Local, Deformable Precomputed Radiance Transfer

Peter-Pike Sloan[*] Ben Luna[†] John Snyder[‡]

Abstract

Precomputed radiance transfer (PRT) captures realistic lighting effects from distant, low-frequency environmental lighting but has been limited to static models or precomputed sequences. We focus on PRT for local effects such as bumps, wrinkles, or other detailed features, but extend it to arbitrarily deformable models. Our approach applies zonal harmonics (ZH) which approximate spherical functions as sums of circularly symmetric Legendre polynomials around different axes. By spatially varying both the axes and coefficients of these basis functions, we can fit to spatially varying transfer signals. Compared to the spherical harmonic (SH) basis, the ZH basis yields a more compact approximation. More important, it can be trivially rotated whereas SH rotation is expensive and unsuited for dense per-vertex or per-pixel evaluation. This property allows, for the first time, PRT to be mapped onto deforming models which re-orient the local coordinate frame. We generate ZH transfer models by fitting to PRT signals simulated on meshes or simple parametric models for thin membranes and wrinkles. We show how shading with ZH transfer can be significantly accelerated by specializing to a given lighting environment. Finally, we demonstrate real-time rendering results with soft shadows, inter-reflections, and subsurface scatter on deforming models.

Keywords: lighting environments, nonlinear optimization, spherical harmonics, soft shadows, subsurface scattering, texture maps, zonal harmonics.

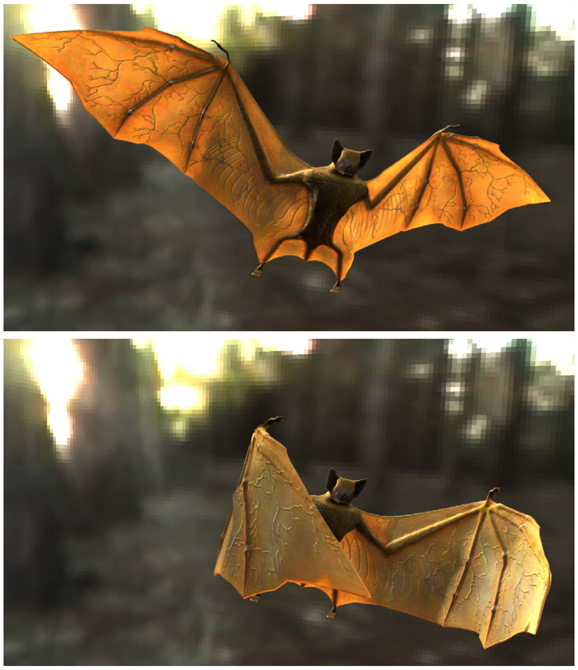


Figure 1: As the bat flaps its wings, LDPRT at any point on the bat can quickly be rotated from the rest coordinate frame where it is stored to the current, deformed orientation. A single-lobe ZH model represents diffuse PRT including subsurface scatter. Real-time rendering rates (240Hz) are achieved on graphics hardware.

1 Introduction

Real-time lighting effects like soft shadows and subsurface scattering can be obtained with precomputed radiance transfer (PRT). Images generated using the technique exhibit a compelling realism lacking from traditional computer graphics with its emphasis on hard shadows and opaque surfaces. Until now PRT has been limited to static objects or highly constrained, precomputed sequences. Our goal is to extend it to dynamic, arbitrarily deformable objects.

We restrict our attention to *local* PRT: how to encapsulate the realistic shading effects mentioned above but from *detailed* surface features, such as wrinkles and bumps commonly represented as texture maps, where the feature's shading influence is *spatially nearby*. Distant effects, such as shadowing from one limb of an articulated figure onto its trunk or another limb, are difficult because the required PRT simulation depends on the figure's pose. When this pose is parameterized by more than one or two degrees of freedom, both the preprocessing and run-time costs become overwhelming. Such effects remain for future work.

Even for local PRT, previous methods are inadequate. We introduce the first method for mapping PRT over a deformable surface which we call *local, deformable PRT* (LDPRT). To understand the difficulty involved, consider a diffuse object, whose self-transfer at some point can be represented as a transfer vector which is dotted with the light vector to produce the diffuse shade at that point. In previous work, transfer and lighting vectors were represented using spherical harmonics (SH), a compact basis for low-frequency lighting. If the object is static, we simply record a PRT vector at each point as a preprocess, assuming the lighting is in a global coordinate frame. If the object rotates rigidly, we can rotate the lighting vector from the rotated into the original (rest) orientation of the object, once for all its points, before computing the per-point shading dot product. But if the object deforms nonrigidly, we must rotate the lighting to the rest frame, on-the-fly at every surface point. SH rotation is costly and unsuited for real-time evaluation on graphics hardware over dense point sets.

To solve this problem, we rotate transfer to the deformed frame instead of rotating lighting to the rest frame. We approximate transfer with *zonal harmonics* (ZH), a general technique that represents spherical functions as a sum of *lobes*: circularly symmetric functions [Whittaker and Watson 1990]. The ZH representation allows on-the-fly rotation, simply by rotating each lobe's axis, and is easily evaluated per-pixel and in real-time on graphics hardware. With

[*] Microsoft Corporation, e-mail: ppsloan@microsoft.com
[†] Microsoft Corporation, e-mail: benluna@microsoft.com
[‡] Microsoft Research, e-mail: johnsny@microsoft.com

a naive method, shading speed is comparable to previous methods using the SH basis, but can be significantly accelerated by a precomputation over the lighting environment.

Our main contribution is a new representation for local, low-frequency, quickly-rotatable radiance transfer. LDPRT has two advantages over previous methods. It allows local PRT effects to be mapped onto any static or deforming surface. And it performs the precomputation over a single mesh or texture patch, not over every pose of a deforming model. We novelly apply sums of zonal harmonics with spatially varying directions and coefficients to allow fast transfer rotation and shading. Our model generalizes previous convolution and "bent normal" methods, providing controllably low approximation error when fit to PRT signals from arbitrary local geometry. Our results show realistic effects such as soft shadows, inter-reflections, and subsurface scattering from textural features, mapped onto dynamic models and synthesized in real-time on graphics hardware (see Figure 1).

2 Previous Work

Precomputed Radiance Transfer (PRT) PRT methods for static objects can be categorized as *low-frequency* [Sloan et al. 2002; Sloan et al. 2003a] and *all-frequency* [Ng et al. 2003; Liu et al. 2004; Wang et al. 2004], according to the lighting and shading frequencies handled. Though more general, all-frequency PRT is problematic because it tabulates much more data per point to represent transfer from higher-dimensional lighting vectors, and many more points per object to represent higher-frequency shading variation over that object. The resulting data sets are enormous, and remain unwieldy even after sophisticated compression techniques are applied. We focus on low-frequency PRT because it is more practical and achieves soft shading effects that are most lacking from existing methods. Rather than tabulating an object's response to high-frequency lighting, it may be more profitable to combine low-frequency PRT methods with existing point or directional light source methods like shadow buffers/volumes.

