Eurographics Symposium on Rendering (2007)
Jan Kautz and Sumanta Pattanaik (Editors)

Precomputed Radiance Transfer for Real-Time Indirect Lighting using A Spectral Mesh Basis

Rui Wang[1], Jiajun Zhu[2], and Greg Humphreys[2]

[1] University of Massachusetts Amherst
[2] University of Virginia

Abstract

Simulating indirect lighting effects has been a challenging topic in many real-time rendering and design applications. This paper presents a novel method, based on precomputed radiance transfer, for rendering physically based, multi-bounce indirect lighting in real-time. Our key idea is to represent both the direct lighting and precomputed diffuse indirect transfer using a spectral mesh basis set derived from an arbitrary scene model [KG00]. The complete spectral basis set can approximate a spatially varying function to any degree of accuracy. For indirect lighting, we show that only $60 \sim 100$ *sparse basis coefficients suffice to achieve high accuracy, due to the low-frequency nature of indirect illumination. This reduces the run-time computation of per-vertex diffuse indirect lighting to simple inner products of two sparse vectors: one representing the dynamic direct lighting, and the other representing the precomputed direct to indirect transfer. The key advantage using this approach is that we are not restricted to parameterized models or any particular mesh topology. Our method simulates multiple diffuse interreflections while at the same time permitting dynamically changing surface albedos. In addition, we approximate the final bounce of glossy interreflection using a standard BRDF SH projection. Finally, we demonstrate high-quality indirect lighting effects rendered at* $15 \sim 30$ *fps with dynamically changing lighting and materials.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism;

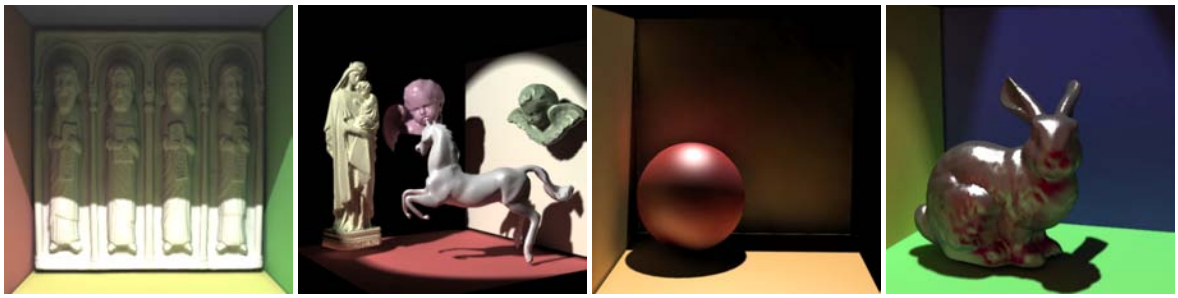
1. Introduction

Accurate simulation of indirect lighting effects is crucial to many real-time applications such as games, lighting design, and cinematic relighting. Many current approaches use expensive global illumination algorithms such as ray tracing or photon mapping to simulate physically correct indirect lighting, but these algorithms are impractical to use directly in real-time. Recent advances in Precomputed Radiance Transfer (PRT) [SKS02, NRH03] have enabled real-time realistic global illumination by assuming static scenes. PRT represents precomputed illumination effects using basis functions such as spherical harmonics (SH) [SKS02] or wavelets [NRH03], and linearly combines the results to simulate novel rendering at run-time. Because projection onto a basis requires an appropriate domain parameterization, existing PRT techniques typically assume that the direct lighting is distant and can be adequately represented as an 2D environment map. This assumption makes PRT difficult to apply in applications that require dynamic local lighting, such as computer games or indoor scene design.

We present a PRT-based method for interactive rendering of high-quality, multi-bounce indirect lighting under arbitrary direct lighting models. Our technique parameterizes the direct illumination over the 2D domain of mesh surfaces. This domain supports a frequency space projection method by using a *spectral basis* [KG00], which exists for meshes with arbitrary topology. Our key idea is to project both the direct lighting and the precomputed diffuse indirect transfer functions onto this spectral basis set. The low-frequency nature of indirect illumination permits a high fidelity approximation with only a small set ($60 \sim 100$) of sparse basis coefficients. The key advantage is that we are not restricted to parameterized models or any particular mesh topology. The diffuse indirect lighting can then be computed in real time

© The Eurographics Association 2007.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org



(a) Statues (b) Horse (c) Sphere (d) Glossy Bunny

Figure 1: *Realistic indirect lighting effects rendered using our system.*

as a single inner product of two sparse vectors, representing the dynamic direct lighting and precomputed diffuse indirect transfer. In addition, we permit the final bounce of glossy interreflection effects using a standard BRDF SH projection.

Figure 1 shows examples of our rendering results. Conceptually, our technique generalizes previous SH-based PRT [SKS02] to support local lighting effects: where they use an SH basis to approximate distant environment illumination, we use a spectral basis over the mesh to approximate arbitrary illumination, including both distant and local lighting. Using our method, we demonstrate high-quality indirect lighting effects on complex geometry at $15 \sim 30$ fps with dynamically changing viewpoint, lighting, and BRDFs.

