ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/221474899

Approximation of Glossy Reflection with Prefiltered Environment Maps.

Conference Paper · January 2000

Source: DBLP

CITATIONS
58

READS
198

2 authors, including:

Michael D. Mccool
Intel
99 PUBLICATIONS **2,256** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

RapidMind/ArBB View project

All content following this page was uploaded by Michael D. Mccool on 11 July 2014.

The user has requested enhancement of the downloaded file.

Approximation of Glossy Reflection with Prefiltered Environment Maps

Jan Kautz
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Michael D. McCool
Department of Computer Science
University of Waterloo, Waterloo, Canada

Abstract

A method is presented that can render glossy reflections with arbitrary isotropic bidirectional reflectance distribution functions (BRDFs) at interactive rates using texture mapping. This method is based on the well-known environment map technique for specular reflections.

Our approach uses a single- or multilobe representation of bidirectional reflectance distribution functions, where the shape of each radially symmetric lobe is also a function of view elevation. This approximate representation can be computed efficiently using local greedy fitting techniques. Each lobe is used to filter specular environment maps during a preprocessing step, resulting in a three-dimensional environment map. For many BRDFs, simplifications using lower-dimensional approximations, coarse sampling with respect to view elevation, and small numbers of lobes can still result in a convincing approximation to the true surface reflectance.

Key words: Environment map, glossy reflection, texture mapping, bidirectional reflectance distribution function.

1 Introduction

The environment map technique [2] is widely used to approximate reflections in real-time rendering. Environment maps are an approximation technique because they make an assumption that is often not true, namely that the environment is far from the reflecting surface. Despite this, they are effective and efficient, and can be used to build more sophisticated techniques.

Heidrich and Seidel [12] preblurred environment maps with the Phong model [18] to approximate glossy reflectance in real time. However, the technique was limited to Phong lobes. Variation of lobe shape with incident angle was not supported, although self-shadowing and Fresnel scale factors that varied with the incident and view direction have been demonstrated.

We have extended and refined this idea to support the generation of glossy reflections with a relatively general class of isotropic bidirectional reflectance distribution functions (BRDFs). For a number of elevation angles we approximate the given BRDF with a sum of radially symmetric lobes. In our representation each lobe must be symmetric around some axis, but need not be a Phong lobe: the radial shape function is arbitrary, and is derived from the data. Furthermore, the axis of each lobe may be offset from the reflection direction.

For "glossy" BRDFs with large peaks, lobe-fitting can be performed using a greedy technique that is more efficient than the global optimization techniques previously used for multilobe BRDF representations [15]. While not optimal, the greedy fitting technique is easy to implement and produces usable results quickly.

Once the lobes have been obtained we prefilter a specular environment map with each one. This results in a set of prefiltered two-dimensional environment maps, for each lobe and for each sampled elevation angle.

Lobes are then tracked over multiple elevation angles, to find consistent and coherent sequences of lobes that can be interpolated. The resulting stacks of prefiltered two-dimensional environment maps (each stack resulting from a sequence of coherent lobes filtered against the original environment map) can be stored in three-dimensional environment maps.

The new third dimension corresponds to the view elevation angle while the other two dimensions correspond to the incident elevation and azimuth angles. Use of three-dimensional environment maps permits the representation of important and common effects such as increased reflection sharpness at glancing angles.

By summing up multiple lobes an arbitrary BRDF can be approximated to whatever precision is desired; for the isotropic examples we have tested, a small number of lobes (less than 5, often as few as 1) have been visually satisfactory. Furthermore, for machines that do not support three-dimensional filtering, we have found that ignoring the dependence of the shape of the lobe on elevation angle is adequate for some BRDFs.

2 Prior Work

In Heidrich and Seidel's technique [12], specular environment maps are prefiltered with view-independent, radially symmetric Phong (exponentiated cosine) lobes. During rendering, view-dependent Fresnel scale factors are applied but the basic shape of the lobe is not varied.

Each entry of the Phong-prefiltered environment map

contains not only the radiance coming from $\hat{\omega}_i$ (like normal environment maps do), but also the radiance from a larger region integrated against the Phong lobe, as depicted in Figure 1.

The original Phong model was chosen because (after weighting by the cosine of the incident angle) Phong lobes are symmetric around the reflected view direction and the shape does not change with view elevation. However, Phong lobes are not realistic *exactly* because their shape does not change! In particular, at glancing angles Heidrich and Seidel's technique actually includes energy from below the horizon in the reflection. Normally, the lobe should become thinner and sharper.