PRT has also been applied to deformable models [James and Fatahalian 2003] but the preprocessing was essentially done for each pose of the object in a low-dimensional control space: the space of elastic deformations caused by an impulse at an arbitrary instant (i.e., a "poke"). Models can only be poked, they can't be bent or twisted, walk, or flap their wings. Our aim is to capture local effects for models where the deformations are not known in advance and so can have arbitrarily many degrees of freedom.

Bi-scale radiance transfer [Sloan et al. 2003b] and environnmental lighting with BTFs [Muller et al. 2004] capture both local and distant transfer effects on static objects. The distant level, represented as transfer matrices, is required not only to obtain distant effects (which we neglect here) but also to rotate the lighting vector to the local frame. If the object deforms, this rotation is impractical to tabulate and so must be performed dynamically at each surface point, a problem we address.

Local Parallax Effects Bi-scale radiance transfer and BTFs also capture view-dependent parallax effects at a local scale using 4D textures. Other methods for obtaining parallax effects (also called visual masking) include parallax mapping [Kaneko et al. 2001], relief texture mapping [Oliveira et al. 2000], view-dependent displacements mapping [Wang et al. 2003], and volume textures [Meyer and Neyret 1998]. LDPRT captures only shading effects, not parallax, but can easily be combined with such methods.

Unshadowed Irradiance Irradiance maps [Ramamoorthi and Hanrahan 2001] are a related idea, in which samples are taken from a pre-convolved lighting environment represented using SH. This is similar to our single lobe ZH approximation, except that we do the convolution on the fly with spatially-varying coefficients. By summing multiple lobes, we fit to precomputed transfer signals which in general are not circularly symmetric. This allows spatially-varying transfer effects like shadowing and scattering from arbitrary local geometry.

Transfer Methods for Directional Lighting Precomputed visibility [Heidrich et al. 2000], polynomial texture maps [Malzbender et al. 2001], horizon maps [Max 1988], and steerable illumination textures [Ashikhmin and Shirley 2002] are pretabulated methods for shading locally-varying geometry. Constant [Westin et al. 1992] and spatially-varying [McCallister et al. 2002] BRDF models can also be fit to local geometry. Bidirectional texture functions (BTFs) [Dana et al. 1999] are similar to spatially-varying BRDFs but are designed for meso-scale rather than micro-scale surface properties by capturing the surface's shading response as a function of spatial position, view, and lighting direction. These methods are all based on the directional lighting basis rather than environmental lighting and so require integration over many lighting directions at run-time when used with area light sources or environmental lighting. By fitting to transfer parameterized by the SH lighting basis, we avoid impractical light integration.

Transfer Methods for Environmental/Area Lighting [McCallister et al. 2002] also deals with environmental lighting by combining an irradiance map with a specular table that preconvolves lighting as a function of specular peak direction and power falloff. This model "leaks" lighting below the hemisphere and doesn't properly handle cosine modulation. It also uses densely-sampled 3D tables for lighting; we use low-resolution 2D tables. Most important, its basis functions simply decay away from a central axis. Our zonal harmonics are a general basis for low-frequency, circularly symmetric functions, exhibiting complex falloff and ringing (see Figure 2). They provide a more powerful fit to transfer where nearby geometry modulates or blocks the incoming light in a fairly arbitrary way. Their model is thus more appropriate for highly specular surfaces and shading from extremely fine micro-geometry, while we target shadows and other transfer effects from diffuse or semi-glossy meso-geometry.

Hemispherical rasterization [Kautz et al. 2004] captures distant shadowing on dynamic geometry by rasterizing blockers at each receiving vertex. It achieves interactive performance (≤15Hz) at 1-2 thousand vertices. We capture only local effects but with transfer shading per pixel and at 10x higher frame rates. The techique could be combined with ours by applying a coarsely-sampled radiance field to our local transfer model.

Accessibility shading [Miller 1994] and ambient occlusion [Fernando 2004] can be viewed as a simple form of PRT restricted to the DC (average) lighting component. By including higher-order lighting frequencies, we obtain more realistic shading variation; for example, shadows trail peak lighting directions. The "bent normal" [Fernando 2004] is an ad-hoc model in which the diffuse shading normal is replaced by the weighted average visibility vector in the shading dot product; the result is then scaled by ambient occlusion. This is spatially-varying direction manipulation to capture transfer effects and so represents a precursor to LDPRT. We fit to actual transfer signals with a general model (zonal harmonics) that drives error to 0 using multiple lobes with spatially-varying lobe coefficients as well as directions. We also handle environmental lighting rather than directional and general transfer effects like inter-reflections and subsurface scatter.

Curvature Change Precomputed visibility [Heidrich et al. 2000] is also notable in that it handles small curvature changes to smooth underlying geometry under the assumption that such changes preserve local visibility. View-dependent displacement maps [Wang et al. 2003] handle larger curvature changes by explicitly tabulating rendering parameters as a function of curvature. Our method neglects this second-order deformation effect, which is typically subtle compared to the first-order effect of re-orienting the local frame.

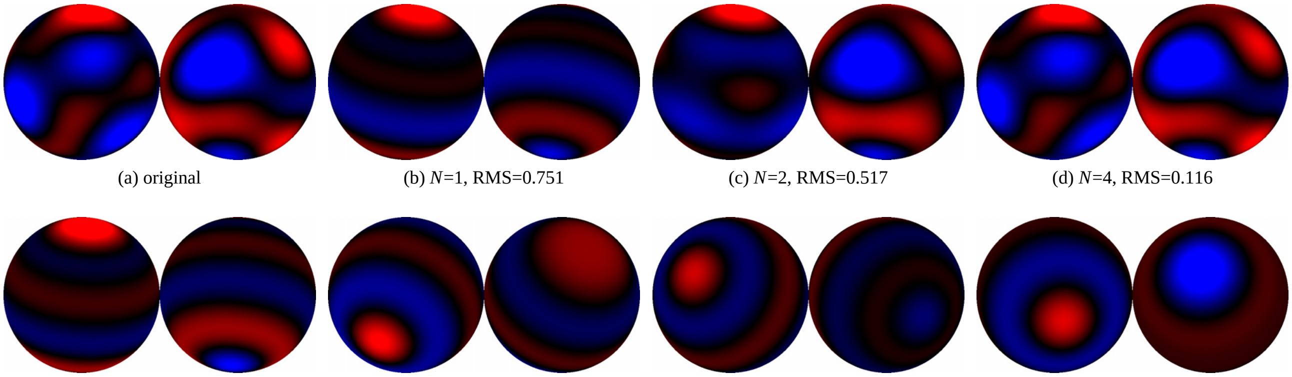


Figure 2: Sums of zonal harmonics accurately match difficult signals. A unit-length SH vector (n=6) was randomly generated as the original signal (a) with positive values colored red and negative blue. ZH approximations (b,c,d) with increasing number of lobes N are shown on the right. For N=6 (not shown), RMS error further decreases to 0.00908, or less than 1% of the original signal's L^2 norm. The bottom row of the table shows the 4 lobes whose sum yields the N=4 approximation in (d) of the top row.

