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 hemispherical or spherical 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, hemispherical harmonics, 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]. Even for the radiosity method, high quality images are created by final gathering, often using Monte Carlo ray tracing [3].

Monte Carlo ray tracing is, however, expensive when it comes to computing indirect illumination on surfaces with low-frequency BRDFs (bi-directional reflectance distribution functions). Too many rays have to be traced to get a reasonably precise estimate of the outgoing radiance at a point. Fortunately, a high degree of coherence in the outgoing radiance field on those surfaces [1], [4]–[6] can be exploited by interpolation [1], [7] to obtain a significant performance gain.

Our goal is to accelerate Monte Carlo ray tracing-based global illumination computation in the presence of surfaces with low-frequency glossy BRDFs. We achieve it through sparse sampling, caching, and interpolating radiance on those surfaces. In particular, we extend Ward et al.'s irradiance caching [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. However, this applies to surfaces with arbitrary low-frequency BRDFs. Motivated by this observation, we extend Ward et al.'s work to cache and interpolate the *directional* incident radiance instead of the irradiance. This allows us to accelerate indirect lighting computation on surfaces with glossy BRDFs. We call the new method *radiance caching*.

The incoming radiance at a point is represented by a vector of coefficients with respect to spherical or hemispherical harmonics [9]. Due to the basis orthogonality, the illumination integral evaluation (Eq. 1) reduces to a dot product of the interpolated incoming radiance coefficients and the BRDF coefficients. Radiance interpolation is carried out by interpolating the coefficients. 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 Ward et al.'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 BRDF represented by (hemi)spherical harmonics, including measured BRDFs.

This paper extends the initial description of radiance caching given in [9] (included with the submission). The main contributions of this paper are the extension of irradiance caching to glossy surfaces, an automatic method for selecting BRDFs suitable for radiance caching, new methods for computing translational radiance gradient, and integration of radiance caching in a 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 ray tracer. Section IV details different aspects 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.

Manuscript received July XXX, 2004.

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

P. Gautron is with IRISA/INRIA Rennes and UCF. E-mail: pgautron@irisa.fr

S. Pattanaik is with UCF. E-mail: sumant@cs.ucf.edu

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

II. Related work

a) Interpolation: Interpolation can be used in global illumination whenever there is a certain level of smoothness in the quantity being computed. Radiosity method uses interpolation in the form of surface discretization *e.g.* [10]–[12]. In the context of Monte Carlo ray tracing, approaches have been proposed for *screen space interpolation* [5], [13]–[15]. The goal of these methods is to display an approximate solution quickly. 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 fast previewing [16], [17]. Sparse sampling and interpolation for high quality rendering was used in [1], [7]. 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 directional distributions has been used to extend the radiosity method to support glossy surfaces, *e.g.* [18]–[25]. It has also been used in Monte Carlo ray tracing on diffuse surfaces [6], [26], [27]. Slusallek et al. [6] and Kato [26] use reprojection of radiance samples. Tawara et al. [27] selectively update a radiance sample list in time to exploit temporal coherence. Storing light particles in the scene can also be thought of as caching a directional distribution [28], [29].

c) Spherical Function Representation: A 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 memory demanding. Unless higher order wavelets are used, even wavelet representation [21]–[23], [25], [30] does not remove the aliasing problems. Nevertheless, wavelets are a viable alternative to the use of spherical or hemispherical harmonics, especially for higher frequency BRDFs.

Spherical Harmonics [18], [20], [31]–[37] remove the aliasing problem and are efficient for representing low-frequency functions. However, representation of sharp functions require many coefficients and ringing might appear. Hemispherical harmonics [9] are better suited for representing functions on a hemisphere. Basis functions similar to spherical harmonics are Zernike polynomials [38], [39] and hemispherical harmonics of Makhotkin [40]. Unlike for spherical harmonics, the rotation procedure is not available for these basis functions. We choose hemispherical and spherical harmonics because they are the only basis for which an efficient rotation procedure is available. They also have good anti-aliasing properties, low storage cost and are easy to use.

d) Illumination Gradient Computation: Arvo [41] computes the irradiance Jacobian 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 using the information from 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 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 ray tracer [2]. They sample the irradiance sparsely over surfaces, cache the results and interpolate them. For each ray hitting a surface, the irradiance cache is queried. If irradiance records are available, the irradiance is interpolated. Otherwise a new irradiance record is computed by sampling the hemisphere and added to the cache. 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 us to apply the interpolation to glossy surfaces.

