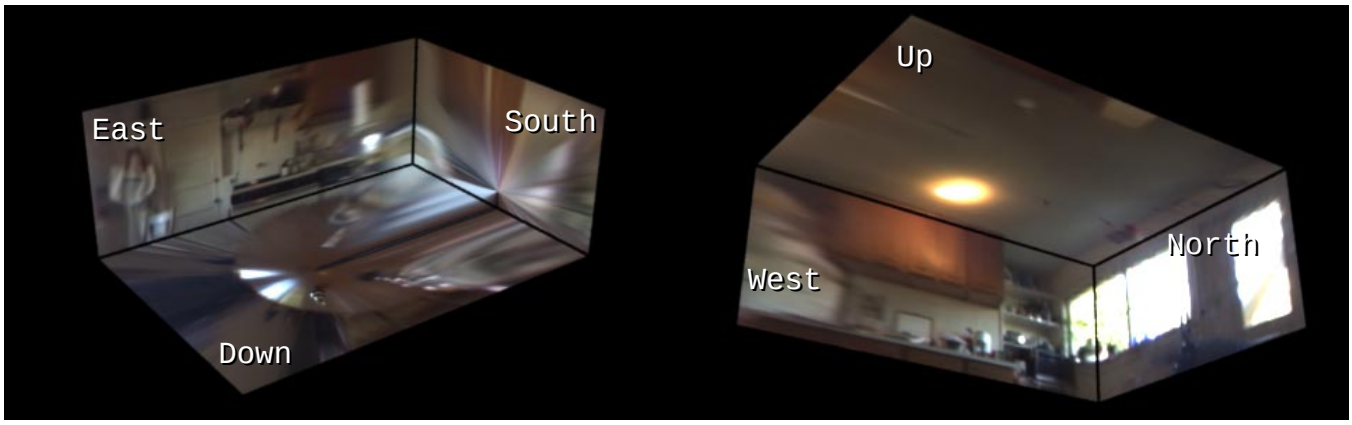


Figure 5: **A Light-Based Model** *A simple light-based model of a room is constructed by mapping the image from a light probe onto a box. The box corresponds to the upper half of the room, with the bottom face of the box being coincident with the top of the table. The model contains the full dynamic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

and when LS_{obj} is lighter than LS_{noobj} 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 abitrarily 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: $LS_{final} = LS_b(LS_{obj}/LS_{noobj})$. Inaccuracies in the local scene BDRF 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:

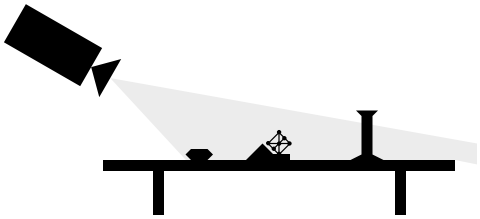
$$L_{r2}(\theta_r, \phi_r) = \int_0^{2\pi} \int_0^{\pi/2} L_i(\theta_i, \phi_i) \cos\theta_i \sin\theta_i \, d\theta_i \, d\phi_i$$

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

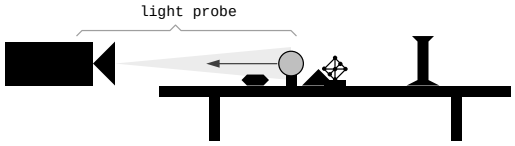
$$\rho_d' = \frac{L_{r1}(\theta_r, \phi_r)}{L_{r2}(\theta_r, \phi_r)}$$

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.

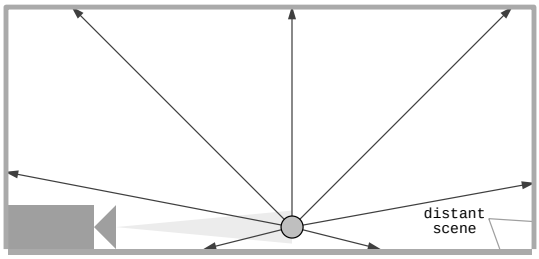
For a trichromatic image, the red, green, and blue diffuse reflectance values are computed independently. The diffuse characteristics of the background material used to produce Fig. 8(c) were



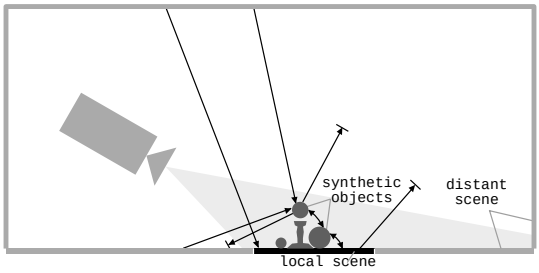
(a) Acquiring the background photograph



(b) Using the light probe



(c) Constructing the light-based model



(d) Computing the global illumination solution

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.

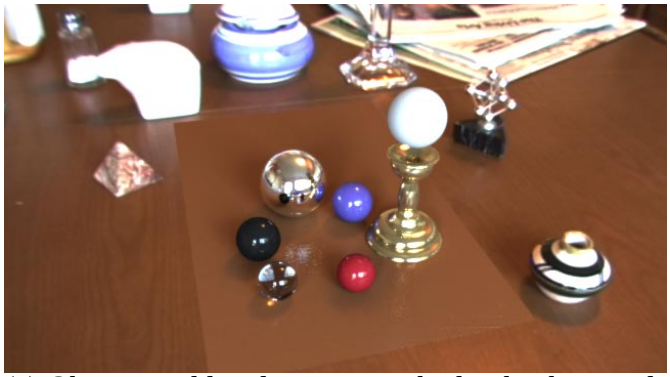
Fig. 8(f) shows the difference between the appearance of the local scene rendered with (8(c)) and without (8(d)) the objects. For illustration purposes, the difference in radiance values have been offset so that zero difference is shown in gray. The objects have been masked out using image 8(e). This 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.

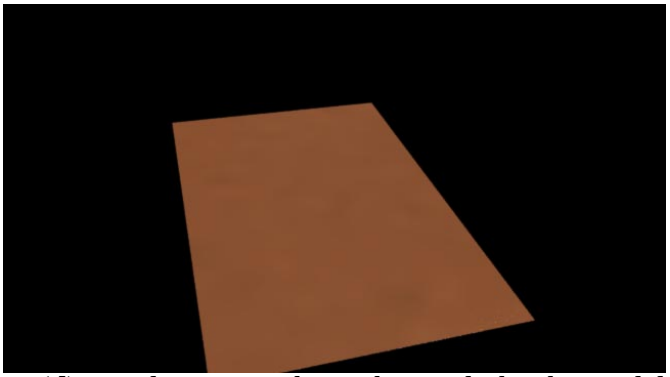
In this final rendering, the synthetic objects exhibit a consistent appearance with the real objects present in the background image 8(a) in both their diffuse and specular shading, as well as the direction and coloration of their shadows. The somewhat speckled nature of the object reflections seen in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

(a) Background photograph

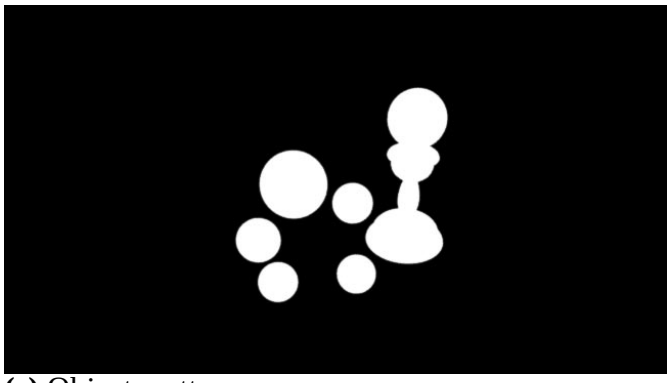
(b) Camera calibration grid and light probe



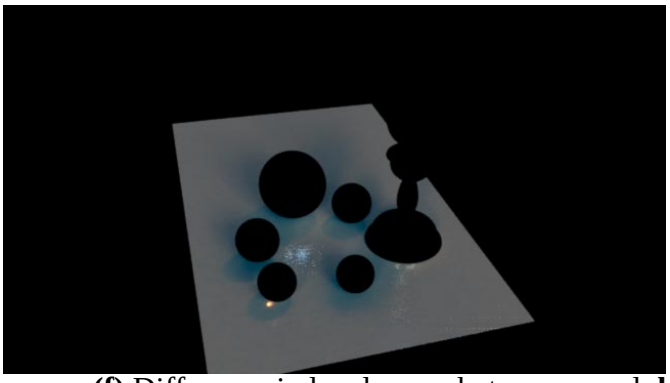
(c) Objects and local scene matched to background



(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 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 ilumination 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] CURLESS, 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. Zippered 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 Berkeley[1] [‡]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, rerendering 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-

[1]Computer 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*:

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

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 coindicent 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).

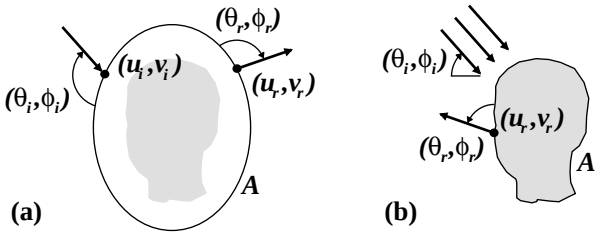


Figure 1: **The Reflectance Field** *(a) describes how a volume of space enclosed by a surface A transforms an incident field of illumination $R_i(u_i, v_i, \theta_i, \phi_i)$ into a radiant field of illumination $R_r(u_r, v_r, \theta_r, \phi_r)$. In this paper, we acquire a non-local reflectance field (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.*

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.*

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 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

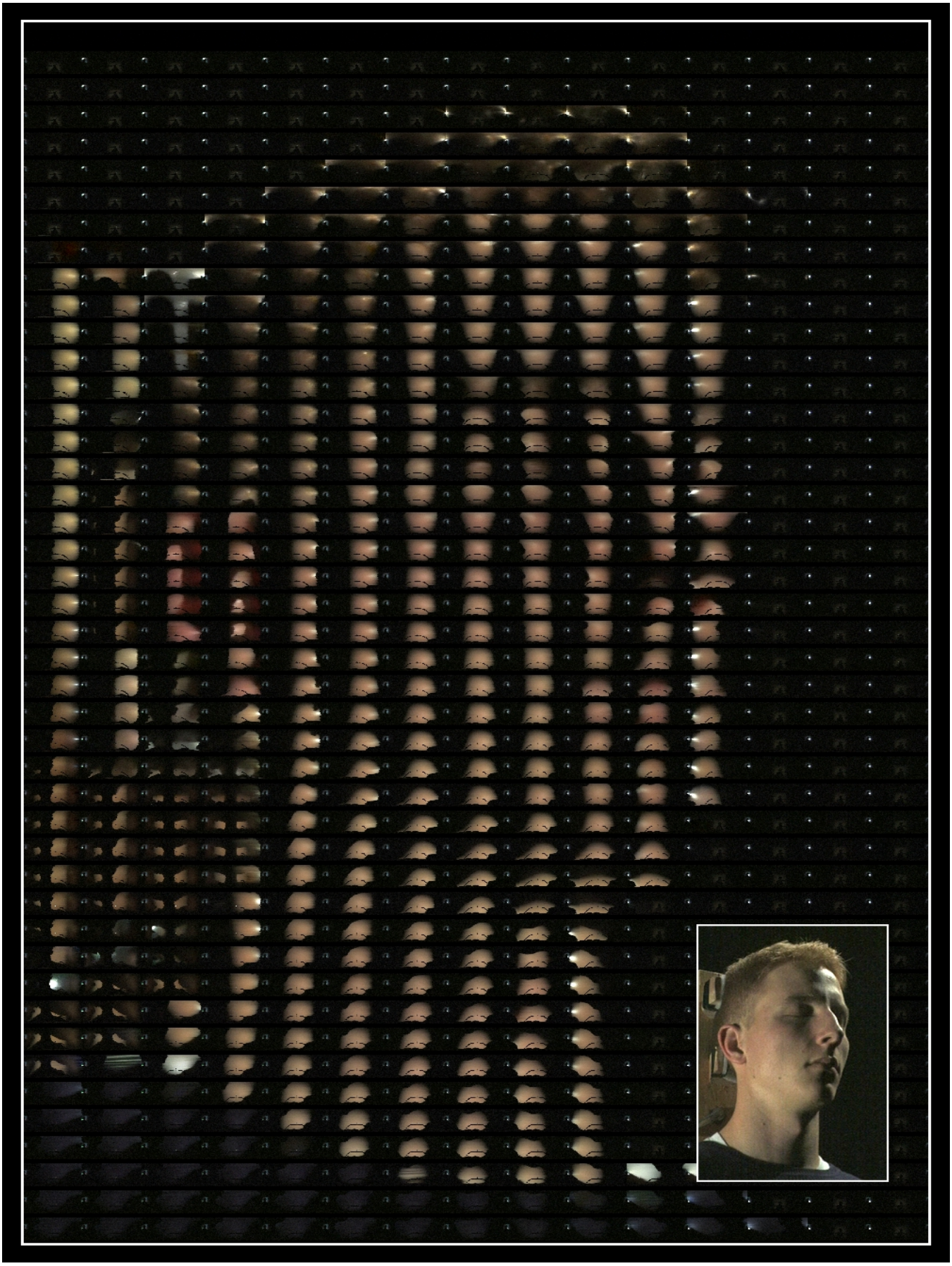


Figure 3: **Reflectance Functions for a Face** *This mosaic is formed from the reflectance functions of a* 15×44 *sampling of pixels from the original* 480×720 *image data. Each* 64×32 *reflectance function consists of the corresponding pixel location's appearance under two thousand lighting directions 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:

$$R_{xy}(\theta, \phi) = L_{\theta,\phi}(x, y) \tag{2}$$

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

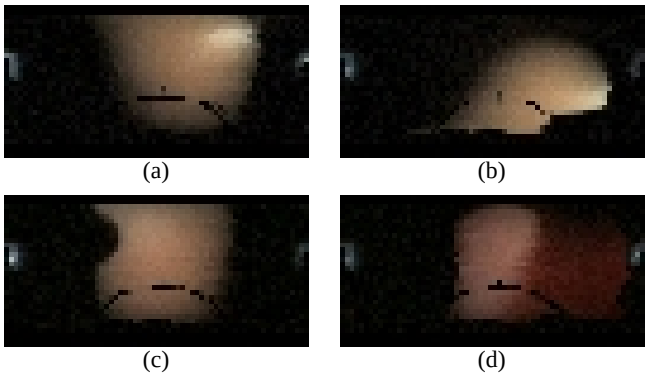


Figure 4: **A Sampling of Facial Reflectance Functions** *The above reflectance functions appear in the mosaic of Fig. 3. The middle of each function corresponds to the pixel being illuminated from the direction of the camera; as one moves within the reflectance function the light direction moves in 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 obscurs 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.*

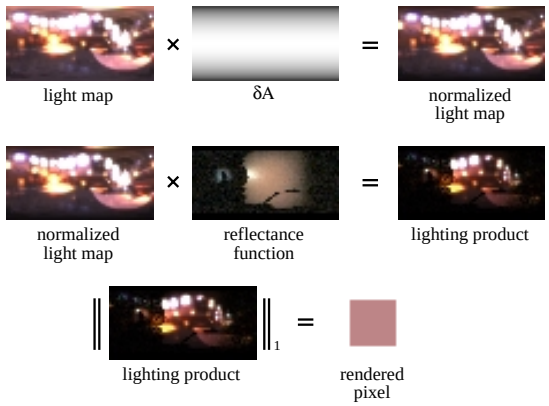
Suppose that we wish to generate an image of the face in a novel form of illumination. Since each $R_{xy}(\theta, \phi)$ represents how much light is reflected toward the camera by pixel (x, y) as a result of illumination from direction (θ, ϕ), and since light is additive, we can compute an image of the face $\hat{L}(x, y)$ under any combination of the original light sources $L_i(\theta, \phi)$ as follows:

$$\hat{L}(x, y) = \sum_{\theta,\phi} R_{xy}(\theta, \phi) L_i(\theta, \phi) \tag{3}$$

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:

$$\hat{L}(x, y) = \sum_{\theta,\phi} R_{xy}(\theta, \phi) L_i(\theta, \phi) \delta A(\theta, \phi) \tag{4}$$

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.



(5)

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, translucence, 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.

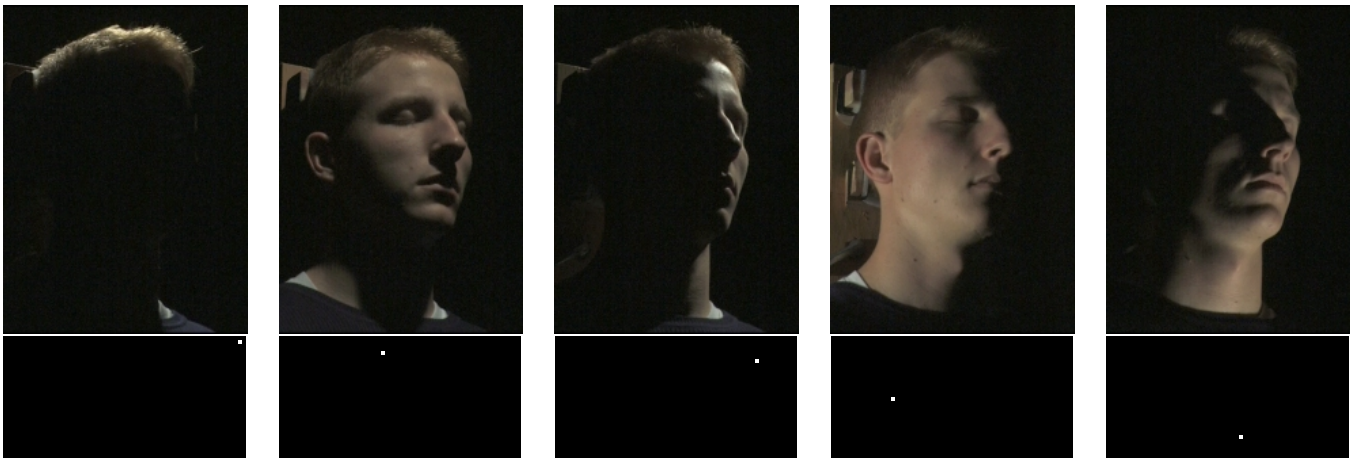


Figure 5: **Light Stage Images** *Above are five of the 2048 images taken by one camera during a run of the light stage. The pixel values of each location on the face under the 2048 illumination directions are combined to produce the mosaic images in Fig. 3. Below each image is the impulse light map that would generate it.*

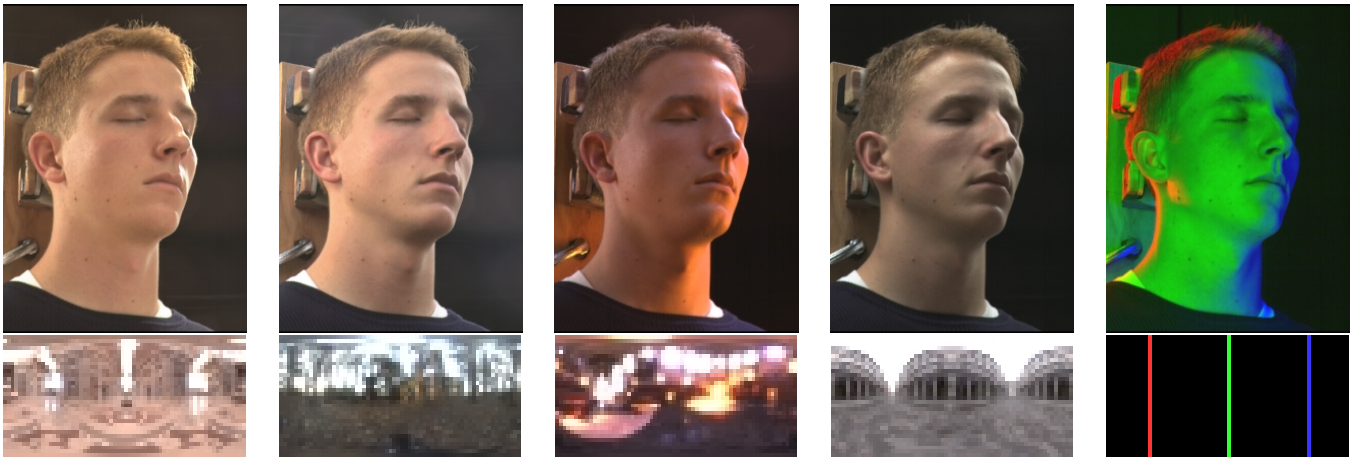


Figure 6: **Face Rendered under Sampled Illumination** *Each of the above images shows the face synthetically illuminated with novel lighting, with the corresponding light map shown below. Each image is created by taking the dot product of each pixel's reflectance function with the light map. The first 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.

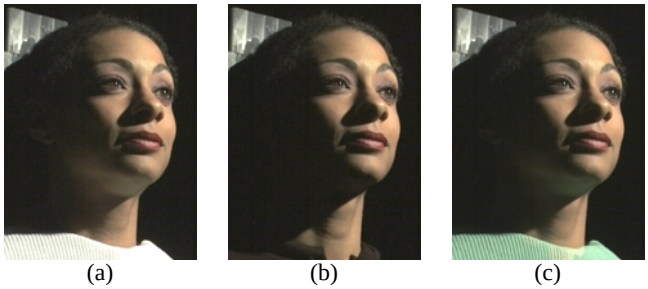
By recording the light stage images in high dynamic range [9] and using the process of environment matting [42], we can apply this technique to translucent and refractive objects and reproduce the appearance of the environment in the background; this process is described in the Appendix.

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.

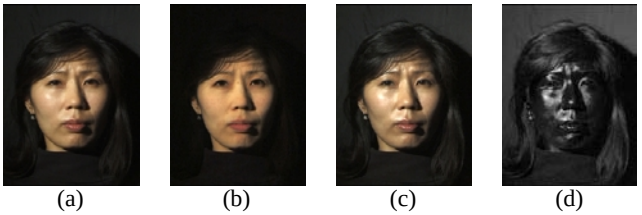
To accomplish this, we make use of a skin reflectance model which we introduce in Section 4.1. This model is used to guide the shifting and scaling of measured reflectance function values as the viewpoint changes. As such, our technique guarantees that the resynthesized reflectance function will agree 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.



(a) (b) (c)

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.*



(a) (b) (c) (d)

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.*

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 camera[1]. Figure 9 shows separated specular and diffuse reflection components of a face using this technique.

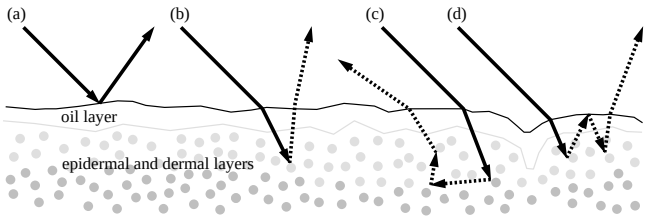


Figure 8: **Skin Reflectance** *Light reflecting from skin must have reflected specularly off the surface (a) or at some point entered one or more of the scattering layers (b, c, d). If the incident light is polarized, the specularly reflected light will maintain this polarization; however, light which 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 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.

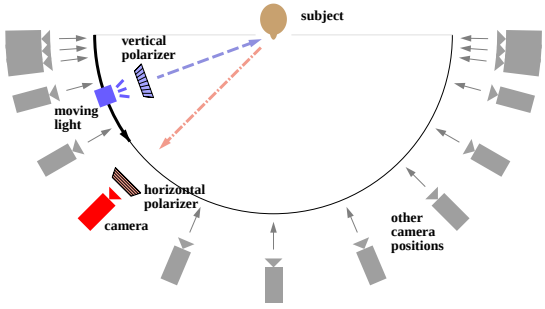


Figure 10: **Reflectometry Experiment** *In this experiment, the diffuse and specular reflectance of an area of skin on the subject's forehead was recorded from sixty-four illumination directions for each of fifteen camera positions. Polarizers on the light and camera were used to separate the reflection 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

[1] In 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.

