Radiance Caching for Efficient Global Illumination Computation

Jaroslav Křivánek, Pascal Gautron, Sumanta Pattanaik, and Kadi Bouatouch

Abstract— **In this paper we present a ray tracing based method for accelerated global illumination computation in scenes with low-frequency glossy BRDFs. The method is based on sparse sampling, caching, and interpolating radiance on glossy surfaces. In particular we extend the irradiance caching scheme proposed by Ward *et al.* [1] to cache and interpolate directional incoming radiance instead of irradiance. The incoming radiance at a point is represented by a vector of coefficients with respect to a spherical or hemispherical basis. The surfaces suitable for interpolation are selected automatically according to the roughness of their BRDF. We also propose a novel method for computing translational radiance gradient at a point.**

Index Terms— **Global illumination, ray tracing, spherical harmonics, directional distribution.**

I. Introduction

MONTE Carlo ray tracing is the method of choice for computing images of complex environments with global illumination [2]. It produces high quality images, handles general lighting phenomena and scene descriptions, and scales well with the scene size. Even for the radiosity method, high quality images are created by the final gathering, often making use of Monte Carlo (MC) ray tracing [3].

MC ray tracing is, however, expensive when it comes to computing indirect illumination on surfaces with low frequency BRDFs (bi-directional reflectance distribution functions). Low-frequency BRDFs are wide and therefore too many rays have to be traced to get a reasonably precise estimate of the outgoing radiance at a point. Fortunately, there is a high degree of coherence in the outgoing radiance field on those surfaces [1], [4]–[6], which demonstrates itself as a smooth indirect illumination. The coherence of radiance can be exploited by interpolation [1], [7] to obtain a significant performance gain.

Our goal is to accelerate global illumination computation in the presence of surfaces with low-frequency glossy BRDFs in the context of MC ray tracing. We achieve it by sparse sampling, caching, and interpolating radiance on those surfaces. In particular we extend Ward *et al.*'s irradiance caching scheme [1], [8] to glossy surfaces. Irradiance caching is based on the observation that reflected radiance on diffuse surfaces due to indirect illumination changes very slowly with position. This can be, however, generalized for all surfaces with low-frequency BRDFs, even if they are glossy. Motivated by this observation, we extend Ward *et al.*'s work to cache and interpolate the *directional* incident radiance instead of the irradiance. This provides the necessary information for the illumination integral evaluation (Eq. 1) in the presence of a glossy BRDF and allows for radiance caching and interpolation on glossy surfaces. We dub the new method *radiance caching*.

The incoming radiance at a point is represented by spherical or hemispherical harmonics (HSH, see Appendix I) and radiance interpolation reduces to interpolating the HSH coefficients. The illumination integral evaluation reduces to a dot product of the interpolated incoming radiance coefficients and the coefficients of the BRDF representation. The dot product is fast to compute and saves many BRDF evaluations that would have to be used in MC importance sampling.

We enhance the interpolation quality by the use of translational gradients. We propose novel methods for computing gradients that are more general than the method of Ward and Heckbert [8].

Radiance caching shares all the advantages of the original Ward's work. Computation is concentrated in visible parts of the scene; no restrictions are imposed on the scene geometry; implementation and integration with a ray tracer is easy. Our approach can be directly used with any measured BRDF represented by spherical or hemispherical harmonics.

This paper extends the initial description of radiance caching given in [9]. The main contributions of this paper are extension of the irradiance caching algorithm

Manuscript received July XXX, 2004.

J. Křivánek is with the Czech Technical University, IRISA/INRIA Rennes, and the University of Central Florida. E-mail: jkrivane@irisa.fr

P. Gautron is with IRISA/INRIA Rennes and the University of Central Florida. E-mail: pgautron@irisa.fr

S. Pattanaik is with the University of Central Florida. E-mail: sumant@cs.ucf.edu

K. Bouatouch is with IRISA/INRIA Rennes. E-mail: kadi@irisa.fr

to support glossy surfaces, an automatic method for selecting BRDFs for which radiance caching is suitable, new methods for computing translational radiance gradient at a point, and integration of radiance caching in a distribution ray tracer.

The rest of the paper is organized as follows. Section II summarizes the related work. Section III gives an overview of how radiance caching works and how it is integrated in a distribution ray tracer. Section IV details different aspect of radiance caching. Section V presents the results. Section VI discusses various topics not covered in the algorithm description. Section VII concludes the paper and summarizes our ideas for future work.

II. Related work

a) Interpolation: Interpolation can be used in global illumination whenever there is a certain level of smoothness in the radiometric quantity being computed. All radiosity methods use interpolation in the form of surface discretization. They adapt to the irradiance smoothness by adaptive geometry subdivision, *e.g.* [10]–[12]

In the context of MC ray tracing many approaches have been proposed for *screen space interpolation* [5], [13]–[15]. The goal of these methods is to display an approximate solution quickly, possibly at interactive frame rates. However, they do not accelerate the computation of the final high quality solution, which is the objective of our work. *Object space interpolation* has also been used for the purpose of interactive previewing [16], [17]. Sparse sampling and interpolation for high quality rendering was used in [7] and [1]. The approach of Bala et al. [7] is suitable only for deterministic ray tracing. Ward et al. [1] use interpolation only for diffuse surfaces. Our approach extends this work to support caching and interpolation of the directional incoming radiance on glossy surfaces.

b) Caching Directional Distributions: Caching the directional representation of radiometric quantities has been used to extend the radiosity method to support glossy surfaces, *e.g.* [18]–[25].

Directional distribution caching has also been used in the context of MC ray tracing [6], [26], [27]. In those papers, the incoming radiance at a point is represented by keeping the list of radiance samples used for hemisphere sampling. Slusallek et al. [6] and Kato [26] use reprojection of the samples for improved interpolation. Tawara et al. [27] selectively update the sample list in time to exploit temporal coherence. Slusallek et al. and Tawara et al. mention the suitability of their representation for accelerating rendering with glossy surfaces. The papers discuss only the diffuse case, though. Storing light particles in the scene can also be thought of as caching the directional distribution, since the incoming direction is usually retained with the particle [28], [29].

c) Spherical Function Representation: A suitable representation of functions on a (hemi)sphere is necessary for incoming radiance caching. Piecewise constant representation [6], [24], [26], [27] is simple but prone to aliasing and usually very memory demanding. Unless higher order wavelets are used, even wavelet representation [21]–[23], [25], [30] does not remove the aliasing problems. But with higher order wavelets the mathematics becomes complicated and hence discourages their use.

Spherical Harmonics [18], [20], [31]–[37] remove the aliasing problem and are efficient for representation of low-frequency functions. However, representation of sharp functions require large number of coefficients and ringing artifacts might appear. Hemispherical harmonics [9] are better suited for representing functions on a hemisphere. Basis functions very similar to spherical harmonics are Zernike polynomials [38], [39] and hemispherical harmonics of Makhotkin [40]. Unlike for spherical harmonics, the rotation matrices are not available for these basis functions. We choose spherical and hemispherical harmonics for their good anti-aliasing properties, low storage cost and ease of use.

d) Illumination Gradient Computation: Arvo [41] computes the Jacobian of the vector irradiance at a point due to partially occluded polygonal emitters of constant radiosity. Holzschuch and Sillion [42] handle polygonal emitters with arbitrary radiosity. Ward and Heckbert [8] compute irradiance gradient at a point using the information from stochastic hemisphere sampling. Our gradient computation is also based on hemisphere sampling. We use gradients to improve the smoothness of the radiance interpolation. One of the algorithms that we use for gradient computation was independently developed by Annen *et al.* [43].

e) Irradiance Caching: Ward et al. [1] propose irradiance caching as a means of computing indirect diffuse inter-reflections in a distribution ray tracer [2]. They exploit the smoothness of the indirect illumination by sampling the irradiance sparsely over surfaces, caching the results and interpolating them.

For each ray hitting a surface, the irradiance cache is queried. If one or more irradiance records are available, the irradiance is interpolated. Otherwise a new irradiance record is computed by sampling the hemisphere and is added to the cache. In this way, the cache gets filled lazily, progressively in a view dependent manner. As it gets filled, more and more irradiance computations can be carried out by interpolation. Ward uses an octree for storing the irradiance records. In [8] the interpolation quality is improved by the use of irradiance gradients.

We retain the basic structure of the original algorithm, but each record stores the incoming radiance function over the hemisphere. This allows interpolation to be applied to glossy surfaces.

III. Algorithm Overview

Radiance caching is a part of a distributed ray tracing approach to global illumination. Rays are cast from the camera into the scene through pixels. At every ray-surface intersection, the outgoing (reflected) radiance has to be computed by evaluating the *illumination integral*:

$$L(\theta_o,\phi_o) = \\ = \int_0^{2\pi}\int_0^{\pi/2} L_i(\theta_i,\phi_i) f(\theta_i,\phi_i,\theta_o,\phi_o)\cos\theta_i \sin\theta_i \mathrm{d}\theta_i \mathrm{d}\phi_i, \tag{1}$$

where L is the outgoing radiance, L_i is the incoming radiance and f is the BRDF. The integral is split into parts and each of them is solved by a different technique:

- Direct illumination uses deterministic method for point light sources and area sampling for area light sources [44].
- Perfect specular reflections/refractions are solved by tracing a single deterministic secondary ray.
- Ward's irradiance caching computes the indirect diffuse term for *purely* diffuse surfaces.
- Two different techniques may be used for glossy surfaces.
  Low-frequency BRDF. Our radiance caching computes the indirect glossy *and diffuse* terms.
  High-frequency BRDF. MC importance sampling computes the indirect glossy term and irradiance caching computes the indirect diffuse term.

