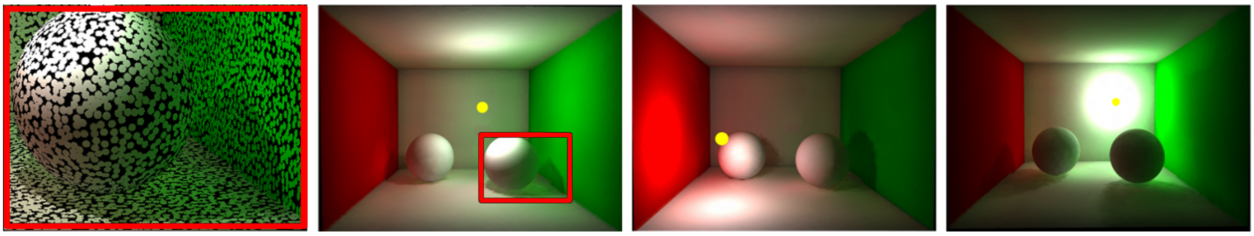
Precomputed Illumination for Virtual Reality Scenarios

Mathias Leonhardt and Christian-A. Bohn

Wedel University of Applied Sciences
Feldstr. 143
22880 Wedel
Tel.: +49 (0)4103 / 80 48 - 810
Fax: +49 (0)4103 / 80 48 - 91810
E-Mail: mle@fh-wedel.de



Precomputed Radiance Transfer (PRT) rendered at 30 fps using the proposed function basis. Just 10 coefficients are sufficient for capturing indirect light, 50 coefficients are used for the visibility information. Data is precomputed at $45K$ sample points (see scene detail on the left) requiring 30 MB of data storage.

Abstract

In this work we present a lightweight and easy to implement approach to enhance static scenes with diffuse surfaces and local light sources by soft shadows and indirect illumination based on precomputed radiance transfer (PRT).

Targeting the easy integration of global illumination effects into existing virtual reality applications, we compared recent PRT based approaches concerning their efficiency and fast implementability. We developed orthonormal bases for the representation of both precomputed indirect illumination and precomputed visibility using PCA of example data yielding a high compression ratio and fast reconstruction of precomputed data.

Unlike most comparable approaches featuring local area and point light sources we do not rely on shadow mapping or stencil shadows in order to determine the direct lighting, although we do not limit the light source to hemispherical directions of a shaded surface point. Further, we apply a meshless basis construction to avoid typical problems when tesselating a given scene.

1 Introduction

Global illumination is a critical factor for the quality of immersion in virtual reality scenarios. However, its computation in real time remains a challenging topic in computer graphics. A comprehensive overview of work which have been done up to today is presented in [RDGK12].

Precomputed radiance transfer (PRT) was introduced by [SKS02]. Global illumination calculations are precomputed offline and retrieved during rendering taking into account the current light source position. Since a large amount of data is precalculated, compression methods are applied in the precomputation step. The obvious drawbacks of PRT is the lengthy precomputation step and the restriction to static scenes. However, these approaches provide a consistent mathematical model to integrate global illumination with low effort into existing scanline rendering methods. For the precomputation step, an arbitrary global illumination renderer is required supplemented by a method for compressing and storing the illumination data. In order to render the precalculated lighting the data needs to be decompressed and linked to the scene geometry which can be accomplished efficiently using GPU splatting methods [GBP04, BeG$^+$05] or by simply coloring the vertices of the mesh. Most PRT approaches deal with static scenes and diffuse surfaces, but also extensions to dynamic scenes have been presented in [IDYN07, NSK$^+$07, LAM$^+$11, LNJS12], extensions to arbitrary BRDFs in [GKMD06, WZH07, XJF$^+$08, AUW07].

2 Previous Work

Existing PRT approches mainly differ in the way precomputed data is compressed and reconstructed. The precomputed data contains the discretized *global transfer function*, which maps either an environmental light source or the direct illumination of the scene geometry to the global illumination. We therefore distinguish between techniques using *environment lighting* and techniques using *direct-to-indirect transfer*.

2.1 Environment Lighting

In these techniques environment light describes the illumination function at each vertex of the scene, being discretized as an illumination vector by e.g. hemispherical sampling. Assuming the illumination function at infinite distance from the scene, the identical illumination vector is used at each vertex. The *global transfer matrix* is a linear mapping from the high dimensional illumination vector to a global illumination vector, consisting of a component for each vertex of the scene geometry. Most PRT approaches utilize orthonormal basis functions to perform a dimension reduction of the illumination vectors and the corresponding transfer vectors since the rendering integral can then be determined using inner products of the projected vectors without explicitly decompressing the data.

