The Natural Look

Alain Fournier

Department of Computer Science
University of British Columbia
Vancouver, BC
V6T 1W5
Canada
fournier@cs.ubc.ca

Abstract

The modelling of natural phenomena is a challenging and endless task in computer graphics. As difficult as it is, even with a good shape model our job is still not done. What is now known as "rendering" remains a challenge for natural objects and natural scenes.

All the local illumination models currently in use are insufficient to render convincingly the reflective properties of surfaces such as cloth, wood, grass fields, water, etc.. Even more difficult is to accomplish this at all scales, and blend smoothly from uniform surfaces to textures to surface displacements to geometric models.

The correct handling of global illumination is also essential for a "natural" look. There has been much activity recently to develop accurate and efficient algorithms to solve that problem. As much progress as has been made, we still have some way to go to be able to incorporate arbitrary local reflection models and complex participating media into them.

We will discuss these various challenges, illustrate the shortcomings of the current solutions as far as the rendering of natural scenes is concerned, and present some of the specific solutions we are currently working on. These solutions include the filtering of bump maps, yielding improved hierarchical local illumination models and in particular interesting reflection models for cloth, and *FIAT*, a global illumination model which can use arbitrary local illumination models.

CR Categories: I.3.3 [**Computer Graphics**]: Picture/Image Generation I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism

General Terms: algorithms, models.

Additional Keywords and Phrases: local illumination, bump map filtering, global illumination.

1. Introduction

I do not think it necessary to try to convince anybody that the modelling of natural phenomena is both a necessary and a challenging task. Just a look outside of your window (if you’re inside and have a window not controlled by X) should in addition convince anybody of the distance still left to cover in computer graphics. So far most of the attention has been focussed on *shape modelling*, and it is indeed the first thing to get right. In a previous survey [1] I addressed specifically some of the issues and results in this area. The other two important areas are *animation* of natural phenomena and *modelling the interaction of light* (of course itself a "natural" phenomenon) and natural objects. In this paper I will address some problems related to the latter. It does not mean that the three aspects can be always neatly separated. In many cases realistic animation cannot be achieved unless the motion is built into the geometric model, and, as will be obvious from the following, there are complex interactions between shape models and illumination models.

This is not a survey, as it concentrates around two rather specific topics, but is (I think) indicative of the range of problems we face in this area. I will assume known the current state of the art in modelling natural phenomena. *Textures* and texture mapping techniques have been very useful to augment geometric modelling. Among recent relevant development are [2][3][4][5].

2. The Modelling Hierarchy

The appearance of most objects results from a complex interaction between their *shape*, as modelled by geometry, and the way they reflect light. This dichotomy is only for descriptive convenience, however, and there is no sharp boundary between the two. Even if we keep to a level of details well above the wavelength of light to avoid considering interference and diffraction, we can either observe the effect of individual surface details or merely the average of such, depending on the scale and viewing distance from the surface. Taking woven cloth as an example, we might have to consider a geometric model for each strand of material, each yarn, each thread, or the average of the whole weaving pattern, depending on viewing conditions. For each level different modelling and rendering techniques have been developed in computer graphics.

If we note D_g the level of the geometry at which objects are usually defined (i.e. polygons or parametric surfaces), then displacement and bump mappings [6][7] approximate what the geometric model D_{g+1} would define at a higher resolution or magnification. As the details become smaller, even surface mapping becomes highly subject to aliasing problems. At this geometric level, D_{g+2}, the details cannot be seen individually and appear only because of the way they modulate the light reflected off them. These details are then modelled by local reflection models[1].