2. Background and Related Work

Global illumination: Global illumination has been one of the central themes in computer graphics research for over two decades. Standard algorithms such as radiosity, Monte Carlo ray tracing, and photon mapping are known for their high computational cost. Even with recent advances in interactive ray tracing [RSH05, WIK*06, WBS07], real-time image synthesis under complex lighting and illumination models remains a challenging topic. Several projects have taken advantage of cached sparse illumination samples to enable interactive global illumination [WDP99, TPWG02, BWG03]. These types of techniques are typically hampered by interpolation or visibility artifacts. GPU-based global illumination algorithms [Kel97, PDC*03, NPG05, CHL04, DS06] have also been presented but are typically limited to small scenes or suffer from degraded rendering quality. Incremental instant radiosity [LSK*07] enables real-time indirect lighting on modern GPUs, but only supports simple lighting and single-bounce interreflection, and assumes that the motion of the light source is smooth.

Radiosity algorithms [CWH93] use approximation bases and finite element methods to solve the global illumination problem. The choice of basis have been extensively studied, including piecewise constant functions, linear functions [Hec92], higher order polynomials [Zat93, TM93], and wavelets [GSCH93]. These basis construction schemes require either a parameterized model or a remeshed model, while our method does not impose these limitations.

Precomputed Radiance Transfer (PRT): PRT was first introduced in [SKS02, NRH03] as a general technique for real-time rendering of global illumination effects. PRT encodes precomputed light transport effects of a static scene model by frequency space projection onto a orthonormal basis set. Run-time rendering under dynamic lighting can then be expressed linearly using precomputed illumination data. Subsequent work has extended PRT to handle changing viewpoints [LSSS04, WTL04, NRH04], local lighting [ZHL*05], deformable geometry [SLS05, RWS*06] and changing BRDFs [BAOR06]. However, existing PRT techniques typically assume distant environment lighting represented as 2D environment maps. This has limited their usefulness in many practical applications such as computer games or indoor lighting design. Annen et al. [AKDS04] extended PRT to handle mid-range illumination effects, but their method does not support arbitrary local lighting. Kristensen et al. [KAMJ05] simulate indirect lighting by parameterizing local incident light using a 3D volume and applying clustering techniques to reduce the precomputed data size. Their system is limited to omnidirectional lights.

In a recent paper, Hašan et al. [HPB06] presented a precomputed direct-to-indirect transfer technique for simulating cinematic lighting effects. They use an unstructured point cloud to sample direct illumination over the mesh surface and apply 2D wavelets to compress the precomputed diffuse indirect transfer. Mapping point clouds to a regular 2D domain for applying wavelets is non-trivial, and they propose a hierarchical construction technique to solve this problem. Their method limits the number of sample points to be an integer power of four, and they only demonstrate image relighting with a fixed viewpoint. Kontkanen et al. [KTHS06] use a 4D wavelet basis to express indirect transfer information that also accounts for glossy view-dependent effects. Their method is limited to simple param-

© The Eurographics Association 2007.

eterized geometry, while our method does not impose this limitation and can thus handle complex geometry with arbitrary mesh topology.

Spectral Mesh Basis: The spectral mesh basis was introduced by Karni and Gotsman [KG00]. They showed how classical Fourier analysis can be extended to represent functions defined over an arbitrary 3D mesh. To reduce the construction cost, they subdivide a large input mesh into submeshes using a fixed partitioning scheme. The application they explore is geometry compression, where they approximate mesh data with a small number of spectral basis coefficients. In recent years, spectral-based geometry processing has received much attention in computer graphics, such as mesh simplification [KG00], resampling [DBG*06], morphing [Ale01], and watermarking [WK05]. Our technique is the first to make use of the spectral basis to solve the real-time relighting problem.

3. Algorithms and Implementation

In this section we describe our algorithm in detail and also explain the implementation of our rendering system. Our technique makes the following assumptions:

- **Static scenes:** As in standard PRT, we assume that the scene models are fixed. It is possible to extend our technique to rigid body dynamic scenes using an approach similar to the Shadow Fields presented by Zhou et al [ZHL*05].
- **Fast direct illumination:** We assume that the direct illumination can be quickly computed using an existing technique such as shadow mapping for point lights or PRT for environment lighting and area lights.
- **Diffuse indirect transfer:** We assume multi-bounce diffuse-to-diffuse indirect transfer and ignore intermediate glossy transfer paths such as caustics. Real-time rendering of glossy transfer is a challenging topic itself and remains an open problem. We do, however, support a single final bounce of glossy interreflection by using standard SH projection of BRDFs.

3.1. Review of Spectral Mesh Basis

We first give a brief review of the spectral mesh basis [KG00]. Considering an arbitrary piecewise linear mesh as our domain of interest, the second derivative over this domain can be approximated by the Laplacian of the binary mesh adjacency matrix. The eigenvectors of this Laplacian will serve as an analog of the Fourier basis defined over the mesh. Specifically, given a graph representing the topology of a mesh consisting of n vertices, the mesh Laplacian is defined as a $n \times n$ matrix L such that:

$$L_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\frac{1}{d_i} & \text{if vertices } i \text{ and } j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases}$$

