Thirteenth Eurographics Workshop on Rendering (2002)
P. Debevec and S. Gibson (Editors)

Fast, Arbitrary BRDF Shading for Low-Frequency Lighting Using Spherical Harmonics

Jan Kautz[1], Peter-Pike Sloan[2] and John Snyder[2]

[1] Max-Planck-Institut für Informatik, Saarbrücken, Germany
[2] Microsoft Research, Redmond, WA, USA

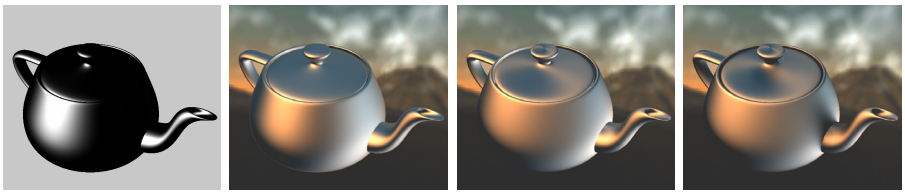
Abstract

Real-time shading using general (e.g., anisotropic) BRDFs has so far been limited to a few point or directional light sources. We extend such shading to smooth, area lighting using a low-order spherical harmonic basis for the lighting environment. We represent the 4D product function of BRDF times the cosine factor (dot product of the incident lighting and surface normal vectors) as a 2D table of spherical harmonic coefficients. Each table entry represents, for a single view direction, the integral of this product function times lighting on the hemisphere expressed in spherical harmonics. This reduces the shading integral to a simple dot product of 25 component vectors, easily evaluatable on PC graphics hardware. Non-trivial BRDF models require rotating the lighting coefficients to a local frame at each point on an object, currently forming the computational bottleneck. Real-time results can be achieved by fixing the view to allow dynamic lighting or vice versa. We also generalize a previous method for precomputed radiance transfer to handle general BRDF shading. This provides shadows and interreflections that respond in real-time to lighting changes on a preprocessed object of arbitrary material (BRDF) type.

Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Color, shading, shadowing, and texture.

1. Introduction

Fully general BRDFs are necessary to describe many interesting kinds of materials, including brushed metals and fabrics. Interactive shading with such BRDFs has taken two approaches. The first simply evaluates the BRDF at a number of points[9,11,14]. While efficient for a few point lights, this approach is slow for large lights since it requires light integration over many directions. The second approach precomputes and tabulates the lighting integral by pre-convolving the incident lighting. Although this table reduces to two dimensions for mirror-like surfaces[8], high-dimensional tables are required for general BRDFs[13]. Phong-like BRDFs require 3D tables where the incident lighting is convolved with kernels of variable radius; anisotropic BRDFs require as much as 5D tables. Pre-convolved tables are costly to compute and store and thus difficult to use with dynamic lighting. Pre-convolution also ignores shadowing and interreflections[19].



(a) point light *(b) glossy* *(c) anisotropic* *(d) shadowed*

Figure 1: *Previous real-time shading methods are limited to point lights (a) or allow smoother lighting environments but restricted to Phong-like glossy BRDF models (b). We generalize to area lighting of arbitrary BRDF models, like anisotropic brushed metal (c) and include shadows (d) at little extra cost. Our method renders this model at 2Hz; 10/50Hz after fixing either the view/light.*

Our approach differs by using a spherical harmonic (SH) basis for lighting and BRDF functions. Essentially, the BRDF is tabulated not in terms of response to directional lights, but to large lights covering the sphere via the orthonormal SH basis. The lighting environment also is projected into the SH basis using a fast, on-the-fly calculation[19]. For low-frequency lighting, very few coefficients are required to accurately compute the lighting integral (25 components). The calculation is simple and fast even for arbitrary BRDFs, requiring no high-dimensional tables of pre-convolved lighting. Our approach allows spatially varying materials by tabulating the BRDF with varying parameters. Furthermore, our method can also be easily combined with precomputed transfer[19] to obtain real-time shadows and interreflections over preprocessed objects. Our method's benefits are highlighted in Figure 1.