Traditionally, reflection models have been divided into two components: *diffuse* and *specular* [8][9][10][11]. The diffuse component takes into account the light that inter-reflects onto elements of a same surface and is reemitted equally in all directions. To model the specular reflection, Torrance and Sparrow[8] assume that the surface is made of highly reflective microscopic facets distributed in v-grooves. If the facets are randomly distributed over the surface, shadowing and masking functions can be statistically estimated, and, for a given distribution of slopes of the facets, the light reflected in a particular direction can be approximated. This shows clearly that the local illumination model usually is based on some model for the geometry of the surface at a smaller scale than the visible details. In recent work 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) end added interreflection and self-shadowing/masking factors derived from statistical techniques (the surface height is assumed to follow a Gaussian distribution). While their model extends considerably the flexibility and accuracy of traditional local models, there are still many surfaces which do not meet these assumptions.

In [13] we used a "hidden" level of geometry made of cylinders to allow the computation of anisotropic reflection. Our anisotropic reflection model corresponds to inserting two new geometric levels between the mapping (displacement or bump) and the isotropic reflection model, now identified as D_{g+4}. The isotropic reflection model D_{g+4}, like the facets model of Torrance and Sparrow, characterizes the surface nature of each cylinder. A group of adjacent cylinders oriented in one direction defines the D_{g+3} level while a set of groups of adjacent cylinders represents the D_{g+2} level. Figure 1). illustrates the interaction between these various levels.

Figure 1.
Levels of geometric modelling

While all these techniques are effective within their intented scale, a major unsolved problem is

---

[1] It is important to note that *local* is a relative term. In practice local means at a scale of the order of the area which is to be represented by a single illumination computation.

to allow their simultaneous use, and in particular ensure smooth transition between models when the scale changes. For instance in our anisotropy model one cannot see the individual cylinders, no matter how close one gets to the surface. 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.

3. Local Illumination Models

To begin to solve that problem, it is important to understand what "average" means in this context. The smallest scale I will consider is the one at which I can define and compute an effective *bidirectional reflectance distribution function* (BDRF). Now of course this function can be itself the result of considering a surface with a structure at a scale below the one considered. The important point is that this structure will never be explicit or visible in the rendering. The BDRF is then probably itself an average. The trouble right away is that a complete BDRF is a function of 4 variables $\rho(\theta_i, \phi_i, \theta_r, \phi_r)$, 2 for the incident direction and two for the reflected direction. This can (and most often is) reduced to two variables for *isotropic* surfaces, but most natural surfaces are not isotropic (and even more important we are very sensitive to the case where they are not). To be able to develop models above this, we have to compute the light reflected by a given area of the surface given the local BDRF, the local geometric characteristics of the surface (tangent plane, curvatures), the self blocking/shadowing and the amount of interreflection. This has to be averaged over the area considered, preferably as a close form solution, but barring this in a form that can be quickly computed. In addition, since at this scale the structure can actually be seen under some viewing conditions, we should be able to vary smoothly from the values given by the original BDRF to the average given by the new model. The same requirements and exercise can be repeated at an higher level.

4. Bump Map Filtering

As an example of problems and (partial) solution, let's consider *bump map filtering*.

4.1. The problem

Texture filtering has been investigated often in computer graphics, and in addition to the surveys by Greene and Heckbert [14], Greene [15], and Heckbert [1617] techniques are described in [18^{19}20] and [21].

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 introduced by Blinn [6] and have been widely used since. Unfortunately the filtering techniques listed above 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. This is the case of colour, whether it is represented in some **RGB** space, spectral samples or linear combinations of basis functions. This can be the case of some other values such as the *solid horizon angle* used by F. Taillefer to modulate the ambient term. It is definitively not the case for normals, since they are used in non-linear formulae by all current illumination models.