III. Algorithm Overview

Radiance caching is a part of a ray tracing approach to global illumination. At every ray-surface intersection, the outgoing radiance is evaluated with 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. Monte Carlo 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 coefficients would be needed. Moreover, high-frequency BRDFs are well localized and importance sampling provides good accuracy with a few secondary rays. The distinction between low- and high-frequency BRDFs is done automatically as described in Section IV-A. The steps of the rendering algorithm related to radiance caching are shown in Figure 1.

```
// 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
    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 hemisphere 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. (HSH stands for hemispherical harmonics.)

The i-th *radiance cache record* contains:

- position $\mathbf{p}_i$,
- local coordinate frame $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{n}_i)$,
- hemispherical harmonics 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 $\mathbf{p}_i$.

Λ denotes a coefficient vector and λ_l^m denotes a coefficient, that is $\Lambda = \{\lambda_l^m\}$. Each record stores the incident radiance function in a view-independent manner so that it 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 represent the 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 hemispherical harmonics to represent the BRDF over the hemisphere of incoming directions. The n-th order representation of a cosine weighted[1] BRDF $f_{(\theta_o,\phi_o)}$ for an outgoing direction (θ_o, ϕ_o) is

$$f_{(\theta_o,\phi_o)}(\theta_i,\phi_i) \approx \sum_{l=0}^{n-1} \sum_{m=-l}^{l} c_l^m(\theta_o,\phi_o) H_l^m(\theta_i,\phi_i) \ , \tag{2}$$

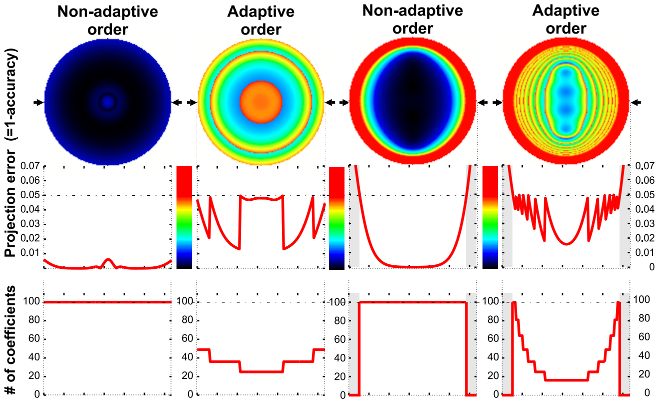
where

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

H_l^m are the hemispherical harmonics basis functions [9]. We sample the outgoing hemisphere for (θ_o, ϕ_o) using the parabolic parametrization [45].

Adaptive BRDF Representation: Hemispherical harmonics representation for all scene BRDFs is computed before the rendering starts. For each outgoing direction, the adaptive representation of $f_{(\theta_o,\phi_o)}$ uses the minimum order n sufficient to avoid exceeding the user specified maximum error, measured as described in [33] (see Figure 2). If no order $n < n_{\max}$ is sufficient for the specified error, the hemispherical harmonics 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 the harmonics and radiance caching is used for them. This constitutes an automatic criterion for discerning low- and high-frequency BRDFs in our rendering framework. $n_{\max}$ 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 Monte Carlo importance sampling. While $n_{\max} = 10$ was a good value for our scenes, it would not have to be so in other ones. Higher $n_{\max}$ allows using radiance caching for higher frequency BRDFs. The higher frequency BRDFs, the more rays will be needed to sample the hemisphere for a new record and the less reuse will be possible for that record.