[SKS02] utilize *spherical harmonics* to reduce the dimension of the transfer matrix. These orthonormal basis functions capture low frequency global illumination effects like very soft shadows and smooth indirect light. The environment light is projected onto the orthonormal basis only once at the beginning of the rendering process. Due to the rotational invariance of spherical harmonics the projected illumination vector can be rotated without leaving the subspace it has been projected onto.

Similar techniques have been developed, aiming at supporting higher frequency illumination effects like well-defined shadows and even glossy reflections [NRH03, GKPB04, GKMD06, XJF$^+$08, AUW07].

[NSKF07] construct an orthonormal basis by performing a PCA of the local transfer matrix. Only direct illumination is supported but sharp shadow boundaries are yielded using only very few coefficients. We partially adopt this approach due to the high efficiency and simple construction of the function basis, but in contrast to [NSKF07] we do not limit the light source to positions outside of the convex hull of the scene geometry, making this approach suitable for combination with direct-to-indirect transfer techniques.

2.2 Direct-to-Indirect Transfer

Since PRT based on environment lighting only depends on incident light directions, the light sources are implicitly assumed to be located outside of the convex hull of the scene geometry — unlike the infinte distance lighting property of most environment lighting approaches the convex hull restriction is an inherent property of the approach. Supporting dynamic local light sources inside of a complex indoor scene requires a different approach, known as *direct-to-indirect transfer* [HPB06]. The direct illumination is captured at locations of the scene geometry resulting in a high dimensional illumination vector, which is then mapped to indirect lighting using transfer matrices. The direct and indirect illumination is subsequently combined to the resulting global illumination. Since these approaches require a technique to determine the direct illumination, they often are combined with shadow mapping [Wil78] resulting in unnaturally sharp shadows.

[KTHS06] parameterize the rectangular faces of the scene geometry using an orthonormal wavelet basis to capture the incident direct illumination, which is mapped to indirect illumination by the global transfer matrix. Since these basis functions are only suitable for flat rectangular faces, the supported scenes are quite limited.

[WZH07] introduce an orthonormal basis defined over an arbitrary mesh for the compression of the transfer matrices consisting of the eigenvectors of the mesh Laplacian. Thus, arbitrary scenes are supported, but a high and regular mesh tesselation is required, since precomputations are made at the vertex locations.

[LZT$^+$08] introduce a meshless hierarchical basis for representing light transport in arbitrary scenes without requiring a tesselated geometry. The different levels of hierarchy provide a different approximation accuracy of the transfer function. However, the reconstruction of compressed illumination data is not as straightforward as by using orthonormal basis functions and it is difficult to be implemented on a GPU since a hierarchical data structure needs to be traversed.

[LAM+11] present a PRT technique for dynamic scenes based on radiance transfer of simple basic shapes. Since they aim at using only very few memory ressources and very little time for retrieving indirect illumination, they coarsely approximate light transfer, e.g. by ignoring interreflections of smaller objects. So-called *lighting prior bases* are constructed in order to model direct-to-indirect transport operators of these basic shapes.

We propose a function basis for direct-to-indirect tranfer similar to the *lighting prior basis* due to its simple construction scheme utilizing a PCA algorithm only, paired with very high compression capabilities and a fast reconstruction scheme. We further adopt the idea of using a meshless basis in order to profit from the independence of the global illumination quality from the mesh resolution, but combined with an orthonormal basis enabling better GPU support. We also provide a model for capturing shadows as well, without being limited to either infinite small light sources or to positions outside of the convex hull of the scene.

3 Foundations

3.1 Precomputed Radiance Transfer

In computer graphics global illumination is modeled by the well-known *rendering equation* [Kaj86]

$$L_o(x,\vec{\omega}) = L_e(x,\vec{\omega}) + \int_\Omega f_r(\vec{\omega},x,\vec{\omega}')\, L_i(x,\vec{\omega}')\, \cos\theta\, d\omega'\,. \quad (1)$$

Discretization and considering only outgoing radiances yields

$$\vec{L}_o = \vec{L}_e + \mathbf{T}\,\vec{L}_o \quad (2)$$

with the vector of outgoing radiances $\vec{L}_o$ at each of the n sample points, the direct illumination $\vec{L}_e$ of each of the points due to a light source, and the local transfer matrix $\mathbf{T}$ with

$$T_{i,j} = f_{r_d}(x_i,\vec{\omega}_j)\, cos\theta\, \Delta\omega_j\,. \quad (3)$$