2. Previous Work

Preconvolved Environment Maps. Blinn and Newell[4] proposed environment maps to approximate specular reflections. Greene[8] observed that a pre-convolved environment map could be used to simulate diffuse reflections.

Since then, several approaches have been proposed to simulate glossy reflections based on pre-filtered environment maps[6,8,9,12,13]. These algorithms assume a simple, fixed BRDF model[8,9,13,16] (e.g, Phong model, Banks model[2]) or generalize but only to isotropic, radially symmetric BRDFs[12]. More precisely, not the BRDF itself but the *BRDF product function*, i.e. the BRDF multiplied by the cosine factor (Section 3), must be an isotropic, radially symmetric function of the light direction; in other words, a Phong-like model. Such pre-filtering methods thus neglect the cosine factor unless the parameterization is view-dependent, as proposed by Cabral et al.[6]. But their method is restricted to isotropic BRDFs about a central reflection direction. None of these methods handle spatially varying BRDFs well since they require even higher-dimensional tables to account for the varying parameters.

Generally, these methods cannot change the lighting environment on-the-fly since an expensive convolution must be applied to the environment map. Graphics hardware has been used to accelerate this convolution, again for the restricted class of radially symmetric BRDF product functions[13]. For the special case of a diffuse BRDF, Ramamoorthi and Hanrahan[16] propose a fast method based on spherical harmonics for computing the convolution and performing real-time rendering.

Spherical Harmonics for Shading. Cabral et al.[5] use spherical harmonics to derive isotropic BRDFs and make the observation that their use reduces the lighting integral to a dot product. We apply this insight to real-time rendering. Our BRDF product coefficients are derived from analytic models[1,15] or measured materials[20] rather than height field geometry. Their method assumes a constant view direction per object; we generalize to arbitrary views by using hardware-supported 2D textures that parameterize BRDF product coefficients in terms of the view direction. Their method is also limited to isotropic BRDFs while we generalize to arbitrary BRDFs. Finally, we combine anisotropic BRDF shading with precomputed transfer to obtain real-time self-shadows and self-interreflections.

Sillion et al.[18] use spherical harmonics to represent exit radiance at many points in an off-line, progressive radiosity simulation. In contrast, our goal is real-time rendering. Their method represents the *view-dependence* of exit radiance in the SH-basis, implicitly assuming this dependence is smooth and so requires few coefficients. As does Cabral, we make the converse assumption that incident lighting is smooth and represent incident lighting and the BRDF product function's *light-dependence* in the SH-basis in order to accelerate the lighting integral which forms the bottleneck in real-time rendering. We tabulate the BRDF product function's view-dependence using high-resolution 2D textures of SH coefficients, providing high-frequency view-dependence which can be significant even in low-frequency lighting.

Westin et al.[21] use spherical harmonics for off-line BRDF inference from geometric models. Both view and light dependence of the BRDF is represented using the SH basis via a large matrix. Encoding view dependence using the SH basis requires on-the-fly evaluation of high-order basis functions which we avoid by using a table.

Precomputed Radiance Transfer. Sloan et al.[19] use the SH basis to represent how an object casts shadows and interreflections onto itself, called *precomputed transfer*. Only radially symmetric BRDF models are considered, precluding anisotropic materials and Fresnel effects. Such restricted BRDFs permit accelerated shading by exploiting the simplicity of convolution in the SH basis[3,16,19].

We generalize precomputed transfer to *arbitrary* BRDFs, for which the convolution property of the SH basis cannot be used. Nevertheless, little extra cost is incurred since it is possible to fold the necessary rotation of lighting coefficients per point on the preprocessed object into the glossy transfer matrix already needed. We just need to do an additional BRDF coefficient vector lookup indexed by the view direction at each point.

Alternatively, the methods of this paper can be used without precomputed transfer, so that no storage or preprocessing is needed for transfer vectors or matrices over an object. In this case, the shading result will lack shadows as in the case of pre-filtering methods, but works with arbitrary BRDFs and dynamic lighting.