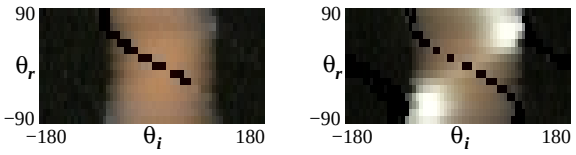


Figure 11: **Reflectometry Results** *The left image shows the measured diffuse (sub-surface) component of the skin patch obtained from the experiment in Fig. 10 for incident illumination angle θ_i and viewing direction θ_r. θ_r is nonuniformly spaced at angles of $\pm(0, 22.5, 45, 60, 75, 82.5, 86.25, 89)$ 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:

$$normalize(\vec{d_0} + f(\theta_i, \theta_r)(\vec{d_0} - \vec{s})) \quad (6)$$

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 $Rxy(\theta, \phi)$, 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 $\vec{e}$:

$$R = \mu_d \vec{d} + \mu_s \vec{s} + \mu_e \vec{e}$$

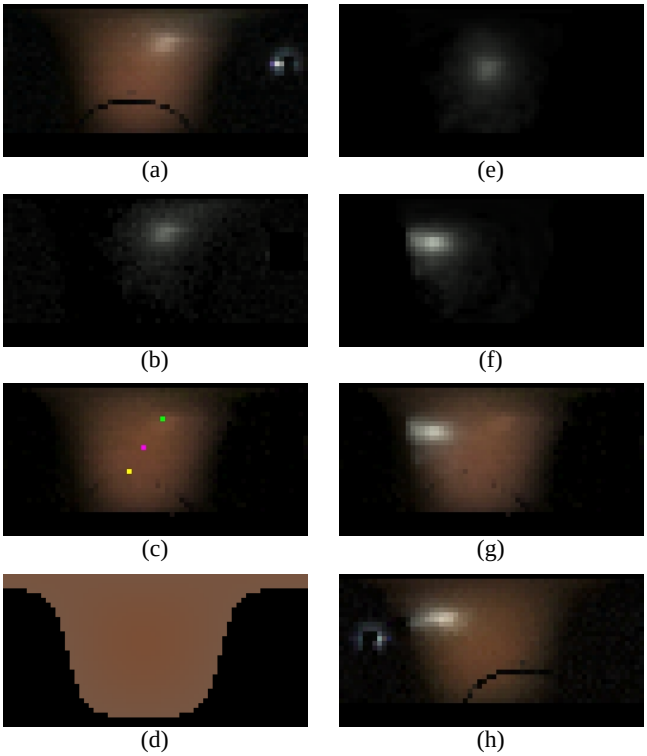


Figure 12: **Analyzing and Resynthesizing Reflectance Functions** *Reflectance functions (a) can be decomposed into specular (b) and diffuse (c) components using colorspace analysis based on a model of the variation in diffuse chromaticity (d). We compute a surface normal $\vec{n}$ based on the diffuse 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 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:

$$\vec{H} = normalize(\vec{v}_n + \vec{l}_p)$$

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:

$$L_{\vec{v}} = \int \frac{PL_{\omega_i}GF}{4\cos\theta_r} d\omega_i \tag{8}$$

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 $\vec{l}_q$:

$$L_{\vec{v}_n} = L_{\vec{v}_0} \frac{G(\vec{v}_n, \vec{l}_p, \vec{n})F(\vec{v}_n, \vec{l}_p)\ (\vec{v}_0 \cdot \vec{n})}{G(\vec{v}_0, \vec{l}_q, \vec{n})F(\vec{v}_0, \vec{l}_q)\ (\vec{v}_n \cdot \vec{n})} \tag{9}$$

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 resynthesized 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.

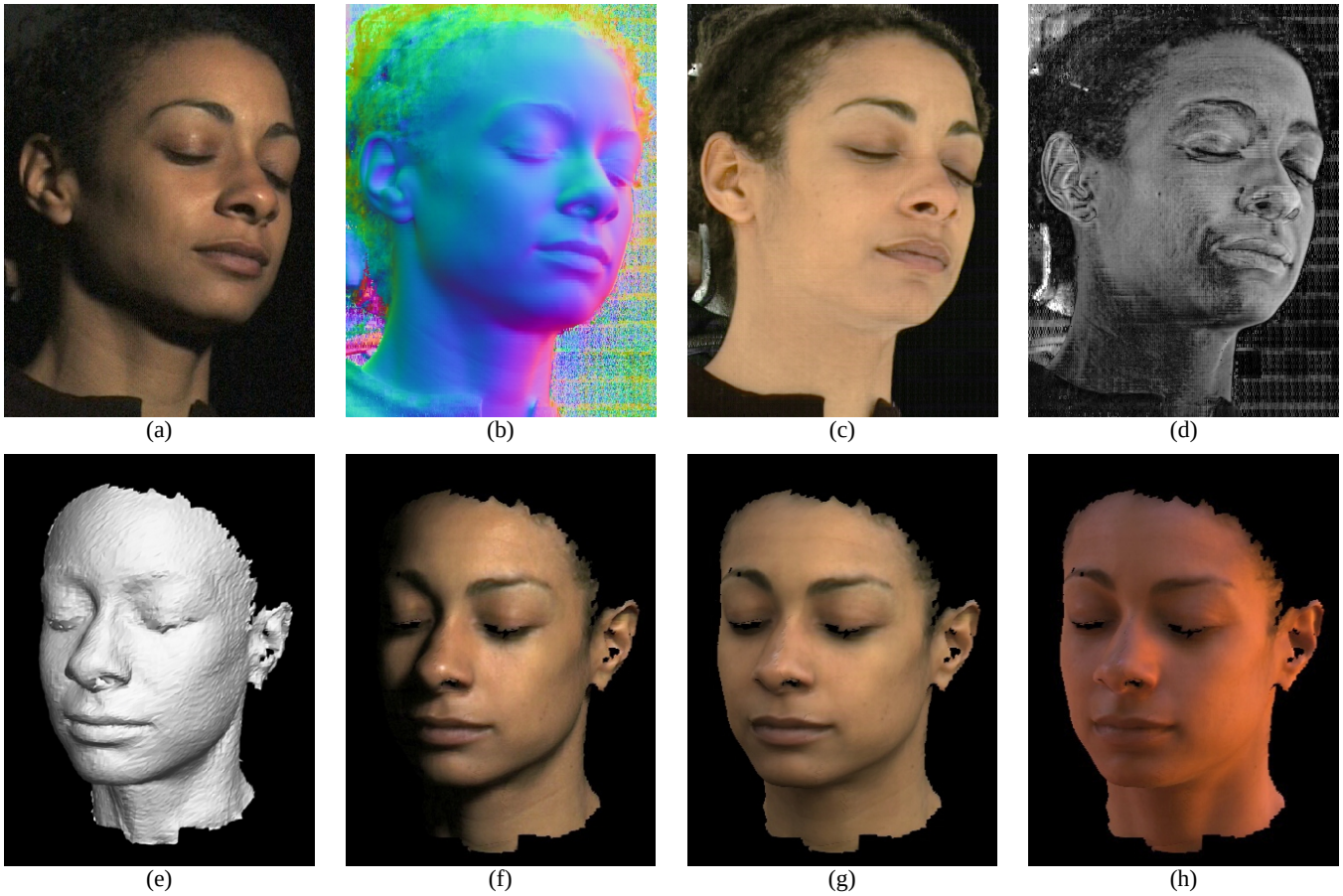


Figure 14: **Analyzing Reflectance and Changing the Viewpoint** **(a)** An original light stage image taken by the left camera. **(b)** Recovered surface normals n_d derived from the fitted diffuse reflectance lobe for each pixel; the RGB value for each pixel encodes the X, Y, and Z direction of each 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.

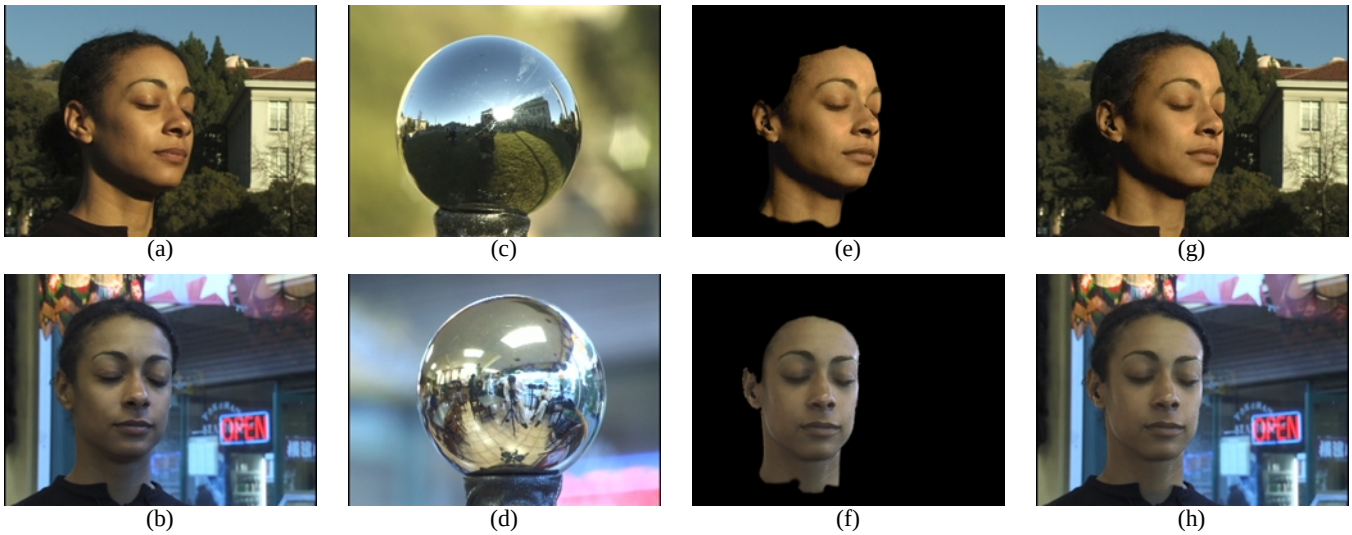


Figure 15: **Matching to Real-World Illumination** **(a,b)** Actual photographs of the subject in two different environments. **(c,d)** Images of a light probe placed in the position of the subject's head in the same environments. **(e,f)** Synthetic renderings of the face matched to the photographed 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 orginal 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.

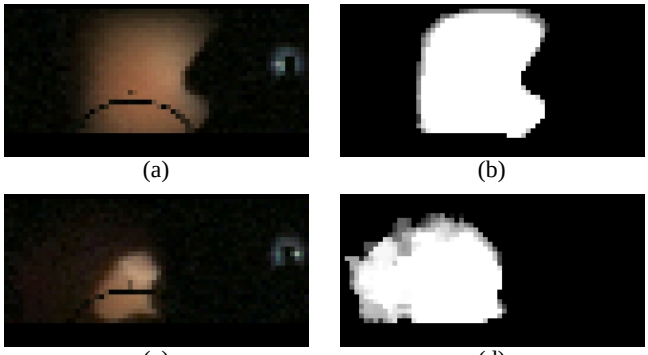


Figure 13: **Reflectance Function Shadow Maps** *The reflectance function of a point near the nose (a) and the corresponding shadow map (b) computed using brightness thresholding. (c) shows a point in the ear which receives strong indirect illumination, causing the non-shadowed region in (d) to be overestimated. 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 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] HAEBERLI, 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 decompression 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 rerendering 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.

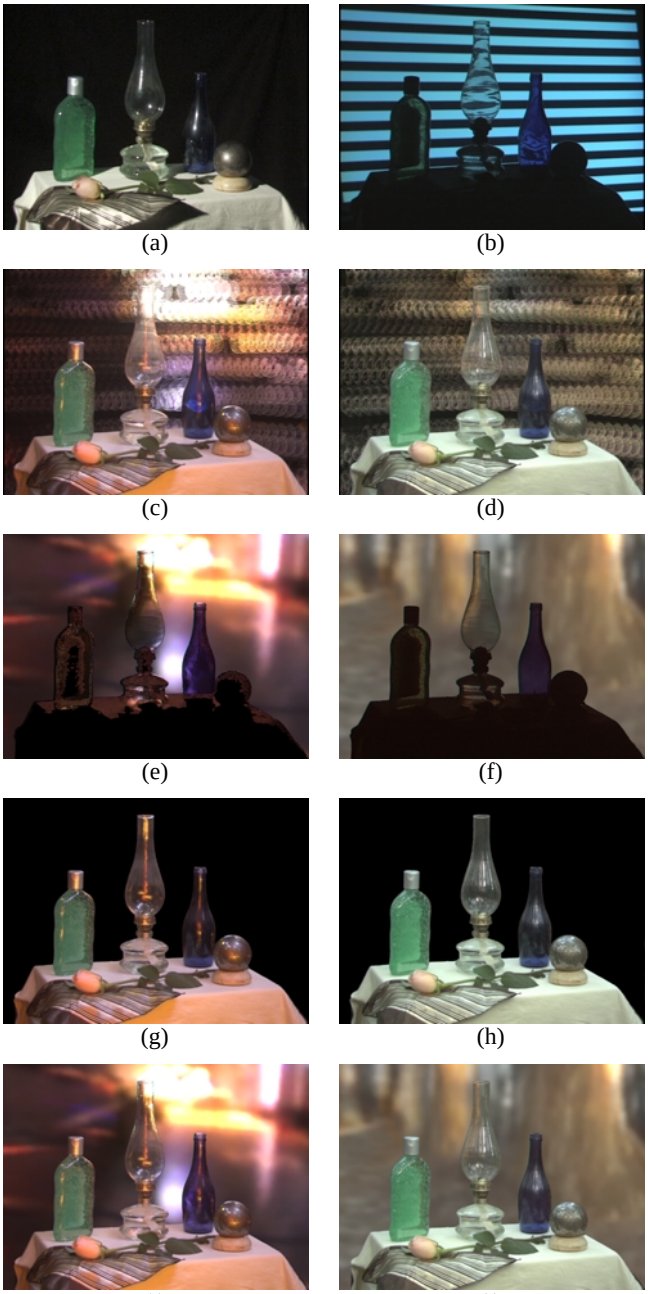
[38] WILLIAMS, L. Performance-driven facial animation. *Computer Graphics (Proceedings of SIGGRAPH 90) 24*, 4 (August 1990), 235–242.

[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.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

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).

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 daylit 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 imperceivable 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, the we use the formula given in Eq. 1a. To compute real luminance from L_e, we use the inverse formula given in Eq. 1b.

$$L_e = \left\lfloor 64\left(\log_2 L + 12\right)\right\rfloor \qquad (1a)$$

$$L = \exp_2\left[\left(L_e + 0.5\right)/64 - 12\right] \qquad (1b)$$

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.

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

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

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.

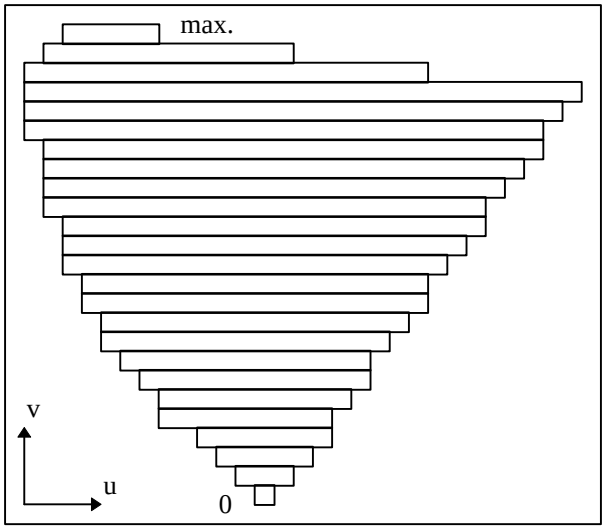


Figure 2. Scanline traversal of (u,v) coordinate space for 14-bit chromaticity encoding.

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.)

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

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

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.

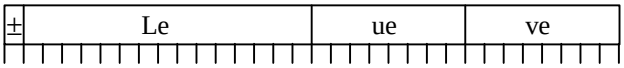


Figure 3. Bit allocation for 32-bit pixel encoding. MSB is a sign bit, and the next 15 bits are used for a log luminance encoding. The uv coordinates are separate 8-bit quantities.

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.

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

$$L = \exp_2\left[(L_e + 0.5)/256 - 64\right] \qquad (4b)$$

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.

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

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

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

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

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/decompresson 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 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.

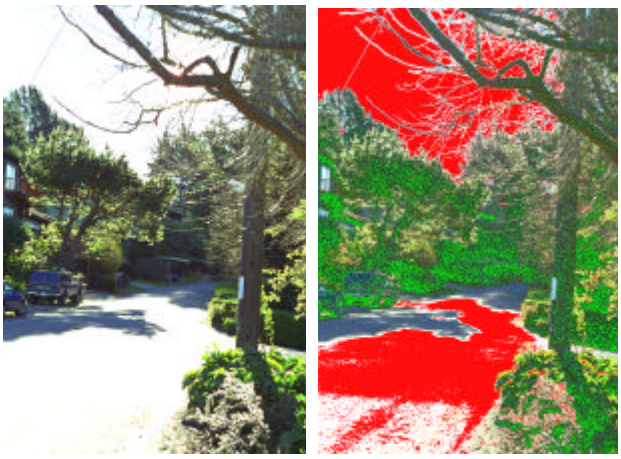


Figure 4. The left image (a) shows a PhotoYCC encoding of a color photograph tone-mapped with a linear operator. The right image (b) shows the out-of-gamut regions. Red areas are too bright or too dim, and green areas have inaccurate color.

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*.

This paper appeared in IEEE Transactions on Visualization and
Computer Graphics, Vol. 3, No. 4, December 1997.

A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes

Gregory Ward Larson†
Building Technologies Program
Environmental Energy Technologies Division
Ernest Orlando Lawrence Berkeley National Laboratory
University of California
1 Cyclotron Road
Berkeley, California 94720

Holly Rushmeier
IBM T.J. Watson Research Center

Christine Piatko††
National Institute for Standards and Technology

January 15, 1997

This paper is available electronically at:
http://radsite.lbl.gov/radiance/papers

† Author's current address: Silicon Graphics, Inc., Mountain View, CA.
†† Author's current address: JHU/APL, Laurel, MD.

LBNL 39882
UC 400

A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes

Gregory Ward Larson
Lawrence Berkeley National Laboratory

Holly Rushmeier
IBM T.J. Watson Research Center

Christine Piatko
National Institute for Standards and Technology

ABSTRACT

We present a tone reproduction operator that preserves visibility in high dynamic range scenes. Our method introduces a new histogram adjustment technique, based on the population of local adaptation luminances in a scene. To match subjective viewing experience, the method incorporates models for human contrast sensitivity, glare, spatial acuity and color sensitivity. We compare our results to previous work and present examples of our techniques applied to lighting simulation and electronic photography.

Keywords: Shading, Image Manipulation.

1 Introduction

The real world exhibits a wide range of luminance values. The human visual system is capable of perceiving scenes spanning 5 orders of magnitude, and adapting more gradually to over 9 orders of magnitude. Advanced techniques for producing synthetic images, such as radiosity and Monte Carlo ray tracing, compute the map of luminances that would reach an observer of a real scene. The media used to display these results -- either a video display or a print on paper -- cannot reproduce the computed luminances, or span more than a few orders of magnitude. However, the success of realistic image synthesis has shown that it is possible to produce images that convey the appearance of the simulated scene by mapping to a set of luminances that can be produced by the display medium. This is fundamentally possible because the human eye is sensitive to relative rather than absolute luminance values. However, a robust algorithm for converting real world luminances to display luminances has yet to be developed.

The conversion from real world to display luminances is known as *tone mapping*. Tone mapping ideas were originally developed for photography. In photography or video, chemistry or electronics, together with a human actively controlling the scene lighting and the camera, are used to map real world luminances into an acceptable image on a

display medium. In synthetic image generation, our goal is to avoid active control of lighting and camera settings. Furthermore, we hope to improve tone mapping techniques by having direct numerical control over display values, rather than depending on the physical limitations of chemistry or electronics.

Consider a typical scene that poses a problem for tone reproduction in both photography and computer graphics image synthesis systems. The scene is a room illuminated by a window that looks out on a sunlit landscape. A human observer inside the room can easily see individual objects in the room, as well as features in the outdoor landscape. This is because the eye adapts locally as we scan the different regions of the scene. If we attempt to photograph our view, the result is disappointing. Either the window is over-exposed and we can't see outside, or the interior of the room is under-exposed and looks black. Current computer graphics tone operators either produce the same disappointing result, or introduce artifacts that do not match our perception of the actual scene.

In this paper, we present a new tone reproduction operator that reliably maps real world luminances to display luminances, even in the problematic case just described. We consider the following two criteria most important for reliable tone mapping:

1. Visibility is reproduced. You can see an object in the real scene if and only if you can see it in the display. Objects are not obscured in under- or over-exposed regions, and features are not lost in the middle.
2. Viewing the image produces a subjective experience that corresponds with viewing the real scene. That is, the display should correlate well with memory of the actual scene. The overall impression of brightness, contrast, and color should be reproduced.

Previous tone mapping operators have generally met one of these criteria at the expense of the other. For example, some preserve the visibility of objects while changing the impression of contrast, while others preserve the overall impression of brightness at the expense of visibility.

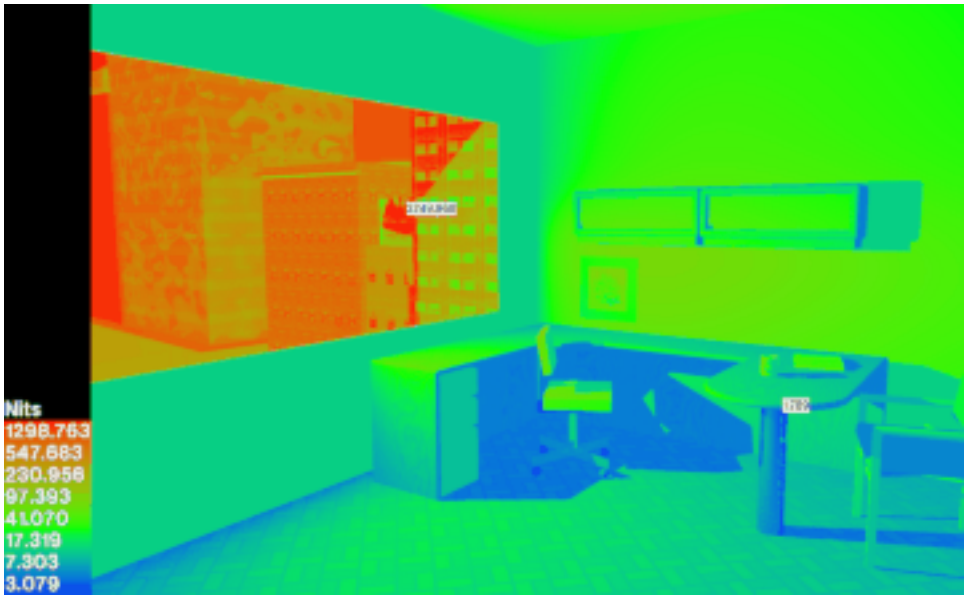


Figure 1. A false color image showing the world luminance values for a window office in candelas per meter squared (cd/m^2 or Nits).

The new tone mapping operator we present addresses our two criteria. We develop a method of modifying a luminance histogram, discovering clusters of adaptation levels and efficiently mapping them to display values to preserve local contrast visibility. We then use models for glare, color sensitivity and visual acuity to reproduce imperfections in human vision that further affect visibility and appearance.

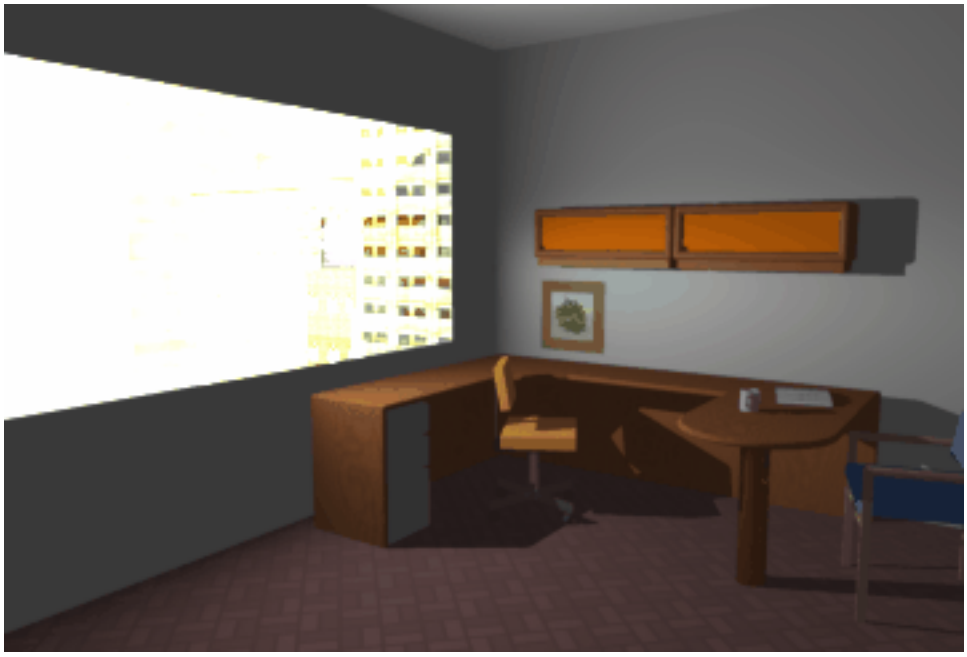


Figure 2. A linear mapping of the luminances in Figure 1 that over-exposes the view through the window.