By applying Neumann series expansion, $\vec{L}_o$ can be expressed in a closed form introducing the global transfer matrix $\mathbf{S}$

$$\begin{aligned}\vec{L}_o &= \vec{L}_e + \mathbf{T}\,\vec{L}_e + \mathbf{T}^2\,\vec{L}_e + ...\\ \vec{L}_o &= (\mathbf{I} + \mathbf{T} + \mathbf{T}^2 + ...)\,\vec{L}_e\\ \vec{L}_o &= \mathbf{S}\,\vec{L}_e\,.\end{aligned} \quad (4)$$

Precomputed radiance transfer techniques aim at precomputing and compressing the global transfer matrix $\mathbf{S}$.

3.2 Orthonormal Bases

Let $\mathfrak{B} = \{B_1(x), B_2(x), B_3(x), ...\}$ an orthonormal basis of $\mathbb{R}^{\mathbb{R}}$, implying

$$\int B_j(x)B_k(x)\,\mathrm{d}x = \begin{cases}1, & \text{if } j = k\\ 0, & \text{if } j \neq k\end{cases} \quad (5)$$

the inner product of two functions modeled by this basis can be written as

$$\begin{aligned}
&\int f(x)g(x)\mathrm{d}x \\
\approx &\int \left(\sum_{i=0}^{n} f_i\, B_i(x)\right)\left(\sum_{i=0}^{n} g_i\, B_i(x)\right)\mathrm{d}x \\
= &\sum_{i=0}^{n}\sum_{j=0}^{n} f_i g_j \int B_i(x)\, B_j(x)\mathrm{d}x \\
= &\sum_{i=0}^{n} f_i g_i .
\end{aligned} \tag{6}$$

Thus, projecting $\mathbf{S}$ and $\vec{L}_e$ into an orthonormal basis enables Equ. 4 to be computed using a dot product of n-dimensional coefficient vectors.

3.3 Principal Component Analysis

We utilize principal component analysis to reduce the dimensionality of the precomputed visibility information and the transfer matrices. Given a $n \times d$ matrix $\mathbf{A}$, consisting of n d-dimensional row vectors $\vec{a}_i$, and letting $\vec{\mu} \in \mathbb{R}^d$ be the arithmetic mean of all columns of $\mathbf{A}$, the PCA is computed by subtracting $\vec{\mu}^T$ from each row of $\mathbf{A}$ and performing a singular value decomposition of the resulting matrix

$$\mathbf{A}_c = \mathbf{A} - \mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{7}$$

where $\mathbf{M}$ is the mean vector $\vec{\mu}^T$ repeated n times. The first d' rows of $\mathbf{V}$ form an orthonormal basis $\mathbf{B} \in \mathbb{R}^{d' \times d}$ that can be used to project d-dimensional vectors onto a d' dimensional subspace, yielding a good approximation for vectors that correlate to the row vectors of $\mathbf{A}_c$ since the d'-dimensional hyperplane spanned by the row vectors of $\mathbf{B}$ forms a least squares fitting into these d-dimensional vectors.

Let $\mathbf{X}$ be a $m \times d$ matrix, consisting of m row vectors, $\mathbf{X}$ can be projected onto the subspace

$$\hat{\mathbf{X}} = (\mathbf{X} - \mathbf{M}) \times \mathbf{B}^T . \tag{8}$$

The reconstruction yields an approximation of $\mathbf{X}$

$$\mathbf{X} \approx \hat{\mathbf{X}} \times \mathbf{B} + \mathbf{M} . \tag{9}$$

4 Precomputation

The precomputation process consists of three phases as follows.

1. Generation of sample locations on the scene geometry and clustering.
2. Construction of the visibility function basis and the projection of the visibility information onto it.

3. Construction of the basis used for capturing direct illumination and construction of the compressed transfer matrices using this basis.

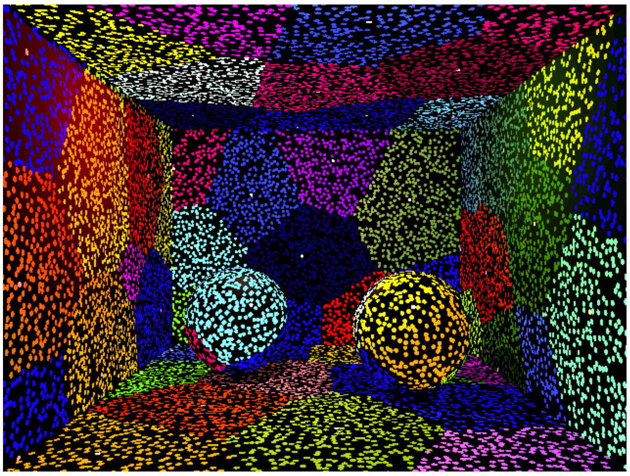


Figure 1: Sample points of the example scene. Clusters are colored randomly, white dots expose the cluster root points.