Anisotropic lighting models. Analytic models for anisotropic BRDFs have been proposed by Kajiya[10], Poulin and Fournier[15], Banks[2], and recently by Ashikmin and Shirley[1]. Our approach allows any BRDF model as well as measured data[20] in the presence of arbitrary low-frequency lighting.

3. Accelerating Shading with the SH basis

To shade a point p, we need to compute the *lighting integral*, given by

$$Q_p(v) = \int L(s)\, f(s,v) \max(0, s \bullet N_p)\, ds = \int L(s)\, f^*(s,v)\, ds$$

where $Q_p(v)$ is the outgoing radiance for viewing direction v, L is the incident illumination function, f is the BRDF, f^* is the BRDF product function, and s is the incident illumination direction representing the variable of integration. Both s and v are on $\boldsymbol{S}^2$.

We first parameterize f^* by local view direction to get spherical functions, which we represent in the SH basis via

$$f_v^*(s) = \sum_{i=1}^{n} c_i(v)\, y_i(s)$$

where $y_i(s)$ are the SH basis functions. (Formulas for $y_i(s)$ appear in [7,18].) We then tabulate the c_i in terms of the view direction v. For each v, the result is a vector of 25 coefficients representing a spherical function (Figure 2). We use a parabolic parameterization[9] of the hemisphere of directions v to get a 2D texture image where each texel has 25 components, obtaining good results with 128×128 textures.

By representing the lighting function in the SH basis

$$L(s) = \sum_{i=1}^{n} L_i\, y_i(s)\,,$$

the lighting integral then reduces to the simple dot product

$$Q_p(v) = \sum_{i=1}^{n} L_i\, c_i(v)\,. \qquad (1)$$

Inconveniently, f is represented using a local coordinate frame which varies over the object while the incident lighting typically uses a global coordinate system shared across the whole object. Thus to perform the integration, we first need to rotate the lighting into this local frame at

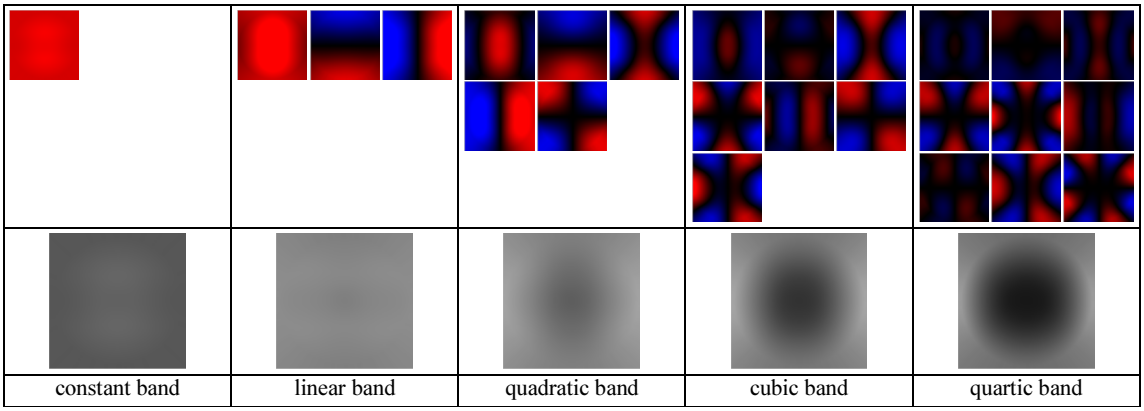


Figure 2: *BRDF product textures for an example Ashikhmin-Shirley model[1]. The top rows of smaller images show component textures for each frequency band, where band i has 2i+1 components (red=positive, blue=negative). The bottom row shows squared energy summed over all the band's components. The parabolic parameterization of a hemisphere is used to map view directions to samples in a square. Note the smoothness of the textures and signal attenuation at higher bands.*

each p. Fortunately, this can be performed with a simple computation (see Appendix).

