Capturing and Simulating Physically Accurate Illumination in Computer Graphics

PAUL DEBEVEC
Institute for Creative Technologies
University of Southern California
Marina del Rey, California

Anyone who has seen a recent summer blockbuster has witnessed the dramatic increases in computer-generated realism in recent years. Visual effects supervisors now report that bringing even the most challenging visions of film directors to the screen is no longer a question of what's possible; with today's techniques it is only a matter of time and cost. Driving this increase in realism have been computer graphics (CG) techniques for simulating how light travels within a scene and for simulating how light reflects off of and through surfaces. These techniques—some developed recently, and some originating in the 1980's—are being applied to the visual effects process by computer graphics artists who have found ways to channel the power of these new tools.

RADIOSITY AND GLOBAL ILLUMINATION

One of the most important aspects of computer graphics is simulating the illumination within a scene. Computer-generated images are two-dimensional arrays of computed pixel values, with each pixel coordinate having three numbers indicating the amount of red, green, and blue light coming from the corresponding direction within the scene. Figuring out what these numbers should be for a scene is not trivial, since each pixel's color is based both on how the object at that pixel reflects light and also the light that illuminates it. Furthermore, the illumination does not only come directly from the lights within the scene, but also indirectly from all of the surrounding surfaces in the form of "bounce" light. The complexities of light's behavior is a reason the world around us appears rich and interesting, and it makes generating images that are "photoreal" both conceptually and computationally complex.

As a simple example, suppose we stand at the back of a square white room, where the left wall is painted red, the right wall is painted blue, with light coming from the ceiling (Figure 1). If we take a digital picture of this room and examine its pixel values, we will indeed find that the red wall is red, and the blue wall is blue. But looking closely at the white wall in front of us, we'll notice that it isn't perfectly white. Toward the right it becomes bluish, and to the left it becomes pink. The reason for this is indirect illumination: toward the right, blue light from the blue wall adds to the illumination on the back wall, and toward the left, red light does the same.

Indirect illumination is responsible for more than the somewhat subtle effect of white surfaces picking up the colors of nearby objects—it is often responsible for the majority or all of the illumination on an object or within a scene. If I sit in a white room illuminated by a small skylight in the morning, it is the indirect light from the patch of sunlight on the wall that lights the rest of the room, not the direct light from the sun itself. If light did not bounce between surfaces, the room would be nearly dark!

In early computer graphics, such interreflections of light between surfaces in a scene were poorly modeled. The lighting falling on each surface was computing solely as a function of the light coming directly from a scene's light sources, perhaps plus a roughly determined amount of "ambient" light added irrespective of the actual colors of light in the scene. The groundbreaking publication that showed that indirect illumination could be modeled and computed accurately came at SIGGRAPH 84, when Goral, Torrance, and Greenberg at Cornell

University simulated the appearance of the red, white, and blue room example using a technique known as *Radiosity*. Their resulting image is shown below in Figure 1:

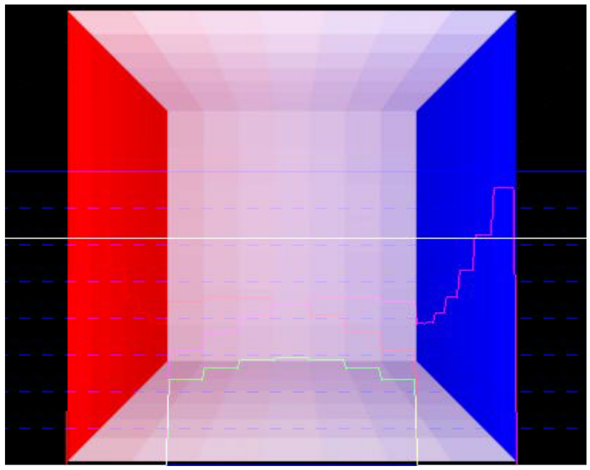


Figure 1. A simulation of indirect illumination within a scene, known as the "Cornell Box," from (Goral et al., 1984). The white wall at the back picks up color from the red and blue walls at the sides. (The Cornell box may be seen in color at: http://www.graphics.cornell.edu/online/box/history.html.)

