Temporal Radiance Caching

Pascal Gautron, Kadi Bouatouch, *Member*, *IEEE*, and Sumanta Pattanaik, *Member*, *IEEE*

Abstract—We present a novel method for fast high-quality computation of glossy global illumination in animated environments. Building on the irradiance caching and radiance caching algorithms, our method leverages temporal coherence by sparse temporal sampling and interpolation of the indirect lighting. In our approach, part of the global illumination solution computed in previous frames is reused in the current frame. Our reusing scheme adapts to the change of incoming radiance by updating the indirect lighting only where there is a significant change. By reusing data in several frames, our method removes the flickering artifacts and yields a significant speedup compared to classical computation in which a new cache is computed for every frame. We also define temporal gradients for smooth temporal interpolation. A key aspect of our method is the absence of any additional complex data structure, making the implementation into any existing renderer based on irradiance and radiance caching straightforward. We describe the implementation of our method using graphics hardware for improved performance.

Index Terms—Global illumination, animation, temporal coherence, graphics processors.

1 Introduction

Efficient computation of global illumination in complex animated environments is one of the most important research challenges in computer graphics. Achievements in this field have many applications, such as visual effects for computer-assisted movies and video games. Many approaches for such computation have been proposed in the last 20 years. Most of these methods are based on radiosity, such as those in [1], [2], [3], and [4]. However, radiosity methods are prone to visual artifacts due to undersampling or high memory consumption and computational cost due to high-quality sampling. Therefore, other approaches, such as an extension of photon mapping [5] and irradiance caching [6], [7], have been proposed.

We propose a simple and accurate method based on temporal caching for efficient computation of global illumination effects in animated environments. Our method provides rapid generation of image sequences for environments in which viewer, objects, and light sources move.

Our approach focuses on a temporal optimization for lighting computation based on the irradiance caching [8] and radiance caching [9] techniques. These caching-based techniques use sparse sampling and interpolation in the spatial domain to reduce the computational cost of indirect illumination. Our extension introduces sparse sampling and interpolation in the temporal domain to reduce the computational cost in dynamic environments. We define a temporal weighting function and temporal radiance gradients for accurate temporal interpolation, which are based on an estimate of the change of incoming radiance in the course of time. This estimate requires a basic knowledge of the incoming radiance at future time steps. To this end, we propose an inexpensive graphics processing unit (GPU)-based method using reprojection to compute this estimate.

Unlike many previous approaches, our method does not introduce any new data structure and adds very little overhead to the existing memory requirements. Since the same record in the cache can be used in several frames, the computational cost related to the computation of the records is significantly reduced. Furthermore, the records used to render an animation segment can be kept within a small memory space. Once the records are computed in the scene, our GPU-based renderer can display the dynamic globally illuminated scene in real time.

This paper is organized as follows: Section 2 presents some significant previous works in the field of global illumination for animations in dynamic scenes. Section 3 describes the key aspects of the irradiance and radiance caching algorithms. In Section 4, we highlight the problems related to using those algorithms for animation rendering and present our main contributions: the temporal weighting function, the estimation of future incoming radiance by reprojection, and the temporal gradients (TGs). Section 5 contains implementation details for efficient computation using graphics hardware. Our results are presented in Section 6.

2 Previous Work

This section describes the most significant approaches to the computation of high-quality global illumination in dynamic scenes. We also describe several methods aiming at computing approximate solutions at interactive frame rates. A thorough description of many existing methods can be found in [4]. Most significant algorithms for global illumination computation are detailed in [10], [11].

- *P. Gautron is with France Telecom, R&D Division, TECH/IRIS Lab, 4, rue du Clos Courtel, 35512 Cesson-Sévigné, France. E-mail: pgautron@irisa.fr.*
- *K. Bouatouch is with IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France. E-mail: kadi@irisa.fr.*
- *S. Pattanaik is with the University of Central Florida, School of Electrical Engineering and Computer Science, Computer Science Building, Room 251, Orlando, FL 32816-2362. E-mail: sumant@cs.ucf.edu.*

Manuscript received 1 June 2006; revised 10 Nov. 2006; accepted 24 Jan. 2007; published online 28 Feb. 2007.
Recommended for acceptance by P. Dutre.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-0078-0606.
Digital Object Identifier no. 10.1109/TVCG.2007.1039.

2.1 Interactive Methods

The methods described in this section focus on interactive approximate rendering in which the quality may be compromised to enhance interactivity. The Render Cache [12], [13] is based on the high coherence of the points visible between two subsequent frames: Most of the points visible in frame n are also visible in frame $n+1$. Therefore, the render cache stores the points visible in frame n and reprojects them onto frame $n+1$. As the visibility may change between two successive frames, some pixels may not have any corresponding visible point stored in the cache. In that case, those pixels are either filled by computing the actual corresponding hit point or by hole filling. However, the artifacts due to missing information make the Render Cache usable only in the context of fast approximate rendering. Note that we use a similar reprojection technique to estimate the temporal change of incoming lighting.

Another approach based on interactive ray tracing is Frameless Rendering [14]. This method is based on parallel asynchronous ray tracing. Each processor is assigned a fixed set of pixels to render continuously. The computation of each pixel updates the resulting image immediately. The computational power of several processors working in parallel allows interactive frame rates. This method has been enhanced by Dayal et al. [15] for using temporal coherence. As in our approach, the authors compute TGs. However, those gradients are computed in image space.

In the Shading Cache method [16], the irradiance at the vertices of a tesselated scene is adaptively computed and cached using a parallel computing cluster. The scene is rendered and displayed at interactive frame rates using hardware-based interpolation. The vertices within the region affected by object and camera movement are localized using an image-space sampling scheme and updated. The rendering shows ghosting artifacts in regions affected by moving objects; for example, color bleeding due to an object may remain behind even though the object has moved away. These artifacts eventually disappear if the viewer and objects remain static for several seconds.

A similar drawback is present in the work by Dmitriev et al. [5]. The authors present a density estimation technique using photon mapping [17] and quasi Monte Carlo. The photon map is updated when the user moves an object. For each frame, the algorithm detects which photon paths should be recomputed. The update consists in using quasi Monte Carlo to trace a group of photons following a similar path. This method is well suited for interactive manipulation of objects but suffers from delays between the object motion and the update of the photon map.

2.2 Irradiance Caching and Final Gathering

Several approaches have been proposed to accelerate the final gathering process used to render photon maps in dynamic scenes. The methods described in this section exploit temporal coherence in the context of final gathering using irradiance caching [8].

Tawara et al. [18] propose a two-step method for computing indirect lighting in dynamic scenes. In the first step, they compute an irradiance cache using only the static objects of the scene. Then, for each frame, they update the irradiance cache by introducing the dynamic objects at their frame-specific positions. This method shows significant speedup but is restricted to animations during which "lighting conditions do not change significantly" [18]. Otherwise, the static irradiance cache has to be recomputed from scratch, leading to the temporal artifacts inherent in this method.

Another approach by Tawara et al. [6] is based on the detection of the incoming radiance changes related to the displacement of objects. The rays traced for computing irradiance records are stored and updated using either a user-defined update rate or an adaptive rate based on the number of rays hitting a dynamic object. This method requires a large amount of memory to store the rays, making it difficult to use in complex scenes where many records have to be computed.

The method described in [7] is based on a data structure called *anchor*. Using this structure, the algorithm detects changes in visibility and incoming radiance for each irradiance record during an animation sequence. Therefore, the irradiance values stored in the cached records can be efficiently updated. However, even though this method provides high-quality results, it requires the explicit storage of the rays used to compute each irradiance record and, hence, is highly memory intensive.