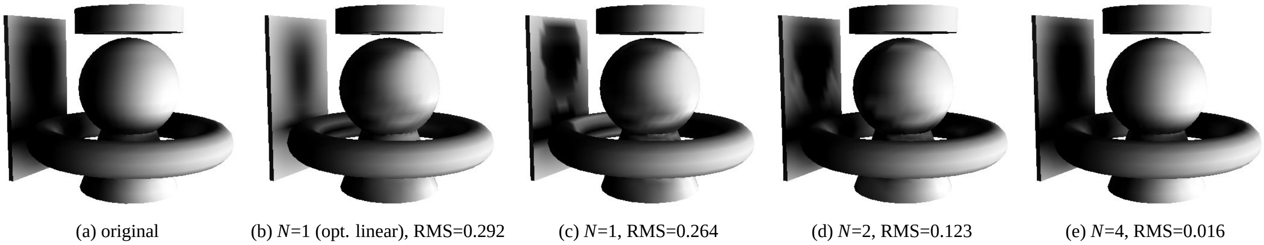


Figure 3: LDPRT on a diffuse example, n=6. RMS error is normalized by total L^2 signal length over all mesh vertices. For N=3 (not shown), RMS error is 0.045.

3 Zonal Harmonics for PRT

3.1 ZH Representation for Spherical Functions

Spherical harmonics represent a scalar spherical function $f(s)$, $s = (s_x, s_y, s_z) \in S^2$ via

$$f(s) = \sum_{l=0}^{n-1} \sum_{m=-l}^{+l} f_{lm}\, y_{lm}(s) = f \cdot y(s) \qquad (1)$$

where n is the *order* of the SH expansion, $y(s)$ is the vector of SH basis functions, and f is the vector of coefficients for $f(s)$. Both vectors have n^2 components. One property of the SH representation is that arbitrary rotations of $f(s)$ can be represented exactly using independent rotations of the coefficients f_{lm} in each *band*, which is indexed by a constant value of l and has $2l+1$ components.

A function $g(s)$ which is circularly symmetric about the z axis can be represented more simply as

$$g(s) = g(s_z) = \sum_{l=0}^{n-1} g_l\, y_{l0}(s_z) \qquad (2)$$

where s_z is the z component of the spherical point s, and the basis functions $y_{l0}(s)$ are a subset of the SH basis functions (m=0) given by Legendre polynomials and dependent only on s_z. Such functions are also called *zonal harmonics* because their zeroes form parallels (lines of constant latitude), which divide the sphere into zones. Note that s_z is just the cosine of the angle between s and the z axis and is often written $\cos\theta$ when spherical functions are parameterized by the angular variables (θ, ϕ).

Spherical convolution of the functions f and g is then given by [Driscoll and Healy 1994]

$$f \star g(s) = \sum_{lm} y_{lm}(s) \left(\sqrt{\frac{4\pi}{2l+1}}\, g_l \right) f_{lm} = \sum_{lm} y_{lm}(s)\, g_l^*\, f_{lm} \qquad (3)$$

where

$$g_l^* = \sqrt{\frac{4\pi}{2l+1}}\, g_l. \qquad (4)$$

In matrix/vector notation, we write this as $f \star g(s) = y^T(s)\, G^*\, f$ where y^T is the transpose of y, and G^* is a diagonal matrix composed of the convolution coefficients g_l^*, each repeated $2l+1$ times in successive bands l=0, l=1, etc. For example, for n=3 the diagonal of G^* is the 9D vector $(g_0^*,\, g_1^*, g_1^*, g_1^*,\, g_2^*, g_2^*, g_2^*, g_2^*, g_2^*)$.

The convolution property allows us to rotate any circularly symmetric function and find its resulting SH coefficients. Define the function $\hat{g}(s) = g(R^{-1}s)$ where the rotation R maps the z axis to an arbitrary axis s^*. To find the coefficients of $\hat{g}$, we have

$$\int_{S^2} f(s)\, \hat{g}(s)\, ds = f \star g(s^*) = \sum_{lm} f_{lm}\, \hat{g}_{lm} \qquad (5)$$

where the last equality is true by the orthogonality of the SH basis. Since this is true for any function f, substituting the definition of $f \star g$, we obtain

$$\hat{g}_{lm} = y_{lm}(s^*)\, g_l^* \qquad (6)$$

yielding the rotated coefficients, where g_l^* is defined in equation (4).

Equation (6) tells us how to rotate a ZH function (circularly symmetric about the z axis) so that it becomes symmetric about an arbitrary axis s^*. We now approximate a spherical function $f(s)$ as the sum of N rotated ZH functions about arbitrary axes, indexed by i. Therefore by (6), a component of f is approximated by $\tilde{f}$ via

$$f_{lm} \approx \tilde{f}_{lm} = \sum_{i=1}^{N} y_{lm}(s_i^*)\, g_{il}^* \tag{7}$$

We write this in matrix/vector notation as

$$f \approx \tilde{f} = \sum_{i=1}^{N} y^T(s_i^*)\, G_i^* \tag{8}$$

The continuous reconstruction of $f(s)$ is then approximated by

$$f(s) \approx \tilde{f}(s) = \sum_{i=1}^{N} y^T(s_i^*)\, G_i^*\, y(s) \tag{9}$$

Here, the s_i^* are called the *lobe axes* and the set of n coefficients g_i^* per lobe are called the *lobe coefficients*. When $N = 1$, we call the ZH approximation *single-lobe*, and *multi-lobe* if $N > 1$.

The advantage of this representation is its simplicity of spherical rotation. Define a rotated version of $f(s)$ via $\hat{f}(s) = f(R^{-1}s)$ where R is a 3D rotation matrix. Given the vector f of SH coefficients, this rotation can be performed by applying a $(2l+1)\times(2l+1)$ rotation matrix in each band, an operation that is $O(n^3)$ in the SH order n and computationally very costly (see digression below).

Using the ZH representation however, the operation is $O(N)$ where typically $N \leq 4$ and trivial to evaluate:

$$\hat{f} \approx \sum_{i=1}^{N} y^T(R s_i^*)\, G_i^* \tag{10}$$

In other words, the rotated ZH representation rotates the lobe axes to $R s_i^*$ while the lobe coefficients g_i^* and thus the diagonal matrices G_i^* remain unchanged (compare to equation (8)).

Each ZH lobe is axially symmetric. Negating the lobe direction and all its odd l lobe coefficients results in an identical function.

3.2 Applying ZH to PRT

Diffuse PRT Applying these ideas to to PRT, assume that a low-frequency lighting environment is represented in the SH basis as a vector e. For a diffuse surface, the transfer vector can be represented by another SH vector t_p [Sloan et al. 2002] which varies over points on the surface p. The shading result is then given by $r_p = e \cdot t_p$, which is simply the spherical integral of source radiance $e(s)$ times the object's *transfer*, $t_p(s)$, which encapsulates how the object shadows and scatters light onto itself at point p. In this work we use an order 6 SH expansion, so that these vectors are 36D. For simplicity in the notation, we will drop the subscript p in the following though transfer remains a spatially-varying quantity.