Inspired by physics techniques for simulating heat transfer, the Cornell researchers divided each wall of the box into a 7 x 7 grid of patches, and for each patch determined its degree of visibility to every other patch, noting that patches reflect less light if they are further apart or facing away from each other. The final light color of each patch could then be written as its inherent surface color times the sum of the light coming from every other patch in the scene. Despite the fact that the illumination arriving at each patch depends on the illumination arriving (and thus leaving) every other patch, this *Radiosity* equation could be solved in a straightforward manner as a linear system of equations.

The result that Goral et al. obtained (Figure 1) correctly modeled that the white wall would subtly pick up the red and blue of the neighboring surfaces. Soon after this experiment, Cornell researchers actually constructed such a box with real wood and paint, and found that photographs of the box closely matched their simulations, to the point where people could not tell the difference under controlled conditions. The first "photoreal" image had been rendered!

A limitation of this work was that the time required to solve the resulting linear system increased with the cube of the number of patches within the scene, making the technique difficult to apply to complex models (especially in 1984). Another limitation was that in the radiosity model, all of the surfaces were assumed to be painted matte colors, with no shine or gloss. A subsequent watershed work in the field came at SIGGRAPH 86, when Jim Kajiya from Caltech published the "The Rendering Equation," which generalized the ideas of light transport to any kind of geometry and any sort of surface reflectance. The titular equation of his paper stated in general terms that the light leaving a surface in each direction is a function of the light arriving from all directions upon the surface, convolved by a function that describes how the surface reflects light. This latter function, called a surface's Bidirectional Reflectance Distribution Function, or BRDF, is constant for diffuse surfaces, and varies according to the incoming and outgoing directions of light for surfaces with shine and gloss.

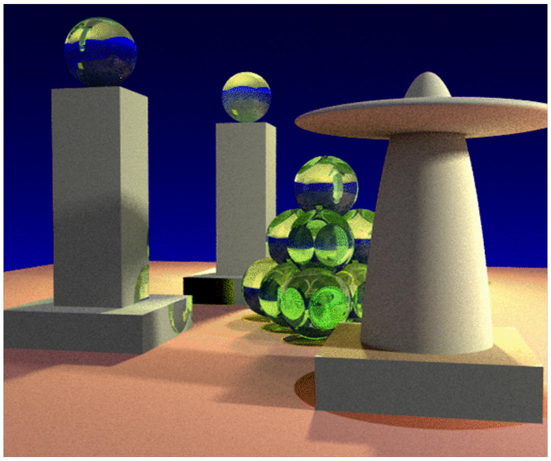


Figure 2. A synthetic scene rendered using path tracing from Jim Kajiya's SIGGRAPH 86 paper "The Rendering Equation."

Kajiya described a process for rendering images according to this equation using a randomized numerical technique known as *path tracing*. Like the earlier fundamental technique of *ray tracing* (Whitted, 1980), path tracing generates images by tracing rays from the camera to the surfaces in the scene, and then traces rays out from these surfaces to determine the incident illumination upon the surfaces. The difference from ray tracing is that these rays are traced not only in the direction of light sources, but also randomly in all other directions, which can account for indirect light from the rest of the scene as well. The demonstration rendering Kajiya produced for his paper is shown in Figure 2. Though a simple scene, Kajiya's example showed realistic light interactions among both diffuse and glossy surfaces, and other complex effects such as light refracting through translucent objects. While still computationally intensive, Kajiya's randomized process for estimating solutions to the rendering equation made the problem tractable both conceptually and computationally.

