Smooth Transitions between Bump Rendering Algorithms

Barry G. Becker[1]
Nelson L. Max[2]
University of California, Davis
and
Lawrence Livermore National Laboratory

Abstract

A method is described for switching smoothly between rendering algorithms as required by the amount of visible surface detail. The result will be more realism with less computation for displaying objects whose surface detail can be described by one or more bump maps. The three rendering algorithms considered are a BRDF, bump mapping, and displacement mapping. The bump mapping has been modified to make it consistent with the other two. For a given viewpoint, one of these algorithms will show a better trade-off between quality, computation time, and aliasing than the other two. The decision as to which algorithm is appropriate is a function of distance, viewing angle, and the frequency of bumps in the bump map.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.5 [Computer Graphics]: Three-Dimensional Graphics and Realism.

Keywords: animation, BRDF, bump map, displacement map, rendering, surface detail, volume texture.

[1] (510) 422-3724 becker@mozart.llnl.gov
[2] (510) 422-4074 max2@llnl.gov

1 Introduction

Objects in animation are sometimes distant specks; at other times a tiny part of one will fill the whole screen. If these objects have rough surfaces, the same rendering algorithm should not be used in both cases. Almost all real materials have a hierarchy of surface detail. We assume that the macro-structure of all objects is described by parameterized patches or a polygonal mesh. The micro-structure is then described by one or more bump tables for each level of detail below the geometrical, each giving bump height as a function of the 2-D surface parameters. An alternative way to describe the surface detail is through the use of volume textures to specify bump height as a function of 3-D coordinates [10, 12].

The Bidirectional Reflection Distribution Function or BRDF [13, 14, 6] captures the surface properties which are too small to be visible. Most real surfaces are neither purely specular (mirror-like) nor purely diffuse, but rather somewhere in between. To represent this non-trivial distribution of light reflectance a BRDF is used. It can be represented by a table indexed by a lighting direction and a viewing direction, to give the reflectance as a function of these directions. The BRDF used for this research is constructed from distributions of normals recorded from various views of a single displaced surface patch.

Bump-mapping [2] is an inexpensive way to achieve a good approximation to macroscopic surface roughness. The parameterized surface is treated as smooth for the purpose of visible surface determination, while the surface normals are perturbed to a first order approximation of what the actual bump normals would be.

The third algorithm, displacement-mapping [4, 5], is used when any shortcut in computation will be noticeable to the eye. Displacement mapping is different in that the surface is actually offset by the appropriate bump height so that the full 3-D geometry can be rendered. For purposes of maintaining consistent shading, the same approximated normal is used to shade the displaced surface as was used in the bump map. However, now it is applied to the displaced surface rather than to the flat parametric one.

Bump-mapping is good for economically rendering bumps which can be described as a height field. Unfortunately it does not account for occlusion. It is necessary to modify flat bump-mapping so that it yields images statistically similar to images produced by the other two methods. This revised procedure will be termed 'redistribution bump-mapping' because it redistributes the normals in a way that is statistically similar to those seen on the displaced surface viewed from a specific direction.

The three methods are blended together so that the parts of the scene which are close to the viewer, or close to the extreme edge (silhouette), would be displacement-mapped, since this is where missing detail would be noticed most. Smooth silhouette edges are an artifact of bump mapping which is easy to detect. Parts farther away, or whose normals are parallel to the viewing direction, will be bump-mapped. When surfaces have microscopic material-specific qualities or are very far from the viewer, they are rendered using a BRDF. More specifically, for a given scene, those features with a spatial frequency higher than one half cycle per pixel (the Nyquist limit) are considered in the BRDF. At the other end of the spectrum, features that are large enough to cause noticeable occlusion need to be displacement-mapped. The parts in between are rendered with varying degrees of redistributed bump-mapping. Most importantly, there is a smooth transition among the three. The effect is that the whole scene looks as if it were displacement-mapped, when in fact much of it was rendered with cheaper algorithms. Extending this concept we can have high frequency rough surfaces on top of low frequency rough surfaces, each bumpy level of detail having three rendered representations.

