Advanced Real-Time Illumination Techniques

Naty Hoffman
Westwood Studios
Email: naty@westwood.com

Introduction

All of computer graphics boils down to the simulation of light transport. Real-time graphics has always lagged far behind pre-rendered graphics in the accuracy of this simulation. Real-time methods for handling more accurate reflectance models (BRDFs) have been introduced over the last few years, but reflectance is just part of the rendering equation. Current real-time applications usually model the lighting environment as a collection of point or directional sources, plus an ambient term. Global illumination effects are usually unaccounted for or used only in the most restrictive cases (constant lighting, shadows only, etc.). These limitations must be overcome to achieve real-time photorealism. Fortunately recent advances in algorithms, hardware and APIs now make this possible.

This paper accompanies the GDC lecture "Advanced Real-Time Illumination Techniques". The lecture will discuss several mathematical tools, algorithms and techniques which can be used to handle arbitrary light environments and model complex light / object interactions. This paper will summarize some of the background theory for the lecture.

Radiometry

We will start by defining some basic background concepts. Light is electromagnetic radiant energy. The measurement and study of this energy is called *radiometry*. The total energy over time is averaged, so wave phase or polarization are not taken into account. For example, the total light pouring through a surface section (through all points and in all directions) is measured as energy over time, or power, for which the unit of measurement is the Watt. We call this quantity *radiant flux* (Φ). Another important quantity is the amount of light in a ray (or through a single surface point in a single direction). We call this quantity *radiance* (L), and it is measured as power per area[1] per solid angle[2], for which the unit of measurement is Watt per square meter per steradian. A quantity of particular significance for the important case of diffuse surfaces is *irradiance* (E), which is defined as the incident (incoming) radiant flux per unit surface area. Irradiance is measured as power per area[3], for which the unit of measurement is Watt per square meter.

---

[1] More precisely, projected area, which is area projected perpendicularly to the surface in question. It is equal to surface area times the cosine of the angle between the light direction and the surface normal.

[2] Solid angles are similar to angles in that they can be seen as measuring a continuous range of directions, but they extend the concept to 3D. While angles are defined as arcs on a unit circle and measured in radians (2π in a complete circle), solid angles are defined as patches on a unit sphere and measured in steradians (4π in a complete sphere).

[3] In this case, not projected area.

Radiometric quantities can be measured over the entire electromagnetic spectrum or (more commonly) over a restricted range of frequencies of interest. For rendering purposes we usually only deal with radiant flux in the visible part of the electromagnetic spectrum. However, this flux (and any radiometric quantity derived from it) is distributed continuously over the visible spectrum. How do we get from this spectral distribution to the familiar RGB triplet used for display? The correct method (commonly used in the more physically precise offline rendering applications) is to perform all rendering computations on a dense sampling of frequencies, and to convert to RGB at the end using color matching functions and color-space conversions. This is too expensive for most real-time applications where we want to avoid handling more than three frequency-dependent values. These three values can be obtained either by applying the "correct" conversion to quantities such as colors and light intensities ahead of time, or by simply sampling the spectral distributions at three (R,G,B) sample values. In any case from this point we will simply treat all radiometric quantities as RGB triplets and not worry about spectral distributions.

When rendering, we find the radiance values incident to the center of projection through the pixel centers. These radiance values need to be converted to pixel values for rendering. This conversion involves mapping linear, high dynamic range quantities to output pixel values which have extremely limited dynamic range and are in a non-linear space. The details are beyond the scope of this paper.

The Rendering Equation

The rendering equation was first formulated by Kajiya [1986] (in a slightly different form from the one here).

$$L_{\mathrm{ex}}\left(x,\vec{\Theta}\right)=L_{\mathrm{em}}\left(x,\vec{\Theta}\right)+\int_{\Omega(x)}L_{\mathrm{in}}\left(x,\vec{\Psi}\right)f_{\mathrm{r}}\left(x,\vec{\Theta},\vec{\Psi}\right)\vec{N}(x)\cdot\vec{\Psi}\,d\omega_{\Psi} \tag{1}$$

Where $L_{\mathrm{ex}}\left(x,\vec{\Theta}\right)$ is the exitant (outgoing) radiance from point x in direction $\vec{\Theta}$, $L_{\mathrm{em}}\left(x,\vec{\Theta}\right)$ is the emitted radiance from point x in direction $\vec{\Theta}$, $\vec{N}(x)$ is the surface normal at point x, $\Omega(x)$ is the hemisphere centered on $\vec{N}(x)$, $L_{\mathrm{in}}\left(x,\vec{\Psi}\right)$ is the incident radiance at point x in direction $\vec{\Psi}$, $f_{\mathrm{r}}\left(x,\vec{\Theta},\vec{\Psi}\right)$ is the BRDF (Bidirectional Reflectance Distribution Function) at from point x evaluated for incident direction $\vec{\Theta}$ and exitant direction $\vec{\Psi}$, and $d\omega_{\Psi}$ is the infinitesimal solid angle over which $\vec{\Psi}$ is integrated.