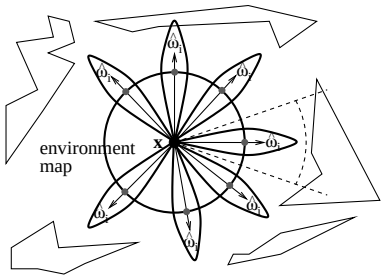

Figure 1: *Phong environment maps.*

Cabral et al. [3] proposed a technique that uses a set of prefiltered environment maps, which they call radiance environment maps. An original specular environment map is prefiltered with a BRDF using a specific viewing position. This is done for a number of viewing positions, generating a set of intrinsically view-dependent prefiltered environment maps.

In order to render a reflective object a new environment map is generated for every new viewpoint using three of the prefiltered environment maps, which are warped and blended together. This new environment map is then applied to the reflective object. In order to compute the warping the central reflection direction of the BRDF's lobe has to be known, and an assumption is made that the BRDF is radially symmetric about this direction. This technique also assumes an orthographic viewer. See the conclusions section for a comparison with our method.

Other researchers have looked at deriving multilobe representations for BRDFs. Lafortune et al. [15] used a generalized Phong lobe model and Walter et al. [21] used multiple Phong lobes to approximate the reflection of a global illumination solution from a surface with an arbitrary BRDF. Fournier [5] used Phong lobes to approximate the distribution of normals. Our work differs from these: we drop the restriction to Phong lobes entirely, and use local greedy fitting techniques rather than global optimization.

There have been other techniques for representing interactive glossy reflections that are not as closely related to our work as the previous methods. We will briefly mention and comment on a few that attempt to target high-performance rendering.

Stürzlinger and Bastos [19] used photon maps for interactive glossy reflections; photons were "splatted" and weighted with an arbitrary BRDF. Diefenbach and Badler [4] used multi-pass methods (stochastic multi-sampling) to generate glossy reflections; this performs the filtering of the environment map (that we do in a prepass) using Monte Carlo integration at runtime. Miller et al. [17] used surface light fields to store and reconstruct glossy reflections. Lischinski and Rappoport [16] used layered depth images to produce glossy reflections. Bastos et al. [1] used a space-variant convolution filter in screen-space to produce glossy reflections; we apply our convolution in environment-map space during a preprocess, and so have fewer image quality limitations.

Finally, previous work by Kautz and McCool [14] addressed the problem of computing local illumination and reflectance from point sources in real time from surfaces with arbitrary BRDFs, using a separable decomposition of the BRDF. The current technique is complementary to that work, in that we address the problem of using arbitrary BRDFs in real-time rendering, but with environment maps instead of point sources for illumination.

The algorithm we will present has three phases: multilobe BRDF approximation (which is done once per BRDF), environment map prefiltering (which is done once per BRDF for each new specular environment map), and rendering. These are presented separately in the following sections, and will be followed by a sampling of the resulting images.

3 BRDF Approximation

A shift-invariant BRDF $f(\hat{\omega}_o, \hat{\omega}_i)$ can be approximated using a sequence of lower-dimensional approximations for a set of sampled viewing directions $\hat{\omega}_o$. The viewing direction $\hat{\omega}_o$ is set to a number of different values and then a two-dimensional approximation algorithm is run for each value. We will denote a BRDF "slice" with $\hat{\omega}_o$ fixed as $f_{\hat{\omega}_o}(\hat{\omega}_i)$, which we will call a *lobe*; this is in fact a two-dimensional function of $\hat{\omega}_i$.

For the prefiltering of environment map we *must* use radially symmetric lobes for each approximation. Assume we are given a BRDF with a non-radially symmetric lobe for a given local viewing direction $\hat{\omega}_o$, as seen on the left side of Figure 2. Assume the reflected viewing vector is used to index the environment map. The environment map stores the radiance prefiltered with (integrated against) that BRDF. If we rotate the surface and look at it from a different viewpoint so that we get the same reflected viewing vector (see right side of Figure 2),

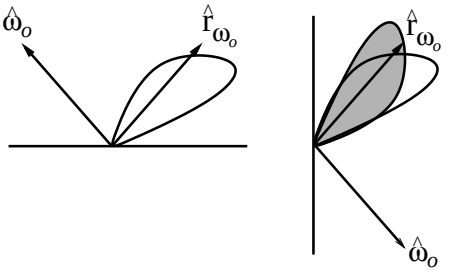

Figure 2: *Radially symmetric lobes are necessary.*

we will of course obtain the same value from the environment map, as we are indexing it with the same texture coordinates. However, the orientation of the actual lobe (grey) now differs from the orientation of the lobe with which the environment map was filtered. To avoid this problem we have to use lobes which are radially symmetric.