[1] All BRDFs are multiplied by the cosine term $\cos\theta_i$ before computing the harmonics 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 order of the hemispherical harmonics representation adapts to the BRDF without ever exceeding the specified maximum representation error (here 5%). The color disks represent BRDF representation 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). In the case of the Ward BRDF, radiance caching is not used for some directions, since the representation error would be too high.

B. Incoming Radiance Computation

Whenever interpolation is not possible at a point $\mathbf{p}$, a new radiance record is computed and stored in the cache. We represent the incoming radiance L_i by a vector of hemispherical harmonics coefficients $\Lambda = \{\lambda_l^m\}$ as $L_i(\theta, \phi) \approx \sum_{l=0}^{n-1} \sum_{m=-l}^{l} \lambda_l^m H_l^m(\theta, \phi)$, where n is the representation order. The coefficients λ_l^m for an analytical L_i would be computed with the integral $\lambda_l^m = \int_0^{2\pi}\int_0^{\pi/2} L_i(\theta, \phi) H_l^m(\theta, \phi) \sin\theta \mathrm{d}\theta \mathrm{d}\phi$. Our knowledge of L_i is based only on sampling (ray casting). Hence, we compute λ_l^m by a Monte Carlo quadrature with uniform sampling:

$$\lambda_l^m = \frac{2\pi}{N} \sum_{k=1}^{N} L_i(\theta_k, \phi_k) H_l^m(\theta_k, \phi_k), \tag{4}$$

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. We use a fixed N but adaptive hemisphere sampling [47], [48] is desirable.

The order n for the incoming radiance representation is equal to the order of the BRDF representation at $\mathbf{p}$. This cuts off high frequencies from the incoming radiance. The approach is justified by a low-frequency BRDF acting as a low-pass filter on the incoming radiance [34]. If incoming radiance vectors with different number of coefficients are interpolated, the shorter vectors are padded with zeros.

C. Translational Gradient Computation

We want to compute the translational gradient $\nabla\lambda_l^m$ for each λ_l^m computed with Eq. (4). The gradient is used to improve the interpolation smoothness. We did not succeed in extending the gradient computation of Ward and Heckbert [8] to handle our case since their method tightly couples the cosine probability density and the cosine weighting used for irradiance computation.

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, analytical, is based on differentiating the terms of Eq. (4). 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 typical displacements along z are very small and using it does not visually improve the interpolation quality. We compute the 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 (4) 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

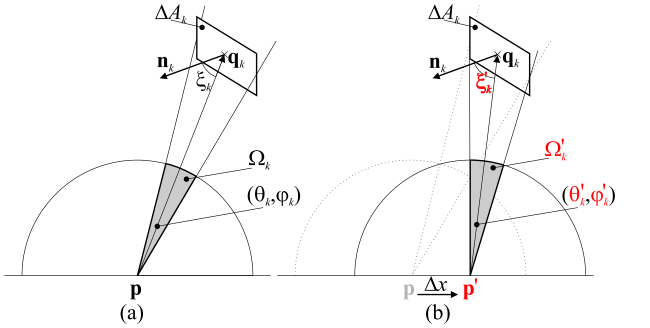
$$\Omega'_k = \Delta A_k \frac{\cos\xi'_k}{r'^2_k} = \frac{2\pi}{N}\frac{r_k^2}{r'^2_k}\frac{\cos\xi'_k}{\cos\xi_k}.$$

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 $\partial \lambda_l^m / \partial x = (\lambda_l^{m\prime} - \lambda_l^m)/\Delta x$.

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.



(a) (b)

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) Analytical Gradient Computation: We rewrite Equation (4) as

$$\lambda_l^m = \sum_{k=1}^{N} \Omega_k L_i(\theta_k, \phi_k) H_l^m(\theta_k, \phi_k), \tag{5}$$

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 partial derivative $\partial \lambda_l^m / \partial x$ is computed by differentiating the terms of the sum in Eq. (5)