Radiance caching is not used for high frequency BRDFs since many HSH coefficients would be needed. High-frequency BRDFs are also well localized and importance sampling provides good accuracy with few secondary rays. The distinction between low- and high-frequency BRDF is done automatically as described in Section IV-A. Steps of the rendering algorithm related to radiance caching are shown in Figure 1.

The i-th *radiance cache record* contains:

- position $\mathbf{p}_i$,
- local coordinate frame $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{n}_i)$,
- HSH coefficient vector Λ_i representing the incoming radiance,
- two derivative vectors $\frac{\partial \Lambda_i}{\partial x}$ and $\frac{\partial \Lambda_i}{\partial y}$ representing the translational gradient,
- harmonic mean distance R_i of objects visible from $\mathbf{p}_i$.

```
// preprocessing - BRDF conversion
for (every surface in the scene) do
  if (surface suitable for radiance caching) then
    Compute and store the HSH representation of the surface's
    BRDF.
  end if
end for

// rendering with radiance caching
for (every ray-surface intersection $\mathbf{p}$) do
  Retrieve the HSH representation of the BRDF at $\mathbf{p}$.
  if ( HSH representation of the BRDF not available) then
    // high-frequency BRDF
    MC importance sampling computes the indirect glossy term.
    Irradiance caching computes the indirect diffuse term.
  else
    // low-frequency BRDF; use radiance caching
    if ( radiance cache records exist near $\mathbf{p}$ ) then
      Compute the HSH coefficients of the incident radiance at
      $\mathbf{p}$ by gradient based interpolation.
    else
      Compute incident radiance at $\mathbf{p}$ by sampling the hemi-
      sphere above $\mathbf{p}$.
      Compute HSH coefficients of the incoming radiance.
      Compute the translational gradient of the coefficients.
      Store the new radiance record in the radiance cache.
    end if
    Compute the outgoing radiance at $\mathbf{p}$ as the dot product of
    the coefficient vector of the incoming radiance and that of
    the BRDF.
  end if
end for
```

Fig. 1. Outline of the radiance caching algorithm.

In the rest of the paper Λ denotes a HSH coefficient vector and λ_l^m denotes the coefficients, that is $\Lambda = \{\lambda_l^m\}$.

Each record stores an incident radiance function which can be reused for different viewpoints. The records are stored in an octree as described by Ward et al. [1].

IV. Radiance Caching Details

A. BRDF Representation

We precompute the HSH representation of surface BRDFs using the method of Kautz et al. [35] which we briefly describe here.

We discretize the hemisphere of outgoing directions. For each discrete outgoing direction (θ_o, ϕ_o) we use HSH to represent the BRDF values over the hemisphere of incoming directions (*i.e.* one HSH coefficient vector per one discretized (θ_o, ϕ_o)). The n-th order representation of a cosine weighted[1] BRDF $f_{(\theta_o,\phi_o)}$ for an outgoing

[1] All BRDFs are premultiplied by the cosine term $\cos\theta_i$ of the illumination integral (Eq. 1) before computing the HSH representation.

direction (θ_o, ϕ_o) is

$$f_{(\theta_o,\phi_o)}(\theta_i, \phi_i) \approx \sum_{l=0}^{n} \sum_{m=-l}^{l} c_l^m(\theta_o, \phi_o) H_l^m(\theta_i, \phi_i) \ , \quad (2)$$

where

$$c_l^m(\theta_o, \phi_o) = \\ = \int_0^{2\pi}\int_0^{\pi/2} f(\theta_o, \phi_o, \theta_i, \phi_i) H_l^m(\theta_i, \phi_i) \sin\theta_i \mathrm{d}\theta_i \mathrm{d}\phi_i. \quad (3)$$

H_l^m are the hemispherical harmonics basis functions (Appendix I). We sample the outgoing hemisphere for (θ_o, ϕ_o) using the parabolic parametrization proposed by Heidrich and Seidel [45].

f) Adaptive BRDF Representation: BRDFs are converted to HSH before the rendering starts. For each outgoing direction, the representation of $f_{(\theta_o,\phi_o)}$ uses the minimum order n sufficient for not exceeding the user specified maximum error. The error is measured as described in [33]. If no order $n < n_{\max}$ is sufficient for the specified error, the HSH representation is discarded: radiance caching will not be used for that BRDF and that outgoing direction. After applying this procedure, only low-frequency BRDFs are represented using HSH and radiance caching is used for them. $n_{\max}$ is user specified; we use $n_{\max} = 10$ for our examples. The aim is to have $n_{\max}$ such that a BRDF is classified as low-frequency if and only if radiance caching is more efficient than MC importance sampling. The required HSH order may vary significantly with the outgoing direction, higher order (more coefficients) is usually required for grazing angles (see Figure 2).

B. Incoming Radiance Computation

Whenever interpolation is not possible at $\mathbf{p}$ because none of the radiance cache records meet the error criterion (Section IV-D), a new record has to be computed and stored in the cache.

We represent the incoming radiance function L_i at a point $\mathbf{p}$ by a vector of HSH coefficients $\Lambda = \{\lambda_l^m\}$:

$$L_i(\theta, \phi) \approx \sum_{l=0}^{n} \sum_{m=-l}^{l} \lambda_l^m H_l^m(\theta, \phi), \quad (4)$$

where n is the order of HSH representation. The representation coefficients λ_l^m for an known, analytical L_i would be computed with the integral

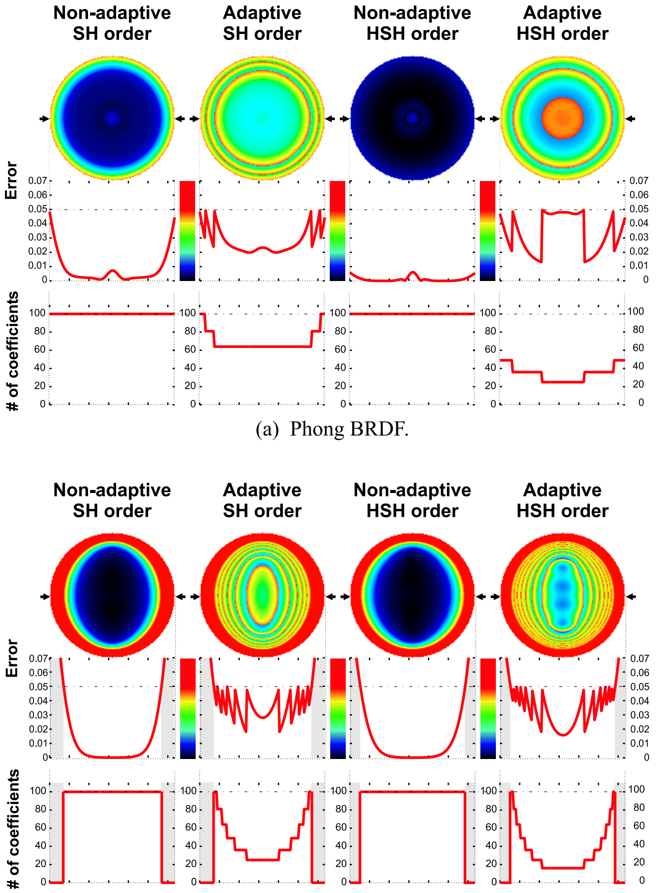
$$\lambda_l^m = \int_0^{2\pi} \int_0^{\pi/2} L_i(\theta, \phi) H_l^m(\theta, \phi) \sin\theta \mathrm{d}\theta \mathrm{d}\phi. \quad (5)$$

Our knowledge of L_i is based only on sampling (ray casting). Hence, we compute λ_l^m by a MC quadrature with uniform sampling of the hemisphere of incoming directions:

$$\lambda_l^m = \frac{2\pi}{N} \sum_{k=1}^{N} L_i(\theta_k, \phi_k) H_l^m(\theta_k, \phi_k), \quad (6)$$

where $L_i(\theta_k, \phi_k)$ is the incoming radiance coming from the sampled direction (θ_k, ϕ_k) and N is the number of sampled directions. Direct light source contribution is excluded from L_i. We use a fixed number N of directions but adaptive hemisphere sampling [47], [48] is desirable.

The HSH order n for the incoming radiance representation is equal to the order of the BRDF representation



(a) Phong BRDF.

(b) Anisotropic Ward BRDF.

Fig. 2. Adaptive BRDF representation for (a) Phong BRDF with exponent $h = 15$ and (b) anisotropic Ward BRDF [46] with $k_d = 0$, $k_s = 1$, $\alpha_x = 0.6$, $\alpha_y = 0.25$. The color disks represent BRDF error for different outgoing directions (θ_o, ϕ_o). Directions are mapped on the disk with the parabolic parametrization. (One can imagine the disks as looking at the hemisphere from the top.) The graphs represent one scanline from the disk images (*i.e.* fixed y and varying x component of the outgoing direction). Adaptive BRDF representation (columns 2 and 4) chooses the minimum number of coefficients sufficient for not exceeding the user specified maximum error (here 5%). For the Ward BRDF, radiance caching is not used for some directions, since the error would be too high. Hemispherical harmonics (columns 3 and 4) give lower errors or fewer coefficients than spherical harmonics (columns 1 and 2).

at $\mathbf{p}$. This cuts off high frequencies from the incoming radiance. The approach is justified by the fact that a low-frequency BRDF acts as a low-pass filter on the incoming radiance [34].

