From Local to Global Illumination and Back

ALAIN FOURNIER

Department of Computer Science, University of British Columbia
Vancouver, BC, V6T 1Z4, Canada. fournier@cs.ubc.ca

Abstract: The following being musings about illumination problems and illumination answers, more particularly about the evolution and the interplay between local and global illumination concerns.

Keywords: global illumination, local illumination.

1. Introduction

Alchemists[1] had a saying "*What is above is like what is below, and what is below is like what is above*". It could be interpreted as a general statement about the self-similarity of the universe, but I prefer for now to see it as an affirmation that one can and should not separate global and local illumination too much, because they are not that different (note I do not claim that they are identical).

Let's first look at some hierarchy of things rendered (Figures 1 and 2). There are many other ways to describe and arrange this, but the basic message is that we build global behaviour from basic elements (*primitives*). So at first glance there is not much difference between computing the *bidirectional reflectance distribution function* (BRDF) of a surface from basic elements or computing the global illumination in a room to extract a view-independent light distribution. In fact one can easily conceive of a *bidirectional room reflectance distribution function* (BRRDF) which for any directional light source gives the reflected light towards an eye at infinity (either for every visible point in that direction, or just an average radiance)[2]. A lot has already been said about these hierarchies, either explicitly or implicitly, but I will mention first some general issues, and then discuss some specifics related to the work I am currently involved with.

1.1. Tight and Loose Coupling

A big difference (maybe the only valid one considering local vs global illumination) is the amount of coupling between elements contributing to the illumination. In local cases, one generally assumes that they don't move with respect to each other, and therefore a pre-computation of the results (again often in the form of a BRDF) is quite

1. Alchemists, at least the serious ones, were interested in the secrets of life and achieving immortality. Turning base metals into gold was only a side effect. Serious scientists still feel the same.

2. You could object that the reflected light could come out far from its point of incidence. But I would ask: what do you think the r is for in the $f_r()$ notation for the BRDF? Many references forget about that. Glassner, in his wonderful (if heavy) new book [9] gets it right.

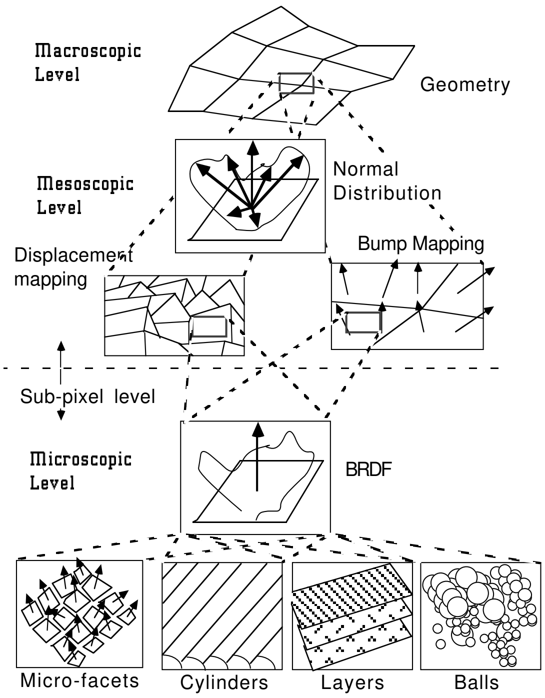

Figure 1. The main hierarchy of local illumination

reasonable. The reason BRRDF are not too popular so far is that it is not worth the investment in most applications, especially if one often rearranges the furniture. The unfortunate (and challenging) situation is with these "local" cases where the elements indeed move with respect to each other (leaves in a tree, hair in a fur, small scale waves on water, etc..). Some work in that direction has been done (the paper by Krueger [14] was an excellent start, though time was not a parameter) but more is needed (see below).