$$\begin{aligned} \frac{\partial \lambda_l^m}{\partial x} &= \sum_{k=1}^{N} \frac{\partial}{\partial x} \Big(\Omega_k L_i(\theta_k, \phi_k) H_l^m(\theta_k, \phi_k) \Big) = \\ &= \sum_{k=1}^{N} L_i(\theta_k, \phi_k) \left(\frac{\partial \Omega_k}{\partial x} H_l^m(\theta_k, \phi_k) + \Omega_k \frac{\partial H_l^m(\theta_k, \phi_k)}{\partial x} \right) \end{aligned} \tag{6}$$

The derivative of the basis function is

$$\frac{\partial H_l^m(\theta_k, \phi_k)}{\partial x} = \frac{\partial \theta_k}{\partial x} \frac{\partial H_l^m(\theta_k, \phi_k)}{\partial \theta_k} + \frac{\partial \phi_k}{\partial x} \frac{\partial H_l^m(\theta_k, \phi_k)}{\partial \phi_k}, \tag{7}$$

with [49]

$$\begin{aligned} \partial \theta_k / \partial x &= -\cos \theta_k \cos \phi_k / r_k, \\ \partial \phi_k / \partial x &= \sin \phi_k / (r_k \sin \theta_k). \end{aligned} \tag{8}$$

Those derivatives with respect to y would be

$$\begin{aligned} \partial \theta_k / \partial y &= -\cos \theta_k \sin \phi_k / r_k, \\ \partial \phi_k / \partial y &= -\cos \phi_k / (r_k \sin \theta_k). \end{aligned} \tag{9}$$

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

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). 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

$$\begin{aligned} \frac{\partial}{\partial q_x} \frac{\cos \xi_k}{r_k^2} &= -\frac{\partial}{\partial q_x} \frac{n_x q_x + n_y q_y + n_z q_z}{(q_x^2 + q_y^2 + q_z^2)^{3/2}} \\ &= -\frac{r_k n_x + 3 q_x \cos \xi_k}{r_k^4}. \end{aligned} \tag{10}$$

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

$$\frac{\partial \Omega_k}{\partial x} = \frac{2\pi}{N} \frac{r_k n_x + 3 q_x \cos \xi_k}{r_k^2 \cos \xi_k}. \tag{11}$$

Plugging Equations (11) and (7) into (6) we get the complete formula for $\partial \lambda_l^m / \partial x$. The formulas for $\partial \lambda_l^m / \partial y$ are similar; only Eq. (8) must be replaced by Eq. (9).

A similar gradient calculation was also proposed in [29]. This method disregards the change of Ω_k and hence does not provide good results. The analytical method was also independently developed by Annen et al. [43]. A code fragment evaluating one term of the sum in Eq. (6) is given in the accompanying material [50].

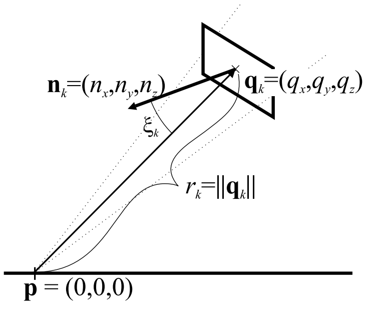


Fig. 4. Quantities in the computation of $\frac{\partial}{\partial q_x} \frac{\cos \xi_k}{r_k^2}$.

3) Discussion: For the derivation of both numerical and analytical 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 numerical and analytical 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 analytical method is numerically more stable near edges and corners and also slightly faster to evaluate.

4) Irradiance Gradient Computation: Note that both methods we propose can still be 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 analytical 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 [8] 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 some surfaces are 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 provides 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 our method is more easily amenable for parallelization or hardware implementation. We do 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.

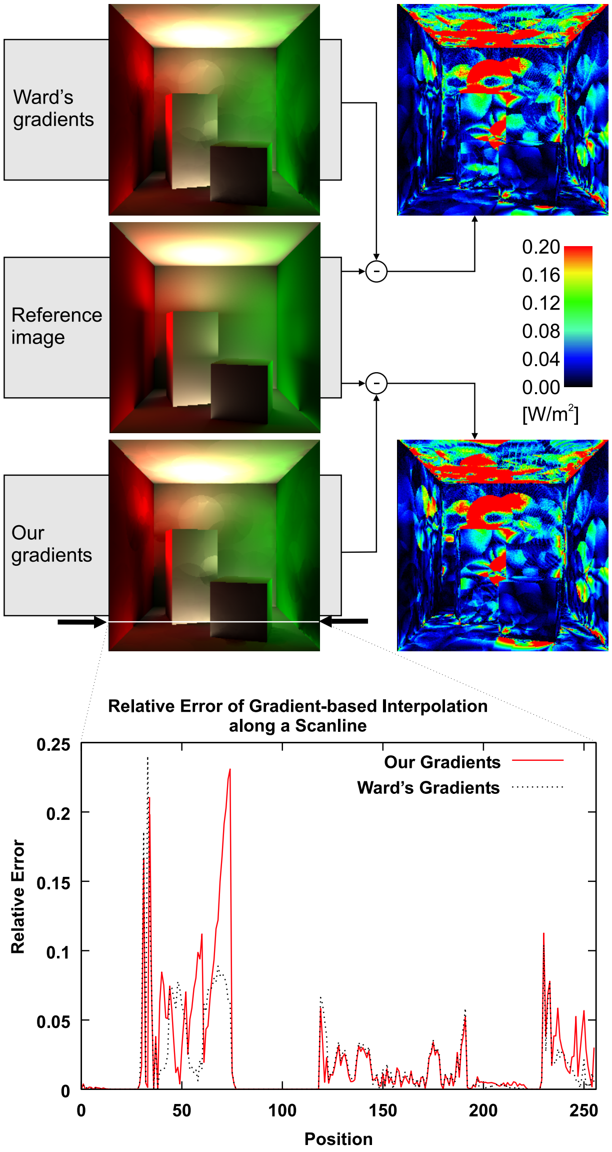


