Normal Mapping for Precomputed Radiance Transfer

Peter-Pike Sloan
Microsoft Corporation

Figure 1: Comparison of PRT, and four variants of normal mapping: Gold Standard, Separable, Half-Life 2 basis, Shifted Associated Legendre Polynomials

Abstract

Normal mapping is a variant of bump mapping that is commonly used in computer games. It models complex surface variations by explicitly storing a surface normal in a texture map. However, it has not been used with precomputed radiance transfer (PRT), a technique for modeling an object's response to a parameterized model of lighting, which enables interactive rendering of complex global illumination effects such as soft shadows and inter-reflections. This paper presents several techniques that effectively combine normal mapping and precomputed radiance transfer for rigid objects. In particular, it investigates representing the convolved radiance function in various bases and borrowing concepts from the separable decomposition of BRDF's.

Keywords: Graphics Hardware, Illumination, Normal Mapping, Rendering, Shadow Algorithms

1. Introduction

Generating realistic images of complex scenes at interactive rates is a challenging problem. Games traditionally have used a combination of static light maps and multiple dynamic point or directional light sources. The surfaces generally are textured with both reflectance properties and normal maps [Blinn 1978; Peercy et al. 1997] that approximate complex surface details. These textures tend to be sampled at high spatial sampling rates and are often layered and tiled to create even higher effective sampling rates. In contrast, precomputed lighting techniques require unique representations of a signal and are stored at much lower sampling rates. It is impractical to sample these unique details at the effective composite sampling rate.

A recent technique, precomputed radiance transfer [Sloan et al. 2002], enables complex global illumination effects like soft shadows and inter-reflections to be rendered in real time for rigid objects under dynamic distant lighting. While a technique has been proposed to model these higher frequencies [Sloan et al. 2003b], it is too heavyweight for computer games.

Normal maps have been successfully integrated with global illumination in both off-line production [Tabellion and Lamorlette 2004] and interactive games [McTaggart 2004] under static global lighting. Inspired by this work, we propose several simple techniques that extend diffuse PRT for rigid objects to handle normal maps. They are lightweight, and they only model global illumination effects at coarse scales, neglecting fine scale shadowing of the surface details.

2. Background and Related Work

Static global illumination and normal mapping have been successfully integrated both off-line [Tabellion and Lamorlette 2004] and in the recent game Half-Life 2 [McTaggart 2004]. Our work is heavily influenced by the latter paper, which is effectively a vector irradiance formulation of radiosity that makes normal mapping effective. We shall investigate several hemispherical bases for representing parameterized global illumination, and include a novel basis based on concepts used for representing BRDF's [Kautz and McCool 1999].

2.1 Precomputed Radiance Transfer

PRT [Sloan et al. 2002] enables interactive rendering of complex global illumination effects for rigid objects. For completeness, we briefly describe PRT and discuss why previous work does not adequately address normal mapping. In this paper scalar quantities and low dimensional vectors are denoted in italics, matrices with upper case bold letters and vectors with lower case bold letters. The general mathematical form of PRT is:

$$e(v) = \mathbf{b}(v)^T \mathbf{R}\mathbf{M}\boldsymbol{l}$$

Here $e(v)$ is outgoing radiance, $\mathbf{b}(v)$ is a vector that represents the materials response to lighting in view direction $\mathbf{v}$ expressed in a local coordinate system; $\mathbf{R}$ is a matrix that rotates lighting into the local frame; and $\mathbf{M}$ is a matrix that maps distant lighting $\boldsymbol{l}$ to transferred incident radiance. For diffuse surfaces, a single transfer vector $\mathbf{t}$ has been used that directly models the relationship between distant lighting and outgoing radiance. These spatially varying operators are computed using slightly modified off-line global illumination techniques; for details see [Sloan et al. 2002;Sloan et al. 2003a;Lehtinen 2004].

