Exploiting Temporal Coherence in Global Illumination

T. Tawara, K. Myszkowski, K. Dmitriev, V. Havran, C. Damez, and H.-P. Seidel

MPI Informatik, Saarbrücken, Germany

Abstract

Producing high quality animations featuring rich object appearance and compelling lighting effects is very time consuming using traditional frame-by-frame rendering systems. In this paper we present a number of global illumination and rendering solutions which exploit temporal coherence in lighting distribution for subsequent frames to improve the computation performance and overall animation quality. Our strategy relies on extending into temporal domain well-known global illumination techniques such as density estimation photon tracing, photon mapping, and bi-directional path tracing, which were originally designed to handle static scenes only.

Keywords: Global illumination, temporal coherence, density estimation, irradiance cache, bi-directional path tracing

1 Introduction

Synthesis of images predicting the appearance of the real world has many important engineering applications including product design, architecture, and interior design. One of the major components of such predictive image synthesis is global illumination, which is very costly to compute. The reduction of those costs is an important practical problem in particular for the production of animated sequences because a vast majority of the existing global illumination algorithms were designed for rendering static scenes. In practice this means that when such algorithms are used for a dynamic scene, all computations have to be repeated from scratch even for minor changes in the scene. This leads to redundant computations which could be mostly avoided by taking into account the temporal coherence of global illumination in the sequence of animation frames. Another important problem is the temporal aliasing which is more difficult to combat efficiently if temporal processing of global illumination is not performed. Many small errors in lighting distribution cannot be perceived by the human observer when they are coherent in the temporal domain. However, they may cause unpleasant flickering and shimmering effects when such a coherence is lost.

In this paper we discuss global illumination and rendering algorithms, which were developed by us and designed specifically to exploit temporal coherence in lighting distribution between the subsequent animation frames. For a complete survey of research on off-line and interactive global solutions for dynamic environments refer to a recent paper by Damez et al. [2003].

In Section 2 we recall a density estimation particle tracing algorithm, which was originally designed for static scenes. We extend this algorithm into temporal domain by sharing photon hit points between the subsequent frames and using advanced spatio-temporal filters for lighting reconstruction (Section 2.1). Even more efficient identification and update of invalid photon paths (due to changes in the scene) can be obtained using selective photon tracing (Section 2.2).

In Section 3 we discuss the photon mapping technique and we focus on the problem of efficient rendering using the so-called irradiance cache approach. We extend the irradiance cache into the temporal domain by re-using cache locations (Section 3.1) and selectively updating their values whenever affected by changes in the scene (Section 3.2).

In all solutions introduced so far temporal coherence was exploited in the object space. In Section 4 we present a rendering architecture for computing multiple frames at once by re-using global illumination samples in the image space. For each sample representing a given point in the scene we update its view-dependent components for each frame and add its contribution to pixels identified through the compensation of camera and object motion. We demonstrate that precise and costly global illumination techniques such as bi-directional path tracing become affordable in this rendering architecture.

In Section 5 we conclude this paper we suggest some applications for which our techniques presented in Sections 2–4 are well suited.

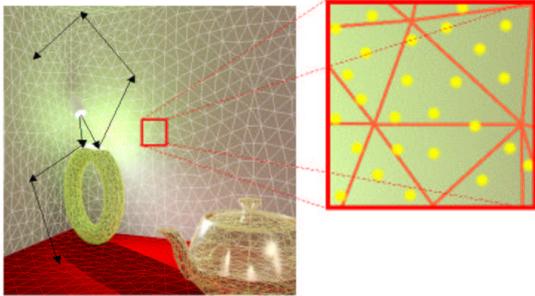# 2 Space-Time Photon Density Estimation

Figure 1: Density estimation photon tracing algorithm: the lighting function is known implicitly as the density of photon hit points for each mesh element in the scene.

In this section we describe an extension of a simple histogram-based photon density estimation algorithm into the temporal domain. The original algorithm [Volevich et al. 2000] for static scenes (refer to Figure 1) is similar to other stochastic solutions in which photons are traced from light sources towards surfaces in the scene. The energy carried by every photon is deposited at the hit point locations on those surfaces [Heckbert 1990; Shirley et al. 1995; Walter 1998]. A simple photon bucketing on a dense triangular mesh is performed, and every photon is discarded immediately after its energy is distributed to the mesh vertices. Efficient object space filtering substantially reduces visible noise. Excessive smoothing of the lighting function can be avoided by adaptively controlling the local filter support, which is based on stochastically-derived estimates of the local illumination error [Volevich et al. 2000; Walter 1998].

