

Spatial Directional Radiance Caching

Václav Gassenbauer[†]
IRISA / INRIA Rennes
Czech Technical University in Prague

Jaroslav Křivánek
Cornell University
Charles University, Prague

Kadi Bouatouch
IRISA / INRIA Rennes

Abstract

We present a new approach for accelerated global illumination computation in scenes with glossy surfaces. Our algorithm combines sparse illumination computation used in the radiance caching algorithm with BRDF importance sampling. To make this approach feasible, we extend the idea of lazy illumination evaluation, used in the caching approaches, from the spatial to the directional domain. Using importance sampling allows us to apply caching not only on low-gloss but also on shiny materials with high-frequency BRDFs, for which the radiance caching algorithm breaks down.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism – Rendering, Global Illumination—

1. Introduction

Global illumination (GI) effects constitute an important aspect of generating realistic images for applications spanning film production, video games, industrial design or architecture. Many algorithms for simulating GI effects have been proposed, however, these methods are very time-consuming for general environments. Therefore, full global illumination solution is often restricted to a simpler case, such as ambient occlusion or predominantly diffuse indirect illumination, for which efficient algorithms are known [WRC88, WABG06, HPB07]. In this paper, we focus on a more difficult case of indirect illumination on surfaces with arbitrary material properties, coming from both diffuse and glossy objects. We refer to those effects as *directional indirect illumination*.

Monte Carlo (MC) ray tracing algorithms are almost exclusively used for rendering scenes with directional indirect effects. Unfortunately variance of MC estimators gives rise to image noise that decreases only with the square-root of the number of samples. Therefore, research in computer graphics has focused on variance reduction techniques, of which the most widely accepted is importance sampling [PH04, CAE08, CAM08]. Other acceleration techniques trade efficiency for bias in the resulting solution. Some of these techniques calculate illumination only at several locations in the scene exactly. The resulting image is then generated

using a reconstruction function from these points. The pioneering work in this area was the irradiance caching algorithm [WRC88], the extensions of which include *radiance caching* on glossy surfaces with low-frequency BRDFs [KGPB05] and in participating media [JDZJ08].

In this paper, we propose an algorithm that combines the sparse computation of indirect illumination used in radiance caching with the variance reduction offered by BRDF importance sampling. The main idea is to extend the sparse illumination evaluation from the spatial to the directional domain: The indirect radiance at a point in a direction is evaluated by interpolating radiance samples from neighboring directions and locations. Using this strategy we obtain an algorithm that has the following advantages: (1) exploits spatial and directional illumination coherence, (2) ensures a smooth integration of the view-dependent BRDFs through interpolation in both these domains, (3) avoids conversion of the scene BRDFs into a special-purpose representation, such as spherical harmonics [KGPB05], thereby making the algorithm more flexible, (4) gains efficiency by exploiting BRDF importance sampling.

The proposed algorithm computes the first bounce of indirect illumination on glossy surfaces. Multiple bounces can be added e.g. by the use of photon mapping [Jen01]. The algorithm can be used as a part of a full solution to global illumination computation. Diffuse inter-reflections could be handled using irradiance caching, highly specular reflections

[†] e-mail: gassev1@fel.cvut.cz

by classical MC sampling methods, while the reflections on glossy surfaces could be computed using our algorithm.

The remainder of this paper is organized as follows. Section 2 summarizes the prior work, Section 3 reviews the radiance caching algorithm, Section 4 describes our algorithm while Section 5 provides its evaluation. The algorithm is discussed in Section 6 and Section 7 concludes the paper.

2. Related Work

Our algorithm is a variant of illumination caching techniques. The seminal work in this area is irradiance caching [WRC88], that accelerates global illumination computation on diffuse surfaces. Indirect illumination is computed by interpolating previous irradiance values if these are available. If none of the previously cached irradiance values can be used, a new one is computed by sampling the hemisphere and cached. Ward and Heckbert [WH92] propose to use the translation and rotation gradient to improve the quality of interpolation. Tabellion and Lamorlette [TL04] use irradiance caching combined with an approximate lighting model in cinematic lighting. Brouillat et al. [BGB08] propose to use a photon map [Jen01] for fast construction of a coarse approximation of irradiance cache. Arikan et al. [AFO05] speed up irradiance caching by decomposing illumination into near and distant terms. The above mentioned methods work for diffuse illumination only while radiance caching proposed by Křivánek et al. [KGPB05] is designed for the use on glossy surfaces with low-frequency BRDFs. The goal of our work is to modify radiance caching in such a way that caching can be applied for higher-frequency BRDFs.

Other extensions of the (ir)radiance caching algorithms include the support for caching in dynamic environments [GBP07]. Instead of building (ir)radiance cache for each frame from scratch, they reuse and update existing records for several frames. Gautron et al. [GKBP05] also propose speeding up irradiance caching using the graphic accelerator by reformulating the algorithm to better fit the GPU architecture. Jarosz et al. [JDZJ08] extend the radiance caching algorithm to cache lighting in participating media. Later they reformulated the gradient computation to account for changes of occlusion [JZJ08]. These extensions show the potential of the caching methods for faithfully rendering global illumination effects.

The idea of reusing partial results of illumination computation is not limited to the caching approaches. For example, photon mapping proposed by Jensen [Jen01] and instant radiosity proposed by Keller [Kel97] reuse the same set of paths initiated from light sources for all image pixels. Approaches for reusing camera-paths and bidirectional paths have also been proposed [BSH02, HDMS03, CSH08]. Other techniques [PBSP08, PWP08, LW95] reuse the information gathered during illumination sampling to build an importance function to be used at other locations.

3. Background: Radiance Caching

Radiance caching [KGPB05] is based on the observation that indirect illumination tends to change slowly on glossy surfaces. Therefore, it can be computed only at some points, stored in a cache, and later reused for fast interpolation. This approach can significantly speed up the computation.