These formulations have fine surface details burned into the model of transfer, which precludes their use with normal mapping. One exception to this is the bi-scale radiance transfer technique [Sloan et al. 2003b], which, like this work, models coarse effects using operators that map from distant lighting to smooth incident radiance. However, in that work the fine scale was modeled using a heavyweight representation that modeled both masking effects and arbitrary materials. We represent convolved incident radiance with a more compact basis which makes our technique much lighter weight, but can only handle diffuse materials and does not model shading effects at the fine scales of the normal map. Local, Deformable Precomputed Radiance Transfer (LDPRT) [Sloan et al. 2005] models only fine

scale transfer effects for deformable objects, while this work models only coarse scale effects for rigid objects.

2.2 Spherical Harmonics

Spherical harmonics are the spherical analog of the fourier basis on the unit circle and have been used extensively in computer graphics. In this paper they are used to represent both the distant lighting environment and the convolved local lighting environment. The general form is:

$$Y_l^m(\theta,\varphi) = K_l^m\, e^{im\varphi} P_l^{|m|}(\cos\theta),\ l \in \boldsymbol{N},\ -l \le m \le l$$

where P_l^m are the associated Legendre polynomials and K_l^m are the normalization constants

$$K_l^m = \sqrt{\frac{(2l+1)}{4\pi}\frac{(l-|m|)!}{(l+|m|)!}}$$

The above definition forms a complex basis; a real-valued basis is given by the simple transformation

$$y_l^m = \begin{cases} \sqrt{2}\,\mathrm{Re}(Y_l^m), & m>0 \\ \sqrt{2}\,\mathrm{Im}(Y_l^m), & m<0 \\ Y_l^0, & m=0 \end{cases} = \begin{cases} \sqrt{2}\,K_l^m \cos(m\varphi) P_l^m(\cos\theta), & m>0 \\ \sqrt{2}\,K_l^m \sin(-m\varphi) P_l^{-m}(\cos\theta), & m<0 \\ K_l^0 P_l^0(\cos\theta), & m=0 \end{cases}$$

Low values of *l* (called the *band* index) represent low-frequency basis functions over the sphere. The basis functions for band *l* reduce to polynomials of order *l* in *x*, *y*, and *z*.

Spherical harmonics can only efficiently represent smooth lighting environments. Smooth lighting environments induce low spatial sampling rates, making them more practical for applications like computer games.

2.3 Irradiance Environment Maps

An irradiance environment map [Ramamoorthi and Hanrahan 2001] enables interactive rendering of diffuse objects under any illumination environment without shadowing. This is done by representing the lighting environment and the normalized cosine kernel[1] directly in spherical harmonics. Convolution can be performed efficiently in this space, and only the first 9 terms of the convolved lighting environment are needed to represent it accurately. Irradiance environment maps are used in this paper to efficiently decouple normal variation from lighting variation.

3. Normal Mapping for PRT

At any point on the surface of an object a transfer matrix represents how distant lighting expressed in some lighting basis (spherical harmonics for this work) maps to transferred incident radiance. Even if the object has complex normal variation, this incident radiance is a smooth function, while the outgoing or exit radiance has much more variation. Decoupling normal variation from incident radiance, makes practical rendering objects with complex surface variation.

The straightforward way to address this for diffuse objects is: exploit the observation in other papers [Ramamoorthi and Hanrahan 2001; Basri and Jacobs 2001], and have transfer matrices that project into quadratic spherical harmonics followed by a convolution with the clamped cosine kernel. Then, interpolate irradiance at some coarse scale, and simply evaluate this using a normal, looked up in a normal map. This can be expressed mathematically as follows:

$$e(n) = \mathbf{y}(n)^T\ \mathbf{CRM}\ \boldsymbol{l} \qquad (3.1)$$

Here, $e(n)$ is the outgoing radiance as a function of the surface normal n; $\mathbf{y}(n)$ is a vector generated by evaluating the quadratic spherical harmonic basis functions in the normal direction; **C** is a diagonal matrix that convolves quadratic spherical harmonics with the normalized cosine kernel; **R** is a spherical harmonic rotation matrix that rotates a quadratic SH function into the local frame; and **M** is a transfer matrix that maps the varying distant lighting ***l*** of some order into quadratic local lighting.