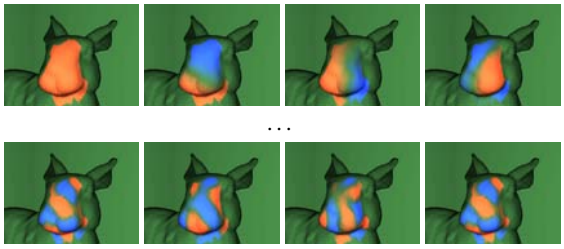


Figure 2: *Example of spectral basis functions generated on a small partition of the bunny model. Red indicates positive values and blue negative. The upper row shows some of the lowest frequency spectra, starting from the first (DC) component; the bottom row shows high frequency examples.*

where d_i is the valence of vertex i. The eigenvectors of L form an orthonormal basis that spans the mesh, and we can compute these eigenvectors using the singular value decomposition (SVD) of L. Note that the construction of such a basis set is determined only by the topology of the mesh, and is independent of the actual spatial location of each vertex. The associated eigenvalues can be thought of as frequencies in the sense of classical Fourier analysis, with the smallest eigenvalues corresponding to the lowest frequency bases. Due to the computational cost of SVD, a big mesh is typically partitioned into smaller submeshes, and the spectral basis for each submesh can then be computed individually. Figure 2 shows several representative spectral basis functions constructed for a small partition of the mesh.

Once computed, this basis set can then be used to represent any per-vertex varying signal defined over the mesh. Much like a Fourier basis, a spectral mesh basis has non-compact support and is not well suited for representing high frequency signals. However, it is very suitable for the representation of indirect illumination, which is a smoothly-varying, low-frequency effect by its nature. This property of indirect lighting has been exploited frequently by previous work to accelerate global illumination calculations, such as irradiance caching [WRC88] or photon mapping [Jen01]. The key observation is that smoothly varying spatial functions should yield a very compact representation using only a small subset of the basis functions.

3.2. Precomputing Direct-to-Indirect Transfer

In this section we derive algorithms for precomputing the direct-to-indirect transfer function. In the following, we use the function $T(x_o, x_i)$ to denote the differential form factor between two surface points x_i and x_o, taking into account the binary visibility $V(x_o, x_i)$:

$$T(x_o, x_i) \equiv V(x_o, x_i) \frac{\cos\theta_i \cdot \cos\theta_o}{\|x_i - x_o\|^2} \qquad (1)$$

One-bounce diffuse interreflection: Assuming diffuse

© The Eurographics Association 2007.

materials, the reflection equation can be written as:

$$L_o(x_o) = f_r(x_o) \int_{\mathcal{M}^2} L(x_i)\, T(x_o, x_i)\, dA(x_i) = f_r(x_o)\, E(x_o) \tag{2}$$

where $L(x_i)$ is the incoming radiance at x_i due to direct lighting, f_r is the diffuse albedo of x_o, and $E(x_o)$ represents the total irradiance received by x_o. The domain of integration $\mathcal{M}^2$ is the entire surface area of the scene. Assuming static scenes, T is a purely geometric term and can therefore be precomputed and represented as a matrix. Thus Eq. 2 can be written in matrix form as:

$$\mathbf{L}_o = \rho \cdot (\mathbf{T} \times \mathbf{L}) \tag{3}$$

where ρ is a vector containing the diffuse surface albedos, and $\mathbf{T}$ and $\mathbf{L}$ are matrix and vector versions of $T(x_o, x_i)$ and $L(x_i)$ respectively. We use the same set of mesh vertices for x_i and x_o, so the matrix $\mathbf{T}$ is an $n \times n$ symmetric matrix where n is the total number of vertices.

Let S be our set of spectral mesh basis functions. We can then project both the lighting vector $\mathbf{L}$ and the rows of $\mathbf{T}$ onto this basis, achieving a projected lighting vector $\mathbf{L}^s$ and transport matrix $\mathbf{T}^s$. Because S is orthonormal, the reflected radiance L_o is still computed in the same way:

$$\mathbf{L}_o = \rho \cdot (\mathbf{T} \times \mathbf{L}) = \rho \cdot (\mathbf{T}^s \times \mathbf{L}^s) \tag{4}$$

Recall that $\mathbf{L}^s$ represents the *dynamic* direct lighting that is projected onto the spectral basis set s at run-time. We could further project the columns of $\mathbf{T}^s$ onto the spectral basis, resulting in a new matrix encoding basis-to-basis indirect transfer. However, doing so would eliminate our ability to dynamically change the per-vertex diffuse albedos on the fly.

Multi-bounce diffuse interreflections: Multiple interreflection bounces are simulated using a straightforward gathering approach from radiosity literature:

$$\mathbf{L}^{i+1} = \rho \cdot (\mathbf{T}^s \times (\mathbf{L}^i + \mathbf{L})^s)$$

where $\mathbf{L}$ is direct illumination, $\mathbf{L}^i$ is the indirect lighting evaluated at the i-th bounce, with $\mathbf{L}^0$ set to zero. Note that the resulting $\mathbf{L}^i$ from each bounce is added to direct lighting $\mathbf{L}$ at the start of the next bounce, and projected again onto the spectral basis set s. Because our spectral basis set is small, this projection can be computed quickly in real-time.

By computing global illumination in a multi-bounce fashion, we can easily modify the BRDF diffuse albedos at runtime. Note that if the BRDF f_r were known to be fixed at precomputation time, we could combine the multi-bounce transfer into a single matrix T^*, encoding the linear relationship between direct lighting and the total indirect lighting.