Whenever indirect illumination needs to be computed at a point \mathbf{p} , the cache is queried for nearby *records* (i.e. the cached illumination values) available for interpolation, formally defined by the set:

$$S(\mathbf{p}) = \left\{ i \mid w_i^s(\mathbf{p}) > \frac{1}{a} \right\}, \quad (1)$$

where a is a user defined maximum interpolation error. The greater the value of a , the more allowance for interpolation. The spatial weight $w_i^s(\mathbf{p})$ determines the contribution of the i -th record to illumination at a given point \mathbf{p} with normal \mathbf{n} . It is given by:

$$w_i^s(\mathbf{p}) = \left(\frac{\|\mathbf{p} - \mathbf{p}_i\|}{R_i} + \sqrt{1 - \mathbf{n} \cdot \mathbf{n}_i} \right)^{-1}, \quad (2)$$

where R_i is the harmonic mean distance to the objects visible from the i -th record's location \mathbf{p}_i , and \mathbf{n}_i is the normal at \mathbf{p}_i .

If the set $S(\mathbf{p})$ is empty, i.e. no records in the vicinity of \mathbf{p} have been found, it is necessary to compute a new record. The full hemisphere above \mathbf{p} is sampled and the directional incoming radiance is approximated by a vector of spherical harmonics coefficients Λ_i . A SH approximation of $\frac{\partial \Lambda_i}{\partial x}$ and $\frac{\partial \Lambda_i}{\partial y}$, the derivatives of Λ_i with respect to translation along the local x and y axes, is also computed. These vectors are stored as a new record in the radiance cache.

If $S(\mathbf{p})$ is not empty, the total outgoing radiance is computed by spatial interpolation of the contributions $L_i^{\text{out}}(\mathbf{p}, \omega^{\text{out}})$ due to the records in $S(\mathbf{p})$:

$$L^{\text{out}}(\mathbf{p}, \omega^{\text{out}}) = \frac{\sum_{i \in S(\mathbf{p})} w_i^s(\mathbf{p}) \cdot L_i^{\text{out}}(\mathbf{p}, \omega^{\text{out}})}{\sum_{i \in S(\mathbf{p})} w_i^s(\mathbf{p})}$$

The contribution of the i -th radiance record is given by:

$$L_i^{\text{out}}(\mathbf{p}, \omega^{\text{out}}) = \mathbf{R}_i \left(\Lambda_i + d_x \frac{\partial \Lambda_i}{\partial x} + d_y \frac{\partial \Lambda_i}{\partial y} \right) \cdot \mathbf{C}(\mathbf{p}, \omega^{\text{out}}),$$

where \mathbf{R}_i is a rotation matrix used to align coordinate frames at \mathbf{p}_i and \mathbf{p} , d_x and d_y is the displacement from \mathbf{p}_i to \mathbf{p} in the local coordinate frame of record i . Finally, $\mathbf{C}(\mathbf{p}, \omega^{\text{out}})$ is the vector of SH coefficient representing the BRDF lobe at \mathbf{p} for the outgoing direction ω^{out} .

4. Spatial Directional Radiance Caching

4.1. Motivation

The major limitation of radiance caching consists of the *uniform* sampling of full hemisphere used to estimate incoming radiance. With uniform sampling, the rays traced outside the

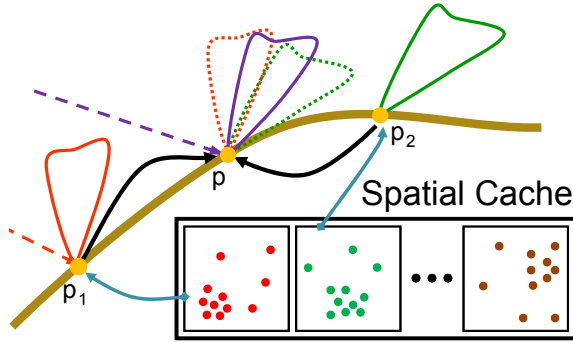


Figure 1: Lazy evaluation scheme in the directional domain. Our goal is to compute the outgoing radiance at point \mathbf{p} . Suppose that there are two cache records at points \mathbf{p}_1 and \mathbf{p}_2 close to \mathbf{p} , that store the incoming radiance samples with high density around the BRDF peaks. The incoming radiance at \mathbf{p} may be obtained by merging the information in the records at \mathbf{p}_1 and \mathbf{p}_2 . If, after the merging, some parts of the hemisphere still do not have high enough sample density, additional rays are traced.

lobe of a glossy BRDF produce wasted effort. Nonetheless, the use of spherical harmonics in radiance caching necessitates a uniform sampling pattern.

To avoid unnecessary computation for shiny glossy surfaces, our algorithm employs BRDF importance sampling when a new cache record is created. Doing so, however, requires to keep track of the sample density on the hemisphere of the cached records, so that they can be reused at other locations, where the BRDF lobe may be slightly different. We approach this issue by extending the radiance caching’s lazy evaluation scheme from the spatial to the directional domain, adding directional samples on the fly as needed. Figure 1 illustrates our lazy evaluation procedure. Using this approach, we are able to handle shiny materials for which the original radiance caching would not be efficient. As an added benefit, there is no need to convert the BRDFs into the SH basis.

Since we use non-uniform sampling to create cache records, we need a representation for the incoming radiance that offers directional localization. This is why spherical harmonics are not an option. Wavelets do offer localization but the rotation \mathbf{R}_i is a limiting factor. Although it can be made relatively fast by pre-computing the rotation matrices [WNLH06], the memory requirements for a reasonable directional resolution (256×256 or more) render this approach impractical. Furthermore, adding new radiance samples to existing records is not simple. Our choice is, therefore, to retain the individual radiance samples and organize them in a kd-tree for fast access.

4.2. Overview

Our algorithm is based on the following caching scheme. To compute outgoing radiance at a point \mathbf{p} , we search for nearby records available for interpolation in the *spatial* cache. If no records are available we create a new cache record as follows. We generate random directions using BRDF importance sampling, compute incoming radiance for each direction by ray tracing, project these directions onto a 2D domain, \mathcal{D} , and build a kd-tree over the radiance samples. We call this kd-tree the *L-tree*. We store the entire *L-tree* in the spatial cache as a single record. Given the *L-tree*, we then continue the computation as though the spatial cache query succeeded.

If there are one or more *L-trees* available close to \mathbf{p} , we interpolate their contributions. We generate random directions using BRDF importance sampling as before. But instead of computing the incoming radiance for these directions by ray tracing, we try to find close radiance samples stored in the *L-trees* for each direction and possibly reuse it. If no suitable radiance sample is available for a direction, we shoot a ray to obtain a new radiance sample and update an existing *L-tree*. Finally, the outgoing radiance is computed as a weighted average of the radiance samples from the individual *L-trees*. The pseudo-code for our spatial directional radiance caching is given in Algorithm 1.