In the following sections we describe our extensions of the algorithm by Volevich et al. to handle dynamic environments. The key idea is to re-use photon hit points between the subsequent frames. The main problem here is to avoid re-using invalid photon paths due to changes in the scene. We address this problem using various space-time photon filtering techniques (Section 2.1) and selectively updating those invalid paths (Section 2.2).

2.1 Space-Time Bilateral Filtering

In our space-time density estimation algorithm [Myszkowski et al. 2001] direct lighting is computed from scratch for each frame. To reconstruct indirect illumination photons hitting each mesh element are collected in temporal domain for the previous and subsequent frames until a significant change of illumination due to moving objects is detected or a

sufficient number of photons are collected. Initially, the indirect lighting function is sparsely sampled in space for all frames within a given animation segment. Then, based on the obtained results, a decision is made whether the segment can be expanded/contracted in the temporal domain. Since the validity of samples may depend on the particular region in the scene for which indirect lighting conditions change more rapidly, different segment lengths are chosen locally for each mesh element (used to store photon hits), based on the variations of the lighting function. Energy-based statistical measures of such local variations are used to calculate the number of preceding and following frames for which samples can be safely used for a given region. More samples are generated if the quality of the frames obtained for a given segment length is not sufficient. The perception-based Animation Quality Metric is used to decide upon the stopping condition for the photon tracing depending on the perceptibility of stochastic noise (resulting from collecting too few photons) in the reconstructed illumination.

In [Myszkowski et al. 2001] we emphasize on the perceptual aspects of steering the global illumination computation and the processing of photons was simply limited to sharing those photons between frames. In the follow-up work [Weber et al. 2004] we focus on spatio-temporal photon filtering. It is aimed at improving the quality of reconstructed surface illumination by reducing the amount of spatial and temporal blur, while keeping the stochastic noise below the perceivable level.

The temporal blur is a result of collecting photons in temporal domain in those scene regions in which lighting changes quickly. In such regions some photon paths computed for previous frames can be invalid for the currently processed frame due to dynamic changes in the scene. To reduce the temporal blur such invalid photons should not be considered. If too few photons are collected in the temporal domain to satisfy noise error criteria, the remaining photons must be collected in the spatial domain. This in turn may lead to spatial blur in the reconstructed illumination. To reduce this effect the missing photons must be collected only in the neighboring scene regions exhibiting similar illumination. Collecting photons across the edges of high illumination contrast inherently leads to lighting patterns blurring. Similarly, collecting photons in the temporal domain for abruptly changing lighting leads to temporal blur. Clearly, a photon density estimation method capable of reducing stochastic noise while detecting such abrupt spatio-temporal changes of lighting over meshed surfaces is needed.

We propose to extend traditional photon density estimation methods by using spatio-temporal bilateral filtering to reduce stochastic noise, while preventing excessive

blurring in reconstructed lighting. For lighting estimation of each mesh element, photons collected for neighboring mesh elements in space-time domain are considered. However the influence of elements with significantly different illumination is suppressed by bilateral filtering. This means that extremely noisy illumination estimates (outliers) as well as estimates resulting from abrupt changes of illumination distribution both in space and time domains are filtered out. As a result better space-time resolution of lighting patterns is obtained while at the same time spatio-temporal noise is significantly reduced. Also, the computation efficiency is significantly improved by reducing the number of photons required to achieve high-quality animations when compared to traditional frame-by-frame rendering approaches.

In the first case study experiment we evaluate the quality of a reconstructed caustic for a scene shown in Figure 2d. Figure 2a shows an unfiltered caustic that results from simple bucketing of photon hit points for each mesh element. Figure 2b shows the result of Gaussian filtering with adaptive spatial support σ_S. The result of bilateral filtering shown in Figure 2c was obtained for the same setting of σ_S. As it can be seen bilateral filtering produces the best results. The noise is significantly reduced with respect to the non-filtered image in Figure 2a. Note also that the excessive blur visible in Figure 2b is avoided. All images shown in Figure 2 were obtained using 557,000 photons.