C. Translational Gradient Computation

The coefficients λ_l^m of the HSH projection of the incoming radiance at a point are computed with Eq. (6). We want to compute the translational gradient $\nabla\lambda_l^m$ for each λ_l^m, *i.e.* the rate of change of λ_l^m with a differential displacement of the sampling point $\mathbf{p}$ on the surface. We will use the gradient to improve the interpolation smoothness.

Our original intent was to build on the gradient computation of Ward and Heckbert [8] since the form of Equation (6) is similar to Ward and Heckbert's formula for computing irradiance (Equation (2) in [8]). The difference is in the choice of the probability density for the hemisphere sampling: we use uniform density whereas Ward and Heckbert use cosine density. Another difference is the weighting function for incoming radiance samples: we use $H_l^m(\theta,\phi)$ while no weight is used in [8]. We did not succeed in extending Ward and Heckbert's gradient computation to handle our case since their method tightly couples the cosine probability density and the weighting.

Instead, we have developed two new methods for computing translational gradient $\nabla\lambda_l^m$. The first, numerical, displaces the center of the hemisphere. The second, analytic, is based on differentiating the terms of Eq. (6). In both cases we compute the gradient $\nabla\lambda_l^m = \left[\frac{\partial\lambda_l^m}{\partial x}, \frac{\partial\lambda_l^m}{\partial y}, 0\right]$ by computing the partial derivatives $\partial\lambda_l^m/\partial x$ and $\partial\lambda_l^m/\partial y$. The gradient is defined in the local coordinate frame at the point $\mathbf{p}$. The derivative with respect to z is not computed since we assume displacement only in the tangent plane. We compute the translational gradients simultaneously with the computation of the coefficients λ_l^m during the hemisphere sampling.

1) Numerical Gradient Computation: To compute the derivative $\partial\lambda_l^m/\partial x$ numerically, we displace the point $\mathbf{p}$, along the local X axis, by Δx to $\mathbf{p}'$ (Figure 3). For each Monte Carlo sample $L_i(\theta_k,\phi_k)$ we:

1) Compute the new direction (θ_k',ϕ_k') at $\mathbf{p}'$ as $(\theta_k',\phi_k') = \frac{\mathbf{q}_k-\mathbf{p}'}{r_k'}$. Here $\mathbf{q}_k$ is the point hit by the ray from $\mathbf{p}$ in direction (θ_k,ϕ_k) and $r_k' = \|\mathbf{q}_k-\mathbf{p}'\|$. We will also denote $r_k = \|\mathbf{q}_k-\mathbf{p}\|$. See Figure 3 for the various terms used here.
2) Compute the solid angle Ω_k' associated with the new direction (θ_k',ϕ_k'). The solid angle Ω_k associated with each direction in Equation (6) is uniform and equal to $2\pi/N$. With the displacement of the point $\mathbf{p}$, the solid angles no longer remain uniform. The change in solid angle is due to the change in distance $r_k = \|\mathbf{q}_k-\mathbf{p}\|$ and orientation of the surface at $\mathbf{q}_k$, as seen from the hemisphere center $\mathbf{p}$ or $\mathbf{p}'$. The solid angle before the displacement is

$$\Omega_k = \Delta A_k \frac{\cos\xi_k}{r_k^2} = \frac{2\pi}{N},$$

where ξ_k is the angle between the surface normal at $\mathbf{q}_k$ and the vector from $\mathbf{q}_k$ to $\mathbf{p}$. The area $\Delta A_k = \frac{2\pi}{N}\frac{r_k^2}{\cos\xi_k}$ is the part of the environment visible through Ω_k. It does not change with the displacement because we assume that the environment visible from $\mathbf{p}$ and $\mathbf{p}'$ is the same. After the displacement the solid angle subtended by ΔA_k becomes

$$\Omega_k' = \Delta A_k \frac{\cos\xi_k'}{r_k'^2} = \frac{2\pi}{N}\frac{r_k^2}{r_k'^2}\frac{\cos\xi_k'}{\cos\xi_k}.$$

We now estimate the coefficient $\lambda_l^{m\prime}$ at $\mathbf{p}'$ as

$$\lambda_l^{m\prime} = \frac{2\pi}{N}\sum_{k=1}^{N}\frac{r_k^2}{r_k'^2}\frac{\cos\xi_k'}{\cos\xi_k}\;L_i(\theta_k,\phi_k)\;H_l^m\left(\theta_k',\phi_k'\right),$$

and finally we compute $\partial\lambda_l^m/\partial x$ as

$$\frac{\partial\lambda_l^m}{\partial x} = \frac{\lambda_l^{m\prime}-\lambda_l^m}{\Delta x}.$$

The computation of $\partial\lambda_l^m/\partial y$ proceeds in a similar way. This completes the numerical estimation of the translational gradient $\nabla\lambda_l^m$ at the point of interest.

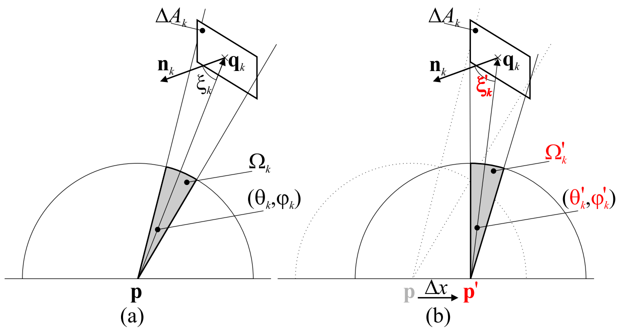


Fig. 3. Gradient computation by displacing the hemisphere center from $\mathbf{p}$ to $\mathbf{p}'$ ((a) before and (b) after the displacement). The quantities changing with the displacement are (shown in red): sample ray direction (θ_k,ϕ_k), the solid angle Ω_k associated with this sample, and the angle ξ_k between the sample direction and the surface normal at the hit point $\mathbf{q}_k$. Neither the hit point $\mathbf{q}_k$ nor the area ΔA_k visible through Ω_k change with the displacement.

2) Analytic Gradient Computation: We rewrite Equation (6) as

$$\lambda_l^m = \sum_{k=1}^{N} \Omega_k L_i(\theta_k, \phi_k) H_l^m(\theta_k, \phi_k), \tag{7}$$

with $\Omega_k = \frac{2\pi}{N}$ for uniform hemisphere sampling. We have seen that Ω_k does not remain constant with displacement of $\mathbf{p}$ and therefore it has to be included in the sum and differentiated. The notation in this section is as above.

The partial derivative $\partial\lambda_l^m/\partial x$ is computed by differentiating the terms of the sum in Eq. (7)

$$\begin{aligned}\frac{\partial\lambda_l^m}{\partial x} &= \sum_{k=1}^{N} \frac{\partial}{\partial x}\Big(\Omega_k L_i(\theta_k, \phi_k) H_l^m(\theta_k, \phi_k)\Big) = \\ &= \sum_{k=1}^{N} L_i(\theta_k, \phi_k)\left(\frac{\partial\Omega_k}{\partial x} H_l^m(\theta_k, \phi_k) + \Omega_k \frac{\partial H_l^m(\theta_k, \phi_k)}{\partial x}\right)\end{aligned} \tag{8}$$

The derivative of the basis function is

$$\frac{\partial H_l^m(\theta_k, \phi_k)}{\partial x} = \frac{\partial\theta_k}{\partial x}\frac{\partial H_l^m(\theta_k, \phi_k)}{\partial\theta_k} + \frac{\partial\phi_k}{\partial x}\frac{\partial H_l^m(\theta_k, \phi_k)}{\partial\phi_k}, \tag{9}$$

with [49]

$$\begin{aligned}\partial\theta_k/\partial x &= -\cos\theta_k \cos\phi_k / r_k, \\ \partial\phi_k/\partial x &= \sin\phi_k/(r_k \sin\theta_k).\end{aligned} \tag{10}$$

Those derivatives with respect to y would be

$$\begin{aligned}\partial\theta_k/\partial y &= -\cos\theta_k \sin\phi_k / r_k, \\ \partial\phi_k/\partial y &= -\cos\phi_k/(r_k \sin\theta_k).\end{aligned} \tag{11}$$

Derivatives $\partial H_l^m/\partial\theta_k$ and $\partial H_l^m/\partial\phi_k$ are given in the Appendix II.

The derivative of the solid angle Ω_k is

$$\frac{\partial\Omega_k}{\partial x} = \frac{\partial}{\partial x}\Delta A_k \frac{\cos\xi_k}{r_k^2} = \Delta A_k \frac{\partial}{\partial x}\frac{\cos\xi_k}{r_k^2},$$

The area $\Delta A_k = \frac{2\pi}{N}\frac{r_k^2}{\cos\xi_k}$ is the part of the environment visible through Ω_k. It does not change with the displacement. The change of $\cos\xi_k/r_k^2$ with the displacement of $\mathbf{p}$ is opposite to its change with the displacement of $\mathbf{q}_k = (q_x, q_y, q_z)$, *i.e.*

$$\frac{\partial}{\partial x}\frac{\cos\xi_k}{r_k^2} = -\frac{\partial}{\partial q_x}\frac{\cos\xi_k}{r_k^2}.$$

The derivative $\frac{\partial}{\partial q_x}\frac{\cos\xi_k}{r_k^2}$ can be computed with the assumption that $\mathbf{p}$ lies at the origin because only the relative position of $\mathbf{p}$ and $\mathbf{q}_k$ matters (see Figure 4).