BRINGING REALITY INTO THE COMPUTER

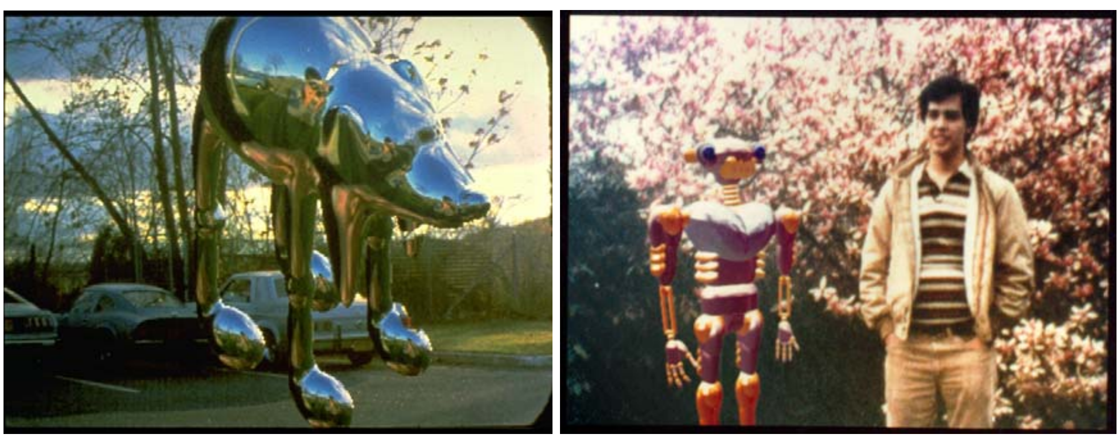
Despite their sophistication, the breakthroughs in rendering techniques of the mid-1980s did not make it a simple endeavor to produce synthetic images with the full richness and realism of images of the real world. In a photograph, the realism that we perceive also comes from the fact that shapes in the real world are typically distinctive and detailed, and that surfaces in the real world reflect light in interesting ways, with different characteristics that vary across surfaces. And also very importantly, light in the real world is interesting because typically there is different color and intensity of light coming from every direction, dramatically and subtly shaping the appearance of the forms within a scene. Computer-generated scenes, when constructed from simple shapes, textured with ideal plastic and metallic reflectance properties, and illuminated by simple point and area light sources, will lack “realism” no matter how accurate or computationally intensive the lighting simulation. As a result, creating photoreal images was still often a matter of skilled artistry rather than of advanced technology, with digital artists manually adjusting the appearance of scenes.

The dearth of realistic geometry in computer-generated scenes was considerably ameliorated in the mid 1980s when 3-D digitizing techniques became available for scanning the shapes of real-world objects into computer graphics programs. The Cyberware 3-D scanner was an important part of this evolution, moving a stripe of laser light across objects and human faces to transform them into 3-D polygon models in a matter of seconds. An early use of this scanner for a motion picture was in *Star Trek IV* for an abstract time-travel sequence showing a collage of 3-D models of the main characters’ heads. 3-D digitization techniques were also used to capture artists’ models of extinct creatures to build the impressive digital dinosaurs for *Jurassic Park*.

DIGITIZING AND RENDERING WITH REAL-WORLD ILLUMINATION

A recent advance in computer graphics realism comes from new techniques that are able to capture illumination from the real world, and to use this illumination to create lighting in computer-generated scenes. If we consider a particular place in a scene, then the light at that place can be described as the set of all the colors and intensities of light coming toward it from every direction. As it turns out, it is relatively straightforward to capture this function for a real-world location by taking an image of a mirrored sphere, which reflects light coming from the whole environment toward the camera. Other techniques for capturing omnidirectional images include using fisheye lenses, tiled panoramas, or scanning panoramic cameras.

The first and simplest form of lighting to be done from images taken in this manner is known as *environment mapping*. In this technique, the image from the mirrored sphere is directly warped and applied to the surface of the synthetic object. The technique was first demonstrated using images of a real scene in the early 1980s independently by Gene Miller and Mike Chou seen in Miller et al., 1984, and Williams, 1983, respectively. Soon after, the technique was used to simulate the reflections on the silvery computer-generated spaceship seen in the 1986 film *Flight of the Navigator*, and most famously, on the metallic T1000 "terminator" character in the 1991 film *Terminator 2*. In all of these examples, the technique not only produced realistic reflections on the computer graphics (CG) object, but also had the desirable effect of making the CG object appear to truly have been within the background environment. This demonstrated the fulfillment of an important cue for realism in visual effects: CG objects should appear to be illuminated by the light of the environment that they are in.



(a) (b)

Figure 3. Environment-mapped renderings from the early 1980s. (a) An environment-mapped shiny dog rendered by Gene Miller (Miller and Hoffman, 1984). (b) An environment-mapped shiny robot rendered by Mike Chou (from (Williams, 1993).

Environment mapping produced convincing results for shiny objects, but new innovations were necessary in order to extend the technique to more common CG models such as creatures, digital humans, and cityscapes. One limitation of environment mapping is that it is unable to reproduce the effects of how object surfaces shadow themselves, or the effects of light reflecting between surfaces. The reason for the limitation is that the lighting environment is applied directly to the object surface according to its surface orientation, irrespective of the degree of visibility that each surface point may have to the environment. For surface points on the convex hull of the object, correct answers can be obtained. However, for more typical points on an object, their appearance depends both by which directions of the environment they are visible to as well as light received from other points on the object.