4.1 Sample Point Generation

Sample points are uniformly distributed over the faces of the scene geometry. The selected sample density corresponds to the possible accuracy of global illumination effects, since every sample represents an illumination value during rendering. Thus, in scenes with mainly low frequency lighting effects a lower sample density can be chosen than in scenes, where, e.g., sharp shadow boundaries are expected. Decoupling the samples from the vertices of the scene primitives ensures that the tradeoff between rendering speed and accuracy is a scalable property of the algorithm. We adopt this idea from [LZT+08] where point samples are used to construct a hierarchical representation of light transport. Particles are sent out from a point inside of the scene and are traced through the scene for several random reflections. In order to enable the creation of an uniform distribution, after the fifth reflection the following collision points of the particles with the scene are stored — first as candidates — into a kd-tree. When a sufficient number of candidates is reached, the candidates are thinned by eliminating all candidates in a certain radius of a selected sample point (see Fig. 1). This radius depends on the desired sample point density and is also affected by the surface curvature allowing more samples to be placed on curvy objects. The process continues until there are no candidates left, but only sample points. Then the points are grouped into clusters. *Root points* forming the middle of a cluster are identified in the same way using a larger radius. Each sample point is then associated with the nearest root point. Thus, the radius influences the number of sample points per cluster. In our example implementation, we used about 1000 sample points in each cluster.

4.2 Precomputation of Visibility

[NSKF07] present a method for precomputing shadowing information for light sources placed outside of the convex hull of a scene. We adopt the use of a PCA basis being optimally suited for compression, but since we aim at placing a light source at arbitrary positions inside of the scene, this approach is not exactly suitable to our problem. Thus we construct a hemispherical depth buffer (see Fig. 2) for each sample point by carrying out a ray casting calculation for a couple of directions of the hemisphere. The generated depth values embody the visibility of that sample point through the fact that a light source in a certain direction is not visible if its distance from the sample point is bigger than the related depth buffer entry. Let the cluster c contain n_c sample points and let each depth buffer contain d distance

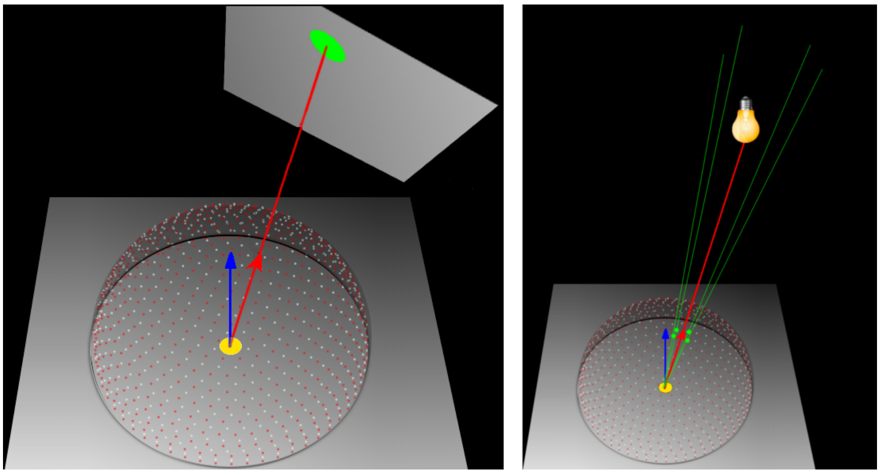


Figure 2: The computation of one value in the hemispherical depth buffer depends on the four nearest sample rays.

values corresponding to d hemispherical directions resulting in an $n_c \times d$ matrix $\mathbf{S}_c$. We construct a d_1-dimensional orthonormal basis $\mathbf{B}_c$ by applying a PCA to $\mathbf{S}_c$. In the example implementation we yielded good results with $d \approx 2000$ and $d_1 = 50$. The depth buffers are then projected onto the d_1-dimensional subspace using Equ. 8 resulting in a compressed matrix $\hat{\mathbf{S}}_c$ that is stored in a file. During rendering, the depth values are reconstructed using Equ. 9 for the four directions closest to the direction aiming at the light source. For more complex light sources accordingly more depth values have to be reconstructed.