In this paper, we present a method exploiting temporal coherence for using irradiance and radiance caching [8], [9] in dynamic scenes. Section 3 presents a brief overview of irradiance and radiance caching.

3 Irradiance and Radiance Caching: An Overview

The irradiance and radiance caching algorithms are based on the following observation: "The indirect illuminance tends to change slowly over a surface" [8]. These methods take advantage of spatial coherence by sparsely sampling and interpolating indirect incoming radiance. Irradiance caching [8] is designed for the computation of indirect diffuse lighting. Radiance caching [9] extends the approach proposed by Ward et al. [8] to glossy interreflections.

The irradiance and radiance caching algorithms are very similar. In irradiance caching, a record stores the irradiance at a given point on a surface. In radiance caching, a record stores the projection coefficients of the incoming radiance for the hemispherical harmonics basis [19]. The interpolation schemes are also similar: The irradiance stored in the irradiance cache and the coefficients stored in the radiance cache undergo weighted interpolation. The method described in this paper is similarly applicable to both caching techniques. For the simplicity of explanation, we only consider irradiance caching in most of the following discussions. Specific details about radiance caching are given in Section 4.6.

Irradiance Interpolation. Let us consider a point $\mathbf{p}$ in a scene. In [8], an estimate of the irradiance $E(\mathbf{p})$ at point $\mathbf{p}$ is given by

$$E(\mathbf{p}) = \frac{\sum_{K \in S_r} w_K(\mathbf{p}) E_K(\mathbf{p})}{\sum_{K \in S_r} w_K(\mathbf{p})}, \quad (1)$$

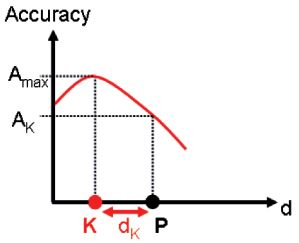


Fig. 1. The *accuracy* of the reconstructed lighting is maximum at the location of actual computation of record $\mathbf{K}$. The accuracy decreases when the lighting is extrapolated at a point $\mathbf{p}$ in the neighborhood of $\mathbf{K}$.

where S_r represents the set of irradiance records surrounding $\mathbf{p}$. For each record $K \in S_r$, $w_K(\mathbf{p})$ is the weighting function of record K evaluated at point $\mathbf{p}$. $E_K(\mathbf{p})$ is the contribution of record K to the computed incoming lighting at point $\mathbf{p}$. In the next two paragraphs, we describe the irradiance weighting function and the computation of the contribution of the records using irradiance gradients.

Weighting Function. In the irradiance caching algorithm, a given record contributes to the indirect lighting of points with similar lighting. If the incoming lighting at the record location changes rapidly in the neighborhood of the record, its weight should be small in the surrounding area. Conversely, if the lighting changes slowly, the weight of the record should be high. Therefore, the weighting function is based on an approximation of the potential change in incoming radiance with respect to displacement and rotation.

The weighting function is chosen as the inverse of this estimated change. At a point $\mathbf{p}$ with normal $\mathbf{n}$, the weighting function of record K is defined as

$$w_K(\mathbf{p}) = \frac{1}{\frac{\|\mathbf{p}-\mathbf{p_K}\|}{R_K} + \sqrt{1-\mathbf{n}\cdot\mathbf{n_K}}}, \tag{2}$$

where $\mathbf{p_K}$, $\mathbf{n_K}$, and R_K are, respectively, the location of record K, its normal, and the harmonic mean distance to the objects visible from $\mathbf{p_K}$.

This weighting function is not only used for interpolation but also to determine the set of records $S_r(\mathbf{p})$ contributing to the incoming radiance at point $\mathbf{p}$:

$$S_r(\mathbf{p}) = \{K | w_K(\mathbf{p}) \geq a\}, \tag{3}$$

where a is a user-defined accuracy value.

Irradiance Gradients for Accurate Extrapolation. The irradiance contribution of record K to the irradiance at point $\mathbf{p}$ depends on the translation and rotation gradients of the irradiance. The gradient computations proposed in [20], [9], and [21] present several methods to estimate the change in incoming radiance with respect to translation and rotation. Figs. 1a and 1b in [20] illustrate the quality improvement obtained using gradients. Ward and Heckbert [20] define the contribution of record K to point $\mathbf{p}$ with normal $\mathbf{n}$ as

$$E_K(\mathbf{p}) = E_K + (\mathbf{n_K} \times \mathbf{n}) \cdot \nabla_\mathbf{r} + (\mathbf{p} - \mathbf{p_K}) \cdot \nabla_\mathbf{p}, \tag{4}$$

where E_K, $\nabla_\mathbf{r}$, and $\nabla_\mathbf{p}$ are, respectively, the irradiance, the rotation gradient, and the translation gradient stored in record K.

The irradiance and radiance caching algorithms provide an accurate and efficient way of computing global illumination in static scenes. Furthermore, several methods have been proposed to enhance the quality and efficiency of these algorithms [22], [23]. However, problems arise when using this approach for global illumination computation in dynamic environments. Section 4 presents the typical problems encountered when using the irradiance caching algorithm in dynamic scenes and describes our temporal optimization method.

4 Temporal Irradiance Caching

4.1 Quality Measurement

Both our method and the (ir)radiance caching algorithms rely on sparse sampling and approximations (interpolations and extrapolations) for fast global illumination computation. Intuitively, the quality of the reconstructed lighting is high at the location and time of actual computation and decreases as the extrapolation point and time get away from the record. In this paper, the quality of the reconstructed lighting is named *accuracy*. The maximum (100 percent) accuracy is obtained when the lighting is explicitly computed (that is, at the location and time of record creation) and decreases as the point of interest gets away from the record (Fig. 1).

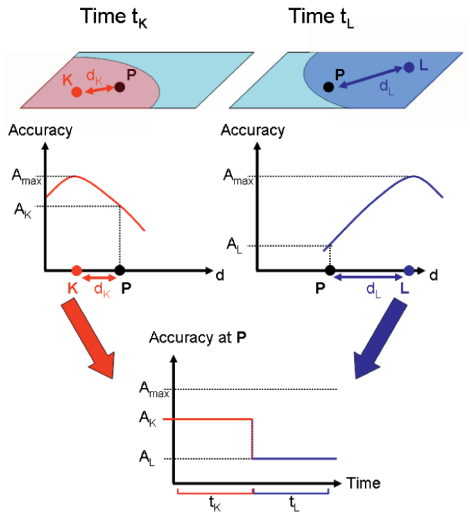
4.2 Irradiance Caching in Dynamic Scenes

As described in the previous section, irradiance caching leverages spatial coherency of indirect lighting and, thus, reduces the computation time of global illumination. In dynamic scenes, a simple and commonly used approach is to discard all the cached records and start with an empty cache at the beginning of each frame. This indiscriminate discard of records amounts to significant waste of computational effort. Additionally, the resulting animation video quality may be poor. The reason for this is that the distributions of record location in frames n and $n+1$ are likely to be different. Fig. 2 illustrates the consequence of the change of record distribution: Since the gradients extrapolate the change of incoming radiance, they are not completely accurate compared to the ground truth. Therefore, the accuracy of the lighting reconstructed by irradiance caching is not constant over a surface: The accuracy is maximum at the record location and decreases as the extrapolation point gets away from the record. Therefore, changing the distribution of records also changes the distribution of the accuracy, yielding high-frequency flickering artifacts. Thus, the simple use of irradiance caching in dynamic scenes gives rise to poor animation quality.