A second limitation of the traditional environment mapping process is that a single digital or digitized photograph of an environment rarely captures the full range of light visible within a scene. In a typical scene, directly visible light sources are usually hundreds or thousands of times brighter than indirect illumination from the rest of the scene, and both types of illumination must be captured to represent the lighting accurately. This wide dynamic range typically exceeds the dynamic range of both digital and film cameras, which are designed to capture a range of brightness values of just a few hundred to one. As a result, light sources in such images typically become "clipped" at the saturation point of the image sensor, leaving no record of their true color or intensity. For shiny metal surfaces this is not a major problem, since the shiny reflections would become clipped anyway in the final rendered images. However, when lighting more typical surfaces—surfaces that blur the incident light before reflecting it back toward the camera—the effect of incorrectly capturing the intensity of direct light sources in a scene can be very significant.

To solve the dynamic range problem, we used a technique (Debevec and Malik, 1997) to capture the full dynamic range of light within a scene, up to and including direct light sources.

In this technique, photographs are taken using a series of varying exposure settings on the camera, where brightly exposed images record indirect light from the surfaces within the scene, and the dimly exposed images record the direct illumination from the light sources without clipping. Using techniques to derive the response curve of the imaging system (i.e., how recorded pixel values correspond to levels of scene brightness), we can assemble this series of limited dynamic range images into a single high dynamic image representing the full range of illumination for every point in the scene. These High Dynamic Range Images (called HDR Images or HDRI) can be represented using IEEE floating-point numbers for their pixel values, allowing ranges exceeding even one to a million to be captured and stored.

The following year, we presented an approach to illuminating synthetic objects with measurements of real-world illumination (Debevec, 1998) known as Image-Based Lighting (IBL) that addresses the remaining limitations of environment mapping. The first step in IBL is to map the image onto the inside of a surface, such as an infinite sphere, surrounding the object, rather than to map the image directly onto the surface of the object. Then, we use a global illumination system (such as the path tracing approach described by [Kajiya, 1986]) to simulate this image of incident illumination actually lighting the surface of the object. In this way, the global illumination algorithm traces rays from each object point out into the scene to determine what is lighting it. Some of these rays have a free path away from the object and as a result strike the environmental lighting surface, allowing the illumination from each visible part of the environment to be accounted for. Other rays will strike other parts of the object, blocking the light it would have received from the environment in that direction. If the system computes additional ray bounces, the color of the object at this occluding surface point is computed in a similar manner; otherwise, the algorithm approximates the light arriving from this direction as zero. The algorithm sums all of the light arriving directly and indirectly from the environment at each surface point, and uses this sum as the point's illumination. The elegant part of this approach is that it produces all of the effects of how the real object would appear illuminated by the light of the environment, including its self-shadowing, and that it can be applied to any sort of object material, from metal to plastic to glass.

We first demonstrated these techniques of HDRI and IBL in a short computer animation called *Rendering with Natural Light*, shown at the SIGGRAPH 98 computer animation festival (Figure 4, left). The film featured a still life of diffuse, shiny, and translucent spheres on a pedestal illuminated by an omnidirectional, high dynamic range image of the light within UC Berkeley's eucalyptus grove. We later used our lighting capture techniques to record the light within St. Peter's Basilica, which allowed us to virtually add tumbling monoliths and gleaming spheres to the Basilica's interior for our SIGGRAPH 99 film *Fiat Lux* (Figure 4, right). In *Fiat Lux*, we also used the same lighting techniques to compute how much light the new objects would obstruct

Figure 4. Still frames from the animations *Rendering with Natural Light* (left) and *Fiat Lux* (right), showing computer-generated objects illuminated by and integrated into real-world lighting environments.

from hitting the ground, allowing the synthetic objects to cast shadows in the same manner that they would have if they had actually been there.