Now instead of recording transfer vectors in the SH basis, t, we record N lobe axes s_i^*, and for each axis its n coefficients g_i^* which determine a diagonal matrix G_i^*. The transfer vector is approximated using equations (7,8) as

$$t \approx \tilde{t} = \sum_{i=1}^{N} y^T(s_i^*)\, G_i^*, \qquad \tilde{t}_{lm} = \sum_{i=1}^{N} y_{lm}(s_i^*)\, g_{il}^* \tag{11}$$

A continuous spherical function $\tilde{t}(s)$ can be reconstructed from the vector $\tilde{t}$ via equation (9) as

$$\tilde{t}(s) = \sum_{i=1}^{N} y^T(s_i^*)\, G_i^*\, y(s) \tag{12}$$

and the shading result is then given by $r = \tilde{t} \cdot e$ or

$$r = \sum_{i=1}^{N} y^T(s_i^*)\, G_i^*\, e = \sum_{i=1}^{N} \sum_{lm} y_{lm}(s_i^*)\, g_{il}^*\, e_{lm} \tag{13}$$

The ZH transfer model therefore makes shading evaluation simple and, what's more, allows the transfer vector to be rotated easily to any frame using (10). Note that the N lobe axes and coefficients vary over the surface as did the original PRT vector t.

Glossy PRT For glossy objects, exit radiance is view-dependent rather than constant. The linearity of light transport means that exit radiance can be represented as a linear operator (matrix) on the source lighting vector e, given by $r(v) = y^T(v)\, M\, e$ where M is an $n^2 \times n^2$ transfer matrix and v is the view direction [Sloan et al. 2003a]. M takes source lighting e in the SH basis and produces an exit radiance spherical function, also in the SH basis.

We can reduce the number of rows in M by specializing the output basis to the object's reflectance properties using a view/light factoring of its BRDF [Liu et al. 2004; Wang et al. 2004]. If $b(v,s)$ is the BRDF, parameterized by view direction v and light direction s, the idea is to sample it over many view and light directions and perform a singular value decomposition (SVD) over the resulting matrix of samples. Using the SH basis for lighting, define a $n_v \times n^2$ matrix B where n_v is the number of view samples, given by the following integral over the hemisphere $H = \{s | s_z \geq 0\}$:

$$B_{ij} = \int_H b(v_i, s)\, y_j(s)\, s_z\, ds = (U\,\Sigma\,V)_{ij} \approx (\tilde{U}\,\tilde{\Sigma}\,\tilde{V})_{ij} \tag{14}$$

$U\Sigma V$ is the SVD of B. Selecting only the n_b largest singular values yields the approximation $\tilde{U}\tilde{\Sigma}\tilde{V}$, where $\tilde{U}$ is $n_v \times n_b$, $\tilde{\Sigma}$ is a $n_b \times n_b$ diagonal matrix, and $\tilde{V}$ is $n_b \times n^2$.

Given B, glossy shading may be expressed as $r(v) = \gamma^T(v)\, B\, T\, e$ where T is the transfer matrix going from source to transferred incident radiance [Kautz et al. 2002]. $\gamma(v)$ is an n_v-dimensional vector that simply interpolates the correct components of exit radiance for a specified view direction v, and depends on the view parameterization. Using the SVD, shading is approximated by

$$r(v) = \left(\gamma^T(v)\,\tilde{U}\right)\left(\tilde{\Sigma}\,\tilde{V}\,T\right) e = u(v)\, M'\, e \tag{15}$$

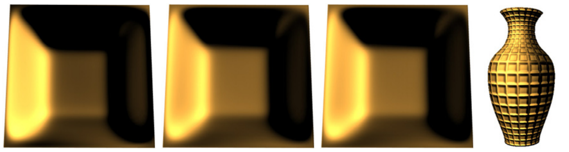
M' is a $n_b \times n^2$ *BRDF-specialized transfer matrix* and $u(v) = \gamma^T(v)\,\tilde{U}$ is an n_b-dimensional *view factor vector* generated by the SVD. The approximation reduces the number of rows in the transfer matrix from n^2 (M) to n_b (M').

Now a ZH model can be fit to each of M''s n_b rows, just as we did for the diffuse transfer vector t. In other words, a diffuse transfer vector is just a single-row transfer matrix. The result is a quickly-rotatable representation for glossy transfer that can be applied to deformable models. The final shade is given by evaluating each of the n_b scalars using equation (13) for each row of M', and then dotting that result with the n_b-dimensional vector $u(v)$, or

$$r(v) = \sum_{j=1}^{n_b} u_j(v) \sum_{i=1}^{N} \sum_{lm} y_{lm}(s_{ji}^*)\, g_{jil}^*\, e_{lm} \tag{16}$$

where j varies over M' rows. $u(v)$ is a 32×32 texture, which composes a mapping from square to disk [Shirley and Chiu 1997] with an area-preserving mapping from disk to hemisphere.

Rotating Transfer vs. Light Another possible solution to the problem posed by deformable models may occur to the reader. It is to fit a ZH model to the lighting e rather than the transfer vector t or matrix M'. Instead of rotating transfer into the deformed coordinate



(a) original, n=4 (b) SVD, n_b=3 RMS=0.33 (c) ZH, N=3 RMS=0.0074 (d) mapped

Figure 4: Glossy LDPRT. RMS error in (b) is due to the BRDF SVD with n_b=3 rows; error in (c) is the *additional* error due to the ZH fit.

frame, it is also possible to rotate lighting into the rest coordinate frame. The advantage is that a ZH fit is no longer needed over each point on the object or texture patch. The rotational simplicity of the ZH model is similarly leveraged.

The primary reason to rotate transfer rather than lighting is to admit the light-specialized optimization described in Section 5, which signicantly accelerates rendering. This optimization relies on the fact that a single lighting environment simultaneously applies to multiple shaded points and so can't be similarly applied to spatially-varying transfer. The fact that lighting is spectral,[1] is high dynamic range rather than clamped between 0 and 1, and is not modulated by a cosine or BRDF means that more lobes will be needed to approximate it relative to transfer, slowing the rendering even further. In addition, we consider dynamic lighting to be a critical application. The cost of nonlinear optimization to fit the ZH model will be prohibitive if done per-frame, while the cost of building the light specialized table per-frame is quite small and is included in our peformance results (Section 6).

3.3 ZH Error Analysis

A constant (DC) signal (n=1, l=0) is represented by a single coefficient, so we store it explicitly and fit ZH lobes to the remaining components, l>0. A linear SH function (l=1) can be exactly reproduced using any three linearly independent vectors as lobe directions. Therefore, if we're using three or more lobes, we store the linear SH components directly to reduce storage.

A linear SH function is also circularly symmetric about an axis, because the basis functions are linear polynomials in s_x, s_y, and s_z. That axis of symmetry for a linear SH vector f is given by normalizing the 3D vector of its linear basis coefficients $(-f_{11}, -f_{1-1}, f_{10})$. We call this the *optimal linear lobe* for f. Higher order SH expansions however can not be represented exactly as a finite sum of rotated Legendre functions of the same order. Fortunately, error approaches 0 rapidly as N increases.

Figure 2 shows ZH fit results for a difficult, randomly-generated SH function. For diffuse PRT, the ZH model fits even better as shown in Figure 3. This example uses sixth order SH lighting (n=6). For single-lobe models (N=1), using the optimal linear lobe results in higher error than fitting optimally to all 36 components of the transfer vector, as would be expected. The optimal linear approximation is spatially smoother though. A four lobe approximation matches the original signal well, both visually and in terms of error measurement.