4.2. A Solution

The solution proposed is to compute a new "model" for each level for which an average is needed. To be specific, assume a discrete bump map which is used in the usual manner in a rendering system. The goal is to produce a pyramid of increasing texture areas, until the whole texture is reduced to a single texel, exactly as in done in *MIP* maps [20]. Again simple averaging to produce the new levels won't work. If we look at the mapped normals as discrete samples of an underlying distribution of normals $\mathbf{D}(\mathbf{P})$ defined for each point P on the surface, we see that the problem is one of resampling that function at different resolutions. One could use a variety of techniques to accomplish this (spline interpolation, spherical harmonics, etc.), but I found best for pragmatic reasons to represent the distribution of normals as a weighted sum of *n Phong peaks*, that is of functions of the form $\mathbf{D}(\mathbf{P}) = \sum_{i=0}^{n-1} a_i \cos^{n_i}(\alpha_i)$ where α_i is the angle between the vector $\mathbf{OP}$ and the direction of peak i (θ_i, ϕ_i). In practice n is kept small (≤ 7). Each peak has therefore 4 variables to fit, and non-linear least squares are used to compute the fit. It is remarkable that most bump maps yield a fit to within a fraction a percent under these circumstances. This is done for each level of the pyramid (normally each half the linear size of the preceding), but each level is computed from the bottom, not from each other (of course this is to be done only one per texture as a pre-processing step). To use a given level of the pyramid, one extracts the variables of the peaks for the appropriate texel of the level, and compute the shade as a weighted sum of Phong shading. This of course integrates easily into any current shading software/hardware. In practice for smooth transitions one has to interpolate between texels and between levels, and this is done in manner similar to [20]. The important difference is that the contributing peaks are not averaged, but their amplitude a_i is multiplied by the suitable interpolation weights.

4.3. Results and Further Problems

This approach effectively solves the initial problem of bump map filtering. It has a not totally unexpected side benefit: at each level one gets local illumination models, and at the top level one gets a model valid when all of the original bump maps is comprised within the local "sample". This model can be seen as a new BDRF extracted from the bump map, but it has the advantage of being made of rather 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. One can at little additional cost get anisotropy, and some of the complex reflection behaviour that makes velvet different from silk, and both different from taffetas. In collaboration with F. Taillefer I am currently working on extracting the necessary bump maps from geometric models deduced from the individual weaving patterns of various materials.

It is not all good news, however. None of the secondary reflection effects (self-blocking/shadowing, interreflection) are in the model[2]. Considered from the point of view of bump map filtering this is acceptable, since the bump map does not take this into account anyway, but seen as a method to generate a new illumination model this could be a serious drawback. Partial remedies are possible. A good example is the technique already mentioned developed by F. Taillefer to modulate the ambient component. More difficult is the case of self-shadowing, and unfortunately it is essential to solve this, since many types of surfaces such as fur and hair cannot be properly modelled without it. Of course a solution for repeated bump maps is to compute such effects directly from the original geometric model, if available. In the case of woven material this means for given illumination and viewing conditions one can compute on the geometric tile the reflection, shadowing and interreflection at the cost of sampling the tile at a fine enough resolution (cost which is acceptable only for relatively small repeated patterns). This approach, intermediary between local and global illumination could be called *context sensitive local illumination*.

5. Global Illumination

5.1. Introduction

It has always been known that behind the *ambient* term, which is set to a constant by most illumination models, lurks a problem of devastating complexity. If we want to achieve a convincing level of realism, however, it is indispensable to get at least a reasonable approximation of the amount and distribution of light falling on any object from every direction. This is so because our judgement about the nature, the quality and the orientation of the surfaces we see, and about the characteristics of the atmosphere around us, in short our whole sense of space, critically depends on it.

The reasons for the difficulties are fairly evident. A local illumination model defines the intensity distribution of the outgoing light given the intensity distribution of the incoming light, and does so with only that local information. On the other hand a global illumination model has to define the intensity distribution of the incoming light, which is in turn a function of the intensity distribution of the outgoing light.

5.2. Current Techniques

The first illumination models used light falling directly on the objects from the light (often ignoring shadows) to compute the light reflected in the direction of the viewer Since it is obvious that in most environments a significant amount of light actually comes indirectly from other surfaces, a constant term, the *ambient intensity* is usually added.

