

Normal Distribution Functions and Multiple Surfaces

Alain Fournier

Department of Computer Science
University of British Columbia
Vancouver, BC
V6T 1Z2
Canada
fournier@cs.ubc.ca

Abstract

A *normal distribution function* for a surface is a function defined for each point of the surface and giving the density of normals as a function of direction. We examine the role such functions can play in local illumination models, in particular how they can be used as a compact representation of *bidirectional reflectance distribution functions*.

Such functions can be computed from the geometry of the surface at some intermediate level between the micro and the macroscopic level (the *mesoscopic* level) (in particular from bump-maps), computed from the overall BRDF and some *basis* BRDF, or custom-designed for special effects.

We discuss the pluses and minuses of such an approach, the connection with bump-map filtering and geometric modelling, and show examples of the use of the concept with *multiple surfaces*.

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, normal distribution function, mesoscopic level, multiples surfaces.

1. Introduction

To achieve any degree of realism in computer generated images, shape modelling is only the beginning of the struggle. Another fundamental step is *modelling the interaction with light*. Of course these steps cannot always be neatly separated. In most cases there are complex interactions between shape models and illumination models. In fact the appearance of most objects results from such interactions between their *shape*, as modelled by geometry, and the way they reflect light. This dichotomy is only for descriptive convenience. This paper addresses more particularly the modelling of local interaction between material surfaces and light. It is easy to overlook the importance of these effects for our interpretation of the world, the subtlety of our visual system and the distance between our current illumination models and the behaviour of real surfaces. As an example, consider how easily we recognize material such as velvet, silk or denim, by their colour and reflectivity structures, even at large distances where the details are not distinguishable (see Figure 1).

Figure 1.
Types of material (real; shown in presentation)

1.1. Scales

The type of interactions between light and matter is largely dependent on the scale at which we consider the material. 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 most levels of details modelling and rendering techniques have been developed in computer graphics. The best defined and studied are the the *geometric model* (such as polygons, parametric surfaces), which can be called the *macroscopic level*, and the *local reflection models*¹, which can be called the *microscopic level*. The level intermediary between these two, which we will dub the *mesoscopic level* (from the Greek *mesos*, middle), is less explored and the level of most concern in this paper. It includes such things as *displacement maps* and *bump maps* [Blin78, Max88].

Traditionally, reflection models have been divided into two components: *diffuse* and *specular* [Torr66, Blin77, Cook82, Beck87]. 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 [Torr66] 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 is a clear example where the local illumination model is based on some model for the geometry of the surface at a smaller scale than the visible details. One of course can go further. In recent work to improve the generality of illumination models, He *et al* [He91] 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). 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 [Poul90a] 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. The isotropic reflection model, like the facets model of Torrance and Sparrow, characterizes the surface nature of each cylinder. Set of groups of adjacent cylinders can be used to represent yet another intermediary level.

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

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.

above. As it will turn 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.

1.2. Local Illumination Models

To begin to solve that problem, it is important to understand what "average" means in this context. The smallest scale we will consider at the microscopic level is the one at which we can define and compute an effective *bidirectional reflectance distribution function* (BRDF). 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 BRDF is then probably itself an average. The trouble right away is that a complete BRDF 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 direction². 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 at the mesoscopic level we have to compute the light reflected by a given area of the surface given the local BRDF, the local geometric characteristics of the surface (tangent plane, curvatures), the masking/shadowing and the amount of interreflection. In other words we need information from both the microscopic and the macroscopic level. 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 BRDF to the average given by the new model.

2. Normal Distribution Function

A *normal distribution function* (NDF) for a surface is a function $\mathbf{n}(\theta, \phi)$ defined for each point on the surface, and giving for each direction (θ, ϕ) at this point the density of normal vectors pointing in that direction. The direction is defined in the surface local frame of reference (the *Frenet frame*). In the following we will assume that the function has been normalized so that

$$\int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} \mathbf{n}(\theta, \phi) \sin\theta \, d\theta \, d\phi = 1 \quad \text{Eq.(1)}$$

Most of the functions considered have a value of 0 at directions below the tangent plane.