The techniques of high dynamic range imaging and image-based lighting, and techniques and systems derived from them, are now widely used within the visual effects industry, and have given visual effects artists new lighting and compositing tools for giving the appearance that digital actors, airplanes, cars, and creatures were actually present during filming rather than added in later using computer graphics. Examples of elements illuminated in this way include the transforming mutants in *X-Men* and *X-Men 2*, virtual cars and stunt actors in *The Matrix Reloaded*, and whole cityscapes in *The Time Machine*. In our latest computer animation, we extended the techniques to be able to capture the full range of light of an outdoor illumination environment—from the pre-dawn sky to a direct view of the sun—to illuminate a virtual 3-D model of the Parthenon on the Athenian Acropolis (Figure 5).

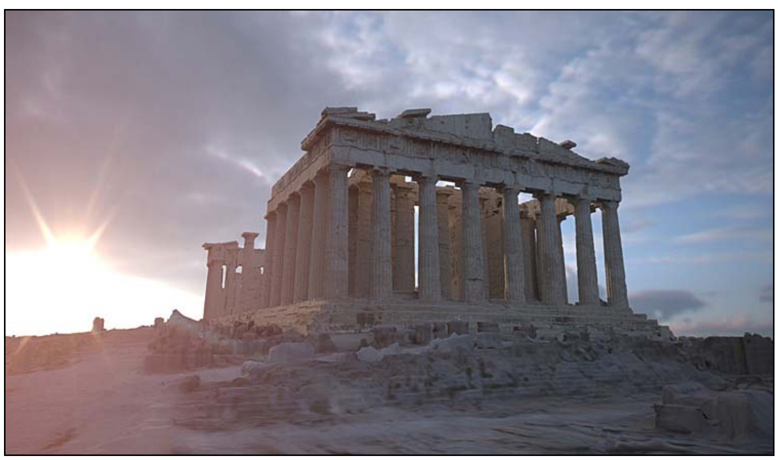


Figure 5. A virtual image of the Parthenon, seen synthetically illuminated with a lighting environment captured at USC ICT in Southern Calfornia.

APPLYING IMAGE-BASED LIGHTING TO ACTORS

In my laboratory's most recent line of work we have examined the problem of illuminating real objects and people, rather than CG models, with light captured from real-world environments. To accomplish this we have used a series of Light Stages to directly measure how an object transforms incident environmental illumination into reflected radiance, which is a dataset that we refer to as the *reflectance field* of an object. The first version of the light stage (Debevec et al., 2000) consisted of a spotlight attached to a two-bar rotation mechanism to rotate the light in a spherical spiral about a person's face in approximately one minute. During this time, one or more digital video cameras record the object's appearance under every form of directional illumination. From this set of data, we can then render the object under any form of complex illumination by computing linear combinations of the color channels of the acquired images. In particular, the illumination can be chosen to be measurements of illumination in the real world (Debevec, 1998) or the illumination present within a virtual environment, allowing the image of a real person to be photorealistically composited into such a scene with the correct illumination.

An advantage of this photometric approach for capturing and rendering objects is that the object need not have well-defined surfaces or easy to model reflectance properties. The object can have arbitrary self-shadowing, interreflection, translucency, and fine geometric detail. This is helpful for modeling and rendering human faces, which exhibit all of these properties, as do many of the objects that we encounter in our everyday lives.

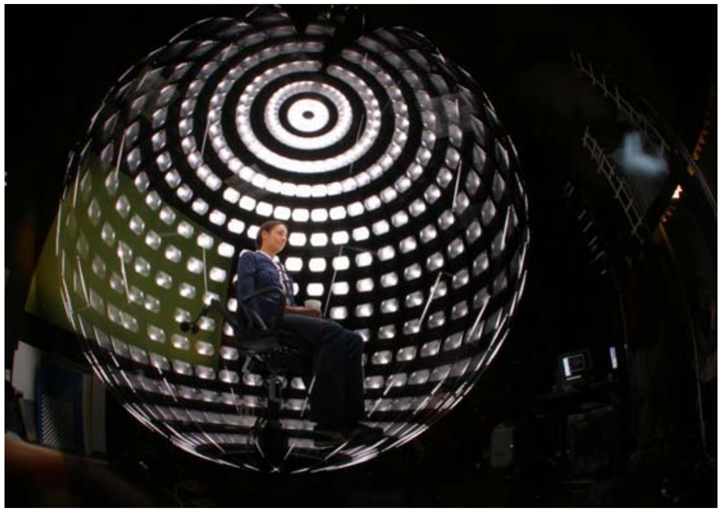


Figure 6. Light Stage 2 (Hawkins et al., 2004) is designed to illuminate an object or a person from all possible directions in a short period of time. This allows a digital video camera to directly capture the subject's *reflectance field*: how they transform incident illumination into radiant illumination. We can then synthetically illuminate the subject under any form of complex illumination directly from this captured data.