Figure 3. A linear mapping of the luminances in Figure 1 that under-exposes the view of the interior.

Figure 4. The luminances in Figure 1 mapped to preserve the visibility of both indoor and outdoor features using the new tone mapping techniques described in this paper.

2 Previous Work

The high dynamic range problem was first encountered in computer graphics when physically accurate illumination methods were developed for image synthesis in the 1980's. (See Glassner [Glassner95] for a comprehensive review.) Previous methods for generating images were designed to automatically produce dimensionless values more or less evenly distributed in the range 0 to 1 or 0 to 255, which could be readily mapped to a display device. With the advent of radiosity and Monte Carlo path tracing techniques, we began to compute images in terms of real units with the real dynamic range of physical illumination. Figure 1 is a false color image showing the magnitude and distribution of luminance values in a typical indoor scene containing a window to a sunlit exterior. The goal of image synthesis is to produce results such as Figure 4, which match our impression of what such a scene looks like. Initially though, researchers found that a wide range of displayable images could be obtained from the same input luminances -- such as the unsatisfactory over- and under-exposed linear reproductions of the image in Figures 2 and 3.

Initial attempts to find a consistent mapping from computed to displayable luminances were ad hoc and developed for computational convenience. One approach is to use a function that collapses the high dynamic range of luminance into a small numerical range. By taking the cube root of luminance, for example, the range of values is reduced to something that is easily mapped to the display range. This approach generally preserves visibility of objects, our first criterion for a tone mapping operator. However, condensing the range of values in this way reduces fine detail visibility, and distorts impressions of brightness and contrast, so it does not fully match visibility or reproduce the subjective appearance required by our second criterion.

A more popular approach is to use an arbitrary linear scaling, either mapping the average of luminance in the real world to the average of the display, or the maximum non-light source luminance to the display maximum. For scenes with a dynamic range similar to the display device, this is successful. However, linear scaling methods do not maintain visibility in scenes with high dynamic range, since very bright and very dim values are clipped to fall within the display's limited dynamic range. Furthermore, scenes are mapped the same way regardless of the absolute values of luminance. A scene illuminated by a search light could be mapped to the same image as a scene illuminated by a flashlight, losing the overall impression of brightness and so losing the subjective correspondence between viewing the real and display-mapped scenes.

A tone mapping operator proposed by Tumblin and Rushmeier [Tumblin93] concentrated on the problem of preserving the viewer's overall impression of brightness. As the light level that the eye adapts to in a scene changes, the relationship between brightness (the subjective impression of the viewer) and luminance (the quantity of light in the visible range) also changes. Using a brightness function proposed by Stevens and Stevens [Stevens60], they developed an operator that would preserve the overall impression of brightness in the image, using one adaptation value for real scene, and another adaptation value for the displayed image. Because a single adaptation level is used for the scene, though, preservation of brightness in this case is at the expense of visibility. Areas that are very bright or dim are clipped, and objects in these areas are obscured.

Ward [Ward91] developed a simpler tone mapping method, designed to preserve feature visibility. In this method, a non-arbitrary linear scaling factor is found that preserves the impression of contrast (i.e., the visible changes in luminance) between the real and displayed image at a particular fixation point. While visibility is maintained at this adaptation point, the linear scaling factor still results in the clipping of very high and very low values, and correct visibility is not maintained throughout the image.

Chiu et al. [Chiu93] addressed this problem of global visibility loss by scaling luminance values based on a spatial average of luminances in pixel neighborhoods. Values in bright or dark areas would not be clipped, but scaled according to different values based on their spatial location. Since the human eye is less sensitive to variations at low spatial frequencies than high ones, a variable scaling that changes slowly relative to image features is not immediately visible. However, in a room with a bright source and dark corners, the method inevitably produces display luminance gradients that are the opposite of real world gradients. To make a dark region around a bright source, the transition from a dark area in the room to a bright area shows a decrease in brightness rather than an increase. This is illustrated in Figure 5 which shows a bright source with a dark halo around it. The dark halo that facilitates rendering the visibility of the bulb disrupts what should be a symmetric pattern of light cast by the bulb on the wall behind it. The reverse gradient fails to preserve the subjective correspondence between the real room and the displayed image.

Inspired by the work of Chiu et al., Schlick [Schlick95] developed an alternative method that could compute a spatially varying tone mapping. Schlick's work concentrated on improving computational efficiency and simplifying parameters, rather than improving the subjective correspondence of previous methods.

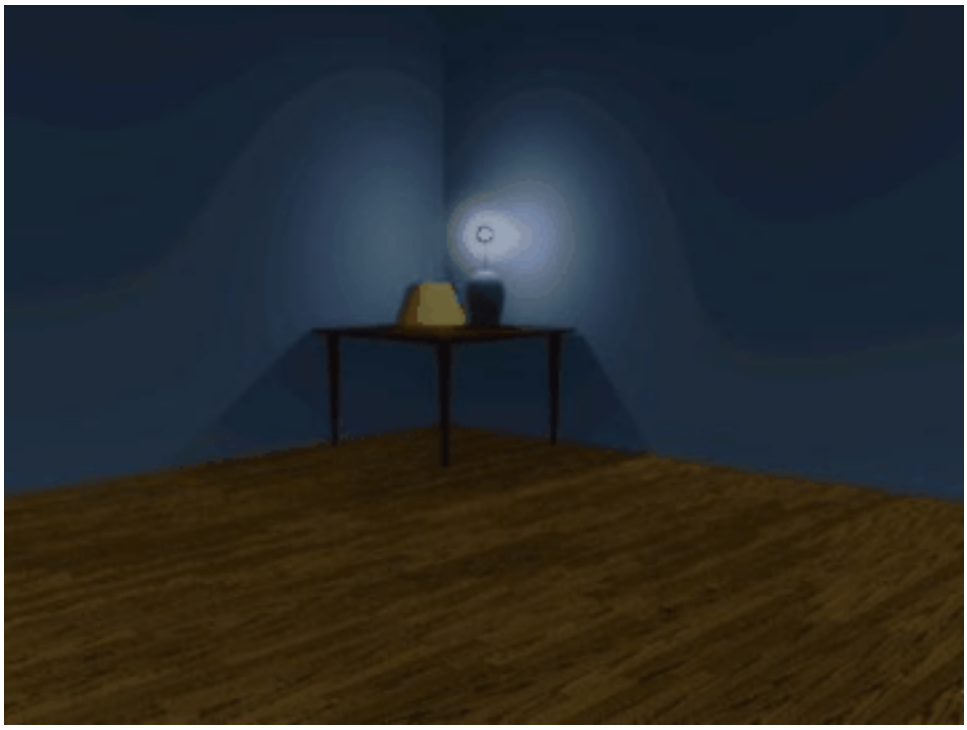


Figure 5. Dynamic range compression based on a spatially varying scale factor (from [Chiu93]).

Contrast, brightness and visibility are not the only perceptions that should be maintained by a tone mapping operator. Nakamae et al. [Nakamae90] and Spencer et al. [Spencer95] have proposed methods to simulate the effects of glare. These methods simulate the scattering in the eye by spreading the effects of a bright source in an image. Ferwerda et al. [Ferwerda96] proposed a method that accounts for changes in spatial acuity and color sensitivity as a function of light level. Our work is largely inspired by these papers, and we borrow heavily from Ferwerda et al. in particular. Besides maintaining visibility and the overall impression of brightness, the effects of glare, spatial acuity and color sensitivity must be included to fully meet our second criterion for producing a subjective correspondence between the viewer in the real scene and the viewer of the synthetic image.

A related set of methods for adjusting image contrast and visibility have been developed in the field of image processing for image enhancement (e.g., see Chapter 3 in [Green83]). Perhaps the best known image enhancement technique is histogram equalization. In histogram equalization, the grey levels in an image are redistributed more evenly to make better use of the range of the display device. Numerous improvements have been made to simple equalization by incorporating models of perception. Frei [Frei77] introduced histogram hyperbolization that attempts to redistribute perceived brightness, rather than screen grey levels. Frei approximated brightness using the logarithm of luminance. Subsequent researchers such as Mokrane [Mokrane92] have introduced methods that use more sophisticated models of perceived brightness and contrast.

The general idea of altering histogram distributions and using perceptual models to guide these alterations can be applied to tone mapping. However, there are two important

differences between techniques used in image enhancement and techniques for image synthesis and real-world tone mapping:

1. In image enhancement, the problem is to correct an image that has already been distorted by photography or video recording and collapsed into a limited dynamic range. In our problem, we begin with an undistorted array of real world luminances with a potentially high dynamic range.
2. In image enhancement, the goal is to take an imperfect image and *maximize* visibility or contrast. Maintaining subjective correspondence with the original view of the scene is irrelevant. In our problem, we want to maintain subjective correspondence. We want to *simulate* visibility and contrast, not maximize it. We want to produce visually accurate, not enhanced, images.

3 Overview of the New Method

In constructing a new method for tone mapping, we wish to keep the elements of previous methods that have been successful, and overcome the problems.

Consider again the room with a window looking out on a sunlit landscape. Like any high dynamic range scene, luminance levels occur in clusters, as shown in the histogram in Figure 6, rather than being uniformly distributed throughout the dynamic range. The failure of any method that uses a single adaptation level is that it maps a large range of sparsely populated real world luminance levels to a large range of display values. If the eye were sensitive to absolute values of luminance difference, this would be necessary. However, the eye is only sensitive to the fact that there are bright areas and dim areas. As long as the bright areas are displayed by higher luminances than the dim areas in the final image, the absolute value of the difference in luminance is not important. Exploiting this aspect of vision, we can close the gap between the display values for high and low luminance regions, and we have more display luminances to work with to render feature visibility.

Another failure of using a uniform adaptation level is that the eye rapidly adapts to the level of a relatively small angle in the visual field (i.e., about 1°) around the current fixation point [Moon&Spencer45]. When we look out the window, the eye adapts to the high exterior level, and when we look inside, it adapts to the low interior level. Chiu et al. [Chiu93] attempted to account for this using spatially varying scaling factors, but this method produces noticeable gradient reversals, as shown in Figure 5.

Rather than adjusting the adaptation level based on spatial location in the image, we will base our mapping on the population of the luminance adaptation levels in the image. To identify clusters of luminance levels and *initially* map them to display values, we will use the cumulative distribution of the luminance histogram. More specifically, we will start with a cumulative distribution based on a logarithmic approximation of brightness from luminance values.

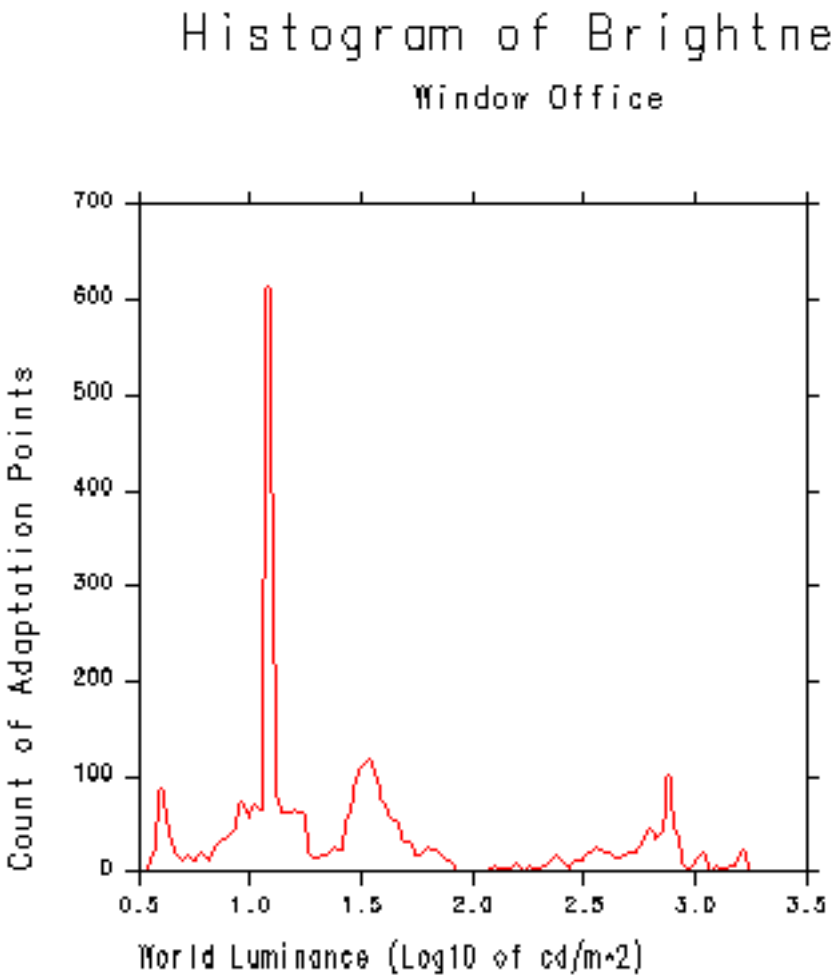


Figure 6. A histogram of adaptation values from Figure 1 (1° spot luminance averages).

First, we calculate the population of levels from a luminance image of the scene in which each pixel represents 1° in the visual field. We make a crude approximation of the brightness values (i.e., the subjective response) associated with these luminances by taking the logarithm of luminance. (Note that we will not display logarithmic values, we will merely use them to obtain a distribution.) We then build a histogram and cumulative distribution function from these values. Since the brightness values are integrated over a small solid angle, they are in some sense based on a spatial average, and the resulting mapping will be local to a particular adaptation level. Unlike Chiu's method however, the mapping for a particular luminance level will be consistent throughout the image, and will be order preserving. Specifically, an increase in real scene luminance level will always be represented by an increase in display luminance. The histogram and cumulative distribution function will allow us to close the gaps of sparsely populated luminance values and avoid the clipping problems of single adaptation level methods. By deriving a single, global tone mapping operator from locally averaged adaptation levels, we avoid the reverse gradient artifacts associated with a spatially varying multiplier.

We will use this histogram only as a starting point, and impose restrictions to preserve (rather than maximize) contrast based on models of human perception using our knowledge of the true luminance values in the scene. Simulations of glare and variations in spatial acuity and color sensitivity will be added into the model to maintain subjective correspondence and visibility. In the end, we obtain a mapping of real world to display luminance similar to the one shown in Figure 7.

For our target display, all mapped brightness values below 1 cd/m^2 (0 on the vertical axis) or above 100 (2 on the vertical axis) are lost because they are outside the displayable range. Here we see that the dynamic range between 1.75 and 2.5 has been compressed, yet we don't notice it in the displayed result (Figure 4). Compared to the two linear operators, our new tone mapping is the only one that can represent the entire scene without losing object or detail visibility.

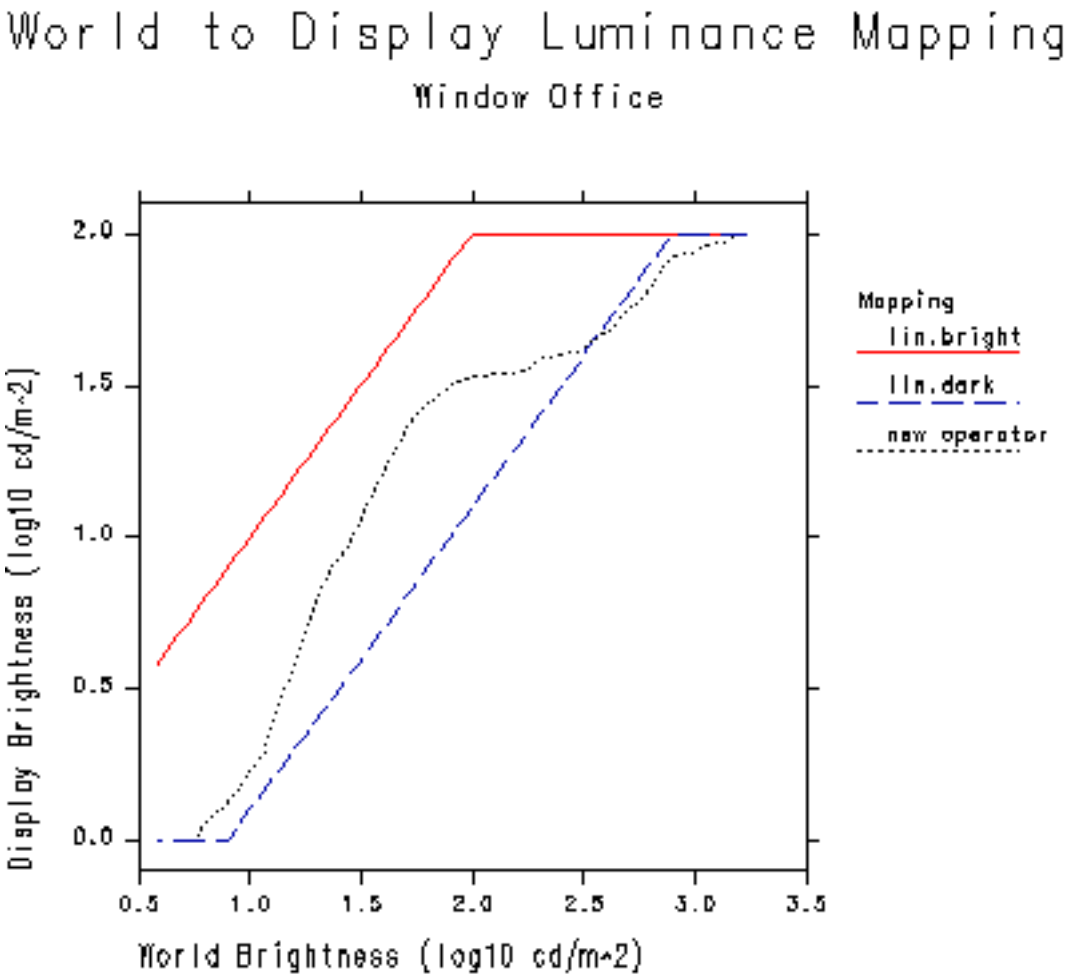


Figure 7. A plot comparing the global brightness mapping functions for Figures 1, 2, and 3, respectively.

In the following section, we illustrate this technique for histogram adjustment based on contrast sensitivity. After this, we describe models of glare, color sensitivity and visual

acuity that complete our simulation of the measurable and subjective responses of human vision. Finally, we complete the methods presentation with a summary describing how all the pieces fit together.

4 Histogram Adjustment

In this section, we present a detailed description of our basic tone mapping operator. We begin with the introduction of symbols and definitions, and a description of the histogram calculation. We then describe a naive equalization step that partially accomplishes our goals, but results in undesirable artifacts. This method is then refined with a linear contrast ceiling, which is further refined using human contrast sensitivity data.

4.1 Symbols and Definitions

L_w	= world luminance (in candelas/meter^2)
B_w	= world brightness, $\log(L_w)$
L_{wmin}	= minimum world luminance for scene
L_{wmax}	= maximum world luminance for scene
L_d	= display luminance (in candelas/meter^2)
L_{dmin}	= minimum display luminance (black level)
L_{dmax}	= maximum display luminance (white level)
B_{de}	= computed display brightness, $\log(L_d)$ [Equation (4)]
N	= the number of histogram bins
T	= the total number of adaptation samples
$f(b_i)$	= frequency count for the histogram bin at b_i
Δb	= the bin step size in $\log(cd/m^2)$
P(b)	= the cumulative distribution function [Equation (2)]
log(x)	= natural logarithm of x
log10(x)	= decimal logarithm of x

4.2 Histogram Calculation

Since we are interested in optimizing the mapping between world adaptation and display adaptation, we start with a histogram of world adaptation luminances. The eye adapts for the best view in the fovea, so we compute each luminance over a 1° diameter solid angle corresponding to a potential foveal fixation point in the scene. We use a logarithmic scale for the histogram to best capture luminance population and subjective response over a wide dynamic range. This requires setting a minimum value as well as a maximum, since the logarithm of zero is -∞. For the minimum value, we use either the minimum 1° spot average, or 10^{-4} cd/m^2 (the lower threshold of human vision), whichever is larger. The maximum value is just the maximum spot average.

We start by filtering our original floating-point image down to a resolution that roughly corresponds to 1° square pixels. If we are using a linear perspective projection, the pixels on the perimeter will have slightly smaller diameter than the center pixels, but they will still be within the correct range. The following formula yields the correct resolution for

1° diameter pixels near the center of a linear perspective image:

$$S = 2 \tan(\theta/2) / 0.01745 \quad (1)$$

where:

S = width or height in pixels

θ = horizontal or vertical full view angle

0.01745 = number of radians in 1°

For example, the view width and height for Figure 4 are 63° and 45° respectively, which yield a sample image resolution of 70 by 47 pixels. Near the center, the pixels will be 1° square exactly, but near the corners, they will be closer to 0.85° for this wide-angle view. The filter kernel used for averaging will have little influence on our result, so long as every pixel in the original image is weighted similarly. We employ a simple box filter.

From our reduced image, we compute the logarithms of the floating-point luminance values. Here, we assume there is some method for obtaining the absolute luminances at each spot sample. If the image is uncalibrated, then the corrections for human vision will not work, although the method may still be used to optimize the visible dynamic range. (We will return to this in the summary.)

The histogram is taken between the minimum and maximum values mentioned earlier in equal-sized bins on a log(luminance) scale. The algorithm is not sensitive to the number of bins, so long as there are enough to obtain adequate resolution. We use 100 bins in all of our examples. The resulting histogram for Figure 1 is shown in Figure 6.

4.2.1 Cumulative Distribution

The cumulative frequency distribution is defined as:

$$P(b) = \frac{\sum_{b_i < b} f(b_i)}{T} \quad (2)$$

where:

$T = \sum_{b_i} f(b_i)$ (i.e., the total number of samples)

Later on, we will also need the derivative of this function. Since the cumulative distribution is a numerical integration of the histogram, the derivative is simply the histogram with an appropriate normalization factor. In our method, we approximate a continuous distribution and derivative by interpolating adjacent values linearly. The derivative of our function is:

$$\frac{dP(b)}{db} = \frac{f(b)}{T \Delta b} \quad (3)$$

where:

$\Delta b = \frac{\left[\log(L_{wmax}) - \log(L_{wmin})\right]}{N}$ (i.e., the size of each bin)

Figure 8. Rendering of a bathroom model mapped with a linear operator.

4.3 Naive Histogram Equalization

If we wanted all the brightness values to have equal probability in our final displayed image, we could now perform a straightforward histogram equalization. Although this is not our goal, it is a good starting point for us. Based on the cumulative frequency distribution just described, the equalization formula can be stated in terms of brightness as follows:

$$B_{de} = \log(L_{dmin}) + \left[\log(L_{dmax}) - \log(L_{dmin})\right] P(B_w) \qquad (4)$$

The problem with naive histogram equalization is that it not only compresses dynamic range (contrast) in regions where there are few samples, it also *expands* contrast in highly populated regions of the histogram. The net effect is to exaggerate contrast in large areas of the displayed image. Take as an example the scene shown in Figure 8. Although we cannot see the region surrounding the lamps due to the clamped linear tone mapping operator, the image appears to us as more or less normal. Applying the naive histogram equalization, Figure 9 is produced. The tiles in the shower now have a mottled appearance. Because this region of world luminance values is so well represented, naive

histogram equalization spreads it out over a relatively larger portion of the display's dynamic range, generating superlinear contrast in this region.

Figure 9. Naive histogram equalization allows us to see the area around the light sources but contrast is exaggerated in other areas such as the shower tiles.