Unfortunately this is still too heavyweight; in particular, 27 numbers are required to represent the irradiance environment map. If this is computed at the vertices of the mesh, 7 of the 8 interpolators have to be consumed for this alone. The rest of this paper is focused on dealing with efficient approximations to the above expressions.

3.1 Projection into Hemispherical Basis

One possible solution that we will briefly discuss here is to simply project the irradiance environment map into a hemispherical basis. Two obvious choices are the basis used in the paper [McTaggart 2004], which consist of three clamped linear basis functions and the basis used in the paper [Tabellion and Lamorlette 2004] which is effectively determined from the local linear spherical harmonic lighting coefficients but does not include the DC term[2].

In this paper we investigate using the Half-Life 2 basis. In the GDC presentation, shader code was presented with and without clamping values to zero. For PRT, not clamping produced better visual results and that is what is used in the paper. The three basis functions are:

$$b_0 = \left\{-\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}\right\} \quad b_1 = \left\{-\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}\right\}$$

$$b_2 = \left\{\sqrt{\frac{2}{3}}, 0, \frac{1}{\sqrt{3}}\right\}$$

They are orthogonal when integrated over the hemisphere, and can be normalized by scaling by $3/(2\pi)$.

Another basis choice is the one proposed in Gautron's paper [Gautron 2004]. This is an orthogonal basis generated by applying the shifting theorem to the spherical harmonic basis. These basis functions can be expressed based on the Euclidean coordinates of a point on the unit hemisphere. The first four basis functions are:

$$H_0^0 = \frac{\sqrt{2}}{2\sqrt{\pi}} \qquad H_1^{-1} = -\frac{\sqrt{6}\sqrt{z-z^2}\,y}{\sqrt{\pi}\sqrt{1-z^2}}$$

$$H_1^0 = \frac{\sqrt{6}(2z-1)}{2\sqrt{\pi}} \qquad H_1^1 = -\frac{\sqrt{6}\sqrt{z-z^2}\,x}{\sqrt{\pi}\sqrt{1-z^2}}$$

These functions are clearly not polynomials except for the L=0 basis functions, but can be evaluated efficiently using shaders. When z=1 the limit of the functions with z in the denominator

[1] Irradiance needs to be converted to outgoing radiance to be visualized. This can be done by convolving, using a normalized cosine kernel (divide by Pi) and keeping the albedo in [0,1].

[2] The dominant lighting direction that is referred to in this paper can be determined from the linear SH coefficients. However, the cosine term would have to be neglected to model the same result (it is later added in using the actual surface normal).

equals 0, $x/\text{sqrt}(1-z^2) = \cos(\text{phi})$, $y/\text{sqrt}(1-z^2) = \sin(\text{phi})$, and so the functions are defined over the complete hemisphere.

Projection matrices from spherical harmonics for both bases are in the appendices.

3.2 Separable Approximation

An alternative approach is to use the concepts from separable BRDF factorization [Kautz and McCool 1999] to build a minimal specialized basis. Separable approximations have been used with PRT to model glossy surfaces [Liu et al. 2004;Wang et al. 2004]. The lighting is treated the same way, but in this case normal variation in tangent space for diffuse surfaces is being modeled instead of view variation for arbitrary materials. As in the glossy case, the goal is to reduce the dimensionality of the matrix, reducing both the size of the dataset and the amount of computation required in the shaders.

Moving to a directional basis, Equation 3.1 can be re-written as:

$$e(n) = \mathbf{b}(n)^T \, \mathbf{ACRM}\, \boldsymbol{l} \tag{3.2}$$

Where $\mathbf{b}(n)$ is a vector of coefficients for bi-linear basis functions on the unit square mapped to the hemisphere [Shirley and Chiu 1997], which has at most 4 non-zero values for a given normal. Coefficient a_{ij} of the matrix **A** represents evaluating the quadratic spherical harmonic basis function j in normal direction i. In this paper 1024 discrete normal directions are used (32x32 samples on the unit square) so this matrix is 1024x9.

