DOI: 10.1111/j.1467-8659.2011.02001.x
COMPUTER GRAPHICS *forum*
Volume 30 (2011), number 8 pp. 2270–2278

Pre-computed Gathering of Multi-Bounce Glossy Reflections

Jurgen Laurijssen[1], Rui Wang[2], Ares Lagae[1] and Philip Dutré[1]

[1]Katholieke Universiteit Leuven, Belgium
{jurgen.laurijssen, ares.lagae, philip.dutre}@cs.kuleuven.be
[2]University of Massachussetts at Amherst, MA, USA
ruiwang@cs.umass.edu

Abstract

Recent work in interactive global illumination addresses diffuse and moderately glossy indirect lighting effects, but high-frequency effects such as multi-bounce reflections on highly glossy surfaces are often ignored. Accurately simulating such effects is important to convey the realistic appearance of materials such as chrome and shiny metal. In this paper, we present an efficient method for visualizing multi-bounce glossy reflections at interactive rates under environment lighting. Our main contribution is a pre-computation–based method which efficiently gathers subsequent highly glossy reflection passes modelled with a non-linear transfer function representation based on the von Mises–Fisher distribution. We show that our gathering method is superior to scattered sampling. To exploit the sparsity of the pre-computed data, we apply perfect spatial hashing. As a result, we are able to visualize multi-bounce glossy reflections at interactive rates at a low pre-computation cost.

Keywords: global Illumination, real-time rendering

ACM CCS: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture.

1. Introduction

Interactive global illumination has become a popular topic in recent graphics research. Today, rendering systems simulate multi-bounce reflections on diffuse or perfectly specular surfaces at interactive rates. However, it remains a challenge to efficiently simulate global illumination effects resulting from highly glossy materials. As a result, multi-bounce, high-frequency glossy reflections are typically missing in games and similar interactive applications. Although these reflection effects are typically more subtle than diffuse inter-reflections, they are very important to convey the realistic appearance of materials such as chrome and shiny metal. Therefore, our goal in this paper is to simulate these effects interactively. In particular, we are interested in rendering multi-bounce glossy reflections in scenes illuminated by a distant environment map, such as in Figure 1 .

If a scene is static, we can apply pre-computed radiance transfer (PRT) [SKS02] to sample the linear relationship between the light sources and the radiance reflected off a surface point in any view direction. This relationship can be pre-computed as light transport data and compressed with a suitable basis, allowing for relighting of the scene on the fly from any arbitrary viewpoint. Typically, PRT methods are very efficient at pre-computing indirect lighting through gathering, which is an iterative procedure that samples a single reflection pass during each iteration, exploiting the data computed during the previous iteration. Thus, because of the reuse of samples gathering increases the speed of the pre-computation phase significantly.

Indirect lighting samples can be projected onto an orthonormal basis. Thus, a standard linear basis allows gathering. However, standard linear basis functions are not suitable for approximating highly glossy materials. A non-linear mixture of functions such as spherical Gaussians [GKMD06] or the von Mises–Fisher (vMF) distribution [HSRG07] has been shown to fit this purpose better. Unfortunately without orthonormal basis, a non-linear transfer function

(a) (b) (c)

Figure 1: *We present a pre-computation based method, which efficiently computes multi-bounce highly glossy reflections under environment lighting. Notice the reflections of the glasses on the plane, which preserve their glossy appearance. This example renders at 6.19 frames per second. (a) Direct lighting, (b) first bounce indirect lighting and (c) second bounce.*

representation does not support gathering. Thus, current approaches [GKMD06] rely on scattering samples through the scene with bidirectional reflection distribution function (BRDF) importance sampling to compute multi-bounce glossy reflections, which is less efficient.

In this paper, we present for the first time a PRT method which both computes a non-linear transfer function and supports gathering. The key element of our approach is the use of the vMF distribution [MJ00] and vMF merging [LWDB10]. During each iteration, vMF distributions modelling contributions from the previous reflection pass are gathered and merged into a small number of vMFs. We also propose to use perfect spatial hashing [LH06] to reduce the size of the pre-computed data sets independently of the number of vMFs used to model the transfer function. As a result, we are able to pre-compute multi-bounce highly glossy reflections under environment lighting very efficiently.

2. Previous Work

Global illumination based on the rendering equation [Kaj86] has been a central challenge in computer graphics research for a long time. An overview of modern techniques can be found in standard textbooks such as [DBB06]. A fundamental difficulty in solving global illumination is the computation of indirect lighting, where many surface points can contribute illumination to a shading point. Common techniques such as irradiance caching [WRC88] and photon mapping [Jen01] are used to simulate accurate multi-bounce indirect lighting, but are too expensive for interactive applications. In addition, their efficiency can drop significantly for very glossy inter-reflections or self-reflections, due to the highly view-dependent nature of such effects.