Final-bounce glossy interreflection: We include the final bounce of glossy interreflection by using an SH projection of the BRDF. In this case, a general glossy BRDF $f_r(\omega_i, \omega_\mathbf{o})$ is approximated using a linear sum of 16 SH basis h_k:

$$f_r(\omega_i, \omega_o) \approx \sum_{k=1}^{K} \rho_k(\omega_o) \cdot h_k(\omega_i) \tag{5}$$

Figure 3: *Example of partitions generated by MeTis [Met].*

This allows us to approximate glossy interreflection by:

$$\begin{aligned} L_o(x_o, \omega_o) &= \int_A L^i(x_i)\, f_r(x_i \to x_o, \omega_o)\, T(x_o, x_i)\, dA(x_i) \\ &\approx \sum_k \rho_k(\omega_o) \int_A L^i(x_i)\, h_k(x_i \to x_o)\, T(x_o, x_i)\, dA(x_i) \\ &= \sum_k \rho_k(\omega_o) \int_A L^i(x_i)\, T_k(x_o, x_i)\, dA(x_i) \\ &= \sum_k \rho_k(\omega_o) \cdot (\mathbf{T}_k^s \times (\mathbf{L}^i)^s) \end{aligned} \tag{6}$$

where $\mathbf{T}_k^s$ represents the precomputed indirect transfer matrix after double projections: first onto the spectral mesh basis (spatial projection), and then onto the SH basis (BRDF projection). $\rho_k(\omega_o)$ is computed on the fly as the user changes the viewpoint; this is done with a simple texture access using modern graphics hardware. Note, however, the SH projection (Eq. 5) cannot currently be computed in real-time, we therefore precompute and store the projections of several glossy BRDFs, and switch between them on the fly.

The use of only low-order SH coefficient limits us to low-frequency reflection effects, which means high-frequency BRDFs such as shiny metals may not be approximated with sufficient accuracy. Note that we are only limited by this approximation for the glossy *interreflection* component – our direct lighting component uses programmable shaders to compute reflections from analytic BRDFs. Therefore, our overall rendering still demonstrates convincing shiny reflections with subtle glossy interreflection effects (please see the accompanying video). Alternatively, we could use other BRDF bases such as wavelets [WNLH06] or RBFs [GKMD06], which are proven to be more accurate for high-frequency effects.

3.3. Implementation Notes

Constructing the spectral basis: We compute a spectral basis for our models using the algorithm described by [KG00]. Similar to them, we use a static algorithm MeTis [Met] to partition our models into submeshes. Each submesh consists of 500 $\sim$ 1000 vertices depending on the

© The Eurographics Association 2007.

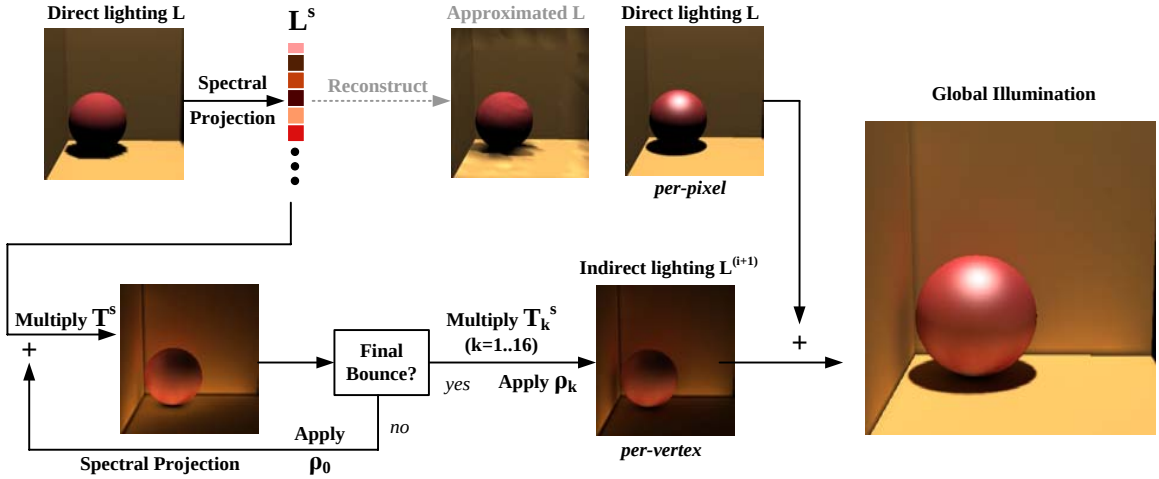


Figure 4: *Overview of our relighting algorithm. Dynamic direct lighting is first computed at each vertex and projected onto the spectral mesh basis. The dotted line shows the approximated direct lighting (for illustrative purpose). The projected lighting is then multiplied with the precomputed diffuse indirect transfer matrix and the per-vertex diffuse albedo. This indirect transfer step is repeated several times to simulate multi-bounce transfer, and eventually the per-vertex SH-BRDF is applied to gather final-bounce glossy interreflections. Finally, a hardware shader computes per-pixel direct shading color, sums it up with the per-vertex indirect color, and displays the global illumination result onto the screen.*

scene complexity. This reduces the computational cost of SVD, and also has the advantage that each object gets localized mesh bases to better capture local illumination changes. If the partition size is too small, a large number of patches will be generated, each having only a few basis functions budgeted. On the other hand, if the partition size is too big, the computational cost and numerical stability (in finding the smallest eigenvalues) will become an issue for the SVD routine. We found $500 \sim 1000$ vertices per patch a good tradeoff between both. In general, the rendering results are insensitive to the particular partition size or scheme we choose.