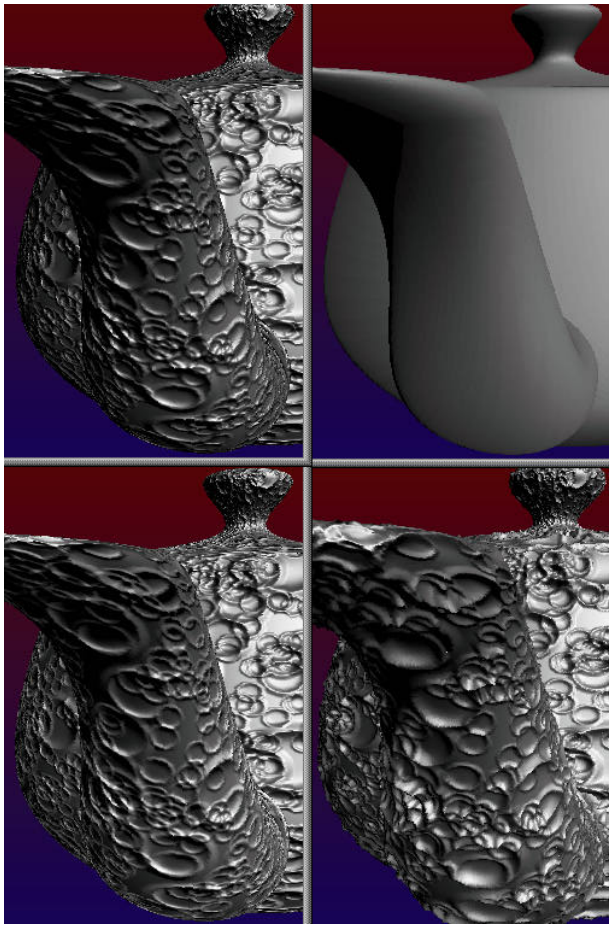

Figure 1: Counter-clockwise from upper left: bump-mapping, redistribution mapping, displacement mapping, BRDF.

In Figure 1 we see a teapot rendered in the four different ways. All renderings are based on the same height function. A major consideration for a smooth transitions among these is the consistency of the shading between methods. The amount of light emitted by a surface rendered with one method does not necessarily equal that amount emitted by the same surface rendered with another. Nor is the distribution of that light necessarily equivalent. A key aspect of this research is the determination of how the varying algorithms need to be modified in order to have their overall area-averaged light intensity contributions consistent.

There are five reasons why the average reflected intensity from a bump-mapped image is inconsistent with the reflected intensity from either the BRDF rendered image or the displacement-mapped image of the same object. Usually the BRDF is constructed under the assumption that the microfeatures of the surface are composed entirely of specular, mirrored facets. Bump- and displacement-mapping contain both specular and diffuse components. The easy solution to this inconsistency is to include a diffuse component for each microfacet when constructing the BRDF for the highest frequency bumps. Usually there is an inconsistency between bump- and displacement-mapping because actual surface displacement creates a geometrically computed facet normal for the shader while the perturbed normals for bump maps are only approximations. As previously mentioned this is overcome by using the approximated bump-mapped normals on the displaced surface. The approximated bump normals also vary more smoothly than the facet normals, especially with our quadratic interpolation, which is smoother than Blinn's approximation [2]. Note that if a procedural displacement function is employed, it is possible to compute the surface normal analytically. Since the BRDF is constructed from a displacement-mapped patch, the same inconsistency may arise for it. Again the solution is remedied by using the bump normal for tabulating the BRDF. The most difficult consistency problem is caused by occlusion. Occlusion, which is the hiding of some bumps by others, can change the distribution of visible surface normals. A solution is presented which redistributes bump normals so they match a distribution of normals similar to one derived from displacement-mapping. Lastly, there is the problem of consistency of shadowing. We have not yet found a general solution for shadowing, so we draw our images and compute our BRDF without it.

The concept of blending between methods is not new. The difficulty in overcoming the intensity distribution inconsistencies is perhaps the main reason why there are few coded examples. Kajiya [8] mentioned a hierarchy of scale which is appropriate for modelling the complexity of nature. He states that each level of detail contains the three subscales discussed above. Westin *et al.* [14] describes these levels as the geometrical, milliscale, and microscale. Perlin [11] proposed a method to shift between the BRDF and perturbed normals. Perlin's method does not include an explicit height table for determining the new normals, making displacement-mapping difficult. Fournier [7] has presented a promising approach for filtering normal maps by recording a discrete number of Phong peaks.

The software for each of the three algorithms described in this paper has been combined according to the previously discussed considerations. The result is an animation which explores a surface from changing distances and directions, showing that there are no significant side effects while transitioning between renderers. For more detail concerning the implementation refer to Becker [1].

2 Basic Algorithms

2.1 Bidirectional Reflection Distribution Functions

The BRDF is used to capture the microscopic reflectance properties of a surface. The BRDF itself can be a table of reflectivities or it can

be represented by a spherical harmonic series approximation [3, 14]. It is a function of either three or four variables representing the polar and azimuthal angles of the light rays. The polar angle is called θ and it measures the angle away from the normal. Its domain is $[0, \pi/2]$. The azimuthal angle is denoted by ϕ and has domain $[0, 2\pi)$, with 0 and 2π both in the direction of the viewer. An isotropic surface is one for which the emitted intensity does not vary as the surface is rotated radially about its surface normal. If only isotropic textures are used, then the arguments to the BRDF reduce to the two polar viewing directions and the difference in the azimuthal angle between the viewing and lighting directions. In the most general anisotropic case, the BRDF is a function of viewing direction and lighting directions, requiring all four angles.

There are several different ways to construct a BRDF. Cabral [3] constructed the BRDF directly from a bump map using horizon tables. Westin *et al*. [14] ray traced a generalized 3-D surface sample in order to calculate the intensities for their BRDF. Our method uses normal distributions. They are already required in order to create redistribution functions for the new bump-mapping method. The same normal distributions are used to create the BRDF. Fournier [7] has also discussed normal distributions.