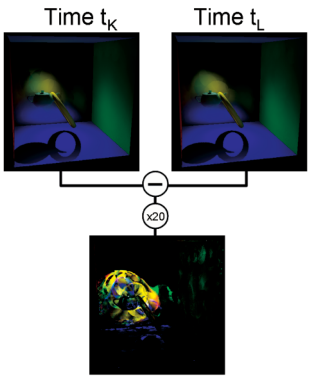
In this paper, we propose a simple and unified framework for view-dependent global illumination in dynamic environments with predefined animation in which objects, light sources, and cameras can move.

4.3 Overview of Temporal Radiance Caching

In [8], [20], the interpolation scheme of Ward et al. converts the high-frequency noise of Monte Carlo path tracing into low frequency error. As shown above, this result is achieved by sparse sampling, extrapolation, and interpolation in space. In this paper, our aim is to convert the high-



(a)



(b)

Fig. 2. (a) Changing the location of the records between successive frames can significantly modify the accuracy of the lighting at a point $\mathbf{p}$. (b) Therefore, the incoming radiance at this point can change noticeably between two frames, yielding flickering artifacts (b). Note that the maximum accuracy A_{max} is obtained at the point and time of actual computation of the incoming radiance.

frequency temporal noise (that is, the flickering artifacts) into low-frequency temporal errors by sparse sampling, extrapolation, and interpolation in the temporal domain. More precisely, we amortize the cost of record computation over several frames by performing a sparse temporal sampling and temporal interpolation of the irradiance (Algorithm 1). When a record K is created at frame n, the future incoming lighting is estimated. This estimate is first used to compute the ratio between the current and future lightings. In the spirit of [8], we define our temporal weighting function w_K^t as the inverse of the temporal change. Hence, the number of frames in which K can contribute is inversely proportional to the future change of incoming radiance. Since the actual computation of the future incoming lighting may be expensive, we use a simple reprojection technique for estimating the future lighting using the data sampled at the current frame. As the irradiance at a point is extrapolated using the irradiance and gradients of neighboring records, we propose TGs for smooth temporal interpolation and extrapolation of the irradiance.

Algorithm 1 Temporal Radiance Caching
```
for all frames n do
  for all existing records K do
    if w_K^t(n) is greater than a threshold then
      Use K in frame n
    end if
  end for
  for all points p where a new record is needed do
    Sample the hemisphere above p
    Estimate the future incoming lighting (Section 4.7)
    Generate w_K^t (Section 4.4)
    Compute the TGs (Section 4.5)
    Store the record in the cache
  end for
end for
```

4.4 Temporal Weighting Function

The temporal weighting function expresses the confidence on a given record as a function of time. Using a derivation similar to that in [8], we define the temporal change ϵ^t of incoming radiance between time t and t_0 as

$$\epsilon^t = \frac{\partial E}{\partial t}(t_0)(t - t_0). \tag{5}$$

This derivative $\frac{\partial E}{\partial t}(t_0)$ can be approximated using estimates of incoming radiance at two successive times t_0 and t_1, denoted E_0 and E_1:

$$\frac{\partial E}{\partial t}(t_0) \approx \frac{E_1 - E_0}{t_1 - t_0} \tag{6}$$

$$= \frac{\tau E_0 - E_0}{t_1 - t_0} \quad \text{where } \tau = E_1/E_0 \tag{7}$$

$$= E_0 \frac{\tau - 1}{t_1 - t_0}. \tag{8}$$

In our method, the time range of the animation is discretized into integer frame indices. Therefore, we always choose $t_1 - t_0 = 1$, that is, E_1 and E_0 represent the estimated irradiance at two successive frames.

As in [8], we define the temporal weighting function as the inverse of the change, excluding the term E_0:

$$w_K^t(t) = \frac{1}{(\tau - 1)(t - t_0)}, \tag{9}$$

where $\tau = E_1/E_0$ is the *temporal irradiance ratio*.

This function can be evaluated and tested against a user-defined accuracy value a^t. A record K created at t_0 is allowed to contribute to the image at t if

$$w_K^t(t) \geq 1/a^t. \tag{10}$$

The temporal weighting function is used to adjust the time segment during which a record is considered valid. Since a given record can be reused in several frames, the computational cost can be significantly reduced.

However, (7) shows that if the environment remains static starting from frame t_0, we obtain $\tau = 1$. Therefore, (10) shows that w_K^t is infinite for any frame and, hence, record K is allowed to contribute at any time $t > t_0$. However, since part of the environment is dynamic, the inaccuracy becomes

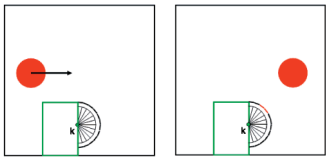


Fig. 3. When record K is created at time t_K, the surrounding environment is static $(\tau_K = 1)$. However, the red sphere is visible from the K n frames later. The user-defined value $\delta_{t_{max}}$ prevents the record from contributing if $n > \delta_{t_{max}}$, then reducing the risk of using obsolete records. (a) Time t_K. (b) Time $t_K + n$.

significant when $t - t_0$ gets high (see Fig. 3). This is a limitation of our technique for estimating the temporal change of incoming radiance, which determines the lifespan of a record by considering only the change between E_t and E_{t+1}. Therefore, we introduce a user-defined value $\delta_{t_{max}}$ limiting the length of the validity time segment associated with each record. If (11) does not hold, we decide that the record cannot be reasonably used:

$$t - t_0 < \delta_{t_{max}}. \tag{11}$$

This reduces the risk of using obsolete records, which allows the user to control the artifacts due to residual global illumination effects also known as "ghosts," which commonly appear in interactive methods. However, as a and a^t, this value must be user-defined by trial and error to obtain the best results. If $\delta_{t_{max}}$ is too low, many records may be recomputed unnecessarily. Setting $\delta_{t_{max}} = 1$ implies the recomputation of each record for each frame. In this case, the resulting performance would be similar to the classical per-frame computation. Nevertheless, this would completely avoid the ghosting artifacts, since the indirect lighting is continually computed from scratch. If $\delta_{t_{max}}$ is set too high, the same records might be reused in too many frames. Hence, artifacts due to the residual global illumination effects are likely to appear in the vicinity of the moving objects, degrading the quality of the rendered frame. Consequently, such a high value significantly reduces the rendering time at the detriment of the temporal accuracy.

However, abrupt discarding of a record reintroduces the flickering problem described in Section 4.2. As proposed in [18], we avoid this problem by keeping track of the location of the records over time. Let us consider a record K located at point $\mathbf{p_K}$. If K was allowed to contribute to the previous frame and cannot be reused in current frame, a new record l is created at the same location, that is, $\mathbf{p_l} = \mathbf{p_K}$ (Fig. 4b). Since the location of visible records remains constant in space, the accuracy at a given point tends to be constant over time, hence reducing the flickering artifacts. Note that the locations of records remain constant even though they lie on dynamic objects (Fig. 5).