We emphasize the fact that all data being compressed is previously used for the construction of the basis. This is a vital difference compared to the kind of construction of the transfer matrices basis (see next section).

4.3 Precomputation of the Transfer Matrices

The computation of the direct-to-indirect transfer matrices requires a clustering of the sample points to reduce the runtime and numerical instabilities of the SVD algorithm. Each pair of clusters share a transfer matrix encoding the radiance transfer between them, consisting of d_2 rows and n_c columns, where d_2 is the number of basis vectors used to compress the matrix and n_c is the number of samples of the receiver cluster. A compression of the row vectors of the transfer matrix would not be reasonable since a radiance value for every sample point of the receiver cluster is needed. Note that an uncompressed patch-to-patch transfer matrix would contain as many rows as there are sample points in the sender cluster — about 1000 in our case. However, d_2 is required to be 10 only in our example implementation.

4.3.1 Basis Construction

During rendering, direct illumination will be determined using realtime local lighting calculations and the precomputed visibility information. Subsequently, the direct lighting is projected onto our basis and mapped to indirect lighting using the transfer matrices. Therefore, the basis needs good approximation properties for capturing direct illumination. Unlike the basis used for compressing the visibility information, this basis cannot be constructed using the same data being compressed due to the fact that not all possible direct lighting situations are known, too. Also, constructing the transfer matrices before compressing them would entail an enormous precomputation effort. Thus, we construct a basis based on example data, enabling computing the transfer matrices in already compressed form. For this purpose, we place a point light source successively at uniformly distributed locations in the scene and capture the direct illumination of each cluster c through a direct illumination vector $\vec{l}_c$ with n_c components. This yields l direct illumination vectors $\vec{l}_c$ forming a matrix $\mathbf{L}_c$ for each cluster, with l being the number of example light sources used ($l \approx 1000$ in our example implementation). Applying PCA on $\mathbf{L}_c$ yields an orthonormal basis $\mathbf{D}_c$ well suited for compression of incident direct light on cluster c.

4.3.2 Construction of the Transfer Matrices

We utilize photon mapping for building the transfer matrices. Each of the d_2 basis vectors of a cluster s is used to enlighten the whole scene, leading to global illumination values for each cluster r. These global illumination values are the components of the compressed transfer matrices $\hat{\mathbf{S}}_{s,r}$ describing the global illumination of a receiver patch r due to a sender patch s. This way the linear system shown in Equ. 4 is solved, taking into account that the illumination vector is already projected onto our basis, like

$$\begin{aligned}\vec{L}_r &= \mathbf{S}_{s,r}\,\vec{L}_s \\ \vec{L}_r &= \hat{\mathbf{S}}_{s,r}\,\left((\vec{L}_s^T - \vec{\mu}_c^T) \times \mathbf{D}_c^T\right) \\ \vec{L}_r &= \hat{\mathbf{S}}_{s,r}\,\hat{L}_s\end{aligned} \tag{10}$$

where $\vec{L}_r$ is the indirect illumination of the receiver cluster, $\vec{L}_s$ is the direct illumination of the sender cluster and $\vec{\mu}_c$ is the mean vector of the columns of $\mathbf{L}_c$. We emphasize that $\mathbf{S}_{s,r}$ is not actually determined, but only the compressed matrix $\hat{\mathbf{S}}_{s,r}$ whereas the projection of $\vec{L}_s$ has to be accomplished during rendering for each frame. However, this does not influence rendering performance significantly due to the small number of basis functions required. Like pointed

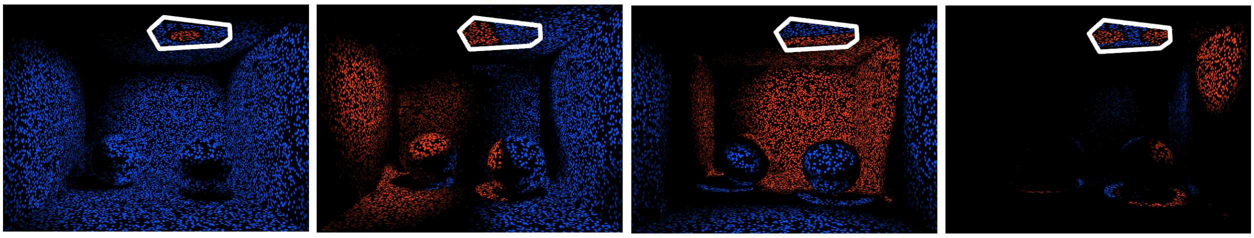


Figure 3: Construction of transfer matrices. A sending cluster illuminates all other clusters by emitting light according to its basis functions. Red areas illustrate positive values, blue areas illustrate negative values. For higher frequency basis functions dark areas are getting larger, leading to a sparse matrix representation.