Recently, our group has constructed two additional light stages. Light Stage 2 (Figure 6) uses a rotating semicircular arm of strobe lights to illuminate the face from a large number of directions more quickly than Light Stage 1, in about eight seconds. This is a short enough period of time that an actor can hold a steady facial expression for an entire capture session. By blending the geometry and reflectance of faces with different facial expressions, we have been able to create novel animated performances that can be realistically rendered from new viewpoints and under arbitrary illumination (Hawkins et al., 2004). Related techniques were recently employed by Mark Sagar and his colleagues at Sony Pictures Imageworks to create the digital stunt actors of Tobey Maguire and Alfred Molina from Light Stage datasets for the film *Spider-Man 2*.

For Light Stage 3 (Debevec et al, 2002), we built a complete sphere of light sources, able to illuminate an actor from all directions simultaneously. Each of the 156 lights consists of a collection of red, green, and blue LEDs, interfaced to a computer so that any light can be set to any color and intensity. In this way, the stage can be used to reproduce the illumination from a captured lighting environment, using it as a 156-pixel display device for the spherical image of incident illumination. A person standing inside the sphere then becomes illuminated by a close approximation of the light that was originally captured, and when composited over a background image of the environment, appears nearly as if they were there. This technique may improve on how green screens and virtual sets are used today, as actors in a studio can be filmed lit as if they were somewhere else, giving visual effects artists much more control over the realism of the lighting process.

In our latest tests, we have used a high frame camera to capture how an actor appears under several rapidly cycling basis lighting conditions throughout the course of their performance (Debevec et al., 2004). In this way, we can simulate their appearance under a wide variety of different illumination conditions after they have been filmed, allowing directors and cinematographers never-before-available control of the actor's lighting during postproduction.

A REMAINING FRONTIER: DIGITIZING REFLECTANCE PROPERTIES

Significant challenges remain in capturing and simulating physically accurate illumination in computer graphics. While techniques for capturing object geometry and lighting are maturing, techniques are still weak for capturing object reflectance properties—the way that the surfaces of a real-world object respond to light. In a recent project, our laboratory presented a relatively simple technique for digitizing surfaces with varying color and shininess components (Gardner et al., 2003). We found that by moving a neon tube light source across a relatively flat object, and recording the light's reflections using a video camera, that we could independently estimate the diffuse color and the specular properties of every point on the object. For one test object, we were able to digitize a 15th-century illuminated manuscript with colored inks and embossed gold lettering (Figure 7a). Using the derived maps for diffuse and specular reflection, we were able to render a computer graphics version of the manuscript under any sort of lighting environment, and observe realistic glints and reflections from the different object surfaces.

(a) (b) (c)

Figure 7. (a) A digital model of a digitized illuminated manuscript lit by a captured lighting environment (Gardner et. al., 2003). (b) A synthetic model of a face rendered using an efficient simulation of subsurface scattering (Jensen et al., 2001). (c) A digital model of a translucent alabaster horse sculpture recorded and rendered using techniques in (Goesele et al., 2004).

A central complexity in digitizing reflectance properties for more general objects is that the way each point on an object's surface responds to light is a complex function of the incident light direction and the viewing direction—the surface's four-dimensional Bidirectional Reflectance Distribution Function. In fact, the behavior of many materials and objects is even more complicated than this, in that light incident upon other parts of the object may scatter within the object material, an effect known as subsurface scattering (Hanrahan and Krueger, 1993). This effect has received increased interest in the visual effects industry, since this effect is a significant component of the appearance of human skin. New efficient techniques for simulating subsurface scattering effects on computer-generated models (Jensen et al., 2001) have been notably influential in achieving more realistic renderings of computer-generated actors (Figure 7b) and creatures, such as the Gollum character in the *Lord of the Rings*.