To achieve interactive global illumination, PRT [SKS02] is a well-known technique that samples and compresses light transport functions for a static scene and exploits the data at run-time to permit relighting under arbitrary environment lighting. Early work in PRT [SKS02, KSS02, LK03] uses a spherical harmonics (SH) basis for approximating pre-computed data, which limits the materials to diffuse or moderately glossy, and only supports low-frequency lighting. Using a wavelet basis instead, Ng *et al.* [NRH03] achieves all-frequency lighting effects, but only for a fixed camera view. Later, [WTL04, LSSS04] proposed using separable BRDF decomposition [KM99] to allow for view-dependent reflections. Wang *et al.* [WTL06] extended this approach to include multi-bounce indirect lighting by a technique that efficiently accumulates transport vectors. However, these methods essentially approximate the BRDF onto a low-frequency basis similar to SH, hence requires many terms in case of highly glossy materials [MTR08]. Another approach is to use piecewise constant basis to directly sample BRDFs at run-time [CPWAP08], but it too has trouble handling glossy BRDFs due to aliasing artefacts.

In [GKMD06] and [TS06], light transport functions are approximated with a mixture of spherical Gaussian (SGs) functions. Unlike SH, the interpolation of spherical Gaussians is non-linear, therefore it is more appropriate for modelling highly glossy BRDFs. Our method builds upon the vMF distribution proposed by Han *et al.* [HSRG07]. It is similar to SGs but with a proper normalization factor. This representation is also studied in [WRG*09] for PRT-based rendering, but with direct illumination only. PRT methods based on a non-linear representation do not allow gathering of indirect lighting. In contrast, we propose to use vMF merging [LWDB10] for gathering indirect lighting represented with an vMF distribution [MJ00].

Recently, Huang *et al.* [HR10] propose an adaptive approach for computing the transfer function, both spatially and angularly. The input for their method is a uniform, very dense mesh. Their strategy is to pick a small set of vertices from this mesh and perform a full PRT computation at these positions. Additional so-called 'dense' vertices are added to this set iteratively. Computation of the transfer function in the remainder of the vertices is focused in directions where the transfer function differs significantly from the solution for the dense set.

PRT techniques also have been extended to allow interactive editing of glossy BRDFs with global illumination for a fixed view direction [BAOR06, BAEDR08, NKLN10] or with a dynamic camera [SZC*07].

Another area of related work is a series of techniques built upon gathering illumination from many lights [WFA*05, HPB07]. These methods approximate the source of indirect lighting as many point lights called virtual point lights (VPLs). Then, the contribution from each VPL is gathered at every shading point to approximate indirect lighting. One main advantage of this approach is that it is fast and suitable for dynamic scenes, as no pre-computation is required. Combined with modern GPUs and hierarchical VPL evaluation, it can achieve impressive performance with high-rendering quality, as demonstrated by Ritchel *et al.* [REG*09].

However, a common drawback of these methods is that they are restricted to diffuse to moderately glossy materials. One reason is that the VPLs are typically assumed diffuse; another reason is to reduce computation, the number of VPLs must be limited, which directly limits the sampling rate and high frequencies of illumination. Recent work [HKWB09, DKH*10] extends the many lights approach to include glossy reflections. Although efficient, these methods run offline, taking several minutes to render a frame. In this paper, we aim to achieve interactive performance by exploiting pre-computed data.

Recently, Wang *et al.* [WWZ*09] combined GPU-based photon mapping with adaptive irradiance sampling and photon tree cut to achieve interactive global illumination. This method is efficient for multi-bounce reflections on primarily diffuse or perfectly specular materials. Simulating medium to high-frequency range reflections remains a challenge.

2.1. The von Mises–Fisher distribution

2.1.1. *Definitions*

The vMF distribution [MJ00] is a spherical probability density function that describes the distribution of directions centred around a mean direction:

$$\gamma(\mathbf{s}) = c(\kappa)\,\mathrm{e}^{\kappa(\mu\cdot\mathbf{s})}, \quad (1)$$

where μ is the normalized mean direction of the distribution ($\|\mu\| = 1$), and κ describes the angular width (concentration) of the distribution. Higher values of κ correspond to more concentrated distributions. $c(\kappa) = \frac{\kappa}{4\pi\sinh(\kappa)}$ is a normalization constant for the spherical integration to evaluate to 1. Note that, except for the normalization constant, the vMF distribution is the same as a spherical Gaussian.