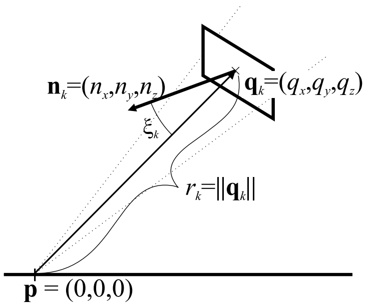


Fig. 4. Quantities in the computation of $\frac{\partial}{\partial q_x}\frac{\cos\xi_k}{r_k^2}$.

Since $\cos\xi_k = -\frac{\mathbf{n}_k\cdot\mathbf{q}_k}{r_k}$ and $r_k = \|\mathbf{q}_k\| = \sqrt{q_k^2 + q_y^2 + q_z^2}$ we have

$$\begin{aligned}\frac{\partial}{\partial q_x}\frac{\cos\xi_k}{r_k^2} &= -\frac{\partial}{\partial q_x}\frac{n_x q_x + n_y q_y + n_z q_z}{(q_x^2 + q_y^2 + q_z^2)^{3/2}} \\ &= -\frac{r_k^2 n_x - 3q_x(\mathbf{n}_k \cdot \mathbf{q}_k)}{r_k^5} \\ &= -\frac{r_k n_x + 3q_x \cos\xi_k}{r_k^4}.\end{aligned} \tag{12}$$

Here $\mathbf{n}_k = (n_x, n_y, n_z)$ is the surface normal at $\mathbf{q}_k$. Combining this result with $\Delta A_k = \frac{2\pi}{N}\frac{r_k^2}{\cos\xi_k}$, we get

$$\frac{\partial\Omega_k}{\partial x} = \frac{2\pi}{N}\frac{r_k n_x + 3q_x \cos\xi_k}{r_k^2 \cos\xi_k}. \tag{13}$$

3) Discussion: For the derivation of both numerical and analytic methods we assumed:

- The radiance $L_i(\theta_k, \phi_k)$ from the point $\mathbf{q}_k$ incident at $\mathbf{p}$ does not change with the displacement of $\mathbf{p}$.
- Visibility of ΔA_k, the small area around $\mathbf{q}_k$, does not change with the displacement of $\mathbf{p}$.

Though none of these assumptions is necessarily valid in all scenes, they are reasonable for small displacements.

The quantities changing with the displacement are:

- Direction (θ_k, ϕ_k) of the MC sample contributing to the sum (6). As a consequence, the value of $H_l^m(\theta_k, \phi_k)$ changes too.
- The size of the solid angle Ω_k through which the contribution to the sum (6) is brought. The change of Ω_k is due to the change in distance r_k and in the apparent orientation of the surface ΔA_k around the hit point $\mathbf{q}_k$.

The numerical and analytic methods are equivalent, their results are indistinguishable. The numerical method is easier to implement since we do not need to evaluate the basis function derivatives. The analytic method is numerically more stable near edges and corners and also slightly faster to evaluate.

Plugging Equations (13) and (9) into (8) we get the complete formula for $\partial\lambda_l^m/\partial x$. The formulas for

$\partial\lambda_l^m/\partial y$ are similar; only Equation (10) must be replaced by Equation (11).

A similar gradient calculation was also proposed in [29]. Their method disregards the change of Ω_k and hence does not give good results. The analytic method was also independently developed by Annen *et al.* [43].

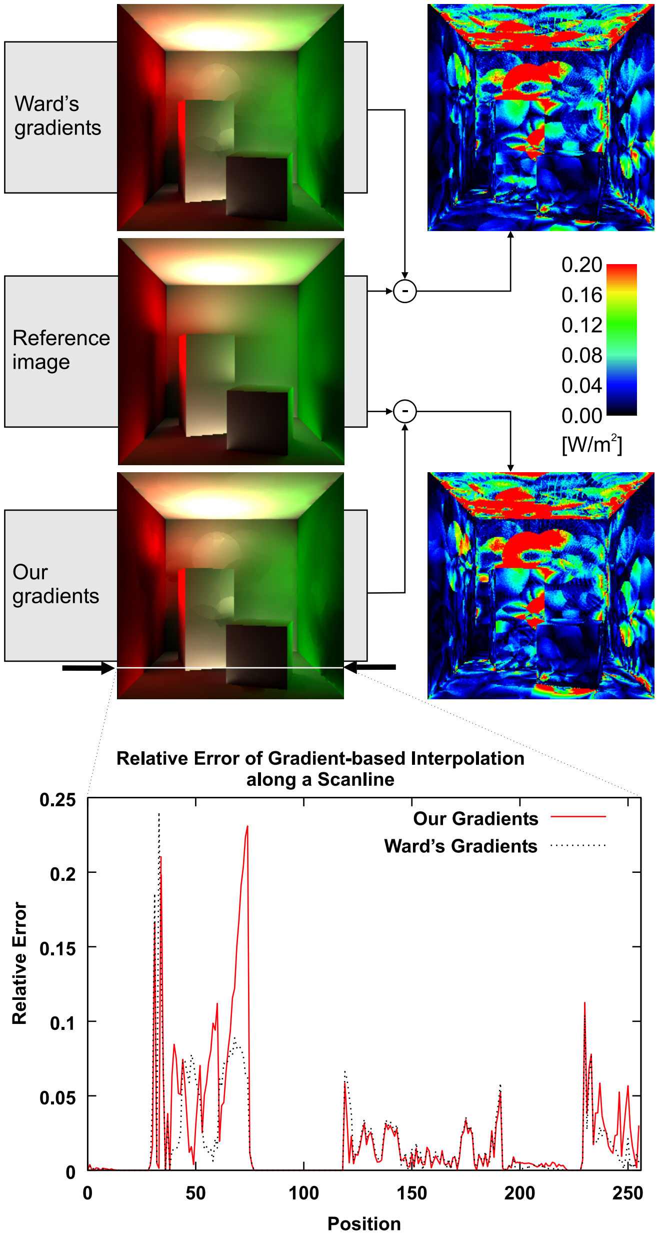


Fig. 5. Comparison of irradiance gradient computation. The scene is a diffuse Cornell box; only first bounce indirect illumination is computed. The color-coded images show the difference between the gradient-based interpolation and the reference solution (10,000 samples per hemisphere at each pixel). RMS error of the images is 0.125 for Ward's method and 0.131 for our method. The graph shows relative error of the interpolation along a single scanline as compared to the reference solution. Ward's method gives better results when there are surfaces seen at very sharp grazing angles from the sampling point. Otherwise our method gives slightly lower error.

4) Irradiance Gradient Computation: Note that both of the methods we propose can be still used if H_l^m is replaced by any other hemispherical function. We also do not rely on uniform hemisphere sampling. Any probability density $p(\theta,\phi)$ can be used for sampling. The only change is that the solid angle becomes $\Omega_k = \frac{1}{Np(\theta_k,\phi_k)}$ instead of $\Omega_k = \frac{2\pi}{N}$ used for the uniform sampling.

As an example we compute the irradiance gradient ∇E with a cosine-weighted hemisphere sampling. $H_l^m(\theta,\phi)$ is replaced by $\cos\theta$, the probability density of sampling in direction (θ,ϕ) is $p(\theta,\phi) = \frac{\cos\theta}{\pi}$ and therefore $\Omega_k = \frac{\pi}{N\cos\theta_k}$. The resulting formula for the analytic method is

$$\frac{\partial E}{\partial x} = \sum_{k=1}^{N} L_i(\theta_k,\phi_k)\left(\frac{\partial\Omega_k}{\partial x}\cos\theta_k - \frac{\pi}{N}\frac{\sin\theta_k}{\cos\theta_k}\frac{\partial\theta_k}{\partial x}\right)$$

with

$$\frac{\partial\Omega_k}{\partial x} = \frac{\pi}{N\cos\theta_k}\,\frac{r_k n_x + 3q_x\cos\xi_k}{r_k^2\cos\xi_k}.$$

We implemented this irradiance gradient computation method and that of Ward and Heckbert and we compared them on a sample scene (Figure 5). The results were similar for both methods. Ward and Heckbert's method gives better results when there are surfaces seen at very sharp grazing angles from the sampling point $\mathbf{p}$. Otherwise our method gives slightly smoother results.

Even though the quality of Ward and Heckbert's method is subtly superior to ours, our method provide many advantages. It works with any sampling distribution and with any function used to weight the radiance samples. The contribution of radiance samples to the gradient is independent of each other and therefore more easily amenable for parallelization or hardware implementation. Our method also does not assume any stratification of the hemisphere. This allows one to use our gradient computation with different sampling strategies, *e.g.* quasi Monte Carlo sampling.

D. Radiance Interpolation

When a ray hits a glossy surface with a low-frequency BRDF, the radiance cache is queried for records available for interpolation. If the query succeeds, the incoming radiance stored with those records is interpolated as described in this section.

We use a weighted interpolation scheme similar to the one proposed in [8] for interpolating the coefficient vectors $\Lambda_i = \{\lambda_{l,i}^m\}$ at any required surface point $\mathbf{p}$. The difference is that we replace the use of the rotational gradient by applying a rotation to the coefficient vector Λ_i of each cache record i used for interpolation at $\mathbf{p}$.