The temporal weighting function provides a simple and adaptive way of leveraging temporal coherence by introducing an aging method based on the change of incoming radiance. Nevertheless, Fig. 4c shows that the accuracy of the computation is still not continuous in the course of time. Replacing obsolete records with new ones creates a discontinuity of accuracy, which causes the visible flickering

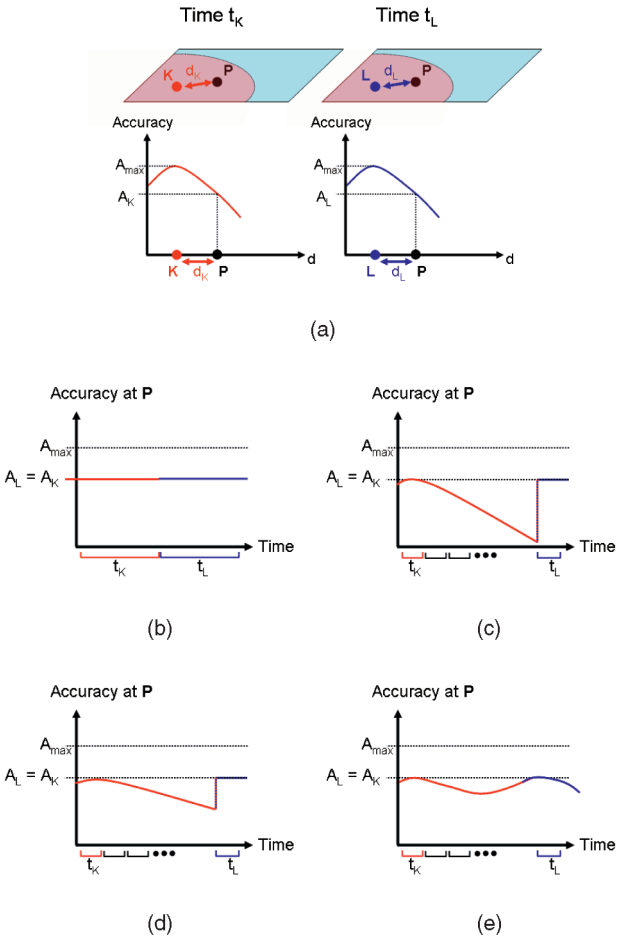


Fig. 4. Empirical shape of the accuracy (a) at a fixed time with respect to space and (b), (c), (d), (e) at a fixed location $\mathbf{p}$ with respect to time. (b) New record after $\delta = 1$. (c) New record after $\delta = n$ frames with no temporal gradient. (d) New record after $\delta = n$ frames with extrapolated temporal gradient. (e) New record after $\delta = n$ frames with an interpolated temporal gradient. If the records are located at the same point between successive frames as in (a), the temporal accuracy is improved as shown in (b). However, as shown in (c), flickering artifacts due to the temporal discontinuities of accuracy may appear when a record is reused in several frames, then recomputed. (d) Extrapolated TGs decrease the amplitude of the discontinuity in one pass, reducing flickering. (e) Interpolated TGs use two passes to eliminate the discontinuity by smoothing out the temporal changes.

artifacts. Therefore, we propose TGs to generate a smooth and less noticeable transition between successive records.

4.5 Temporal Gradients

TGs are conceptually similar to classical irradiance gradients. Instead of representing the incoming radiance change with respect to translation and rotation, those gradients represent how the incoming radiance gets altered over time.

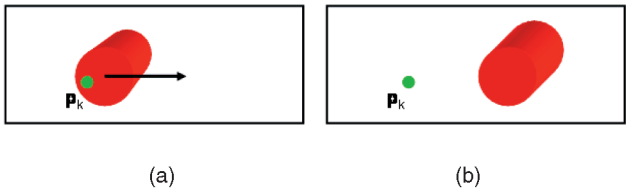


Fig. 5. Record K created at time t_K remains at point $\mathbf{p_K}$ even though it lays on a dynamic object. (a) Time t_K. (b) Time $t_K + n$.

In the context of irradiance caching, (4) shows that the irradiance at point $\mathbf{p}$ is estimated using rotation and translation gradients. The temporal irradiance gradient of record K at a given point $\mathbf{p}$ with normal $\mathbf{n}$ is derived from (4) as

$$\nabla^{\mathbf{t}}(\mathbf{p}) = \frac{\partial}{\partial t}(E_K + (\mathbf{n_K} \times \mathbf{n}) \cdot \nabla_{\mathbf{r}} + (\mathbf{p} - \mathbf{p_K}) \cdot \nabla_{\mathbf{p}}), \quad (12)$$

where

- $\nabla_{\mathbf{r}}$ and $\nabla_{\mathbf{p}}$ are the rotation and translation gradients.
- $\mathbf{p_K}$ and $\mathbf{n_K}$ are the location and normal of record K.

As described in Section 4.4, we keep the location of all records constant over time. We choose any point of interest $\mathbf{p}$ with normal $\mathbf{n}$, which is also constant over time. Therefore, $\mathbf{n_K} \times \mathbf{n}$ and $\mathbf{p} - \mathbf{p_K}$ are constant with respect to time. The equation for TGs becomes

$$\nabla^{\mathbf{t}}(\mathbf{p}) = \nabla^{\mathbf{t}}_{\mathbf{E_K}} + (\mathbf{n_K} \times \mathbf{n}) \cdot \nabla^{\mathbf{t}}_{\nabla_{\mathbf{r}}} + (\mathbf{p} - \mathbf{p_K}) \cdot \nabla^{\mathbf{t}}_{\nabla_{\mathbf{p}}}, \quad (13)$$

where

- $\nabla^{\mathbf{t}}_{\mathbf{E_K}} = \frac{\partial E_K}{\partial t}$ is the *TG of irradiance*.
- $\nabla^{\mathbf{t}}_{\nabla_{\mathbf{r}}} = \frac{\partial \nabla_{\mathbf{r}}}{\partial t}$ is the *TG of rotation gradient*.
- $\nabla^{\mathbf{t}}_{\nabla_p} = \frac{\partial \nabla_{\mathbf{p}}}{\partial t}$ is the *TG of translation gradient*.

Using (13), the contribution of record K created at time t_K to the incoming radiance at point $\mathbf{p}$ at time t is estimated by

$$\begin{aligned} E_K(\mathbf{p}, t) = {} & E_K + \nabla^{\mathbf{t}}_{\mathbf{E_K}}(t - t_K) \\ & + (\mathbf{n_K} \times \mathbf{n}) \cdot (\nabla_{\mathbf{r}} + \nabla^{\mathbf{t}}_{\nabla_{\mathbf{r}}}(t - t_K)) \\ & + (\mathbf{p_K} - \mathbf{p}) \cdot (\nabla_p + \nabla^{\mathbf{t}}_{\nabla_{\mathbf{p}}}(t - t_K)). \end{aligned} \quad (14)$$

This formulation represents the temporal change of the incoming radiance around p_K as three vectors. These vectors represent the change of the incoming radiance at point $\mathbf{p_K}$ and the change of the translation and rotation gradients over time. The values of these vectors can be easily computed using the information generated in Section 4.7. Since our method estimates the incoming radiance at time $t+1$ using the information available at time t, we define *extrapolated* TGs as

$$\nabla^{\mathbf{t}}_{\mathbf{E_K}} \approx E(t_K + 1) - E(t_K) \quad (15)$$

$$\nabla^{\mathbf{t}}_{\nabla_{\mathbf{r}}} \approx \nabla_{\mathbf{r}}(t_K + 1) - \nabla_{\mathbf{r}}(t_K) \quad (16)$$

$$\nabla^{\mathbf{t}}_{\nabla_{\mathbf{p}}} \approx \nabla_{\mathbf{p}}(t_K + 1) - \nabla_{\mathbf{p}}(t_K). \quad (17)$$

However, as illustrated in Fig. 4d, these TGs do not remove all the discontinuities in the animation. When a record K is replaced by record l, the accuracy of the result exhibits a possible discontinuity, yielding some flickering artifacts. As explained in Section 4.4, this problem can be avoided by keeping track of the history of the records: When record K becomes obsolete, a new record l is created at the same location. Since $t_l > t_K$, we can use the value of incoming radiance stored in l to compute *interpolated* TG for record K:

$$\nabla^{\mathbf{t}}_{\mathbf{E_K}} \approx (E_l - E_K)/(t_l - t_K) \quad (18)$$

$$\nabla^{\mathbf{t}}_{\nabla_{\mathbf{r}}} \approx (\nabla_{\mathbf{r}}(t_l) - \nabla_{\mathbf{r}}(t_K))/(t_l - t_K) \quad (19)$$

$$\nabla^{\mathbf{t}}_{\nabla_{\mathbf{p}}} \approx (\nabla_{\mathbf{p}}(t_l) - \nabla_{\mathbf{p}}(t_K))/(t_l - t_K). \quad (20)$$

As illustrated in Fig. 4e, the TGs enhance the continuity of the accuracy, hence removing the flickering artifacts. However, the gradients only account for the first derivative of the change of incoming lighting, which temporally smoothes the changes. Although this method proves accurate in scenes with smooth changes, it should be noted that the gradient-based temporal interpolation may introduce ghosting artifacts when used in scenes with very sharp changes of illumination. In this case, a^t and $\delta_{t_{max}}$ must be reduced to obtain a sufficient update frequency.