Fig. 5. Comparison of irradiance gradient computation. The scene is a diffuse Cornell box; only first bounce indirect illumination is computed. The color-coded images show the difference between the gradient-based interpolation and the reference solution (10,000 samples per hemisphere at each pixel). 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.

D. Radiance Interpolation

If a query to the radiance cache succeeds, the incoming radiance 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 Λ_i at any required surface point $\mathbf{p}$. The difference is that we replace the use of the rotational gradient by truly rotating the incident radiance function. 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 $w_i(\mathbf{p}) = \left(\frac{\|\mathbf{p}-\mathbf{p}_i\|}{R_i} + \sqrt{1-\mathbf{n}\cdot\mathbf{n}_i}\right)^{-1}$, 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:

$$\Lambda(\mathbf{p}) = \frac{\sum_S \mathbf{R}_i \left(\Lambda_i + d_x \frac{\partial \Lambda_i}{\partial x} + d_y \frac{\partial \Lambda_i}{\partial y}\right) w_i(\mathbf{p})}{\sum_S w_i(\mathbf{p})}, \quad (12)$$

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. If S is non-empty, interpolation (or extrapolation) is possible. d_x and 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. Displacement along z is not taken into account for the gradient-enhanced interpolation, since it is typically very small. $\mathbf{R}_i$ is the hemispherical harmonics rotation matrix [9] that aligns the coordinate frame at $\mathbf{p}_i$ with the frame at $\mathbf{p}$.

The interpolation scheme is borrowed from Ward et al. [1]. They derived it from the 'split sphere' model that estimates an upper bound on the magnitude of the change of irradiance. Although this model does not apply for radiance caching, the results are satisfying. We observe that a lower a has to be used for higher-frequency BRDFs and interpolation errors are more apparent when surfaces are viewed from grazing angles. In future work we want to devise an interpolation scheme suited for radiance caching based on these observations.

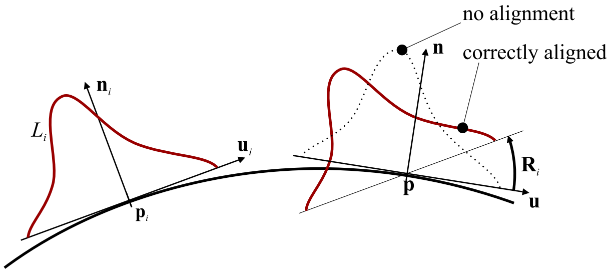


Fig. 6. Rotation $\mathbf{R}_i$ aligns the coordinate frame at $\mathbf{p}_i$ with that at $\mathbf{p}$ to make interpolation possible.

E. Outgoing Radiance Computation

The incoming radiance obtained by interpolation or hemisphere sampling is integrated against the BRDF to compute the outgoing radiance. With an orthonormal basis, the integral evaluation reduces to the dot product [51]:

$$L(\theta_o, \phi_o) = \sum_{l=0}^{n-1} \sum_{m=-l}^{l} \lambda_l^m c_l^m(\theta_o, \phi_o). \quad (13)$$

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

V. Results

Figure 7 gives the breakdown of rendering times for the the three scenes we used to test radiance caching (Cornell Box, Walt Disney Hall, Flamingo). The timings were measured on a 2.2GHz Pentium 4 with 1 GB RAM running Windows XP. The resulting renderings are shown in figures 8, 9 and 10, and in the accompanying video [50]. The maximum hemispherical harmonics order for radiance caching was set to $n = 10$, which corresponds to approx. 3.6 kB sized radiance cache records.

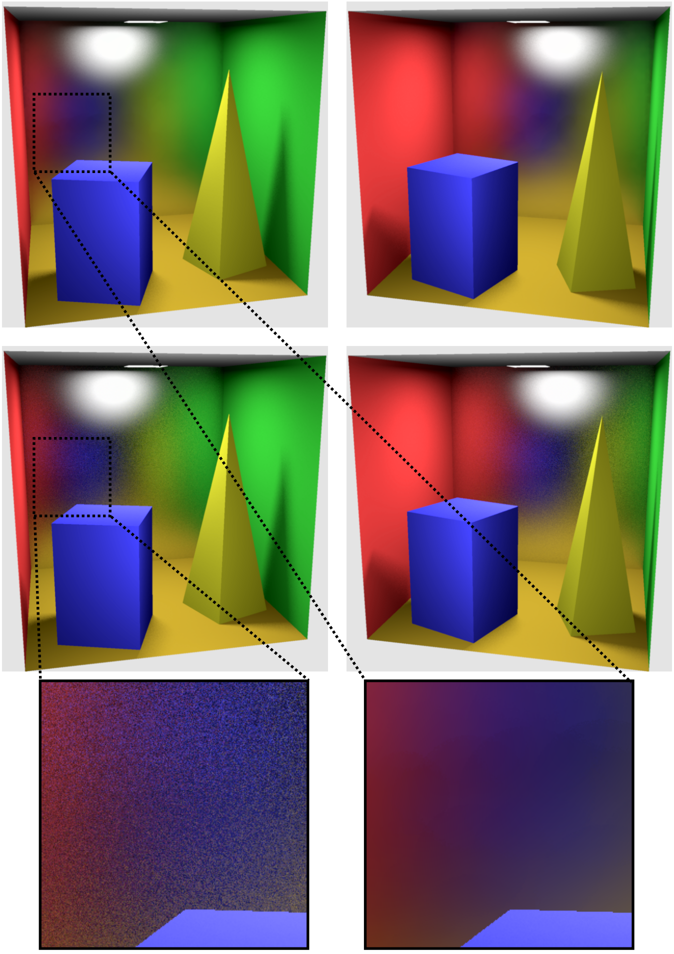


Fig. 8. Two views of a Cornell Box with glossy back wall rendered using radiance caching (top) and Monte Carlo importance sampling (bottom).

	Cornell Box 1280 × 1280		Disney Hall 1440 × 840		Flamingo 1280 × 1280		
	Frame I	Frame II	Frame I	Frame II	Frame I	Frame II	Frame III
RC filling	28.7	9.2	218.0	169.0	63.5	53.2	63.9