Shown first in its full glory by Whitted [22], ray-tracing traces paths from the eye into the environment through refracted and reflected rays. Ray-tracing models very well pure reflection and refraction, and since in many circumstances this is what we notice most in scenes, it is a successful

---

[2] Note that even the average colour of the surface is affected by self-shadowing/blocking.

technique. Another reason for its wide appeal is that it is relatively simple to implement. Ray-tracing, however, has two serious drawbacks. First it is a *ray-driven* process, and as such it inherently point-samples. Various modifications have been proposed to alleviate this problem, some of them very effective, such as cone-tracing [23], distributed ray-tracing [24], stochastic sampling [25] [26], stratified sampling [27] and pencil-tracing [28]. Second, ray-tracing traces from the eye, and as such cannot determine illumination coming indirectly from the light sources after reflection/refraction off other surfaces (or in general media). That means still being limited to a constant ambient light term. Various fixes have been suggested [29], most involving the elusive "ray-tracing from the light", none too successful.

Radiosity is a physical quantity expressing the *power* (quantity of energy per unit time per area sent by a surface into the environment. The radiosity of a surface is a function of the radiosities of other surfaces in the environment, and therefore to obtain it one has to solve a system of n linear equations, n being the number of surface elements in the scene. From the introduction of an algorithm based on radiosity by Goral, Torrance and Greenberg [30] [31], to its first practical application through the use of *hemi-cubes* [32], to the introduction of volume elements to include participating media [33], to an adaptive implementation that proves it can be a realistic choice for a rendering method [34], radiosity-based methods have produced remarkable pictures, and have shown that the global illumination problem for diffuse surfaces is essentially solved.

The obvious limitation of radiosity is that it is a model of diffuse reflections. Several attempts have been made to introduce specular reflection within a radiosity approach, notably by Immel, Cohen and Greenberg [35], but this proves to be extremely costly, because of the necessity to use bidirectional reflectance which forces a jump in the size of the solution matrix from $O(n^2)$ to $O(n^4)$, where n is the number of elements in the scene. A very interesting and (after the fact) simple observation made by Neumann and Neumann [36] is that the same system of equations obtained for diffuse surfaces is obtained for a more general class of *separable* illumination models, that is where the function of the light direction is separable from the function of the eye direction. They thus obtain a limited range of specular models "for free" within the radiosity solution.

Essentially, radiosity methods are *patch driven* (actually there can be surface patches or volume elements), and anything that forces an increase in the number n of surface elements will dramatically increase the storage requirements and the running time (roughly as $\Theta(n^2)$).

The fact that ray-tracing and radiosity model different aspects of the global illumination problem has lead several researchers to attempt combining the two techniques. Wallace, Cohen and Greenberg [37] and [38] [39] have met with some success in that direction, but the gain is limited by the fact that they consider only a limited number of diffuse/specular surface interactions. The most recent result along this line [40] is a two-pass solution which handles quite arbitrary BDRF through the clever use of spherical harmonic decomposition.

In a quite different vein, Buckalew and Fussell [41] have discretized light beam directions to replace the hemicube by discrete patch to patch links. Other approaches to improve on the hemicube solutions are found in [42] and [43]. They are relevant to this work because they address discretization problems similar to the ones found in our approach.

5.3. Analysis

An obvious conclusion from this brief survey is that each class of methods has good and bad qualities. Most of the drawbacks of ray-tracing stem from its *ray-driven* character, that is it works only for perfect mirrors and refractors, and from the fact it traces from the eye. Most of the drawbacks of radiosity come from the fact that it is *patch-driven*, and introduces the light only after the form factors have been computed, and from the fact it really works only for separable sources and reflectors. They are both global illumination solutions which critically depend 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.

The 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 [3]. One

---

[3] We hasten to point out that it will not be "ray-

way to achieve this goal is to use wave propagation, and it has been suggested before [44] [45] but we think that it is excessive and impractical at this time. 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 we are advocating, therefore, is a *light-driven, volume-oriented* approach.

6. FIAT

We suggest here a new paradigm. Its main characteristics are: it is a light-driven approach, it balances the light energy budget of adaptively determined cells, and it uses the result as "incremental local light sources" in a conventional rendering.