1.2. Is It Linear?

Linear in this context means that all the operators $\mathbf{O}(I)$ on light intensity (or any quantities proportional to the light power) are linear:

$$\mathbf{O}(\ a\ I_1 + b\ I_2) = a\ \mathbf{O}(I_1) + b\ \mathbf{O}(I_2)$$

This is a basic (and not always stated) assumption of every illumination computation so far, in particular *ray-tracing* and *radiosity*. Examples of real non-linear light effects range from the universal (but usually negligible) such as the variation of BRDF as a function of temperature (which is itself a function of the amount of light absorbed) to the more exotic, such as the photo-cell on my TV set which adjusts the brightness of

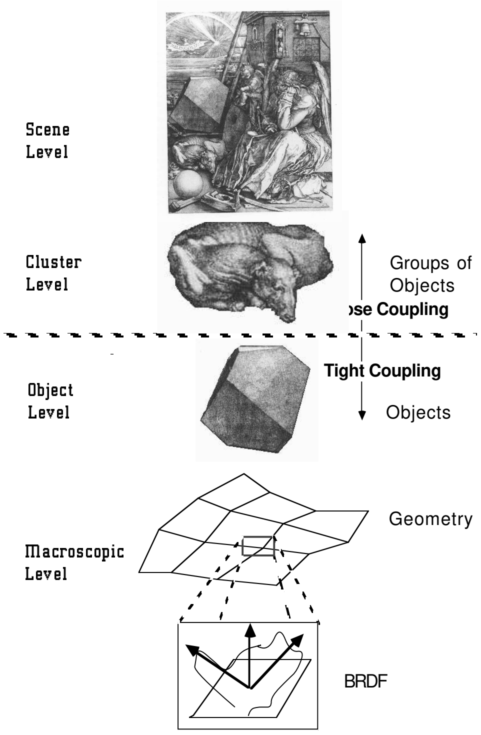

Figure 2. The main hierarchy of global illumination

the CRT as a function of the ambient light (fortunately for my eyes it is not a linear feed-back). If reflectance were not generally linear, we of course would be in deep trouble to compute most of anything.

1.3. Does It Scale

If the subtext is to use global illumination techniques to compute local illumination effects or the other way around, we have to make sure that our models and computations are independent of scale (or that we know how to scale them) in spatial terms. Let's have a quick glance at the *rendering equation*, here given in the notation of Cohen and Wallace [4]:

$$L(\mathbf{x}', \omega') = L_e(\mathbf{x}', \omega') + \int_S f_r(\mathbf{x}', \omega, \omega') L(\mathbf{x}, \omega) G(\mathbf{x}, \mathbf{x}') V(\mathbf{x}, \mathbf{x}') \, dA$$

where $G()$ is a geometric factor and $V()$ the visibility factor.

The radiances are *intensive* quantities, that is they are scaleless (not dimensionless, of course), and the BRDF has a dimension of sr^{-1}, so the only spatial extensive quantities are in $G()$ and $V()$. For $V()$ it is clear that any reasonable definition of visibility should be scaleless. For $G()$ one has to look a little more closely at the integral, but lengths appear only explicitly square in dA and square in the denominator of $G()$, so they do cancel out. In the classic radiosity simplification, this is obvious, since the *form factors* are scaleless and dimensionless.

1.4. Compared to What

A relatively recent trend in illumination computation is to obtain formal error bounds on the results or to assess the images by experimental comparisons. The work of James Arvo, Dani Lischinski, Gershbein/Schröder/Hanrahan for the former, and Holly Rushmeyer and Greg Ward for the latter is especially notable, and show a degree of rigour and sophistication unequaled in any other area of computer graphics (of course IMHO).

There is a *caveat*, though. Since most illumination algorithms are based either on diffuse reflectors, which in reality are few and far between (see remark and reference "[18]" in [11]) or on "un-natural" specular reflection, since the models of luminaires used are far the real ones, and since the real ones are rarely measured with any accuracy (see for example Ashdown [1]) a bound on the error for the computation does not mean much in terms of difference between the result and real scenes. Of course we have to start somewhere, but it's good not to lose sight of the goal. As an exercise, consider what is farther from the truth: a rendering with inter-reflection but only diffuse reflectors, or a scene with realistic reflectors but no inter-reflection? Assume three exterior scene: Andrew Wyeth's "Christina's World", Claude Monet's "Cathédrale de Rouen" (any one) and Hiroshige's "Kanbara" (from *Fifty-three stages of the Tokaido*). Now repeat for three interior scenes: Hopper's "The Night Hawks", Vincent Van Gogh's "Bedroom in Arles" and Rembrandt's "The Night Watch".

2. Act Locally

Indeed the only way the light acts (at our rendering level, anyway) is locally. For this discussion I will largely ignore participating media, but even for these it all starts locally.