4.6 Extension to Radiance Caching

4.6.1 Temporal Weighting Function

Our temporal weighting function is based on an estimate of the ratio between the current and future incoming radiances. In the context of radiance caching, the directional information of the incoming radiance must be very accurate. Therefore, we define the temporal radiance ratio as

$$\tau_{radiance} = max\{\lambda^i_1/\lambda^i_0, 0 \geq i < n\}, \quad (21)$$

where λ^i_0 and λ^i_1, are respectively the ith projection coefficient of the current and future incoming lighting. By using the maximum ratio, our method can account for directional change of incoming radiance without loss of accuracy.

4.6.2 Temporal Gradients

As described in [9], the rotational gradient is not necessary in the radiance caching algorithm. The incoming radiance function and the translation gradient being represented using projection coefficients, we compute the TG of each coefficient independently. Thus, the problem reduces to the computation of the above equations for each coefficient.

However, as shown in (7), the determination of our temporal weighting function and extrapolated gradients relies on the knowledge of the incoming radiance at the next time step, E_{t_0+1}. Since the explicit computation of E_{t_0+1} would introduce a significant computational overhead, we propose a simple and accurate estimation method based on reprojection.

4.7 Estimating E_{t_0+1}

We use the reprojection and hole-filling approach proposed by Walter et al. [12]. However, it must be noted that, while Walter et al. use reprojection for interactive visualization using ray tracing, our aim is to provide a simple and reliable estimate of the incident radiance at a given point at time $t_0 + 1$ by using the data acquired at time t_0 only. This estimate will be used to determine the lifespan of the records by evaluating our temporal weighting function.

In the context of predefined animation, the changes in the scene are known and accessible at any time. When a record K is created at time t_0, the hemisphere above $\mathbf{p_K}$ is sampled (Fig. 6a) to compute the incoming radiance and gradients at this point. Since the changes between times t_0 and $t_0 + 1$ are known, it is possible to reproject the points visible at time t_0 to obtain an estimate of the visible points at time $t_0 + 1$ (Fig. 6b). The outgoing radiance of the reprojected visible points can be estimated by accounting for the rotation and displacement of both objects and light

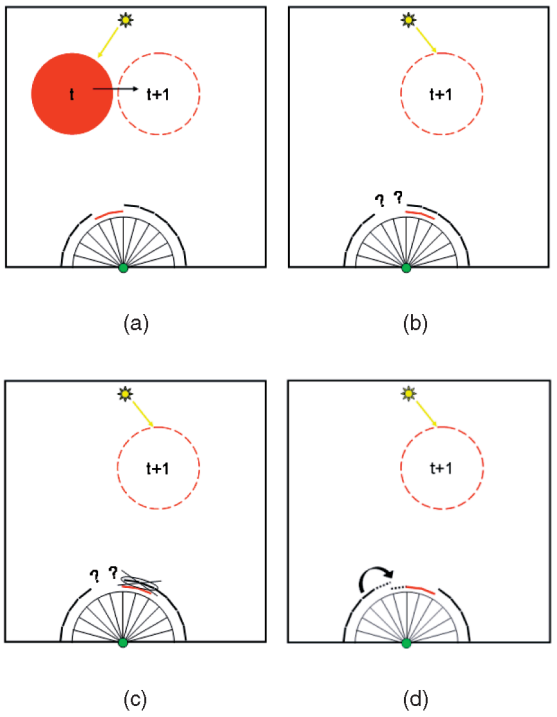


Fig. 6. (a) The hemisphere is sampled at time t as in the classical irradiance caching process. (b) For each ray, our method determines where each visible point will be located at time $t+1$ by reprojection. (c) Distant overlapping points are removed using a depth test, whereas (d) resulting holes are filled using the farthest neighboring values.

sources. In overlapping areas, a depth test accounts for the occlusion change (Fig. 6c).

However, some parts of the estimated incoming radiance may be unknown (holes) due to displacement and overlapping of visible objects (Fig. 6d). As proposed in [12], we use a simple hole-filling method: Each hole is filled using the background values, yielding a plausible estimate of the future indirect lighting.

As shown in Fig. 7 and Table 1, the reprojection reduces the error in the estimate of the future incoming lighting. Those errors were measured by comparing the irradiances

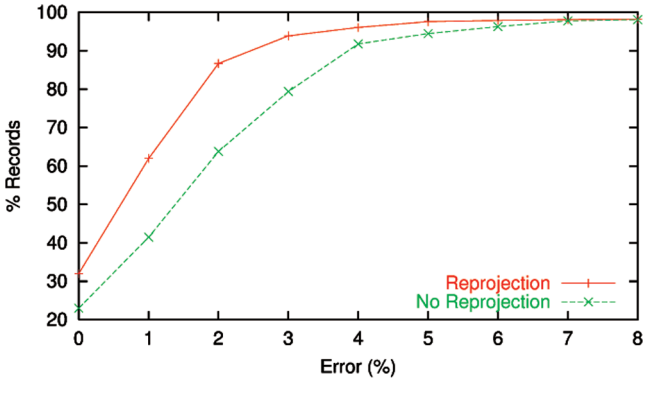


Fig. 7. Error between the actual lighting at $t+1$ and the lighting at $t+1$ estimated with and without reprojection. This plot represents the percentage of records for which the estimate of the future incoming lighting is below a given RMS error level. The reprojection reduces the overall error in the estimate compared to a method without reprojection (that is, where the lighting is considered temporally constant). Errors are computed using 4,523 records.

TABLE 1
RMS Error Between the Actual Lighting at $t+1$ and the Lighting at $t+1$ Estimated with and without Reprojection

Error	Reprojection	No Reprojection
Min	0%	0%
Max	30%	32%
Mean	2.9%	3.7%
Median	1.6%	2.4%

(Based on 4,523 values).

at time $t+1$ with the irradiances estimated using records created at time t.

Our method provides a simple way of extending irradiance caching to dynamic objects and dynamic light sources by introducing a temporal weighting function and TGs. In Section 5, we discuss the implementation of our method on a GPU for increased performance.

5 GPU Implementation

Our method has been implemented within a GPU-based renderer for irradiance and radiance caching. First, we detail the implementation of the incoming radiance estimate by reprojection (Section 4.7). Then, we describe how the GPU can be simply used in the context of radiance cache splatting to discard useless records and avoid their replacement.