6.1. 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. This means that we assume that all the operators $\mathbf{O}(I)$ on light intensity (or any quantities proportional to the light power) are linear:

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

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.

6.2. Power Balance

Given a volume **V**, how do we know whether its *internal global illumination* has correctly been determined? We can by considering its *power balance*. If we call P_I the power we have so far determined to be going into **V** from outside, P_O the power we have so far determined to be going from **V** to the outside, P_A the power we have so far determined to be absorbed inside **V** (of course energy is conserved and P_A is transformed into non-visible forms of radiation, or non-radiative forms of energy) and P_E the power we have so far determined to be emitted within **V**, then the quantity:

$$\Delta P = P_E + P_I - P_A - P_O$$

is a measure of our ignorance about the power budget of this volume (see Figure 2).

Figure 2.
Power Budget for volume **V**

A positive ΔP means that some power is not correctly re-emitted into the environment. A negative ΔP means that some power is lost by **V** into the environment and not properly accounted for. Only when ΔP is zero can we claim to have balanced power budget of **V**. It is crucial to note that once we have done this, because of our linearity assumption we do not have to remember any of the power components. After balancing **V**, we have to consider it again only if there is a modification to one of the power values, and then we only need to use its increment.

In the initial state no power is exchanged between volumes, and the only unbalanced volumes are the ones containing a light source ($P_E = \Delta P > 0$).

---

tracing from the light", since we won't use "rays", and we won't follow just one reflection/refraction.

After balancing any of these volumes, new volumes will become unbalanced because their P_I will become positive ($P_I = \Delta P > 0$). If when dealing with a volume we are careful not to "create" energy, that is we have a local illumination and transmission model with some physical credibility, then a "treated" volume will be balanced, and remain so until new light is transmitted into it from an adjacent volume. This amount of light cannot be more than its neighbour received (in power), and in fact will be less after any reflection/refraction, so we are guaranteed to eventually reach a state where all volumes are balanced. In particular we will not become trapped in local minima, and general global optimization algorithms are not necessary.

An issue of concern is the existence of cycles. They are very possible, as for instance when a light source is between two parallel mirrors, each one of these three objects being in distinct volumes. We in practice *preempt* cycles by limiting the amount of power that can be reflected to something below 100%, so that the amount of power transmitted in a cycle will be damped rapidly.

6.3. Basic Algorithm

The straightforward implementation of this paradigm can then be described by the following pseudo-code:

```
begin Main
   Scene_Pointer Scene
   Volume_Pointer Volume
   List of Volume_Pointer V_List
   Get_Scene_Description(Scene)
   Choose_Initial_Volume(Scene, Volume)
   Add_To_V_List(Volume, V_List)
   Balance_Everything(V_List)
   Display(Scene)
   Enjoy(Scene)
end Main

begin Balance_Everything(V_List)
   Volume_Pointer Volume, Sub_Volume
   List of Volume_Pointer Sub_V_List
   while  (V_List not Empty)
      Chose_Next_Unbalanced(V_List, Volume)
      if (Compute_DELTA_P(Volume)
        < P_TOLERANCE) then
            Remove_From_List(V_List, Volume)
      elsif Can_Deal_With_Volume(Volume)
         then
         Balance_Volume(Volume)
         Update_Neighbours(Volume, V_List)
         /* Update_Neighbours can cause some
          of the neighbours to be put on V_List */
         Update_Local_Illumination(Volume)
         Remove_From_List(Volume, V_List)
      else
         Split_Into_Sub_V(Volume, Sub_V_List)
         for each Sub_Volume in Sub_V_List
            Add_To_V_List(Sub_Volume)
         end for
      end if
   end while
end Balance_Everything
```

6.4. Results So Far

We have an implementation based on adaptive decomposition of the space in an octree of cells, light propagation within cells modelled by sampled beams with discretized directions, and a light update buffer for progressive rendering. See [46] for some details on the implementation on a parallel architecture.