2.1. Models of Microstructure

A lot of reflection models, starting with Bouguer and Lambert, continuing with the Torrance/Sparrow model, or in more recent work by Hanrahan and Krueger[11], Gondek *et al*[10] or Oren and Nayar[19] was developed by considering a simple (or not so simple) assemblage of primitives[3]. Similarly to improve the generality of illumination models, Xiao *et al* [12] started from Kirchhoff equation for the light reflected at a surface (approximated by its tangent plane at the point considered) and added interreflection and self-shadowing/masking factors derived from statistical techniques (the surface height is assumed to follow a Gaussian distribution). In [20] we used a "hidden" level of geometry made of cylinders to allow the computation of anisotropic reflection. A similar approach is effective for modelling cloth, since the

---

3. Pat Hanrahan and Julie Dorsey recently devised models simulating corrosion and aging processes with layers, inverting the alchemists' dream and turning gold into base metal.

the geometry of the weave (or even the thread) is well known and available.

While all these techniques are effective within their intented scale, a major unsolved problem is to allow their simultaneous use, and in particular ensure smooth transition between models when the scale changes. Another way to formulate the same problem is that one should be able to find a way to represent the average effect of one level in terms of the level above. As it turns out, one very constraining, though unnecessary, assumption made in all the local models is that a single normal at each sampled point or area has to account for all the reflection effects.

2.2. Local or Parochial[4]

Most fractal objects are easy to render if we assume their surface is fully realized. Let us take as a simple example a fractal surface as a sample of two-variable fractional Brownian motion $fBm(x, y) = z$. For any point **P** on the surface to be visible from the eye (or from a point on a light source), a line segment from that point to the eye cannot intersect the surface. (see Figure 3).

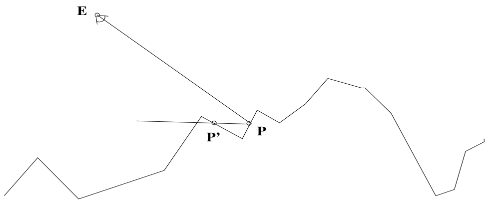

Figure 3. A fractal surface cannot be seen.

It means that for any point **P′** on the surface the segment **PP′** has a slope less than the slope of **PE**. As **P′** gets closer to **P**, however, that means that there is a limit to the slope of **PP′**, which contradicts the property of fBm that the limit slope goes to ∞ with probability 1 [15]. Therefore no point on the surface can be seen from the eye or from the lights (exercise left to the reader, what if the line **PE** is parallel to the Z axis?). This frees us from the worry of designing an illumination model for such surfaces[5]. Also note that it is not in contradiction with the fact that one standing on the surface would see a finite horizon, as long as the observer is a finite height above the surface. I convinced John Hart of the above (I think) but he said: "What about non-random fractals, like Koch island?". I leave that as an exercise to the reader.

2.3. Bump Maps and Filtering

Bump maps is the traditional (if inelegant) name given to texture maps when the values (discrete values of texture maps are often called *texels*) are used to define or modify the normal to the surface prior to shading computations. They were originally

4. I have said that before, but I can’t resist.

5. Should you think it does not apply to the real world, recall that one can make approximations of a blackbody by stacking tightly together several old-fashioned double-edged razor blades.

introduced by Blinn [2] and have been widely used since. Unfortunately the traditional filtering techniques do not apply and they have not been successfully filtered yet. The reason is straightforward. All the pre-filtering techniques developed so far rely on the fact that they filter some quantity that can be factored out of the shading computation when the result is averaged over an area or discrete samples. The rendering equation for such surfaces over some finite area A becomes, if A_i is an element of A with a given normal direction (assuming for now they do not emit light):

$$L(\mathbf{x}', \omega') = \frac{1}{\sum_i A_i} \sum_i \int_{all\ A_i} dA_i \int_S f_r(\mathbf{x}', \omega, \omega') L(\mathbf{x}, \omega) G(\mathbf{x}, \mathbf{x}') V(\mathbf{x}, \mathbf{x}')\ dA$$

Any quantity that enters only linearly in the integrals can be brought out and integrated separately. For each element A_i, unfortunately, there is a different normal, and therefore nothing can be brought out of the second integral since every term is a function of the local frame, and therefore of the normal direction.

As a trivial counter-example, consider the case where the normals over the area considered are of two kinds, symmetric with respect to the average normal and equally distributed. The average of the normals will be the normal of the average plane, and this will cause highlights where the real surface has none, and not show highlights where the real surface has strong ones.