RC interpolation	4.6	4.6	17.3	17.0	4.42	11.4	8.7
Total	82.9	68.7	357.8	295.6	108.0	101.0	104.0

Fig. 7. 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 Disney Hall and Flamingo). All times given in seconds.

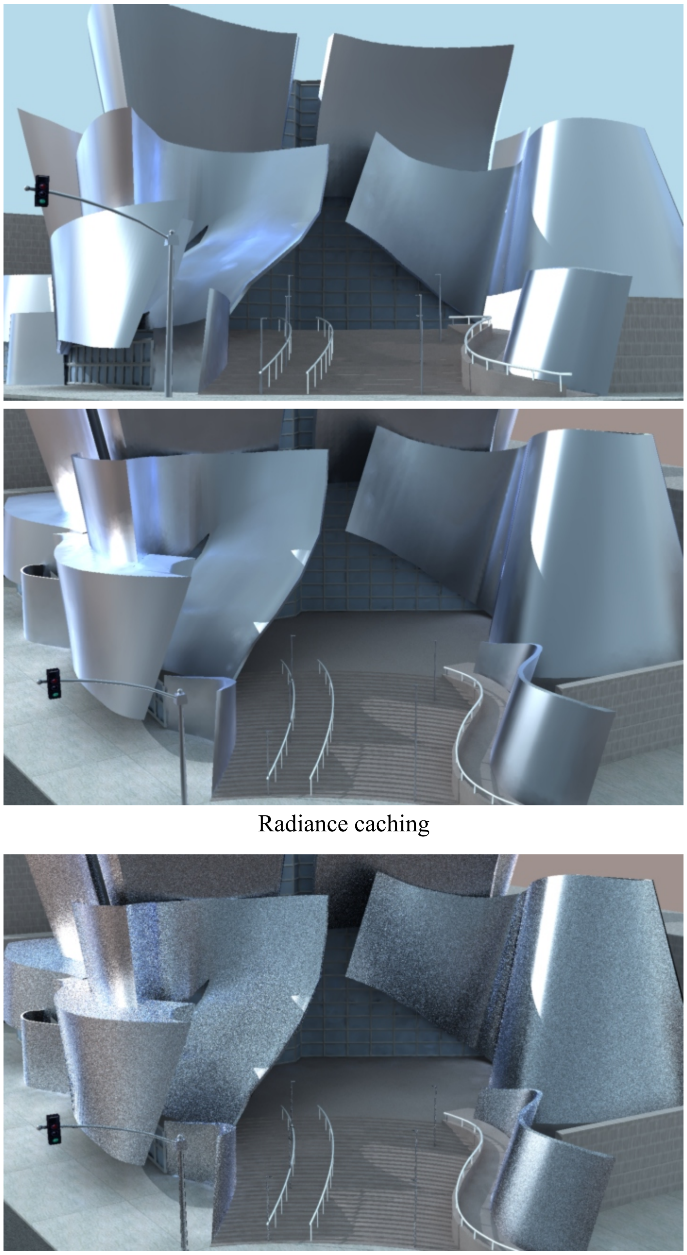
We compared the solutions obtained by radiance caching with those obtained by Monte Carlo importance sampling. Importance sampling uses the surface BRDF as the importance function. The two rendering methods exhibit artifacts with very different characteristics: high-frequency noise for importance sampling (‘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. Instead, we have chosen to fix the rendering time and compare the image quality delivered by the two methods.

a) Cornell Box: Figure 8 shows renderings of a Cornell Box with a 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 the caching accuracy set to $a = 0.15$ and the number of rays cast to sample each hemisphere set to $N = 6000$. 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 radiance cache 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 Monte Carlo importance sampling with 12 reflected rays per pixel on a glossy surface. Those rendering methods exhibit high noise level, 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. The average frame time with Monte Carlo importance sampling is 35 seconds since this


Radiance caching

Monte Carlo sampling

Fig. 9. Rendering of a simple model of Walt Disney Hall in Los Angeles. The top and the middle images, computed with radiance caching, show the building from two different viewpoints. The bottom image was computed with Monte Carlo importance sampling.

method does not reuse any information from the previous frames. Even though this time is 7 times longer than for radiance caching, the quality is much lower.

b) Walt Disney Hall: Figure 9 shows renderings of a simple model of Walt Disney Hall in Los Angeles. The curved walls of the real building are covered with brushed metal tiles, whose BRDF we approximate by a three-lobe Lafortune model [52]. The illumination is due to the sun (modeled as a directional light), the sky (modeled as a constant blue light) and the surrounding urban environment (modeled as a constant brownish light). Similarly to the real building (see [50]), walls reflect the sky or the surroundings depending on their normals and the viewpoint (compare the top and the middle image).

Full global illumination solution with one ray per pixel was computed at resolution 1440×840 and then scaled down. Indirect illumination and the illumination coming from the sky and from the surroundings was computed in the same way: with irradiance caching on diffuse surfaces, with radiance caching on glossy surfaces in the top and the middle image, and with Monte Carlo sampling on glossy surfaces in the bottom image.

To capture the high indirect illumination variations on the metal walls, the caching accuracy was set to $a = 0.1$ leading to as many as 38,000 radiance cache records. We used 800 rays for hemisphere sampling. The rendering time was 357.8 s and 295.6 s for the top and the middle image respectively. Monte Carlo importance sampling in the bottom image used 15 reflected rays on each visible glossy pixel (fist bounce) and one ray for other bounces. The rendering time was similar to that for radiance caching, but the quality of radiance caching results is higher.