Algorithm 1 Spatial directional radiance caching

```

 $S \leftarrow \text{LOOKUPSPATIAL}(\mathbf{p}, \mathbf{n})$ 
if ( $S$  is empty) then
     $[\omega_j^{\text{in}}]_{j=1}^N \leftarrow \text{SAMPLEBRDF}(\mathbf{p}, \omega^{\text{out}}, N)$ 
    for each  $\omega_j^{\text{in}}$  compute  $L^{\text{in}}(\mathbf{p}, \omega_j^{\text{in}})$  using ray tracing
     $L\text{-tree} \leftarrow \text{build kd-tree over } [\omega_j^{\text{in}}]_{j=1}^N$ 
    Store  $L\text{-tree}$  in spatial cache
     $S \leftarrow \{L\text{-tree}\}$ 
end if

 $[\omega_j^{\text{in}}]_{j=1}^M \leftarrow \text{SAMPLEBRDF}(\mathbf{p}, \omega^{\text{out}}, M)$ 
for each  $\omega_j^{\text{in}}$  do
    for each  $i$  in  $S$  do
         $T_i \leftarrow \text{LOOKUPDIRECTIONAL}(\overline{\omega_j^{\text{in}}})$ 
    end for
    if ( $\cup T_i$  is empty) then
        Choose  $L\text{-tree}_u \in S$  for updating
        if (no suitable  $L\text{-tree}$  exists) continue
         $L^{\text{in}}(\mathbf{p}, \omega_j^{\text{in}}) \leftarrow \text{TRACERAY}(\mathbf{p}, \omega_j^{\text{in}})$ 
        Insert  $L^{\text{in}}(\mathbf{p}, \omega_j^{\text{in}})$  into  $L\text{-tree}_u$ 
    end if
     $\tilde{L}^{\text{in}}(\mathbf{p}, \omega_j^{\text{in}}) \leftarrow \text{INTERPOLATERADIANCE}(\{T_i\}_{i \in S})$ 
    Update  $\tilde{L}^{\text{out}}(\mathbf{p}, \omega^{\text{out}})$ 
end for

```

4.3. New Record Computation

To create a new spatial record, we first generate N random directions using BRDF importance sampling and compute incoming radiance for each direction by ray tracing. We then map these samples from the sphere to the domain \mathcal{D} using paraboloid mapping [HS98] which has low distortion and fast analytical transform. We construct a kd-tree over the radiance samples mapped to \mathcal{D} . To keep the memory requirements low, we quantize sample position inside \mathcal{D} to 2 bytes (resolution of 256×256) and use Ward's RGBE format to represent the incoming radiance. Together with 2 bytes for bookkeeping information, each L -tree node takes 8 bytes. The record computation is completed by inserting the whole L -tree into the spatial cache organized as an octree [WRC88].

The number of rays N used for creating a new record is derived from the number of directions M used to compute the outgoing radiance during interpolation. If N is too low ($N < 8M$), the query to the contributing L -trees often fails and triggers many L -tree updates, negatively affecting performance. The number of L -tree updates stops decreasing for $N > 16M$. A good compromise that works well in our scenes is to use $N \approx 12M$.

4.4. Incoming Radiance Interpolation

To compute outgoing radiance at point \mathbf{p} we start by querying the spatial cache. The definition of the set $S(\mathbf{p})$ of records used for interpolation and their spatial weights are borrowed from the original radiance caching algorithm, see Equations (1) and (2). If the set $S(\mathbf{p})$ is empty, we compute a new record as described above and insert it into $S(\mathbf{p})$.

In the next step we generate M random directions ω_j^{in} using BRDF importance sampling at \mathbf{p} . We map these directions to \mathcal{D} and for each direction ω_j^{in} we collect nearby radiance samples stored in the contributing records' L -trees. These samples are used for the directional interpolation.

Given a direction ω_j^{in} and an L -tree i , we use a range query to collect nearby radiance samples stored in the L -tree as shown in Figure 2. The query radius $r_d(\omega_j^{\text{in}})$ is given by:

$$r_d(\omega_j^{\text{in}}) = \min \left\{ r_{\max}, \frac{1}{2\pi} \frac{1}{M \sqrt{p(\omega_j^{\text{in}})}} \right\},$$

where $p(\omega_j^{\text{in}})$ is the probability density of the BRDF sampling in direction ω_j^{in} . The radius is designed to adapt both to the number of rays M and to the reflectance properties at \mathbf{p} . In particular, the radius will be smaller when the pdf value is high, i.e. around the peak of the BRDF lobe. We use a ceiling of $r_{\max} = 0.15$ to avoid errors when the pdf value is very small. The collected radiance samples are formally defined by the set $T_i(\overline{\omega_j^{\text{in}}})$:

$$T_i(\overline{\omega_j^{\text{in}}}) = \left\{ k \mid w_{ik}^d(\overline{\omega_j^{\text{in}}}) > 0 \right\},$$

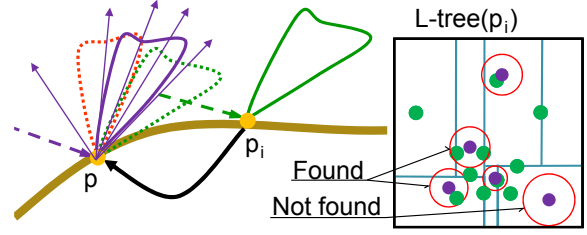


Figure 2: Directional interpolation. We generate M directions ω_j^{in} by BRDF sampling at \mathbf{p} . The directions are mapped to points in \mathcal{D} . For each point $\overline{\omega_j^{\text{in}}}$ (blue dots) we find directionally close radiance samples (green dots) in the contributing L -trees. These radiance samples are used for interpolation. The directions for which no radiance samples are found yield ray tracing and L -tree update.