Calculating BRDF Coefficients. Our approach allows any BRDF model (isotropic or anisotropic) or tabulated data (which we interpolate using radial basis functions to create a smooth BRDF). Because the SH basis is orthonormal, the coefficients g_i of the least-squares best approximation of a function $g(s)$ by a finite series is given by the spherical integral of g times the basis function $y_i(s)$. Thus, as a preprocess we compute with numerical integration

$$c_i(v) = \int y_i(s)\, f(s,v) \max(0, s_z)\, ds$$

where s_z denotes the z coordinate of the direction s, assuming the local coordinate frame maps the normal to the z axis. Computing these coefficients via the above integral is called *SH-projecting* the function, in this case f*.

To vary the BRDF's parameters across the object, we just need a higher-dimensional $c_i(v,q)$ table indexed by whichever parameters q are to be spatially varied.

Calculating Lighting Coefficients. SH-projection is also applied to the incident lighting $L(s)$ to obtain its coefficients L_i. This can be done either as a precomputation if the lighting is fixed, or on-the-fly for dynamic lighting. Hardware renderings can be used to quickly sample incident lighting at many points[19].

We have also implemented arbitrary rotations of fixed environments via on-the-fly SH rotation (Appendix), and circular light sources of parameterized solid angle and direction converted to SH coefficients via a simple formula.

4. Rendering Algorithm

The rendering algorithm is fairly simple; we only need to evaluate equation (1). In the following, we use a primed symbol (e.g., v'_p) for quantities expressed in a global coordinate system for a whole object and an unprimed (e.g., v_p) for the corresponding quantity transformed into the local coordinate frame. At each point p to be shaded, having view vector v'_p and local coordinate frame R_p mapping p's normal N_p to the positive z axis and two chosen tangent vectors to the x and y axes, we:

1. rotate lighting coefficients $L'_{p,i}$ to frame R_p to obtain rotated coefficients $L_{p,i}$ (see Appendix)
2. rotate view vector v'_p to local frame to obtain v_p
3. lookup BRDF coefficients at v_p yielding $c_i(v_p)$
4. compute dot product of $L_{p,i}$ with $c_i(v_p)$

In the simplest case, the global (untransformed) incident lighting coefficients $L'_{p,i}$ can be constant for all points p over an object. This is analogous to using an environment map representing lighting at infinity. For finite lighting (that varies over points p on the object), we can use an incident light *field* which interpolates lighting sampled at various points around the object[19].

Rotation of lighting coefficients (step 1) is too complicated for current graphics hardware, so we currently perform this operation on the host and upload the rotated lighting coefficients to the GPU. The remaining operations (steps 2-4) are performed on the GPU. View vector rotation is done in a vertex shader, the BRDF lookup is just a texture access, and the final dot product is computed in a pixel shader. Current hardware (we use an ATI Radeon 8500) computes the dot product for only a single color channel in one pass due to an instruction count limit in the pixel shader, so three rendering passes are needed.

By fixing either light or view, this rendering algorithm can be accelerated by pre-rotating the SH coefficients.

In the case of fixed lighting, we first rotate the lighting coefficients into the local coordinate system at every vertex p and store the resulting $L_{p,i}$. We can then reuse these coefficients for a new view; i.e., we only need to perform steps 2-4 of the rendering algorithm.

The fixed view case is similar. For a given view, we rotate the view vector v'_p into the local coordinate frame at every vertex p and do a lookup into the BRDF texture with the local view vector v_p to retrieve BRDF coefficients $c_i(v_p)$. These coefficients are then rotated into the global coordinate system and stored per vertex. This is done so that we do not need to rotate the lighting coefficients, which are then the same for every vertex in the case of incident lighting at infinity. After a lighting change, we only need to compute the dot-product between the stored, pre-rotated

BRDF coefficients and the lighting coefficients, performed by a vertex shader. An added advantage is that all 3 color channels can be done in one pass.

Spatially varying BRDFs are handled with the same rendering algorithm, where the BRDF lookup now depends on additional parameters that are varied spatially across the object as well as the view direction.

5. Combining with Precomputed Transfer

Sloan et al.[19] use a linear transformation to represent how an object casts shadows and interreflections onto itself. This linear transformation, called a *transfer matrix*, is stored on many points p over the object and applied to the SH coefficients of the incident lighting. The result of this transformation is a new set of SH coefficients that represent the incident light at the point p but account for self-shadowing and interreflections due to the object, called *transferred radiance*. Transfer matrices are derived using an offline global illumination simulation but can be applied at run-time to arbitrary low-frequency lighting.