Computing the singular value decomposition [Press et al. 1992] of **A**, factors the matrix into a product of 3 matrices **USV**, where **U** is a 1024x9 orthogonal matrix, **S** is a 9x9 diagonal matrix and **V** is a 9x9 orthogonal matrix. Using the first M singular values, truncates the matrix **U** to have M columns and the matrix **V** M rows. This leaves the final approximation:

$$e(n) = \left(b(n)^T \, \mathbf{U}_m\right)\mathbf{S}_m\mathbf{V}_m\mathbf{CRM}\, \boldsymbol{l} \tag{3.3}$$

The term in parentheses is an M dimensional row vector that is a function of the surface normal. This can be efficiently evaluated using texturing hardware by packing the columns of $\mathbf{U}_m$ into individual color channels of a texture, and, if M>4, into multiple textures. In this paper we used M=4, and instead of using a texture that computes the hemisphere to unit square mapping (or complex shader code), we resample the columns of $\mathbf{U}_m$ into a texture parameterized by an orthographic projection of the normal vector (ignoring z in tangent space) which is represented at a higher resolution (64x64.) This texture is then sampled directly.

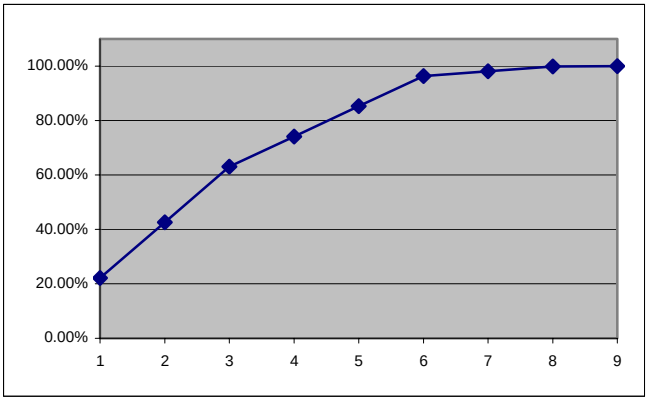

Figure 2: Accuracy vs. number of singular values

The figures in the paper use 4 rows, requiring 3 interpolants to be passed from the vertex shader to the pixel shader. If the convolution matrix **C** is absorbed into the matrix **A** we believe the SVD will result in higher accuracy.

4. Rendering

Shading is straightforward using these techniques. In every frame, the transfer matrices at the vertices have to be multiplied by the distant lighting environment. Then the coefficients for the correct basis are passed down to the pixel shaders, where the current normal is looked up in the normal map and used to evaluate the particular basis. The separable approximation is more involved, since the normal dependent textures have to be de-referenced as well. The shader code below is the pixel shader when M=4. The vertex shader passes down the texture coordinates and three registers that contain the basis coefficients. There are three textures that are sampled: the normal dependent texture, the normal map, and the albedo of the surface.

```
float4 StandardSVDPS( VS_OUTPUT_SVD In )
{
  float4 RGBColor(0,0,0,1);
  // sample albedo/normal map
  float4 vAlbedo = tex2D(AlbedoTex, In.uv);
  float4 vNormal = tex2D(NormTex, In.uv);

  // sample normal dependent texture
  float4 vU = tex2D(USampler,vNormal);

  // compute irradiance
  RGBColor.r = dot(In.cR,vU);
  RGBColor.g = dot(In.cG,vU);
  RGBColor.b = dot(In.cB,vU);

  // scale by albedo and return
  return RGBColor*vAlbedo;
}
```

The transfer matrices can be compressed using CPCA [Sloan et al. 2003a] shifting more of the workload to the GPU, decreasing the amount of data and increasing performance.