Suppose that $\mathbf{s}_i$, $i \in \{1, \ldots, M\}$ is a set of directions resulting from a stochastic process and that it can be modelled as a single vMF distribution. We can estimate the parameters μ and κ of this distribution by computing the *unnormalized* average direction $\mathbf{r}$ of the set $\mathbf{s}_i$:

$$\mathbf{r} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{s}_i. \quad (2)$$

Intuitively, $\mathbf{r}$ points in the mean direction of the lobe, and the magnitude $\|\mathbf{r}\|$ indicates the angular width of the lobe. For example, a uniform distribution of directions on the sphere results in $\|\mathbf{r}\| = 0$, while an extremely concentrated distribution results in $\|\mathbf{r}\| \approx 1$. Mathematically, an unbiased estimator for μ and κ based on $\mathbf{r}$ is given by Banerjee *et al.* [BDGS05] as

$$\langle\mu\rangle = \frac{\mathbf{r}}{\|\mathbf{r}\|},\ \langle\kappa\rangle \approx \frac{3\|\mathbf{r}\| - \|\mathbf{r}\|^3}{1 - \|\mathbf{r}\|^2}. \quad (3)$$

2.1.2. *vMF merging*

Laurijssen *et al.* [LWDB10] propose a simple method that computes, given a large set of vMF distributions, the parameters of a much smaller set of vMFs that effectively approximates the overall input distribution. Similar vMFs are combined into a single vMF by estimating its mean unnormalized direction $\mathbf{r}$. Clustering on the mean directions μ is used to find subsets of similar vMFs. Their method applies vMF merging to efficiently estimate indirect highlights created by two reflections off highly glossy surfaces. No pre-computation is required.

In contrast, our method pre-computes transfer functions represented by vMF distributions and uses vMF merging to be able to gather indirect transfer components efficiently.

Suppose a large set of similar vMF distributions γ_i, $i \in 1, \ldots, M$ is given. In this context, similar means that the overall distribution represented by this set can be described by a single vMF distribution γ_m. The individual vMF parameters κ_i and μ_i are not necessarily identical. To estimate the parameters κ_m and μ_m of the overall distribution, it suffices to compute its mean unnormalized direction $\mathbf{r}_m$ given $\mathbf{r}_i$:

$$\mathbf{r}_m = \frac{1}{M}\sum_{i=1}^{M}\mathbf{r}_i \quad (4)$$

and apply Equation (3)s. In our application, we store each vMF distribution by its mean unnormalized direction $\mathbf{r}$ to enable fast gathering.

3. Overview

To simulate multi-bounce glossy reflections under environment lighting, we choose a pre-computation-based approach. We assume isotropic BRDFs and use the vMF distribution as a model: For each of a fixed set of viewing elevation angles θ_o we store an vMF, which approximates the corresponding BRDF slice, and an associated colour α which equals the hemispherical integral of the slice [GKD07].

In the absence of blocking geometry, our transfer function representation is equivalent to this BRDF representation. However, because of blockers some transfer functions contain additional contributions from subsequent reflection passes, also stored with one or more vMFs with associated colour (Section 4).

We initiate our method by sampling visibility across the scene's surface and store transfer functions containing only direct lighting contributions. We proceed iteratively by gathering and merging contributions from the previous reflection pass (Section 5).

As stated before in convex regions of the scene, our transfer function is identical to the BRDF in that position. Hence it need not be stored. We exploit this sparsity by using perfect spatial hashing (Section 6).

4. A vMF Transfer Function Representation

All light in the scene ultimately originates from the distant lighting environment, after a series of reflections. The relationship between the lighting environment $L_e(\mathbf{s})$ defined in the sphere of directions $\mathbf{S}$, and radiance reflected off a surface point $\mathbf{p}$ in direction $\mathbf{s}_o$ is described by the transfer function $\mathbf{T}(\mathbf{p}, \mathbf{s}, \mathbf{s}_o)$ [SKS02]:

$$L_o(\mathbf{p}, \mathbf{s}_o) = \int_{\mathbf{S}} \mathbf{T}(\mathbf{p}, \mathbf{s}, \mathbf{s}_o)\, L_e(\mathbf{s})\, d\omega_{\mathbf{s}} \tag{5}$$

We pre-compute a slice of this transfer function for a discrete set of surface sample points $\mathbf{p}$ and view directions $\mathbf{s}_o$, and approximate each such slice $\mathbf{T}_{\mathbf{p},\mathbf{s}_o}(\mathbf{s})$ with a set of N vMF distributions:

$$\mathbf{T}_{\mathbf{p},\mathbf{s}_o}(\mathbf{s}) \approx \sum_{i=1}^{N} \alpha_i\, \gamma_i(\mathbf{s}). \tag{6}$$

We choose the points $\mathbf{p}$ to be the vertices of the scene and make sure that the vertex density in concave areas is sufficient to show highly glossy reflections (Section 6). We sample the directions $\mathbf{s}_o$ defined as the vertices of a geodesic grid computed by icosahedral subdivision. These directions are defined in a global frame. We can interpolate $\mathbf{T}_{\mathbf{p},\mathbf{s}_o}(\mathbf{s})$ both spatially and angularly by interpolating the parameters of the set [GKMD06]:

$$\tilde{\mu} = \frac{T(\mu)}{\|T(\mu)\|}, \qquad \tilde{\kappa} = T(\kappa), \qquad \tilde{\alpha} = T(\alpha). \tag{7}$$

Our pre-computation algorithm proceeds iteratively. First it computes all direct contributions, and then it gathers indirect contributions from the previous iteration which it obtains through vMF merging. We store I vMFs for each of the R reflection passes. Thus, for each direction at each surface point, we store $N = IR$ lobes. Usually, $I = 2$ and $R = 3$, unless stated otherwise.