Figure 3 shows an example of generated partitions. MeTis is optimized such that each partition consists of roughly the same number of vertices. After partitioning, we run the spectral basis construction algorithm on each submesh, keeping roughly equal number of basis functions per patch. We typically generate a total of about 1024 spectral bases for the entire scene, resulting in $20 \sim 40$ bases budgeted for each submesh. Note that we cannot optimize the partition algorithm to preserve illumination discontinuities, as we do not have any information about the dynamic direct lighting at precomputation time.

Because we use model vertices to sample illumination functions, significantly under-tesselated surfaces can cause under-sampling artifacts. We therefore require the user to reasonably tessellate the models to maintain uniform accuracy. Our algorithm is not restricted to any particular scheme for tessellation, and is flexible to any mesh topology.

Precomputing transfer matrices: In precomputation, we evaluate the transfer function T (Eq. 1) and project it onto the spectral basis set:

$$T^s(x_o,\ell) = \int_A T(x_o,x_i)\, s_\ell(x_i)\, dA(x_i) \qquad (7)$$

where $s_\ell(x_i)$ is the ℓ-th spectral basis function. This is equivalent to computing the total irradiance at x_o due to a spatially varying illumination function exactly equal to $s_\ell(x_i)$. Therefore, we can easily implement the precomputation algorithm using an existing Monte Carlo ray tracer [PH04]. To do so, we use each basis $s_\ell(x_i)$ as a spatially varying source to illuminate the scene, then cast rays from x_o to evaluate the integral above. Using linear properties of the basis, we compute the results for all basis functions in a single pass. All our demonstrated scenes are computed within one hour (including the construction of spectral mesh basis) using 128×128 Monte Carlo samples.

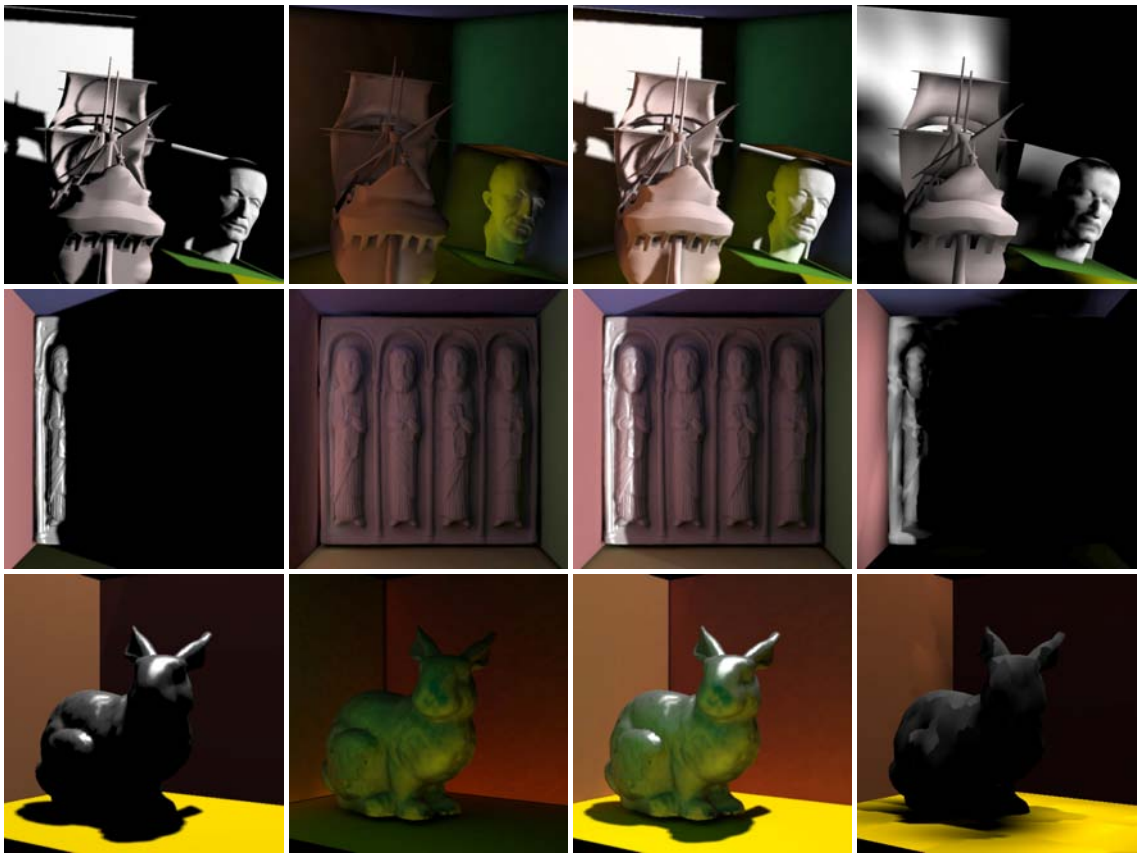
For glossy interreflections, we further include BRDF SH basis h_k into precomputation, thus Eq. 7 becomes:

$$T_k^s(x_o,\ell) = \int_A T(x_o,x_i)\, s_\ell(x_i)\, h_k(x_i \rightarrow x_o)\, dA(x_i) \qquad (8)$$

This results in a higher dimensional transfer matrix, each slice of which corresponds to a different SH basis. We typically use 16 SH functions, which is similar to what has been used previously for low-frequency BRDFs [KSS02].