This function can be used in formulas where the "normal" normal appears. To take for instance a generic function of two directions, where the directions are in the surface's Frenet frame, one uses:

$$R_{\mathbf{n}}(\theta_R, \phi_R, \theta_I, \phi_I) = \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} \mathbf{n}(\theta, \phi) R(T(\theta, \phi, \theta_R, \phi_R), T(\theta, \phi, \theta_I, \phi_I)) \sin\theta \, d\theta \, d\phi \quad \text{Eq.(2)}$$

where $T(\theta, \phi, \theta_A, \phi_A)$ is the direction (θ_A, ϕ_A) transformed to the frame of reference of the direction (θ, ϕ) . This is of course a form of the convolution of the two functions. It is not the ordinary convolution, since the transformations are not affine, but might be when on the right domain. It is easy to see that when the NDF is a delta function, for instance³ $\mathbf{n}(\theta, \phi) = \frac{\delta(\theta, \phi)}{\sin\theta}$ from the $\delta()$ sifting property $R()$ regains its original definition, and the NDF corresponds to the usual concept of a surface normal. The same observation applies when $\mathbf{n}()$ is the sum of delta functions:

$$\mathbf{n}(\theta, \phi) = \sum_i \frac{\delta(\theta - \theta_i, \phi - \phi_i)}{\sin\theta_i} \quad \text{Eq.(3)}$$

In this case $R_{\mathbf{n}}()$ is the sum of instances of the function $R()$:

2. It is also a function of the wavelength. We will ignore this for the time being.
3. We assume everything is correct for $\theta = 0$.

$$R_{\mathbf{n}}(\theta_R, \phi_R, \theta_I, \phi_I) = \sum_i R(T(\theta_i, \phi_i, \theta_R, \phi_R), T(\theta_i, \phi_i, \theta_I, \phi_I)) \quad \text{Eq.(4)}$$

Consider a more general case when we have a function $R_{\mathbf{n}}()$ that is exactly or approximately equal to a linear combinations of some basis functions:

$$R_{\mathbf{n}}(\theta_R, \phi_R, \theta_I, \phi_I) = \sum_i a_i B_i(\theta_R, \phi_R, \theta_I, \phi_I) \quad \text{Eq.(5)}$$

We can equate the right hand side term by term to expressions of the kind:

$$B_i(\theta_R, \phi_R, \theta_I, \phi_I) = \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} \mathbf{n}_i(\theta, \phi) F_i(T(\theta, \phi, \theta_R, \phi_R), T(\theta, \phi, \theta_I, \phi_I)) \sin\theta \, d\theta \, d\phi \quad \text{Eq.(6)}$$

This can be accomplished in a trivial way by making the $\mathbf{n}_i()$ functions be $\delta()$ functions and the $F_i()$ be identical to the $B_i()$.

A quite different way is to have all the $F_i()$ functions be the same only with parameters depending on i and the $\mathbf{n}_i()$ functions be in effect selectors of the $F_i()$, and therefore the $B_i()$:

$$\mathbf{n}(\theta, \phi) = \sum_i b_i \mathbf{n}_i(\theta, \phi) \quad \text{Eq.(7)}$$

and:

$$R_{\mathbf{n}}(\theta_R, \phi_R, \theta_I, \phi_I) = \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} \sum_i b_i \mathbf{n}_i(\theta, \phi) F_i(T(\theta, \phi, \theta_R, \phi_R), T(\theta, \phi, \theta_I, \phi_I)) \sin\theta \, d\theta \, d\phi \quad \text{Eq.(8)}$$

We have traded a sum of four-variable functions for a sum of two-variable functions plus a family of four-variable "transfer" functions (the $F_i()$ functions). For arbitrary basis functions $B_i()$ it might not be always possible to find $F_i()$ and a family of $\mathbf{n}_i()$ to verify equation (8), but we will see that it is more useful to consider the inverse problem, and we will assume the latter to fit to the function $R_{\mathbf{n}}()$ according to equation (8):

2.1. Simple Normal Distribution Functions

We can see equation (8) as stating that a sum of simple NDFs (the $\mathbf{n}_i()$) and a transfer function approximates an arbitrary BRDF, or as stating that a sum of simple BRDFs (the $B_i()$) approximates an arbitrary BRDF. In fact there can be a shift in boundary between the simple NDFs and the simple BRDFs. To take a familiar example, assume that the simple NDFs are delta functions:

$$\mathbf{n}_i(\theta, \phi) = \delta(\theta - \theta_i, \phi - \phi_i) \quad \text{Eq.(9)}$$

and the transfer function is the Phong specular term:

$$F_i(T(\theta, \phi, \theta_R, \phi_R), T(\theta, \phi, \theta_I, \phi_I)) = (\mathbf{N}(\theta, \phi) \cdot \mathbf{H}(\theta_R, \phi_R, \theta_I, \phi_I))^{\eta_i} \quad \text{Eq.(10)}$$

$\mathbf{N}(\theta, \phi)$ is the unit vector in the (θ, ϕ) direction, and $\mathbf{H}(\theta_R, \phi_R, \theta_I, \phi_I)$ is the unit vector bisector of the directions (θ_R, ϕ_R) and (θ_I, ϕ_I) . In this case equation (8) becomes:

$$R_{\mathbf{n}}(\theta_R, \phi_R, \theta_I, \phi_I) = \sum_i b_i (\mathbf{N}(\theta_i, \phi_i) \cdot \mathbf{H}(\theta_R, \phi_R, \theta_I, \phi_I))^{\eta_i} \quad \text{Eq.(11)}$$

in other words a sum of "Phong peaks" functions.