The BRDF we used for the metallic walls [53] was relatively sharp; the Lafortune lobes had exponents of 16, 88 and 186. It is impossible to represent such a BRDF accurately using hemispherical harmonics of order 10 — the average representation error was over 20%. Nonetheless, radiance caching renderings show no serious distortion of the material appearance compared to Monte Carlo sampling.

c) Glossy Flamingo: Figure 10 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. The caching accuracy was set to $a = 0.15$ and the number of rays for hemisphere sampling was $N = 1000$. Table 7 gives the rendering times for the three images when rendered independently, without retaining radiance cache records from previous renderings. Figure 11 shows the time spent on radiance caching in a 280 frames long animation with camera moving between the shown images (see the accompanying video).

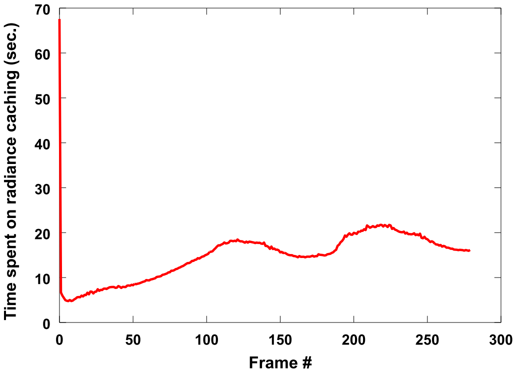


Fig. 11. Time spent on radiance caching for the Flamingo animation (see the accompanying video). Cached records are shared between the frames.

The images in the bottom row use Monte Carlo importance sampling instead of radiance caching. In order to keep the rendering time the same as for radiance caching, the number of reflected rays per pixel on a glossy surface was set to 12, 4, 6 respectively (from left to right).

This scene is particularly challenging for radiance caching since the glossy surface is curved. On such a surface, radiance cache records cannot be used for interpolation at as many pixels as on a flat surface. Moreover, costly alignment is required before each interpolation. 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 advantage of radiance caching over Monte Carlo importance sampling can be seen only by a very close inspection (see the images in the accompanying material [50]). However, in the other two images the noise introduced by Monte Carlo 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 using Monte Carlo importance sampling with only 2 reflected rays on a glossy pixel leads to 27 seconds per frame spent on indirect glossy lighting computation. Notice in the accompanying video, that the quality obtained by radiance caching is considerably better.

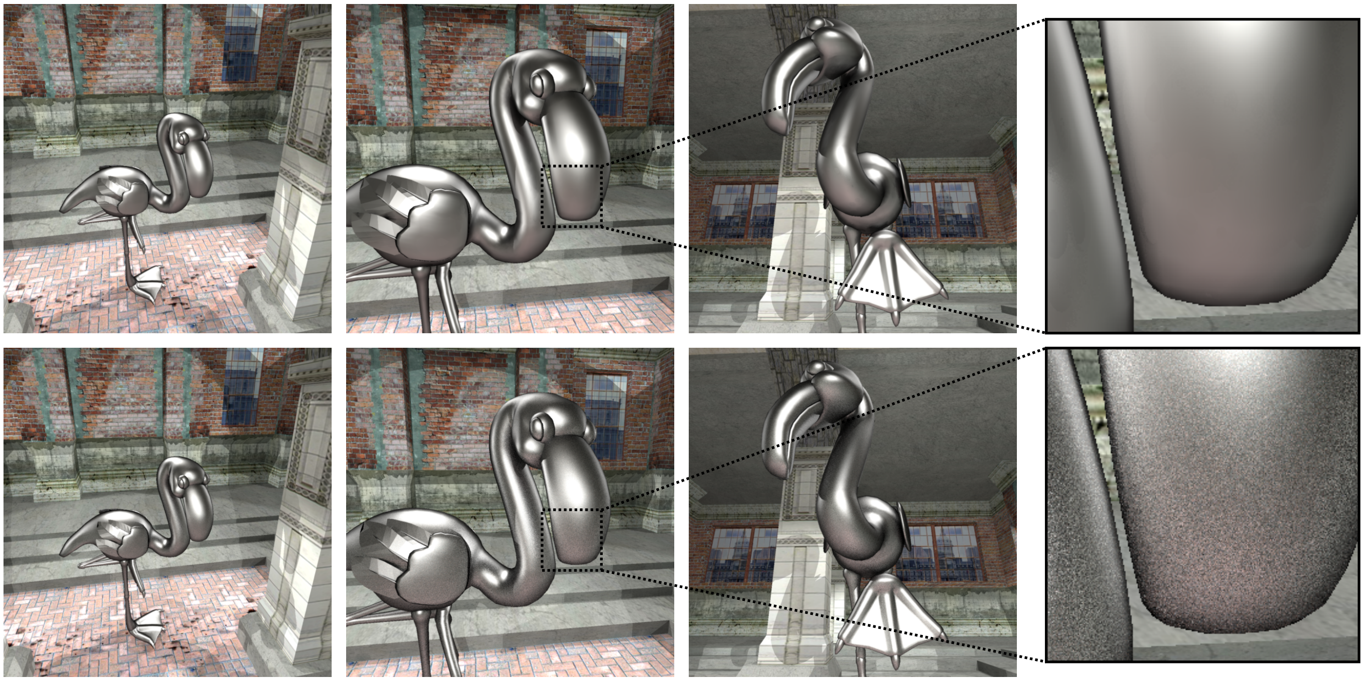


Fig. 10. Frames from the Flamingo animation rendered using radiance caching (top row) and Monte Carlo importance sampling (bottom row).

VI. Discussion

a) Rotation Loss: There is a loss of information when radiances are interpolated on a curved surface (Figure 12). A part of the radiance incident at $\mathbf{p}_i$ should disappear under the surface (marked 'a' in Fig. 12) 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. 12) and is 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, instead of hemispherical, harmonics 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 Ward et al.'s irradiance caching suffers from the same problem.

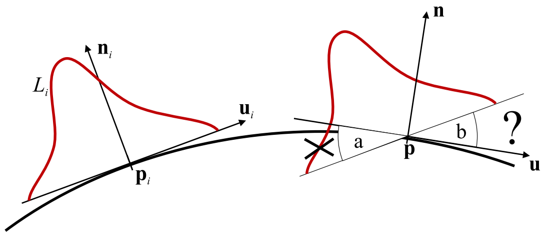


Fig. 12. Information loss when radiances are interpolated on a curved surface.

b) Global vs. Local Coordinates: Incoming radiance at a point can be represented either in the local frame at that point or in the global frame. This influences how the interpolation at $\mathbf{p}$ is performed:

- *Incoming radiance in the global frame*