2.4. Reconstructing a Distribution of Normals

The simple example above suggests a solution. If we somehow knew that the surface has only two values of normal vectors equally distributed, these two normals alone are enough to compute the reflected light at any scale. In general, we can look at the normals over the area as defining an underlying distribution of normals $\mathbf{N}(\theta, \phi)$ defined for each direction (θ, ϕ) on the hemisphere . The problem is to approximate this distribution by a weighted linear sum of a small number of discrete functions, each function being used then to compute their contribution to the total reflected light. One could use a variety of techniques to accomplish this. The method chosen and implemented is to represent the reconstructed distribution of normals as a weighted sum of a small number of *Phong peaks*, that is of functions of the form:

$$\mathbf{N}(\theta, \phi) = \mathbf{U}(\theta, \phi) \sum_{i=0}^{n-1} a_i \cos^{n_i}(\alpha_i)$$

where $\mathbf{U}(\theta, \phi)$ is the unit vector in the direction (θ, ϕ), α_i is the angle between the direction (θ, ϕ) and the direction of peak i: (θ_i, ϕ_i). Each peak has therefore 4 variables to fit, and non-linear least squares are used to compute the fit.

The original function $\mathbf{N}\,()$ is computed by *spreading* the normals given by the initial bump map, or the relevant section of it. Spreading consists in computing for every point on the hemisphere the sum of the density of normals given by the Phong $\cos^n(\alpha)$ formula. In effect the Phong function serves as a reconstruction filter for the definition of $\mathbf{N}\,()$. This process is carried at various levels to achieve filtering, in a manner now familiar from *MIP maps* [21] and *NIL maps* [7].

2.5. Multiple Surfaces

The building of the pyramid has a not totally unexpected side benefit: at each texel of each level of the pyramid one gets a local illumination model, and at the top level one gets a model valid when all of the original bump map is comprised within the local "sample". The model is expressed simply as a surface with more than one normal per sample, which we can call a *multiple surface*. The description of the multiple surface implies a new BDRF extracted from the bump map. The BDRF can be obtained by computing the reflected intensity for every direction of incoming and reflected light. It is not as general as an arbitrary BDRF but it has the advantages of being very compact and made of simple elements, namely Phong peaks. This makes the shading computation with this model a trivial modification of the computation normally done with a single Phong peak. Of course values for multiple surfaces do not have to be extracted from bump maps. One can deduce them in simple cases, such as a sphere or a cylinder, or invent them for "special effects".

2.6. Shadowing, Masking and Other Bothers

When the surface structure is really three-dimensional some of the details block other from the light (*self-shadowing*) or from the view (*masking*, or *self-blocking*). It is important to model these effects for convincing rendering. Textures simulating hair or fur can be geometrically adequate, for example, but not very realistic if masking and shadowing are missing. Such problems, and solutions, has been addressed notably by Max and his collaborators [17][3]. I don't have the space here to describe our solution, but it is simple, slightly sleazy and effective.

We have in all obtained an effective technique to filter bump maps, bridging the gap between bump maps and local illumination models. It lead us to a simple local illumination model where each point on a surface can have multiple normal vectors. We have also developed a technique to compute and filter the effect of self-shadowing and masking, again from basic and compact information extracted from the geometric model of the microstructure.

The relationship between BDRF and the distributions of normals we compute are also to explore. One issue is to characterize the BDRF which cannot be effectively represented that way, another is to try to extract automatically multiple surface characteristics from experimentally measured BDRF. Work on this and in general efficient representation of BRDF for direct use in renderers is being pursued by Paul Lalonde and me. Again, most surfaces are not static. The surface of water, as an important example, moves constantly. Statistical approaches to the computation of the light reflected by water exist [14] but are not well suited to animation. As another example, my initial interest in the problem of filtering surfaces arose in the context of stochastic modelling [6] where a partially evolved surface "stands in" for the fully subdivided one. In this case the shading of each polygon should be the average of the shade of all the surface details not yet produced. These two problems have similar elements and are shimmering targets above the horizon.