5. Pre-computed Gathering of Glossy Reflections

We initialize pre-computation by approximating each BRDF slice $\tilde{\rho}_{\theta_o}(\mathbf{s})$ in the scene with an vMF distribution. Based on this approximation, we compute the direct lighting contributions of the transfer function. Then we proceed iteratively by gathering indirect lighting from the previous iteration. Finally, small or zero vMF distributions can be discarded. In Section 6, we describe a simple method of storing the resulting sparse data sets.

5.1. Direct lighting

For each point $\mathbf{p}$ and viewing direction $\mathbf{s}_o$, we compute the elevation angle with respect to the normal $\mathbf{n}_\mathbf{p}$ to find the vMF approximating the corresponding BRDF slice $\gamma_{\mathbf{p},\mathbf{s}_o}$ with $\alpha_{\mathbf{p},\mathbf{s}_o}$. We then generate reflection rays by importance sampling this lobe. This can be performed using the following identities:

$$\theta = \frac{\arccos(\ln(1-u)+\kappa)}{\kappa}, \qquad \phi = 2\pi v, \tag{8}$$

where u and v are two uniform random variables, and θ, ϕ are the spherical angles in the coordinate system centred around the lobe centre. The direct transfer function component $\mathbf{T}^0_{\mathbf{p},\mathbf{s}_o}(\mathbf{s})$ is equal to the BRDF multiplied by the binary visibility function V:

$$\mathbf{T}^0_{\mathbf{p},\mathbf{s}_o}(\mathbf{s}) \approx \alpha_{\mathbf{p},\mathbf{s}_o}\gamma_{\mathbf{p},\mathbf{s}_o}(\mathbf{s})(1 - V_{\mathbf{p},\mathbf{s}_o}(\mathbf{s})). \tag{9}$$

To compute $\mathbf{T}^0_{\mathbf{p},\mathbf{s}_o}(\mathbf{s})$, we only consider rays that intersect with the environment, thus discard those that are occluded by the scene. The directions of these rays are then used to find the mean unnormalized direction $\mathbf{r}^0_{\mathbf{p},\mathbf{s}_o}$ (Equation 2). The associated colour $\alpha^0_{\mathbf{p},\mathbf{s}_o}$ is equal to the hemispherical integral of $\mathbf{T}^0_{\mathbf{p},\mathbf{s}_o}(\mathbf{s})$, which is estimated as the ratio of unoccluded rays to the total number of rays, multiplied by $\alpha_{\mathbf{p},\mathbf{s}_o}$.

We have found during our experiments that for highly glossy materials a single vMF lobe is usually sufficient to approximate a BRDF slice appropriately. This is also the case for transfer slices $\mathbf{T}^0_{\mathbf{p},\mathbf{s}_o}(\mathbf{s})$ due to their narrow support.

5.2. Indirect lighting

Following the direct lighting pass, each surface point in the scene that receives direct illumination will have non-zero vMF distributions representing $\mathbf{T}^0_{\mathbf{p},\mathbf{s}_o}(\mathbf{s})$. Now we use multi-pass gathering to accumulate the direct transfer function o obtain indirect lighting contributions. To do so, we again generate reflection rays at each surface point p and direction $\mathbf{s}_o$ during each iteration j. If the ray intersects a point in the scene, at the intersection point we can then obtain a set of existing vMF distributions by interpolating the transfer function using Equation (7).

These vMF distributions $\gamma_i, i = 1, \ldots, M$ are then merged into I new vMFs. Laurijssen *et al.* [LWDB10] show that usually these vMFs are very similar due to the narrow support of $\gamma_{\mathbf{p},\mathbf{s}_o}$. Therefore, we choose not to cluster the input vMFs and to merge them into a single lobe. For most of our scenes we store $I = 2$ lobes. The second lobe is used to store diffuse components of the transfer function, which are modelled separately. However, including more lobes is straightforward due to the sparse nature of the pre-computed data sets (Section 6), which we demonstrate further on in Section 7.

To merge the input vMF distributions, we perform weighted averaging:

$$\mathbf{r}_j = \frac{\sum_i^M \mathbf{r}_i \bar{\alpha}_i}{\sum_i^M \bar{\alpha}_i} \tag{10}$$

with $\bar{\alpha}_i$ the luminance of α_i. The associated colour α_j is computed as the spherical integral of $\mathbf{T}^j_{\mathbf{p},\mathbf{s}_o}(\mathbf{s})$ and estimated as

$$\alpha_j = \frac{\alpha_{\mathbf{p},\mathbf{s}_o}}{M} \sum_i^M \alpha_i. \tag{11}$$

6. Compact Representation with Spatial Hashing

For rendering glossy inter-reflections, the transfer function should be sampled densely both angularly and spatially. On the other hand, there is a lot of coherence in both the angular ($\mathbf{s}_o$) and spatial ($\mathbf{p}$) sampling dimensions that we should exploit. For example, regions that are largely unoccluded, result in transfer functions that are equivalent to the BRDF approximation, due to the absence of blocking geometry. Moreover, after a few inter-reflections, many lobes will become small therefore can be merged or discarded. Finally, in convex regions of the scene which do not have substantial indirect contributions, a dense spatial sampling is typically not necessary.