A normal distribution is obtained by tabulating sampled normals from a projected displacement-mapped flat patch. The range of normals is a hemisphere. The hemisphere can be discretized into a finite number of (θ_N, ϕ_N) bins. When the displacement map is projected, each pixel of the projected image represents a sample normal, and the count for the bin containing that normal is incremented. If bump-mapping is used to draw the flat patch, then the approximated normal distribution is independent of θ. However, when looking from some direction with $\theta > 0$, self-occlusion may occur in the displacement-mapped image. This occlusion is accounted for by rendering the displacement-mapped geometry with a hardware z-buffer, coding the normal directions into the pixel colors. For grazing angles many potentially occluding patches may have to be rendered in order to get the occlusion correct on a single patch. The problem is solved by rendering a single patch using parallel projection, and then using a block read from the screen buffer to copy the patch to all the positions where it is needed, in a back to front ordering. In a postprocess the sample normals are scanned in and the distributions are created. These distributions will be used to find the redistribution functions and to make the BRDF. The normal distributions are stored in a 3-D table. The first index is the viewing polar angle θ_V. The second and third indices are the θ_N, ϕ_N angles specifying the normal direction. For simplicity a table access is described by $distr[\theta_V, N]$, where $N = (\theta_N, \phi_{N-V})$, and ϕ_{N-V} denotes $\phi_N - \phi_V$. The difference between viewing and lighting ϕ's is denoted by ϕ_{V-L}. To improve the statistics of the distribution, the patch is viewed in many ϕ_V directions for each θ_V. The result is normal distributions for each θ_V which account for proper occlusion. To use these distributions in constructing the BRDF, the algorithm in Figure 2 is used.

Note that there are two components to the BRDF, one for the diffuse information and one for the specular. This way the amount of diffusivity and specularity chosen can be used as a parameter later. The θ'_V and θ'_L represent the angles between the viewing or lighting direction and the bin normal N, rather than with the flat surface patch normal. The angle ϕ'_{V-L} is the difference between L and V when projected to the plane perpendicular to the bump normal. It is computed by

$$\begin{aligned}\phi'_L &= \arctan((L \cdot (N \times x)), (L \cdot (y \times N))) \\ \phi'_V &= \arctan((V \cdot (N \times x)), (V \cdot (y \times N))) \\ \phi'_{V-L} &= mod((\phi'_V - \phi'_L + \pi), 2\pi) - \pi \end{aligned} \quad (1)$$

where $x = (1, 0, 0)$ and $y = (0, 1, 0)$ are the axis directions of the bump table. This technique will give the same BRDF as if the combined displacement maps were used, as long as there is no correlation between the bumps at the different levels.

```
for each level n from highest to lowest frequency
 for each θ_V
  for each θ_L
   for each φ_{V-L}
    {H = (V + L)/|V + L|
     for each θ_N
      for each φ_{N-V}
       if highest frequency BRDF
          { increment BRDF^n_diff[θ_V, θ_L, φ_{V-L}] by
              (L · N) distr^n[θ_V, N]
            increment BRDF^n_spec[θ_V, θ_L, φ_{V-L}] by
              (H · N)^Phong distr^n[θ_V, N]
          }
       else
          { compute θ'_V, θ'_L and φ'_{V-L}
            increment BRDF^n_diff[θ_V, θ_L, φ_{V-L}] by
              BRDF^{n-1}_diff[θ'_V, θ'_L, φ'_{V-L}] distr^n[θ_V, N]
            increment BRDF^n_spec[θ_V, θ_L, φ_{V-L}] by
              BRDF^{n-1}_spec[θ'_V, θ'_L, φ'_{V-L}] distr^n[θ_V, N]
          }
    }
```

Figure 2: The algorithm to compute the BRDF using a table of normal distributions.

A smooth surface patch is rendered by interpolating the BRDF trilinearly in the angles θ_V, θ_L, and ϕ_{V-L}. The indices for the table are computed from a local coordinate on the patch surface. The smooth surface normal points in the direction of $\theta = 0$. The origin of the azimuthal angle is the projection of the viewing direction onto the surface.

For a given patch parameterization, $P(u, v)$, the partial derivatives, $P_u = \frac{\partial P}{\partial u}$ and $P_v = \frac{\partial P}{\partial v}$, are rarely the same length (causing stretching), and not always perpendicular (causing warping). For these reasons special care must be taken when indexing the BRDF to determine an intensity. The method for computing the difference in azimuthal angle is as follows:

$$\begin{aligned} V_n &= [V \cdot P_u, V \cdot P_v, 0] \\ L_n &= [L \cdot P_u, L \cdot P_v, 0] \\ \Phi_{V-L} &= \arccos(\frac{V_n}{|V_n|} \cdot \frac{L_n}{|L_n|}) \end{aligned} \quad (2)$$

The stretching will actually change the normal directions making the BRDF inaccurate. The BRDF would need to be recalculated to yield a theoretically correct result, but equation (2) does get the occlusion correct and gives nice anisotropic highlight effects in places where they would be expected.