The use of only radially symmetric BRDFs might seem to limit this approach, but in practice many BRDFs have lobes that are close to radially symmetric for fixed viewing angles. We can use multiple radially symmetric lobes as basis functions to approximate BRDFs that do not show this symmetry.

3.1 Approximation with a Single Lobe

An existing *isotropic* BRDF $f(\hat{\omega}_o, \hat{\omega}_i)$ can be approximated with a separate lobe for each $\hat{\omega}_o = (\theta_o, 0)$, $\theta_o \in \{\theta_{o,1}, \theta_{o,2}, \ldots, \theta_{o,k}\}$, where $\theta_o = \hat{n} \cdot \hat{r}_v$; see Figure 6. In this paper we only consider isotropic BRDFs to keep the dimensionality of the result under control.

The actual fitting uses a greedy, heuristically-driven algorithm. This approach works in this case because "glossy" surfaces that give a recognizable reflection of the environment will be likely to have a peak near the direction of the reflection vector.

First, we find the maximum peak of $f_{\hat{\omega}_o}(\hat{\omega}_i)$; denote the direction in which the maximum lies using $\hat{\omega}_{i/\mathrm{peak}}$. The offset elevation angle will be denoted by $\theta_{i/\mathrm{off}}$ and the offset azimuth by $\phi_{i/\mathrm{off}}$. These angles are the difference between the expected reflection direction (i.e. the reflected viewing direction) and the actual peak direction. For a single lobe approximation and an isotropic BRDF, the offset azimuth angle $\phi_{i/\mathrm{off}}$ must in fact must be 0, but this will not be true for multilobe approximations.

We assume that $\hat{\omega}_{i/\mathrm{peak}}$ lies in the center of the lobe given by $f_{\hat{\omega}_o}(\hat{\omega}_i)$, and create profile curves for the lobe by taking samples in radial directions away from the peak; see Figure 3. Every curve that is generated by the intersection of a plane going through $\hat{\omega}_{i/\mathrm{peak}}$ and the lobe itself is a profile curve; see Figure 3. Then we average all the profile curves and create a mean profile. This mean profile is used as the profile curve for our radially symmetric approximation lobe. The approximation lobe $p_{\theta_o}(\vartheta_i) \approx f_{\hat{\omega}_o}(\hat{\omega}_i)$ (where $\vartheta_i = \hat{\omega}_{i/\mathrm{peak}} \cdot \hat{\omega}_i$) is computed by rotating the mean profile around $\hat{\omega}_{i/\mathrm{peak}}$. This obviously makes the lobe radially symmetric; see Figure 3. The approximation is compact: the approximation's parameters consist of the profile curve and peak offset angles. The profile curves are stored using discrete samples.

Because of the averaging process the overall energy of our approximation lobe might be different from the original BRDF lobe, which results in reflections with the wrong brightness. To correct that error we compute the total hemispherical reflectivity of the approximation, compare this with the original BRDF, and scale our approximation accordingly.

3.2 Using a Single 2D Lobe

We can further approximate a given BRDF to save memory. Some glossy BRDFs maintain roughly the shape of their lobe as θ_o changes. When $\hat{\omega}_o$ varies, only a scale factor change is required. So another obvious approximation calculates only *one* average lobe $p(\vartheta_i)$, either by averaging all profile curves or by choosing a "characteristic" profile for a specific $\hat{\omega}_o$. To approximate the actual lobes $f(\hat{\omega}_o, \hat{\omega}_i)$ we assign a weight for each $\hat{\omega}_o$ to the single lobe $p(\hat{\omega}_i)$, to get $f(\hat{\omega}_o, \hat{\omega}_i) \approx k(\theta_o)p(\vartheta_i)$.

This representation has the advantage that it can be rendered with only a single two-dimensional environment map. The weights can be either assigned as colors at the vertices or put in an extra one-dimensional texture map—much like the Fresnel term used by Heidrich [12]. However, with this approximation we will not be able to capture such effects as increased sharpness or reflections at glancing angles.

3.3 Using Multiple Lobes

Some BRDFs might be difficult to approximate by a single lobe. In that case we can use multiple lobes. Each of these "basis lobes" has to be radially symmetric, as explained earlier, but may be offset by various amounts from the reflection vector.

There are two different approaches to fit multiple lobes to a BRDF: global methods and greedy iterative methods.