6.1. Spatial redundancy

We exploit spatial redundancy by subdividing the scene adaptively using an approach similar to Křivánek *et al*. [KPŽ04]. As a pre-process, we iteratively compute visibility masks for each vertex. These masks are created by tracing shadow rays created by subdividing an icosahedron 7 times (163842 directions). We subdivide a face if it contains at least one edge whose visibility masks differ more than a pre-defined threshold. In practice, this threshold is crossed when more than 0.5% of the respective visibility entries differ. This method creates a high vertex density in regions where lots of inter-reflections are to be expected, and a lower density in convex regions.

In contrast with the approach of Huang *et al*. [HR10], the input of our method is a mesh which is dense in regions where inter-reflections are likely. This mesh is computed as a pre-process. At each of the vertices we perform a full PRT computation. We believe that it is possible to apply many of their ideas to our method to reduce computation further both spatially and angularly.

6.2. Angular redundancy

After pre-computation, there is still significant angular redundancy present in the transfer function data, for example most transfer slices do not differ from the BRDF approximation due to the absence of blockers and do not need to be stored. Thus, we pack the sparse transfer function data into a compact table, and apply perfect spatial hashing [LH06] to maintain efficient random access speed during rendering. We use the row displacement compression algorithm [LD08]. The two-dimensional (2D) version of this algorithm replaces a sparse matrix A by a smaller 1D offset table O and 1D hash table H, such that $A[i][j] = H[O[i] + j]$.

An important consequence of this hashing is that it allows us to add more lobes per reflection pass to improve the quality of the transfer function, without excessive memory costs. Indeed, there are but very few regions where more than 1 lobe might be beneficial, such as when the sampled vMF distributions contain contributions from more than 1 object, or when these distributions are not similar. The hash table ensures that these lobes do not need to be stored for the entire scene.

7. Results and Discussion

7.1. Implementation

We have implemented both our pre-computation and rendering algorithm on an Intel(R) Core(TM)2 Quad Q9450 2.66GHz processor with an NVidia Quadro FX5800 graphics card. Both algorithms run entirely on the GPU using NVidia's OptiX framework [PBD*10]. As a pre-process, the environment maps are pre-filtered and each material in the scene is approximated with vMF distributions for a fixed set of elevation angles [GKD07]. The pre-computation step is divided into several small kernels: tracing shadow rays, interpolating the transfer function, merging vMFs. The renderer traces primary rays from the camera and stores hit records on intersecting the scene. Then N passes are performed to interpolate each lobe of the transfer function both angularly and spatially. Each of the pre-computed lobes is either stored in the hash table, found by a lookup in the BRDF approximation table (for direct lighting), or has $\alpha = 0$. Each interpolated lobe is then used to index a pre-filtered environment map. The result of this lookup is added to the pixel's colour. Each frame is tone mapped as a post-process [RSSF02].

7.2. Performance

In all our scenes, we use Phong specular BRDFs with exponents 32–512. Some of the results are shown in Figure 2. These renderings show very distinct glossy reflections such

Figure 2: *Some results of our method. Notice the reflection of the bird's tail on its head, the indirect lighting in the crinkles of the Stanford dragon and the reflections of the glasses on the glossy plane. Each of these results are rendered with a transfer function that models two bounces of indirect lighting and contains $N = 6$ vMF lobes per slice. Precomputation takes less than 1 h. All renderings run at interactive frame rates.*

as the reflection of the bird's tail on its head, the reflections of the glasses onto the glossy plane or the indirect lighting on the Stanford Dragon. During each pass j, we trace 144 rays for each $\mathbf{T}^j_{\mathbf{p},\mathbf{s}_o}(\mathbf{s})$. Most transfer functions contain $I = 2$ vMFs per reflection pass and we pre-compute 3 reflection passes. We achieve interactive frame rates for every scene tested in our experiments. Refer to Table 1 for performance data. Pre-computation times are all less than 1 h and dependent on the number of vertices and the complexity of the scene. The size of the regions where inter-reflections may occur determines not only the vertex density but also the number of vMFs that need to be stored in the hash table. For scenes with less inter-reflections, less transfer functions need to be interpolated and merged and thus are faster to pre-compute.