```
for (each available record $i$) do
    Update the interpolation sums in (12).
end for
Align the result with the local frame at $\mathbf{p}$.
Compute the dot product.
```

- *Incoming radiance in the local frame*

```
for (each available record $i$) do
    Align the frame at $\mathbf{p}_i$ with the frame at $\mathbf{p}$.
    Update the interpolation sums in (12).
end for
Compute the dot product.
```

The final dot product (Eq. 13) is always carried out in the local frame at $\mathbf{p}$. On curved surfaces, fewer rotations are needed if the incoming radiance is represented in the global 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 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.* using spherical harmonics) is needed to represent the incoming radiance in the global frame.

c) Suitability of Hemispherical Harmonics: We use (hemi)spherical harmonics since they are simple, computationally efficient (manipulations of vectors of floats), avoid aliasing, and rotation is available for them. 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 would need to be sampled. For this purpose one can use piecewise constant representation [6], [26], [27], but it would presumably introduce aliasing. The use of spherical wavelets [30] is probably a good choice, and is left for further investigation.

d) General limitations: The BRDF frequencies used in the example scenes are near the limit of what our technique can currently handle. The restriction to low-frequency BRDFs is the main limitation of our technique, because we use hemispherical harmonics. We believe that higher frequency BRDFs can be handled within the presented framework by using a localized basis such as wavelets. Additionally, the radiance caching, as a gathering technique, cannot solve certain types of light transfer phenomena, such as caustics. Those have to be solved by a more adapted technique.

VII. Conclusion and Future Work

We have presented radiance caching, a method for accelerating the 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 hemispherical or spherical harmonics in our approach. The interpolation quality is enhanced by the use of translational gradients that can be computed with two novel methods we have presented in this paper. We have also proposed an automatic criterion to decide for which BRDFs radiance caching is suitable. We have shown on several examples that radiance caching is more efficient than pure Monte Carlo sampling at every surface point and delivers images of superior quality.

In the future work we would like to use adaptive hemisphere sampling to compute the incoming radiance coefficients [29], [48], [54]; 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 wish to investigate the relationship between the frequency content of a BRDF and the suitability of interpolating radiance on surfaces with that BRDF.

Acknowledgements

Thanks to Vlastimil Havran for letting us use his ray tracer, Marina Casiraghi and Eric Risser for the Walt Disney Hall model. This work was partially supported by ATI Research, I-4 Matching fund and Office of Naval Research.

References

[1] G. J. Ward, F. M. Rubinstein, and R. D. Clear, “A ray tracing solution for diffuse interreflection,” in *Proceedings of SIGGRAPH ’88*, 1988, pp. 85–92.

[2] G. J. Ward, “The Radiance lighting simulation and rendering system,” in *Proceedings of SIGGRAPH ’94*, 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 *Proceedings of SIGGRAPH ’98*, 1998.

[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*, 2004.

[10] P. Hanrahan, D. Salzman, and L. Aupperle, “A rapid hierarchical radiosity algorithm,” in *Proceedings of SIGGRAPH ’91*, 1991.

[11] P. S. Heckbert, “Simulating global illumination using adaptive meshing,” Ph.D. dissertation, University of California.

[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 SIGGRAPH ’91*, 1991, pp. 187–196.

[19] L. Aupperle and P. Hanrahan, “A hierarchical illumination algorithm for surfaces with glossy reflection,” in *Proceedings of SIGGRAPH ’93*, 1993, pp. 155–162.

[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 *Proceedings of Fifth Eurographics Workshop on Rendering*, 1995.

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

[23] R. R. Lewis and A. Fournier, "Light-driven global illumination with a wavelet representation," in *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*, 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*, 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," in *Eurographics*, 1998.

[30] P. Schröder and W. Sweldens, "Spherical wavelets: efficiently representing functions on the sphere," in *Proceedings of SIGGRAPH '95*, 1995, pp. 161–172.

[31] B. Cabral, N. Max, and R. Springmeyer, "Bidirectional reflection functions from surface bump maps," in *Proceedings of SIGGRAPH '87*, 1987, pp. 273–281.

[32] S. H. Westin, J. R. Arvo, and K. E. Torrance, "Predicting reflectance functions from complex surfaces," in *Proceedings of SIGGRAPH 92*, 1992, pp. 255–264.

[33] R. Ramamoorthi and P. Hanrahan, "Frequency space environment map rendering," in *Proceedings of SIGGRAPH*, 2002.

[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," in *Proceedings of SIGGRAPH*, 2002.

[37] P.-P. Sloan, J. Hall, J. Hart, and J. Snyder, "Clustered principal components for precomputed radiance transfer," in *Proceedings of SIGGRAPH 2003*, 2003, pp. 382–391.

[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 SIGGRAPH '94*, 1994.

[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*, 2004.

[44] P. Shirley and C. Wang, "Direct lighting calculation by Monte Carlo integration," in *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 SIGGRAPH '99*, 1999.

[46] G. J. Ward, "Measuring and modeling anisotropic reflection," in *Proceedings of SIGGRAPH '92*, 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).

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

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

[50] http://www.cs.ucf.edu/graphics/RCache/index.html, the Web page accompanying this paper.

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

[52] E. P. F. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg, "Non-linear approximation of reflectance functions," in *Proceedings of SIGGRAPH '97*, 1997.

[53] S. H. Westin, "Lafortune BRDF for RenderMan," http://www.graphics.cornell.edu/ westin/lafortune/lafortune.html.

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

APPENDIX I
DERIVATIVES OF SPHERICAL AND HEMISPHERICAL HARMONICS

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:

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

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

PLACE
PHOTO
HERE

Jaroslav Křivánek is a Ph.D. student at IRISA / INRIA Rennes and the Czech Technical University in Prague and a visiting research associate at the University of Central Florida. He earned his Masters in computer science from the Czech Technical University. His research focuses on real-time and offline lighting simulation and using visual perception for efficient lighting computation.

PLACE
PHOTO
HERE

Pascal Gautron is a Ph.D. student at IRISA / INRIA Rennes, France. He works in collaboration with the University of Central Florida and the University of Poitiers. He earned his Masters in computer science from the University of Poitiers. His main research interest is high quality real-time rendering using graphics hardware.

PLACE
PHOTO
HERE

Sumanta Pattanaik is an Associate Professor in the Computer Science Department of University of Central Florida. From 1995 to 2001 he was a research associate in the Program of Computer Graphics of Cornell University. Before that he was a post-doctoral researcher in IRISA/INRIA (1993-1995), France. His main fields of research are: realistic image synthesis and digital imaging. His current research focuses on: real-time realistic rendering, and the application of visual perception for efficient lighting computation and accurate display. He is a member of ACM SIGGRAPH, Eurographics and IEEE.

PLACE
PHOTO
HERE

Kadi Bouatouch is an electronics and automatic systems engineer (ENSEM 1974). He was awarded a PhD in 1977 and a higher doctorate on computer science in the field of computer graphics in 1989. He research interest is: global illumination, lighting simulation and remote rendering for complex environments, real-time high fidelity rendering, parallel radiosity, virtual and augmented reality and computer vision. He applied his research work to infrared simulation and tunnel lighting. He is currently Professor at the university of Rennes 1 (France) and researcher at IRISA. He is member of Eurographics, ACM and IEEE. He is and was member of the program committees of several conferences and workshops, and referee for several Computer Graphics journals like: The Visual Computer, IEEE Computer Graphics and Applications, IEEE Transaction on Visualization and Computer Graphics, IEEE transaction on image processing, ACM Transaction on Graphics, etc. He also acted as a referee for many conferences and workshops.