Reprojection of incoming radiance. As shown in Section 4.4, the computation of the temporal weighting function and TGs for a given record K requires an estimate of the radiance reaching point $\mathbf{p_K}$ at the next time step. This estimate is obtained through reprojection (Section 4.7), provided that the position of the objects at the next time step is known. Therefore, for a given vertex v of the scene and a given time t, we assume that the transformation matrix corresponding to the position and normal of v at time $t+1$ is known. We assume that such matrices are available for light sources as well. Using the method described in [24], a record K can be generated by rasterizing the scene on a single plane above point $\mathbf{p_K}$. In a first pass, during the rasterization at time t, shaders can output an estimate of the position and incoming lighting at time $t+1$ of each point visible to $\mathbf{p_K}$ at time t. This output can be used to reconstruct an estimate of the incoming radiance function at time $t+1$.

This estimate is obtained in a second pass: Each projected visible point generated in the first pass is considered a vertex. Each of those vertices is sent to the graphics pipeline as a pixel-sized point. The result of the rasterization process is an estimate of the incoming radiance function at time $t+1$. Since the size of the sampling plane is usually small (typically, 64×64), this process is generally much faster than resampling the whole scene.

During the reprojection process, some fragments may overlap. Even though the occlusion can be simply solved by classical Z-Buffer, the resulting image may contain holes (Fig. 6d). These holes are created at the location of dynamic objects. Since the time shift between two successive frames is very small, the holes are also small. As described in Section 4.7, we use a third pass to fill the holes using the local background (that is, the neighboring value with the

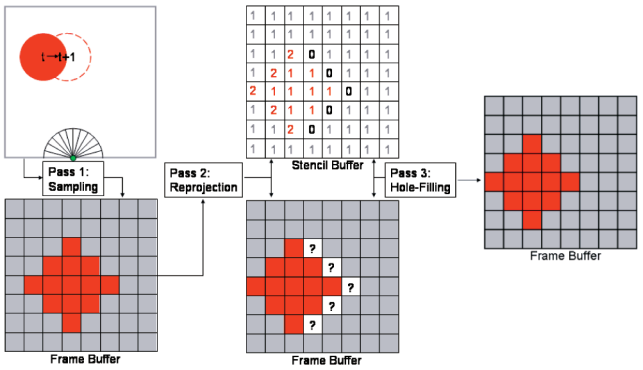


Fig. 8. Reprojection using the GPU. The first pass samples the scene to gather the required information. Then, the visible points are reprojected to their estimated position at next time step. During this pass, each rendered fragment increments the stencil buffer. Finally, the holes (that is, where the stencil value is 0) are filled using the deepest neighboring values.

highest depth). This computation can be performed efficiently on the GPU using the stencil buffer with an initial value of 0. During the reprojection process, each rasterized point increments the stencil buffer. Therefore, the hole-filling algorithm must be applied only on pixels where the stencil buffer is still 0. The final result of this algorithm is an estimate of the incoming radiance at time $t+1$, generated entirely on the GPU (Fig. 8). This estimate is used in the extrapolated TGs and the temporal weighting function. As shown in (10), this latter defines a maximum value of the lifespan of a given record and triggers its recomputation. However, this recomputation is not always necessary.

Radiance cache splatting. The radiance cache splatting method [24] is based on a simple observation of the spatial weighting function described in [8]: An (ir)radiance record K cannot contribute to the lighting of points located outside a sphere of influence centered at p_K. Therefore, the indirect lighting at visible points can be obtained by splatting the sphere corresponding to record K on the image plane. This splatting is performed on graphics hardware by rasterizing a quadrilateral tightly bounding the sphere. For each fragment within this quadrilateral, a fragment program evaluates the weighting function for record K and verifies whether record K is allowed to contribute to the indirect lighting of the visible point corresponding to this fragment (see (3)). The weighted average described in (1) is computed using simple hardware blending. Using this method, high-quality global illumination can be displayed at interactive frame rates.

Replacement/deletion method. As described in the previous sections, the flickering artifacts of the lighting come from the temporal discontinuities of the accuracy. Therefore, if a record cannot contribute to the current image (that is, it is out of the view frustum or occluded), it can be simply deleted instead of being replaced by a novel, up-to-date record. This avoids the generation and update of a "trail" of records following dynamic objects (Fig. 9), hence reducing the memory and computational costs. In the context of radiance cache splatting [24], this decision can be easily made using hardware occlusion queries: During the last frame of the lifespan of record K, an occlusion query is issued as the record is rasterized. In the next frame, valid records are first rendered. If a record K is now obsolete, the result of the occlusion query is read from the GPU. If the number of covered pixels is 0, the record is discarded. Otherwise, a new record l is computed at location $\mathbf{p_l} = \mathbf{p_K}$.

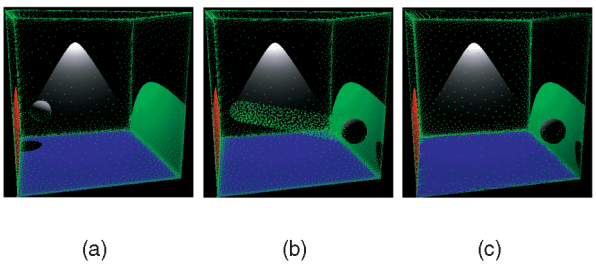


Fig. 9. The sphere moves from the left to the right of the Cornell Box. (a) At time 1, records (represented by green points) are generated to compute the global illumination solution. When the sphere moves, new records are created to evaluate the incoming radiance on the sphere. (b) If every record is permanently kept up to date, a "trail" of records lies on the path of the dynamic sphere. (c) Using our method, only useful records are updated.

The hardware occlusion queries are very useful, but they suffer from high latency. However, in our method, the result of a query is not needed immediately. Between the query issue and the reading of the record coverage, the renderer renders the other records, then switches the scene to next frame and renders valid records. In practice, the average latency appeared to be negligible (less than 0.1 percent of the overall computing time). Besides, in our test scenes, this method reduces the storage and computational costs by up to 25-30 percent.

6 Results

This section discusses the results obtained using our method and compares them with the classical method in which a new cache is computed for each frame. This latter method is referred to as *per-frame computation* in the remainder of this section. The images, videos, and timings have been generated using a 3.8 GHz Pentium 4 with 2 Gbytes of RAM and an nVidia GeForce 7800 GTX with 512 Mbytes. The scene details and timings are summarized in Table 2. The animations are presented in the accompanying video.

Cube in a box. This very simple diffuse scene (Fig. 12a) exhibits high flickering when no TGs are used. Along with a significant speedup, our method reduces the flickering artifacts by using extrapolated TGs. Such artifacts are unnoticeable with interpolated gradients. The animations are generated using $a^t = 0.05$ and a maximum lifespan of

TABLE 2
Test Scenes and Timings

Scene	Nb. Poly	Nb. Frames	Per-Frame Comp.	Our Method	Speedup
Cube in a Box	24	400	2048s	269s	7.62
Moving Light	24	400	2518s	2650s	0.95
Flying Kite	28K	300	5109s	783s	6.52
Japanese Interior	200K	750	13737s	7152s	1.9
Spheres	64K	200	3189s	753s	4.24

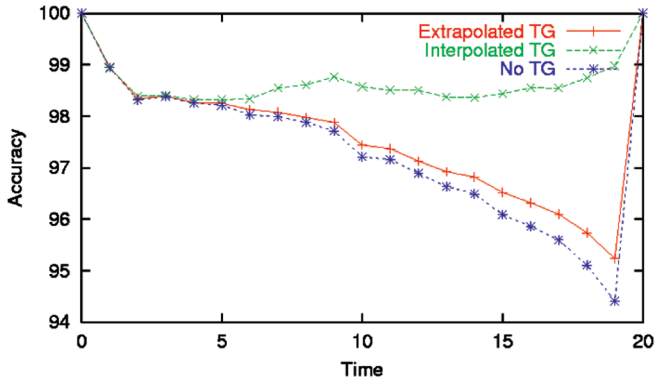


Fig. 10. Plot of the temporal accuracy as a function of time obtained in scene Cube in a Box by creating records at time 0 and extrapolating their value until time 19. The accuracy value at a time t is obtained by computing the error between the extrapolated values and the actual lighting at time t. At time 20, the existing records are discarded and replaced by up-to-date records with maximum accuracy. The TGs provide a better approximation compared to the approach without those gradients. Using interpolated gradients, the accuracy is continuous and remains above 98 percent.