7.3. Density of the hash table

Overall, the density of the hash table is low (10–20%), which suggests that for most scenes there are few regions with inter-reflections. An interesting aspect of our implementation shows in the teapot**xx** scenes. The number in the name of the scene is the Phong exponent of the teapot's material. Table 1 shows that the hash table is sparser for higher exponents. The glasses scene is typically the worst case scenario for our method due to the very large regions showing glossy reflections from the glasses on the plane and the glasses on each other. Also, the density is approximately identical given the number of directions, which shows in the glasses example generated with 2562 directions compared to the same scene with 642 directions. Another experiment with the glasses scene which uses $I = 6$ lobes per iteration (1 for diffuse components, 5 for each different material) shows that there is a sub-linear increase in the density of the hash table. This demonstrates how we can improve on the quality of the transfer function without excessive memory costs.

7.4. Memory

Nevertheless, the memory requirements of our method are considerable (typically hundreds of Mb). However, contemporary GPUs such as the NVidia Quadro FX5800 support 4 GB of on-board memory. All our scenes render on the GPU

Table 1: *Performance of our PRT application.*

Scene	N	No. of directions	No. of vertices	Density (%)	Size (Mb)	Pre-computation time (s)	Framerate (fps)
Bird	6	642	25403	12	275.1	1120	6.21
Dragon	6	642	41097	17	652.2	2101	6.02
Glasses	6	642	23491	18	385.5	789	6.19
Glasses	6	2562	23491	17	1510.0	1836	5.85
Glasses	18	642	23491	9	578.9	862	2.79
Teapot32	6	642	24269	18	413.3	819	6.26
Teapot64	6	642	24269	16	352.0	844	6.33
Teapot128	6	642	24269	14	303.0	867	6.36
Teapot256	6	642	24269	12	264.5	881	6.11

Note: The columns list the number of pre-computed directions, the number of vertices of each model, the density of the hashmap, the compressed PRT data size, the pre-computation time and the rendering speed. N is the number of lobes we allocate to represent each transfer function slice. Each image is rendered at 1024×768 resolution.

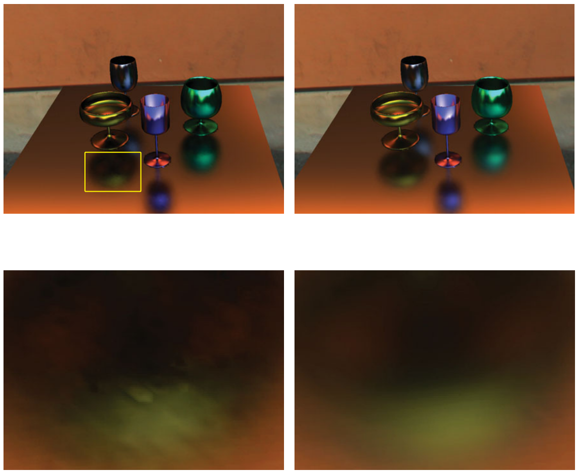
without problems. Also, our experiments using the half floating point format show that it reduces memory consumption by a factor of 2 without significant quality loss. Because of lack of support for the half floating point buffers in OptiX, we have nevertheless used floats for all results in this paper. Note that our storage is lossless; it is difficult to apply additional lossy compression techniques based on a linear decomposition such as PCA or the meshless hierarchical representation proposed by Lehtinen *et al.* [LZT*08] on our data set because even a small error on some of the components of the **r** vectors could result in unpredictable artefacts (Equation 3). In future work, we plan to investigate vector quantization as a more appropriate lossy approach for memory-constrained scenarios.

7.5. Rendering speed

All our results render at interactive frame rates. The speed is primarily dependent on the number of lobes used to approximate the transfer function and on the camera position. The latter is a factor because it determines the number of glossy objects that are visible in the image. Our simple, sub-optimal renderer performs the full transfer function interpolation (angularly and spatially) for each pixel. A better approach would be to interpolate angularly in each vertex, and interpolate this intermediate solution for each pixel.

7.6. Comparison with previous work

The closest to our method is the work of Green *et al.* [GKMD06]. Our vMF transfer function representation is identical to their SG representation. Spherical Gaussians are equivalent to the vMF distribution up to the constant factor $c(\kappa)$. However, in contrast to our gathering approach, they propose to scatter samples through the scene until they are reflected into the distant environment. These samples are then used in a fitting procedure to find lobes describing indirect lighting contributions. To compare both methods, we have implemented a simple scattering approach. Samples that are reflected into the environment are sorted with respect to the number of reflections of the corresponding path. For each such set of samples a lobe is computed. Our simple implementation is far from optimal; thus, we compare results from both methods generated with the same amount of rays traced. Figure 3 shows that our method produces less sampling noise due to the constant reuse of samples during each iteration. Moreover, by scattering samples there is no guarantee that each reflection pass is sampled sufficiently.



(a) Green *et al.* (b) Our method

Figure 3: *Our method compared to Green et al. [GKMD06] for an equal number of rays traced. In contrast to their method which scatters samples through the scene, our gathering method efficiently reuses samples from previous iterations. Our results suffer less from sampling noise.*

7.7. Validation

To test the accuracy of our method, we have compared our results to path traced references (Figure 4). Overall, there are barely visible differences. In some cases, we do notice that our reflections are slightly broader, such as on the teapot's handle. This is due to the fact that the distributions that are merged are not completely axisymmetric.