This aligns the coordinate frame at the position $\mathbf{p}_i$ of the cache record and the frame at $\mathbf{p}$ (see Figure 6). The weight $w_i(\mathbf{p})$ of record i with respect to $\mathbf{p}$ is

$$w_i(\mathbf{p}) = \left(\frac{\|\mathbf{p} - \mathbf{p}_i\|}{R_i} + \sqrt{1 - \mathbf{n} \cdot \mathbf{n}_i} \right)^{-1}, \tag{14}$$

where $\mathbf{n}$ is the surface normal at $\mathbf{p}$, $\mathbf{n}_i$ is the surface normal at $\mathbf{p}_i$, and R_i is the harmonic mean distance to objects visible from $\mathbf{p}_i$. The coefficient vector of the interpolated radiance is computed as a weighted average:

$$\Lambda(\mathbf{p}) = \frac{\sum_{\mathrm{S}} \mathbf{R}_i \left(\Lambda_i + d_x \frac{\partial \Lambda_i}{\partial x} + d_y \frac{\partial \Lambda_i}{\partial y} \right) w_i(\mathbf{p})}{\sum_{\mathrm{S}} w_i(\mathbf{p})}, \tag{15}$$

where the set S of radiance records used for interpolation at $\mathbf{p}$ is defined as $S = \{i | w_i(\mathbf{p}) > 1/a\}$ and a is a user defined desired accuracy. The definition of the set S effectively represents the criterion used to decide which radiance cache records can be used for interpolation. Derivatives $\partial \Lambda_i / \partial x$ and $\partial \Lambda_i / \partial y$ are the translational gradient components and (d_x, d_y) are the displacements of $\mathbf{p} - \mathbf{p}_i$ along the X and Y axes of the record i's local coordinate frame. $\mathbf{R}_i$ is the HSH rotation matrix [9] that aligns the coordinate frame at $\mathbf{p}_i$ with the frame at $\mathbf{p}$. It transforms the whole coefficient vector Λ_i after the translational gradient has been applied.

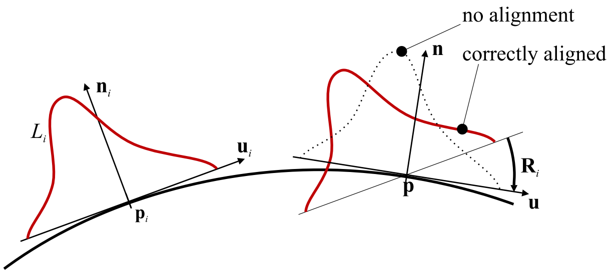


Fig. 6. Rotation $\mathbf{R}_i$ has to align the coordinate frame at $\mathbf{p}_i$ with that at $\mathbf{p}$ before interpolation is possible.

E. Outgoing Radiance Computation

The result of the interpolation procedure or of the hemisphere sampling is a vector of coefficients representing the incoming radiance at a point. The incoming radiance has to be integrated against the BRDF to obtain the outgoing radiance. In general the outgoing radiance at a point $\mathbf{p}$ is computed by evaluating the illumination integral (1). Since both the incoming radiance L_i and the BRDF f in the integral are represented as coefficient vectors with respect to an orthonormal basis, the integral evaluation reduces to the dot product [50]:

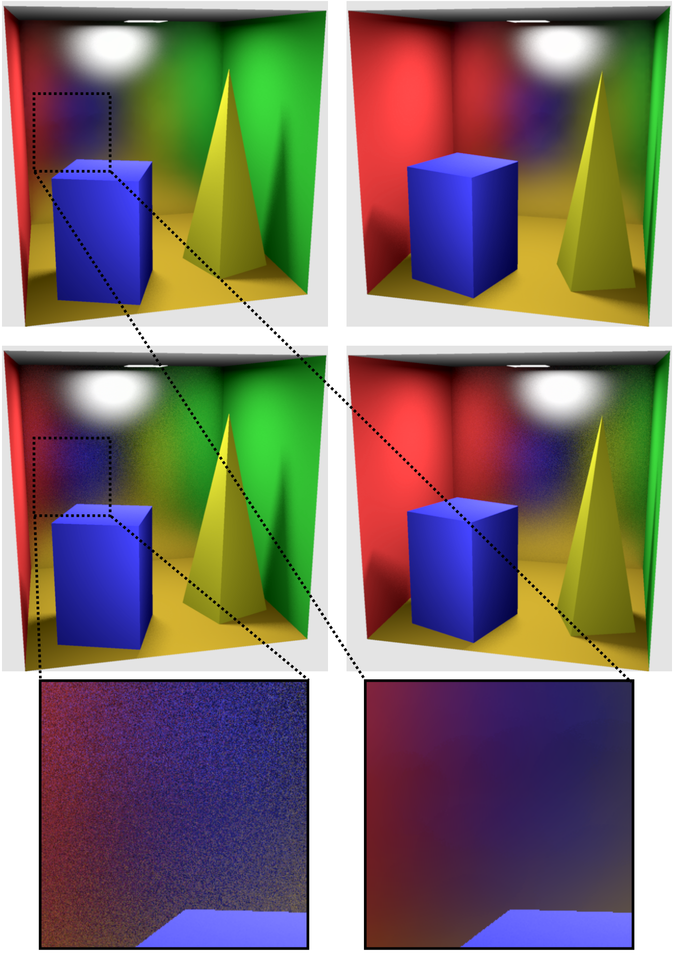


Fig. 7. Two views of a Cornell box with glossy back wall rendered using radiance caching (top) and MC importance sampling (bottom). Time spent on radiance caching was 33.3 seconds (600 records) for the left image and 13.8 seconds (164 additional records) for the right image. The indirect glossy term for each of the two bottom images was computed in 35 seconds using MC importance sampling with 12 reflected rays per pixel. Image resolution was 1280 × 1280.

$$L(\theta_o, \phi_o) = \sum_{l=0}^{n-1} \sum_{m=-l}^{l} \lambda_l^m f_l^m(\theta_o, \phi_o).$$

λ_l^m is an interpolated incoming radiance coefficient and $f_l^m(\theta_o, \phi_o)$ is a BRDF coefficient at $\mathbf{p}$ for the outgoing direction (θ_o, ϕ_o). This computation saves many BRDF evaluations.

V. Results

Table I gives the breakdown of rendering times for the the three scenes we used to test radiance caching (Cornell box, Office Space, Flamingo). The timings were measured on a 2.2GHz Pentium 4 with 1GB RAM running Windows XP; C++ code was compiled with Microsoft

	Cornell box 1280 × 1280		Office Space 1920 × 1299	Flamingo 1280 × 1280		
	Frame I	Frame II		Frame I	Frame II	Frame III
RC filling	28.7	9.2	106.7	63.5	53.2	63.9
RC interpolation	4.6	4.6	4.6	4.42	11.4	8.7
Total	82.9	68.7	155.0	108.0	101.0	104.0

TABLE I.

Timing breakdown for the test scenes. 'RC filling' is the time spent on computing and adding radiance cache records. 'RC interpolation' is the time spent on looking up the existing radiance cache records and interpolating the radiance. 'Total' is the total rendering time. The difference between the total time and the time spent by radiance caching consists of primary ray casting, direct lighting, specular reflections (in Office Space) and irradiance caching (in Flamingo). All times given in seconds.

Visual C++ 6.0. The resulting renderings are shown on figures 7, 8 and 9, and also in the accompanying video. The maximum HSH order for radiance caching was set to $n = 10$, which corresponds to approx. 3.6 kB sized radiance cache records.

We compare the solutions obtained by radiance caching (RC) with Monte Carlo (MC) importance sampling according to BRDF. The two rendering methods exhibit errors of very different characteristics: high-frequency noise for importance sampling ('specks' in images) and low-frequency error in radiance caching (uneven illumination gradients). It is therefore difficult to compare rendering times needed to attain the same visual quality. We have rather opted for using the image quality obtained in the same time as the comparison criterion between the two methods.

a) Cornell Box: Figure 7 shows renderings of a Cornell box with glossy back wall (Phong BRDF, exponent 22), taken from two viewpoints at resolution 1280 × 1280. Except for the back wall, all objects are Lambertian. Only direct lighting and first bounce indirect glossy lighting for the back wall were computed.

Images in the top row were computed using radiance caching with $N = 6000$ rays cast to sample each hemisphere. The indirect glossy term took 33.3 seconds to compute for the left image; total rendering time was 82.9 sec (direct illumination uses 8 samples per pixel to sample the area source). The number of RC records was 600. The time spent on the indirect glossy term computation in the right image was only 13.8 sec, since the records from the left rendering were retained and only 164 additional record were required.

The indirect glossy term for each of the two bottom images was computed in 35 seconds using MC importance sampling with 12 reflected rays per pixel on a glossy surface. Those rendering methods exhibit high noise lever, whose perception is even amplified in the temporal domain as shown in the video.