5. Results

We have implemented all of the techniques in the paper. Table 1 presents a synopsis of the results for the two models shown in the paper. The precomputation of the transfer matrices is essentially the same as in earlier papers [Sloan et al. 2002; Sloan et al. 2003b] and projection into either the light specialized separable or the analytic basis is a simple matrix multiply.

Scene	#f	#v	GS	Sep	SAL	HL2	GPUGS	GPUSep
Simple	1296	722	233	476	477	480	1040	1327
Complex	60126	31726	13.7	30	30	30	255	487

Table 1: Performance results (fps), GS: Gold Standard, Sep: Separable, SAL: Shifted Associated Legendre, HL2: Half-Life 2 basis, GPUGS: Gold Standard+PCA, GPUSep: Separable+PCA

Visually all the techniques have reasonable fidelity, the shifted associated legendre polynomials subjectively looks the worst. Not much time has been spent optimizing the CPCA parameters for these datasets; in fact, the examples are all reported with the simple parameters of a single cluster and 24 PCA basis vectors. All of these examples should require fewer clusters compared to what was used in previous work [Sloan et al. 2003a], because the transfer matrices represent a signal that has been convolved with a normalized cosine lobe and have fewer rows. All timings are in

frames per second and were recorded on a 2.2ghz AMD Opteron with a nVidia GeForce 6800 graphics card.

Figure 1 shows a comparison on the complex model between conventional PRT and these normal mapping variants. Note the complex details on the side of the object that are completely encoded in the normal map.

Figure 3 shows three normal maps mapped onto the simple object rendered with PRT and all these techniques. Note the differences on the "bumpy" object.

6. Conclusions

Normal mapping is a compelling technique that has been used with analytic lighting and with static precomputed lighting. We extend these techniques to practical implementations for more general precomputed lighting techniques, and investigate a family of solutions that trade off cost with accuracy. Of the three techniques presented, the analytic basis used in the paper [McTaggart 2004] and the separable basis with a four term approximation seem the most promising. While local transfer is neglected with the presented techniques, ideas from ambient occlusion can be used to generate a crude approximation. In particular, the product of the lighting environment and the DC approximation to the visibility function, when projected into spherical harmonics, results in simply scaling the lighting environment by this DC term. This turns out to be a mathematical justification for a common ambient occlusion technique.

In the future it is worth investigating compression of the transfer matrices more carefully, and possibly integrating a more sophisticated model of local transfer while remaining lightweight enough to be practical for game applications. Also, folding the convolution into the **A** matrix could lead to better results.

7. Acknowledgements

The Drone model and textures were created by Shannon Drone, and the normal maps were created by Ben Luna. I would like to thank James Sloan for editing and the anonymous reviewers for their helpful comments.

References

BASRI, R., AND JACOBS, D. Lambertian Reflectance and Linear Subspaces, ICCV 2001.

BLINN, J. Simulation of Wrinkled Surfaces. SIGGRAPH 1978.

GAUTRON, P., KRIVANEK, K. PATTANAIK, S., AND BOUATOUCH K. A Novel Hemispherical Basis for Accurate and Efficient Rendering, Eurographics Symposium on Rendering 2004.

KAUTZ J., AND MCCOOL, M. Interactive Rendering with Arbitrary BRDFs Using Separable Approximations. Eurographics Workshop on Rendering 1999.

LEHTINEN, J. Foundations of Precomputed Radiance Transfer, Master's Thesis, Helsinki University of Technology, September 2004.

LIU, X., SLOAN, P., SHUM, H., AND SNYDER, J. All-Frequency Precomputed Radiance Transfer for glossy objects. Eurographics Symposium on Rendering 2004.

MCTAGGART, G. Half-Life 2 Source Shading, GDC 2004 .

PEERCY, M., AIREY, J., AND CABRAL, B. Efficient Bump Mapping Hardware, SIGGRAPH 1997

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

RAMAMOORTHI, R., AND HANRAHAN, H. An Efficient Representation for Irradiance Environment Maps, SIGGRAPH 2001.