We tried two global techniques (simulated annealing and branch and bound), which both failed to yield a satisfying solution, even when we restricted ourselves to Phong lobes[1]. Therefore, we will only present an extension to the greedy technique that we have found works well in practice.

Our iterative greedy fitting scheme works almost like the single shape lobe fitting. First, find the maximum peak at $\hat{\omega}_{i/\mathrm{peak}}$ of $f_{\hat{\omega}_o}(\hat{\omega}_i)$ and represent this direction us-

[1]This has also been noted by Walter et al. [21] in a similar situation.

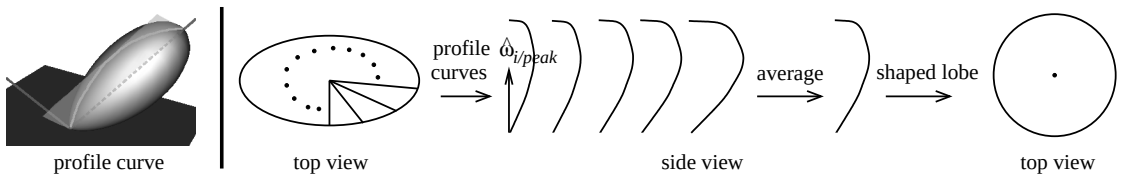

Figure 3: *Single lobe approximation.*

ing the offset angles $\theta_{i/\text{off}}$ and $\phi_{i/\text{off}}$, as described earlier. Then, create profile curves around $\hat{\omega}_{i/\text{peak}}$ and choose the *minimum* profile curve for the approximation lobe $p_{\theta_o}(\vartheta_i)$. A minimum profile curve is defined as the profile curve that generates a lobe $p_{\theta_o}(\vartheta_i)$ which is smaller than $f_{\hat{\omega}_o}(\hat{\omega}_i)$ for all ϑ_i. We are going to use additional lobes that can pick up the not-yet approximated parts, but we do not want negative residuals[2]. This lobe is subtracted from the original BRDF (both are represented using samples). On the remaining unapproximated energy, we rerun the algorithm to get another lobe; see Figure 4. We stop the algorithm when we reach a given number of lobes.

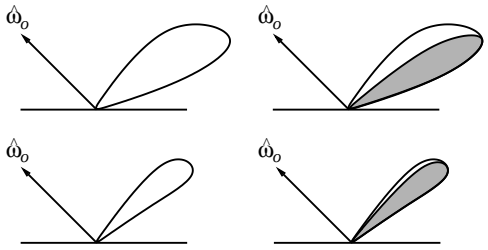

Figure 4: *Approximation with multiple lobes.*

As explained earlier, approximation is performed for each $\hat{\omega}_o$ separately, then we blur an existing environment map and store stacks of these maps in three-dimensional environment maps. In the multiple-lobe case we will get more than one lobe and therefore we will need to create more than one three-dimensional environment map. Each of these three-dimensional environment maps should be generated using a consistent and coherent selection of lobes.

What this means is shown in Figure 5. You can see a lobe varying over θ_o. In our representation this variation in shape is approximated by linearly interpolating approximations found for specific values of $\hat{\omega}_o$. We want the variation in shape to be gradual, and to do this we have to interpolate the "right" approximations. Profile lobes that are consistent and spatially coherent are marked with the same letter (L, M, and R). Each of the three environment maps should be generated with only spatially coherent profile lobes, i.e. environment map one only with L lobes, environment map two only with M lobes, and so on.

This means that after finding different lobes for different $\hat{\omega}_i$, we cluster them by similar $\theta_{i/\text{off}}$ and $\phi_{i/\text{off}}$ values.

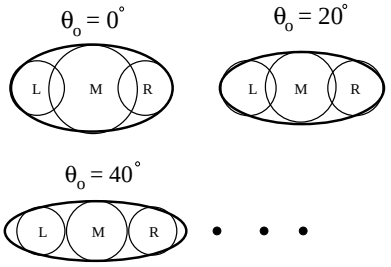

Figure 5: *Spatial coherence of multiple lobes.*

4 Prefiltering

Once we have a BRDF approximation an existing high-dynamic range specular environment map $E(\hat{r}_v)$ needs to be filtered by it. A specular environment map only depends on the reflected viewing vector. We integrate these against the radially symmetric lobes to obtain the three-dimensional environment map representation of the glossy reflection:

$$E_{\text{glossy}}(\hat{r}_v, \hat{n}\cdot\hat{r}_v) = \int_\Omega p_{\hat{n}\cdot\hat{r}_v}(\hat{r}_v\cdot\hat{l})\ (\hat{n}\cdot\hat{l})E(\hat{l})\ d\sigma(\hat{l}) \approx \quad (1)$$

$$(\hat{n}\cdot\hat{r}_v)\int_\Omega p_{\hat{n}\cdot\hat{r}_v}(\hat{r}_v\cdot\hat{l})E(\hat{l})\ d\sigma(\hat{l}) \quad (2)$$