Figure 4(a) shows a glossy model with fourth-order lighting (n=4), using [Cook and Torrance 1982] with glossiness=0.3. BRDF-specialized transfer (b) provides a good approximation with only n_b=3 rows instead of n^2=16. ZH with N=3 lobes per row (c) then provides an accurate fit with $n_b N$=9 total lobes.

3.4 Digression on SH Rotation

General SH rotation can be computed most efficiently by converting to *zyz* Euler angles. This is because rotation around the z axis by an angle θ of a component band l has a special form that can be performed by l-1 2D rotations of the $\pm m, |m| > 0$ components by an angle $m\theta$. Rotation around z is therefore $O(n^2)$.

Unfortunately, the required y rotation is more difficult. A simple method to rotate around y by an angle ϕ is to further decompose it into a series of three rotations: a rotation of $+90^\circ$ around x, followed by a z rotation by ϕ, followed by a rotation of -90° around x. These two x rotations are inverses and thus tranposes because they are themselves high-dimensional rotation matrices. It also turns out that these x rotations are sparse: asympotically only 25% of the $(2l+1)\times(2l+1)$ rotation matrix components are nonzero as $l \rightarrow \infty$. Exploiting this sparsity reduces computation, but it is only a constant factor reduction, so x rotation remains $O(n^3)$. Note that the sparsity of $\pm 90^\circ$ rotation around x means that the method asymptotically requires a number of multiplies equal to 50% of the number of components in the SH rotation matrix, and thus compares favorably to methods for instantiating that matrix through recursive rules [Choi et al. 1999].

We include a table of total multiplication counts for rotating various SH orders using the *zxzxz* method:[2]

$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$
53	121	217	345	509	713	961

SH rotation is very costly for the sixth order lighting (n=6) we use and rapidly gets worse at higher order. Three such rotations must be computed for colored lighting, amounting to more than a thousand multiplies. This should be compared to $N \leq 4$ 3×3 rotations of a lobe axis from equation (10) for diffuse transfer.

4 Creating ZH Transfer Models

4.1 Tabulated Models

Using nonlinear optimization, the ZH model may be fit to an arbitrary spherical functions expressed in the SH basis. The objective function ψ to be minimized is simply the L^2 norm of the vector difference of the ZH approximation $\tilde{t}$ from equation (11) with the original transfer vector t being approximated, or

$$\psi = \|\tilde{t} - t\|^2 = \int_{S^2} \left(\tilde{t}(s) - t(s)\right)^2 \, ds \qquad (17)$$

By the orthogonality of the SH basis, this vector distance equals the squared difference between the continuous functions $\tilde{t}(s)$ and $t(s)$ integrated over the sphere.

Note that the ZH transfer model must be fit over many points on an object or in a patch of local texture, so that this optimization is applied many times. Fitting glossy PRT is similar, except that it is done for each row in M' from equation (15).

Single-Lobe Fitting For a single-lobe ZH model, if the lobe axis s^* is fixed, then the lobe coefficients that best approximate a given

[1] Colored lighting requires a separate ZH model per color channel while transfer can be approximated by a single-channel ZH model modulated by a simple RGB texture map. A single-channel transfer approximation is exact for shadowing and accurate for other effects assuming spatial color variation is low.

[2] The table assumes the linear band l=1 is rotated using the more efficient method of a 3×3 rotation matrix. The constant band l=0 is not changed by rotation. Conversion of a rotation matrix to Euler angles is not included here, but requires 5 multiplies, a reciprocal, and a square root.

SH vector t are given in closed form as

$$g_l^* = \frac{\sum_{m=-l}^{+l} y_{lm}(s^*)\, t_{lm}}{\sum_{m=-1}^{+l} y_{lm}^2(s^*)} \tag{18}$$

This formula simply computes the least-squares best match of the single lobe coefficient g_l^* to the $2l+1$ components of the basis function evaluation $y(s^*)$ in each band l. The denominator above is a constant, $\sum_m y_{lm}^2(s) = (2l+1)/(4\pi)$, so we precompute a table of reciprocals of this sum of squares for each band index l.

We fit a single-lobe model to a given SH vector t by looping over a set of directions formed by uniformly sampling a hemi-cube. A hemi-cube suffices because opposite directions produce the same ZH model (see Section 3.1). We also add the optimal direction for just the linear part of the vector (Section 3.3). For each direction used as a lobe axis, we compute the lobe coefficients using (18). We then run BFGS minimization [Press et al. 1992] using this result as a starting point and both the lobe axis and lobe coefficients as minimization variables. The optimized direction yielding the smallest error (17) is the final result.

Optimization Notes Since BFGS is a derivative-based method, we include the derivatives of the objective function (17) with respect to the minimization variables:

$$\frac{\partial \psi}{\partial g_l^*} = 2\sum_m y_{lm}(s^*)\,(\tilde{t}_{lm} - t_{lm}) \tag{19}$$

$$\frac{\partial \psi}{\partial s^*} = 2\sum_{lm} \nabla y_{lm}(s^*)\, g_l^*\,(\tilde{t}_{lm} - t_{lm}) \tag{20}$$

The gradients of the SH basis functions y_{lm} with respect to the spherical point s^* were computed using Maple but can also be derived using the recurrence relations for SH evaluation.

To constrain 3D points s^* to lie on the sphere, we substitute $h(s^*) = s^*/\|s^*\|$ for s^* in the above equations. The gradient of $y_{lm} \circ h$ replaces the gradient of y_{lm} in (20), given by

$$\nabla' y_{lm} = J_h^T\, \nabla y_{lm} \tag{21}$$

where J_h is the 3×3 Jacobian of the normalization function h.

Multi-Lobe Fitting To fit a multi-lobe model, we iteratively apply the above algorithm. We fit a single lobe, subtract its reconstruction (equation 11) from the vector being approximated, and then fit the next lobe to the residual, up to N lobes. A local minimum can then be reached by applying the BFGS algorithm to all lobe parameters simultaneously. To speed things up, we first freeze lobe directions and optimize only the lobe coefficients to get closer to a local minimum before applying BFGS over all lobe parameters.

Not surprisingly, additional searching often finds a better minimum. We apply further computation when the fit error is high, by trying randomly-generated sets of N lobe directions and then locally optimizing with BFGS.

Solving the multi-lobe, non-linear optimization problem was simpler than we expected. We believe our problem is easier than similar model-fitting problems in computer graphics, such as the multi-lobe Lafortune model [Lafortune et al. 1997], for several reasons:

1. Our objective function is very smooth since the transfer vectors to be fit are low-frequency.

2. The Lafortune model has the undesirable property that several degrees of freedom are highly coupled, notably the matrix diagonals and powers. In our problem, the ZH coefficients of different bands have no influence on each other.

3. For single lobe fitting, we have a good initial guess (optimal linear lobe). Furthermore, if we freeze the lobe axis, we have a closed form solution for the ZH coefficients. For multi-lobe fitting, the simple iterative algorithm (approximate a single lobe, compute residual, repeat) finds reasonably low-error results, which can be improved by additional "multi-start" searching.