The upper bar denotes the paraboloid mapping from the sphere to \mathcal{D} , i.e. $\overline{\omega}$ is ω mapped to \mathcal{D} . The directional weight $w_{ik}^d(\overline{\omega_j^{\text{in}}})$ of the k -th radiance sample in the i -th L -tree with respect to $\overline{\omega_j^{\text{in}}}$ is defined as:

$$w_{ik}^d(\overline{\omega_j^{\text{in}}}) = \max \left\{ 0, 1 - \frac{\|\overline{\omega_{ik}^{\text{in}}} - \overline{\omega_j^{\text{in}}}\|^2}{r_d(\overline{\omega_j^{\text{in}}})^2} \right\},$$

where $\overline{\omega_{ik}^{\text{in}}}$ is the coordinate of the k -th radiance sample in the i -th L -tree.

All the collected radiance samples in sets $T_i(\overline{\omega_j^{\text{in}}})$ from all the contributing L -trees participate in the directional interpolation. Suppose we have two contributing L -trees. The first one is spatially close to \mathbf{p} but the radiance sample found in it lies almost by the edge of the directional search radius. The second one has exactly the opposite property, i.e. it is spatially far from \mathbf{p} but the radiance sample found is incident with the direction sample. For proper evaluation of both these cases it is necessary to find a relationship that correlates between the spatial and the directional weights. We use the following weighted sum:

$$\tilde{L}^{\text{in}}(\mathbf{p}, \omega_j^{\text{in}}) = \frac{\sum_{i \in S} \sum_{k \in T_i} w_i^s(\mathbf{p}) w_{ik}^d(\overline{\omega_j^{\text{in}}}) L_{ik}^{\text{in}}}{\sum_{i \in S} \sum_{k \in T_i} w_i^s(\mathbf{p}) w_{ik}^d(\overline{\omega_j^{\text{in}}})}$$

where L_{ik}^{in} is the k -th radiance sample stored in the i -th L -tree and $\tilde{L}^{\text{in}}(\mathbf{p}, \omega_j^{\text{in}})$ is the interpolated incoming radiance.

If, however, no radiance sample close to $\overline{\omega_j^{\text{in}}}$ was found in any L -tree, a new radiance sample is computed using ray tracing. A question arises how to choose the ray origin and update the contributing L -trees. The first option is to place the ray origin at the interpolation location \mathbf{p} . However, such a sample cannot be used to update the contributing L -trees

since that could potentially lead to its reuse at a too distant spatial location. Instead, one could create a new L -tree at \mathbf{p} with all the added radiance samples. However, this approach generates many new L -trees with only a few radiance samples and the interpolation becomes inefficient.

The second option, which we use in our algorithm, is to place the ray origin at the location of one of the L -trees. The L -tree is selected randomly from among the contributing L -trees for which ω_j^{in} is above their tangent plane. The computed radiance sample is then inserted into the selected L -tree. A simple implementation of the insertions turned out to be the most efficient one: We keep a buffer of added radiance samples; when it is full, the tree is rebuilt. If no L -tree can be used for update, which occurs very rarely, we reject the sample ω^{in} . We have opted for the second technique since it distributes the additional radiance samples well among the existing L -trees.

4.5. Outgoing Radiance Computation

The interpolated incoming radiances $\tilde{L}^{\text{in}}(\mathbf{p}, \omega_j^{\text{in}})$ calculated for each ω_j^{in} are used to compute the outgoing radiance $\tilde{L}^{\text{out}}(\mathbf{p}, \omega^{\text{out}})$. The value of $\tilde{L}^{\text{out}}(\mathbf{p}, \omega^{\text{out}})$ at a point \mathbf{p} in the direction ω^{out} is computed using the following Monte Carlo estimator:

$$\tilde{L}^{\text{out}}(\mathbf{p}, \omega^{\text{out}}) = \frac{1}{M} \sum_{j=1}^M \frac{\tilde{L}^{\text{in}}(\mathbf{p}, \omega_j^{\text{in}}) f_r(\mathbf{p}, \omega_j^{\text{in}}, \omega^{\text{out}}) \cos \theta_j^{\text{in}}}{p(\omega_j^{\text{in}})} \quad (3)$$

where $f_r(\mathbf{p}, \omega_j^{\text{in}}, \omega^{\text{out}})$ is the BRDF at \mathbf{p} and θ_j^{in} is the angle between the surface normal at \mathbf{p} and ω_j^{in} .

4.6. Cache Record Density Control

For faithful reconstruction of indirect illumination, the distribution of the records should be proportional to its change rate. In the original irradiance cache algorithm the rate of change is estimated based on the information about surrounding geometry obtained during hemisphere sampling [WRC88]. Estimating the rate of change of indirect illumination for glossy surfaces is more difficult, though, since it should take into account the reflectance properties and the viewing direction. The formula is difficult to derive even for the simplest reflectance models. Instead, our interpolation criterion is based on the original formulas for diffuse surfaces with the addition of the following heuristics.

While creating a new cache record, we estimate the derivatives of outgoing radiance with translation, $\vec{\nabla}_x \tilde{L}^{\text{out}}(\mathbf{p}, \omega^{\text{out}})$ and $\vec{\nabla}_y \tilde{L}^{\text{out}}(\mathbf{p}, \omega^{\text{out}})$. The derivatives are used for clamping the harmonic mean distance R_i of the new record, similar to regular irradiance caching [KG09]:

$$\text{if } \Delta_i > 1/R_i, \quad \text{then } R_i := 1/\Delta_i,$$

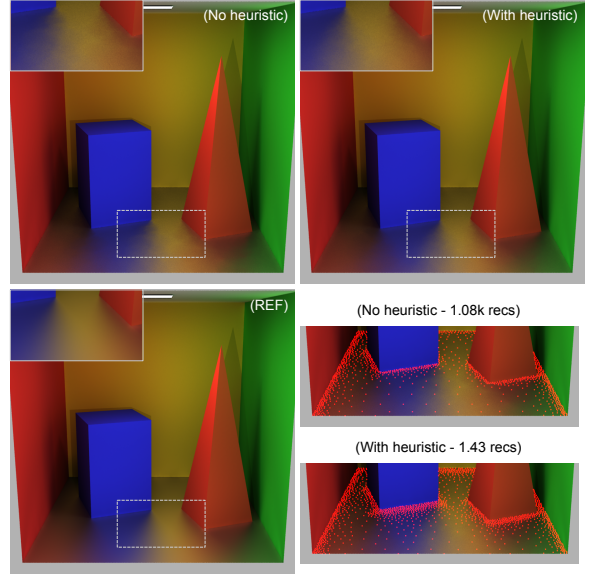


Figure 3: Cache record density control. Individual image lists rendering of a simple scene using our algorithm without detection of high changes of indirect illumination, rendering with the detection of high changes and a reference image. Last picture shows the distribution of cache records for both images. Note the higher density of cache records in places with large change of indirect illumination.

where

$$\Delta_i = \frac{\sqrt{\|\vec{\nabla}_x \tilde{L}^{\text{out}}(\mathbf{p}, \omega^{\text{out}})\|^2 + \|\vec{\nabla}_y \tilde{L}^{\text{out}}(\mathbf{p}, \omega^{\text{out}})\|^2}}{\tilde{L}^{\text{out}}(\mathbf{p}, \omega^{\text{out}})}.$$

This heuristic automatically decreases the radius of the cache record where indirect illumination changes quickly. Although the method is formally not correct, it gives plausible results in practice as shown in Figure 3.