The advantages of the approach are the greater range of effects, over both radiosity-based techniques and ray-tracing, that can be included (indirect illumination, specular reflection, contributing media), the adaptive and progressive nature of the process (in particular unlit areas are not considered) and the flexibility achieved (parameters can be tuned to control the time spent on the global component). The disadvantages are the relatively large memory requirements, the slow running (real-time is only a distant hope) and the problems with discretization (especially for the directions). The good points of the bad points is that the overall complexity is low with respect to the size of the scene. It will take a large scene, however, to begin to reap the benefits of this slow growth. We have had limited experience with the integration of participating media to the computations. The main drawbacks is that they greatly increase the number of active elements within each cell, and therefore the storage and computational needs.

7. Conclusion

After reiterating the complexity of simulating the interaction of light and matter, I isolated two specific class of problems, and presented beginnings of solutions in each of them. Techniques for the filtering of bump maps lead us to a simple improved model of local illumination, and we explored a new approach to solve a very general case of global illumination.

In addition to the shortcomings and remaining problems noted in the above, there are still many large questions to be addressed. 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 [47] but are not well suited to animation. My initial interest in the problem of filtering surfaces arose in the context of stochastic modelling [48] 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. Here again statistical methods help, but one should be able to create and maintain differences between areas with similar statistics.

I have not discussed the issue of light representation, that is spectral representation, and physically correct light transport, both to help account for diffraction, interference, dispersion, polarization, etc. Some work is under way [12][49][50][51][52] in this direction.

To conclude with a plea: never propose techniques or algorithms to simulate natural phenomena without pictures of the "real thing" for comparisons. This is still very rarely done (the last reference is a notable exception), but as the jazz pianist and composer Les McCann wrote: "Try to make it real, compared to what?".

Acknowledgements

I gratefully acknowledge the support of NSERC through operating grants and an equipment grant which considerably facilitated the research described here. The support of the Province of Ontario through an Information Technology Research Centre Grant during my tenure at the University of Toronto is also gratefully acknowledged. The support of the University of British Columbia in establishing a computer graphics laboratory in our department is greatly appreciated, as is the support of IBM Canada, which established GraFiC, an invaluable facility. FIAT was developed in collaboration with Eugene Fiume, Marc Ouellette and Chuan Chee. Frederic Taillefer and Guang Huo collaborated on the development of models for woven materials. Pierre Poulin and John Buchanan often helped with suggestions, criticisms and advice (not always taken).

References

1. A. Fournier, ‘‘The Modelling of Natural Phenomena,’’ *Proceedings of Graphics Interface ’89*, pp. 191-202, held in London, Ontario; 19-23 June 1989, 1989.
2. K. Perlin and E. Hoffert, ‘‘Hypertexture,’’ *Computer Graphics*, vol. 21, July 1989.
3. J. P. Lewis, ‘‘Algorithms for Solid Noise Synthesis,’’ *Computer Graphics*, vol. 21, July 1989.
4. Greg Turk, ‘‘Generating textures on arbitrary surfaces using reaction-diffusion,’’ *Computer Graphics (SIGGRAPH ’91 Proceedings)*, vol. 25, no. 4, pp. 289-298, 1991.
5. Andrew Witkin and Michael Kass, ‘‘Reaction-diffusion textures,’’ *Computer Graphics (SIGGRAPH ’91 Proceedings)*, vol. 25, no. 4, pp. 299-308, 1991.
6. J.F. Blinn, ‘‘Simulation of Wrinkled Surfaces,’’ *Computer Graphics*, vol. 12, no. 3, pp. 286-292, August 1978.
7. N.L. Max, ‘‘Horizon mapping: shadows for bump-mapped surfaces,’’ *The Visual Computer*, vol. 4, 1988.
8. K.E. Torrance and E.M. Sparrow, ‘‘Polarization, Directional Distribution, and Off-Specular Peak Phenomena in Light Reflected from Roughened Surfaces,’’ *J.Opt.Soc.Am.*, vol. 56, no. 7, 1966.
9. J.F. Blinn, ‘‘Models of Light Reflection For Computer Synthesized Pictures,’’ *Computer Graphics*, vol. 11, no. 2, pp. 192-198, July 1977.
10. R.L. Cook and K.L. Torrance, ‘‘A Reflectance Model for Computer Graphics,’’ *ACM Trans. on Graphics*, vol. 1, no. 1, pp. 7-24, January 1982.
11. P. Beckmann and A. Spizzichino, *The scattering of electromagnetic waves from rough surfaces*, Artech House Inc, 1987.
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.
14. N. Greene and P. S. Heckbert, ‘‘Creating Raster Omnimax Images from Multiple