All the vectors are depicted in Figure 6; $\hat{n}$ is the surface normal, $\hat{v} = \hat{\omega}_o$ in world coordinates, $\hat{l} = \hat{\omega}_i$ in world coordinates, and the reflection vector $\hat{r}_v = 2(\hat{v}\cdot\hat{n})\hat{n} - \hat{v}$, $d\sigma(\hat{l})$ is the projected solid angle measure. The lobes $p_{\hat{n}\cdot\hat{r}_v} = p_{\theta_o}$ actually only depend on the angle between the reflected viewing direction and the incident direction because they are by definition radially symmetric around $\hat{r}_v$ (we will *index* the new maps by the offset vectors, but filter them with the reflection vector).

[2]If signed arithmetic is supported, approximation techniques that generate signed lobes would be appropriate. Hardware is starting to become available which will be capable of this.

As you can see in Equation 1, we need not only to filter the environment map with the approximation lobe, but also have to include the cosine between the normal $\hat{n}$ and the incident direction $\hat{l}$.

Fortunately, we can approximate $\hat{n} \cdot \hat{l}$ with $\hat{n} \cdot \hat{r}_v$; otherwise, we would have another dependence on $\hat{n}$, which we cannot capture with a three dimensional environment map. At first sight, this approximation of $\hat{n} \cdot \hat{l}$ seems to be very crude. We assume, however, that BRDFs yielding interesting glossy reflections will have a fairly slim lobe in the direction of $\hat{r}_v$. In the cases of interest, $\hat{n} \cdot \hat{l}$ is therefore usually close to the constant value $\hat{n} \cdot \hat{r}_v$ where the integrand is large.

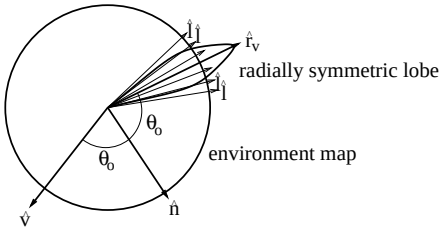

Figure 6: *The vector $\hat{n}$ is the surface normal, $\hat{v}$ is the viewing direction, $\hat{r}_v$ is the reflected viewing direction, and the $\hat{l}$ are the incident directions.*

5 Rendering

For our three different approximation methods (single lobe, single 2D lobe, multiple lobes), we basically use the same rendering algorithm. For each vertex of the reflective object, compute appropriate texture coordinates to represent the appropriate offset of the vector $\hat{r}_v$ and the value of $\hat{n} \cdot \hat{r}_v$. The actual representation of the offset vector as texture coordinates depends on the way the view-independent environment map was stored, e.g. as a parabolic map [11] on standard hardware, or as a cube map on hardware which supports them. Then we render our object with texture mapping turned on, and sum multiple lobes using either the accumulation buffer, multitexturing, or compositing, as appropriate.

5.1 Single Lobe Approach

Assume that we have already filtered an existing environment map $E(\hat{r}_v)$ with a shaped lobe approximation of the BRDF, so we have a three-dimensional environment map $E_{\text{glossy}}(\hat{r}_v, \vec{n} \cdot \hat{r}_v)$; see Equation 2.

This three-dimensional environment map consists of several two-dimensional environment maps, each of which depends only on $\hat{r}_v$. The value $\hat{n} \cdot \hat{r}_v$ is the third parameter of the three-dimensional environment map.

To render an object with this approximation, at each vertex compute $\hat{r}_v$ and compute the offset vector by adding $\theta_{i/\text{off}}$. Place the texture-coordinate representation of $\hat{r}_v$ in the first two texture coordinates, then compute $\hat{n} \cdot \hat{r}_v$ and place it in the third texture coordinate. The factor $\hat{n} \cdot \hat{r}_v$ in Equation 2 can be either incorporated into the environment map or assigned as the color at the vertices.