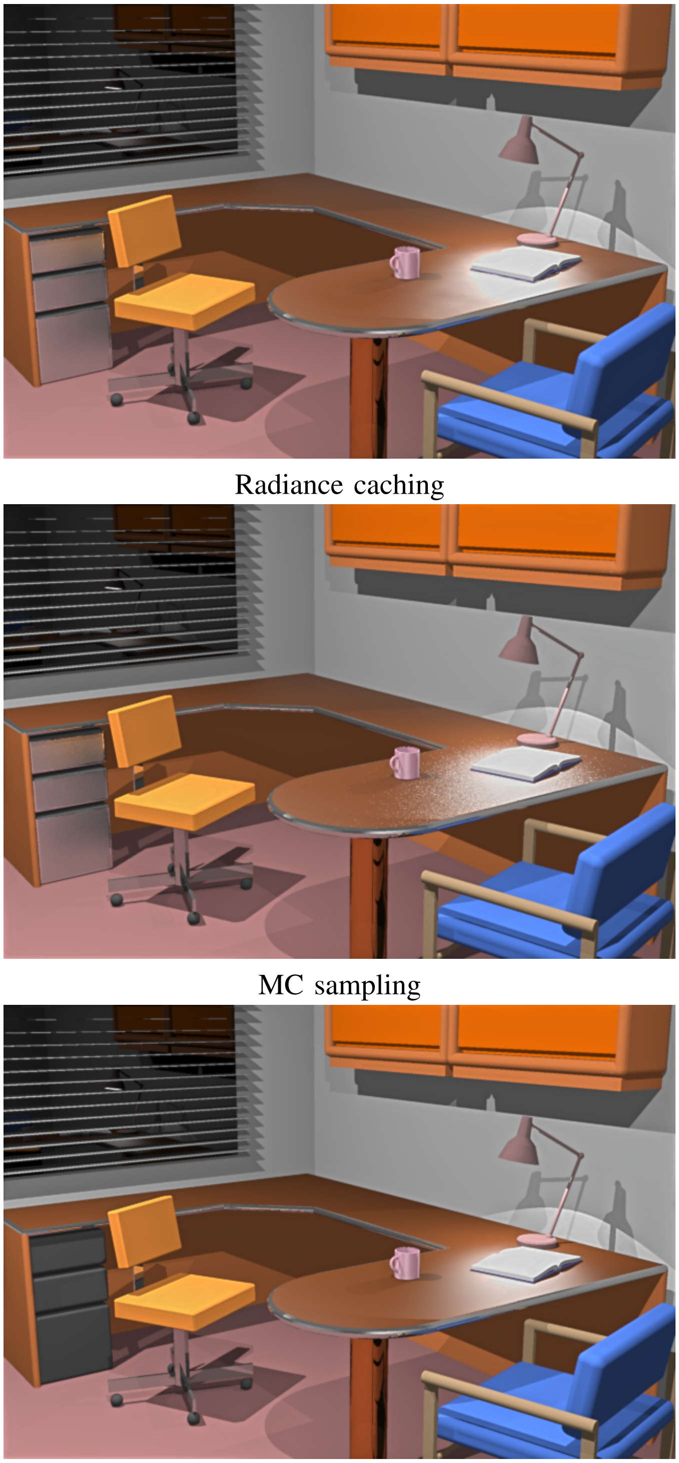
The average time spent on radiance caching for a 180 frames long animation with the camera moving between the position in the left and right images was just 4.9 sec per frame. Most of this time is spent on interpolation since only a few records are needed for additional frames.

b) Office Space: Figure 8 shows three renderings of the modified Office Space scene [51] with glossy desk top and drawers (Ward BRDF with $\alpha = 0.15$ for the drawers and $\alpha = 0.25$ for the desk top). Resolution of these images was 1920 × 1299; the number of rays per hemisphere was set to $N = 6000$. Only direct lighting and first bounce indirect lighting for glossy and specular surfaces were computed.

The top image shows the result of using radiance caching to compute glossy reflection. The number of RC records was 1282. Computation of the indirect glossy term took 111.3 sec., total rendering time was 155 sec.

In the middle image, the glossy indirect lighting was computed using MC importance sampling as described in [46], with 35 reflected rays used on each visible glossy pixel. The rendering time is the same as for radiance caching.

For comparison, the bottom image shows the same scene rendered without indirect glossy reflections whatsoever. The reflection on the drawers is obviously missing. The effect of indirect illumination on the desk top is more subtle. Apart from the slight reflection of the grey wall it demonstrates itself mainly by enlarging the highlight due to the light coming from the lamp. A lot of indirect lighting is due to the reflection of the light emitted by the lamp from the lamp shade. This light creates a high peak in the directional distribution of the indirect light coming to the desk top. Both MC importance sampling and radiance caching have problems to handle this situation correctly. Radiance caching shows a slight variation of intensity near the border of the highlight and MC importance sampling and introduces the white specks that spoil the rendering quality.



Radiance caching

MC sampling

No glossy reflections

Fig. 8. Modified Office Space scene with glossy desk top and drawers. The indirect glossy reflections in the top image were computed using radiance caching. 1282 records were used, the rendering time was 155 seconds (rendering resolution was 1920 × 1299 pixels). The middle image shows the same scene with indirect glossy reflections computed by MC importance sampling rendered in the same time. The bottom image shows the same scene rendered without indirect glossy reflections.

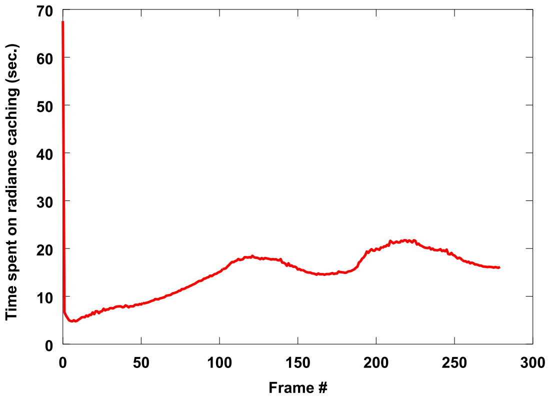


Fig. 10. Time spent on radiance caching for the Flamingo animation (see the accompanying video). Cached records are shared between the frames.

c) Glossy Flamingo: Figure 9 shows 3 frames from the Flamingo animation. The bird was assigned the Phong BRDF (exponent 7) and all other surfaces are purely diffuse. The rendering resolution was 1280 × 1280 pixels. Full global illumination up to 4 bounces was computed. Irradiance caching was used to compute indirect lighting on diffuse surfaces.

First bounce indirect lighting on glossy surfaces for the images in the top row was computed using radiance caching, path tracing was used for deeper recursion levels. Table I gives the rendering times for the three images when rendered independently, without retaining RC records from previous renderings. Figure 10 shows the time spent on radiance caching in a 280 frames long animation with camera moving between the shown images (see the accompanying video).

The images in the bottom row use MC importance sampling instead of radiance caching. The number of reflected rays per pixel on a glossy surface is 12, 4, 6 respectively (from left to right) so that the rendering time is the same as for radiance caching.

This scene is particularly challenging for radiance caching since the object is curved. Costly alignment is required before each interpolation and RC records cannot be used for as many pixels as on a flat surface. In the left image, the flamingo occupies only a small part of the screen and therefore one does not take advantage of radiance caching's independence on image resolution. The quality lead of radiance caching over MC importance sampling can be seen only by a very close inspection. However, in the other two images the noise introduced by MC sampling is more obvious. Notice also, that for the animation rendering the average time for radiance caching is 15 seconds per frame. Rendering the same animation with MC importance sampling with

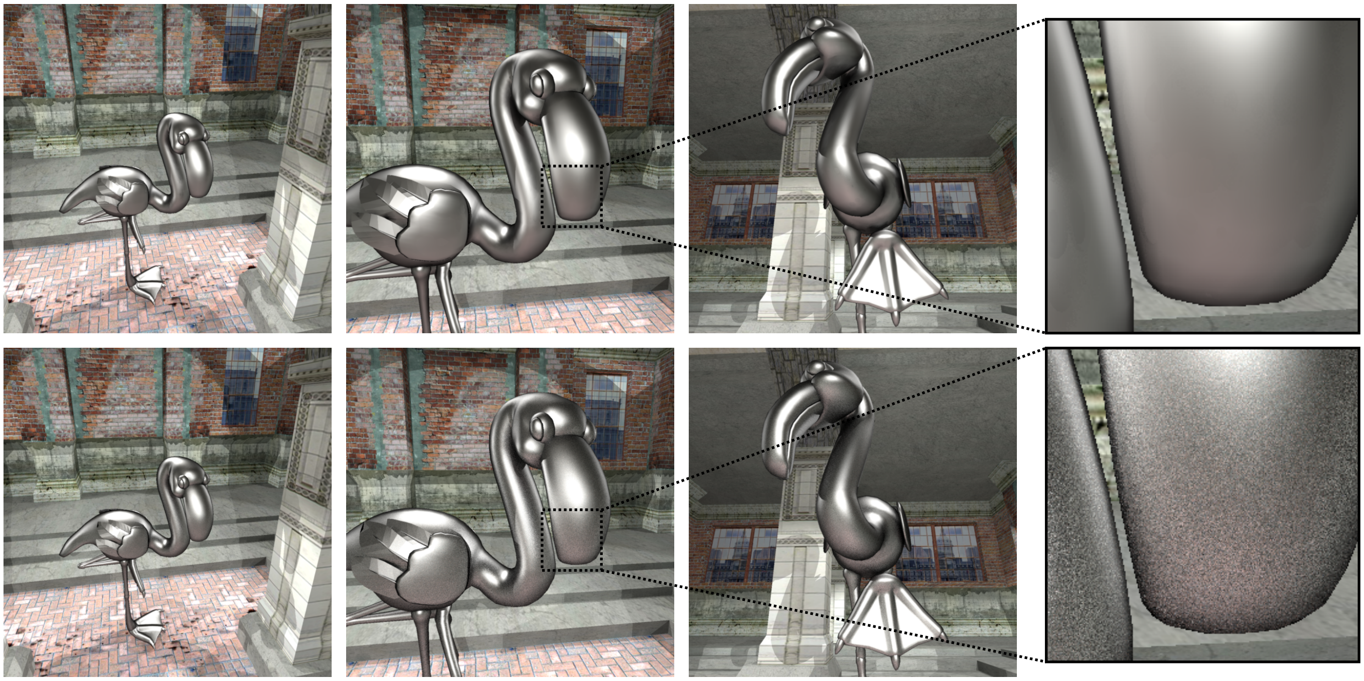


Fig. 9. Frames from the Flamingo animation rendered using radiance caching (top row) and MC importance sampling (bottom row).