Finite differences are used to estimate the derivatives of outgoing radiance. For each incoming radiance sample, we displace the ray origin along local x and y coordinate axes, update the ray direction, and re-evaluate the BRDF. We use the updated BRDF values to compute the approximation of the outgoing radiance at the displaced positions. Finite differences are used to estimate two derivatives $\vec{\nabla}_x \tilde{L}^{\text{out}}(\mathbf{p}, \omega^{\text{out}})$ and $\vec{\nabla}_y \tilde{L}^{\text{out}}(\mathbf{p}, \omega^{\text{out}})$. For the sake of simplicity we assume the incoming radiance does not change with the displacement (which may not be true if the contributing surface is glossy [JZJ08]).

5. Results

We compare our algorithm (spatial directional radiance caching, SDRC) with the original radiance caching algo-

rithm (SHRC) and Monte Carlo importance sampling (MC). The algorithms were implemented as plug-ins in the PBRT ray tracer [PH04]. All images were rendered on a Mac-Book Pro with Intel Core 2 Duo 2.40GHz (using one core). No optimization using the GPU was used.

Images generated by the algorithms exhibits artifacts of very different kinds. Images generated using MC contain high frequency noise. On the other hand, SHRC suffers from low frequency error which can be seen as splotches in the image. The SDRC exhibits both types of errors in lesser amplitude. As a result of the different nature of the errors it is difficult to compare rendering times needed to obtain images of the same visual quality. Instead, we compare the image quality obtained by the algorithms for equal-time computation. For each image we show RMS error although this error measure may not be quite meaningful in terms of visual quality.

Images were rendered with the global illumination effects up to the fourth bounce indirect lighting. The first bounce was computed using the compared algorithms while photon map was used for the higher recursion level. Irradiance cache was used to compute diffuse indirect lighting. We used the following default settings for the algorithms. In SHRC, we used spherical harmonics order of 10. The maximum allowed caching error was set to $a = 0.25$ and the number of rays for hemisphere sampling is set to ensure the same computation time. In SDRC, the default maximum allowed error was $a = 0.22$. Both the SHRC and SDRC used the neighbor clamping heuristic to improve spatial record distribution [KBPv06].

The SHRC and SDRC algorithms render the image in two passes. The first pass, rendered with one camera ray per pixel, populates the spatial cache. The second pass then uses four camera rays per pixel to generate the image. The approximation error a is increased 1.3 times in the second pass to improve the smoothness of interpolated illumination.

Kettle Scene. Figure 5 shows renderings of a glossy kettle in a diffuse box. The kettle material is represented using anisotropic Ward's BRDF with the roughness of $\alpha_x = 0.18$ and $\alpha_y = 0.09$. The box walls are covered with a diffuse texture. The scene is lit by an area light source.

Uniform hemisphere sampling is not efficient for glossy surfaces with a narrow BRDF lobe. A lot of computation time can be saved using importance sampling as we do in the SDRC. The saved time can be used to set smaller the interpolation error for the SDRC, $a = 0.17$. Total number of records generated by SHRC is 2.53; for SDRC, it is 4.67k. Rendering using the SDRC shows 1.58 times smaller RMS error than the SHRC and 1.26 times smaller than MC. See table 1 for details.

Figure 4 shows the RMS error produced by the SHRC and the SDRC as a function of the surface finish. For the sake of simplicity we use Phong's BRDF model instead of

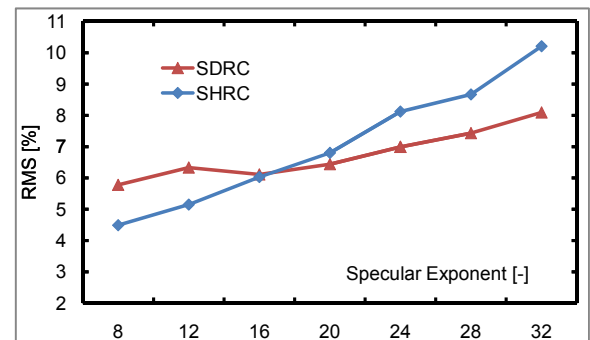
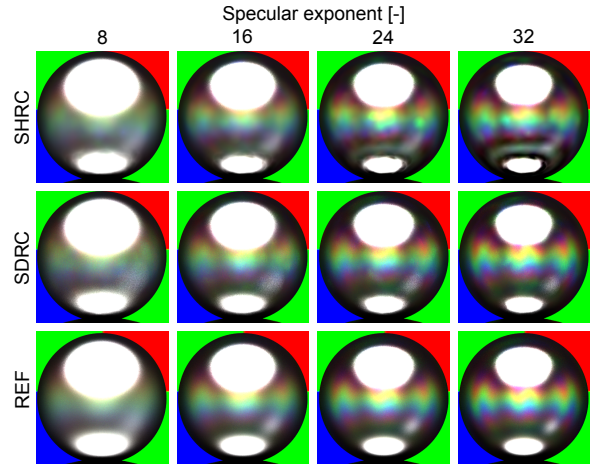


Figure 4: Renderings for different surface finish. The bottom image shows the RMS dependency on the Phong's lobe exponent. The SHRC is suitable for the indirect term computation up to the lobe exponent of 16. For higher exponents, it blurs out the reflection thereby altering the material perception. Our algorithm, on the other hand, provides a sharper image for higher lobe exponent. The top image shows a visual example of the RMS dependency for the lobe exponent. The details show a glossy reflection of the checkered walls on the handle of the teapot lid in Figure 5.

anisotropic one for the kettle material. We can see that the SHRC works well for a low BRDF lobe exponent. Importance sampling has no advantage compared with sampling the whole hemisphere. However, the bigger the exponent, the bigger the RMS error of SHRC. Spherical harmonics are not able to approximate a narrow BRDF lobe accurately. In this case, it is preferable to use the SDRC for the exponent higher than 16.