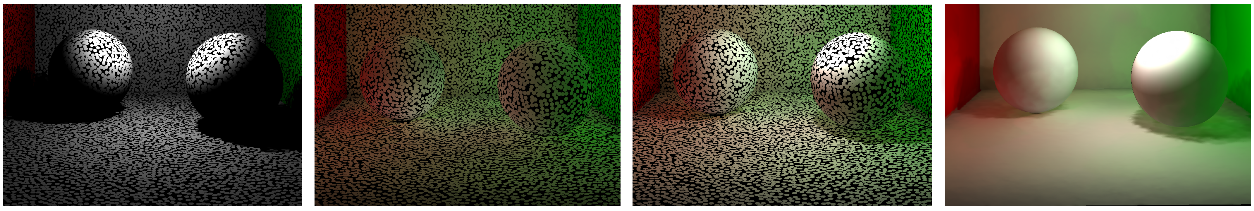


Figure 4: Rendering process. Visibility information is reconstructed at each sample point and combined with local per pixel lighting, resulting in direct illumniation (left picture). Direct illumination is mapped to indirect illumination (second picture from left). Direct and indirect illumination are combined (third picture from left). Splats are enlarged and rendered using elliptical gaussian splats (right picture).

out with Equ. 6 the product of the compressed vectors $\hat{\mathbf{S}}_{s,r}\,\hat{L}_s$ yields an uncompressed vector $\vec{L}_r$.

4.4 Sparse Matrix Representation

In diffuse lighting conditions the coefficients of basis function decrease rapidly with higher frequencies and thus can soon be neglected. This leads to approximations with very few coefficients and to a sparse transfer matrix. Fig. 3 exposes this fact. For higher frequency basis functions not only the resulting illumination is getting very dark but there are large very dark areas as well as some bright areas. This effect also applies to the projected visibility information. In our example scene we found that only less than 20 percent of the projected data is required to keep a comparable visual quality.

Furthermore, applying quantization to the 32 bit floats by mapping them to 16 bit integers additionally reduces the data by 50 percent.

5 Rendering

The rendering process consists of three phases as follows (see Fig. 4).

1. Reconstruction of the visibility information and determination of the direct illumina-

tion.

2. Projection of the direct illumination onto the basis and mapping of the projected direct illumination vector to the indirect illumination.

3. Combination of direct and indirect illumination and shading the mesh accordingly using point rendering techniques.

First, the visibility information is reconstructed for the directions from each sample point to the light source. If a single point light source is used, the visibility information is reconstucted for the four nearest directions. A higher number of light sources and even complex light sources can simply be handled evaluating visibility information for further directions. The distance from a sample point to an occluder is reconstructed by evaluating Equ. 9 for the row vector of $\hat{\mathbf{S}}_c$ corresponding to the considered direction. Precomputed directions are evenly distributed over the hemisphere such that a lookup function is needed to map a direction to the corresponding row vectors of $\hat{\mathbf{S}}_c$. This mapping as well as the reconstruction is implemented on the GPU. The precomputed distances to occluding surfaces are then compared to the current distance to the light source, resulting in whether or not the sample point lies inside of a shadow region. The local illumination is computed per pixel using a fragment shader and combined with the shadow information. For the visibility determined for k directions, k dot products of d_1-component vectors are evaluated per sample point.

Given the direct illumination vector $\vec{L}_s$ of each sender cluster s, the indirect illumination of a sample point is evaluated using p dot products of d_2-component vectors (Equ. 10) where p is the total number of clusters.

The illumination is splatted onto the surface using gaussian elliptical splats, resembling common point techniques.

6 Results

In Fig. 5, a rendering result computed at 30 fps with 1024x768 screen resolution is shown. Precomputation was done using 45K sample points divided into 48 clusters. The precomputation process took about 4 hours using a simple photon mapper, what could be greatly reduced using an instant radiosity-based approach [Kel97], since only global illumination from diffuse surfaces is computed. The visibility information was precomputed using 2016 directions per sample point being compressed to 50 coefficients. Sparse matrix representation and quantizing reduces the data to 4.47 MB of storage containing the compressed visibility information and the basis functions. 10 coefficients for each patch-to-patch transfer matrix (48^2 matrices) were used for capturing the indirect illumination. It requires 21 MB of storage for transfer matrices and basis functions. Using the approach of [WZH07] for supporting dynamically changing BRDFs would further reduce the size of the transfer matrices to 30 percent since color information is not required. Local illumination is calculated using a simple per pixel fragment shader.