2 reflected rays on a glossy pixel leads to 27 seconds per frame spent on indirect glossy lighting computation and the quality gain of radiance caching is considerable (see video).

VI. Discussion

d) The Rotation Loss: There is a loss of information when radiances are interpolated on a curved surface (Figure 11). It is due to the alignment (rotation $\mathbf{R}_i$) of the radiance records with the coordinate frame at the interpolation point $\mathbf{p}$.

A part of the radiance incident at $\mathbf{p}_i$ should disappear under the surface (marked 'a' in Fig. 11) and should not contribute to the interpolated radiance at $\mathbf{p}$. A part of the radiance actually incident at $\mathbf{p}$ is not represented by the radiance record at $\mathbf{p}_i$ (marked 'b' in Fig. 11) and is therefore missing in the interpolated radiance.

This problem is not due to using a hemispherical basis for representing the incoming radiance, but due to the fact the incident radiance at a surface point *is* a hemispherical function. Using spherical harmonics instead of HSH would not solve this problem. In practice the error introduced by this problem is very small because the difference between the normal at $\mathbf{p}$ and the normal at any record used for interpolation at $\mathbf{p}$ is small. Note also that this problem is present in the Ward's irradiance caching as well.

e) Global vs. Local Coordinates: Incoming radiance at a point $\mathbf{p}$ can be represented either in the local frame at $\mathbf{p}$ (*i.e.* aligned with the surface normal at $\mathbf{p}$) or in the global frame. This influences the way the interpolation at $\mathbf{p}$ is performed:

- *Incoming radiance in the global frame*

```
for (each available record $i$ at $\mathbf{p}$) do
    Update the interpolation sum.
end for
Align the interpolation result with the local frame at $\mathbf{p}$.
Compute the dot product.
```

- *Incoming radiance in the local frame*

```
for (each available record $i$ at $\mathbf{p}$) do
    Align the local frame at $\mathbf{p}_i$ with the local frame at $\mathbf{p}$.
    Update the interpolation sum.
end for
Compute the dot product.
```

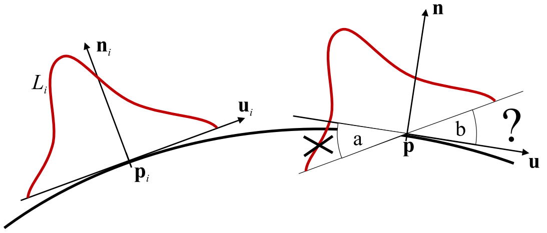


Fig. 11. Loss of information when radiances are interpolated on a curved surface. A part of the radiance actually incident at $\mathbf{p}_i$ should disappear under the surface (marked 'a'). A part of the radiance incident at $\mathbf{p}$ is not represented by the radiance record at $\mathbf{p}_i$ at all (marked 'b').

The final dot product between the BRDF coefficient vector and the interpolated coefficient vector of the incoming radiance is always carried out in the local frame at $\mathbf{p}$.

On curved surfaces, fewer alignment operations (rotations) are needed if the incoming radiance is represented in the global frame than if it is represented in the local frame. On the other hand, if the incoming radiance is represented in the local frame, no alignment (even with the BRDF) is needed on flat surfaces. The best performance (*i.e.* the lowest number of rotations) is obtained if the incoming radiance is represented in the local frame on flat surfaces and in the global frame on curved surfaces. Note that full spherical function representation (*e.g.* spherical harmonics) is needed to represent the incoming radiance in the global frame.

f) Suitability of Hemispherical Harmonics: We use (hemi)spherical harmonics since they are simple, computationally efficient (manipulations of vectors of floats) and avoid aliasing. The essential disadvantage is the lack of directional localization. When we create a new radiance cache record, the full hemisphere must be sampled, whatever the incoming ray direction is. The more directional the BRDF is, the more this approach becomes wasteful. With a basis that localizes in directions, only the required part of the hemisphere need to be sampled. For this purpose one can use piecewise constant representation [6], [26], [27], but it would presumably introduce too much aliasing. Spherical wavelets [30] are probably a good choice. This is left for further investigation. Even though we have implemented radiance caching using spherical and hemispherical harmonics only, the approach should be valid with any other set of basis function constructed in a similar way.

VII. Conclusion and Future Work

We have presented radiance caching, a method for accelerating computation of the indirect illumination on surfaces with low-frequency glossy BRDFs. Radiance caching is based on sparse sampling, caching, and interpolating incoming radiance on those surfaces. Radiance is represented by spherical or hemispherical harmonics in our approach. The interpolation quality is enhanced by the use of translational gradients, for whose computation we have presented two novel methods. We have also presented an automatic criterion to decide for which BRDFs radiance caching is suitable. We have shown on several examples that this approach is more efficient than pure Monte Carlo sampling at every surface point and delivers images of superior quality.

In the future we would like to use adaptive hemisphere sampling to compute the incoming radiance coefficients [29], [48], [52], use a different representation for the incoming radiance that localizes better in directions and devise interpolation criteria better suited for glossy surfaces. In the long term we also want to investigate the relationship between the frequency content of a BRDF and the suitability of interpolating radiance on surfaces with that BRDF.

References

[1] G. J. Ward, F. M. Rubinstein, and R. D. Clear, "A ray tracing solution for diffuse interreflection," in *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*. ACM Press, 1988, pp. 85–92.

[2] G. J. Ward, "The Radiance lighting simulation and rendering system," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM Press, 1994, pp. 459–472.

[3] X. Granier and G. Drettakis, "A final reconstruction approach for a unified global illumination algorithm," *ACM Trans. Graph.*, vol. 23, no. 2, pp. 163–189, 2004.

[4] E. Groeller, "Coherence in computer graphics," Ph.D. dissertation, Technische universität Wien, 1992.

[5] B. Guo, "Progressive radiance evaluation using directional coherence maps," in *Computer Graphics (SIGGRAPH '98 Proceedings)*, 1998, pp. 255–266.

[6] P. Slusallek, W. Heidrich, and H.-P. Seidel, "Radiance maps: An image-based approach to global illumination," SIGGRAPH '98, Technical Sketch, 1998.

[7] K. Bala, J. Dorsey, and S. Teller, "Radiance interpolants for accelerated bounded-error ray tracing," *ACM Trans. Graph.*, vol. 18, no. 3, pp. 213–256, 1999.

[8] G. J. Ward and P. S. Heckbert, "Irradiance gradients," in *Eurographics Workshop on Rendering*, 1992.

[9] P. Gautron, J. Křivánek, S. N. Pattanaik, and K. Bouatouch, "A novel hemispherical basis for accurate and efficient rendering," in *Eurographics Symposium on Rendering*, June 2004.

[10] P. Hanrahan, D. Salzman, and L. Aupperle, "A rapid hierarchical radiosity algorithm," *Computer Graphics (Proceedings of SIGGRAPH 1991)*, vol. 25, no. 4, pp. 197–206, 1991.

[11] P. S. Heckbert, "Simulating global illumination using adaptive meshing," Ph.D. dissertation, University of California, Berkeley, June 1991.

[12] D. Lischinski, F. Tampieri, and D. P. Greenberg, "Discontinuity meshing for accurate radiosity," *IEEE Computer Graphics and Applications*, vol. 12, no. 6, pp. 25–39, November 1992.

[13] B. Walter, G. Drettakis, and S. Parker, "Interactive rendering using render cache," in *Proceedings of the 13th Eurographics Workshop on Rendering*, 1999, pp. 19–30.

[14] B. Walter, G. Drettakis, D. P. Greenberg, and O. Deussen, "Enhancing and optimizing the render cache," in *Proceedings of the 10th Eurographics Workshop on Rendering*, June 2002.

[15] K. Bala, B. Walter, and D. Greenberg, "Combining edges and points for interactive high-quality rendering," *ACM Trans. Graph. (Proceedings of SIGGRAPH 2003)*, vol. 22, no. 3, 2003.

[16] M. Simmons and C. H. Séquin, "Tapestry: A dynamic mesh-based display representation for interactive rendering," in *Proceedings of the 11th Eurographics Workshop on Rendering*, June 2000, pp. 329–340.

[17] P. Tole, F. Pellacini, B. Walter, and D. P. Greenberg, "Interactive global illumination in dynamic scenes," *ACM Trans. Graph. (Proceedings of SIGGRAPH 2002)*, vol. 21, no. 3, pp. 537–546, July 2002.

[18] F. X. Sillion, J. R. Arvo, S. H. Westin, and D. P. Greenberg, "A global illumination solution for general reflectance distributions," in *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. ACM Press, 1991, pp. 187–196.

[19] L. Aupperle and P. Hanrahan, "A hierarchical illumination algorithm for surfaces with glossy reflection," *Computer Graphics*, vol. 27, no. Annual Conference Series, pp. 155–162, 1993.

[20] F. Sillion, G. Drettakis, and C. Soler, "A clustering algorithm for radiance calculation in general environments," in *Rendering Techniques*, June 1995.

[21] S. N. Pattanaik and K. Bouatouch, "Haar wavelet: A solution to global illumination with general surface properties," in *Photorealistic Rendering Techniques (Proceedings of Fifth Eurographics Workshop on Rendering)*. Springer-Verlag, Berlin, June 1995.

[22] P. Schröder and P. Hanrahan, "Wavelet methods for radiance computations," in *Photorealistic Rendering Techniques (Proceedings Fifth Eurographics Workshop on Rendering)*, G. Sakas and P. Shirley, Eds. Springer-Verlag, Berlin, 1995, pp. 310–326.