3. Think Globally

3.1. Whither Fredholm?

Let's look again at the rendering equation:

$$L(\mathbf{x}', \omega') = L_e(\mathbf{x}', \omega') + \int_S f_r(\mathbf{x}', \omega, \omega') L(\mathbf{x}, \omega) G(\mathbf{x}, \mathbf{x}') V(\mathbf{x}, \mathbf{x}')\, dA$$

Do we really have a Fredholm's equation of the second kind? Even since my initiation to recursive procedures, I have been mildly upset at the fact that the most commonly given first example is the computation of the factorial, which can be immediately seen as your basic iteration. Indeed in general *tail recursion* can be unrolled to be replaced by iteration. The rendering equation in a sense can be unrolled. The radiance of an element of surface we want to compute (the left hand side of the redering equation) appears within the integral only "hardly", and in fact the light it receives "from itself" can be decoupled by just storing it on the way to and from all other surfaces. This is the basic approach of *Lucifer* (more below), and it is why we can solve the equation without having to invert the integral operator applied to the radiance.

In his initial paper discussing the rendering equation [13] Jim Kajiya mentioned Neumann series as a method to solve integral equations, and indeed there is a beautiful correspondence between the terms in the series and common illumination algorithms. Fredholm himself also has a series solution (using the *resolvant kernel*[16]). The question is, what is the "physical" meaning of the terms in this kernel. I hope Jim Arvo can answer this, because I can't now.

3.2. Lucifer

Most of the drawbacks of radiosity come from the fact that it is *element-driven*, that is consider specifically pair-wise exchanges of light between elements, and from the fact it really works only for separable sources and reflectors (one can indeed go a long way with that, see my paper in these proceedings on separable BRDFs). It is at heart a global illumination solution which critically depends on a limiting assumption about the local behaviour of light. Hybrid solutions also invariably have problems because there is no neat dichotomy between specular and diffuse reflection, and real reflection/refraction is a continuum between these two extremes. That does not mean of course that there is no hope in that direction. L. Neumann and A. Neumann have a very interesting paper surveying the situation in this respect [18].

An alternative is a *light-driven* approach, where the global illumination problem is solved by propagating the light from sources (we use "source" in the broadest sense of anything radiating light) to other parts of the environment to the eye. Another important criterion is that the amount of effort spent on each part of the scene is somehow related to how much light will come from it. There is no need to compute much about the whole content of a closet whose door remains closed, even less need to compute anything if all the lights are off. What I am advocating, therefore, is a *light-driven, volume-oriented* approach.

To summarize the paradigm, consider a volume **V** within the environment to render, with a boundary surface *S*. If we know for every point of *S* the flux of light crossing it in any direction, then we can study separately the illumination inside and outside of **V**. Furthermore, even if we do not know the true situation at the boundary, but only some amount of light emitted inside **V** and the amount of light coming into **V**, and if we know how to solve the global illumination problem inside **V**, then we can assign to points of *S* the correct amounts of outgoing light. The outside of **V** can then be treated without considering **V** unless it is found that more light comes into **V** from outside. In this case we can solve the global illumination problem inside **V** again *independently* of the previous solution if we assume (and it is the only serious restriction) the *linearity* of the light effects. If we cannot solve the global illumination problem for **V**, we can partition it into two or more sub-volumes, and so on recursively until each section is simple enough so that we can deal with it.

3.3. So What?

Given this paradigm, one has to chose the modalities of implementation. The choices concern the volumes to use, the representation of light flux at the volume boundaries, the rendering techniques, etc.. The choice of volumes is the easiest. Octree cells give simple shapes, hierarchical decomposition of space (nothing much to do if there is nothing there), and adaptability (one is free to do anything reasonable at the level of a leaf of the octree (of course one decides when it *is* a leaf). To represent the light flux is trickier. The original implementation [8][5] used a total discretization of the walls of the *cells* (the cubes of the octree) and the directions. This was quite ugly, but that's a start and a straw-person against which to compare.

Now *wavelets* came to the rescue, and that what we are using (in current work by Bob lewis and myself) for the light flux. They have most of the qualities needed, compactness, respect of discontinuities (important for shadows, in particular) and the "natural" ability to "cluster" the light flux automatically (that is part and parcel of their use in image compression). In this application we need four-variable wavelet transforms.

We have developed the algorithms necessary to compute the propagation through empty cells, the blocking and the reflection while using only a number of operations of the order of the number of wavelet coefficients in the incoming and reflected/transmitted light flux. We hope to be able soon to convince a reluctant world that this approach is not only valid, but actually will eventually perform better than most others, because it is adaptable, accept any reflecting behaviour, is easily parallelizable, and has the asymptotic complexity of the Z-buffer (that is linear in the number of elements in the scene). So what if the constant is *very* large.