7.8. Limitations

In this work, we have decided to merge vMF distributions representing indirect lighting contributions into a single vMF,

(a) Reference (b) Our method

Figure 4: *Comparison with path traced reference.*

assuming that these contributions are very similar due to the narrow support of the glossy BRDFs. In situations with highly complex geometry this assumption might not hold. In [LWDB10], we discuss this issue in more detail. In summary, we can apply clustering on the directions of the vMFs to separate dissimilar distributions. However, for highly complex geometry, the clustering may not find all clusters. Fortunately, even in this situation the results still show smooth and visually pleasing glossy reflections [LWDB10].

8. Conclusion and Future Work

We presented a PRT technique, which allows us to gather multi-bounce glossy indirect lighting efficiently and accurately. As correct simulation of glossy inter-reflections is important to convey the appearance of materials such as metals, we believe that our technique is a nice complement to existing PRT methods that only handle low-frequency reflections.

We demonstrated our technique with several examples. The pre-computation takes less than 1 h and the rendering performs at interactive speed, both entirely on the GPU.

For future work, we would like to examine how certain light paths such as caustics could be sampled more efficiently and accurately. These paths are typically easier to compute if the sampling starts from the light sources. Also we would like to investigate how to support fully dynamic lighting instead of using a pre-filtered environment map. Finally, we are interested in applying techniques such as normal map filtering, spatially varying materials and reflection level-of-details to handle multi-bounce reflection effects on more complex surfaces and materials.

Acknowledgements

The authors thank the anonymous reviewers for their thoughtful comments. J.L. is supported by grant CREA/08/017 from the Research Fund K.U.Leuven (Onderzoeksfonds K.U.Leuven). R.W. is supported in part by NSF CAREER grant CCF-0746577. A.L. is a post-doctoral fellow of the Research Foundation - Flanders (FWO).

References

[BAEDR08] Ben-Artzi A., Egan K., Durand F., Ramamoorthi R.: A precomputed polynomial representation for interactive brdf editing with global illumination. *ACM Transactions on Graphics 27* (May 2008), 13:1–13:13.

[BAOR06] Ben-Artzi A., Overbeck R., Ramamoorthi R.: Real-time brdf editing in complex lighting. *ACM Transactions on Graphics 25* (July 2006), 945–954.

[BDGS05] Banerjee A., Dhillon I. S., Ghosh J., Sra S.: Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6 (2005), 1345–1382.

[CPWAP08] Cheslack-Postava E., Wang R., Akerlund O., Pellacini F.: Fast, realistic lighting and material design using nonlinear cut approximation. *ACM Transactions on Graphics 27*, 5 (2008), 1–10.

[DBB06] Dutré P., Bala K., Bekaert P.: *Advanced Global Illumination (2nd edition)*. A. K. Peters, Ltd., Natick, MA, USA, 2006.

[DKH*10] Davidovič T., Křivánek J., Hašan M., Slusallek P., Bala K.: Combining global and local virtual lights for detailed glossy illumination. *ACM Transactions on Graphics 29* (December 2010), 143:1–143:8.

[GKD07] Green P., Kautz J., Durand F.: Efficient reflectance and visibility approximations for environment map rendering. *Computer Graphics Forum 26*, 3 (2007), 495–502.

[GKMD06] Green P., Kautz J., Matusik W., Durand F.: View-dependent precomputed light transport using nonlinear Gaussian function approximations. In *Proceedings of I3D '06* (Redwood City, USA, 2006), pp. 7–14.

[HKWB09] Hašan M., Křivánek J., Walter B., Bala K.: Virtual spherical lights for many-light rendering of glossy scenes. *ACM Transactions on Graphics 28*, 5 (2009), 1–6.

[HPB07] Hašan M., Pellacini F., Bala K.: Matrix row-column sampling for the many-light problem. *ACM Transactions on Graphics 26*, 3 (2007), 26-1–26-10.

[HR10] Huang F.-C., Ramamoorthi R.: Sparsely precomputing the light transport matrix for real-time rendering. *Computer Graphics Forum 29* (2010), 1335–1345.

[HSRG07] Han C., Sun B., Ramamoorthi R., Grinspun E.: Frequency domain normal map filtering. *ACM Transactions on Graphics 26*, 3 (2007), 28-1–28-11.

[Jen01] Jensen H. W.: *Realistic Image Synthesis using Photon Mapping*. A. K. Peters, Ltd., Natick, MA, USA, 2001.

[Kaj86] Kajiya J. T.: The rendering equation. *SIGGRAPH Computer Graphics 20* (August 1986), 143–150.

[KM99] Kautz J., McCool M. D.: Interactive rendering with arbitrary BRDFs using separable approximations. In *Proceedings of the 10th Eurographics Workshop on Rendering* (Granada, Spain, 1999), pp. 281–292.