Chess scene. Figure 6 shows the renderings of the chess scene. White chess pieces are represented using Ward's BRDF model with the roughness of $\alpha_x = 0.10$ and $\alpha_y = 0.16$. Black ones are represented using Phong's BRDF

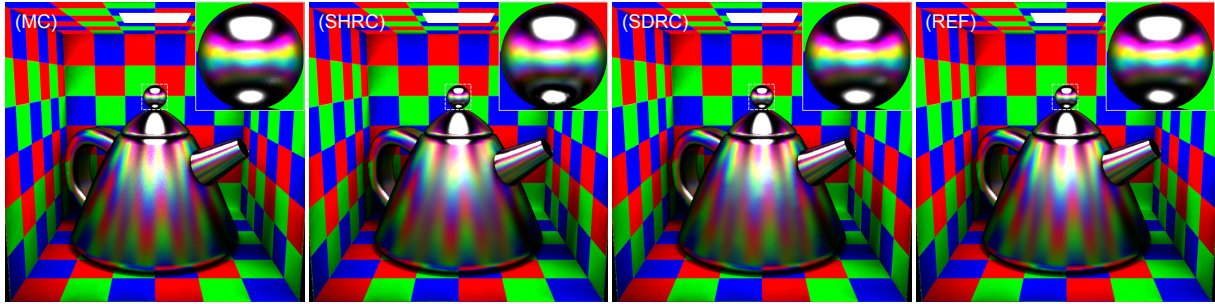


Figure 5: Equal-time comparison of the rendering quality achieved using the MC, SHRC and SDRC algorithms. The images were rendered at a resolution of 800×800 in approximately the same time, 398 seconds. Indirect lighting computation on glossy surfaces up to the 4th bounce took approximately 171 seconds out of the total time. We can see that SDRC is able to faithfully simulate glossy BRDF on the kettle while the SHRC provides a blurrier image. The image rendered using MC exhibits a high noise level. Note that the walls of the Cornell Box are diffuse only. The irradiance cache was used to compute the indirect illumination term on them.

Scene	Method	#Sec rays/ pixel	#Records	#Rays/ record	#Additional rays	Memory	RMS	Time	Total time
Teapot kettle	MC	62.6	-	-	-	-	1.70	168	385
	SHRC	47.7	2.53k	4608	-	9.1M	2.13	174	391
	SDRC	12.8	4.67k	512	101k	19.9M	1.35	171	388
Chess scene	MC	90.7	-	-	-	-	6.56	366	469
	SHRC	54.0	16.0k	2244	-	57.6M	10.25	369	472
	SDRC	22.2	26.1k	512	306k	109.5M	5.50	360	463
Flamingos	MC	41.6	-	-	-	-	3.23	416	625
	SHRC	26.5	14.7k	1986	-	52.9M	4.98	407	616
	SDRC	11.0	22.2k	512	638k	96.2M	2.73	410	619

Table 1: Rendering settings and statistics for the example scenes. The columns list the number of secondary rays traced per pixel, the number of spatial records generated, the number of rays traced to create a new cache record, the number of rays traced to L -trees updation and the memory requirements. The rightmost section of the table shows the RMS error, the time spent on the indirect term computation on glossy surfaces and the total rendering time. The difference consists of building the photon map, primary ray casting, direct lighting computation and the irradiance caching.

model with the lobe exponent of 30. The chessboard is represented using the same model with the lobe exponent of 14. The scene is lit by 5 spot lights and by an environment map of the sky.

The chess scene presents a challenge for the SHRC because of a lot of glossy and curved surfaces. On such surfaces, cache records cannot be reused at many pixels. In addition, the interpolation of the cache record require the costly rotation. See table 1 for the rendering settings and the number of the records generated by the caching algorithms. Note that the rendering using the SDRC shows 1.86 times smaller the RMS error than SHRC and 1.19 times smaller than MC.

Flamingos. Figure 7 shows renderings of a scene with glossy flamingos. The flamingos are represented using the Ward's BRDF model with the roughness in the range of 0.08 to 0.20. Other surfaces are purely diffuse. Table 2 shows the scalability of the rendering time to the number of directional

rays N used. The rendering time spent for indirect lighting computation on glossy surfaces is directly proportional to N .

SDRC	N	32	64	128	256	512
	Time	43	75	150	273	410
	RMS	5.48	4.85	3.88	3.19	2.81
MC	Time	50	77	156	266	416
	RMS	8.71	7.32	5.00	3.76	3.23

Table 2: Rendering time scalability. The rows list the number of directional samples used, rendering time spent for indirect lighting computation on glossy surfaces and the RMS error for both the SDRC and MC algorithms.

6. Discussion and limitations

View dependency. Unlike in the original radiance caching, the cache records generated by our algorithm contain use-