4. The DC/linear part of the signal is reproduced with zero error given enough lobes (1 for DC, 3 for linear).

5. Previous work in fitting Lafortune lobes has used Levenberg-Marquardt optimization (see e.g. [Lensch et al. 2001]). Unlike BFGS, this works poorly if the local minimum has large error, since the Hessian approximation is then inexact [Nocedal and Wright 1999].

4.2 Parametric Models

Translucent Thin-Membrane Model A simple model for diffuse translucency is shown in Figure 1. It uses a cosine weighting function centered around the outward-pointing surface normal direction and an attenuated cosine weighting in the opposite direction, given by the spherical function

$$g(s_\zeta) = \begin{cases} s_\zeta, & \text{if } s_\zeta \geq 0 \\ -\alpha s_\zeta, & \text{otherwise} \end{cases} \tag{22}$$

where $\alpha \leq 1$ is the transparency, and $s_\zeta = s \cdot \zeta$ is the cosine of the angle s makes with the surface normal ζ.

Since this is a circularly symmetric function around ζ, it can be represented using a single-lobe ZH, where the lobe axis is the surface normal ζ and the lobe coefficients are given by

$$g_l^* = \begin{cases} 2/3\,(1-\alpha), & \text{if } l = 1 \\ 2\,(1+\alpha)\left(\frac{(-1)^{l/2-1}}{(l+2)(l-1)}\right)\left(\frac{l!}{2^l((l/2)!)^2}\right), & \text{if } l \text{ even} \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

This is the sum of two cosine-weighting functions, given by equation (8) in [Ramamoorthi and Hanrahan 2001]. One just represents the response of a diffuse surface to low-frequency lighting. The other is scaled by transparency and oriented opposite to the normal direction to account for translucency. These coefficients are also divided by π to convert irradiance to exit radiance.

A spatially-varying translucency map (α) can be applied to enhance realism, as in the bat example. We also note that a subsurface scattering simulation can be performed to parameterize a more sophisticated model based on membrane thickness and scattering properties rather than simple transparency.

Wrinkle Model We model a wrinkle as a simple sinusiodal groove, parameterized by location along the groove cross-section, x, and groove depth d. We extrude this groove and perform a diffuse PRT simulation on the resulting parameterized height field. We then fit a 3-lobe ZH model to its diffuse transfer signal. Our fit uses bilinear basis functions which produce parameters for 3 lobes (axis and coefficients) as a 32×32 map of (x,d). DC and linear SH coefficients are explicitly stored, and fit with bilinear basis functions.

In performing the ZH fit, it is important to establish a correspondence between lobes. Independently fitting a ZH model to each (x,d) point does not match lobes at neighboring (x,d) and so causes artifacts when they are interpolated. We instead perform a single optimization over the entire (x,d) space. Using bilinear basis functions in (x,d) for each lobe parameter and fitting to a more highly-sampled version of the PRT signal automatically penalizes lack of correspondence between lobes. An initial guess for the optimizer is computed by a linear least squares fit of an optimal linear lobe (Section 3.3) with two additional lobes placed symmetrically around it, by a locally optimized angle. The final optimization then uses the conjugate gradient method [Press et al. 1992]. In our experiments, this worked better than BFGS, whose approximate inverse Hessian became too large when dealing with the much greater number of parameters involved.

5 LDPRT Rendering

LDPRT rendering begins by projecting the lighting into the SH basis to get the vector e. Then for each vertex p on the deforming object, we do the following:

1. compute the rotation R_p from the rest coordinate frame into deformed frame,
2. apply R_p to the N ZH axes ($n_b N$ axes for glossy) at p: $R_p s_i^*$,
3. evaluate equation (13) (equation (16) for glossy).

The local deformed frame R_p can be computed on the CPU or in a vertex shader and is then interpolated using rasterization hardware. For example, these frames were computed using skinning in the bat example in Figure 1. The third step is done in a pixel shader.

Evaluating equation (13) can be sped up by storing the SH basis functions $y(s)$ in spherical tables. This is useful when we wish to apply *local* lighting, which varies over points on the object. For distant lighting shared over many surface points, there's an even better speed-up described below.

Lighting-Specialized LDPRT To accelerate shading in (13), we can specialize ZH transfer evaluation to a particular lighting environment e. A spherical table is created, $\hat{e}(s)$ from points $s \in S^2$ into R^n where n is the SH order, via

$$\hat{e}_l(s) = \sum_{m=-l}^{+l} y_{lm}(s)\, e_{lm} \tag{24}$$

Now equation (13) simplifies to the following simple computation

$$r = \sum_{i=1}^{N} \hat{e}(s_i^*) \cdot g_i^* = \sum_{i=1}^{N} \sum_{l=0}^{n-1} \hat{e}_l(s_i^*)\, g_{il}^* \tag{25}$$

requiring just N look-ups of $\hat{e}$ at the lobe axes s_i^* and N n-dimensional dot products between this result and the lobe coefficients g_i^*. Computation in (13) is reduced from $O(n^2 N)$ to $O(nN)$. Tabulated vectors are reduced from n^2-dimensional ($y(s)$) to n-dimensional ($\hat{e}(s)$). For small N, the computation is even simpler than the n^2-dimensional dot products required by the SH basis. The reduction is easily derived from (13) by grouping the e_{lm} and $y_{lm}(s_i^*)$ factors and performing the inner sum over m to generate $\hat{e}_l$. We use a 64×32 map for $\hat{e}(s)$, parameterized by latitude/longitude.

6 Results

"Unshadowed" results (Figures 5ac, 6afh) integrate over a lighting hemisphere, but without blocking or scattering the source radiance. They are equivalent to rendering with spatially-varying normals which index an irradiance map [Ramamoorthi and Hanrahan 2001]. By including transfer effects in the precomputed simulation, LDPRT can incorporate shadowing for both diffuse (Figures 5bd, 6bgi, 7abf) and glossy surfaces (Figure 7d), diffuse inter-reflection (Figure 6e), and subsurface scattering (Figures 1, 5e, 7c). The subsurface waffle texture (Figures 5e, 7c) simulates geometry whose thin bottom layer transmits more light than its sides. It was derived from an actual scattering simulation, not the simple model of Section 4.2. Figure 5a shows the sharpest shadow we can cast with sixth-order SH lighting. All textures can be mapped onto any surface, no matter how it deforms.

LDPRT doesn't include distant effects like shadowing of the bat's wings onto its body, or effects from multiple occluding translucent layers (Figure 7c). The high-frequency local geometry is rendered properly in the frontmost layer, masking these errors and maintaining a realistic appearance.

Two methods generate LDPRT from geometry: simulating a local texture patch which can be mapped onto various objects (Figure 7), or PRT simulation on an entire mesh which can then be deformed. Figure 6cde shows results from the latter method on a Mayan head. The mesh has been bent and twisted; the figure compares ray tracing the deformed model (a), applying the undeformed PRT shading (b), and LDPRT (c). LDPRT matches better by allowing the local coordinate frame to be rotated.