4.4 Histogram Adjustment with a Linear Ceiling

If the contrast being produced is too high, then what is an appropriate contrast for representing image features? The crude answer is that the contrast in any given region should not exceed that produced by a linear tone mapping operator, since linear operators produce satisfactory results for scenes with limited dynamic range. We will take this simple approach first, and later refine our answer based on human contrast sensitivity.

A linear ceiling on the contrast produced by our tone mapping operator can be written thus:

$$\frac{dL_d}{dL_w} \le \frac{L_d}{L_w} \qquad (5a)$$

That is, the derivative of the display luminance with respect to the world luminance must not exceed the display luminance divided by the world luminance. Since we have an expression for the display luminance as a function of world luminance for our naive histogram equalization, we can differentiate the exponentiation of Equation (4) using the chain rule and the derivative from Equation (3) to get the following inequality:

$$\exp(B_{de})\ \frac{f(B_w)}{T\Delta b}\ \frac{\log(L_{dmax})-\log(L_{dmin})}{L_w} \le \frac{L_d}{L_w} \qquad (5b)$$

Since L_d is equal to $\exp(B_{de})$, this reduces to a constant ceiling on f(b):

$$f(b) \le \frac{T\Delta b}{\log(L_{dmax})-\log(L_{dmin})} \qquad (5c)$$

In other words, so long as we make sure no frequency count exceeds this ceiling, our resulting histogram will not exaggerate contrast. How can we create this modified histogram? We considered both truncating larger counts to this ceiling and redistributing counts that exceeded the ceiling to other histogram bins. After trying both methods, we found truncation to be the simplest and most reliable approach. The only complication introduced by this technique is that once frequency counts are truncated, T changes, which changes the ceiling. We therefore apply iteration until a tolerance criterion is met, which says that fewer than 2.5% of the original samples exceed the ceiling.[1] Our pseudocode for `histogram_ceiling` is given below:

```
boolean function histogram_ceiling()
tolerance := 2.5% of histogram total
repeat {
     trimmings := 0
     compute the new histogram total T
     if T < tolerance then
          return FALSE
     foreach histogram bin i do
          compute the ceiling
          if f(b_i) > ceiling then {
               trimmings += f(b_i) - ceiling
               f(b_i) := ceiling
          }
} until trimmings <= tolerance
return TRUE
```

This iteration will fail to converge (and the function will return FALSE) if and only if the dynamic range of the output device is already ample for representing the sample luminances in the original histogram. This is evident from Equation (5c), since Δb is the world brightness range over the number of bins:

$$f(b_i) \le \frac{T}{N}\ \frac{\left[\log(L_{wmax})-\log(L_{wmin})\right]}{\left[\log(L_{dmax})-\log(L_{dmin})\right]} \qquad (5d)$$

[1] The tolerance of 2.5% was chosen as an arbitrary small value, and it seems to make little difference either to the convergence time or the results.

If the ratio of the world brightness range over the display brightness range is less than one (i.e., our world range fits in our display range), then our frequency ceiling is less than the total count over the number of bins. Such a condition will never be met, since a uniform distribution of samples would still be over the ceiling in every bin. It is easiest to detect this case at the outset by checking the respective brightness ranges, and applying a simple linear operator if compression is unnecessary.

We call this method *histogram adjustment* rather than histogram equalization because the final brightness distribution is *not equalized*. The net result is a mapping of the scene's high dynamic range to the display's smaller dynamic range that minimizes visible contrast distortions, by compressing under-represented regions without expanding over-represented ones.

Figure 10 shows the results of our histogram adjustment algorithm with a linear ceiling. The problems of exaggerated contrast are resolved, and we can still see the full range of brightness. A comparison of these tone mapping operators is shown in Figure 11. The naive operator is superlinear over a large range, seen as a very steep slope near world luminances around $10^{0.8}$.

Figure 10. Histogram adjustment with a linear ceiling on contrast preserves both lamp visibility and tile appearance.

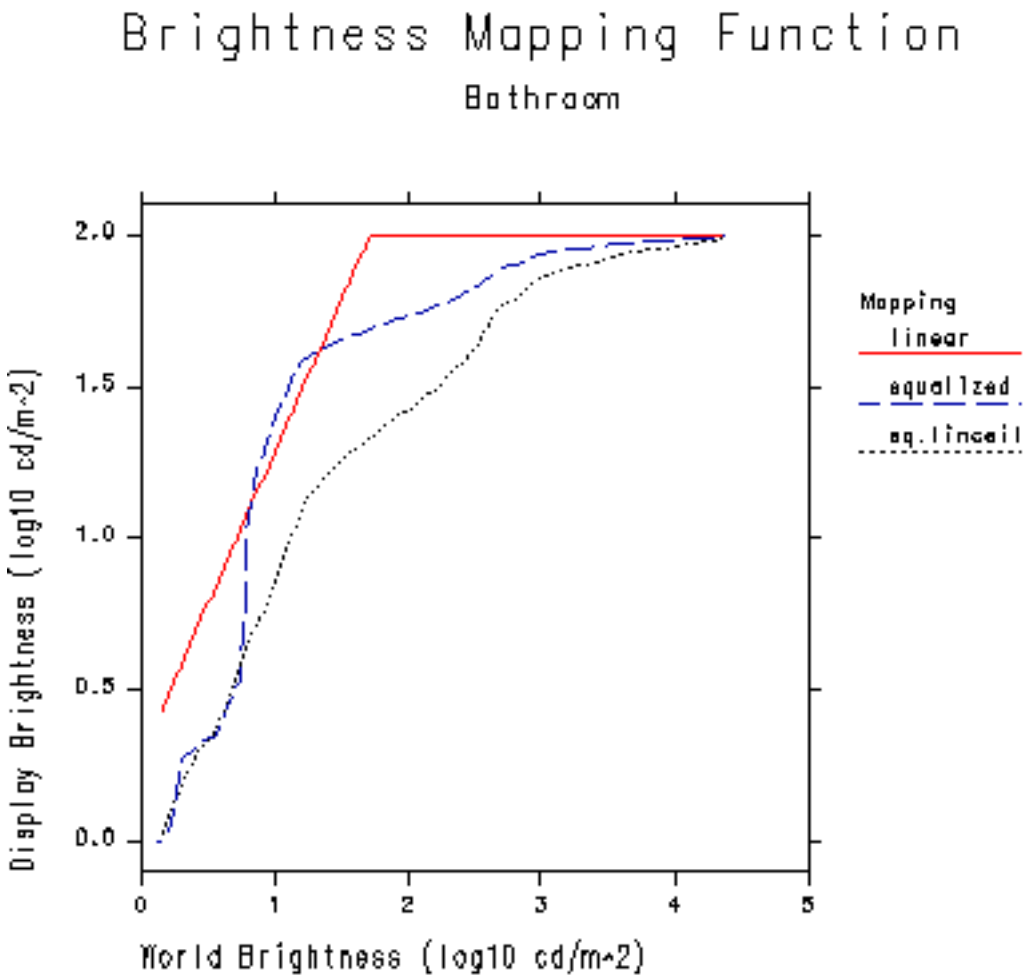


Figure 11. A comparison of naive histogram equalization with histogram adjustment using a linear contrast ceiling.

The method we have just presented is itself quite useful. We have managed to overcome limitations in the dynamic range of typical displays without introducing objectionable contrast compression artifacts in our image. In situations where we want to get a good, natural-looking image without regard to how well a human observer would be able to see in a real environment, this may be an optimal solution. However, if we are concerned with reproducing both visibility and subjective experience in our displayed image, then we must take it a step further and consider the *limitations* of human vision.

4.5 Histogram Adjustment Based on Human Contrast Sensitivity

Although the human eye is capable of adapting over a very wide dynamic range (on the order of 10^9), we do not see equally well at all light levels. As the light grows dim, we have more and more trouble detecting contrast. The relationship between adaptation luminance and the minimum detectable luminance change is well studied [CIE81]. For consistency with earlier work, we use the same detection threshold function used by Ferwerda et al. [Ferwerda96]. This function covers sensitivity from the lower limit of

human vision to daylight levels, and accounts for both rod and cone response functions. The piecewise fit is reprinted in Table 1.

log10 of just noticeable difference	**applicable luminance range**
-2.86	$\log10(L_a) < -3.94$
$(0.405 \log10(L_a) + 1.6)^{2.18} - 2.86$	$-3.94 \le \log10(L_a) < -1.44$
$\log10(L_a) - 0.395$	$-1.44 \le \log10(L_a) < -0.0184$
$(0.249 \log(L_a) + 0.65)^{2.7} - 0.72$	$-0.0184 \le \log10(L_a) < 1.9$
$\log10(L_a) - 1.255$	$\log10(L_a) \ge 1.9$

Table 1. Piecewise approximation for $\Delta L_t(L_a)$.

We name this combined sensitivity function:

$$\Delta L_t(L_a) = \text{"just noticeable difference" for adaptation level } L_a \qquad (6)$$

Ferwerda et al. did not combine the rod and cone sensitivity functions in this manner, since they used the two ranges for different tone mapping operators. Since we are using this function to control the maximum reproduced contrast, we combine them at their crossover point of $10^{-0.0184}$ cd/m^2.

To guarantee that our display representation does not exhibit contrast that is more noticeable than it would be in the actual scene, we constrain the slope of our operator to the ratio of the two adaptation thresholds for the display and world, respectively. This is the same technique introduced by Ward [Ward91] and used by Ferwerda et al. [Ferwerda96] to derive a global scale factor. In our case, however, the overall tone mapping operator will not be linear, since the constraint will be met at all potential adaptation levels, not just a single selected one. The new ceiling can be written as:

$$\frac{dL_d}{dL_w} \le \frac{\Delta L_t(L_d)}{\Delta L_t(L_w)} \qquad (7a)$$

As before, we compute the derivative of the histogram equalization function (Equation (4)) to get:

$$\exp(B_{de}) \; \frac{f(B_w)}{T\Delta b} \; \frac{\log(L_{dmax}) - \log(L_{dmin})}{L_w} \le \frac{\Delta L_t(L_d)}{\Delta L_t(L_w)} \qquad (7b)$$

However, this time the constraint does not reduce to a constant ceiling for f(b). We notice that since L_d equals $\exp(B_{de})$ and B_{de} is a function of L_w from Equation (4), our

ceiling is completely defined for a given P(b) and world luminance, L_w:

$$f(B_w) \le \frac{\Delta L_t(L_d)}{\Delta L_t(L_w)} \frac{T \Delta b L_w}{\left[\log(L_{dmax}) - \log(L_{dmin})\right] L_d} \qquad (7c)$$

where:

L_d = $\exp(B_{de})$, B_{de} given in Equation (4)

Once again, we must iterate to a solution, since truncating bin counts will affect T and P(b). We reuse the `histogram_ceiling` procedure given earlier, replacing the linear contrast ceiling computation with the above formula.

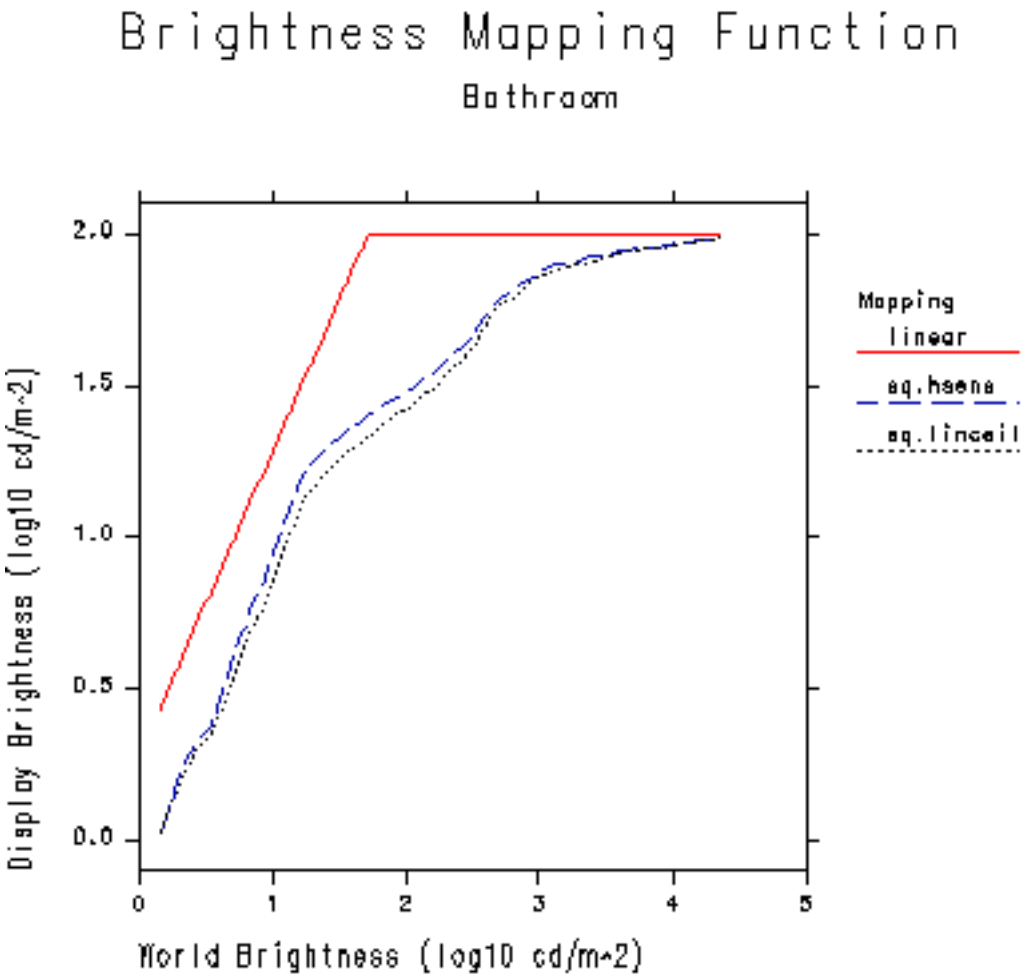


Figure 12. Our tone mapping operator based on human contrast sensitivity compared to the histogram adjustment with linear ceiling used in Figure 10. Human contrast sensitivity makes little difference at these light levels.

Figure 12 shows the same curves for the linear tone mapping and histogram adjustment with linear clamping shown before in Figure 11, but with the curve for naive histogram

equalization replaced by our human visibility matching algorithm. We see the two histogram adjustment curves are very close. In fact, we would have some difficulty differentiating images mapped with our latest method and histogram adjustment with a linear ceiling. This is because the scene we have chosen has most of its luminance levels in the same range as our display luminances. Therefore, the ratio between display and world luminance detection thresholds is close to the ratio of the display and world adaptation luminances. This is known as Weber's law [Riggs71], and it holds true over a wide range of luminances where the eye sees equally well This correspondence makes the right-hand side of Equations (5b) and (7b) equivalent, and so we should expect the same result as a linear ceiling.

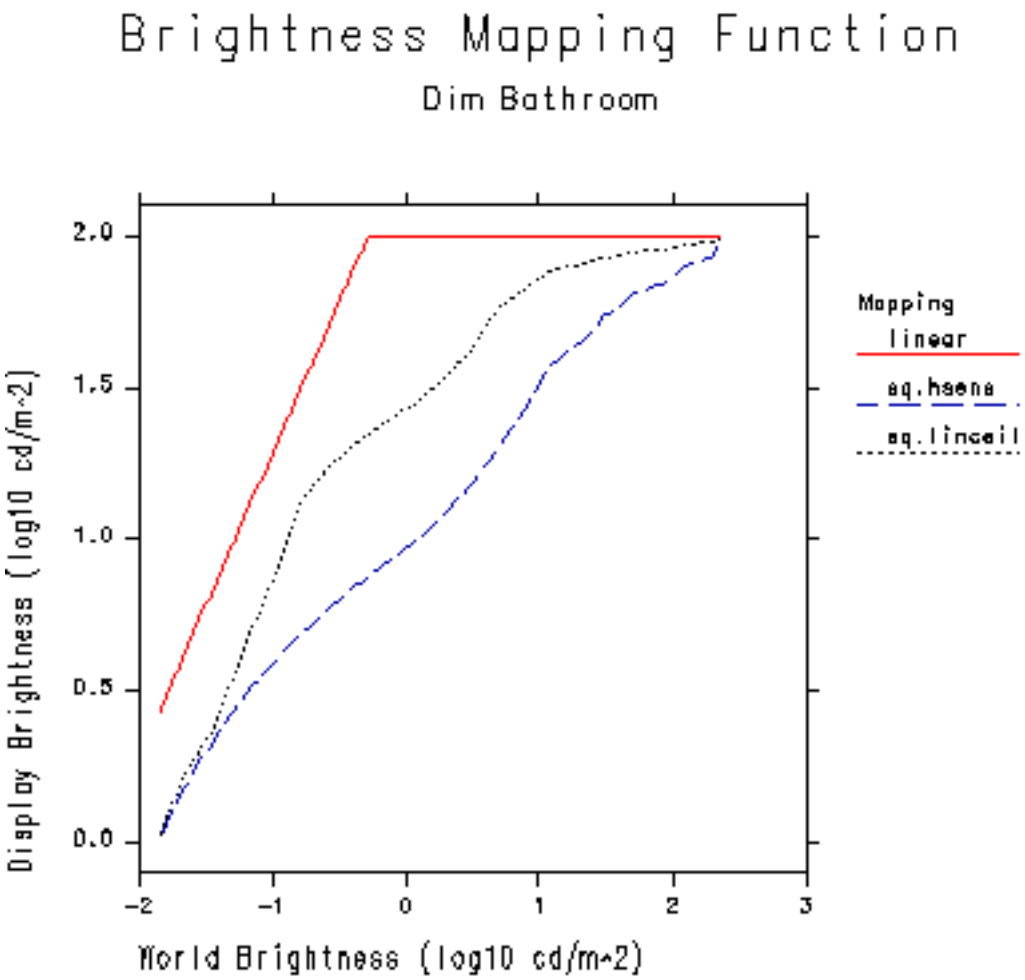


Figure 13. The brightness map for the bathroom scene with lights dimmed to 1/100th of their original intensity, where human contrast sensitivity makes a difference.

To see a contrast sensitivity effect, our world adaptation would have to be very different from our display adaptation. If we reduce the light level in the bathroom by a factor of 100, our ability to detect contrast is diminished. This shows up in a relatively larger detection threshold in the denominator of Equation (7c), which reduces the ceiling for the

frequency counts. The change in the tone mapping operator is plotted in Figure 13 and the resulting image is shown in Figure 14.

Figure 13 shows that the linear mapping is unaffected, since we just raise the scale factor to achieve an average exposure. Likewise, the histogram adjustment with a linear ceiling maps the image to the same display range, since its goal is to reproduce linear contrast. However, the ceiling based on human threshold visibility limits contrast over much of the scene, and the resulting image is darker and less visible everywhere except the top of the range, which is actually shown with higher contrast since we now have display range to spare.

Figure 14 is darker and the display contrast is reduced compared to Figure 10. Because the tone mapping is based on *local adaptation* rather than a single global or spot average, threshold visibility is reproduced *everywhere* in the image, not just around a certain set of values. This criterion is met within the limitations of the display's dynamic range.

Figure 14. The dimmed bathroom scene mapped with the function shown in Figure 13.

5 Human Visual Limitations

We have seen how histogram adjustment matches display contrast visibility to world visibility, but we have ignored three important limitations in human vision: glare, color sensitivity and visual acuity. Glare is caused by bright sources in the visual periphery, which scatter light in the lens of the eye, obscuring foveal vision. Color sensitivity is lost in dark environments, as the light-sensitive rods take over for the color-sensitive cone system. Visual acuity is also impaired in dark environments, due to the complete loss of cone response and the quantum nature of light sensation.

In our treatment, we will rely heavily on previous work performed by Moon and Spencer [Moon&Spencer45] and Ferwerda et al. [Ferwerda96], applying it in the context of a locally adapted visibility-matching model.

5.1 Veiling Luminance

Bright *glare sources* in the periphery reduce contrast visibility because light scattered in the lens and aqueous humor obscures the fovea; this effect is less noticeable when looking directly at a source, since the eye adapts to the high light level. The influence of glare sources on contrast sensitivity is well studied and documented. We apply the work of Holladay [Holladay26] and Moon and Spencer [Moon&Spencer45], which relates the effective adaptation luminance to the foveal average and glare source position and illuminance.

In our presentation, we will first compute a low resolution veil image from our foveal sample values. We will then interpolate this veil image to add glare effects to the original rendering. Finally, we will apply this veil as a correction to the adaptation luminances used for our contrast, color sensitivity and acuity models.

Moon and Spencer base their formula for adaptation luminance on the effect of individual glare sources measured by Holladay, which they converted to an integral over the entire visual periphery. The resulting glare formula gives the effective adaptation luminance at a particular fixation for an arbitrary visual field:

$$L_a = 0.913 L_f + \frac{K}{\pi} \iint_{\theta > \theta_f} \frac{L(\theta,\phi)}{\theta^2} \cos(\theta)\sin(\theta)\, d\theta\, d\phi \qquad (8)$$

where:

L_a	= corrected adaptation luminance (in cd/m^2)
L_f	= the average foveal luminance (in cd/m^2)
$L(\theta,\phi)$	= the luminance in the direction (θ,ϕ)
θ_f	= foveal half angle, approx. 0.00873 radians (0.5°)
K	= constant measured by Holladay, 0.0096

The constant 0.913 in this formula is the remainder from integrating the second part assuming one luminance everywhere. In other words, the periphery contributes less than 9% to the average adaptation luminance, due to the small value Holladay determined for K. If there are no bright sources, this influence can be safely neglected. However, bright sources will significantly affect the adaptation luminance, and should be considered in our model of contrast sensitivity.

To compute the veiling luminance corresponding to a given foveal sample (i.e., fixation point), we can convert the integral in Equation (8) to an average over peripheral sample values:

$$L_{vi} = 0.087 \frac{\sum_{j \neq i} \frac{L_j \cos(\theta_{i,j})}{\theta_{i,j}^2}}{\sum_{j \neq i} \frac{\cos(\theta_{i,j})}{\theta_{i,j}^2}} \qquad (9)$$

where:

L_{vi} = veiling luminance for fixation point i
L_j = foveal luminance for fixation point j
$\theta_{i,j}$ = angle between sample i and j (in radians)

Since we must compute this sum over all foveal samples j for each fixation point i, the calculation can be very time consuming. We therefore reduce our costs by approximating the weight expression as:

$$\frac{\cos\theta}{\theta^2} \approx \frac{\cos\theta}{2 - 2\cos\theta} \qquad (10)$$

Since the angles between our samples are most conveniently available as vector dot products, which is the cosine, the above weight computation is quite fast. However, for large images (in terms of angular size), the L_{vi} calculation is still the most computationally expensive step in our method due to the double iteration over i and j.

To simulate the effect of glare on visibility, we simply add the computed veil map to our original image. Just as it occurs in the eye, the veiling luminance will obscure the visible contrast on the display by adding to both the background and the foreground luminance.[2] This was the original suggestion made by Holladay, who noted that the effect glare has on luminance threshold visibility is equivalent to what one would get by adding the veiling luminance function to the original image [Holladay26]. This is quite straightforward once we have computed our foveal-sampled veiling image given in Equation (9). At each image pixel, we perform the following calculation:

$$L_{pvk} = 0.913 L_{pk} + L_v(k) \qquad (11)$$

where:

L_{pvk} = veiled pixel at image position k
L_{pk} = original pixel at image position k
$L_v(k)$ = interpolated veiling luminance at k

The $L_v(k)$ function is a simple bilinear interpolation on the four closest samples in our veil image computed in Equation (9). The final image will be lighter around glare sources, and just slightly darker *on* glare sources (since the veil is effectively being spread away from bright points). Although we have shown this as a luminance calculation, we retain color information so that our veil has the same color cast as the responsible glare source(s).

[2]The contrast is defined as the ratio of the foreground minus the background over the background, so adding luminance to both foreground and background reduces contrast.

Figure 15 shows our original, fully lit bathroom scene again, this time adding in the computed veiling luminance. Contrast visibility is reduced around the lamps, but the veil falls off rapidly (as $1/\theta^2$) over other parts of the image. If we were to measure the luminance detection threshold at any given image point, the result should correspond closely to the threshold we would measure at that point in the actual scene.

Figure 15. Our tone reproduction operator for the original bathroom scene with veiling luminance added.