We can easily combine precomputed radiance transfer with our technique to obtain self-shadowing and self-scattering effects on anisotropic materials. Instead of using the incident lighting vector $L'_{p,i}$ directly, we apply the transfer matrix to it and substitute the resulting transferred radiance at each p into the rendering algorithm. Steps 1-4 are then performed as before; the BRDF is also tabulated in exactly the same way.

As an optimization, we can fold the required rotation of transferred radiance from the global to local frame into the transfer matrix itself as a preprocess, thus eliminating step 1 but still requiring a linear transformation of the incident lighting at each point. In general, the transfer transformation is less sparse than the rotation transformation.

To include interreflections, the desired BRDF must be fixed and accounted for in the global illumination simulation[19] since interreflections depend on it. If only shadows are considered, then the BRDF can be changed on the fly without additional simulation.

6. Results and Discussion

We achieve interactive frame rates for all models; for the fixed view/fixed light modes we achieve real-time frame rates (Table 1). Performance does not depend on the BRDF model used and is increased only slightly when precomputed radiance transfer is included. Rendering quality can be judged from Figures 1-8, all of which were read back from a PC. Results show monochromatic BRDFs (colors are due solely to colored lighting) but colored BRDFs incur no performance penalty, since we currently compute all color channels separately anyway.

All images except Figure 6 were computed using 25 SH coefficients (fifth order projection) for lighting and BRDF table outputs. Figure 6 varies the SH projection order, showing that a limited number of coefficients support only a limited frequency range in reflections on highly specular surfaces. If both lighting and material BRDF contain higher frequencies, they will be cut off, resulting in blurred reflections. However, even for sharp lighting, accurate results are obtained if the BRDF is not too specular, since more diffuse BRDFs effectively low-pass filter incident lighting[17]. Ringing is also a problem[5,19] which can be mitigated by windowing (i.e., smoothing the lighting) and is masked on bumpy, complex models (compare rows of Figure 8).

Model	#vertices	#fps no transfer	#fps transfer	#fps fix light	#fps fix view
Teapot	152413	2	1.64	9.5	48.9
Head	50060	5.24	4.18	25.3	130
Buddha	49990	4.05	3.22	15.6	127
Bird	48668	6.04	4.74	28.4	128
Tyra	100000	2.62	2.08	9.9	71.5

Table 1: *Timing results. Timings were done on a 2.2Ghz Intel P4 PC with ATI Radeon 8500 card (720x760 image).*

Figure 3 shows the anisotropic AS model applied to a statue head, placed in various lighting environments. Rendering performance is not hampered by dynamic lighting and does not depend on the number or size of light sources; in fact, accuracy increases as lights get bigger.

Figure 1d and 5b illustrate our approach combined with precomputed transfer, which provides self-shadowing. These examples use the Ashikhmin-Shirley (AS) BRDF model[1]. Figure 7 also shows precomputed transfer, focusing on the effect of our generalization to anisotropic BRDFs. All AS model images use a simple anisotropic or "brushing" direction derived from projecting a constant vector to the tangent space at each point.

Figure 4 shows a teapot model with a spatially varying BRDF. We varied a single parameter of the AS model according to a noise function, requiring a 3D texture. The figure's left column shows spatial variation of the glossy power (isotropic model); the right varies the anisotropy. We observe negligible performance decrease when using BRDF spatial variation. In a dynamic lighting environment, such spatial variation would be difficult for pre-filtering methods since a high-dimensional table would have to be built each time the light changed.

Figure 8 compares various BRDF models including anisotropic AS, anisotropic Poulin-Fournier (PF)[15], and measured aluminum foil and vinyl[20]. Note the difference between columns (a) and (b), especially for the teapot (middle row), due to the increased specularity at grazing angles produced by the AS model.

Precomputation times for creating the BRDF table depend on the model used. Analytic models were projected in a few minutes (using 1000 samples for each integration); measured models took up to 90 minutes.