Obtaining models of how real people and objects scatter light in their full generality is an open research problem. A recent step forward in this direction (Goesele et al., 2004) used a computer-controlled laser to shine a narrow beam onto every point of a translucent alabaster sculpture (Figure 7c), and recorded images of the object's resulting light scattering using a specially chosen high-dynamic-range camera. By making the simplifying assumption that any point on the object would respond equally to any incident and radiant light direction, they were able to reduce the dimensionality of the problem from eight to four dimensions and acquire a full characterization of the object's interaction with light under these assumptions. As research in this area continues, we will hopefully obtain the capability to digitize anything—no matter what it is made of, or how it reflects light—to become easily manipulated photoreal computer models. For this we look to new acquisition and analysis techniques, and hope that computing power and memory capacity continue to increase.

REFERENCES

Debevec, P.E., and J. Malik. 1997. Recovering high dynamic range radiance maps from photographs. Pp. 369-378 of the Proceedings of the Association for Computing Machinery Special Interest Group on Computer Graphics and Interactive Techniques 97 International Conference. New York: ACM Press/Addison-Wesley Publishing Company.

Debevec, P. 1998. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. Pp. 189-198 of the Proceedings of the Association for Computing Machinery Special Interest Group on Computer Graphics and Interactive Techniques 98 International Conference. New York: ACM Press/Addison-Wesley Publishing Company.

Debevec, P., T. Hawkins, C. Tchou, H. P. Duiker, W. Sarokin, and M. Sagar. 2000. Acquiring the reflectance field of a human face. Pp. 145-156 of the Proceedings of the Association for Computing Machinery Special Interest Group on Computer Graphics and Interactive Techniques 97 International Conference. New York: ACM Press/Addison-Wesley Publishing Company.

Debevec, P., A. Wenger, C. Tchou, A. Gardner, J. Waese, and T. Hawkins. 2002. A lighting reproduction approach to live-action compositing. Association for Computing Machinery Transactions on Graphics 21(3): 547–556.

Debevec, P., A. Gardner, C. Tchou, and T. Hawkins. 2004. Postproduction re-illumination of live action using time-multiplexed lighting. Institute for Creative Technologies Technical Report No. ICT TR 05.2004.

Gardner, A., C. Tchou, T. Hawkins, and P. Debevec. Linear light source reflectometry. 2003. Association for Computing Machinery Transactions on Graphics 22(3): 749–758.

Goesele, M., H.P.A. Lensch, J. Lang, C. Fuchs, and H-P. Seidel. In press. DISCO–Acquisition of Translucent Objects. Paper from the Association for Computing Machinery Special Interest Group on Computer Graphics and Interactive Techniques 31st International Conference, August 8-14, 2004.

Goral, C.M., K.E. Torrance, D.P. Greenberg, and B. Battaile. 1984. Modelling the interaction of light between diffuse surfaces. Association for Computer Machinery Computer Graphics (3): 213–222.

Hanrahan, P., and W. Krueger. 1993. Reflection from layered surfaces due to subsurface scattering. Pp. 165–174 of the Proceedings of the Association for Computing Machinery Special Interest Group on Computer Graphics and Interactive Techniques 1993 International Conference. New York: ACM Press/Addison-Wesley.

Hawkins, T., A. Wenger, C. Tchou, A. Gardner, F. Goransson, and P. Debevec. In press. Animatable facial reflectance fields. Paper from the 15th Eurographics Symposium on Rendering in Norrkoping, Sweden, June 21-23, 2004.

Jensen, H.W., S.R. Marschner, M. Levoy, and P. Hanrahan. 2001. A practical model for subsurface light transport. Pp. 511-518 of the Proceedings of the Association for Computing Machinery Special Interest Group on Computer Graphics and Interactive Techniques 2001 International Conference. New York: ACM Press/Addison-Wesley Publishing Company.

Kajiya, J.T. 1986. The rendering equation. Association for Computing Machinery Special Interest Group on Computer Graphics and Interactive Techniques Computer Graphics. 20(4): 143–150.

Miller, G.S., and C. R. Hoffman. 1984. Illumination and reflection maps: simulated objects in simulated and real environments. Course Notes for Advanced Computer Graphics Animation, Association for Computing Machinery Special Interest Group on Computer Graphics and Interactive Techniques Computer Graphics.

Whitted, T. 1980. An improved illumination model for shaded display. Communications of the Association for Computing Machinery 23(6): 343–349.

Williams, L. 1983. Pyramidal parametrics. 1983. Association for Computing Machinery Special Interest Group on Computer Graphics and Interactive Techniques Computer Graphics 17(3): 1–11.