5.2 Single 2D Lobe Approach

In this approximation we only have a single filtered two-dimensional environment map, which was computed using the following filter:

$$E_{\text{glossy2D}}(\hat{r}_v) \quad = \quad \int_{\Omega} p(\hat{r}_v) E(\hat{l})\, d\sigma(\hat{l}). \qquad (3)$$

As you can see the weight $k(\theta_o) = k(\hat{n} \cdot \hat{r}_v)$ is not included in the environment map, and neither is the factor $\hat{n} \cdot \hat{r}_v$ (which has to be used because we are dealing with an arbitrary isotropic BRDF).

At each vertex, we assign the weight $k(\theta_o)(\hat{n} \cdot \hat{r}_v)$ as the color. This takes care of the weight $k(\theta_o)$ and the factor $\hat{n} \cdot \hat{r}_v$. Now, we render the object using the two-dimensional environment map. Alternatively we can put the weights $k(\theta_o)$ in a texture map and use a two-pass rendering method, but as the $k(\theta_o)$ usually vary gradually, the first method was sufficiently accurate for the BRDFs we tested.

5.3 Multi-Lobe Approach

For a multilobe approximation we have several three-dimensional environment maps $E^l_{\text{glossy}}(\hat{r}_v, \hat{n} \cdot \hat{r}_v)$, each of has been blurred with a different set of lobes $p^l_{\theta_o}(\vartheta_i)$, as explained in Section 3.

For each environment map we basically run the same rendering algorithm that we described for the single lobe approach, and sum the results (using compositing or the accumulation buffer). The only additional difference is that we have to offset $\hat{r}_v$ by both $\theta_{i/\text{off}}$ and a non-zero $\phi_{i/\text{off}}$.

6 Results

We have validated our technique with several examples, including Ward's model, the HTSG model, and measured car lacquer. Our tests have shown that our multilobe approximations yield RMS errors equivalent to the RMS errors for separable approximations with a similar number of terms [13]; the RMS errors were measured between the original BRDF and the approximated BRDF. However, since only positive lobes were used, negative residual error cannot be corrected and after six to eight lobes the error begins to increase rather than decrease.

Three test cases are shown in Figures 9–18, where they are compared with renderings using the actual BRDF filtered against the same environment map on a per-pixel basis. Note that we are testing the error of the BRDF

approximation, *not* the error of the environment map reflectance approximation. In all cases we used a view-independent dual parabolic environment map of an office scene shown in Figure 8.

The first example applies our technique to the measured Cayman car lacquer from the Computer Graphics Group at Cornell. The results for the single lobe method can be seen in Figure 10 and for the single 2D lobe method in Figure 11. A rendering with the true BRDF (filtering is done for every pixel) is shown in Figure 9. Slices of the prefiltered environment map for the single lobe approximation can be seen at different elevation angles in Figure 8. The 3D texture map used had a resolution of $128 \times 128 \times 16$ slices in the z (elevation angle) direction. You can see that the sharpness and the color of the environment maps changes with elevation angle. This is visible on the lid of the teapot; the reflection in the 2D approximation is blurrier than it should be, although otherwise the approximation is adequate.

The second wavelength dependent BRDF we tried was the HTSG model for rough copper [9]. The results can be seen in Figure 13 for the single lobe method and in Figure 14 for the single 2D lobe method (compare with Figure 12). A few slices of the prefiltered environment map for the single lobe approximation are depicted in Figure 8, where you can again see increased sharpness with decreasing elevation.

Now we would like to demonstrate the differences between the multiple lobe method, the single lobe method, and the two-dimensional lobe method. In Figure 15 – 18 you can see a side-by-side comparison of a teapot rendered with the same BRDF (Ward's model) using these methods. The relative mean square error for this BRDF using the different approximation methods is depicted in Figure 7.

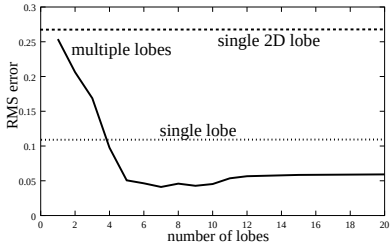

Figure 7: *RMS error for Ward's model.*

These examples show that while the approximations are not exact, they are visually pleasing and adequate for real-time rendering applications. Two dimensional approximations will in general show more error, but in our tests were still visually close to the reflection generated by the original BRDF.

The rendering speed mainly depends on the graphics hardware, since only the texture coordinate generation needs to be done on the host. On an SGI Octane MXE we achieve 25 fps with the teapot model (4000 triangles) and a single lobe approximation. The three-lobe approximation is almost exactly three times slower. The teapot model with a single 2D approximation can be rendered with 33 fps.

7 Extensions

The greedy shaped lobe fitting algorithm may generate an inefficient approximation because it assumes that useful radially symmetric lobes are centered around peaks. Unfortunately, this assumption may be false.