7. Conclusions

We have presented an interactive technique to render objects with general, spatially-varying BRDFs illuminated by dynamic, low-frequency lighting. Combining it with precomputed transfer methods incorporates self-shadowing and self-interreflection effects. Our technique yields fast, very realistic renderings of any material lit by lighting environments containing large light sources, something impossible to achieve with previous techniques.

Our implementation is not yet real-time for the general case where the view and the lighting change simultaneously, since per-vertex rotation of lighting cannot be implemented on current GPUs. We expect next generation hardware to be flexible enough to perform this operation, allowing fully general, real-time performance.

Figure 3: *Brushed metal head in various lighting environments.*

A straightforward extension of this approach would be to decompose lighting into high and low-frequency terms, using our approach for the low-frequency part and existing point light methods[9,11] for the high frequencies.

Acknowledgments

We would like to thank Greg Ward for BRDF data, Paul Debevec for lighting environments, and Hans-Peter Seidel for support.

References

1. ASHIKHMIN, M, AND SHIRLEY, P, An Anisotropic Phong BRDF Model, Journal of Graphics Tools (2000), 5(2):25-32.
2. BANKS, D, Illumination in Diverse Codimensions, SIGGRAPH '94, 327-334.
3. BASRI, R, AND JACOBS, D, Lambertian Reflectance and Linear Subspaces, In International Conference on Computer Vision, 2001.
4. BLINN, J, AND NEWELL, M, Texture and Reflection in Computer Generated Images. Communications of the ACM 19 (1976), 542–546.
5. CABRAL, B, MAX, N, AND SPRINGMEYER, R, Bidirectional Reflection Functions from Surface Bump Maps, SIGGRAPH '87, 273-281.
6. CABRAL, B, OLANO, M, AND NEMEC, P, Reflection Space Image Based Rendering, SIGGRAPH '99, 165-170.
7. EDMONDS, A, Angular Momentum in Quantum Mechanics, Princeton University, Princeton, 1960.
8. GREENE, N, Environment Mapping and Other Applications of World Projections, IEEE CG&A, 6(11):21-29, 1986.
9. HEIDRICH, W, SEIDEL H, Realistic, Hardware-Accelerated Shading and Lighting, SIGGRAPH '99, 171-178.
10. KAJIYA, J, Anisotropic Reflection Models, SIGGRAPH '85, 143-150.
11. KAUTZ, J, AND MCCOOL, M, Interactive Rendering with Arbitrary BRDFs using Separable Approximations, Eurographics Workshop on Rendering 1999, 281-292.
12. KAUTZ, J, AND MCCOOL, M, Approximation of Glossy Reflection with Prefiltered Environment Maps, Proceedings Graphics Interface (May 2000), 119–126.
13. KAUTZ, J, VAZQUEZ, P, HEIDRICH, W, AND SEIDEL, H, A Unified Approach to Pre-filtered Environment Maps, Eurographics Workshop on Rendering 2000, 185-196.
14. MCCOOL, M, ANG, J, AND AHMAD, A, Homomorphic Factorization of BRDFs for High-Performance Rendering, SIGGRAPH '01, 171-178.
15. POULIN, P, AND A. FOURNIER, A Model for Anisotropic Reflection, SIGGRAPH '90, 273–282.
16. RAMAMOORTHI, R, AND HANRAHAN, P, An Efficient Representation for Irradiance Environment Maps, SIGGRAPH '01, 497-500.
17. RAMAMOORTHI, R, AND HANRAHAN, P, A Signal-Processing Framework for Inverse Rendering, SIGGRAPH '01, 117-128.
18. SILLION, F, ARVO, J, WESTIN, S, AND GREENBERG, D, A Global Illumination Solution for General Reflectance Distributions, SIGGRAPH '91, 187-196.
19. SLOAN, P, KAUTZ, J, AND SNYDER, J, Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments, SIGGRAPH '02, to appear.
20. WARD, G, Measuring and Modeling Anisotropic Reflection, SIGGRAPH '92, 265-272.
21. WESTIN, S, ARVO, J, TORRANCE, K, Predicting Reflectance Functions from Complex Surfaces, SIGGRAPH '92, 255-264.