2.2 Bump Mapping

In Blinn's bump-mapping [2], the surface is not actually altered from its smooth parametric form, but it is shaded as though it were.

Blinn used a bump height table B to calculate a linear approximation to the bump normal at a point P on an object surface. If $\vec{P}_u$ and $\vec{P}_v$ are the partial derivatives as above, the unnormalized surface normal is $\vec{N} = \vec{P}_u \times \vec{P}_v$. In the bump map B, the partial derivatives B_u and B_v at the interpolated point corresponding to P can also be computed using finite differences.

$$B_u = (B[u + \epsilon, v] - B[u - \epsilon, v])/(2 * \epsilon) \quad (3)$$

and B_v is similar. Each evaluation of B uses bilinear interpolation.

Truncating insignificant terms, Blinn [2] has showed that the new normalized normal is very close to

$$\vec{N}' = \frac{\vec{N} + B_u(\vec{N} \times \vec{P}_v) - B_v(\vec{N} \times \vec{P}_u)}{|\vec{N} + B_u(\vec{N} \times \vec{P}_v) - B_v(\vec{N} \times \vec{P}_u)|} \quad (4)$$

We have chosen to compute the bump map derivatives by a quadratic rather than linear scheme. Mach bands are eliminated by replacing Blinn's linear formula by a C^1 partial derivative formula, defined by taking the derivative of the C^2 cubic B spline curve approximation to the bump heights as a function of u or of v. Let $du = u - \lfloor u \rfloor$, then

$$B_u = (-du^2/2 + du - .5)B[\lfloor u \rfloor - 1, v] + (3du^2/2 - 2du)B[\lfloor u \rfloor, v]$$

$$+(-3du^2/2 + du + .5)B[\lfloor u \rfloor, v] + (du^2/2)B[\lfloor u \rfloor + 2, v]$$

and B_v is similar. Here each function evaluation requires only a linear interpolation in v. This method uses the same eight neighboring values in the height table as does (3), but with quadratic rather than linear weights.

The normals generated by this process do not lie in a distribution consistent with the other two algorithms. As previously discussed, $\vec{N}'$ must be further modified so that on average it will contribute to a normal distribution similar to displacement-map normals. This new algorithm, redistribution bump-mapping, is described in detail in Section 3.

It should also be noted that Perlin's volume textures [12], with the improvement by Max and Becker [10], can be substituted for bump maps when computing height values. The advantage of this is that there is no explicit parameterization to be concerned with, and thus no stretching to cause singularities or anisotropy. If a square patch has an isotropic texture mapped onto it, the texture becomes anisotropic as soon as the patch is stretched unevenly. Many parameterizations have singularities which lead to degenerate patches. If anisotropy is undesirable, then volume textures should be used. Perlin also used volume textures, and redistributed the normals to make them gaussian (personal communication) in his implementation of [11].

2.3 Displacement Mapping

Displacement-mapping is the direct approach to rendering surface detail. For parameterized surfaces, each patch in the object has a u and v parameterization. The u and v coordinates are used as indices to look up height values in the bump height table. The corresponding vertex is then displaced along its normal vector by that height [4]. The normal generated from the bump approximation is also used on the displaced vertices. There is little loss of accuracy in doing this, and continuity during the transition is assured. Occlusion, the main problem with bump-mapping, is accounted for automatically when the vertices are displaced.

Having multiple bump maps for many levels of detail means the displaced bumps will be rendered with the BRDF constructed from the next bump map of higher frequency. To keep combined displacements consistent with BRDFs representing several combined bump maps, surface perturbations for the i^{th} level must be perpendicular to the $(i-1)^{th}$ displaced surface. This means that for each vertex, P_u and P_v vectors must be computed for each level of detail which has been displaced. Since P_u and P_v are not necessarily perpendicular it is recommended that the following formula be used to compute them, given that the surface normal is N.

$$P_u[level + 1] = P_u[level] + B_u[level]N[level]$$

where $B_u[i], B_v[i]$ are the i^{th} bump map partial derivatives. The equation for P_v is similar.

3 Redistribution Bump Mapping

3.1 Normal Redistribution

The problem of eliminating inconsistencies between the different rendering models lies at the heart of making smooth transitions from one algorithm to another. Primarily we are concerned with keeping the integral of intensities equal over a small area on the surface while the rendering method changes.

Unfortunately, normals from bump-mapping do not yield a distribution similar to that of displacement-mapping or the BRDF. Since the polygon or patch itself is not displaced, it is possible to see normals which ought to be hidden by occluding bumps. In order to overcome this problem a redistribution function q is created. This is a function which accepts as input a normal generated by Blinn's [2] bump approximation, and outputs a normal which is statistically consistent with the distribution used to form the BRDF.

Since the distribution of normals on a displacement-mapped flat patch is different for each viewing angle, it is necessary to have redistribution functions for each one. When the viewing angle is vertical, the identity function is used. When the viewing angle is just above the horizon, the redistribution of bump normals is necessarily quite drastic. The effect is to pull forward normals that might be facing away, and push upward those that might be hidden. This new scheme for doing bump-mapping might appropriately be termed redistribution bump-mapping.