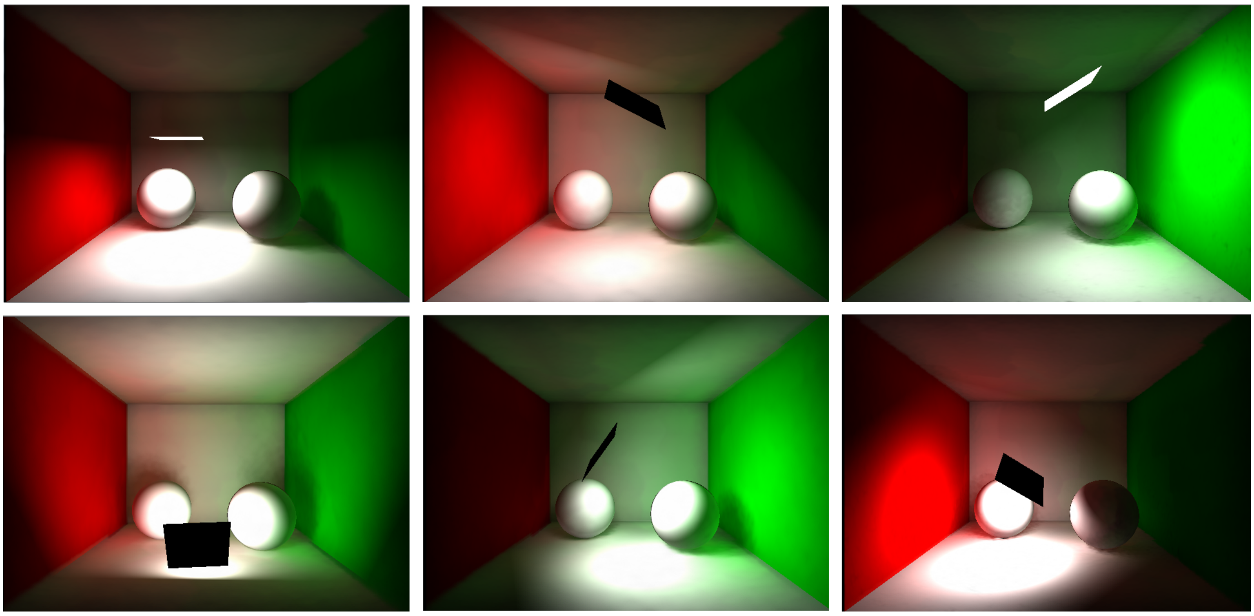


Figure 5: Renderings with different light source positions at 25 fps and 1024x768 resolution. The area light source is sampled at 25 locations. Precomputation of indirect light and visibility information is based on 45K sample points.

6.1 Conclusion

We present a PRT approach easily integrateable into existing virtual reality applications targeting an increase of immersion due to global illumination effects. A precomputation-based technique for the visibility-problem for local area light sources and a direct-to-indirect transfer scheme have been developed, both supporting easy basis construction needing a PCA algorithm only. The orthogonality of the function bases allow convenient GPU support due to simple inner product reconstruction of precomputed data and we avoid the need of mesh tesselation of most PRT techniques due to a meshless basis. Clustering of the data makes our approach potentionally suitable to complex and large scenes.

6.2 Future work

The prototype developed in this work perfectly serves as a proof of concept for our proposed approach. Evaluations regarding cluster size and number of basis functions have to be accomplished to figure out the tradeoff between quality and rendering speed. A promising approach for supporting large scale scenes with huge amounts of sample points would be a multiresolution technique like proposed in [DVS03] which leads to different sized splats depending on the distance from the viewer. The general concept introduced by our approach could also be extended to glossy BRDFs and dynamically changing BRDFs following the approach of [WZH07].

References

[AUW07] Oskar Akerlund, Mattias Unger, and Rui Wang. Precomputed visibility cuts for interactive relighting with dynamic brdfs. *Computer Graphics and Applications, Pacific Conference on*, 0:161–170, 2007.

[BeG+05] Kavita Bala, Philip Dutré (editors, Pascal Gautron, Kadi Bouatouch, and Sumanta Pattanaik. Radiance cache splatting: A gpu-friendly global illumination algorithm, 2005.