Quantization: We compress the precomputed transfer data by quantizing the matrix elements to 16 bit integers and

© The Eurographics Association 2007.



(a) Direct lighting (b) Indirect lighting (c) Global illumination (d) Approximated direct light

Figure 5: *Here we show the different illumination components for three example scenes. From left to right, we show per-pixel direct lighting, per-vertex indirect lighting, global illumination, and approximated direct lighting resulting from our basis project. (d) is for illustration only, not actually computed in our pipeline. Note how indirect lighting adds realistic and convincing effects to the rendering. Also note the final bounce glossy interreflection effects in the bunny scene.*

truncating all elements that are quantized to zero. This results in a sparse representation of $\mathbf{T}$. This approach is especially effective for complex scenes where indirect lighting is dominated by local transfer effects. Our experimental results show that on average we keep only $60 \sim 100$ elements per vertex, and the precomputed data size is roughly 10 MB for a diffuse 30,000-vertex model. This is crucial for the rendering algorithm to perform relighting at real-time rates. The precomputed data size for glossy objects is 16 times larger, but the relighting can still be performed at interactive rates.

Rendering: Our rendering algorithm is similar to other PRT systems. Figure 4 shows an overview of the rendering pipeline. The direct lighting step can use any GPU-based technique such as shadow mapping or shadow volumes. The projection of direct lighting onto the spectral basis is computed efficiently by taking advantage of the fact that each spectral basis is localized within its submesh. For glossy objects, the intermediate interreflection passes only multiply the first matrix slice $\mathbf{T}_0^s$ and the first BRDF component ρ_0, corresponding to diffuse albedo, while the final-bounce pass applies all 16 matrix slices and BRDF components.

4. Results and Discussion

Our results are reported on an Intel Core 2 Duo 1.83GHz computer with 2GB memory and an NVIDIA 8800 GTS graphics card. All programs are compiled using Intel Compiler v9.1. The precomputation is done using a modified version of the `PBRT` ray tracer [PH04], which uses only one processor. The rendering has the majority of its computation spent on matrix-vector multiplication. This is easily parallelizable and therefore we used both CPU cores to accelerate the rendering. Table 1 lists the precomputation and relighting performance for several test scenes.

Precomputation: As seen in Table 1, all test scenes are pre-

© The Eurographics Association 2007.

(a) Our method, ∼25 fps (b) Ray tracing reference, 2.5 hours (c) Approximation error

Figure 6: *A comparison between our method and a ray traced reference. This is a largely closed scene with a small open window on the front wall. Our rendering qualitatively matches the reference but is significantly faster. Subtle differences are noticeable on the ground floor close to the viewer. The blurry effects in our rendering are due to the soft shadow implementation and the per-vertex shading of indirect lighting. The graph on the right plots the L^2 errors of the indirect lighting component compared with the reference. The errors are measured at vertices and with a varying number of spectral basis functions from 32 to 2048. The mean indirect lighting luminance of the scene is 0.1. Through experiments, we found that 1024 basis functions deliver both good rendering accuracy and desirable performance, therefore become our default choice.*

computed under an hour. For all diffuse scenes, the sparse matrix size is under 10 MB. This small footprint makes our technique particularly suitable for real-time applications such as computer games. For scenes that account for glossy BRDFs, the matrix size is substantially bigger due to the inclusion of BRDF bases; nonetheless, these models still perform at interactive relighting rates.

Rendering: Rendering performance is also listed in Table 1. Because our precomputation does not fix the specific BRDFs to be used at run-time, we are able to change materials dynamically on the fly. The multi-bounce diffuse transfer only accounts for diffuse albedo changes, while the final-bounce glossy transfer accounts for low-frequency glossy BRDF changes. As dynamic BRDF projection is costly to compute, we precompute and store several glossy BRDFs and switch between them on the fly, as shown in Figure 7.

Accuracy: Figure 6 shows a comparison of our rendering result vs. a reference image generated using a standard Monte Carlo ray tracer. This scene (31,000 vertices) is largely closed with a small open window on the front wall, therefore the illumination result is dominated by indirect lighting effects. Note how our rendering qualitatively matches the reference. The major difference is that ray tracing generates per-pixel shading, while our technique generates per-vertex shading for the indirect lighting component, combined with per-pixel shading for the direct lighting component. We also use a soft-edge shadow mapping implementation to reduce shadow aliasing artifacts, therefore, our rendering appears slightly blurry overall. In addition, subtle illumination differences are noticeable on the ground floor close to the viewer. These artifacts are caused by our spectral basis approximation.

On the right of Figure 6, we plot the L^2 errors of the indirect lighting component using our method compared to the reference. This indicates the error in the perceived color that is directly visualized. The errors are measured and averaged at scene vertices (we have slightly modified the ray tracer to produce color per vertex). The experiments were done for a varying number of spectral basis functions from 32 to 2048. At 1024 basis, the error is quite small relative to the scene's average indirect lighting luminance, which is 0.1. Note that the accuracy improvement of using 2048 basis functions vs. 1024 basis functions is small. This supports our findings that 1024 basis functions deliver both good rendering results and performance in the experiments.

In Figure 5 we show side by side comparisons of different illumination components. In each example, we show the direct lighting, the simulated indirect lighting using our method, and the global illumination result. We also show the approximated direct lighting resulting from our basis projection. Note that this approximated lighting is not directly visualized, therefore, even when there are non-smooth changes due to mesh partitioning, the indirect lighting result is still nearly artifact free due to the smooth nature of indirect transfer. In all examples, the scenes look unrealistically dark with only direct lighting, while indirect lighting adds important color bleeding effects, making the overall appearance realistic and convincing. Also note the final bounce glossy interreflection effects in the bunny scene.