20 frames. The per-frame computation requires 772,000 records to render the animation. In our method, only 50,000 records are needed, yielding a memory load of 12.4 Mbytes. Fig. 10 shows the accuracy values obtained with and without TGs. The remaining flickering of the extrapolated TGs are due to the discontinuities of accuracy. Since our aim is high-quality rendering, the following results focus on interpolated TGs that avoid discontinuities.

Moving light. A similar scene (Fig. 12b) illustrates the behavior of our algorithm in the context of dynamic light sources. The bottom of the box is tiled to highlight the changes of indirect lighting. Due to the highly dynamic indirect lighting, the lifespan of the records is generally very short, yielding frequent updates of irradiance values. Compared to per-frame computation, our method renders the animation with higher quality in a comparable time.

Flying kite. In a more complicated textured scene (Fig. 12c), our algorithm also provides a significant quality improvement while drastically reducing the computation time. In the beginning of the animation, the change of indirect lighting is small, and, hence, the records can be reused in several frames. However, when the kite comes down, its dynamic reflection on the ceiling and wall is clearly noticeable. Using our temporal weighting function, the global illumination solution of this zone is updated at a fast pace, avoiding ghosts in the final image (Fig. 11).

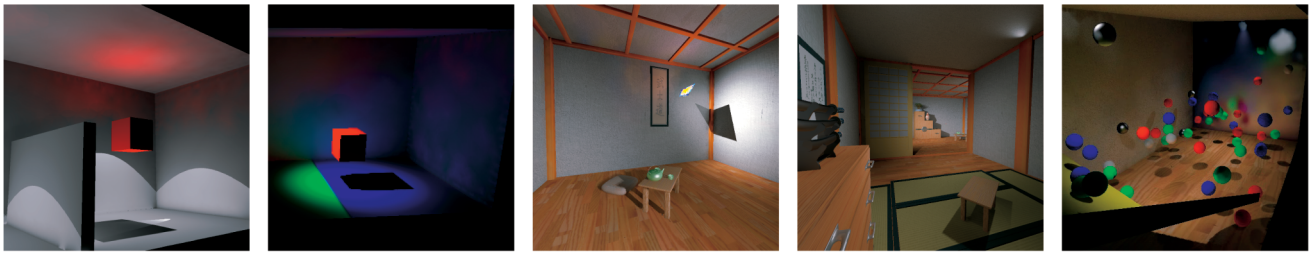
(a) (b)

Fig. 11. (a) Actual sampling frame. When computing the global illumination solution for the current frame, our method estimates where the lighting changes. (b) Records lifespan. The lifespan of each generated record is computed by estimating the future change of lighting. Green and red colors respectively represent long and short lifespans.

Japanese interior. In this more complex scene (Fig. 12d), the glossy and diffuse objects are rendered using, respectively, the radiance and irradiance caching algorithms. The animation illustrates the features of our method: Dynamic nondiffuse environment and important changes of indirect lighting. In the beginning of the animation, the scene is lit by a single dynamic light source. In this case, TGs suppress the flickering artifacts present in the per-frame computation but do not provide a significant speedup ($1.25\times$). In the remainder of the animation, most of the environment is static, even though some dynamic objects generate strong changes in the indirect illumination. Our TGs take advantage of this situation by adaptively reusing records in several frames. The result is the elimination of flickering and a significant speedup (up to $9\times$) compared to per-frame computation. During the generation of this animation, the average latency introduced by occlusion queries is 0.001 percent of the overall rendering time.

Spheres. This scene features complex animation with 66 diffuse and glossy bouncing spheres and a glossy back wall (Fig. 12e). Our method eliminates the flickering while reducing the computational cost of a factor 4.24. We used a temporal accuracy value $a^t = 0.05$ and a maximum record lifespan $\delta_{t_{max}} = 5$.

Comparison with Monte Carlo path tracing. As the (ir)radiance caching algorithms introduce low-frequency spatial errors, our method introduces low-frequency temporal errors. Therefore, the obtained images contain both spatial and temporal errors. In a sequence of 100 images of the *Cube in a Box* scene, the average root-mean-square (RMS) error between our method and the reference solution is 0.139



(a) (b) (c) (d) (e)

Fig. 12. Images of scenes discussed in Section 6. (a) Cube in a Box. (b) Moving light. (c) Flying kite. (d) Japanese Interior. (e) Spheres.

(a) (b) (c)

Fig. 13. Images obtained using (a) Monte Carlo path tracing (16,000 rays per hemisphere at each pixel) and (b) our method. (c) The image obtained by differencing (a) and (b). The pixel values in (c) are multiplied by 5 to highlight the differences.

(Fig. 13). Even though the results obtained using our method exhibit differences compared to the reference solution, the images obtained are a reliable estimate of the global illumination solution.

Computational overhead of reprojection. During the computation of a record, our method evaluates the value of the incoming lighting for both current and next time steps. As shown in Section 4.7, the estimation of the future incoming lighting is performed by simple reprojection. Therefore, the related computational overhead is independent of the scene geometry. In our tests, each record was computed at resolution 64×64. In our system, the reprojection is performed in approximately 0.46 ms. For comparison, the time required to compute the actual incoming lighting at a given point in our 200,000 polygon scene is 4.58 ms. In this case, the overhead due to the reprojection is only 10 percent of the cost of the actual hemisphere sampling. Even though this overhead is not negligible, our estimate enables us reduce the overall rendering time by reusing the records in several frames.

7 Conclusion

In this paper, we presented a novel method for exploiting temporal coherence in the context of irradiance and radiance caching. We proposed an approach for sparse sampling and interpolation of the incoming radiance in the temporal domain. We defined a temporal weighting function and temporal gradients, allowing a simple and accurate temporal interpolation of incoming radiance values. The results show both a significant speedup and an increased quality compared to per-frame computation. Due to our sparse temporal sampling, the incoming radiance values for the entire animation segment can be stored within the main memory. As our method provides a significant quality improvement and is easy to implement, we believe that our approach can be integrated in production renderers for efficient rendering of animated scenes.

Future work includes the design of a more accurate estimation method for extrapolated temporal gradients. Such a method will find use in on-the-fly computation of indirect lighting during interactive sessions. Another improvement would consist in designing an efficient method for faster aging of the records located near newly created records for which important changes have been detected. This would avoid the need for a user-defined maximum validity time while guaranteeing the absence of global illumination ghosts.

References

[1] D.R. Baum, J.R. Wallace, M.F. Cohen, and D.P. Greenberg, "The Back-Buffer Algorithm: An Extension of the Radiosity Method to Dynamic Environments," *The Visual Computer*, vol. 2, no. 5, pp. 298-306, 1986.

[2] X. Pueyo, D. Tost, I. Martín, and B. Garcia, "Radiosity for Dynamic Environments," *J. Visualization and Computer Animation*, vol. 8, no. 4, pp. 221-231, 1997.

[3] G. Besuievsky and X. Pueyo, "Animating Radiosity Environments through the Multi-Frame Lighting Method," *J. Visualization and Computer Graphics*, vol. 12, pp. 93-106, 2001.