3.2 Redistribution Function Construction

Suppose a bumpy surface is viewed from a direction with polar angle θ_V. Let g denote the distribution of normals $distr(\theta, N)$ at this fixed θ_V, computed as above from the displacement map. Let f denote the distribution of normals in a (non-displaced) bump-mapped image. Note that f is the same as $distr(0, N)$. If q is the redistribution function described above, then the requirement that q take the distribution f to the distribution g is that for any region R in the hemisphere H of possible normals,

$$\int_{q(R)} f(\theta, \phi)d\omega = \int_R g(\theta, \phi)d\omega \quad (5)$$

It is easier to explain how to specify q in a 1-D case. So suppose $f(x)$ and $g(x)$ are two distributions on $[0, 1]$, such that

$$\int_0^1 f(x)dx = \int_0^1 g(x)dx = 1 \quad (6)$$

The problem is to find $q : [0, 1] \rightarrow [0, 1]$ such that

$$\int_{q(a)}^{q(b)} f(x)dx = \int_a^b g(x)dx \quad (7)$$

where a and $b \in [0, 1]$. It is enough to guarantee that

$$\int_0^{q(b)} f(x)dx = \int_0^b g(x)dx. \quad (8)$$

Let

$$G(b) = \int_0^b g(x)dx$$

and

$$F(b) = \int_0^b f(x)dx.$$

Then

$$G(b) = F(q(b))$$

hence

$$q(b) = F^{-1}(G(b)). \quad (9)$$

The redistribution function q maps a point b so that the area under the curve before b in g is equal to the area under the curve before the point $q(b)$ in f.

The problem in 2-D can be handled similarly. One method is to define 1-D redistribution functions separately for θ and ϕ. This gives adequate results for most bump maps, whose θ and ϕ distributions are fairly independent. This independence assumption is confirmed by the animation. For a more precise redistribution function, one can first redistribute ϕ, and then for each fixed ϕ, establish a separate redistribution function for θ. For details see Becker [1].

4 Transitions

4.1 Partial Bump Displacement

For control of appearance and for smooth transitions we want the ability to change the height of the bumps in the bump map. This will alter the normal distribution and occlusion information. By close consideration we can see that the change can be accounted for without having to recalculate the redistribution functions every time the bump heights are altered. If the heights are multiplied by a factor t, then the tangent of the angle between the bump normal and the smooth surface normal should also change by a factor t; i.e., $\tan(\theta_{N_t}) = t \cdot \tan(\theta_N)$. The normal, $N = (\theta_N, \phi_N)$, needs to be replaced by $N_t = (\arctan(t \cdot \tan(\theta_N)), \phi)$. In order to keep the visibility information the same, the viewing angle, θ_V, must be replaced with $\theta_W = \text{arccot}(\cot(\theta_V)/t)$. See discussion below concerning Figure 3.

The height of the bumps used to calculate the BRDF and redistribution functions must be the same as that of the bumps being rendered. This is because the BRDF is changed in a non-trivial way as the bump heights change. If we were only concerned with bump- and displacement-mapping, we could change the indexing on the redistribution functions to get the occlusion correct for changing bump heights. Unfortunately there is no easy way to re-index the BRDF to account for scale changes. Between the BRDF and redistribution bump-mapping, an intensity is computed for both methods. The resulting intensity is an interpolation of the two.

For the transition between bump- and displacement-mapping, intensity interpolation is not used, since it would cause the bump shading (particularly the highlights) to cross-dissolve rather than correctly adjust in position. As the bumps go from no displacement to full displacement the surface normals do not change, since they are always represented by Blinn's bump normal. The visible subset of bump normals does change, however, due to changing occlusion. Let $disp$ be the transition parameter which gives the fraction of the full bump height. With $disp = 0$ all normals are seen, even those on the back of bumps. With $disp = 1$, only the visible subset of these normals are seen. In Figure 3 the segments of the visible surface are shown in bold. The redistribution of normals takes normals from standard bump-mapping into this visible subset. For partially displaced bumps there is a different subset of visible normals, but there is a relationship between the bump height and this subset which can be exploited to give the necessary redistribution.