5. Conclusions and Future Work

We have presented a method for interactive rendering of indirect lighting effects using a spectral mesh basis and PRT.

© The Eurographics Association 2007.

	Statues	Horse	Bunny (Diffuse)	Plank	Sphere	Bunny (Glossy)
Vert. count	35 K	25 K	31 K	7 K	4 K	28 K
Tri. count	71 K	50 K	61 K	14 K	8 K	54 K
Partition size	1000	1000	1000	500	500	1000
BRDF SH basis	1	1	1	1	16	16
Prep. time	30 min	32 min	38 min	12 min	4 min	54 min
Basis storage	3 M	3 M	3 M	3 M	3 M	3 M
Coeffs/Vertex	63	62	66	85	1400	1645
Matrix storage	8.7 M	6.0 M	8.0 M	2.5 M	24 M	180 M
Direct Fps	120	180	130	280	300	150
Global (1) Fps	25	35	26	74	30	6.5
Global (3) Fps	18	20	17	45	28	6.0

Table 1: *Precomputation and relighting performance for our test scenes. The first two rows show the total number of vertices and faces in each scene. next two rows show the partition size and the number of BRDF SH basis. All test scenes took less then one hour to precompute. The storage size for the resulting spectral basis data and transfer matrix data are reported in the middle rows. In the last four rows, we report the rendering frame rates (fps) for direct lighting component, the overall (direct + indirect) relighting performance with 1-bounce and 3-bounce indirect lighting respectively.*

Figure 7: *Our system supports dynamic manipulation of diffuse albedos and switching of glossy BRDFs.*

Our system incorporates an arbitrary local lighting model into a real-time high quality global illumination environment. This ability is especially attractive for interactive applications where flexible local lighting is important, such as games or lighting design.

We anticipate several directions for future work. First, to address the limitation of low-frequency glossy interreflections, we plan to experiment with other BRDF bases such as wavelets or RBFs to better capture all-frequency reflection effects. Second, we are extending our technique to handle textured models by representing the spectral basis set as texture functions over the surface geometry. This can better suit our method to applications that require a very low polygon count. Third, the requirement that the scene be fairly evenly tesselated is quite restrictive, and we plan to address this limitation by exploring adaptive mesh partitioning schemes. Finally, our approach should easily extend to rigid body dynamic scenes using a neighborhood sampling approach similar to Shadow Fields [ZHL*05] or Ambient Occlusion Fields [KL05].

Acknowledgements The authors would like to thank Sridhar Mahadevan and Ewen Cheslack-Postava for discussions about the spectral basis, and Oskar Akerlund and Mattias Unger for creating several of the scene models.

References

[AKDS04] ANNEN T., KAUTZ J., DURAND F., SEIDEL H.-P.: Spherical harmonic gradients for mid-range illumination. In *Proceedings of the Eurographics Symposium on Rendering* (2004), pp. 331–336.

[Ale01] ALEXA M.: Local control for mesh morphing. In *Proceedings of the International Conference on Shape Modeling & Applications* (2001), p. 209.

[BAOR06] BEN-ARTZI A., OVERBECK R., RAMAMOORTHI R.: Real-time BRDF editing in complex lighting. *ACM Transactions on Graphics 25*, 3 (2006), 945–954.

[BWG03] BALA K., WALTER B., GREENBERG D. P.: Combining edges and points for interactive high-quality rendering. *ACM Transactions on Graphics 22*, 3 (2003), 631–640.

[CHL04] COOMBE G., HARRIS M. J., LASTRA A.: Radiosity on graphics hardware. In *GI '04: Proceedings of the 2004 conference on Graphics interface* (2004), pp. 161–168.

© The Eurographics Association 2007.

[CWH93] COHEN M. F., WALLACE J., HANRAHAN P.: *Radiosity and realistic image synthesis*. Academic Press Professional, Inc., San Diego, CA, USA, 1993.

[DBG*06] DONG S., BREMER P.-T., GARLAND M., PASCUCCI V., HART J. C.: Spectral surface quadrangulation. *ACM Transactions on Graphics 25*, 3 (2006), 1057–1066.

[DS06] DACHSBACHER C., STAMMINGER M.: Splatting indirect illumination. In *SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games* (2006), pp. 93–100.

[GKMD06] GREEN P., KAUTZ J., MATUSIK W., DURAND F.: View-dependent precomputed light transport using nonlinear gaussian function approximations. In *ACM Symposium on Interactive 3D graphics* (2006), pp. 7–14.

[GSCH93] GORTLER S. J., SCHRÖDER P., COHEN M. F., HANRAHAN P.: Wavelet radiosity. In *Proceedings of SIGGRAPH '93* (1993), pp. 221–230.

[Hec92] HECKBERT P. S.: Radiosity in flatland. *Computer Graphics Forum (EUROGRAPHICS '92 Proceedings) 11*, 3 (1992), 181–192.

[HPB06] HAŠAN M., PELLACINI F., BALA K.: Direct-to-indirect transfer for cinematic relighting. *ACM Transactions on Graphics 25*, 3 (2006), 1089–1097.