SHIRLEY, P, AND CHIU, K, A Low Distortion Map between Disk and Square, Journal of Graphics Tools, vol. 2, no. 3, 1997, 45–52.

SLOAN, P., KAUTZ, J., AND SNYDER, J. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environ-ments, SIGGRAPH 2002.

SLOAN, P,. HALL, J., HART, J., AND SNYDER, J. Clustered Principal Components for Precomputed Radiance Transfer, SIGGRAPH 2003.

SLOAN, P., LIU, X., SHUM, H., AND SNYDER J. Bi-Scale Radiance Transfer, SIGGRAPH 2003.

SLOAN, P, LUNA, B, AND SNYDER J, Local, Deformable Precomputed Radiance Transfer, SIGGRAPH 2005.

TABELLION, E., AND LAMORLETTE, A. An Approximate Global Illumination System for Computer-Generated Films, SIGGRAPH 2004

WANG, R., TRAN, J., AND LUEBKE, D. All-Frequency Relighting of Non-Diffuse Objects Using Separable BRDF Approximation, Eurographics Sumposium on Rendering 2004.

7. Appendix: Other Basis Functions

7.1 Shifted Associated Legendre Polynomials

The matrix below projects the spherical harmonics into the first four shifted associated legendre polynomials in closed form.

$$\begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{6}}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a & 0 & 0 & 0 & b & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{4} & 0 & 0 & 0 & \frac{\sqrt{30}}{8} & 0 & 0 \\ 0 & 0 & 0 & a & 0 & 0 & 0 & b & 0 \end{bmatrix}$$

Where:

$$a = \frac{44 - 9\sqrt{2}\ln\left(3+2\sqrt{2}\right)}{32}, b = \frac{39\sqrt{10}\ln\left(3+2\sqrt{2}\right) - 20\sqrt{5}}{256}$$

7.2 Half-Life 2 Basis

The quadratic spherical harmonics can be projected into this basis using the following matrix:

$$\begin{bmatrix} \frac{\sqrt{3}}{4\sqrt{\pi}} & \frac{\sqrt{6}}{4\sqrt{\pi}} & \frac{1}{2\sqrt{\pi}} & \frac{\sqrt{2}}{4\sqrt{\pi}} & 0 & \frac{3\sqrt{30}}{32\sqrt{\pi}} & \frac{1\sqrt{15}}{16\sqrt{\pi}} & \frac{3\sqrt{10}}{32\sqrt{\pi}} & 0 \\ \frac{\sqrt{3}}{4\sqrt{\pi}} & \frac{-\sqrt{6}}{4\sqrt{\pi}} & \frac{1}{2\sqrt{\pi}} & \frac{\sqrt{2}}{4\sqrt{\pi}} & 0 & \frac{-3\sqrt{30}}{32\sqrt{\pi}} & \frac{1\sqrt{15}}{16\sqrt{\pi}} & \frac{3\sqrt{10}}{32\sqrt{\pi}} & 0 \\ \frac{\sqrt{3}}{4\sqrt{\pi}} & 0 & \frac{1}{2\sqrt{\pi}} & \frac{-\sqrt{2}}{2\sqrt{\pi}} & 0 & 0 & \frac{1\sqrt{15}}{16\sqrt{\pi}} & \frac{-3\sqrt{10}}{16\sqrt{\pi}} & 0 \end{bmatrix}$$

7.3 Numerical Comparison

Below is a table that shows the mean squared error (MSE) integrated over the hemisphere when projecting each of the quadratic spherical harmonic basis functions into these two bases. If the corresponding basis function is in the null space of the basis, the MSE equals 0.5, and as shown below basis functions Y_2^{-2} and Y_2^2 are in the null space of both basis. It is also worth pointing out that those two basis functions are also in the null space of the first four SVD coefficients; the 5th and 6th coefficients are exactly those basis functions.