Different redistribution functions for varying heights are not stored, only different functions for different viewing θ's. Fortunately the two are equivalent. For the fractional bump height, $disp$, we can determine a new θ_W for which the same distribution of full height bump normals will be seen. Figure 3 shows that the distribution of normals for this partially displaced surface, viewed from θ_V, is identical to the distribution of visible normals for the fully displaced surface viewed from θ_W. The slope of the line V in Figure 3 is $disp$ times the slope of line W, so $\cot(\theta_V) = disp \cdot \cot(\theta_W)$

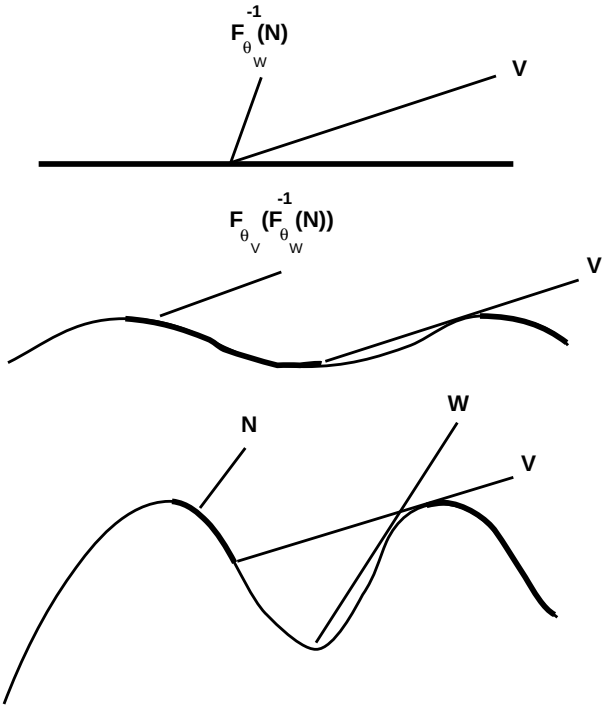

Figure 3: Top: the non-displaced surface. Middle: surfaced sisplaced by bump height fraction $disp$. Bottom: Fully displaced surface.

and the formula for finding θ_W is:

$$\theta_W = \text{arccot}(\cot(\theta_V)/disp).$$

The inverse redistribution function for θ_W is applied to take the visible bump normal from the partially displaced surface into a distribution similar to one from a flat bump-mapped surface. Next the redistribution function for θ_V is applied to that normal to take it all the way forward to match statistically a full displacement-mapped normal. Thus the change from bump-mapping to displacement-mapping is done through two table based function evaluations. Notice that as the bumps decrease in height, the new viewing θ_W approaches vertical. This means that the inverse function needs to alter the normals less in order to get them back to the bump-map distribution.

4.2 Algorithm Selection Criterion

Now that it is known how to modify the algorithms so that they will not deviate from a fundamental reflection model, it must be decided when to apply which algorithm. Clearly displacement-mapping should be applied when the view is close, and the BRDF when the view is far. The relationship is $1/d$, where d is distance, since that is how the projected size of an object relates to distance. Another variable to consider is viewing angle, θ_V. If f is the wavelength of a feature then $f \cos(\theta_V)/d$ is the wavelength of the projected feature (in the direction of maximum foreshortening), and should be no smaller than two pixels. When the object is close, we would like to see a rough silhouette; when it is far, aliasing becomes a problem on the edge so use of the BRDF is desirable. This implies that as the object moves away from the viewer, the transition from displaced bumps to BRDF will be far more rapid on the object silhouette than on that area where the patch normal points toward the viewer. The threshold at which the switch occurs is determined by a constant D.

Summarizing these properties, we define a transition parameter

$$T(d, \theta_V) = (1/d - D)/(\cos(\theta_V) + \epsilon). \quad (10)$$

Here d is the distance from the viewpoint to the surface, θ_V is the angle between the viewing ray and the surface normal, and D is dependent on individual bump maps. To avoid an instantaneous transition on the silhouette an ϵ is added to the cosine term in the denominator. The constant D should be large if the highest frequency component of the bump map is large. Note that D controls where the function changes from positive to negative, and thus lies midway between displacement-mapping and the BRDF. The formula for determining D is

$$D = c \cdot freq \cdot S$$

where $freq$ is the highest frequency in the bump map and S is the amount the u and v values are scaled. If S is large, then the bump map will be repeated more times over the same area, and the partial derivatives, P_u and P_v, are made shorter by a factor of S. The constant c controls computational effort by globally shifting the scene toward more BRDF or alternatively more displacement. If shadows are included, the shadow terminator should be treated just like the silhouette. Areas far from the terminator are likely to be completely illuminated or shadowed, but on the terminator, displacement-mapping will make the shadowing exact. The parameter given by equation (10) determines the algorithm or algorithms used for rendering. Let the threshold values for choice of renderer be $e1 < e2 < 0 < e3 < e4$. If $T < e1$ then use the BRDF, if $T > e4$ then use displacement mapping, and if $e2 < T < e3$ use redistribution bump mapping. Values of T other than these indicate regions where algorithms are blended. Values of -1, -.3, .3, and 1 respectively, were found to give good results.

4.3 Multiple Levels of Detail

With multiple levels of detail there are many more than two possible transition points. Many other cases need to be considered. The displacement-mapped image of the i^{th} layer is rendered using the BRDF for the $(i-1)^{th}$ layer. As the camera continues to zoom in, the BRDF will switch to bump-mapping and then again to displacement-mapping.