Appendix: SH Rotation

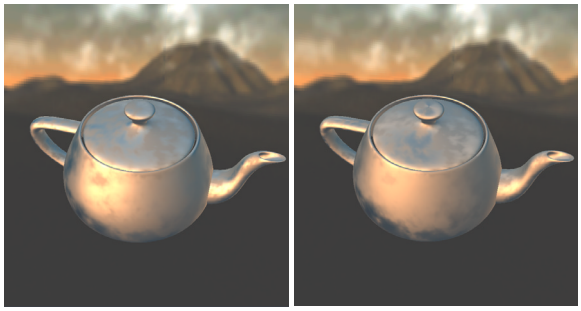
A spherical function represented by a set of SH coefficients can be exactly rotated via a linear transformation of those coefficients[7]. This linear transformation has some additional properties: it is itself a higher-dimensional rotation matrix, and its result for each SH "band" only depends on coefficients from that band (thus implying the transformation is sparse). SH bands have successively 1, 3, 5, …, $2m$-1 coefficients where the total number of coefficients for an order m projection is $n=m^2$.

We have implemented two methods for computing rotations. The first, useful for very low-order projections (up to order 2 or 3), is simply to use symbolic integration to find the linear transformation explicitly as a function of matrix components of the desired rotation R, via

$$M_{ij} = \int y_j(Rs)\, y_i(s)\, ds$$

where the rotated coefficients c_i' are computed from the unrotated c_i via the linear transformation $c_i' = \sum_j M_{ij}\, c_j$.

The second method becomes more efficient as the projection order increases. It first converts the rotation matrix R into its zyz Euler angle decomposition. Rotation around z can be computed using a simple formula that performs a 1D rotation between pairs of coefficients[5,18]. To perform the rotation around y, we further decompose it to a rotation around x by 90°, a general rotation around z, and then a rotation around x by -90°. The two fixed rotations around x are computed using tabulated coefficients derived from numerical integration, and can be represented by very sparse matrices.