Local patch geometry is specified by a mesh (waffle, bumps) or by inferring height fields from images (weave, swirl). We used the simple technique of mapping albedo to height. All LDPRT textures are diffuse except Figure 7d. For diffuse textures, we used n=6 and N=4 except for the bat (Figure 1) where n=3 and N=1. For the glossy texture, n=4, N=3, and n_b=3 for 9 total lobes. Our parametric models are shown in Figure 1 (thin membrane) and Figure 6ab (wrinkle), where the groove location is made a function of angle around a smooth cylinder which is then deformed. For LDPRT textures where N>1, DC and linear SH coefficients are stored directly and can be bilinearly interpolated; ZH lobes are fit to the remaining, higher-order SH coefficients and are point sampled.

Preprocessing is dominated by the ZH fit. The most expensive fits were less than 30 minutes: the Mayan head with 17575 vertices, and the subsurface waffle (64×64) and swirl (128×128) textures. Preprocessing time depends on the size of the texture (or number of mesh vertices), the number of lobes, and the complexity of the PRT signal. Smaller textures (e.g. weave) take just a few minutes; Figure 7 specifies texture resolutions. PRT simulations took from 4-10 minutes with 5000 rays per point. All examples were simulated and rendered with single-channel transfer. Timings are for a 3.0Ghz P4 processor with a Geforce 6800 Ultra graphics card.

We recorded run-time rendering rates of between 100-200Hz for the diffuse examples (3-channel lighting) and 70-100Hz for the glossy example (1-channel lighting). This includes the time to build the light specialized map each frame (1430Hz by itself).

7 Conclusion and Future Work

PRT provides compelling shading effects that until now have been applied only to static models or sequences precomputed at every pose. We lift this restriction by allowing PRT signals from local geometry to be mapped onto any deformable object. Using sums of zonal harmonics about a few axes, we obtain a general transfer approximation that can be trivially re-oriented at every point on a deforming model and rendered in real-time. The approximation accurately fits PRT signals derived from simulations on arbitrary local geometry or simple parametric models.

Though our method is limited to low-frequency lighting and surfaces that aren't highly specular, for local transfer this is a practical form of antialiasing. Properly filtered, high-frequency shading effects like sharp shadows and glints are less noticeable on textural geometry meant to be seen at a distance. Antialiasing these effects with image supersampling is difficult and often results in temporal aliasing; bandlimiting lighting and reflectance is much more effective.

The next, challenging step toward fully dynamic PRT is to simulate distant effects. We are currently working on low-frequency transfer models that can be evaluated on-the-fly, especially for dynamic, articulated characters. This will allow effects like cast soft shadows from moving arms and legs.

Acknowledgments

Thanks to David Thiel for video editing and Shanon Drone for creating the bat and Mayan head models. Jonathan Steed wrote the LDPRT demo that ships with the Microsoft DirectX 9.0 Update (December 2004).

References

ASHIKHMIN, M., AND SHIRLEY, P. 2002. Steerable illumination textures. *ACM Trans. Gr. 2*, 3.

CHOI, C., IVANIC, J., GORDON, M., AND RUEDENBERG, K. 1999. Rapid and stable determination of rotation matrices between spherical harmonics by direct recursion. *The Journal of Chemical Physics 111*, 19 (November), 8825–8831.

COOK, R., AND TORRANCE, K. 1982. A reflectance model for computer graphics. *ACM Trans. Gr. 1*, 1, 7–24.

DANA, K., NAYAR, S., VAN GINNEKEN, B., AND KOENDERINK, J. 1999. Reflectance and texture of real-world surfaces. *ACM TOG 18*, 1–34.

DRISCOLL, J., AND HEALY, D. 1994. Computing fourier transforms and convolutions on the 2-sphere. *Adv. in Appl. Math. 15*, 202–250.

FERNANDO, R. 2004. *GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics*. Addison-Weseley Professional.

HEIDRICH, W., DAUBERT, K., KAUTZ, J., AND SEIDEL, H. 2000. Illuminating micro-geometry based on precomputed visibility. In *Proc. SIGGRAPH '00*, 455–464.

JAMES, D., AND FATAHALIAN, K. 2003. Precomputing interactive dynamic deformable scenes. In *Proc. of SIGGRAPH '03*, 879–887.

KANEKO, T., TAKAHEI, T., INAMI, M., KAWAKAMI, M., YANAGIDA, Y., MAEDA, T., AND TACHI, S. 2001. Detailed shape representation with parallax mapping. In *Proc. of ICAT 2001*, 205–208.

KAUTZ, J., SLOAN, P., AND SNYDER, J. 2002. Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics. *Eurographics Workshop on Rendering*, 291–296.

KAUTZ, J., LEHTINEN, J., AND AILA, T. 2004. Hemispherical rasterization for self-shadowing of dynamic objects. *Proceedings Eurographics Symposium on Rendering*, 179–184.

LAFORTUNE, E., FOO, S., TORRANCE, K., AND GREENBERG, D. 1997. Nonlinear approximation of reflectance functions. In *Proc. of SIGGRAPH '97*, 117–126.

LENSCH, H., KAUTZ, J., GOESELE, M., HEIDRICH, W., AND SEIDEL, H. 2001. Image-based reconstruction of spatially-varying materials. In *EG Rendering Workshop*, 103–114.

LIU, X., SLOAN, P., SHUM, H., AND SNYDER, J. 2004. All-frequency precomputed radiance transfer for glossy objects. In *Proc. of 2004 Eurographics Symposium on Rendering*.

MALZBENDER, T., GELB, D., AND WOLTERS, H. 2001. Polynomial texture maps. In *Proc. of SIGGRAPH '01*, 519–528.

MAX, N. 1988. Horizon mapping: shadows for bump-mapped surfaces. *The Visual Computer 4*, 2, 109–117.

MCCALLISTER, D., LASTRA, A., AND HEIDRICH, W. 2002. Efficient rendering of spatial bidirectional reflectance distribution functions. In *Graphics Hardware '02*, 171–178.

MEYER, A., AND NEYRET, F. 1998. Interactive volumetric textures. In *Eurographics Workshop on Rendering*, 157–168.

MILLER, G. 1994. Efficient algorithms for local and global accessibility shading. In *Proc. of SIGGRAPH '94*, 319–326.

MULLER, G., MESETH, J., AND KLEIN, R. 2004. Fast environmental lighting for local-pca encoded btfs. In *Proc. of Computer Graphics International*, 198–205.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2003. All-frequency shadows using non-linear wavelet lighting approximation. In *Proc. of SIGGRAPH '03*, 376–381.

NOCEDAL, J., AND WRIGHT, S. 1999. *Numerical Optimization*. Springer-Verlag.

OLIVEIRA, M., BISHOP, G., AND MCCALLISTER, D. 2000. Relief texture mapping. In *Proc. of SIGGRAPH '00*, 359–368.

PRESS, W., TEUKOLSKY, S., VETTERLING, W., AND FLANNERY, B. 1992. *Numerical Recipes in C, Second Edition*. Cambridge University Press, Cambridge, England.

RAMAMOORTHI, R., AND HANRAHAN, P. 2001. An efficient representation for irradiance environment maps. In *Proc. of SIGGRAPH '01*, 497–500.

SHIRLEY, P., AND CHIU, K. 1997. A low distortion map between disk and square. *Journal of Graphics Tools 2*, 3, 45–52.

SLOAN, P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proc. of SIGGRAPH '02*, 527–536.