This form of the equation does not take account of translucency of any kind, or participating media, but is otherwise general. We will next examine the deficiencies in the common real-time approximations to this equation.

Lighting Environments

The rendering equation integrates incident radiance from all directions. In real scenes, there are different radiance values for each incident direction, forming a continuous $L_{\mathrm{in}}\left(x,\vec{\Psi}\right)$

function we will call the *lighting environment* for point x. Current real-time rendering models assume that the lighting environment is zero everywhere except for a limited number of delta functions[4] representing ideal directional or point lights. This model has the advantage of reducing the integral in equation 1 to a (small) summation. However, it has several deficiencies.

The first deficiency is that the majority of the light environment is zero, causing some surfaces to receive little or no light. Since this usually leads to unacceptably dark lighting in parts of the scene, applications using this lighting model usually add an ambient term. Note that this ambient term is not the same as adding a constant non-zero incident radiance to the light environment (which would be a reasonable thing to do). The ambient term is a constant which is added directly to the result of the rendering equation. This is very incorrect from a physical standpoint, and leads to extremely “flat” lighting which severely harms realism.

Another deficiency is that delta functions are physically implausible. Real light sources have finite radiance and subtend a non-zero solid angle. Delta functions in the light environment lead to lighting which is unnaturally sharp.

The most important deficiency is that this lighting model is extremely limited compared to the lighting found in actual scenes. It is very rare that an object is lit by a very small number of tiny, high-intensity lights with little or no incident light from other directions. Even in outdoor scenes with strong sunlight, objects are lit by reflected light from their surroundings, by skylight, etc. In indoor scenes objects are typically lit mostly (often exclusively) by indirect lighting, with complex continuous color shifts and intensity changes over the light environment. These rich lighting environments illuminate objects in a way which cannot be achieved by current real-time methods.

There is one exception to the lack of general lighting environments in real-time rendering - environment maps. These are typically used in the special case of an ideal mirrored surface, and demonstrate the potential complexity of real lighting environments compared to the typical “four point light + ambient” model. In the past few years, environment maps have been extended to other types of surfaces [Heidrich and Seidel 1999; Kautz and McCool 2000; Kautz et al. 2000]. However, drawbacks remain to representing the light environment as an environment map; such maps are memory-intensive, so it is hard to store many of them for different locations in a scene. Also, many common operations (such as interpolation) are computationally expensive when performed on environment maps.

Global Illumination

Real-time graphics usually uses the *direct illumination* model, where the value of $L_{\text{ex}}\left(x,\vec{\Theta}\right)$ is dependent only on light sources and the local properties of point x - other points in the scene have no effect. This model has a large performance advantage in that it bypasses the recursive nature of the rendering equation and enables a closed-form solution. In the general

---

[4] A delta function is a function with zero extent and infinite height (in this case a point with infinite radiance) such that the integral over the function results in a finite quantity.

case, to solve the rendering equation and find the exitant radiance from a point x, we need to first find the incident radiance from all directions. Some of these require first finding the exitant radiance from other surface points in the scene. The exitant radiance from these other points depends on yet other points (possibly including the original point x), and so on. This causes algorithms which perform complete global illumination solutions to have an iterative nature – no closed-form solution is possible.

However, the direct illumination model has severe drawbacks in terms of realism. In real scenes, light bounces multiple times between various surface points before reaching the eye. In many cases, such indirect illumination may contribute considerably more to the total lighting solution than the direct illumination does. Since use of the direct illumination model usually leads to unacceptable darkness in parts of the scene (similarly to the “several point lights” model mentioned in the previous section), the “ambient value” hack is often used to simulate indirect lighting. We have already mentioned the drawbacks of this in the previous section.