In another case study experiment we evaluate the performance of bilateral filtering for an animated sequence. Figure 3b shows a sample animation frame obtained using bilateral filtering for 30 temporally processed frames. This scene contains many small mesh elements that collect a small number of photons and therefore exhibit significant noise in the lighting function. This leads to strong fluctuations of illumination between neighboring triangles, which bilateral filtering interprets as important lighting details and tends to preserve. To overcome this problem we relax the influence parameter of the bilateral filter (which decides upon the suppression of outliers) for those mesh elements that collect a very small number of photons. In practice, this means that for such elements filtering characteristics adaptively evolves towards Gaussian filtering when the number of photons decreases. Figure 3a shows the corresponding results for the same number of photons without any filtering. With traditional frame-by-frame computation and filtering performed only in the spatial domain, in order to obtain a quality of frames similar to the one of Figure 3b, at least ten times more photons must be traced. However, temporal flickering is very difficult to eliminate when the processing of photons is limited only to the spatial domain.

In our space-time density estimation solutions [Myszkowski et al. 2001; Weber et al. 2004] dis-

cussed in this section we try to limit the use of invalid photons due to changes in the scene by the adaptive choice of filter parameters. However, this does not guarantee conservative results. To improve on this, in the following section, we discuss an algorithm [Dmitriev et al. 2002] in which photon hit points are also re-used in the temporal domain but invalid photon paths are detected and updated with a very high probability.

2.2   Selective Photon Tracing

In [Dmitriev et al. 2002] we have presented a selective photon tracing (SPT) method for global illumination computation which is specifically designed for interactive applications, such as product design, architecture and interior design, and illumination engineering. The method is embedded into the framework of Quasi-Monte Carlo photon tracing and density estimation techniques. Temporal coherence of illumination is exploited by tracing photons selectively to the scene regions that require indirect illumination update. Such regions are identified with a high probability by a small number of so-called *pilot photons*. Based on the pilot photons which require updating, the remaining photons with similar paths in the scene can be found immediately. This is possible due to the periodicity property inherent to the multi-dimensional Halton sequence [Halton 1960], which is used to generate photons.

The SPT algorithm uses graphics hardware to compute direct lighting with shadows using the shadow volume algorithm. Using the functionality of modern graphics hardware we can process up to 4 light sources with goniometric diagrams during a single rendering pass. The indirect lighting is computed asynchronously using a quasi-random photon tracing (similar to Keller [1996]) and density estimation techniques (similar to Volevich et al. [2000]; refer to a short summary in Section 2 and to Figure 1). The indirect lighting is reconstructed at vertices of a precomputed mesh and can be readily displayed using graphics hardware for any camera position. The most unique feature of the SPT algorithm is exploiting the temporal coherence of illumination by tracing photons selectively into the scene regions that require illumination update.

In the SPT algorithm, pseudo-random sequences, typically used in photon shooting, are replaced by the quasi-random Halton sequence [Halton 1960]. This proves to be advantageous. Firstly, as shown by Keller [1996], quasi-random walk algorithms converge faster than classical random walk ones. Secondly, a periodicity property of the Halton sequence [Niederreiter 1992] provides a straightforward way of updating indirect illumination as the scene changes. Let us briefly discuss this periodic-