[Jen01] JENSEN H. W.: *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., Natick, MA, USA, 2001.

[KAMJ05] KRISTENSEN A. W., AKENINE-MÖLLER T., JENSEN H. W.: Precomputed local radiance transfer for real-time lighting design. *ACM Transactions on Graphics 24*, 3 (2005), 1208–1215.

[Kel97] KELLER A.: Instant radiosity. In *Proceedings of SIGGRAPH '97* (1997), pp. 49–56.

[KG00] KARNI Z., GOTSMAN C.: Spectral compression of mesh geometry. In *Proceedings of SIGGRAPH 2000* (2000), pp. 279–286.

[KL05] KONTKANEN J., LAINE S.: Ambient occlusion fields. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games* (2005), pp. 41–48.

[KSS02] KAUTZ J., SLOAN P.-P., SNYDER J.: Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In *Proceedings of the 13th Eurographics Rendering Workshop* (2002), pp. 291–296.

[KTHS06] KONTKANEN J., TURQUIN E., HOLZSCHUCH N., SILLION F.: Wavelet radiance transport for interactive indirect lighting. In *Proceedings of Eurographics Symposium on Rendering* (2006).

[LSK*07] LAINE S., SARANSAARI H., KONTKANEN J., LEHTINEN J., AILA T.: Incremental instant radiosity for real-time indirect illumination. In *Proceedings of Eurographics Symposium on Rendering 2007* (2007).

[LSSS04] LIU X., SLOAN P.-P., SHUM H.-Y., SNYDER J.: All-frequency precomputed radiance transfer for glossy objects. In *Proceedings of the 15th Eurographics Symposium on Rendering* (2004), pp. 337–344.

[Met] http://www-users.cs.umn.edu/~karypis/metis/.

[NPG05] NIJASURE M., PATTANAIK S. N., GOEL V.: Real-time global illumination on gpus. *Journal of Graphics Tools 10*, 2 (2005), 55–71.

[NRH03] NG R., RAMAMOORTHI R., HANRAHAN P.: All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics 22*, 3 (2003), 376–381.

[NRH04] NG R., RAMAMOORTHI R., HANRAHAN P.: Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics 23*, 3 (2004), 477–487.

[PDC*03] PURCELL T. J., DONNER C., CAMMARANO M., JENSEN H. W., HANRAHAN P.: Photon mapping on programmable graphics hardware. In *Proceedings of Graphics Hardware 2003* (2003), pp. 41–50.

[PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., 2004.

[RSH05] RESHETOV A., SOUPIKOV A., HURLEY J.: Multi-level ray tracing algorithm. *ACM Transactions on Graphics 24*, 3 (2005), 1176–1185.

[RWS*06] REN Z., WANG R., SNYDER J., ZHOU K., LIU X., SUN B., SLOAN P.-P., BAO H., GUO B.: Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Transactions on Graphics 25*, 3 (2006), 977–986.

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *ACM Transactions on Graphics* (2002), vol. 21, pp. 527–536.

[SLS05] SLOAN P.-P., LUNA B., SNYDER J.: Local, deformable precomputed radiance transfer. *ACM Transactions on Graphics 24*, 3 (2005), 1216–1224.

[TM93] TROUTMAN R., MAX N. L.: Radiosity algorithms using higher order finite element methods. In *Proceedings of SIGGRAPH '93* (1993), pp. 209–212.

[TPWG02] TOLE P., PELLACINI F., WALTER B., GREENBERG D. P.: Interactive global illumination in dynamic scenes. In *ACM Transactions on Graphics* (2002), vol. 21, pp. 537–546.

[WBS07] WALD I., BOULOS S., SHIRLEY P.: Ray tracing deformable scenes using dynamic bounding volume hierarchies. *ACM Transactions on Graphics 26*, 1 (2007), 6.

[WDP99] WALTER B., DRETTAKIS G., PARKER S.: Interactive rendering using the render cache. In *Proceedings of the 10th Eurographics Rendering Workshop* (1999), pp. 235–246.

[WIK*06] WALD I., IZE T., KENSLER A., KNOLL A., PARKER S. G.: Ray tracing animated scenes using coherent grid traversal. *ACM Transactions on Graphics 25*, 3 (2006), 485–493.

[WK05] WU J., KOBBELT L.: Efficient spectral wtaermarking of large meshes with orthogonal basis functions. In *Proceedings of Pacific Graphics 2005* (2005), pp. 848–857.

[WNLH06] WANG R., NG R., LUEBKE D., HUMPHREYS G.: Efficient wavelet rotation for environment map rendering. In *Proceedings of the 17th Eurographics Symposium on Rendering* (2006).

[WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. In *Proceedings of SIGGRAPH '88* (1988), pp. 85–92.

[WTL04] WANG R., TRAN J., LUEBKE D.: All-frequency relighting of non-diffuse objects using separable BRDF approximation. In *Proceedings of the 15th Eurographics Symposium on Rendering* (2004), pp. 345–354.

[Zat93] ZATZ H. R.: Galerkin radiosity: a higher order solution method for global illumination. In *Proceedings of SIGGRAPH '93* (1993), pp. 213–220.

[ZHL*05] ZHOU K., HU Y., LIN S., GUO B., SHUM H.-Y.: Precomputed shadow fields for dynamic scenes. *ACM Transactions on Graphics 24*, 3 (2005), 1196–1201.

© The Eurographics Association 2007.