There is one exception in recent years to the “direct illumination” rule for real-time applications. Shadows are a global illumination effect (since they involve parts of the scene occluding light from other parts) and are often simulated, typically using shadow mapping [Williams 1978; Reeves 1987; Segal et al. 1992; Stamminger and Drettakis 2002] or stencil shadow volumes [Crow 1977; Heidmann 1991; Everitt and Kilgard 2002]. These methods have their own drawbacks – they cannot easily do soft shadows, and they cannot cast shadows from details which are modeled in textures rather than in geometry (such as bump maps). There has been some recent work on shadows from bump maps [Sloan and Cohen 2000; Heidrich et al. 2000], but we are interested in solutions which cover all global illumination effects; interreflections as well as shadows.

Advanced Real-Time Illumination Techniques

There are several techniques which overcome the limitations above and enable photorealistic real-time scenes. We will go into much more depth on these techniques in the accompanying lecture – here we will only summarize them.

Polynomial Texture Maps [Malzbender et al. 2001] use biquadratic polynomials to approximate the change in surface lighting as a point or directional light source moves over a surface. The polynomials are fitted to data which may come from photographs or from high-quality renders of a dense model. The coefficients are stored in texture maps (six coefficients per pixel). Since this technique in its basic form does not take account of view-dependence (though there are some extensions for that in the paper), it is best suited for diffuse surfaces. However, other than that, there is no limit to the effects that can be modeled – interreflections, sub-surface scattering, etc. In some cases, the polynomial may not be a very good fit to the data, but in practice the technique produces very convincing images. PTMs provide a full global illumination solution (to be precise, they provide a good approximation to one). They are limited to a small number of light sources, so they do not enable general light environments.

Ramamoorthi and Hanrahan [2001a; 2001b] introduced a new approach to lighting which represents the light environment and the BRDF in frequency space using spherical harmonic basis functions, and treats the integral in equation 1 as a convolution between the SH coefficients of the two. This leads to several interesting results. In [Ramamoorthi and Hanrahan 2001c] it is shown that for diffuse surfaces, nine spherical harmonic coefficients are sufficient to approximate any arbitrary light environment to a high degree of accuracy (in [Ramamoorthi 2002] it is shown that in certain cases where the normals lie in a hemisphere as few as five coefficients may suffice). They then develop a closed-form formula (incorporating the SH coefficients as constants) which takes the world-space normal as input and directly calculates the rendering integral. This formula is only moderately more expensive than the traditional "four light + ambient" computation and can be used with either per-vertex or per-pixel normals. This work is very interesting since it enables real-time apps to easily extend to much richer light environments that those currently used. However, it is strictly a direct illumination solution and does not take account of shadowing or interreflections. In [Ramamoorthi and Hanrahan 2002] this work is extended to handle arbitrary BRDFs.

Starting from the same theoretical basis, [Sloan et al. 2002] use spherical harmonic coefficients to represent not only the lighting environment, but also radiance transfer functions which are precomputed over the surface of an object. These can represent any view-independent global illumination phenomena, in which they are similar to polynomial texture maps. Rendering is then simply a matter of performing dot-products between the light and transfer coefficient vectors, thus computing the rendering integral. It is shown that in most cases, fifth-order SH (25 coefficients) suffice to represent the radiance transfer to a high degree of accuracy. In some cases (depending on the object and the light environment) as few as 16 or even 9 coefficients can be used. One limitation of this technique is that it cannot handle completely arbitrary light environments – if the light environment has very high-frequency components then errors will be introduced. However, the result will be the same as if the light environment had been low-pass filtered previous to rendering, which can be quite acceptable in many cases. The paper also shows how to handle glossy surfaces, but in the general case those cannot be rendered in real-time. In [Kautz et al. 2002] the method is extended to arbitrary BRDFs, but again the result is not quite ready for real-time rendering.

The lecture will explain these techniques in detail, and will show how they can be used in the context of a game rendering engine.

References

CROW, F. 1977. Shadow Algorithms for Computer Graphics. In *Computer Graphics (Proceedings of ACM SIGGRAPH 77), 11*, 3, ACM, 143-150.