Since each bump map has its own independent transition regions, some areas may have bump-mapping from two or more different levels. Perlin [11] suggests that each set of bumps be limited to a narrow range of frequencies. The result of implementing two levels of detail is shown in Figure 4. The bump map describing the surface detail is broken up into high and low order band-limited frequencies. The low frequencies compose the first level bump map and the high frequencies compose the second level. The left half of Figure 4 is color coded according to the algorithm used to render the most refined level of detail visible. Hence one can see bumpy sections colored yellow to indicate the BRDF from the next lower level was used to render the displaced bumps.

5 Results

5.1 Consistency Comparison

In Figure 5 we can see the four rendering methods compared. The difference between the lighting and viewing ϕ is zero. Note that since the lighting and viewing directions are in alignment the patch becomes brighter for grazing angles. The rows are rendered with bump-mapping, redistribution bump-mapping, BRDF, and displacement mapping respectively. Note that redistribution bump-mapping is far more consistent with the BRDF and displacement-mapping than is ordinary bump-mapping. Figure 6 is a table which

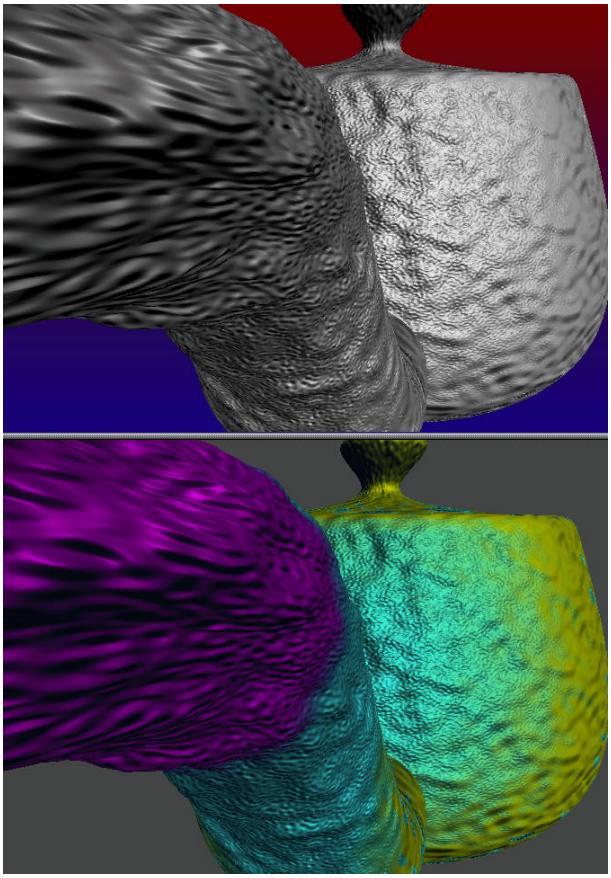

Figure 4: Two levels of bumpy detail. Colors in the bottom half indicate BRDF (yellow), redistribution bump mapping (blue), and displacement mapping (red) in the higher frequency bumps.

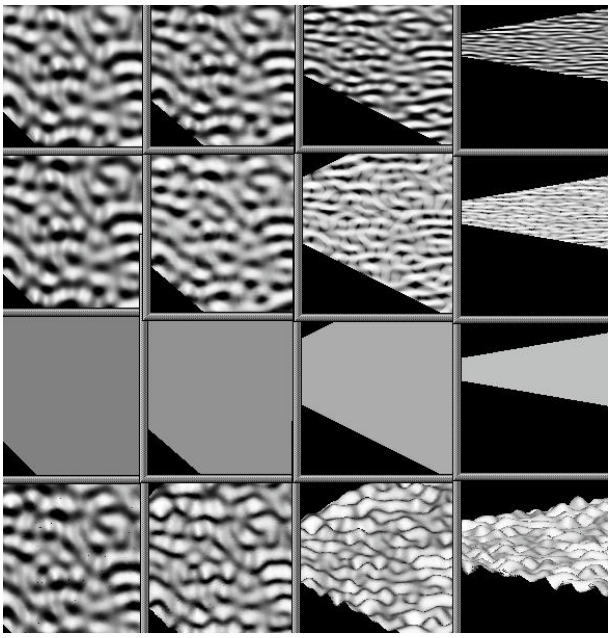

Figure 5: Intensity comparisons. The lighting direction is consistently $\theta_L = \pi/4$. The rows from top to bottom represent bump-mapping, redistribution bump mapping, BRDF, and displacement mapping.

	Θ_v=0	Θ_v=π/6	Θ_v=π/3	Θ_v=4π/9
Bump	128	129	129	129
Redistribution	128	143	170	194
BRDF	129	146	172	192
Displacement	128	146	175	194

Figure 6: Area averaged intensities for the diffuse component.

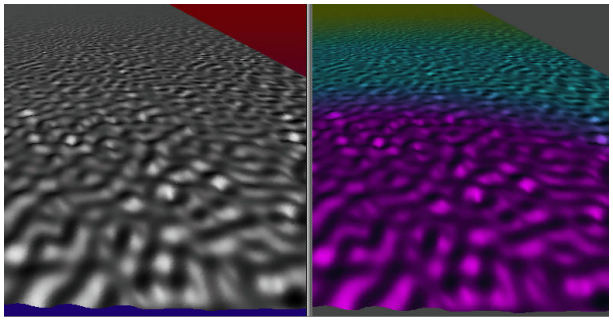

Figure 7: Transitions on a flat surface. BRDF (yellow) in the back, redistribution bump mapping (blue) in the middle, and displacement mappign in the foreground.

shows quantitative results for viewing angles corresponding to those shown in Figure 5.