Finally, it ties everything back together, because of course the same transform used for the light flux on the cell walls can be used to represent BRDF (which again is done with Paul Lalonde) and we are right back to local illumination.

4. Conclusion

When it seems that it finally all comes together, maybe it is because we are standing too close to a black hole. Nevertheless that is how it feels. Thanks you.

Acknowledgements

I am very thankful and truly honoured to have been invited to address this workshop. I gratefully acknowledge the support of NSERC through operating grants and graduate scholarships. IBM and Silicon Graphics were generous with donations of workstations to keep graduate students happy. Many people helped me and inspired me in my research,but one pontificates alone.

References

1. I. Ashdown, ‘‘Near-Field Photometry: A New Approach,’’ *Journal of the Illuminating Engineering Society of North America*, vol. 22, no. 1, pp. 163-180, 1993.
2. J.F. Blinn, ‘‘Simulation of Wrinkled Surfaces,’’ *Computer Graphics*, vol. 12, no. 3, pp. 286-292, August 1978.
3. B. Cabral, N. Max, and R. Springmeyer, ‘‘Bidirectional Reflection Functions from Surface Bump Maps,’’ *Computer Graphics*, vol. 21, no. 4, pp. 273-281, July 1987.

4. M.F. Cohen and J.R. Wallace, *Radiosity and Realistic Image Synthesis,* Academic Press, 1993.

5. G. Drettakis, E. Fiume, and A. Fournier, *Tightly-Coupled Multiprocessing for a Global Illumination Algorithm,* pp. 387-398, Eurographics ’90, September 1990.

6. A. Fournier, D. Fussell, and L. Carpenter, “Computer Rendering of Stochastic Models,” *Comm. of the ACM*, vol. 25, no. 6, pp. 371-384, June 1982.

7. A. Fournier and E.L. Fiume, “Constant-Time Filtering with Space-Variant Kernels,” *Computer Graphics*, vol. 22, no. 4, pp. 229-238, August 1988.

8. A. Fournier, E. Fiume, M. Ouellette, and C. K. Chee, “FIAT LUX: Light Driven Global Illumination,” DGP Technical Memo DGP89-1, 1989.

9. A.S. Glassner, in *Principles of Digital Image Synthesis*, Morgan Kaufmann, 1995.

10. J. S. Gondek, G. W. Meyer, and J. G. Newman, “Wavelength Dependent Reflectance Function,” *Computer Graphics Proceedings (Siggraph ’94)*, pp. 213-220, 1994.

11. P. Hanrahan and W. Krueger, “Reflection from Layered Surfaces due to Subsurface Scattering,” *Computer Graphics Proceedings (Siggraph ’93)*, pp. 165-174, 1993.

12. Xiao D. He, Kenneth E. Torrance, Francois X. Sillion, and Donald P. Greenberg, “A comprehensive physical model for light reflection,” *Computer Graphics (SIGGRAPH ’91 Proceedings)*, vol. 25, no. 4, pp. 175-186, 1991.

13. J.T. Kajiya, “The Rendering Equation,” *Computer Graphics*, vol. 20, no. 4, pp. 143-150, August 1986.

14. W. Krueger, “Intensity fluctuations and natural texturing,” *Computer Graphics*, vol. 22, no. 4, August 1988.

15. B. B. Mandelbrot and J. W. Van Ness, “Fractional Brownian Motion, Fractional Noises and Applications,” *SIAM Review*, vol. 10, no. 4, pp. 422-437, October 1968.

16. J. Mathews and R. L. Walker, *Mathematical Methods of Physics,* W. A. Benjamin, 1970.

17. N.L. Max, “Horizon mapping: shadows for bump-mapped surfaces,” *The Visual Computer*, vol. 4, 1988.

18. L. Neumann and A. Neumann, “Radiosity and Hybrid Methods,” *ACM Transactions on Graphics (to appear)*, 1995.

19. M. Oren and S. K. Nayar, “Generalization of Lambert’s Reflectance Model,” *Computer Graphics Proceedings (Siggraph ’94)*, pp. 239-246, 1994.

20. P. Poulin and A. Fournier, “A Model for Anisotropic Reflection,” *Computer Graphics (SIGGRAPH ’90 Proceedings)*, vol. 24, no. 4, pp. 273-282, held in Dallas, Texas; 6-10 August 1990, July 1990.

21. L. Williams, “Pyramidal Parametrics,” *Computer Graphics*, vol. 17, no. 3, pp. 1-11, July 1983.