EVERITT, C. AND KILGARD, M. 2002. Practical and Robust Shadow Volumes, NVIDIA developer web site, http://developer.nvidia.com/view.asp?IO=robust_shadow_volumes

HEIDMANN, T. 1991. Real Shadows Real Time, *IRIS Universe,* 18, 28-31. (http://developer.nvidia.com/view.asp?IO=robust_shadow_volumes)

HEIDRICH, W. AND SEIDEL, H-P. 1999. Realistic, Hardware-accelerated Shading and Lighting. In *Proceedings of ACM SIGGRAPH 99*, ACM Press / Addison-Wesley Publishing Co. / ACM SIGGRAPH, New York. A. Rockwood, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 171-178. (www.cs.ubc.ca/~heidrich/Projects/HWShading/)

HEIDRICH, W., DAUBERT, K., KAUTZ, J., AND SEIDEL, H-P. 2000. Illuminating Micro Geometry Based on Precomputed Visibility. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH, New York. K. Akeley, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 455-464. (http://www.mpi-sb.mpg.de/~jnkautz/projects/shadowbumpmaps/index.html)

KAJIYA, J. 1986. The Rendering Equation. In *Computer Graphics (Proceedings of ACM SIGGRAPH 86), 20*, 4, ACM, 143-150.

KAUTZ, J. AND MCCOOL, M. 2000. Approximation of glossy reflection with prefiltered environment maps. In *GI 00 (Proceedings of Graphics Interface 2000)*, 119-126. (http://www.mpi-sb.mpg.de/~jnkautz/publications/index.html)

KAUTZ, J., VÁZQUEZ, P-P., HEIDRICH, W., AND SEIDEL, H-P. 2000. A Unified approach to Prefiltered Environment Maps. In *Rendering Techniques 2000 (Proceedings of the 10th Eurographics Workshop on Rendering)*, 185-196. (http://www.mpi-sb.mpg.de/~jnkautz/projects/unifiedenvmaps/index.html)

KAUTZ, J., SLOAN, P-P., AND SNYDER, J. 2002. Fast, Arbitrary BRDF Shading for Low-Frequency Lighting Using Spherical Harmonics. In *Rendering Techniques 2002 (Proceedings of the 12th Eurographics Workshop on Rendering)*, 301-308. (http://research.microsoft.com/~ppsloan/)

MALZBENDER, T., GELB, D., AND WOLTERS, H. 2001. Polynomial Texture Maps. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, New York. E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 519-528. (http://www.hpl.hp.com/ptm/)

RAMAMOORTHI, R. AND HANRAHAN, P. 2001. On the Relationship Between Radiance and Irradiance: Determining The Illumination From Images of a Convex Lambertian Object, *Journal of the Optical Society of America (JOSA A)* Oct 2001, 2448-2459. (http://graphics.stanford.edu/papers/invlamb/)

RAMAMOORTHI, R. AND HANRAHAN, P. 2001. A Signal Processing Framework for Inverse Rendering. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, New York. E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 117-128. (http://graphics.stanford.edu/papers/invrend/)

RAMAMOORTHI, R. AND HANRAHAN, P. 2001. An Efficient Representation for Irradiance Environment Maps. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, New York. E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 497-500. (http://graphics.stanford.edu/papers/envmap/)

RAMAMOORTHI, R. 2002. Analytic PCA Construction for Theoretical Analysis of Lighting Variability in Images of a Lambertian Object, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Oct. 2002, 1322-1333. (http://graphics.stanford.edu/papers/pca/)

RAMAMOORTHI, R. AND HANRAHAN, P. 2002. Frequency Space Environment Map Rendering. *ACM Transactions on Graphics, 21*, 3, 517-526. (http://graphics.stanford.edu/papers/freqenv/)

REEVES, W. 1987. Rendering Antialiased Shadows with Depth Maps. In *Computer Graphics (Proceedings of ACM SIGGRAPH 87), 21*, 4, ACM, 283-291.

SEGAL, M., KOROBKIN, C., VAN WIDENFELT, R., FORAN, J., AND HAEBERLI, P. 1992. Fast Shadow and Lighting Effects Using Graphics Hardware. In *Computer Graphics (Proceedings of ACM SIGGRAPH 92), 26*, 2, ACM, 249-252.

SLOAN, P-P. AND COHEN, M. 2000. Interactive Horizon Mapping. In *Rendering Techniques 2000 (Proceedings of the 10th Eurographics Workshop on Rendering)*, 281-298. ((http://research.microsoft.com/~ppsloan/)

SLOAN, P-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. *ACM Transactions on Graphics, 21*, 3, 527-536. (http://research.microsoft.com/~ppsloan/)

STAMMINGER, M. AND DRETTAKIS, G. 2002. Perspective Shadow Maps. *ACM Transactions on Graphics, 21*, 3, 557-562. (http://www-sop.inria.fr/reves/publications/data/2002/SD02)

WILLIAMS, L. 1978. Casting Curved Shadows on Curved Surfaces. In *Computer Graphics (Proceedings of ACM SIGGRAPH 78), 12*, 3, ACM, 270-274.