Since glare sources scatter light onto the fovea, they also affect the local adaptation level, and we should consider this in the other parts of our calculation. We therefore apply the computed veiling luminances to our foveal samples as a correction *before* the histogram generation and adjustment described in Section 4. We deferred the introduction of this correction factor to simplify our presentation, since in most cases it only weakly affects the brightness mapping function.

The correction to local adaptation is the same as Equation (11), but without interpolation, since our veil samples correspond one-to-one:

$$L_{ai} = 0.913 L_i + L_{vi} \qquad (12)$$

where:

L_{ai} = adjusted adaptation luminance at fixation point i
L_i = foveal luminance for fixation point i

We will also employ these L_{ai} adaptation samples for the models of color sensitivity and visual acuity that follow.

5.2 Color Sensitivity

To simulate the loss of color vision in dark environments, we use the technique presented by Ferwerda et al. [Ferwerda96] and ramp between a scotopic (grey) response function and a photopic (color) response function as we move through the mesopic range. The lower limit of the mesopic range, where cones are just starting to get enough light, is approximately 0.0056 cd/m^2. Below this value, we use the straight scotopic luminance. The upper limit of the mesopic range, where rods are no longer contributing significantly to vision, is approximately 5.6 cd/m^2. Above this value, we use the straight photopic luminance plus color. In between these two world luminances (i.e., within the mesopic range), our adjusted pixel is a simple interpolation of the two computed output colors, using a linear ramp based on luminance.

Since we do not have a value available for the scotopic luminance at each pixel, we use the following approximation based on a least squares fit to the colors on the Macbeth ColorChecker Chart™:

$$Y_{scot} = Y \left[1.33 \left(1 + \frac{Y+Z}{X} \right) - 1.68 \right] \qquad (13)$$

where:

Y_{scot} = scotopic luminance
X,Y,Z = photopic color, CIE 2° observer (Y is luminance)

This is a very good approximation to scotopic luminance for most natural colors, and it avoids the need to render another channel. We also have an approximation based on RGB values, but since there is no accepted standard for RGB primaries in computer graphics, this is much less reliable.

Figure 16 shows our dimmed bathroom scene with the human color sensitivity function in place. Notice there is still some veiling, even with the lights reduced to 1/100th their normal level. This is because the relative luminances are still the same, and they scatter in the eye as before. The only difference here is that the eye cannot adapt as well when there is so little light, so everything appears dimmer, including the lamps. The colors are clearly visible near the light sources, but gradually less visible in the darker regions.

Figure 16. Our dimmed bathroom scene with tone mapping using human contrast sensitivity, veiling luminance and mesopic color response.

5.3 Visual Acuity

Besides losing the ability to see contrast and color, the human eye loses its ability to resolve fine detail in dark environments. The relationship between adaptation level and foveal acuity has been measured in subject studies reported by Shaler [Shaler37]. At daylight levels, human visual acuity is very high, about 50 cycles/degree. In the mesopic range, acuity falls off rapidly from 42 cycles/degree at the top down to 4 cycles/degree near the bottom. Near the limits of vision, the visual acuity is only about 2 cycles/degree. Shaler's original data is shown in Figure 17 along with the following functional fit:

$$R(L_a) \approx 17.25\arctan(1.4\log10(L_a) + 0.35) + 25.72 \qquad (15)$$

where:

$R(L_a)$ = visual acuity in cycles/degree

L_a = local adaptation luminance (in cd/m^2)

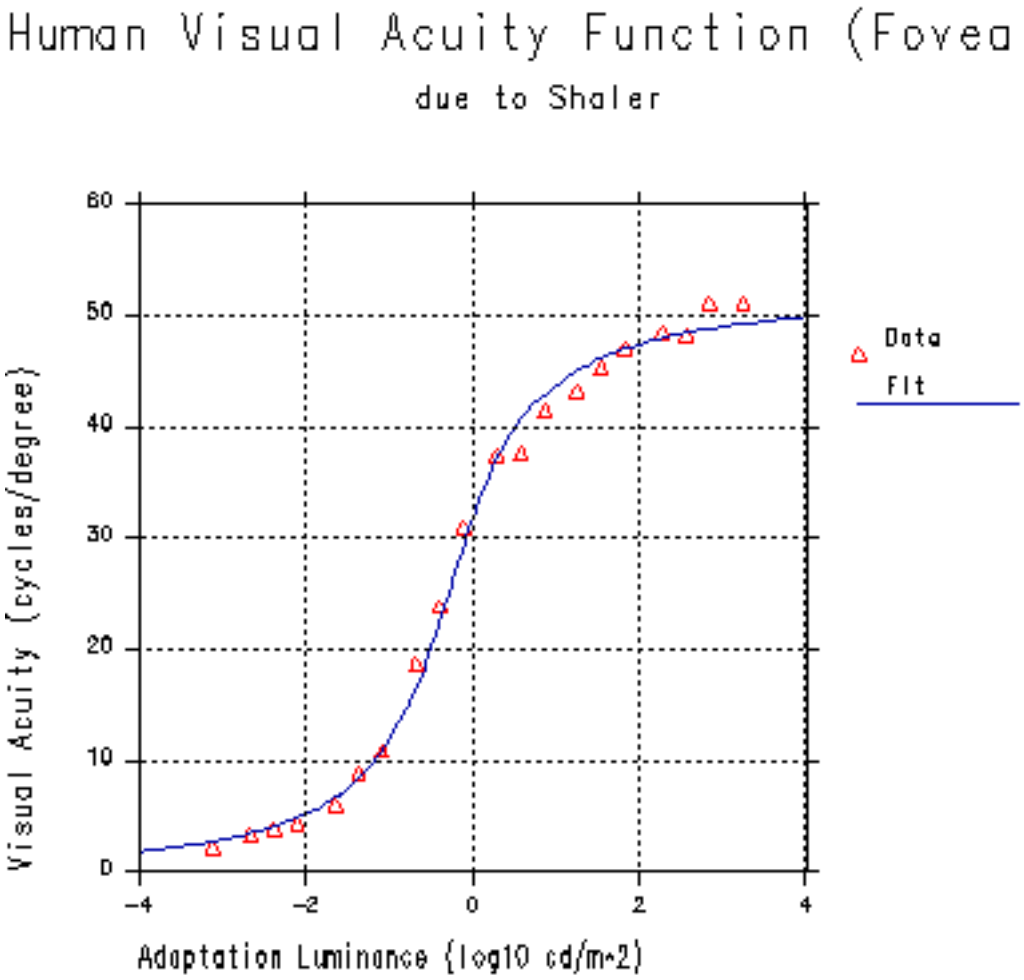


Figure 17. Shaler's visual acuity data and our functional fit to it.

In their tone mapping paper, Ferwerda et al. applied a global blurring function based on a single adaptation level [Ferwerda96]. Since we wish to adjust for acuity changes over a wide dynamic range, we must apply our blurring function locally according to the foveal adaptation computed in Equation (12). To do this, we implement a variable-resolution filter using an image pyramid and interpolation, which is the *mip map* introduced by Williams [Williams83] for texture mapping. The only difference here is that we are working with real values rather than integers pixels.

At each point in the image, we interpolate the local acuity based on the four closest (veiled) foveal samples and Shaler's data. It is very important to use the foveal data (L_{ai}) and not the original pixel value, since it is the fovea's adaptation that determines acuity. The resulting image will show higher resolution in brighter areas, and lower resolution in darker areas.

Figure 18 shows our dim bathroom scene again, this time applying the variable acuity operator applied together with all the rest. Since the resolution of the printed image is low, we enlarged two areas for a closer look. The bright area has an average level around

25 cd/m^2, corresponding to a visual acuity of about 45 cycles/degree. The dark area has an average level of around 0.05 cd/m^2, corresponding to a visual acuity of about 9 cycles/degree.

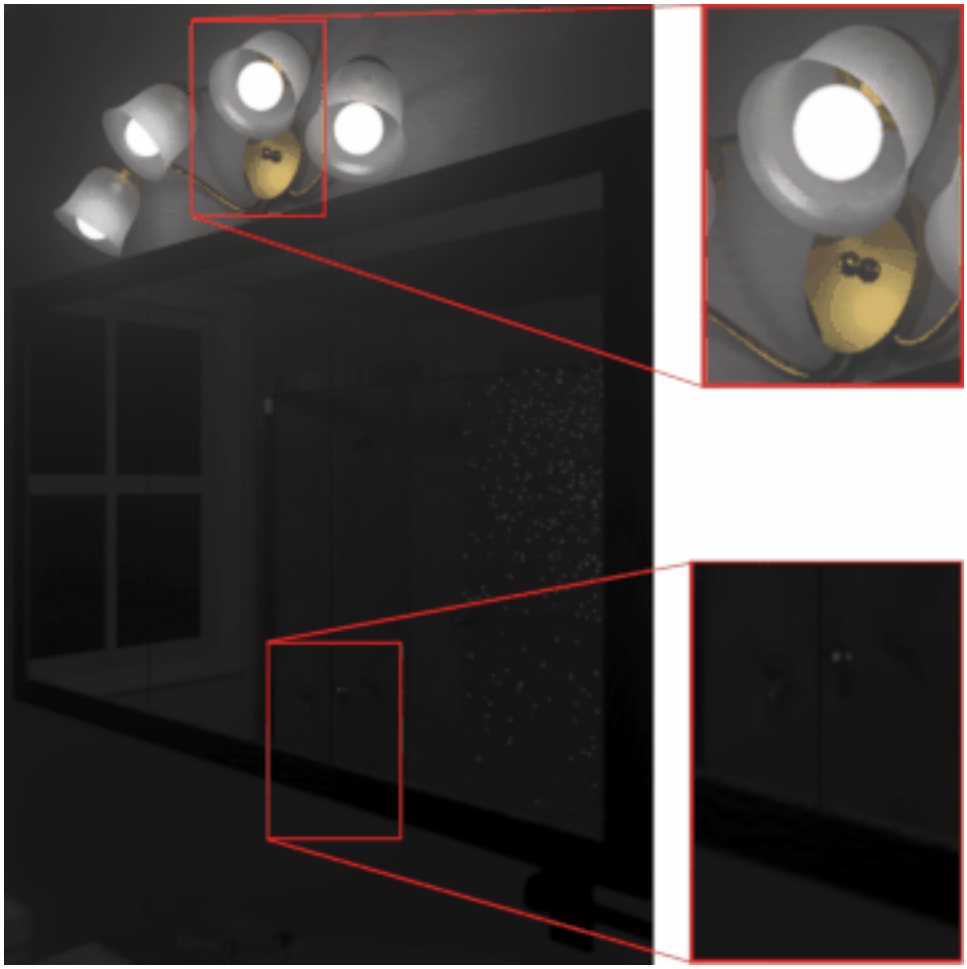


Figure 18. The dim bathroom scene with variable acuity adjustment. The insets show two areas, one light and one dark, and the relative blurring of the two.

6 Method Summary

We have presented a method for matching the visibility of high dynamic range scenes on conventional displays, accounting for human contrast sensitivity, veiling luminance, color sensitivity and visual acuity, all in the context of a local adaptation model. However, in presenting this method in parts, we have not given a clear idea of how the parts are integrated together into a working program.

The order in which the different processes are executed to produce the final image is of critical importance. These are the steps in the order they must be performed:

```
procedure match_visibility()
compute 1° foveal sample image
compute veil image
add veil to foveal adaptation image
add veil to image
blur image locally based on visual acuity function
apply color sensitivity function to image
generate histogram of effective adaptation image
adjust histogram to contrast sensitivity function
apply histogram adjustment to image
translate CIE results to display RGB values
end
```

We have not discussed the final step, mapping the computed display luminances and chrominances to appropriate values for the display device (e.g., monitor RGB settings). This is a well studied problem, and we refer the reader to the literature (e.g., [Hall89]) for details. Bear in mind that the mapped image accounts for the black level of the display, which must be subtracted out before applying the appropriate gamma and color corrections.

Although we state that the above steps must be carried out in this order, a few of the steps may be moved around, or removed entirely for a different effect. Specifically, it makes little difference whether the luminance veil is added before or after the blurring function, since the veil varies slowly over the image. Also, the color sensitivity function may be applied anywhere after the veil is added so long as it is before histogram adjustment.

If the goal is to optimize visibility and appearance without regard to the limitations of human vision, then all the steps between computing the foveal average and generating the histogram may be skipped, and a linear ceiling may be applied during histogram adjustment instead of the human contrast sensitivity function. The result will be an image with all parts visible on the display, regardless of the world luminance level or the presence of glare sources. This may be preferable when the only goal is to produce a nice-looking image, or when the absolute luminance levels in the original scene are unknown.

7 Results

In our dynamic range compression algorithm, we have exploited the fact that humans are insensitive to relative and absolute differences in luminance. For example, we can see that it is brighter outside than inside on a sunny day, but we cannot tell how much brighter (3 times or 100) or what the actual luminances are (10 cd/m^2 or 1000). With the additional display range made available by adjusting the histogram to close the gaps between luminance levels, visibility (i.e., contrast) within each level can be properly preserved. Furthermore, this is done in a way that is compatible with subjective aspects of vision.

In the development sections, two synthetic scenes have served as examples. In this section, we show results from two different application areas -- lighting simulation and electronic photography.

Figure 19. A simulation of a shipboard control panel under emergency lighting.

Figure 20. A simulation of an air traffic control console.

Figure 21. A Christmas tree with very small light sources.

7.1 Lighting Simulation

In lighting design, it is important to simulate what it is like to *be* in an environment, not what a photograph of the environment looks like. Figures 19 and 20 show examples of real lighting design applications.

In Figure 19, the emergency lighting of a control panel is shown. It is critical that the lighting provide adequate visibility of signage and levers. An image synthesis method that cannot predict human visibility is useless for making lighting or system design judgments.

Figure 20 shows a flight controller's console. Being able to switch back and forth between the console and the outdoor view is an essential part of the controller's job. Again, judgments on the design of the console cannot be made on the basis of ill-exposed or arbitrarily mapped images.

Figure 21 is not a real lighting application, but represents another type of interesting lighting. In this case, the high dynamic range is not represented by large areas of either high or low luminance. Very high, almost point, luminances are scattered in the scene. The new tone mapping works equally well on this type of lighting, preserving visibility

while keeping the impression of the brightness of the point sources. The color sensitivity and variable acuity mapping also correctly represent the sharp color view of areas surrounding the lights, and the greyed blurring of more dimly lit areas.

Figure 22. A scanned photograph of Memorial Church.

7.2 Electronic Photography

Finally, we present an example from electronic photography. In traditional photography, it is impossible to set the exposure so all areas of a scene are visible as they would be to a human observer. New techniques of digital compositing are now capable of creating images with much higher dynamic ranges. Our tone reproduction operator can be applied to appropriately map these images into the range of a display device.

Figure 22 shows the interior of a church, taken on print film by a 35mm SLR camera with a 15mm fisheye lens. The stained glass windows are not completely visible because the recording film has been saturated, even though the rafters on the right are too dark to see. Figure 23 shows our tone reproduction operator applied to a high dynamic range version of this image, called a *radiance map*. The radiance map was generated from 16 separate exposures, each separated by one stop. These images were scanned, registered, and the full dynamic range was recovered using an algorithm developed by Debevec and Malik

[Debevec97]. Our tone mapping operator makes it possible to retain the image features shown in Figure 23, whose world luminances span over 6 orders of magnitude.

The field of electronic photography is still in its infancy. Manufacturers are rapidly improving the dynamic range of sensors and other electronics that are available at a reasonable cost. Visibility preserving tone reproduction operators will be essential in accurately displaying the output of such sensors in print and on common video devices.

Figure 23. Histogram adjusted radiance map of Memorial Church.

8 Conclusions and Future Work

There are still several degrees of freedom possible in this tone mapping operator. For example, the method of computing the foveal samples corresponding to viewer fixation points could be altered. This would depend on factors such as whether an interactive system or a preplanned animation is being designed. Even in a still image, a theory of probable gaze could be applied to improve the initial adaptation histogram. Additional modifications could easily be made to the threshold sensitivity, veil and acuity models to simulate the effects of aging and certain types of visual impairment.

This method could also be extended to other application areas. The tone mapping could be incorporated into global illumination calculations to make them more efficient by

relating error to visibility. The mapping could also become part of a metric to compare images and validate simulations, since the results correspond roughly to human perception [Rushmeier95].

Some of the approximations in our operator merit further study, such as color sensitivity changes in the mesopic range. A simple choice was made to interpolate linearly between scotopic and photopic response functions, which follows Ferwerda et al. [Ferwerda96] but should be examined more closely. The effect of the luminous surround on adaptation should also be considered, especially for projection systems in darkened rooms. Finally, the current method pays little attention to absolute color perception, which is strongly affected by global adaptation and source color (i.e., white balance).

The examples and results we have shown match well with the subjective impression of viewing the actual environments being simulated or recorded. On this informal level, our tone mapping operator has been validated experimentally. To improve upon this, more rigorous validations are needed. While validations of image synthesis techniques have been performed before (e.g., Meyer et al. [Meyer86]), they have not dealt with the level of detail required for validating an accurate tone operator. Validation experiments will require building a stable, non-trivial, high dynamic range environment and introducing observers to the environment in a controlled way. Reliable, calibrated methods are needed to capture the actual radiances in the scene and reproduce them on a display following the tone mapping process. Finally, a series of unbiased questions must be formulated to evaluate the subjective correspondence between observation of the physical scene and observation of images of the scene in various media. While such experiments will be a significant undertaking, the level of sophistication in image synthesis and electronic photography requires such detailed validation work.

The dynamic range of an interactive display system is limited by the technology required to control continual, intense, focused energy over millisecond time frames, and by the uncontrollable elements in the ambient viewing environment. The technological, economic and practical barriers to display improvement are formidable. Meanwhile, luminance simulation and acquisition systems continue to improve, providing images with higher dynamic range and greater content, and we need to communicate this content on conventional displays and hard copy. This is what tone mapping is all about.

9 Acknowldgments

The authors wish to thank Robert Clear and Samuel Berman for their helpful discussions and comments. This work was supported by the Laboratory Directed Research and Development Funds of Lawrence Berkeley National Laboratory under the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

10 References

[Chiu93]

K. Chiu, M. Herf, P. Shirley, S. Swamy, C. Wang and K. Zimmerman "Spatially nonuniform scaling functions for high contrast images," *Proceedings of Graphics Interface '93,* Toronto, Canada, May 1993, pp. 245-253.

[CIE81]
CIE (1981) *An analytical model for describing the influence of lighting parameters upon visual performance*, vol 1. Technical foundations. CIE 19/2.1, Techical committee 3.1

[Debevec97]
Debevec, Paul and Jitendra Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," *Proceedings of ACM SIGGRAPH '97*.

[Ferwerda96]
J. Ferwerda, S. Pattanaik, P. Shirley and D.P. Greenberg. "A Model of Visual Adaptation for Realistic Image Synthesis," *Proceedings of ACM SIGGRAPH '96*, p. 249-258.

[Frei77]
W. Frei, "Image Enhancement by Histogram Hyperbolization," *Computer Graphics and Image Processing*, Vol 6, 1977 286-294.

[Glassner95]
A. Glassner, *Principles of Digital Image Synthesis*, Morgan Kaufman, San Francisco, 1995.

[Green83]
W. Green *Digital Image Processing: A Systems Approach*, Van Nostrand Reinhold Company, NY, 1983.

[Hall89]
R. Hall, *Illumination and Color in Computer Generated Imagery*, Springer-Verlag, New York, 1989.

[Holladay26]
Holladay, L.L., *Journal of the Optical Society of America*, 12, 271 (1926).

[Meyer86]
G. Meyer, H. Rushmeier, M. Cohen, D. Greenberg and K. Torrance. "An Experimental Evaluation of Computer Graphics Imagery," *ACM Transactions on Graphics*, January 1986, Vol. 5, No. 1, pp. 30-50.

[Mokrane92]
A. Mokrane, "A New Image Contrast Enhancement Technique Based on a Contrast Discrimination Model," *CVGIP: Graphical Models and Image Processing*, 54(2) March 1992, pp. 171-180.

[Moon&Spencer45]
P. Moon and D. Spencer, "The Visual Effect of Non-Uniform Surrounds", *Journal of the Optical Society of America*, vol. 35, No. 3, pp. 233-248 (1945)

[Nakamae90]
E. Nakamae, K. Kaneda, T. Okamoto, and T. Nishita. "A lighting model aiming at drive simulators," *Proceedings of ACM SIGGRAPH 90*, 24(3):395-404, June, 1990.

[Rushmeier95]
H. Rushmeier, G. Ward, C. Piatko, P. Sanders, B. Rust, "Comparing Real and Synthetic Images: Some Ideas about Metrics," *Sixth Eurographics Workshop on Rendering*, proceedings published by Springer-Verlag. Dublin, Ireland, June 1995.

[Schlick95]
C. Schlick, "Quantization Techniques for Visualization of High Dynamic Range Pictures," *Photorealistic Rendering Techniques* (G. Sakas, P. Shirley and S. Mueller, Eds.), Springer, Berlin, 1995, pp.7-20.

[Spencer95]
G. Spencer, P. Shirley, K. Zimmerman, and D. Greenberg, "Physically-based glare effects for computer generated images," *Proceedings ACM SIGGRAPH '95*, pp. 325-334.

[Stevens60]
S. S. Stevens and J.C. Stevens, "Brightness Function: Parametric Effects of adaptation and contrast," *Journal of the Optical Society of America*, 53, 1139. 1960.

[Tumbline93]
J. Tumblin and H. Rushmeier. "Tone Reproduction for Realistic Images," *IEEE Computer Graphics and Applications*, November 1993, 13(6), 42-48.

[Ward91]
G. Ward, "A contrast-based scalefactor for luminance display," In P.S. Heckbert (Ed.) *Graphics Gems IV*, Boston, Academic Press Professional.

[Williams83]
L. Williams, "Pyramidal Parametrics," *Computer Graphics*, v.17,n.3, July 1983.

High Dynamic Range Imaging

Greg Ward
Exponent – Failure Analysis Assoc.
Menlo Park, California

Abstract

The ultimate in color reproduction is a display that can produce arbitrary spectral content over a 300-800 nm range with 1 arc-minute resolution in a full spherical hologram. Although such displays will not be available until next year, we already have the means to calculate this information using physically-based rendering. We would therefore like to know: how may we represent the results of our calculation in a device-independent way, and how do we map this information onto the displays we currently own? In this paper, we give an example of how to calculate full spectral radiance at a point and convert it to a reasonably correct display color. We contrast this with the way computer graphics is usually done, and show where reproduction errors creep in. We then go on to explain reasonable short-cuts that save time and storage space without sacrificing accuracy, such as illuminant discounting and human gamut color encodings. Finally, we demonstrate a simple and efficient tone-mapping technique that matches display visibility to the original scene.

Introduction

Most computer graphics software works in a 24-bit RGB space, with 8-bits allotted to each of the three primaries in a power-law encoding. The advantage of this representation is that no tone-mapping is required to obtain a reasonable reproduction on most commercial CRT display monitors, especially if both the monitor and the software adhere to the sRGB standard, i.e., CCIR-709 primaries and a 2.2 gamma [1]. The disadvantage of this practice is that colors outside the sRGB gamut cannot be represented, particularly values that are either too dark or too bright, since the useful dynamic range is only about 90:1, less than 2 orders of magnitude. By contrast, human observers can readily perceive detail in scenes that span 4-5 orders of magnitude in luminance through local adaptation, and can adapt in minutes to over 9 orders of magnitude. Furthermore, the sRGB gamut only covers about half the perceivable colors, missing large regions of blue-greens and violets, among others. Therefore, although 24-bit RGB does a reasonable job of representing what a CRT monitor can display, it does a poor job representing what a human observer can see.

Display technology is evolving rapidly. Flat-screen LCD displays are starting to replace CRT monitors in many offices, and LED displays are just a few years off. Micromirror projection systems with their superior dynamic range and color gamut are already widespread, and laser raster projectors are on the horizon. It is an important question whether we will be able to take full advantage and adapt our color models to these new devices, or will we be limited as we are now to remapping sRGB to the new gamuts we have available -- or worse, getting the colors wrong? Unless we introduce new color models to our image *sources* and do it soon, we will never get out of the CRT color cube.