In Figure 7, a single flat patch is drawn in perspective. Regions in the foreground are clearly displacement-mapped. The middle region is redistribution bump-mapped, and the furthest edge is almost completely shaded with the BRDF. It should be apparent that there is no intensity inconsistency between methods and that the transition is smooth.

5.2 Conclusions

Combining displacement-mapping, bump-mapping and a BRDF into one algorithm makes it possible to explore great scale changes, without changing the geometrical data base. Using a series of bump maps we can generate a variety of rough surfaces simulating different material properties. Objects in the scene will have a complex underlying structure but only the minimum amount of effort necessary to give the impression of complete geometrical representation will be expended. Current animations are restricted by the amount of geometrically represented detail. If the view gets too close to a feature, large drab polygons fill the display. With hierarchy of detail, the polygon level need never be reached, no matter how close the viewer gets. Even at intermediate and far distances the light interacts with flat polygonal surfaces as if they were truly composed of millions of smaller micro-polygons. As a result the otherwise drab polygons become alive with texture and interesting highlights. Those smaller micro-polygons may actually get rendered, but only if the viewer zooms in much closer.

5.3 Future Research

Shadowing is the main enhancement yet to be considered. One way to do the shadowing of displaced bumps is to use the two-pass z-buffer method developed by Williams [15]. Horizon mapping [9] has been shown to generate shadows for bump-mapped images. It will also work for redistribution bump-mapping since the horizon is determined by the u and v parameterization, not the normal. However, this may cause a problem since the rendering is according to a redistributed normal, and the shadows are according to the parameterization. The shadowing may look inappropriate for the rendered bumps. The shadowing for BRDFs can be done using horizon mapping, as was demonstrated by Cabral [3]. Another possibility is to use only the unshadowed normals from a displaced, rendered, and shadowed flat patch to generate the distributions for the BRDF and the redistribution function. The result should be consistent in terms of average intensity, but may not look qualitatively correct.

5.4 Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

Bibliography

[1] B. BECKER. Smooth transitions between bump rendering algorithms during animation. Master's thesis, University of California at Davis, December 1992.

[2] J. F. BLINN. Models of light reflection for computer synthesized pictures. J. George, Ed., vol. 11, 192–198.

[3] B. CABRAL, N. MAX, AND R. SPRINGMEYER. Bidirectional reflection functions from surface bump maps. In *Computer Graphics (SIGGRAPH '87 Proceedings)* (July 1987), M. C. Stone, Ed., vol. 21, 273–281.

[4] R. L. COOK. Shade trees. In *Computer Graphics (SIGGRAPH '84 Proceedings)* (July 1984), H. Christiansen, Ed., vol. 18, 223–231.

[5] R. L. COOK, L. CARPENTER, AND E. CATMULL. The Reyes image rendering architecture. In *Computer Graphics (SIGGRAPH '87 Proceedings)* (July 1987), M. C. Stone, Ed., 95–102.

[6] R. L. COOK AND K. E. TORRANCE. A reflectance model for computer graphics. vol. 15, 307–316.

[7] A. FOURNIER. Normal distribution functions and multiple surfaces. In *Graphics Interface '92 Workshop on Local Illumination*. 1992, pp. 45–52.

[8] J. T. KAJIYA. Anisotropic reflection models. In *Computer Graphics (SIGGRAPH '85 Proceedings)* (July 1985), B. A. Barsky, Ed., vol. 19, 15–21.

[9] N. L. MAX. Horizon mapping: shadows for bump-mapped surfaces. *The Visual Computer 4*, 2 (July 1988), 109–117.

[10] N. L. MAX AND B. BECKER. Bump shading for volume textures. *IEEE Computer Graphics and Applications* (1993). To appear.

[11] K. PERLIN. A unified textural reflectance model. Advanced Image Synthesis course notes, SIGGRAPH '84, July 1984.

[12] K. PERLIN. An image synthesizer. In *Computer Graphics (SIGGRAPH '85 Proceedings)* (July 1985), B. A. Barsky, Ed., vol. 19, 287–296.

[13] K. TORRANCE AND E. SPARROW. Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America 57*, 9 (1967), 1105–1114.

[14] S. H. WESTIN, J. R. ARVO, AND K. E. TORRANCE. Predicting reflectance functions from complex surfaces. In *Computer Graphics (SIGGRAPH '92 Proceedings)* (July 1992), E. E. Catmull, Ed., vol. 26, 255–264.

[15] L. WILLIAMS. Casting curved shadows on curved surfaces. In *Computer Graphics (SIGGRAPH '78 Proceedings)* (August 1978), vol. 12, 270–274.