Perspective Views Using the Elliptical Weighted Average Filter,'' *IEEE Computer Graphics and Applications*, vol. 6, no. 6, pp. 21-27, June 1986.

15. N. Greene, ``Environment Mapping and Other Applications of World Projections,'' *IEEE Computer Graphics and Applications*, vol. 6, no. 11, pp. 21-29, November 1986.
16. P.S. Heckbert, ``Filtering by Repeated Integration,'' *Computer Graphics*, vol. 20, no. 4, pp. 315-321, August 1986.
17. P.S. Heckbert, ``Survey of Texture Mapping,'' *IEEE Computer Graphics and Applications*, vol. 6, no. 11, pp. 56-67, November 1986.
18. W. Dungan, A. Stenger, and G. Sutty, ``Texture Tiles Consideration for Raster Graphics,'' *Computer Graphics*, vol. 12, no. 3, pp. 130-134, 1978.
19. F.C. Crow, ``Summed-Area Tables for Texture Mapping,'' *Computer Graphics*, vol. 18, no. 3, pp. 207-212, July 1984.
20. L. Williams, ``Pyramidal Parametrics,'' *Computer Graphics*, vol. 17, no. 3, pp. 1-11, July 1983.
21. A. Fournier and E.L. Fiume, ``Constant-Time Filtering with Space-Variant Kernels,'' *Computer Graphics*, vol. 22, no. 4, pp. 229-238, August 1988.
22. T. Whitted, ``An Improved Illumination Model for Shaded Display,'' *Comm. of the ACM*, vol. 23, no. 6, pp. 343-349, June 1980.
23.
24. R.L. Cook, T. Porter, and L. Carpenter, ``Distributed Ray Tracing,'' *Computer Graphics*, vol. 18, no. 3, pp. 137-145, July 1984.
25. M.A.Z. Dippe and E.H. Wold, ``Antialiasing through Stochastic Sampling,'' *Computer Graphics*, vol. 19, no. 3, pp. 69-78, July 1985.
26. R.L. Cook, ``Stochastic Sampling in Computer Graphics,'' *ACM Trans. on Graphics*, vol. 5, no. 1, pp. 51-72, January 1986.
27. M.E. Lee, R.A. Redner, and S.P. Uselton, ``Statistically Optimized Sampling for Distributed Ray Tracing,'' *Computer Graphics*, vol. 19, no. 3, pp. 61-67, July 1985.
28. M. Shinya, T. Takahashi, and S. Naito, ``Principles and Applications of Pencil Tracing,'' *Computer Graphics*, vol. 21, no. 4, pp. 45-54, July 1987.
29. G.J. Ward, F.M. Rubinstein, and R.D. Clear, ``A Ray Tracing Solution for Diffuse Interreflection,'' *Computer Graphics*, vol. 22, no. 4, pp. 85-92, August 1988.
30. C.M. Goral, K.E. Torrance, D.P. Greenberg, and B. Battaile, ``Modeling the Interaction of Light Between Diffuse Surfaces,'' *Computer Graphics*, vol. 18, no. 3, pp. 213-222, July 1984.
31. T. Nishita and E. Nakamae, ``Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection,'' *Computer Graphics*, vol. 19, no. 3, pp. 23-30, July 1985.
32. M.F. Cohen and D.P. Greenberg, ``The Hemi-Cube: A Radiosity For Complex Environments,'' *Computer Graphics*, vol. 19, no. 3, pp. 31-40, July 1985.
33.
34. M.F. Cohen, S.E. Chen, J.R. Wallace, and D.P. Greenberg, ``A Progressive Refinement Approach to Fast Radiosity Image Generation,'' *Computer Graphics (SIGGRAPH '88 Proceedings)*, vol. 22, no. 4, pp. 75-84, held in Atlanta, Georgia, August 1988.
35. D.S. Immel, M.F. Cohen, and D.P. Greenberg, ``A Radiosity Method for Non-Diffuse Environments,'' *Computer Graphics*, vol. 20, no. 4, pp. 133-142, August 1986.
36. L. Neumann and A. Neumann, ``Photosimulation: Interreflection with Arbitrary Reflectance Models and Illumination,'' *Computer Graphics Forum*, vol. 8, pp. 21-34, 1989.
37. J.R. Wallace, M.F. Cohen, and D.P. Greenberg, ``A Two-Pass solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods,'' *Computer Graphics*, vol. 21, no. 4, pp. 311-320, July 1987.
38. O. Devillers, C. Puech, and F. Sillion, ``CIL: Un Modele d'Illumination Integrant les Reflexions Diffuse et Speculaires,'' Technical Report LIENS-87-12, Laboratoire d'Informatique de l'Ecole Normale Superieure, Paris, October 1987.