The simplest solution to the gamut problem is to adhere to a floating-point color space. As long as we permit values greater than one and less than zero, any set of color primaries may be linearly transformed into any other set of color primaries without loss. The principal disadvantage of most floating-point representations is that they take up too much space (96-bits/pixel as opposed to 24). Although this may be the best representation for color computations, storing this information to disk or transferring it over the internet is a problem. Fortunately, there are representations based on human perception that are compact and sufficiently accurate to reproduce any visible color in 32-bits/pixel or less, and we will discuss some of these in this paper.

There are two principal methods for generating high dynamic-range source imagery: physically-based rendering (e.g., [2]), and multiple-exposure image capture (e.g., [3]). In this paper, we will focus on the first method, since it is most familiar to the author. It is our hope that in the future, camera manufacturers will build HDR imaging principles and techniques into their cameras, but for now, the easiest path to full gamut imagery seems to be computer graphics rendering.

Computer graphics lifts the usual constraints associated with physical measurements, making floating-point color the most natural medium in which to work. If a renderer is physically-based, it will compute color values that correspond to spectral radiance at each point in the rendered image. These values may later be converted to displayable colors, and the how and wherefore of this tone-mapping operation is the main topic of this paper. Before we get to tone-mapping, however, we must go over some of the details of physically-based rendering, and what qualifies a renderer in this category. Specifically, we will detail the basic lighting calculation, and compare this to common practice in computer graphics rendering. We highlight some common assumptions and approximations, and describe alternatives when these assumptions fail. Finally, we demonstrate color and tone mapping methods for converting the computed spectral radiance value to a displayable color at each pixel.

The Spectral Rendering Equation

$$R_o(\omega_o,\lambda) = \iint f_r(\omega_o;\omega_i,\lambda)\, R_i(\omega_i,\lambda)\cos\theta_i \, d\omega_i \quad (1)$$

The spectral rendering Eq. (1) expresses outgoing spectral radiance R_o at a point on a surface in the direction ω_o (θ_o,ϕ_o) as a convolution of the bidirectional reflectance distribution function (BRDF) with the incoming spectral radiance over the projected hemisphere. This equation is the basis of many physically-based rendering programs, and it already contains a number of assumptions:

1. Light is reflected at the same wavelength at which it is received; i.e., the surface is not fluorescent.
2. Light is reflected at the same position at which it is received; i.e., there is no subsurface scattering.
3. Surface transmission is zero.
4. There are no polarization effects.
5. There is no diffraction.
6. The surface does not spontaneously emit light.

In general, these assumptions are often wrong. Starting with the first assumption, many modern materials such as fabrics, paints, and even detergents, contain "whitening agents" which are essentially phosphors added to absorb ultraviolet rays and re-emit them at visible wavelengths. The second assumption is violated by many natural and man-made surfaces, such as marble, skin, and vinyl. The third assumption works for opaque surfaces, but fails for transparent and thin, translucent objects. The fourth assumption fails for any surface with a specular (shiny) component, and becomes particularly troublesome when skylight (which is strongly polarized) or multiple reflections are involved. The fifth assumption fails when surface features are on the order of the wavelength of visible light, and the sixth assumption is violated for light sources.

Each of these assumptions may be addressed and remedied as necessary. Since a more general rendering equation would require a long and tedious explanation, we merely describe what to add to account for the effects listed. To handle fluorescence, the outgoing radiance at wavelength λ_o may be computed from an integral of incoming radiance over all wavelengths λ_I, which may be discretized in a matrix form [4]. To handle subsurface scattering, we can integrate over the surface as well as incoming directions, or use an approximation [5]. To handle transmission, we simply integrate over the sphere instead of the hemisphere, and take the absolute value of the cosine for the projected area [2]. To account for polarization, we add two terms for the transverse and parallel polarizations in each specular direction [4] [6]. To handle diffraction, we fold interactions between wavelength, polarization, amplitude and direction into the BRDF and the aforementioned extensions [7]. Light sources are the simplest exception to handle – we simply add in the appropriate amount of spontaneous radiance output as a function of direction and wavelength.

Participating Media

Implicitly missing from Eq. (1) is the interaction of light with the atmosphere, or participating media. If the space between surfaces contains significant amounts of dust, smoke, or condensation, a photon leaving one surface may be scattered or absorbed along the way. An additional equation is therefore needed to describe this volumetric effect, since the rendering equation only addresses interactions at surfaces.

$$\frac{dR(s)}{ds} = -\sigma_a R(s) - \sigma_s R(s) + \frac{\sigma_s}{4\pi}\int R_i(\theta_i) P(\theta_i)\, d\omega \quad (2)$$

Eq. (2) gives the differential change in radiance as a function of distance along a path. The coefficients σ_a and σ_s give the absorption and scattering densities respectively at position s, which correspond to the probabilities that light will be absorbed or scattered per unit of distance traveled. The scattering phase function, $P(\theta_i)$, gives the relative probability that a ray will be scattered in from direction θ_i at this position. All of these functions and coefficients are also a function of wavelength.

The above differential-integral equation is usually solved numerically by stepping through each position along the path, starting with the radiance leaving a surface given by Eq. (1). Recursive iteration from a sphere of scattered directions can quickly overwhelm such a calculation, especially if it is extended to multiple scattering events. Without going into details, Rushmeier et al. approached the problem of globally participating media using a zonal approach akin to radiosity that divides the scene into a finite set of voxels whose interactions are characterized in a form-factor matrix [8]. More recently, a modified ray-tracing method called the photon map has been applied successfully to this problem by Wann Jensen et al. [9]. In this method, photons are tracked as they scatter and are stored in the environment for later resampling during rendering .

Solving the Rendering Equation

Eq. (1) is a Fredholm integral equation of the second kind, which comes close to the appropriate level of intimidation but fails to explain why it is so difficult to solve in general [10]. Essentially, the equation defines outgoing radiance as an integral of incoming radiance at a surface point, and that incoming radiance is in turn defined by the same integral with different parameters evaluated at another surface point. Thus, the surface geometry and material functions comprise the boundary conditions of an infinitely recursive system of integral equations. In some sense, it is remarkable that researchers have made any progress in this area at all, but in fact, there are many people in computer graphics who believe that rendering is a solved problem.

For over fifteen years, three approaches have dominated research and practice in rendering. The first approach is usually referred to as the *local illumination* approximation, and is the basis for most graphics rendering hardware, and

much of what you see in movies and games. In this approximation, the integral equation is converted into a simple sum over light sources (i.e., concentrated emitters) and a general ambient term. The second approach is called *ray tracing*, and as its name implies, this method traces additional rays to determine specular reflection and transmission, and may be used to account for more general interreflections as well [11] [12]. The third approach is called *radiosity* after the identical method used in radiative transfer, where reflectances are approximated as Lambertian and the surfaces are divided into patches to convert the integral equation into a large linear system that may be solved iteratively [13]. Comparing these three approaches, local illumination is the cheapest and least accurate. Ray tracing has the advantage of coping well with complex geometry and materials, and radiosity does the best job of computing global interactions in simpler, diffuse environments.

In truth, none of the methods currently in use provides a complete and accurate solution to the rendering equation for general environments, though some come closer than others. The first thing to recognize in computer graphics, and computer simulation in general, is that the key to getting a reasonable answer is finding the right approximation. The reason that local illumination is so widely employed when there are better techniques available is not simply that it's cheaper; it provides a reasonable approximation to much of what we see. With a few added tricks, such as shadow maps, reflection maps and ambient lights, local illumination in the hands of an expert does a very credible job. However, this is not to say that the results are correct or accurate. Even in perceptual terms, the colors produced at each pixel are usually quite different from those one would observe in a real environment. In the entertainment industry, this may not be a concern, but if the application is prediction or virtual reenactment, better accuracy is necessary.

For the remainder of this paper, we assume that accuracy is an important goal, particularly color accuracy. We therefore restrict our discussion of rendering and display to physically-based global illumination methods, such as ray-tracing and radiosity.

Tone Mapping

By computing an approximate solution to Eq. (1) for a given planar projection, we obtain a spectral rendering that represents each image point in physical units of radiance per wavelength (e.g., SI units of watts/steradian/meter^2/nm). Whether we arrive at this result by ray-tracing, radiosity, or some combination, the next important task is to convert the spectral radiances to pixel color values for display. If we fail to take this step seriously, it almost doesn't matter how much effort we put into the rendering calculation – the displayed image will look wrong.

Converting a spectral image to a display image is usually accomplished in two stages. The first stage is to convert the spectral radiances to a tristimulus space, such as CIE XYZ. This is done by convolving each radiance spectrum with the three standard CIE observer functions. The second stage is to map each tristimulus value into our target display's color space. This process is called *tone-mapping*, and depending on our goals and requirements, we may take different approaches to arrive at different results. Here are a few possible *rendering intents*:

1. Colorimetric intent: Attempt to reproduce the exact color on the display, ignoring viewer adaptation.[1]
2. Saturation intent: Maintain color saturation as far as possible, allowing hue to drift.
3. Perceptual intent: Attempt to match perception of color by remapping to display gamut and viewer adaptation.

The rendering intents listed above have been put forth by the ICC profile committee, and their exact meaning is somewhat open to interpretation, especially for out-of-gamut colors. Even for in-gamut colors, the perceptual intent, which interests us most, may be approached in several different ways. Here are a few possible techniques:

A. Shrink the source (visible) gamut to fit within the display gamut, scaling uniformly about the neutral line.
B. Same as A, except apply relative scaling so less saturated colors are affected less than more saturated ones. The extreme form of this is gamut-clipping.
C. Scale colors on a curve determined by image content, as in a global histogram adjustment.
D. Scale colors locally based on image spatial content, as in Land's retinex theory.

To any of the above, we may also add a white point transformation and/or contrast adjustment to compensate for a darker or brighter surround. In general, it is impossible to reproduce exactly the desired observer stimulus unless the source image contains no bright or saturated colors or the display has an unusually wide gamut and dynamic range.[2]

Before we can explore any gamut-mapping techniques, we need to know how to get from a spectral radiance value to a tristimulus color such as XYZ or RGB. The calculation is actually straightforward, but the literature on this topic is vast and confusing, so we give an explicit example to make sure we get it right.

Correct Color Rendering

Looking at the simplest case, spectral reflection of a small light source from a diffuse surface in Eq. (1) reduces to the following formula for outgoing radiance:

$$R_o(\lambda) = \frac{\rho_d(\lambda)}{\pi} E_i(\lambda) \qquad (3)$$

[1] The ICC Colorimetric intent is actually divided into *relative* and *absolute* intents, but this distinction is irrelevant to our discussion.

[2] See www.hitl.washington.edu/research/vrd/ for information on Virtual Retinal Display technology.

where $\rho_d(\lambda)$ is the diffuse reflectance as a function of wavelength, and $E_i(\lambda)$ is the spectral irradiance computed by integrating radiance over the projected source. To convert this to an absolute XYZ color, we apply the standard CIE conversion, given below for SI units [16]:

$$X = 683 \int \bar{x}(\lambda) R(\lambda) d\lambda$$
$$Y = 683 \int \bar{y}(\lambda) R(\lambda) d\lambda$$
$$Z = 683 \int \bar{z}(\lambda) R(\lambda) d\lambda \qquad (4)$$

At this point, we may wish to convert to an opponent color space for gamut-mapping, or we may wait until we are in the device color space. If our tone-mapping is a simple scale factor as described in technique A above, we may apply it in any linear color space and the results will be the same. If we convert first to a nonlinear device color space, we need to be aware of the meaning of out-of-gamut colors in that space before we map them back into the legal range of display values. We demonstrate a consistent and reasonable method, then compare to what is usually done in computer graphics.

BlueFlower Example

To compute the absolute CIE color for a surface point, we need to know the spectra of the source and the material. Fig. 1 shows the source spectra for standard illuminant A (2856K tungsten), illuminant B (simulated sunlight), and illuminant D65 (6500K daylight). Fig. 2 shows the reflected spectral radiance of the BlueFlower patch from the MacBeth chart under each of these illuminants. To these curves, we apply the CIE standard observer functions using Eq. (4).

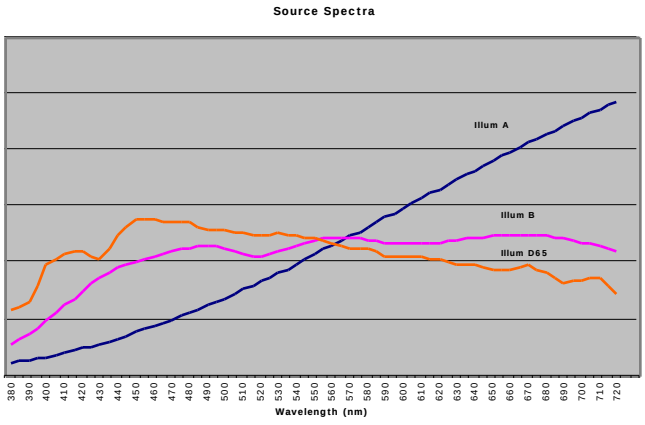


Figure 1. Spectral power of three standard illuminants.

The resulting XYZ values for the three source conditions is given in the first row of Table 1. Not surprisingly, there is a large deviation in color under different illuminants, especially tungsten. We can convert these colors to their RGB equivalents using Eq. (5), as given in the second row of Table 1. If we were to directly display the colors from the illuminant A and B conditions on the screen, they would likely appear incorrect because the viewer would be adapted to the white point of the monitor rather than the white point of the original scenes being rendered. If we assume the scene white point is the same color as the illuminant and the display white point is D65, then a white point adjustment is necessary for the other illuminants (A and B), as given in the third row of Table 1.

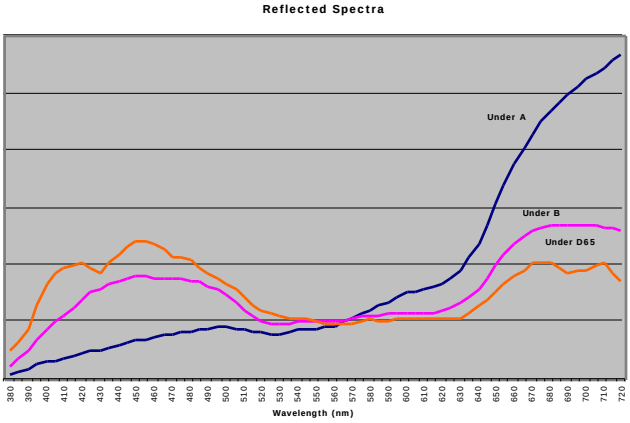


Figure 2. Spectral radiance of MacBeth BlueFlower patch under three standard illuminants.

Source CIE (x,y)	Illum D65 (.3127,.3290)	Illum B (.3484,.3516)	Illum A (.4475,.4075)
BlueFlower CIE XYZ	0.274 0.248 0.456	0.280 0.248 0.356	0.302 0.248 0.145
709 RGB (absolute)	0.279 0.219 0.447	0.349 0.209 0.341	0.525 0.179 0.119
709 RGB (adjusted)	0.279 0.219 0.447	0.285 0.218 0.444	0.306 0.215 0.426

Table 1. Computed color values for BlueFlower under three standard illuminants.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \mathbf{C} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$
$$\mathbf{C}_{709} = \begin{bmatrix} 3.2410 & -1.5374 & -0.4986 \\ -0.9692 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0570 \end{bmatrix} \qquad (5)$$

We use a linear transform to adjust the white point from that of the illuminant to that of the display, which we assume to be D65 in this example. Eq. (5) gives the absolute transformation from XYZ to CCIR-709 linear RGB, and

this is all we need for the D65 illuminant condition. For the others, we apply the transformation shown in Eq. (6).

Eq. (6) is the linear von Kries adaptation model with the CMCCAT2000 primary matrix [14], which does a reasonable job of accounting for chromatic adaptation when shifting from one dominant illuminant to another [15]. The original white point primaries (R_w,G_w,B_w) are computed from the illuminant XYZ using the $\mathbf{M}_{\text{CMCCAT}}$ matrix, and the destination primaries (R_w',G_w',B_w') for D65 are computed using the same transform to be (0.9478,1.0334,1.0850).

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} R_w'/R_w & 0 & 0 \\ 0 & G_w'/G_w & 0 \\ 0 & 0 & B_w'/B_w \end{bmatrix} \mathbf{M} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\begin{bmatrix} R_w \\ G_w \\ B_w \end{bmatrix} = \mathbf{M} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

$$\mathbf{M}_{CMCCAT} = \begin{bmatrix} 0.7982 & 0.3389 & -0.1371 \\ -0.5918 & 1.5512 & 0.0406 \\ 0.0008 & 0.0239 & 0.9753 \end{bmatrix} \quad (6)$$

The combined matrices for a white shift from standard illuminants B and A to D65 (whose chromaticities are given at the top of Table 1) and subsequent conversion from CIE XYZ to CCIR-709 RGB color space, are given in Eq.(7) as $\mathbf{C}_B$ and $\mathbf{C}_A$. Matrix $\mathbf{C}_{709}$ from Eq. (5) was concatenated with the matrix terms in Eq. (6) to arrive at these results, which may be substituted for $\mathbf{C}_{709}$ in Eq. (5) to get the adjusted RGB colors in the third row of Table 1 from the absolute XYZ values in the first row.

$$\mathbf{C}_B = \begin{bmatrix} 3.1273 & -1.6836 & -0.4867 \\ -0.9806 & 1.9476 & 0.0282 \\ 0.0605 & -0.2036 & 1.3404 \end{bmatrix}$$

$$\mathbf{C}_A = \begin{bmatrix} 2.9355 & -2.0416 & -0.5116 \\ -1.0247 & 2.1431 & -.0500 \\ 0.0732 & -0.1798 & 3.0895 \end{bmatrix} \quad (7)$$

Conventional CG Calculation

The standard approach in computer graphics color calculations is to assume all light sources are perfectly white and perform calculations in RGB color space. To display the results, a linear scale factor may be applied to bring the results into some reasonable range, and any values outside the sRGB gamut will be clamped.

We obtain an RGB value for the BlueFlower material from its published (x,y) chromaticity of (0.265,0.240) and reflectance of 24.3%. These published values correspond to viewing under standard illuminant C (simulated overcast), which is slightly bluer than D65. The linear RGB color for the flower material using the matrix $\mathbf{C}_{709}$ from Eq. (5) is (0.246,0.217,0.495), which differs from the D65 results in Table 1 by 10 ΔE* units using the CIE L*uv perceptual metric [16]. Most of this difference is due to the incorrect scene illuminant assumption, since the ΔE* between illuminant C and D65 is also around 10. This demonstrates the inherent sensitivity of color calculations to source color. Using the color corresponding to the correct illuminant is therefore very important.

The reason CG lighters usually treat sources as white is to avoid the whole white balancing issue. As evident from the third row in Table 1, careful accounting of the light source and chromatic adaptation is almost a no-op in the end. For white points close to the viewing condition of D65, the difference is small: a difference of just 1 ΔE* for illuminant B. However, tungsten is very far from daylight, and the ΔE* for illuminant A is more than 5, which is definitely visible. Clearly, if we include the source spectrum, we need to include chromatic adaptation in our tone-mapping. Otherwise, the differences will be very visible indeed -- a ΔE* of 22 for illuminant B and nearly 80 for illuminant A!

What if we include the source color, but use an RGB approximation instead of the full spectral rendering? Errors will creep in from the reduced spectral resolution, and their significance will depend on the source and reflectance spectra. Computing everything in CCIR-709 RGB for our BlueFlower example, the ΔE* from the correct result is 1 for illuminant B and nearly 8 for illuminant A. These errors are at least as large as ignoring the source color entirely, so there seems to be little benefit in this approach.

Relative Color Approximation

An improved method that works well for scenes with a single dominant illuminant is to compute the absolute RGB color of each material under the illuminant using a spectral precalculation from Eqs. (3) and (4). The source itself is modeled as pure white (Y,Y,Y) in the scene, and sources with a different color are modeled relative to this illuminant as (*Rs/Rw*,*Gs/Gw*,*Bs/Bw*), where (*Rw*,*Gw*,*Bw*) is the RGB value of the dominant illuminant and (*Rs*,*Gs*,*Bs*) is the color of the other source. In our example, the RGB color of the BlueFlower material under the three standard illuminants are those given in the second row of Table 1.

Prior to display, the von Kries chromatic adaptation in Eq. (5) is applied to the image pixels using the dominant source and display illuminants. The incremental cost of our approximation is therefore a single transform on top of the conventional CG rendering, and the error is zero by construction for direct reflection from a single source type. There may be errors associated with sources having different colors and multiple reflections, but these will be negligible in most scenes. Best of all, no software change is required – we need only precalculate the correct RGB values for our sources and surfaces, and the rest comes for free.

It is even possible to save the cost of the final von Kries transform by incorporating it into the precalculation,

computing adjusted rather than absolute RGB values for the materials, as in Eq. (7). We would prefer to keep this transform separate to preserve the colorimetric nature of the rendered image, but as a practical matter, it is often necessary to record a white-balanced image, anyway. As long as we record the scene white point in an image format that preserves the full gamut and dynamic range of our tristimulus pixels, we insure our ability to correctly display the rendering in any device's color space, now and in the future.

High Dynamic Range Images

Real scenes and physically-based renderings of real scenes do not generally fit within a conventional display's gamut using any reasonable exposure value (i.e., scale factor). If we compress or remap the colors to fit an sRGB or similar gamut, we lose the ability to later adjust the tone-scale or show off the image on a device with a larger gamut or wider dynamic range. What we need is a truly device-independent image representation, which doesn't take up too much space, and delivers superior image quality whatever the destination. Fortunately, such formats exist.

Since its inception in 1985, the *Radiance* physically-based renderer has employed a 32-bit/pixel RGBE (Red-Green-Blue-Exponent) format to store its high dynamic range output [17]. Predating *Radiance*, Bill Reeves of **Pixar** created a 33-bit log RGB format for the REYES rendering system, and this format has a public version contributed by Dan McCoy in 1996 to Sam Leffler's free TIFF library (www.libtiff.org). While working at **SGI**, the author added to the same TIFF library a LogLuv format that captures 5 orders of magnitude and the full visible gamut in 24 bits using a perceptual color encoding [18]. The 32-bit version of this format holds up to 38 orders of magnitude, and often results in smaller files due to run-length encoding [19]. Both LogLuv formats combine a logarithmic encoding of luminance with a linear encoding of CIE (u',v') chromaticity to cover the full visible gamut as opposed to the gamut of a specific device or medium.

Of the formats mentioned, only SGI's LogLuv TIFF encoding covers the full gamut and dynamic range of perceivable colors. The *Radiance* RGBE format spans a large dynamic range but is restricted to positive RGB values, so there are visible chromaticities it cannot represent. There is an XYZE version of the same format, but the associated quantization errors make it a poor choice. The Pixar 33-bit log format also has a restricted RGB gamut and only covers 3.8 orders of magnitude, which is marginal for human perception. Since the TIFF library is well tested and free, there is really no reason not to use LogLuv, and many rendering packages now output in this format. Even shareware browsers such as *ACDSee* are able to read and display LogLuv TIFF's.

Gamut Mapping

In order to fit a high dynamic range image into the limited color space of a conventional display, we need to apply one of the gamut compression techniques mentioned at the beginning of this section.

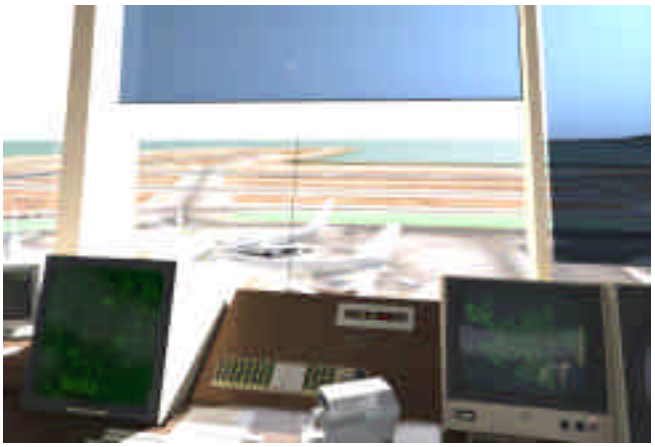


Figure 3. Radiance rendering of control tower clamped to limited display gamut and dynamic range.

Figure 4. The same rendering displayed using a visibility-preserving tone operator including glare effects.

Figure 5. A tone operator designed to optimize print contrast.

Specifically, we show how one might apply the third approach to display an image:

C. Scale colors on a curve determined by image content, as in a global histogram adjustment.

We assume that the rendering system has calculated the correct color at each pixel and stored the result in a high dynamic-range image format. Our task is then to examine this image and choose an appropriate mapping to our display. This is a difficult process to automate, and there is no guarantee we will achieve a satisfactory result in all cases. The best we can do is codify a specific set of goals and requirements and optimize our tone-mapping accordingly.