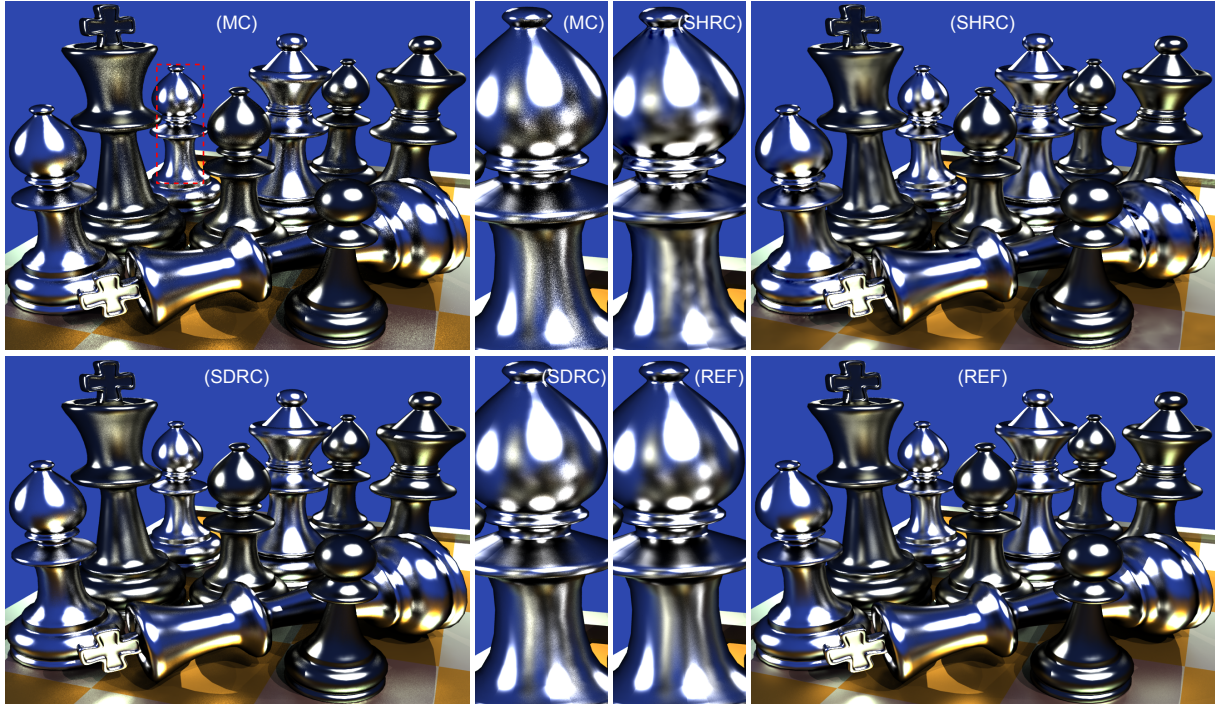


Figure 6: Renderings of the chess scene. Images were rendered at a resolution of 1024×768 in approximately the same time, 468 seconds. The indirect term computation on glossy surfaces took 365 seconds. Note the sharper glossy reflection of black pieces on the white chess piece in the details. Chess pieces courtesy of T. Hachisuka.

ful information only within some parts of the hemisphere—they are view-dependent. Hence, it may seem problematic to reuse them for rendering new views of the scene. However, since new radiance samples are added to the L -trees on the fly as needed, our algorithm behaves in a view-independent manner overall.

Interpolation. The idea of sparse illumination computation and interpolation is based on the assumption that indirect illumination changes slowly on surfaces. However, as the shininess of the reflections goes up, this assumption is less valid. For sharp reflections, Monte Carlo importance sampling performs better than our algorithm.

Gradients. In irradiance and radiance caching, the use of rotation and translation gradients of the incoming illumination significantly improves the interpolation quality. The *rotation* gradient is not necessary in our algorithm due to the way we access individual radiance samples. We have implemented the *translation* gradient according to [KGPB05]. However, the gradient-based interpolation becomes fairly involved. In our scenes, the computational overhead did not pay for the image quality improvement, which is why we decided against the use of translation gradients. Instead, we generate the image in two passes. Increasing the allowed in-

terpolation error a in the second pass generates a smoothed image even without the use of gradients.

Supported Materials. The original radiance caching algorithm is difficult to use on spatially varying glossy surfaces since the scene BRDFs have to be projected onto the spherical harmonics basis. Our algorithm does not require any special BRDF representation, although many sudden changes in material properties may reduce its efficiency due to frequent L -tree updates. Furthermore, spatial directional caching relies on the availability of an efficient and effective sampling procedure for the BRDFs—it cannot be used directly with measured BRDF data without first fitting a model that can be sampled.

7. Conclusion

This paper proposes a new algorithm for indirect lighting computation on glossy surfaces. The algorithm adopts the lazy illumination evaluation scheme used in the irradiance and radiance caching algorithms and extends it from the spatial to the directional domain. Explicit storage of directional incoming radiance samples allows us to exploit BRDF importance sampling for noise reduction and still retain the overall view-independent nature of the algorithm.

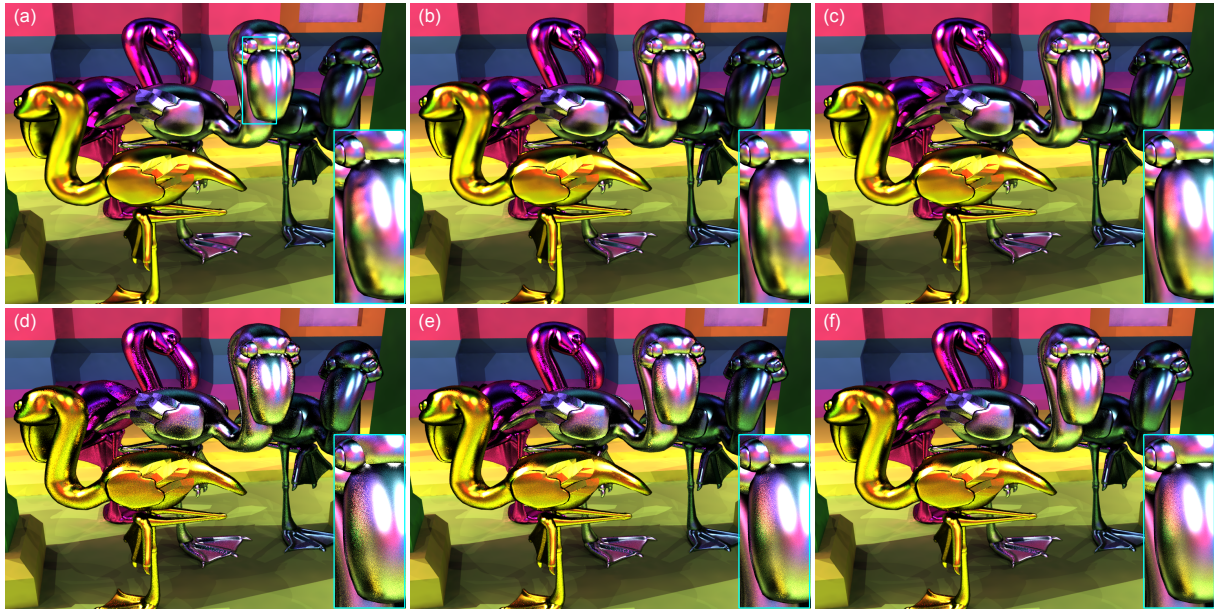


Figure 7: Rendering time scalability. The images were rendered in at a resolution of 1024×768 . Top images shows rendering quality of the SDRC for the value of N set to $N = 64, 128$ and 256 . Bottom images were rendered using the MC in approximately the same time for the indirect term computation on glossy surfaces, as the top images, 76, 153 and 269 seconds respectively.