[4] C. Damez, K. Dmitriev, and K. Myszkowski, "Global Illumination for Interactive Applications and High-Quality Animations," *Proc. Ann. Conf. European Assoc. Computer Graphics (Eurographics '02)*, pp. 55-77, Sept. 2002.

[5] K. Dmitriev, S. Brabec, K. Myszkowski, and H.-P. Seidel, "Interactive Global Illumination Using Selective Photon Tracing," *Proc. Eurographics Workshop Rendering*, pp. 25-36, 2002.

[6] T. Tawara, K. Myszkowski, and H.-P. Seidel, "Exploiting Temporal Coherence in Final Gathering for Dynamic Scenes," *Proc. Computer Graphics Int'l Conf.*, pp. 110-119, June 2004.

[7] M. Smyk, S.-I. Kinuwaki, R. Durikovic, and K. Myszkowski, "Temporally Coherent Irradiance Caching for High Quality Animation Rendering," *Proc. Ann. Conf. European Assoc. for Computer Graphics (Eurographics '05)*, vol. 24, no. 3, pp. 401-412, 2005.

[8] G.J. Ward, F.M. Rubinstein, and R.D. Clear, "A Ray Tracing Solution for Diffuse Interreflection," *Proc. Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '88)*, pp. 85-92, 1988.

[9] J. Křivánek, P. Gautron, S. Pattanaik, and K. Bouatouch, "Radiance Caching for Efficient Global Illumination Computation," *IEEE Trans. Visualization and Computer Graphics*, vol. 11, no. 5, pp. 550-561, Sept.-Oct. 2005.

[10] P. Dutre, P. Bekaert, and K. Bala, *Advanced Global Illumination*. AK Peters, 2003.

[11] M. Pharr and G. Humphreys, *Physically Based Rendering*. Morgan Kaufmann, 2004.

[12] B. Walter, G. Drettakis, and S. Parker, "Interactive Rendering Using the Render Cache," *Proc. Eurographics Workshop Rendering*, pp. 235-246, 1999.

[13] B. Walter, G. Drettakis, and D.P. Greenberg, "Enhancing and Optimizing the Render Cache," *Proc. Eurographics Workshop Rendering*, pp. 37-42, 2002.

[14] G. Bishop, H. Fuchs, L. McMillan, and E.J.S. Zagier, "Frameless Rendering: Double Buffering Considered Harmful," *Proc. Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '94)*, pp. 175-176, 1994.

[15] A. Dayal, C. Woolley, B. Watson, and D. Luebke, "Adaptive Frameless Rendering," *Proc. Eurographics Workshop Rendering*, pp. 265-276, 2005.

[16] P. Tole, F. Pellacini, B. Walter, and D.P. Greenberg, "Interactive Global Illumination in Dynamic Scenes," *Proc. Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '02)*, pp. 537-546, 2002.

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

[18] T. Tawara, K. Myszkowski, and H.-P. Seidel, "Localizing the Final Gathering for Dynamic Scenes Using the Photon Map," *Proc. Vision, Modeling, and Visualization Conf. (VMV '02)*, 2002.

[19] P. Gautron, J. Křivánek, S. Pattanaik, and K. Bouatouch, "A Novel Hemispherical Basis for Accurate and Efficient Rendering," *Proc. Eurographics Symp. Rendering*, pp. 321-330, 2004.

[20] G.J. Ward and P.S. Heckbert, "Irradiance Gradients," *Proc. Eurographics Workshop Rendering*, pp. 85-98, 1992.

[21] J. Křivánek, P. Gautron, K. Bouatouch, and S. Pattanaik, "Improved Radiance Gradient Computation," *Proc. Spring Conf. Computer Graphics (SCCG '05)*, pp. 149-153, 2005.

[22] E. Tabellion and A. Lamorlette, "An Approximate Global Illumination System for Computer-Generated Films," *Proc. Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '04)*, Aug. 2004.

[23] J. Křivánek, K. Bouatouch, S.N. Pattanaik, and J. Žára, "Making Radiance and Irradiance Caching Practical: Adaptive Caching and Neighbor Clamping," *Proc. Eurographics Symp. Rendering,* 2006.
[24] P. Gautron, J. Křivánek, K. Bouatouch, and S. Pattanaik, "Radiance Cache Splatting: A GPU-Friendly Global Illumination Algorithm," *Proc. Eurographics Symp. Rendering,* June 2005.
[25] C.M. Goral, K.E. Torrance, D.P. Greenberg, and B. Battaile, "Modelling the Interaction of Light Between Diffuse Surfaces," *Proc. Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '84),* pp. 212-222, July 1984.
[26] C. Damez, "Simulation Globale de l'Eclairage Pour des Sequences Animees Prenant en Compte La Coherence Temporelle," PhD dissertation, Univ. Joseph Fourier, 2001.
[27] C. Damez, F.X. Sillion, and N. Holzschuch, "Space-Time Hierarchical Radiosity with Clustering and Higher-Order Wavelets," *Proc. Ann. Conf. European Assoc. for Computer Graphics (Eurographics '01),* pp. 129-141, Sept. 2001.
[28] G. Besuievsky and M. Sbert, "The Multi-Frame Lighting Method: A Monte Carlo-Based Solution for Radiosity in Dynamic Environments," *Proc. Eurographics Workshop Rendering,* pp. 185-194, 1996.
[29] I. Martín, X. Pueyo, and D. Tost, "Frame-to-Frame Coherent Animation with Two-Pass Radiosity," *IEEE Trans. Visualization and Computer Graphics,* vol. 9, no. 1, pp. 70-84, Jan.-Mar. 2003.
[30] G. Drettakis and F.X. Sillion, "Interactive Update of Global Illumination Using a Line-Space Hierarchy," *Proc. Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '97),* vol. 31, no. 3, pp. 57-64, 1997.
[31] X. Granier and G. Drettakis, "A Final Reconstruction Approach for a Unified Global Illumination Algorithm," *ACM Trans. Graphics,* vol. 23, no. 2, pp. 163-189, 2004.

Pascal Gautron received the PhD degree from the Institut de Recherche en Informatique et Systèmes Aléatoires/Institut National de Recherche en Informatique et en Automatique (IRISA/INRIA) Rennes and in collaboration with the University of Central Florida. His research focused on the development of fast GPU-accelerated global illumination computation methods based on the irradiance caching algorithm. He is now a postdoctoral researcher at France Telecom R&D Rennes, and his current research work is toward the rendering of photorealistic virtual agents in real time.

Kadi Bouatouch received the degree in electronics and automatic systems engineering from Ecole Nationale Superiore d'Electricité et Mécanique (ENSEM) in 1974, the PhD degree in 1977, and the higher doctorate degree in computer science in the field of computer graphics in 1989. He is currently a professor at the University of Rennes 1, France, and a researcher at the Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA). He is working on global illumination, lighting simulation for complex environments, parallel radiosity, and augmented reality. He is a member of Eurographics, ACM, and the IEEE. He was a member of the program committees of several conferences and workshops and referee for several computer graphics journals, such as *The Visual Computer*, *IEEE Computer Graphics and Applications*, *IEEE Transactions on Visualization and Computer Graphics*, *IEEE Transactions on Image Processing*, and so forth. He has also acted as a referee for many conferences and workshops.

Sumanta Pattanaik received the PhD degree in computer science from Birla Institute of Technology and Science (BITS), Pilani. He is an associate professor of computer science at the University of Central Florida. His research interests include realistic image synthesis and real-time realistic rendering. He is a member of the IEEE, ACM SIGGRAPH, and Eurographics. He is the graphics category editor of *ACM Computing Reviews*.