One possible goal of physically-based rendering is to assess visibility in some hypothetical environment or situation, or to recreate a situation that is no longer readily available (e.g., a plane crash). In such cases, we want to say that anything visible to an observer in the actual scene will be visible on the tone-mapped display. Conversely, if something is not visible on the display, we want to say that it would not be visible to an observer in the actual scene. This kind of visibility-matching operator was described in [20], and we show the result in Fig. 4. Fig. 3 shows the image mapped to an sRGB gamut using technique B to desaturate out-of-gamut colors. As we can see, some of the detail in the planes outside the window was lost to clamping, where it is preserved in the visibility-matching histogram-adjustment procedure in Fig. 4. An optional feature of our tone operator is the ability to simulate disability glare, which reduces visible contrast due to the harsh backlighting in the tower environment. This is visible as a slight haze in front of the monitors in Fig. 4.

Fig. 5 demonstrates another type of tone operator. This is also a histogram adjustment method, but instead of attempting to reproduce visibility, this operator seeks to optimize contrast over the entire image while keeping colors within the printable gamut. Especially in digital photo printers, saturated colors may be difficult to reproduce, so it may be desirable to darken an image to avoid desaturating some regions. We see that this method produces good contrast over most of the image.

Fig. 6 shows the global mapping of these three operators from world (rendered) luminance to display value (fraction of maximum). Where the naive linear operator clamps a lot of information off the top end, the two histogram adjustment operators present this information at a reduced contrast. This compression is necessary in order to bring out detail in the darker regions. We can see that the slopes match the linear operator near black in Fig. 7, deviating from the linear clamping operator above a certain level, where compression begins.

Fig. 8 plots the contrast optimizing tone operator against the world luminance distribution. Peaks in the luminance histogram correspond to increases in contrast, visible in the tone-mapping as a slight increase in slope. Since this is a log-log luminance plot, a small change in slope corresponds to a large change in contrast. The dip between 1.5 and 2.0 corresponds to a more gradual slope in the tone-mapping and lower contrast. In the low end, we see that this operator tends to provide more contrast to compensate for veiling reflection typical of glossy prints.

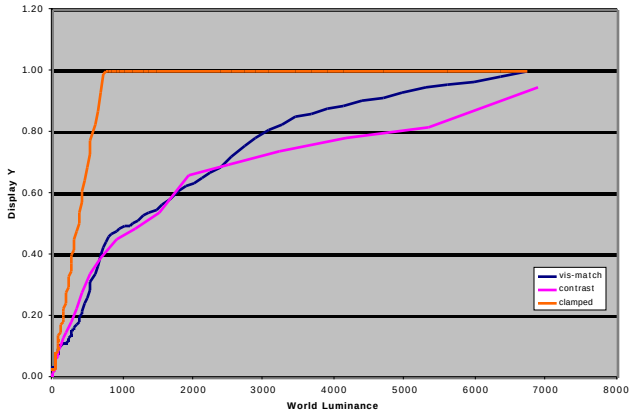


Figure 6. Comparison between three tone-mapping operators.

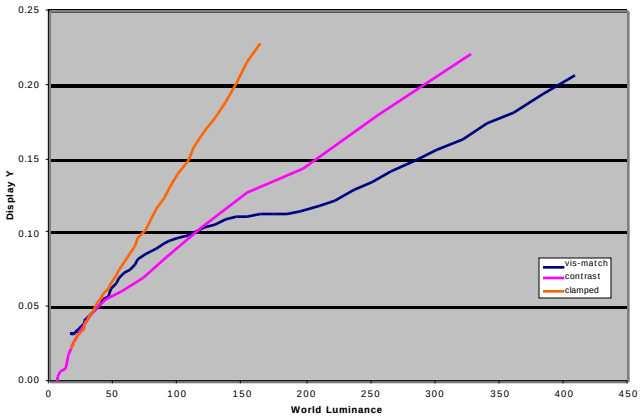


Figure 7. Close-up on darker region of tone-mappings.

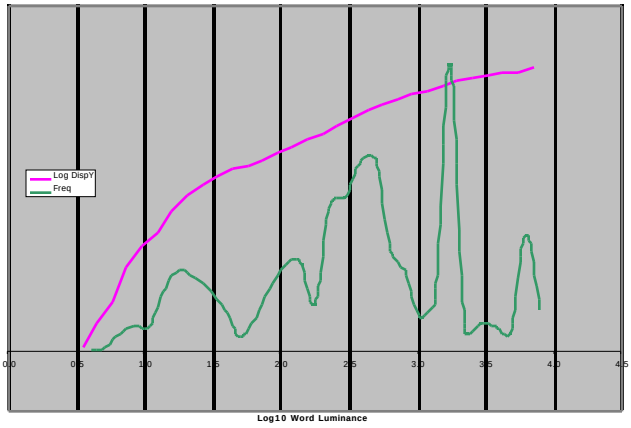


Figure 8. Good global tone operators produce greater contrast at peaks in the input histogram.

Conclusion

The recommendations we make in this paper for accurate color rendering may be summarized as follows:

1. Use a global illumination method with appropriate solutions for all of the phenomena being simulated.
2. Follow accurate spectral calculations with a good chromatic adaptation model to avoid color casts in the displayed image.
3. Substitute full spectral rendering with a relative color approximation for scenes with a single dominant illuminant.
4. Record images in a high dynamic range format to preserve display options (i.e., SGI LogLuv TIFF).
5. Base tone-mapping and gamut-mapping operators on specific goals, such as matching visibility or optimizing color or contrast.

Floating-point spectral calculations and high dynamic-range image manipulation are critical to accurate color rendering. The original approach of rendering directly in 24-bit RGB was recognized as hopeless and abandoned decades ago, but much of the mentality behind it remains with us today.

The methods outlined in this paper are not particularly expensive, neither in terms of implementation effort nor rendering cost. It's simply a matter of applying the right approximation. The author is not aware of any commercial software package that follows more than one or two of these principles, and it seems like a question of priorities.

Most of the money in rendering is spent by the entertainment industry, either in movies or in games. Little emphasis has been placed on accurate color rendering, but with the recent increase in mixed-reality rendering, this is beginning to change. Mixed-reality special effects and games require rendered imagery to blend seamlessly with film or live footage. Since reality follows physics and color science, rendering software will have to do likewise. Those of us whose livelihood depends on predictive rendering and accurate color stand to benefit from this shift.

References

1. Michael Stokes, Matthew Anderson, Srinivasan Chandrasekar, Ricardo Motta, A Standard Default Color Space for the Internet, www.w3.org/Graphics/Color/sRGB
2. Greg Ward, The RADIANCE Lighting Simulation and Rendering System, *Computer Graphics (Proceedings of SIGGRAPH 94)*, ACM, 1994.
3. Paul Debevec, Jitendra Malik, Recovering High Dynamic Range Radiance Maps from Photographs, *Computer Graphics (Proceedings of SIGGRAPH 97)*, ACM, 1997.
4. Alexander Wilkie, Robert Tobler, Werner Purgathofer, Combined Rendering of Polarization and Fluorescence Effects, *Proceedings of 12th Eurographics Workshop on Rendering*, June 2001.
5. Henrik Wann Jensen, Stephen Marschner, Marc Levoy, Pat Hanrahan, A Practical Model for Subsurface Light Transport, *Computer Graphics (Proceedings of SIGGRAPH 01)*, ACM, 2001.
6. Xiaodong He, Ken Torrance, François Sillion, Don Greenberg, A Comprehensive Physical Model for Light Reflection, *Computer Graphics (Proceedings of SIGGRAPH 91)*, ACM, 1991.
7. Jay Gondek, Gary Meyer, Jon Newman, Wavelength Dependent Reflectance Functions, *Computer Graphics (Proceedings of SIGGRAPH 94)*, ACM, 1994.
8. Holly Rushmeier, Ken Torrance, The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium, *Computer Graphics (Proceedings of SIGGRAPH 87)*, ACM, 1987.
9. Henrik Wann Jensen, Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps, *Computer Graphics (Proceedings of SIGGRAPH 98)*, ACM, 1998.
10. Jim Kajiya, The Rendering Equation, *Computer Graphics (Proceedings of SIGGRAPH 86)*, ACM, 1986.
11. Greg Ward Larson, Rob Shakespeare, *Rendering with Radiance*, Morgan Kaufmann Publishers, 1997.
12. Henrik Wann Jensen, *Realistic Image Synthesis Using Photon Mapping*, A.K. Peters Ltd., 2001.
13. Francois Sillion, Claude Puech, *Radiosity and Global Illumination*, Morgan Kaufmann Publishers, 1994.
14. C. Li, M.R. Luo, B. Rigg, Simplification of the CMCCAT97, *Proc. IS&T/SID 8th Color Imaging Conference*, November 2000.
15. Sabine Süsstrunk, Jack Holm, Graham Finlayson, Chromatic Adaptation Performance of Different RGB Sensors, *IS&T/SPIE Electronic Imaging*, SPIE Vol. 4300, January 2001.
16. Günter Wyszecki, W.S. Stiles, *Color Science*, J. Wiley, 1982.
17. Greg Ward, Real Pixels, *Graphics Gems II*, edited by James Arvo, Academic Press, 1992.
18. Greg Ward Larson, Overcoming Gamut and Dynamic Range Limitations in Digital Images, *IS&T/SID 6th Color Imaging Conference*, November 1998.
19. Greg Ward Larson, The LogLuv Encoding for Full Gamut, High Dynamic Range Images, *Journal of Graphics Tools*, 3(1):15-31 1998.
20. Greg Ward Larson, Holly Rushmeier, Christine Piatko, A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 3, No. 4, December 1997.

Biography

Greg Ward (a.k.a. Greg Ward Larson) graduated in Physics from UC Berkeley in 1983 and earned a Master's in Computer Science from SF State University in 1985. Since 1985, he has worked in the field of light measurement, simulation, and rendering variously at the Berkeley National Lab, EPFL Switzerland, Silicon Graphics Inc., Shutterfly, and Exponent. He is author of the widely used *Radiance* package for lighting simulation and rendering.

Submitted to the First Symposium on Applied Perception in Graphics and Visualization, Los Angeles, August 7-8, 2004

Subband Encoding of High Dynamic Range Imagery

Greg Ward[1] and Maryann Simmons[2]
(1) Sunnybrook Technologies, (2) Walt Disney Feature Animation

Abstract

The transition from traditional 24-bit RGB to high dynamic range (HDR) images is hindered by excessively large file formats with no backwards compatibility. In this paper, we propose a simple approach to HDR encoding that parallels the evolution of color television from its grayscale beginnings. A tone-mapped version of each HDR original is accompanied by restorative information carried in a subband of a standard 24-bit RGB format. This subband contains a compressed *ratio image*, which when multiplied by the tone-mapped foreground, recovers the HDR original. The tone-mapped image data may be compressed, permitting the composite to be delivered in a standard JPEG wrapper. To naïve software, the image looks like any other, and displays as a tone-mapped version of the original. To HDR-enabled software, the foreground image is merely a tone-mapping suggestion, as the original pixel data are available by decoding the information in the subband. We present specifics of the method and the results of encoding a series of synthetic and natural HDR images, using various published global and local tone-mapping operators to generate the foreground images. Errors are visible in only a very small percentage of the pixels after decoding, and the technique requires only a modest amount of additional space for the subband data, independent of image size.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image generation – Display Algorithms
I.4 [Image Processing and Computer Vision]: I.4.10 Image Representation

Keywords: high dynamic range image formats, lossy compression, image processing

1. Introduction

Visible light in the real world covers a vast dynamic range. Humans are capable of simultaneously perceiving over 4 orders of magnitude (1:10000 contrast ratio), and can adapt their sensitivity up and down another 6 orders. Conventional digital images, such as 24-bit *sRGB* [Stokes et al. 1996], hold less than 2 useful orders of magnitude. Such formats are called "output-referred" standards because they are tailored to what can be displayed on a common CRT monitor – colors outside this limited gamut are not represented. A "scene-referred" standard is designed instead to represent colors and intensities that are visible in the real world, and though they may not be rendered faithfully on today's output devices, they will be visible on displays in the near future [Seetzen et al. 2003]. Such image representations are referred to as extended gamut or *high dynamic range* (HDR) formats, and a few alternatives have been introduced over the past 15 years, mostly by the graphics research and special effects communities.

Unfortunately, none of the existing or proposed HDR formats supports lossy compression, and only one comes in a conventional image wrapper. These formats yield prohibitively large images that can only be viewed and manipulated by specialized software. Commercial hardware and software developers have been slow to embrace scene-referred standards due to their demands on image capture, storage, and use. Some digital camera manufacturers attempt to address the desire for greater exposure control during processing with their own proprietary RAW formats, but these camera-specific encodings fail in terms of image archiving and third-party support. They are convenient for the manufacturers, but for no one else.

What we really need is a compact image that looks and displays like an output-referred JPEG, but holds the extra information needed to enable it as a scene-referred standard. Future HDR cameras will then be able to write to this format without fear that the software on the receiving end won't know what to do with it. Conventional image manipulation and display software will see only the tone-mapped version of the image, gaining some benefit from the HDR capture due to its better exposure. HDR-enabled software will have full access to the original dynamic range recorded by the camera, permitting large exposure shifts and contrast manipulation during image editing. Establishing such a standard will provide a smooth upgrade path for manufacturers and consumers alike.

1.1. Background

High dynamic range imaging goes back many years. A few early researchers in image processing advocated the use of logarithmic encodings of intensity (e.g., [Jourlin & Pinoli 1988]), though it was global illumination that brought us the first standard. A space-efficient format for HDR images was introduced in 1989 as part of the *Radiance* rendering system [Ward 1991; Ward 1994]. However, the *Radiance* RGBE format was not widely used until HDR photography and environment mapping were developed by Debevec [Debevec & Malik 1997; Debevec 1998]. About the same time, Ward Larson [1998] introduced the LogLuv alternative to RGBE and distributed it as part of Leffler's public TIFF library [Leffler et al. 1999]. The LogLuv format has the advantage of covering the full visible gamut in a more compact representation, whereas RGBE is restricted to positive primary values. A few graphics researchers adopted the LogLuv format, but most continued to use RGBE (or its extended gamut variant XYZE), until Industrial Light and Magic made their EXR format available in 2002 [Kainz et al. 2002]. The OpenEXR library uses the same basic 16-bit floating point data type as modern graphics cards, and is poised to be the new favorite in the special effects industry. Other standards have also been proposed or are in the works, but they all have limited dynamic range relative to their size (e.g., [IEC 2003]). None of these standards is backwards compatible with existing software.

The current state of affairs in HDR imaging parallels the development of color television after the adoption of black and white broadcast. A large installed base must be accommodated as

well as an existing standard for transmission. The NTSC solution introduced a subband to the signal that encoded the additional chroma information without interfering with the original black and white signal [Jolliffe 1950]. We propose a similar solution in the context of HDR imagery, with similar benefits.

As in the case of black and white television, we have an existing, de facto standard for digital images: output-referred JPEG. JPEG has a number of advantages that are difficult to beat. The standard is unambiguous and universally supported. Software libraries are free and widely available. Encoding and decoding is fast and efficient, and display is straightforward. Compression performance for average quality images is competitive with more recent advances, and every digital camera writes to this format. Clearly, we will be living with JPEG images for many years to come. Our chances of large scale adoption increase dramatically if we can maintain backward compatibility with this standard.

Our general approach is to introduce a subband that accompanies a tone-mapped version of the original HDR image. This subband is compressed to fit in a metadata tag that will be ignored by naïve applications, but can be used to extract the HDR original in enabled software. We utilize JPEG/JFIF as our standard wrapper in this implementation, but our technique is compatible with any format that provides client-defined tags (e.g., TIFF, PNG, etc.).

The Method section of our paper describes the basic idea behind HDR subband encoding, followed by a more detailed description of the steps involved and our trial implementation. The Results and Discussion section presents a full set of example images and examines sources of error in our encoding. We end with our conclusions and future directions.

2. Method

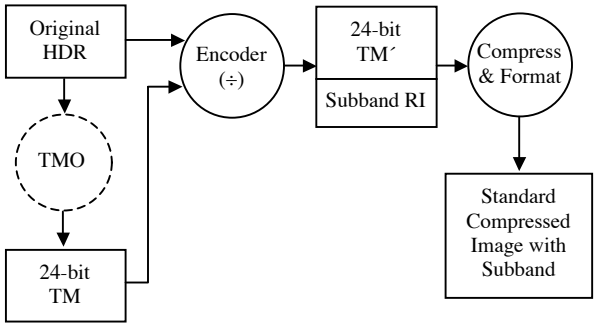


Figure 1. High level HDR subband encoding pipeline.

Figure 1 shows an overview of our HDR encoding pipeline. We start with two versions of our image: a scene-referred HDR image, and an output-referred, tone-mapped version. If we like, we can generate this tone-mapped version ourselves, but in general this is a separable problem, because our technique is designed to work with multiple operators. The encoding stage takes these two inputs and produces a composite, consisting of a potentially modified version of the original tone-mapped image, and a subband ratio image that contains enough information to reproduce a close facsimile of the HDR original. The next stage compresses this information, offering the tone-mapped version as the JPEG base image, and storing the subband as metadata in a standard JFIF wrapper.

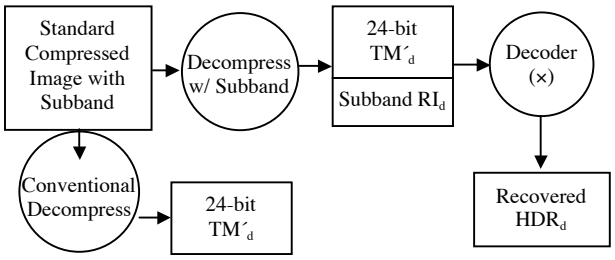


Figure 2. Alternate decoding paths for compressed composite.

Figure 2 shows the two possible decoding paths. The high road decompresses both the tone-mapped foreground image and the subband, delivering them to the decoder to recover the HDR pixels. Naïve applications follow the low road, ignoring the subband in the metadata and reproducing only the tone-mapped foreground image.

In the simplest incarnation of this method, the encoder divides the HDR pixels by the tone-mapped luminances, producing an 8-bit ratio image that is stored as metadata in a standard JPEG compressed image. Decoding follows decompression with a multiplication step to recover the original HDR pixels. Unfortunately, the simplest approach fails for pixels that are mapped to black or white by the tone-mapping operator (TMO), and the limited size of metadata in JPEG makes subband compression a challenge. These are two important issues we must address with our technique.

2.1. The Ratio Image

JPEG/JFIF offers us a limited subband channel, called the "application markers." Sixteen application markers exist in the JFIF specification, three of which are spoken for. A marker may hold up to 64 Kbytes of data, regardless of the image dimensions. This is an important limitation in our method – we want to keep our subband data under 64K, independent of image size. The foreground signal in our application is a tone-mapped version of the original HDR image. This output-referred image is stored in the usual 8×8, 8-bit blocks using JPEG's lossy DCT compression [Wallace 1991]. The subband encodes the information needed to restore the HDR original from this compressed version.

Let us assume that our selected tone-mapping operator possesses the following properties:

A. The original HDR input is mapped smoothly into a 24-bit output domain, with no components being rudely clamped at 0 or 255.

B. Hue is maintained at each pixel, and mild saturation changes may be described by an invertible function of input and output value.

Most tone-mapping operators for HDR images have the first property as their goal, so this should not be a problem. If it is, we can override the operator by locally replacing each clamped pixel with a lighter or darker value that fits the range. Similarly, we can enforce the second property by performing our own color desaturation, using the tone-mapping operator as a guide only for

luminance changes. Most tone-mapping operators address color in a very simple way, if they consider it at all.[1] Exceptions are found in operators that simulate human vision (e.g., [Pattanaik et al. 1998]), whose support we leave as future work.

Given these restrictions, a ratio image may be computed by dividing the HDR original luminance at each pixel by the tone-mapped output luminance:

$$RI(x,y) = \frac{L(HDR(x,y))}{L(TM(x,y))} \quad (1)$$

The ratio image is compressed and sent in our subband along with the saturation formula. The tone-mapped version is then encoded as the foreground image, modified as necessary to avoid clamping. During reconstruction, the ratio image is multiplied by the foreground image to recover the original HDR luminance values. Color is then restored using the recorded saturation formula. (See Appendix for details.)

2.2. Subband Compression

Obviously, we will not meet our goal of fitting our subband into 64 Kbytes if we send our ratio image along uncompressed. Since our goal is to encapsulate our subband in a JPEG wrapper, it would be most convenient if we could compress the ratio image using JPEG as well.[2] If we could also fix the maximum size of our ratio image, we could avoid the problem of needing more space or greater compression for larger input images.

Figure 3 shows the Memorial Church image, tone-mapped using the global zone operator of Reinhard et al. [2002]. Ratio values are mapped into an 8-bit range by a log encoding that captures the extrema, as we show on the right. Because the original image is only 512x768 pixels, we can compress our ratio image into 48 Kbytes with a JPEG quality setting of 90, without resorting to downsampling. We see a before and after close-up of the recovered HDR image near the window, where luminance compression is greatest, in Figure 4. A comparison of the dark ceiling is shown on the left. Qualitatively, we are able to reproduce the original HDR pixels using this method, although JPEG artifacts are beginning to appear. Clearly, we cannot push this approach to much larger images and hope to stay within the 64 Kbyte limit.

Figure 5 shows a 2048x1536 image of the Dyrham Church next to its reduced ratio image. In this composite, we have downsampled the subband image to 768x576 and compressed it to 41 Kbytes using the same JPEG quality setting of 90. Recovery is quite good in darker regions, as shown on the right of Figure 6, but we start to lose focus on bright boundaries, such as the window panes shown on the left. This is due to blur in the ratio image introduced when we upsample prior to multiplication. We need some method of recovering the high frequencies we lost when we downsampled the ratio image. In the following sections, we present two alternate recovery methods: precorrecting the foreground image, and postcorrecting the ratio image.

[1] We can use a fitting function to match desaturation of the exemplar tone-mapped image if it is unknown.

Figure 3. Memorial Church shown tone-mapped with Reinhard's global operator (left) along with the ratio image needed for HDR recovery (right).

Figure 4. Linear displays of original (top) and recovered (bottom) HDR images. On the left is a close-up of the ceiling, and on the right is a close-up of the rightmost windows.

Figure 5. The Dyrham Church image rendered with Reinhard's global operator, shown next to the corresponding downsampled ratio image.

Figure 6. Details of the image recovered using the downsampled subband.

2.3. Precorrection of Foreground Image

One way to retain the high frequency information is to precorrect the foreground image based on the compressed subband. After computing the subband as above, we redivide the HDR original by the decompressed and upsampled ratio image to get a modified foreground image:

$$TM' = \frac{HDR}{RI_d} \tag{2}$$

Substituting this modified foreground image for the original effectively undoes the damage of lossy compression and downsampling. The left image in Figure 7 shows the lattice window after applying Reinhard's global tone-mapping operator to the Dyrham Church image. On the right, we see the result of dividing the HDR original by the computed ratio image. By construction, the result of multiplying this modified foreground image by our downsampled ratio image will be very close to the original, even after JPEG transmission. Figure 8 shows a linear rendition of the HDR original next to the recovered result. The precorrection method is a good complement to Reinhard's operator, because it restores some of the contrast lost at the high end. However, the sharpening produced by this technique may be undesirable when the TMO has already produced an optimal foreground image. In such cases, we may prefer not to modify the foreground image during encoding, choosing instead to recover high frequencies in a post-process.

Figure 7. Reinhard's tone-mapping operator applied to the Dyrham Church image, before and after modification by the downsampled ratio image in our precorrection method.

Figure 8. Linear comparison of HDR image recovery improved by precorrecting the tone-mapped foreground image.

2.4. Postcorrection of Ratio Image

If we invest the time to generate a high quality foreground image with a sophisticated TMO, we may be unwilling to accept the effects of precorrection as described in the previous section. For a sufficiently high resolution original, recomputing TM´ using Eq. (2) can result in small halos that are visible in close-ups, as shown in the sunset of Figure 9.

Figure 9. Reverse gradients visible on a modified high-resolution (3000x1700) foreground image. The original bilateral filter tone-mapping is shown on the left inset.