39. F. Sillion and C. Puech, ‘‘A General Two-Pass Method Integrating Specular and Diffuse Reflection,’’ *Computer Graphics*, vol. 23, no. 3, pp. 335-344, July 1989.

40. Francois X. Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg, ‘‘A global illumination solution for general reflectance distributions,’’ *Computer Graphics (SIGGRAPH ’91 Proceedings)*, vol. 25, no. 4, pp. 187-196, 1991.

41. C. Buckalew and D. Fussell, ‘‘Illumination Networks: Fast Realistic Rendering with General Reflectance Functions,’’ *Computer Graphics*, vol. 23, no. 3, pp. 89-98, July 1989.

42. D. R. Baum, H. E. Rushmeier, and J. M. Winget, ‘‘Improving Radiosity Solutions Through the Use of Analytically Determined Form-Factors,’’ *Computer Graphics*, vol. 23, no. 3, pp. 325-334, July 1989.

43. J. R. Wallace, K. A. Elmquist, and E. A. Haines, ‘‘A Ray Tracing Algorithm for Progressive Radiosity,’’ *Computer Graphics*, vol. 23, no. 3, pp. 315-324, July 1989.

44. H.P. Moravec, ‘‘3D Graphics and the Wave Theory,’’ *Computer Graphics*, vol. 15, no. 3, pp. 289-296, August 1981.

45. E. Bahar and S. Chakrabariti, ‘‘Full-Wave Theory Applied to Computer-Aided Graphics for 3D Objects,’’ *IEEE Computer Graphics and Applications*, vol. 7, no. 7, pp. 46-60, July 1987.

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

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

48. 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.

49. Maria Raso and A. Fournier, ‘‘A Piecewise Polynomial Approach to Shading Using Spectral Distributions,’’ *Proceedings of Graphics Interface ’91*, pp. 40-46, 1991.

50. Gary Meyer and Richard Hale, ‘‘A Spectral Database for Realistic Image Synthesis,’’ *Proceedings of Graphics Interface ’91*, pp. 47-52, held in Calgary, Alberta; 3-7 June 1991.

51. Judith A. Grataganis and John Tartar, *Wave-based spectrum rendering*, Department of Computer Science, University of Alberta, 1990.

52. Atsushi Takagi, Hitoshi Takaoka, Tetsuya Oshima, and Yoshinori Ogata, ‘‘Accurate Rendering Technique Based on Colorimetric Conception,’’ in *Computer Graphics (SIGGRAPH ’90 Proceedings)*, ed. Forest Baskett, vol. 24, pp. 263-272, held in Dallas, Texas; 6-10 August 1990, 1990.