[23] R. R. Lewis and A. Fournier, "Light-driven global illumination with a wavelet representation," in *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, 1996, pp. 11–20.

[24] P. H. Christensen, D. Lischinski, E. J. Stollnitz, and D. H. Salesin, "Clustering for glossy global illumination," *ACM Trans. Graph.*, vol. 16, no. 1, pp. 3–33, 1997.

[25] M. Stamminger, A. Scheel, X. Granier, F. Perez-Cazorla, G. Drettakis, and F. Sillion, "Efficient glossy global illumination with interactive viewing," *Computer Graphics Forum*, vol. 19, no. 1, 2000.

[26] T. Kato, "Photon mapping in Kilauea," in *Siggraph 2002, Course Notes No. 43, A Practical Guide to Global Illumination using Photon Mapping*, 2002, pp. 122–191.

[27] T. Tawara, K. Myszkowski, and H.-P. Seidel, "Exploiting temporal coherence in final gathering for dynamic scenes," in *Proceedings of Computer Graphics International (CGI'04)*, June 2004.

[28] H. W. Jensen, *Realistic Image Synthesis Using Photon Mapping*. AK Peters, July 2001.

[29] J. Zaninetti, X. Serpaggi, and B. Péroche, "A vector approach for global illumination in ray tracing," *Computer Graphics Forum (EUROGRAPHICS '98 Proceedings)*, vol. 17, no. 3, 1998.

[30] P. Schröder and W. Sweldens, "Spherical wavelets: efficiently representing functions on the sphere," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM Press, 1995, pp. 161–172.

[31] B. Cabral, N. Max, and R. Springmeyer, "Bidirectional reflection functions from surface bump maps," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. ACM Press, 1987, pp. 273–281.

[32] S. H. Westin, J. R. Arvo, and K. E. Torrance, "Predicting reflectance functions from complex surfaces," in *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*. ACM Press, 1992, pp. 255–264.

[33] R. Ramamoorthi and P. Hanrahan, "Frequency space environment map rendering," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. ACM Press, 2002, pp. 517–526.

[34] R. Ramamoorthi, "A signal-processing framework for forward and inverse rendering," Ph.D. dissertation, Stanford University, 2002.

[35] J. Kautz, P.-P. Sloan, and J. Snyder, "Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics," in *Proceedings of the 13th Eurographics workshop on Rendering*. Eurographics Association, 2002, pp. 291–296.

[36] P.-P. Sloan, J. Kautz, and J. Snyder, "Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments," *ACM Trans. Graph. (Proceedings of SIGGRAPH 2002)*, vol. 21, no. 3, pp. 527–536, 2002.

[37] P.-P. Sloan, J. Hall, J. Hart, and J. Snyder, "Clustered principal components for precomputed radiance transfer," *ACM Trans. Graph. (Proceedings of SIGGRAPH 2003)*, vol. 22, no. 3, pp. 382–391, 2003.

[38] J. C. Wyant and K. Creath, "Basic wavefront aberration theory for optical metrology," in *Applied optics and Optical Engineering, Vol XI*. Academic Press, Inc., 1992, pp. 27–39.

[39] J. Koenderink, A. van Doorn, and M. Stavridi, "Bidirectional reflection distribution function expressed in terms of surface scattering modes," *ECCV*, vol. B, pp. 28–39, 1996.

[40] O. A. Makhotkin, "Analysis of radiative transfer between surfaces by hemispherical harmonics," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 56, no. 6, pp. 869–879, 1996.

[41] J. Arvo, "The irradiance jacobian for partially occluded polyhedral sources," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM Press, 1994, pp. 343–350.

[42] N. Holzschuch and F. Sillion, "Accurate computation of the radiosity gradient with constant and linear emitters," in *Sixth Eurographics Workshop on Rendering*, June 1995.

[43] T. Annen, J. Kautz, F. Durand, and H.-P. Seidel, "Spherical harmonic gradients for mid-range illumination," in *Proceedings of the Eurographics Symposium on Rendering 2004*. Eurographics Association, 2004, pp. 331–336.

[44] P. Shirley and C. Wang, "Direct lighting calculation by Monte Carlo integration," in *Photorealistic Rendering in Computer Graphics (Proceedings of the Second Eurographics Workshop on Rendering)*, 1994, pp. 54–59.

[45] W. Heidrich and H.-P. Seidel, "Realistic, hardware-accelerated shading and lighting," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 171–178.

[46] G. J. Ward, "Measuring and modeling anisotropic reflection," in *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*. ACM Press, 1992, pp. 265–272.

[47] X. Serpaggi and B. Péroche, "An adaptive method for indirect illumination using light vectors," in *Computer Graphics Forum (EUROGRAPHICS 2001 Proceedings)*, 2001, vol. 20(3), pp. 278–287.

[48] P. Shirley and K. Chiu, "Notes on adaptive quadrature on the hemisphere," Dept. of Computer Science, Indiana University, Tech. Rep. TR-411, July 1994.

[49] E. W. Weisstein, "Spherical coordinates," From *MathWorld* – A Wolfram Web Resource. http://mathworld.wolfram.com/SphericalCoordinates.html, 2004.

[50] G. Szegö, *Orthogonal polynomials*, 4th ed. American Mathematical Society, Providence, Rhode Island, 1975.

[51] G. Ward, "Office space," Part of the MGF Example Scenes available at http://radsite.lbl.gov/mgf/scenes.html.

[52] J. Rigau, M. Feixas, and M. Sbert, "Refinement criteria based on f-divergences," in *Proceedings of the 14th Eurographics workshop on Rendering*. Eurographics Association, 2003, pp. 260–269.

[53] E. W. Weisstein, "Legendre polynomial," From *MathWorld* – A Wolfram Web Resource. http://mathworld.wolfram.com/LegendrePolynomial.html, 2004.

APPENDIX I
HEMISPHERICAL HARMONICS

The hemispherical harmonics (HSH) basis functions are defined as

$$H_l^m(\theta,\phi) = \begin{cases} \sqrt{2}\widetilde{K}_l^m \cos(m\phi) P_l^m(2\cos\theta - 1) & \text{if } m > 0 \\ \sqrt{2}\widetilde{K}_l^m \sin(-m\phi) P_l^{-m}(2\cos\theta - 1) & \text{if } m < 0 \\ \widetilde{K}_l^0 P_l^0(\cos\theta) & \text{if } m = 0 \end{cases}$$

where P_l^m are the associated Legendre polynomials [53] and $\widetilde{K}_l^m$ is the following normalization value:

$$\widetilde{K}_l^m = \sqrt{\frac{(2l+1)(l-|m|)!}{2\pi(l+|m|)!}}.$$

The definition domain is $(\theta,\phi) \in [0,\pi/2] \times [0,2\pi)$, $l \in \{0,1,\ldots\}$, $m = \{-l,\ldots,0,\ldots,l\}$. Further information on hemispherical harmonics is given in [9].

APPENDIX II
SH AND HSH DERIVATIVES

Partial derivatives for spherical harmonics are:

$$\frac{\partial Y_l^m}{\partial\theta}(\theta,\phi) = \begin{cases} -\sqrt{2}K_l^m \cos(m\phi)\sin(\theta)\frac{\mathrm{d}P_l^m}{\mathrm{d}x}(\cos\theta) & \text{if } m > 0 \\ -\sqrt{2}K_l^m \sin(-m\phi)\sin(\theta)\frac{\mathrm{d}P_l^{-m}}{\mathrm{d}x}(\cos\theta) & \text{if } m < 0 \\ -K_l^0 \sin(\theta)\frac{\mathrm{d}P_l^0}{\mathrm{d}x}(\cos\theta) & \text{if } m = 0, \end{cases}$$

$$\frac{\partial Y_l^m}{\partial\phi}(\theta,\phi) = \begin{cases} 0 & \text{if } m = 0 \\ -mY_l^{-m}(\theta,\phi) & \text{otherwise.} \end{cases}$$

The derivative of the associated Legendre polynomials can be found from the recurrence formula:

$$\frac{\mathrm{d}P_l^m}{\mathrm{d}x}(x) = \begin{cases} \frac{1}{x^2-1}\Big(xlP_l^m(x) - (m+l)P_{l-1}^m(x)\Big) & \text{if } m < l \\ -(-1)^m x(2m-1)!!(1-x^2)^{\frac{m}{2}-1} & \text{if } m = l, \end{cases}$$

where $x!!$ is the double factorial (product of all odd integers less than or equal to x).

The partial derivatives for hemispherical harmonics are:

$$\frac{\partial H_l^m}{\partial\theta}(\theta,\phi) = \begin{cases} -2\sqrt{2}\widetilde{K}_l^m \cos(m\phi)\sin(\theta)\frac{\mathrm{d}P_l^m}{\mathrm{d}x}(2\cos\theta - 1) & \text{if } m > 0 \\ -2\sqrt{2}\widetilde{K}_l^m \sin(-m\phi)\sin(\theta)\frac{\mathrm{d}P_l^{-m}}{\mathrm{d}x}(2\cos\theta - 1) & \text{if } m < 0 \\ -2\widetilde{K}_l^0 \sin(\theta)\frac{\mathrm{d}P_l^0}{\mathrm{d}x}(2\cos\theta - 1) & \text{if } m = 0, \end{cases}$$

$$\frac{\partial H_l^m}{\partial\phi}(\theta,\phi) = \begin{cases} 0 & \text{if } m = 0 \\ -mH_l^{-m}(\theta,\phi) & \text{otherwise.} \end{cases}$$