Our new caching algorithm outperforms the original radiance caching for scenes with shiny surfaces, where radiance caching produces blurring of reflections or banding artifacts. Compared to Monte Carlo importance sampling, our algorithm produces less noisy images in the same time. The main disadvantages of our algorithm are higher memory demands and potentially difficult parallelization due to the continual updates of cache records.

In future work, we want to devise a more accurate interpolation criterion for glossy surfaces. Not only should such a criterion adapt to the rate of change of the indirect illumination but also to the surface reflectance properties. In addition we would like to investigate the correlation of illumination coherence in the spatial and directional domains and address flickering in animation rendering.

Acknowledgements

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics), the Internal grant No. CTU0807013 of the Czech Technical University, and the Marie Curie grant number PIOF-GA-2008-221716. Thanks to Jiří Bittner for valuable discussions.

References

- [AFO05] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Fast and detailed approximate global illumination by irradiance decomposition. *ACM Transactions on Graphics* 24, 3 (Aug. 2005), 1108–1114.
- [BGB08] BROUILLAT J., GAUTRON P., BOUATOUCH K.: Photon-driven irradiance cache. *Computer Graphics Forum (Proc. of Pacific Graphics)* 27, 7 (2008), 1971–1978.
- [BSH02] BEKAERT P., SBERT M., HALTON J.: Accelerating path tracing by re-using paths. In *Rendering Techniques 2002: 13th Eurographics Workshop on Rendering* (June 2002), pp. 125–134.
- [CAE08] CLINE D., ADAMS D., EGBERT P.: Table-driven adaptive importance sampling. *Computer Graphics Forum (Proc. of Eurographics Symposium on Rendering)* 27, 4 (2008).
- [CAM08] CLARBERG P., AKENINE-MÖLLER T.: Practical product importance sampling for direct illumination. *Computer Graphics Forum (Proc. of Eurographics)* 27, 2 (2008), 681–690.
- [CSH08] CASTRO F., SBERT M., HALTON J. H.: Efficient reuse of paths for random walk radiosity. *Computers & Graphics* 32, 1 (Feb. 2008), 65–81.
- [GBP07] GAUTRON P., BOUATOUCH K., PATTANAİK S.: Temporal radiance caching. *IEEE Transactions on Visualization and Computer Graphics* 13, 5 (2007).
- [GKBP05] GAUTRON P., KŘIVÁNEK J., BOUATOUCH K., PATTANAİK S. N.: Radiance cache splatting: A GPU-friendly global illumination algorithm. In *Rendering Techniques (Proc. of Eurographics Symposium on Rendering)* (2005), pp. 55–64.
- [HDMS03] HAVRAN V., DAMEZ C., MYSZKOWSKI K., SEIDEL H.-P.: An efficient spatio-temporal architecture for animation

- rendering. In *Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering* (June 2003), pp. 106–117.
- [HPB07] HAŠAN M., PELLACINI F., BALA K.: Matrix row-column sampling for the many-light problem. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 26, 3 (2007).
- [HS98] HEIDRICH W., SEIDEL H.-P.: View-independent environment maps. In *Proc. of Graphics Hardware* (1998).
- [JDZJ08] JAROSZ W., DONNER C., ZWICKER M., JENSEN H. W.: Radiance caching for participating media. *ACM Trans. Graph.* 27, 1 (2008).
- [Jen01] JENSEN H. W.: *Realistic Image Synthesis Using Photon Mapping*. AK Peters, Ltd., July 2001.
- [JZJ08] JAROSZ W., ZWICKER M., JENSEN H. W.: Irradiance gradients in the presence of participating media and occlusions. *Computer Graphics Forum (Proc. of Eurographics Symposium on Rendering)* 27, 4 (2008).
- [KBPv06] KŘIVÁNEK J., BOUATOUCH K., PATTANAIK S., ŽÁRA J.: Making radiance and irradiance caching practical: Adaptive caching and neighbor clamping. In *Rendering Techniques 2006 (Proc. of Eurographics Symposium on Rendering)* (2006), pp. 127–138.
- [Kel97] KELLER A.: Instant radiosity. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 49–56.
- [KG09] KŘIVÁNEK J., GAUTRON P.: *Practical Global Illumination with Irradiance Caching*. Morgan-Claypool, 2009.
- [KGPB05] KŘIVÁNEK J., GAUTRON P., PATTANAIK S., BOUATOUCH K.: Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics* 11, 5 (2005).
- [LW95] LAFORTUNE E. P., WILLEMS Y. D.: A 5D tree to reduce the variance of Monte Carlo ray tracing. In *Rendering Techniques (Proc. of the Sixth Eurographics Workshop on Rendering)* (1995), pp. 11–20.
- [PBSP08] PEGORARO V., BROWNLEE C., SHIRLEY P. S., PARKER S. G.: Towards interactive global illumination effects via sequential Monte Carlo adaptation. In *Proc. of the 3rd IEEE Symposium on Interactive Ray Tracing* (2008), pp. 107–114.
- [PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2004.
- [PWP08] PEGORARO V., WALD I., PARKER S. G.: Sequential Monte Carlo adaptation in low-anisotropy participating media. *Computer Graphics Forum (Proc. of Eurographics Symposium on Rendering)* 27, 4 (2008).
- [TL04] TABELLION E., LAMORLETTE A.: An approximate global illumination system for computer generated films. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 23, 3 (2004), 469–476.
- [WABG06] WALTER B., ARBREE A., BALA K., GREENBERG D. P.: Multidimensional lightcuts. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 25, 3 (2006), 1081–1088.
- [WH92] WARD G. J., HECKBERT P. S.: Irradiance gradients. In *Eurographics Workshop on Rendering* (1992).
- [WNLH06] WANG R., NG R., LUEBKE D., HUMPHREYS G.: Efficient wavelet rotation for environment map rendering. In *Rendering Techniques (Proc. of Eurographics Symposium on Rendering)* (2006), pp. 173–182.
- [WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. In *Proc. of SIGGRAPH* (1988).