We can also have $\mathbf{n}()$ itself be a sum of "Phong peaks":

$$\mathbf{n}(\theta, \phi) = \sum_i c_i (\mathbf{N}(\theta_i, \phi_i) \cdot \mathbf{H}(\theta_i, \phi_i, \theta, \phi))^{\eta_i} \quad \text{Eq.(12)}$$

In this case we want to find a family of transfer functions $F_i()$ such that we obtain the same $R_{\mathbf{n}}()$ as before. This is accomplished by setting $F_i()$ as:

$$F_i(T(\theta, \phi, \theta_R, \phi_R), T(\theta, \phi, \theta_I, \phi_I)) = \frac{\delta(\theta - \theta_i, \phi - \phi_i) (\mathbf{N}(\theta, \phi) \cdot \mathbf{H}(\theta_R, \phi_R, \theta_I, \phi_I))^{\eta_i}}{(\mathbf{N}(\theta_i, \phi_i) \cdot \mathbf{H}(\theta_i, \phi_i, \theta, \phi))^{\eta_i}} \quad \text{Eq.(13)}$$

Another reasonable family of simple NDFs is provided by spherical harmonics, since they can be used easily to approximate a given total NDF.

2.2. BRDF vs NDF

From the preceding equations and examples it is clear that there is a strong relationship between BRDFs and NDFs. It is also trivially true that through the use of the transfer function we can generate a NDF with an effect identical to an arbitrary BRDF. The more interesting issue is whether for all, or for an acceptably wide range, of experimental BRDFs it is possible to define NDFs which are sums of simple NDFs, and which along with a canonical family of transfer functions approximate the BRDF to an acceptable precision⁴. At this point we cannot offer any theoretical results, or even approximations based on non-trivial experimental BRDFs, but our strong feeling is that the answer is affirmative.

3. Where do NDFs come from

It is clear that the motivation for defining NDFs is that for many surfaces⁵ the light reflected by a surface element is the result of reflection from a distribution of surface normals. The next question is therefore how to obtain and store normal distribution functions.

3.1. Geometric Model of the Surface

If the analytical description of the surface over the area to be considered is available, one can consider extracting the NDF from it. For some surfaces it is quite easy to go from one to the other. For example for a sphere:

$$\mathbf{n}(\theta, \phi) = \frac{1}{4\pi} \quad \text{Eq.(14)}$$

or for a cylinder with axis in the $\phi = 0$ direction:

$$\mathbf{n}(\theta, \phi) = \frac{1}{2} \delta(\phi) \quad \text{Eq.(15)}$$

In general, however, it is not easy to go from the surface definition to the NDF. One of the most serious complications is the fact that self-blocking and self-shadowing should be taken into account for distributions on non-convex surfaces. For example consider positive and *negative* cylinders (to use the terminology in [Poul90a]). They have the same distribution of normals, but different BRDF due to different self-blocking characteristics.

3.2. Bump Map Filtering

One way to avoid some of the analytical difficulties, and more importantly to be able to design and manipulate normal distributions at will is to extract them from bump-maps. The bump-map can be in turn extracted from discrete sampling of a geometric model. The techniques to extract NDFs from bump maps are discussed in another paper [Four92] in the context of the filtering of bump maps. The advantages of this route is that self-blocking and self-shadowing information can also be obtained. The NDF so obtained is expressed as a sum of $\delta()$ functions, and the transfer function is a Phong distribution. The presentation of this paper will include pictures illustrating the NDF extracted and the resulting shading. The issue of self-shadowing and self-blocking, along with the related issue of changes of colour of the surface according to the lighting and viewing conditions, are discussed in more details in the previously quoted paper as well.

4. Actually the question is not so much about how well the BRDF is approximated, but how well the resulting distribution of reflected light is approximated.