[KPŽ04] Křivánek J., Pattanaik S., Žára J.: Adaptive mesh subdivision for precomputed radiance transfer. In *SCCG '04: Proceedings of the 20th Spring Conference on*

Computer Graphics (Budmerice, Slovakia, 2004), pp. 106–111.

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

[LD08] LAGAE A., DUTRÉ P.: Compact, fast and robust grids for ray tracing. *Computer Graphics Forum 27*, 4 (2008), 1235–1244.

[LH06] LEFEBVRE S., HOPPE H.: Perfect spatial hashing. *ACM Transactions on Graphics 25* (July 2006), 579–588.

[LK03] LEHTINEN J., KAUTZ J.: Matrix radiance transfer. In *Proceedings of I3D '03* (Monterey, California, 2003), pp. 59–64.

[LSSS04] LIU X., SLOAN P.-P., SHUM H.-Y., SNYDER J.: All-frequency precomputed radiance transfer for glossy objects. In *Proceedings of the Eurographics Symposium on Rendering* (Norrköping, Sweden, 2004). A. Keller and H. W. Jensen (Eds.). The Eurographics Association, Aire-la-Ville, Switzerland, pp. 337–344.

[LWDB10] LAURIJSSEN J., WANG R., DUTRÉ P., BROWN B. J.: Fast rendering and estimation of indirect highlights. *Computer Graphics Forum 29* (2010), 1305–1313.

[LZT*08] LEHTINEN J., ZWICKER M., TURQUIN E., KONTKANEN J., DURAND F., SILLION F. X., AILA T.: A meshless hierarchical representation for light transport. *ACM Transactions on Graphics 27* (August 2008), 37:1–37:9.

[MJ00] MARDIA K. V., JUPP P. E.: *Directional Statistics*. John Wiley and Sons Ltd., Hoboken, New Jersey, 2000.

[MTR08] MAHAJAN D., TSENG Y.-T., RAMAMOORTHI R.: An Analysis of the In-Out BRDF Factorization for View-Dependent Relightings. In *Proceedings of the 10th Eurographics Workshop on Rendering* (Sarajevo, Bosnia and Herzegovina, 2008), pp. 1137–1145.

[NKLN10] NGUYEN C. H., KYUNG M.-H., LEE J.-H., NAM S.-W.: A pca decomposition for real-time brdf editing and relighting with global illumination. *Computer Graphics Forum 29* (2010), 1469–1478.

[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.

[PBD*10] PARKER S. G., BIGLER J., DIETRICH A., FRIEDRICH H., HOBEROCK J., LUEBKE D., MCALLISTER D., MCGUIRE M., MORLEY K., ROBISON A., STICH M.: Optix: a general purpose ray tracing engine. *ACM Transactions on Graphics 29* (July 2010), 66:1–66:13.

[REG*09] RITSCHEL T., ENGELHARDT T., GROSCH T., SEIDEL H.-P., KAUTZ J., DACHSBACHER C.: Micro-rendering for scalable, parallel final gathering. *ACM Transactions on Graphics 28* (December 2009), 132:1–132:8.

[RSSF02] REINHARD E., STARK M., SHIRLEY P., FERWERDA J.: Photographic tone reproduction for digital images. *ACM Transactions on Graphics 21*, 3 (2002), 267–276.

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics 21*, 3 (2002), 527–536.

[SZC*07] SUN X., ZHOU K., CHEN Y., LIN S., SHI J., GUO B.: Interactive relighting with dynamic brdfs. *ACM Transactions on Graphics 26* (July 2007), 27-1–27-10.

[TS06] TSAI Y.-T., SHIH Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Transactions on Graphics 25*, 3 (2006), 967–976.

[WFA*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D. P.: Lightcuts: a scalable approach to illumination. *ACM Transactions on Graphics 24*, 3 (2005), 1098–1107.

[WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. *SIGGRAPH Computer Graphics 22* (June 1988), 85–92.

[WRG*09] WANG J., REN P., GONG M., SNYDER J., GUO B.: All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Transactions on Graphics 28* (December 2009), 133:1–133:10.

[WTL04] WANG R., TRAN J., LUEBKE D.: All-Frequency Relighting of Non-Diffuse Objects using Separable BRDF Approximation. In *Proceedings of the Eurographics Symposium on Rendering* (Norrköping, Sweden, 2004), A. Keller and H. W. Jensen (Eds.). The Eurographics Association, Aire-la-Ville, Switzerland, pp. 345–354.

[WTL06] WANG R., TRAN J., LUEBKE D.: All-frequency relighting of glossy objects. *ACM Transactions on Graphics 25*, 2 (2006), 293–318.

[WWZ*09] WANG R., WANG R., ZHOU K., PAN M., BAO H.: An efficient GPU-based approach for interactive global illumination. *ACM Transactions on Graphics 28*, 3 (2009), 1–8.