First of all, measured BRDF data might have a peak somewhere that might be just a noisy data sample, but we want to maximize the total *volume*, of a lobe, not just its height. Secondly, BRDFs may contain radially symmetric lobes not centered around peaks, i.e. the centers might be "depressed". In this case the greedy fitting algorithm may use a large number of lobes to build a poor approximation when a single lobe would give a good approximation.

A more robust general-purpose fitting algorithm for radially symmetric lobes would be interesting. There are two possible approaches.

First, a heuristically-driven algorithm could be used that would look in several likely places (peaks, around the view direction for retroreflection, around the normal for generalized diffuse reflection, and finally around the reflected view direction), try to find a radially symmetric lobe for each of these vectors, and then pick the best one (in terms of maximum volume).

Secondly, a multiresolution search technique could be used. An exhaustive search could be made for the maximal volume radially symmetric lobe, but against a filtered and down-sampled version of the BRDF. Once one or more candidate lobes have been found, their positions and profile curves can be refined by increasing the resolution. This avoids the cost of exhaustively searching for all possible lobes at high resolution, but should give similar results.

Both these extension would still be greedy sequential algorithms, and so may not find optimal solutions. However, both approaches are fast and should robustly produce useful practical results.

Finally, the current technique can only handle anisotropic BRDFs by using many lobes and a 4D texture, which is inefficient. It would be interesting to use steerable filters [6] or spherical harmonics to implement anisotropic lobes to generalize the technique presented here.

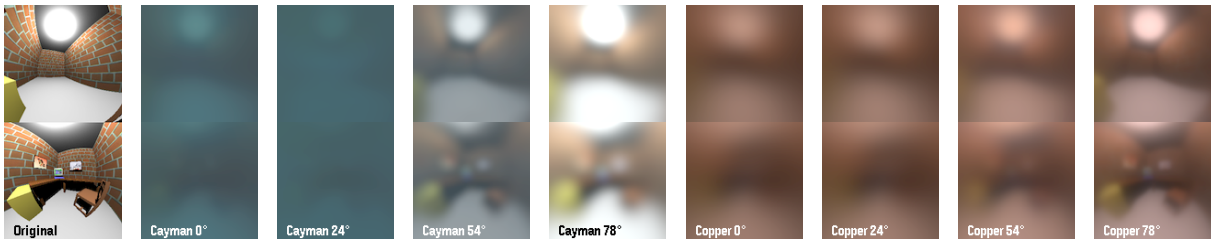

Figure 8: *Parabolic environment maps of an office scene.*

Figure 9: *Teapot with the cayman car lacquer reflecting the office (original).*

Figure 10: *Teapot with the cayman car lacquer reflecting the office (3D).*

Figure 11: *Teapot with the cayman car lacquer reflecting the office (2D).*

Figure 12: *Copper teapot reflecting the office (original).*

Figure 13: *Copper teapot reflecting the office (3D).*

Figure 14: *Copper teapot reflecting the office (2D).*

Figure 15: *Per-pixel filtering with original BRDF.*

Figure 16: *Multiple lobes method (three lobes).*

Figure 17: *Single lobe method.*

Figure 18: *Single 2D lobe method.*

Teapot with Ward's model ($k_d = 0.0, k_s = 0.80, \alpha_x = \alpha_y = 0.04$) reflecting the office.

8 Conclusions

We have presented a technique for representing arbitrary BRDFs using multiple radially symmetric lobes. This representation can be used to prefilter existing specular environment maps in order to render glossy surfaces at interactive rates on existing hardware. The straightforward greedy fitting algorithm we use is most appropriate for BRDFs with large peaks, but these are the BRDFs that produce recognizable reflections.

For several test BRDFs, objects rendered with environment maps that were filtered with our representation are visually convincing and can be rendered at real-time rates on appropriate hardware.

Our method is not restricted to a certain kind of isotropic BRDFs, since we can use a multilobe approximation, but memory requirements are excessive if many lobes are needed, making it unattractive. Therefore it is better to either restrict this method to BRDFs with almost radially symmetric lobes, which is true for many BRDFs, or to accept a loss in quality and use a single lobe or even a single 2D lobe only. Fortunately, our method enables the user to control the accuracy/memory ratio.

Cabral's technique turns out to be closely related to the technique presented here; in fact, our technique can be considered an alternative (and improved) implementation strategy for Cabral's technique. Cabral's method basically uses a sparse representation of the prefiltered environment map. However, we do not require per-frame warping, since we use view-independent environment maps, and we do not make the assumption of an orthogonal (constant) view direction. We have also gone farther in deriving multilobe representations of BRDFs.