5. In fact all surfaces when seen from a sufficient distance.

3.3. Measured BRDF

As shown above, one can directly fit a BRDF with a sum of simple NDFs. We have not done so in practice yet, but my guess is that non-linear least-square techniques (similar to what was used for the NDFs extracted from bump-maps) would be successful in that application as well. The process consists in taking the measured BRDF as discrete data points, and adjusting the positions, amplitudes and exponents of the Phong peaks describing the NDF for an acceptable fit between the measured and the computed reflectance values.

3.4. Making Them Up

The last way is the "traditional" computer graphics way, that is to twiddle with the NDFs until "it looks good". In simple cases it is quite easy to do, and it also can be done for exploring possibilities, but to be effective it requires an interactive setting, or at the very least a fast rendering of the resulting surface under revealing lighting and viewing conditions.

4. Multiple Surfaces, or How to Use NDFs

The general answer to the question of how to use NDFs is to compute the resulting BRDF and use it as part of the local illumination equations. In practice, however, one does not want to use to complex a BRDF, one would like to be able to use some or all of the hardware assist available on modern graphics workstations, and one would like to be able to use formulae and strategies painstakingly developed for extended light sources [Poul90b, Nish85]. This is why in particular the approach described in [Dret90] was chosen. By treating the resulting surface as a *multiple surface*, that is as one where each point has more than one normal, each adding a Phong reflectance function to the total, the shading operation is just a sum of traditional shading operations.

For example Figure 2

Figure 2.
Anisotropic surface and multiple surface with two, three and seven normals
(shown in presentation)

shows a sphere with an anisotropic surface and with a multiple surface with two, three and seven normals. The normal vectors are defined in the surface local frame of reference (this means that a consistent tangent has to be defined for each point on the surface). One can see than an anisotropic surface is obtained in a very straightforward manner. In the case of the seven normals, the values were obtained from the filtering of a bump-map extracted from a geometric model describing a cylinder. Figure 3 and Figure 4 show the same geometry

Figure 3.
Velvet (shown in presentation)

with two different multiple surfaces, one defined to simulate the reflective behaviour of velvet, the other a rough approximation of silk. The geometric models are identical (this is not of course the case for real materials, the geometry is also a strong cue about the nature of the material) for

Figure 4.
Silk (shown in presentation)

both examples.

5. Conclusions

We have introduced for local shading the concept of normal distribution functions, and shown how they fit at the mesoscopic level between the microscopic level of the BRDF and the macroscopic level of the geometric model. We have shown how they relate analytically to the BRDF and the shading computations, and how they can be a much more compact substitute for a completely arbitrary BRDF without losing (too much) on generality. We have discussed the various possible sources of NDFs, including their extraction from the filtering of bump maps and the fitting from experimental BRDFs. Finally we have given few examples of the implementation of simple NDFs in the form of multiple surfaces.

This is only the beginning of the exploration of this approach. The technique allows for the smooth blending from a BRDF to another and for many simulations of natural textures, such as sand, shark skin and leaves.

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 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. Many conversations with Pierre Poulin, John Buchanan and Peter Cahoon helped shape the ideas presented herein.

References

- Beck87.
P. Beckmann and A. Spizzichino, *The scattering of electromagnetic waves from rough surfaces*, Artech House Inc, 1987.
- Blin78.
J.F. Blinn, "Simulation of Wrinkled Surfaces," *Computer Graphics*, vol. 12, no. 3, pp. 286-292, August 1978.
- Blin77.
J.F. Blinn, "Models of Light Reflection For Computer Synthesized Pictures," *Computer Graphics*, vol. 11, no. 2, pp. 192-198, July 1977.
- Cook82.
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.
- Dret90.
G. Drettakis, E. Fiume, and A. Fournier, *Tightly-Coupled Multiprocessing for a Global Illumination Algorithm*, pp. 387-398, Eurographics '90, September 1990.
- Four92.
A. Fournier, "Filtering Normal Maps and Creating Multiple Surfaces," *Imager Tech. Report 92-1*, UBC, Vancouver, BC, January 1992.

- He91. 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.
- Max88.
N.L. Max, "Horizon mapping: shadows for bump-mapped surfaces," *The Visual Computer*, vol. 4, 1988.
- Nish85.
T. Nishita, I. Okamura, and E. Nakamae, "Shading Models for Point and Linear Sources," *ACM Trans. on Graphics*, vol. 4, no. 2, pp. 124-146, April 1985.
- Poul90b.
P. Poulin and J. Amanatides, "Shading and Shadowing with Linear Light Sources," *Eurographics '90*, pp. 377-386, September 1990.
- Poul90a.
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.
- Torr66.
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.