	Y_0^0	Y_1^{-1}	Y_1^0	Y_1^1	Y_2^{-2}	Y_2^{-1}	Y_2^0	Y_2^1	Y_2^2
SAL	0	4.6e-2	0	4.6e-2	5e-1	4.5e-2	3.1e-2	4.5e-2	5e-1
HL2	1.3e-1	0	0	0	5e-1	1.5e-1	3.8e-1	1.5e-1	5e-1

Table 2: Mean squared error integrated over hemisphere. SAL: Shifted Associated Legendre, HL2: Half-Life 2 basis.

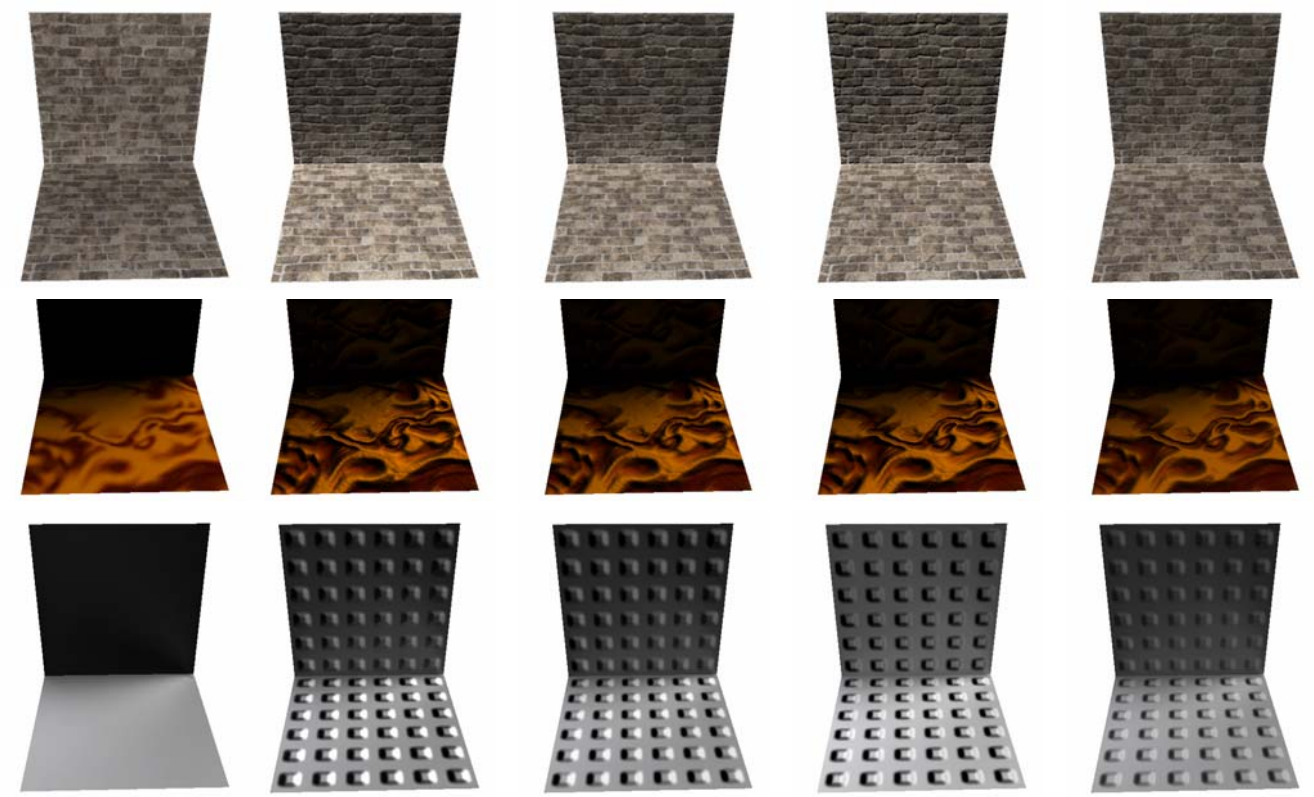

Figure 3: Simple scene, PRT, Gold Standard, Separable, Half-Life 2, Shifted Associated Legendre Polynomials