Despite the different techniques used to store and reconstruct the prefiltered environment map data, our multilobe BRDF representations are compatible and have the same limitation to radially-symmetric lobes and to isotropic BRDFs.

Acknowledgements

Wolfgang Heidrich was very helpful in supplying code for rendering reflective objects using view-independent parabolic environment maps. He also supplied code for filtering environment maps.

The implementation of the He (HTSG) reflectance model was done by Glenn Evans at the University of Waterloo.

The core of this research was performed at the University of Waterloo, Canada, and was supported by a grant from the NSERC (the National Science and Engineering Research Council of Canada).

9 References

[1] R. Bastos, K. Hoff, W. Wynn, and A. Lastra. Increased Photorealism for Interactive Architectural Walkthroughs. *1999 ACM Symposium on Interactive 3D Graphics*, pages 183–190, April 1999.

[2] J. Blinn and M. Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19:542–546, 1976.

[3] B. Cabral, M. Olano, and P. Nemec. Reflection space image based rendering. In *Proc. SIGGRAPH*, pages 165–170, August 1999.

[4] P. Diefenbach and N. Badler. Multi-Pass Pipeline Rendering: Realism For Dynamic Environments . *1997 ACM Symposium on Interactive 3D Graphics*, pages 59–70, April 1997.

[5] A. Fournier. Filtering normal maps and multiple surfaces. Technical Report TR-92-41, University Of British Columbia, Department of Computer Science, 1992.

[6] W. Freeman and E. Adelson. The Design and Use of Steerable Filters. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991.

[7] N. Greene. Applications of World Projections. In *Proceedings of Graphics Interface '86*, pages 108–114, May 1986.

[8] P. Haeberli and M. Segal. Texture Mapping As A Fundamental Drawing Primitive. In *Fourth Eurographics Workshop on Rendering*, pages 259–266. Eurographics, June 1993.

[9] X. He, K. Torrance, F. Sillion, and D. Greenberg. A comprehensive physical model for light reflection. In *Proc. SIGGRAPH*, pages 175–186, July 1991.

[10] W. Heidrich. *High-quality Shading and Lighting for Hardware-accelerated Rendering*. PhD thesis, Universität Erlangen-Nürnberg, 1999.

[11] W. Heidrich and H.-P. Seidel. View-Independent Environment Maps. In *Eurographics/SIGGRAPH Workshop on Graphics Hardware*, pages 39–45, 1998.

[12] W. Heidrich and H.-P. Seidel. Realistic, Hardware-accelerated Shading and Lighting. In *Proc. SIGGRAPH*, pages 171–178, August 1999.

[13] J. Kautz. Interactive Reflection with Arbitrary BRDFs. Master's thesis, University of Waterloo, Waterloo, Canada, 1999.

[14] J. Kautz and M. McCool. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In *Tenth Eurographics Workshop on Rendering*, pages 281–292, June 1999.

[15] E. Lafortune, S.-C. Foo, K. Torrance, and D. Greenberg. Non-Linear Approximation of Reflectance Functions. In *Proc. SIGGRAPH*, pages 117–126, August 1997.

[16] D. Lischinski and A. Rappoport. Image-Based Rendering for Non-Diffuse Synthetic Scenes. *Nineth Eurographics Workshop on Rendering*, pages 301–314, June 1998.

[17] G. Miller, S. Rubin, and D. Ponceleon. Lazy Decompression of Surface Light Fields for Precomputed Global Illumination. *Nineth Eurographics Workshop on Rendering*, pages 281–292, June 1998.

[18] B.-T. Phong. Illumination for computer generated pictures. *Comm. ACM*, 18(6):311–317, June 1975.

[19] W. Stürzlinger and R. Bastos. Interactive Rendering of Globally Illuminated Glossy Scenes. In *Eighth Eurographics Workshop on Rendering Workshop*, pages 93–102. Eurographics, June 1997.

[20] D. Voorhies and J. Foran. Reflection Vector Shading Hardware. In *Proc. SIGGRAPH*, pages 163–166, July 1994.

[21] B. Walter, G. Alppay, E. Lafortune, S. Fernandez, and D. Greenberg. Fitting Virtual Lights for Non-Diffuse Walkthroughs. In *Proc. SIGGRAPH*, pages 45–48, August 1997.

[22] G. Ward. Measuring and modeling anisotropic reflection. In *Proc. SIGGRAPH*, pages 265–272, July 1992.