SLOAN, P., HALL, J., HART, J., AND SNYDER, J. 2003. Clustered principal components for precomputed radiance transfer. In *Proc. of SIGGRAPH '03*, 382–391.

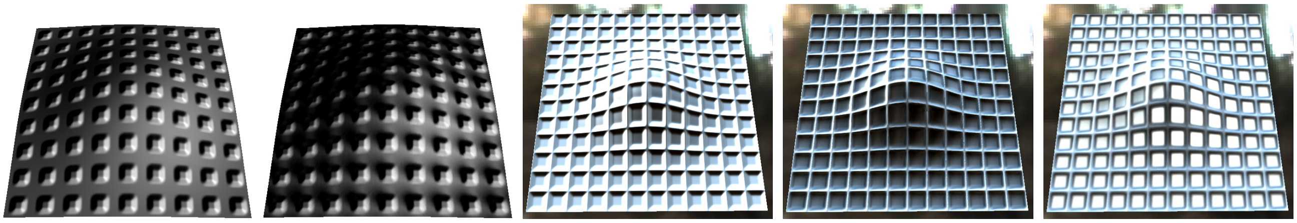
SLOAN, P., LIU, X., SHUM, H., AND SNYDER, J. 2003. Bi-scale radiance transfer. In *Proc. of SIGGRAPH '03*, 370–375.

WANG, L., WANG, X., TONG, X., LIN, S., HU, S., GUO, B., AND SHUM, H. 2003. View-dependent displacement mapping. In *Proc. of SIGGRAPH '03*, 334–339.

WANG, R., TRAN, J., AND LUEBKE, D. 2004. All-frequency relighting of non-diffuse objects using separable brdf approximation. In *Proc. of 2004 Eurographics Symposium on Rendering*.

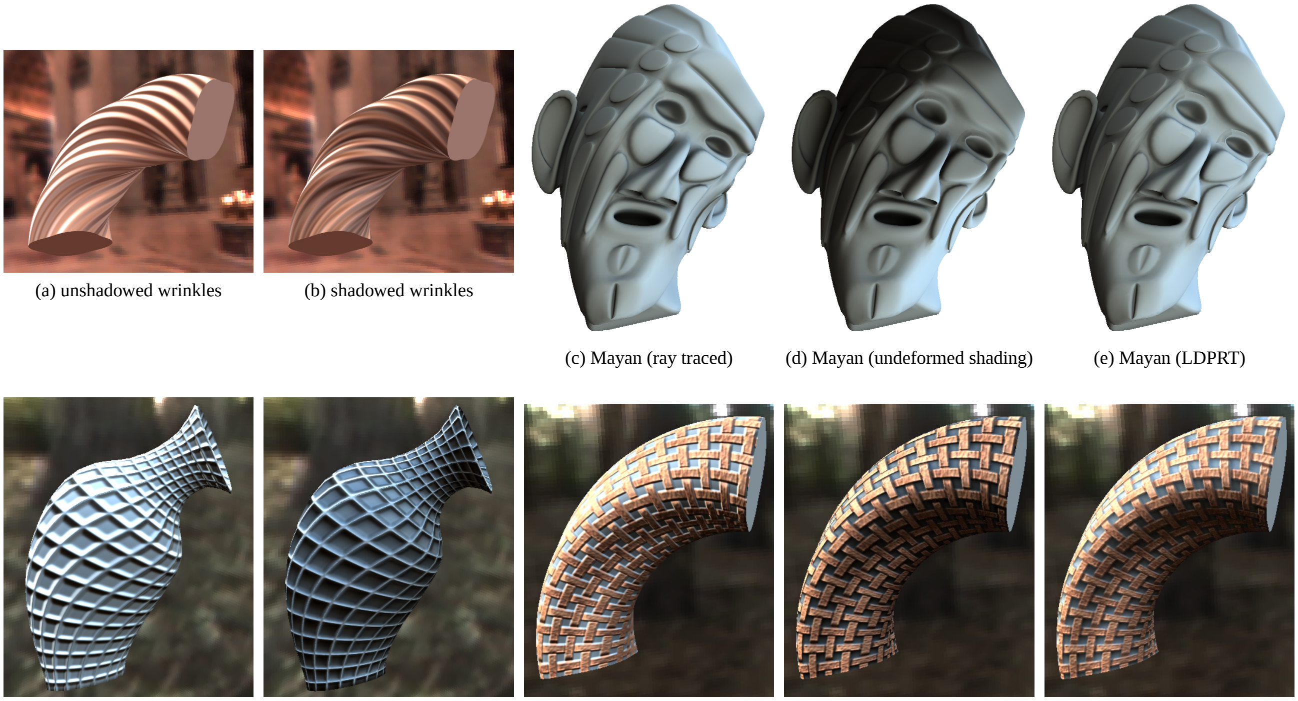
WESTIN, S., ARVO, J., AND TORRANCE, K. 1992. Predicting reflectance functions from complex surfaces. In *Proc. of SIGGRAPH '92*, 255–272.

WHITTAKER, E., AND WATSON, G. 1990. *A Course in Modern Analysis, 4th Ed.* Cambridge University Press, Cambridge, England.



(a) unshadowed bumps (b) shadowed bumps (c) unshadowed waffle (d) shadowed waffle (e) subsurface waffle

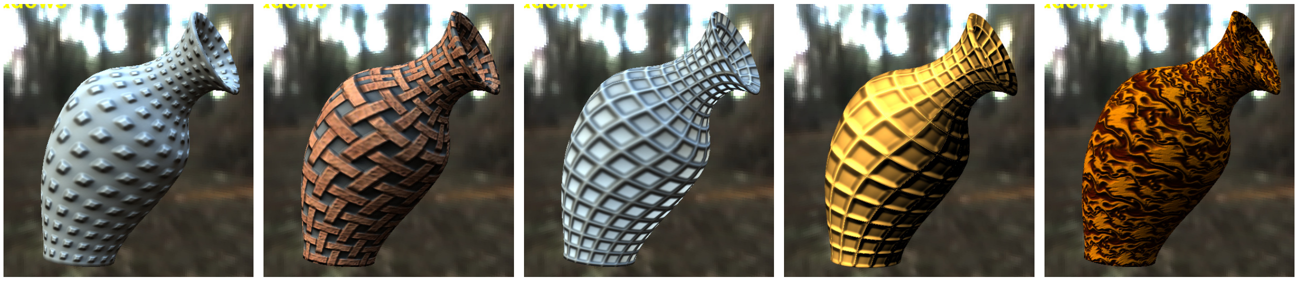
Figure 5: LDPRT on a deformed patch.



(a) unshadowed wrinkles (b) shadowed wrinkles (c) Mayan (ray traced) (d) Mayan (undeformed shading) (e) Mayan (LDPRT)

(f) unshadowed vase (g) shadowed vase (h) unshadowed cylinder (i) shadowed cylinder (j) inter-reflected cylinder

Figure 6: LDPRT on deformed objects.



(a) bumps, 64×64 (b) weave, 32×32 (c) waffle (subsurface), 64×64 (d) waffle (glossy), 128×128 (f) swirl, 128×128

Figure 7: LDPRT textures mapped onto a deformed vase.