Ideally, we would like to preserve the tone-mapped representation in the foreground image without losing resolution in the recovered HDR result. One way to achieve this without exceeding our 64 Kbyte subband limit is to synthesize the missing high frequencies in our resampled ratio image. Since we have a full-resolution foreground image, we can use this as a guide for where edges belong in the ratio image. If the frequency content of the foreground image and the ratio image before downsampling were the same, one could recover the high frequencies in the resampled ratio image with the following simple formula:

$$RI_{synth} = RI_d \frac{L(TM)}{L(TM_r)}$$

In this equation, $L(TM_r)$ is the luminance of the tone-mapped foreground image, resampled in the same way as the encoded ratio image. Of course, there is no guarantee that the frequency content of the foreground and ratio images are the same, especially using a TMO like the bilateral filter, which attempts to

preserve local detail [Durand & Dorsey 2002]. A better approximation therefore attenuates the effect by the ratio between the local variance of RI_d and TM_r. If the ratio image has little or no variance after resampling, then it probably had little in the way of high frequencies beforehand. This improved frequency correction may be written as follows:

$$RI_{synth} = RI_d \cdot \left[\frac{L(TM)}{L(TM_r)}\right]^{\sigma} \qquad (3)$$

$$where: \quad \sigma = \frac{\mathrm{var}(RI_d)}{\mathrm{var}(L(TM_r))}$$

The relative variance is computed over a small neighborhood, equal to the resampling radius, and σ is set to 0 if var($L(TM_r)$) is less than our compression-decompression error. In practice, we do not allow the exponent σ to exceed 1, as this can cause overshooting in the output. We define relative variance as the difference between the maximum and minimum in a neighborhood, divided by the central value.

Figure 10 shows the results of applying high frequency synthesis to the Napa Valley image. The foreground image for this encoding was computed using a bilateral filter, hence there are significant discrepancies between the tone-mapped version and the ratio image. As we can see, the approximation in Eq. (3) does a reasonable job of restoring the high frequency information that has been lost during the resampling process, without introducing objectionable artifacts. However, the errors in the recovered HDR image are greater with this method than those of Eq. (2), so the application should decide whether the benefits of a cleaner foreground image are worth the costs. A flag must be passed in the subband, indicating whether the image was precorrected by Eq. (2), or should be postcorrected by Eq. (3).

Figure 10. The center is the original HDR image with a linear tone-mapping. On the left is our uncorrected ratio image multiplied by the foreground image. On the right, we postcorrected the ratio image with synthetic high frequencies.

3. Results and Discussion

We tested our encoding on the fifteen HDR images shown in Figure 11. Table 1 lists each image size and dynamic range. Eleven images are captures of natural scenes, and four are synthetic. The dynamic ranges of the images are between 3.6 and 9.2 orders of magnitude, with 4.9 being average. The smallest image is 346×512; the largest is 5462×4436, and the median is 0.9 megapixels.

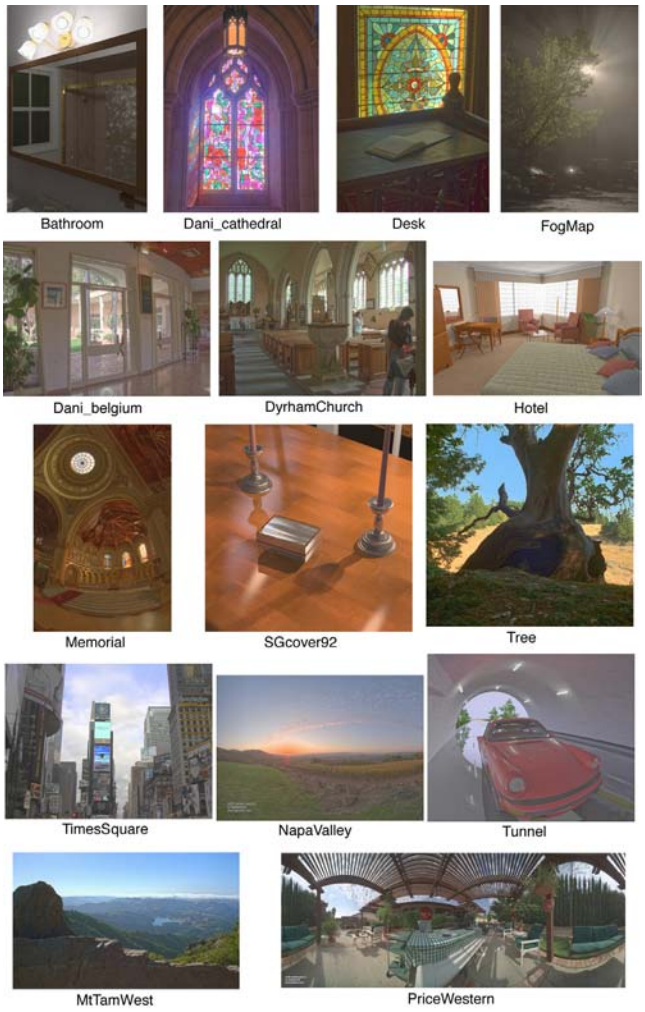


Figure 11. Test image set.

Image	Size	Dynamic Range	Source
Bathroom	346×512	4.8	*Radiance*
Dani_belgium	1025×769	5.8	digital
Dani_cathedral	767×1023	4.8	digital
Desk	644×874	5.2	film
DyrhamChurch	2048×1536	4.0	digital
FogMap	751×1130	4.1	film
Hotel	3000×1950	4.7	*Radiance*
MtTamWest	1214×732	4.1	film
Memorial	512×768	5.5	film
NapaValley	3025×2129	5.3	Spheron
PriceWestern	3272×1280	3.7	Spheron
SGcover92	1024×1024	4.7	*Radiance*
TimesSquare	2272×1704	3.6	digital
Tree	928×906	4.1	film
Tunnel	5462×4436	9.2	*Radiance*

Table 1. Test image sizes. Images were either synthetic (*Radiance* renderings), or captured (multiple film or digital exposures, or panoramic SpheronVR scans).

We tested four different tone-mapping operators to produce the foreground image: Ward Larson et al.'s histogram method [1997], Reinhard et al.'s global zone method [2002], Fattal et al.'s gradient operator [2002], and Durand & Dorsey's bilateral filter method [2002]. Of these, only the bilateral filter was applied consistently to all images, since the global operators did not seem as well-suited, and we did not have implementations of the other local operators, relying instead on downloads of the exemplar images from the authors' websites.

Our test procedure was simple: encode then decode each image using the selected TMO and JPEG compression parameters, then compare the recovered HDR image to the original. We tested two JPEG compression levels (quality settings) on the foreground image: 90 and 100, using Tom Lane's public JPEG implementation. The ratio image was always compressed with the highest JPEG quality setting that kept the result under 60 Kbytes, leaving ample room for other subband data. If the original image was greater than 400,000 pixels, the ratio image was downsampled to this size before compression.

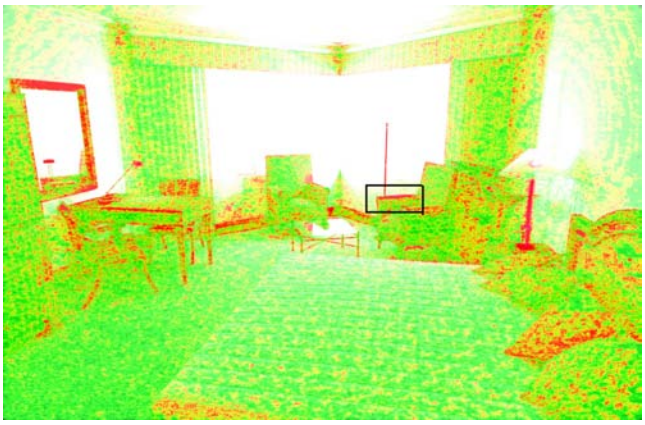


Figure 12. Sample VDP output for Hotel image. Red shows threshold where difference detection probability exceeds 0.75. (Green $p \le 0.5$, Yellow $p \le 0.63$, Purple $p > 0.95$)

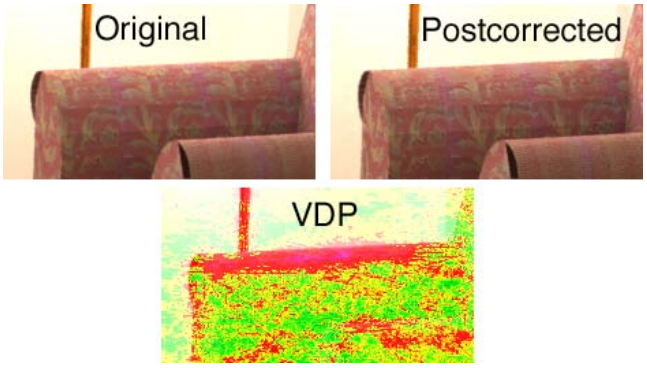


Figure 13. Close-up of Hotel chair's arm from box in Figure 12 with VDP visualization.

To compare our decoded HDR images to their corresponding originals, we employed Daly's Visible Differences Predictor (VDP) [1993]. This metric tells us what percentage of our decoded image pixels are likely (probability $p > 0.75$) to be perceived as different from the originals under standard viewing conditions. Figure 12 visualizes the VDP output for the postcorrected Hotel encoding using the bilateral filter with JPEG quality set to 100. Figure 13 shows a close-up of the indicated region of this image, verifying the correlation between actual perceptible differences and VDP on the arm of the chair. Empirically, we found VDP to be an excellent predictor of where we could see differences in our images.

Our results are summarized in Table 2, where we have averaged our VDP percentages over all images except "Tunnel," which we considered an outlier. We see a sizeable discrepancy in the VDP results for different tone-mapping operators, and for precorrection versus postcorrection. The JPEG quality setting also made a difference, as one would expect.

TMO	Quality	VDP (pre)	VDP (post)
Bilateral Filter	90 100	0.93% 0.02%	5.4% 1.8%
Reinhard Global	90 100	2.5% 0.09%	4.7% 2.8%
Histogram Adj.	90 100	5.9% 0.63%	21% 17%
Gradient	90 100	7.5% 3.0%	36% 34%

Table 2. Percentage of perceptibly different pixels summed over all images for four tone mapping operators using VDP metric on precorrected and postcorrected encodings. Two JPEG quality settings were tested for each foreground image.

Image	Qual.	CR	VDP (pre)	VDP (post)
Bathroom	90 100	9.6 5.7	1.3% 0.00%	0.00% 0.00%
Dani_belgium	90 100	14.4 5.3	0.32% 0.01%	6.2% 3.0%
Dani_cathedral	90 100	12.3 4.5	0.24% 0.00%	2.7% 1.1%
Desk	90 100	10.7 4.6	0.06% 0.02%	2.7% 1.8%
DyrhamChurch	90 100	21.3 6.4	0.03% 0.00%	3.2% 1.5%
FogMap	90 100	16.6 5.9	0.90% 0.00%	1.2% 0.22%
Hotel	90 100	26.0 7.9	0.14% 0.04%	7.1% 2.8%
Memorial	90 100	10.3 4.4	5.3% 0.01%	12.3% 0.76%
MtTamWest	90 100	14.2 4.9	2.4% 0.00%	4.5% 0.17%
NapaValley	90 100	26.3 6.9	0.01% 0.01%	1.9% 1.6%
PriceWestern	90 100	14.5 5.0	2.1% 0.16%	27.3% 9.8%
SGcover92	90 100	18.3 6.2	0.01% 0.00%	0.50% 0.47%
TimesSquare	90 100	22.4 6.7	0.08% 0.01%	4.0% 1.4%
Tree	90 100	9.6 3.9	0.09% 0.00%	1.9% 0.42%
Tunnel	90 100	23.5 7.3	5.3% 3.0%	53% 45%

Table 3. Compression ratios and VDP percentages for each of the test images, mapped with the bilateral filter TMO.

The bilateral filter proved to be an excellent fit to our encoding scheme, with lower associated errors than the other techniques. The Reinhard TMO also fared reasonably well, but the histogram and gradient methods generated many visible errors, especially when coupled with our postcorrection method. Most of these errors occured in darker regions, where pixels were mapped to very small values. This is not a general indictment of these operators – it simply means they are less suited for splitting information between foreground and ratio images as our method requires.

Our results using the bilateral filter are broken out for each image in Table 3. When using a precorrected foreground image with low JPEG compression, only the Tunnel image had more than a small percentage of pixels where differences were discernable. These percentages got larger for the lower quality setting, reaching 5.3% for the Memorial image, but remained acceptable for most of the others. (We found 2% to be a reasonable cut-off for a good side-by-side match.) In all but a few cases, the postcorrected results were acceptable with the higher JPEG quality, but VDP reached a few percent for about half our images on the lower setting, and showed a real failure on the Tunnel image.

We measured our compression performance relative to an uncompressed RGBE original (i.e., 32 bits/pixel). The compression ratios varied so little between precorrected and postcorrected encodings that we averaged the two in Table 3. At a foreground quality setting of 90, we saw compression ratios between 9.6 and 26.3, with an average performance of 17.0. Taking a typical example, the Fog Map image compressed from its original 3.2 Mbytes down to 200 Kbytes. At a foreground quality setting of 100, we saw compression ratios ranging from 3.9 to 7.9, with an average performance of 5.7. Unsurprisingly, the smallest compression ratios were associated with the smallest original images. The highest compression ratios were associated with the Napa Valley image. For comparison, the average compression ratio achieved by the most sophisticated, lossless HDR image format is 1.6 on this data set, using OpenEXR's "PIZ" wavelet encoding [Kains et al. 2002].

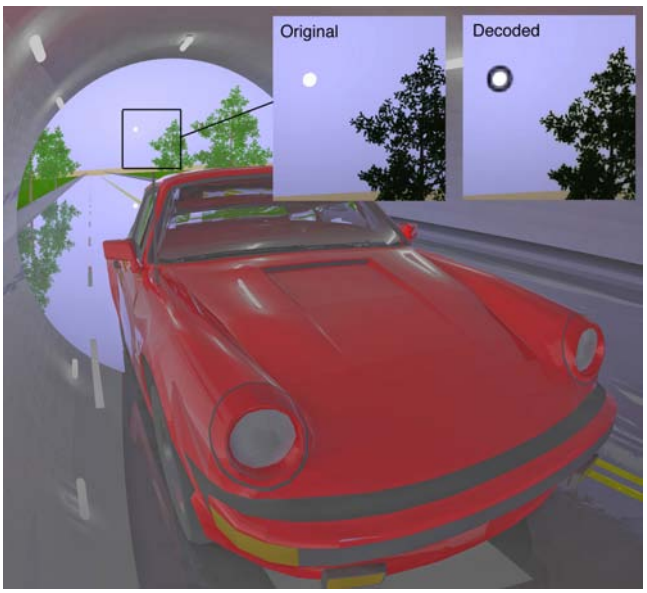


Figure 14. Close-up showing details lost in small regions of the Tunnel due to downsampling of the ratio image.

We would expect our errors to increase somewhat for larger images with higher dynamic range, since the ratio image must be scaled to fit both these input parameters. Examining our bilateral filter results, we searched for correlations between VDP and image size, and VDP and dynamic range, but found no significant trends in our data, with the exception of a single outlier: the Tunnel. This is really the worst case image for our algorithm – not only is it the largest (23 Mpixel), it also has the greatest dynamic range (a billion to 1), and since it came unfiltered out of a stochastic ray-tracing program, it contains abnormal amounts of high frequency, HDR pixel noise. Our postcorrection algorithm had visible errors over half the image, and even precorrection could not cope where sampling discontinuities exceeded the 8-bit carrying capacity of the foreground image. For example, Figure 14 shows a halo around the sun that derives from a huge luminance discontinuity – almost 5 orders of magnitude between neighboring pixels. This is an extreme jump that could only be generated synthetically, and demands an encoding that carries 16 bits at every pixel. Therefore, we may wish to consider ways to bypass the restriction on the ratio image resolution for such extreme images, possibly working around the 64 Kbyte limit for JPEG markers by stringing multiple markers in series, which is permitted by the JFIF standard.

4. Conclusion and Future Directions

By providing a lossy, high dynamic range image format that is backwards compatible with existing JPEG software, we remove an important barrier to the adoption of HDR imaging technology by digital camera manufacturers and web content providers. The subband encoding method we presented couples a high quality, tone-mapped (i.e., output-referred) foreground image with metadata that enables HDR software to recover the original, scene-referred luminances at 16-bit resolution. Naïve applications see only the tone-mapped version, which still encompasses the larger dynamic range, albeit with 8-bit precision. Color is encoded in the foreground image as well, which HDR-enabled applications may resaturate to access a wider gamut than standard RGB. With the current prototype implementation, we obtain predicted visible differences at only a very small percentage of the pixels over a wide range of HDR image sizes, adding just 64 Kbytes of subband data to each tone-mapped JPEG.

Although it will not affect the critical standardization of the subband format and decoding method, there is more work to do in finding an optimal tone-mapping operator for the encoding. We found Durand and Dorsey's bilateral filter [2002] to behave quite well, and Reinhard et al.'s global operator [2002] to perform adequately, but further testing is needed. We would like to incorporate an operator that is fast, robust, and completely automatic. Further, we would like to explore tuning of this operator to minimize problems in the ratio image that could show up as artifacts in the decoded HDR result.

Although our initial implementation is tied to the standard JPEG encoding, there is no reason the same separation of tone-mapped and ratio image could not be applied within other existing and emerging image standards. The advantage to this approach over a direct extension to incorporate an HDR color space is two-fold. Firstly, an output-referred image is immediately available to all applications, avoiding the need for a potentially time-consuming tone-mapping step prior to viewing. Secondly, separate control is possible for the fidelity/bitrate of the output-referred and scene-referred versions, permitting application-tuned encodings. As an

added benefit, web-savvy formats such as PNG and JPEG 2000 can send the ratio image as a separate bundle only when requested by the client, eliminating the associated cost of this additional information where it is not needed. In other words, there may be benefits to backwards compatibility looking forward as well.

5. Acknowledgements

The VDP implementation was provided by Karol Myszkowski, and Dave Shreiner helped process all the data. Thanks to Paul Debevec, Dani Lischinski, Erik Reinhard, Jack Tumblin, Spheron Corporation, and ILM for lending their HDR images. Thanks also to Scott Daly for useful insights into his VDP metric.

6. References

ASHIKHMIN M. 2002. A Tone Mapping Algorithm for High Contrast Images. In *Proceedings of 13th Eurographics Workshop on Rendering*, 145-156.

CHIU, K. HERF, M., SHIRLEY, P., SWAMY, M., WANG, C., and ZIMMERMAN, K., 2002. A Tone Mapping Algorithm for High Contrast Images. In *Proceedings of 13th Eurographics Workshop on Rendering*, 245-253.

DALY, S. 1993. The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity. In *Digital Images and Human Vision*, A.B. Watson, editor, MIT Press, Cambridge, Massachusetts.

DEBEVEC, P., and MALIK, J. 1997. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proceedings of ACM SIGGRAPH 1997*, 369-378.

DEBEVEC, P. 1998. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography. In *Proceedings of ACM SIGGRAPH 1998*, 189-198.

DURAND, F., and DORSEY, J. 2002. Fast Bilateral Filtering for the Display of High-Dynamic Range Images. *ACM Transactions on Graphics*, 21, 3, 249-256.

FATTAL, R., LISCHINSKI, D., and WERMAN, M. 2002. Gradient Domain High Dynamic Range Compression. *ACM Transactions on Graphics*, 21, 3, 257-266.

IEC. 2003. 61966-2-2. Extended RGB colour space – scRGB, *Multimedia systems and equipment – Colour measurement and management* – Part 2-2: Colour management.

JOLLIFFE, C.B., 1950. Answers to Questions about Color Television. members.aol.com/ajaynejr/rca2.htm.

JOURLIN, M., PINOLI, J-C., 1988. A model for logarithmic image processing, *Journal of Microscopy*, 149(1), pp. 21-35.

KAINS, F., BOGART, R., HESS, D., SCHNEIDER, P., ANDERSON, B., 2002. *OpenEXR*. www.openexr.org/.

LEFFLER, S., WARMERDAM, F., KISELEV, A., 1999. *libTIFF*. remotesensing.org/libtiff.

MOON P., and SPENCER, D. 1945. The Visual Effect of Non-Uniform Surrounds. *Journal of the Optical Society of America*, 35, 3, 233-248.

PATTANAIK, S., FERWERDA, J., FAIRCHILD, M., and GREENBERG, D. 1998. A Multiscale Model of Adaptation and Spatial Vision for Realistic Image Display, In *Proceedings of ACM SIGGRAPH 1998*, 287-298.

REINHARD, E., STARK, M., SHIRLEY, P., and FERWERDA, J. 2002. Photographic Tone Reproduction for Digital Images. *ACM Transactions on Graphics*, 21,3, 267-276.

SEETZEN, H., WHITEHEAD, L., and WARD, G. 2003. A High Dynamic Range Display Using Low and High Resolution Modulators. In *Proceedings of the Society for Information Display International Symposium*, Baltimore, MD.

STOKES, M., ANDERSON, M., CHANDRASEKAR, S., and MOTTA, R. A 1996. Standard Default Color Space for the Internet. www.w3.org/Graphics/Color/sRGB.

TUMBLIN, J., and TURK, G. 1999. LCIS: A Boundary Hierarchy for Detail-Preserving Contrast Reduction. *ACM Trans. on Graphics*, 21, 3, 83-90.

WALLACE, G. 1991. The JPEG Still Picture Compression Standard. *Communications of the ACM*, 34, 4, 30-44.

WARD LARSON, G., RUSHMEIER, H., and PIATKO, C. 1997. A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. *IEEE Trans. on Visualization and Computer Graphics*, 3, 4.

WARD LARSON, G. 1998. Overcoming Gamut and Dynamic Range Limitations in Digital Images. *Proc. of IS&T 6th Color Imaging Conf.*

WARD, G. 1991. Real Pixels. In *Graphics Gems II*, edited by James Arvo, Academic Press, 80-83.

WARD, G. 1994. The RADIANCE Lighting Simulation and Rendering System. , In *Proceedings of ACM SIGGRAPH 1994*, 459-472.

7. Appendix: Color Saturation and Gamut

Our treatment of color is straightforward. Any color with a primary component outside the "safe" range for JPEG encoding is scaled back into range, and the ratio image is adjusted for correct recovery. This works only for positive primary values. Negative primary values are legal in some HDR formats, and are indeed necessary for certain real chromaticities outside the standard, triangular RGB gamut.

To accommodate negative primaries, we apply a global desaturation to the image, pulling all colors in towards gray by an amount that is guaranteed to contain the entire visible gamut in the legal JPEG range. This may in fact be beneficial to the appearance of the foreground image, as dynamic range compression tends to leave the colors with an oversaturated appearance. The desaturation process is then reversed during the decoding, rendering the colors back into their original, full gamut glory. We use the following definition of input color saturation:

$$S \equiv 1 - \min(R,G,B)/Y \tag{4}$$

Y in the formula above is the luminance of red, green and blue taken together. We expect saturation to be greater than 1 for negative primaries. Zero saturation implies a neutral value, which is passed. Non-neutral values are desaturated with a two-parameter formula:

$$S' = \alpha \cdot S^{\beta} \tag{5}$$

The α parameter controls how much saturation we wish to keep in the encoded colors, and is generally ≤ 1. The β parameter controls the color "contrast," and is usually ≥ 1. This modified saturation is used with the original saturation from Eq. (4) to determine the encoded primary values. Below is the formula for the desaturated red channel:

$$R' = \left(1 - \frac{S'}{S}\right) \cdot Y + \frac{S'}{S} \cdot R \tag{6}$$

And similarly for the green and blue channels. Note that Y does not change under this transformation, and the primary that was smallest before is the smallest after. Resaturating the encoded color to get back the original pixel is done by inverting these equations. If the smallest primary value were blue for example, this inverse transformation would yield:

$$B = Y - Y \cdot \left(\frac{Y - B'}{\alpha Y}\right)^{1/\beta} \tag{7}$$

The red and green channels would then be determined by:

$$R = Y - \frac{(Y - R')}{\alpha}\left(1 - \frac{B}{Y}\right)^{1-\beta}, \quad G = Y - \frac{(Y - G')}{\alpha}\left(1 - \frac{B}{Y}\right)^{1-\beta} \tag{8}$$

If either red or green were the minimum primaries, these equations would be switched around accordingly.