[DVS03] Carsten Dachsbacher, Christian Vogelgsang, and Marc Stamminger. Sequential point trees. In *ACM Transactions on Graphics*, pages 657–662, 2003.

[GBP04] Gael Guennebaud, Loïc Barthe, and Mathias Paulin. Deferred Splatting. *Computer Graphics Forum*, 23(3):653–660, 2004.

[GKMD06] Paul Green, Jan Kautz, Wojciech Matusik, and Frédo Durand. View-dependent precomputed light transport using nonlinear gaussian function approximations. In *In ACM Symposium on Interactive 3D graphics*, pages 7–14, 2006.

[GKPB04] Pascal Gautron, Jaroslav Křivánek, Sumanta N. Pattanaik, and Kadi Bouatouch. A novel hemispherical basis for accurate and efficient rendering. In *Rendering Techniques 2004, Eurographics Symposium on Rendering*, pages 321–330. Eurographics Association, June 2004.

[HPB06] Miloš Hašan, Fabio Pellacini, and Kavita Bala. Direct-to-indirect transfer for cinematic relighting. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 1089–1097, New York, NY, USA, 2006. ACM.

[IDYN07] Kei Iwasaki, Yoshinori Dobashi, Fujiichi Yoshimoto, and Tomoyuki Nishita. *Precomputed radiance transfer for dynamic scenes taking into account light interreflection.* Eurographics Association, 2007.

[Kaj86] James T. Kajiya. The rendering equation. *ACM Siggraph*, 1986.

[Kel97] Alexander Keller. Instant radiosity, 1997.

[KTHS06] Janne Kontkanen, Emmanuel Turquin, Nicolas Holzschuch, and François Sillion. Wavelet radiance transport for interactive indirect lighting. In Wolfgang Heidrich and Tomas Akenine-Möller, editors, *Eurographics Symposium on Rendering, Rendering Techniques 2006, June, 2006*, pages 161–171, Nicosia, Chypre, June 2006. Eurographics.

[LAM+11] Bradford J. Loos, Lakulish Antani, Kenny Mitchell, Derek Nowrouzezahrai, Wojciech Jarosz, and Peter-Pike Sloan. Modular radiance transfer. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2011)*, 30(6), December 2011.

[LNJS12] Bradford James Loos, Derek Nowrouzezahrai, Wojciech Jarosz, and Peter-Pike Sloan. Delta radiance transfer. In *ACM Siggraph Symposium on Interactive 3D Graphics and Games*, New York, NY, USA, 2012. ACM.

[LZT+08] Jaakko Lehtinen, Matthias Zwicker, Emmanuel Turquin, Janne Kontkanen, Frédo Durand, François Sillion, and Timo Aila. A meshless hierarchical representation for light transport. *ACM Trans. Graph.*, 27(3), 2008.

[NRH03] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph.*, 22:376–381, July 2003.

[NSK+07] Derek Nowrouzezahrai, Patricio Simari, Evangelos Kalogerakis, Karan Singh, and Eugene Fiume. Compact and efficient generation of radiance transfer for dynamically articulated characters. In *GRAPHITE '07: Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, New York, NY, USA, jan 2007. ACM.

[NSKF07] Derek Nowrouzezahrai, Patricio Simari, Evangelos Kalogerakis, and Eugene Fiume. Eigentransport for efficient and accurate all-frequency relighting. In *GRAPHITE 07: Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, page 163169, New York, NY, USA, January 2007. ACM.

[RDGK12] Tobias Ritschel, Carsten Dachsbacher, Thorsten Grosch, and Jan Kautz. The state of the art in interactive global illumination. *Comput. Graph. Forum*, 31(1):160–188, 2012.

[SKS02] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, pages 527–536, New York, NY, USA, 2002. ACM.

[Wil78] Lance Williams. Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph.*, 12:270–274, August 1978.

[WZH07] Rui Wang, Jiajun Zhu, and Greg Humphreys. Precomputed radiance transfer for real-time indirect lighting using a spectral mesh basis. *EuroGraphics Symposium on Rendering*, 2011(12.11.2011), 2007.

[XJF+08] Kun Xu, Yun-Tao Jia, Hongbo Fu, Shi-Min Hu, and Chiew-Lan Tai. Spherical piecewise constant basis functions for all-frequency precomputed radiance transfer. *IEEE Transaction on Visualization and Computer Graphics*, 14(2):454–467, 2008.