(a) varying exponent *(b) varying anisotropy*

Figure 4: *Spatially-Varying BRDFs.*

(a) unshadowed *(b) shadowed*

Figure 5: *Combining with Precomputed Transfer.*

original lighting environment

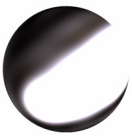
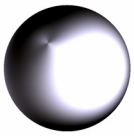
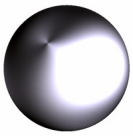
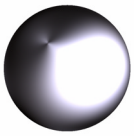
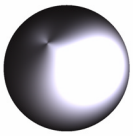
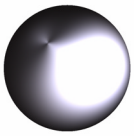
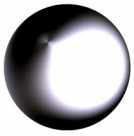
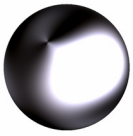
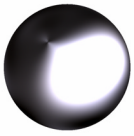
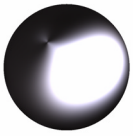
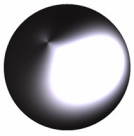
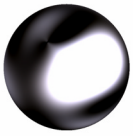
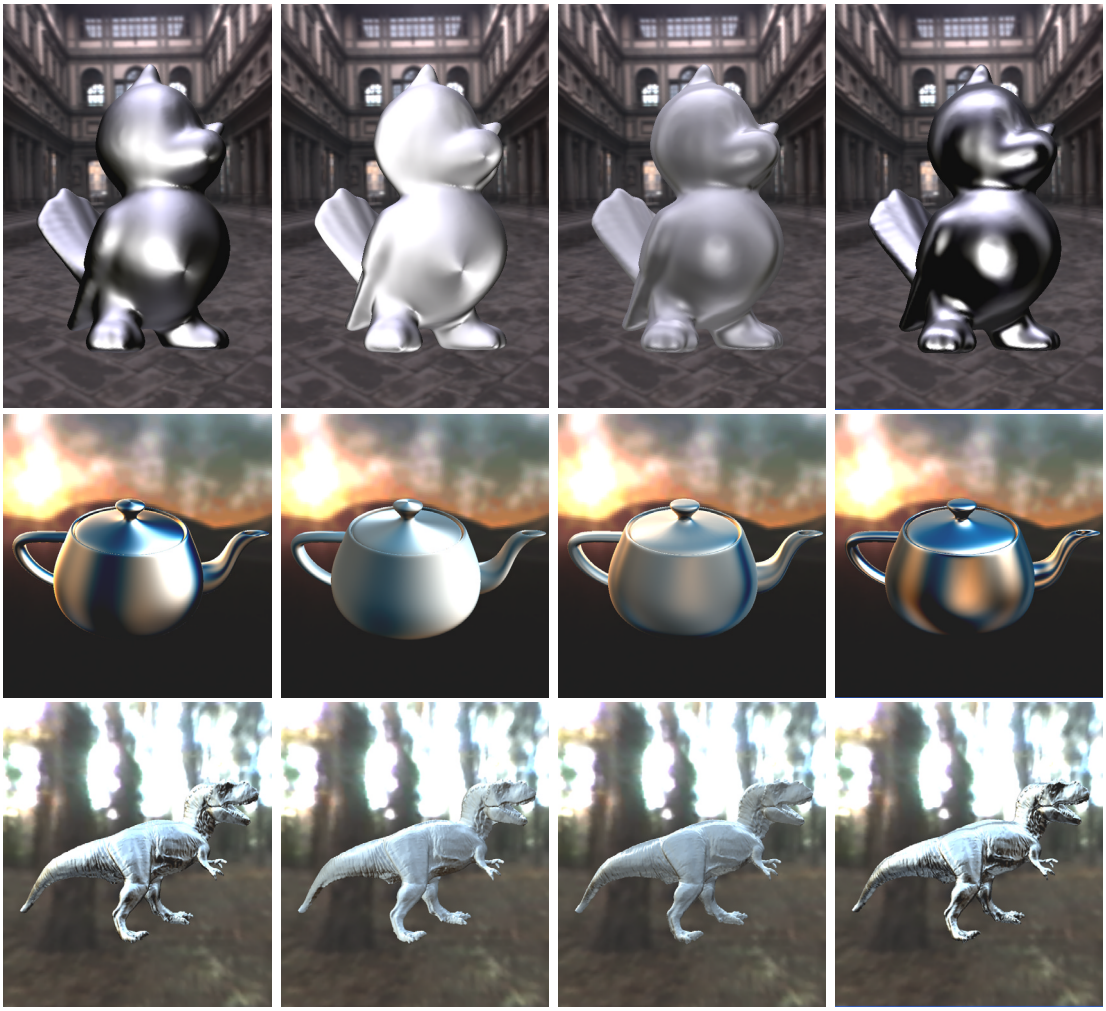
projected lighting environment					
shaded sphere AS (12,4)					
shaded sphere AS (24,8)					
shaded sphere AS (48,16)					
	n=9	*n*=25*	*n*=49	*n*=225,unwindowed	*n*=225, windowed

Figure 6: *Comparison of SH order and glossiness. Columns represent SH projection order for both the lighting environment and BRDF product function: from low (9 coefficients) at left, to high (225 coefficients) at right. Other figures in this paper were computed using 25 coefficients, shown here as the starred column. The last two columns use the same number of SH components, but with and without windowing which reduces ringing artifacts but blurs the lighting. The top row shows the lighting environment; the next three rows then show a shaded sphere lit by that environment using the Ashikmin-Shirley (AS) BRDF model. Various anisotropic glossiness exponents are used: from more diffuse (12,4) at top, to more specular (48,16) at bottom. Low-order projection suffices for less specular BRDFs, even though the lighting environment loses much of its detail.*

(a) isotropic[19] (b) anisotropic (c) anis., different brushed direction

Figure 7: *Generalizing Precomputed Transfer to Anisotropic BRDF Models.*



(a) AS (analytic) (b) PF (analytic) (c) vinyl (measured) (d) alum. foil (measured)

Figure 8: *Comparison of BRDF models applied to various geometric models and lighting environments.*