(a) No filtering

(b) Gaussian filtering

(c) Bilateral filtering
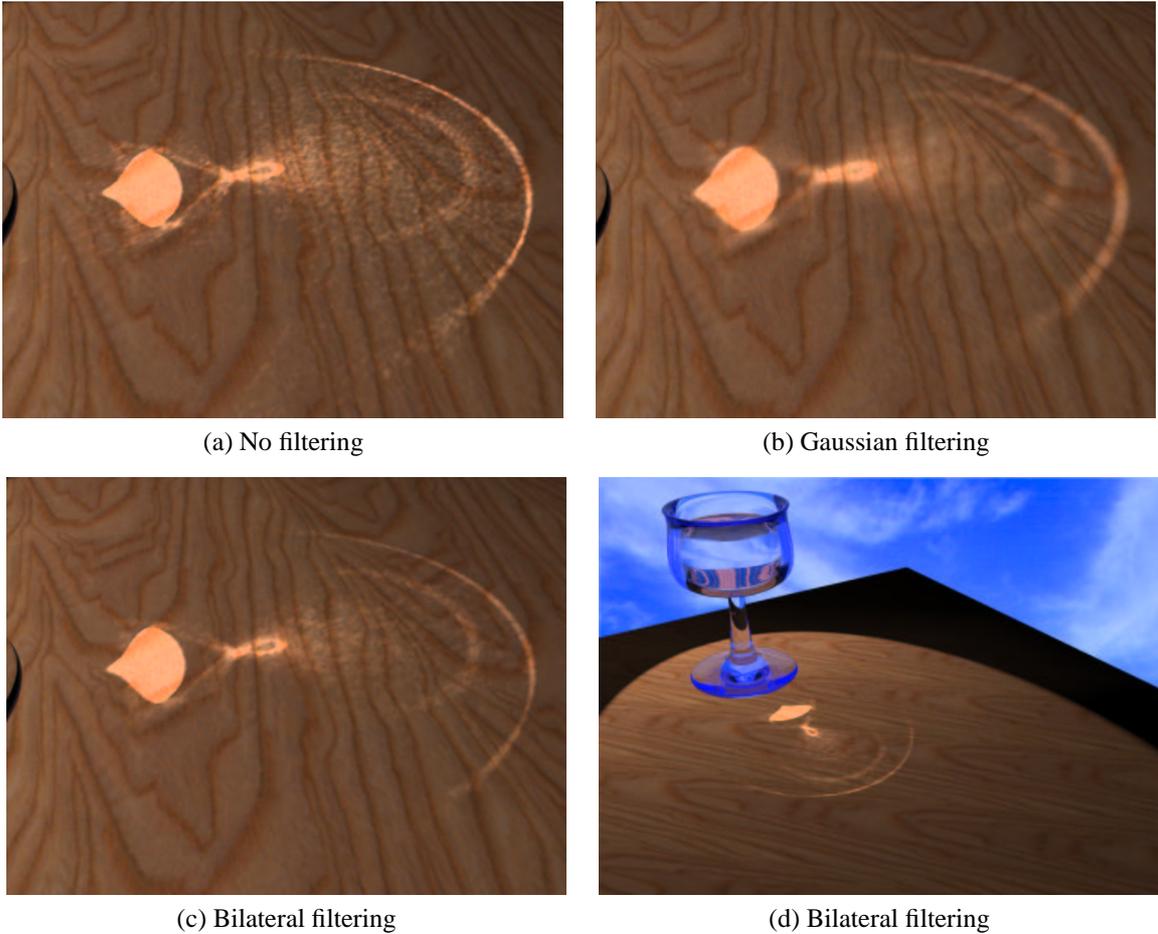

(d) Bilateral filtering

Figure 2: Caustic reconstruction for the GLASS scene. Notice that the shape of caustics is affected by the polygonal structure of the glass.

ity property, which is fundamental for efficient searching of invalid photon paths in dynamic scenes.

The Halton sequence generation is based on the radical inverse function [Halton 1960] h, applied to an integer i. The sequence of multidimensional points it generates can be organized in N_g groups for which the distance between points is bounded. If h_{ij} is the j-th coordinate of the Halton point with index i, it can be shown that [Dmitriev et al. 2002]:

$$|h_{ij} - h_{(i+mN_g)j}| < \frac{1}{b_j^k}, \text{ if } N_g = lb_j^k \qquad (1)$$

where b_j is the base prime number used to compute the j-th coordinate of Halton points, k, l, and m are integers such that $k \geq 0$ and $l > 0$. For instance, setting $N_g = b_0^{k_0} b_1^{k_1} b_2^{k_2} b_3^{k_3}$ yields points in which the first four coordinates closely match. The closeness of this match is governed by the corresponding powers k_0, k_1, k_2, k_3 (the larger power values are selected the closer match is obtained). If the scene surfaces and BSDFs are reason-

ably smooth, quasi-random points with similar coordinates produce photons with similar paths, which is depicted in Figure 4.
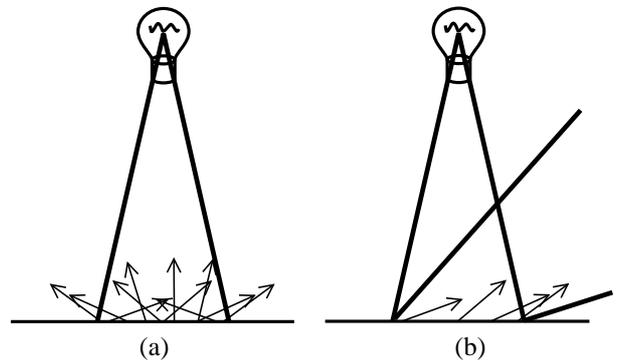

Figure 4: Structure of photon paths, corresponding to quasi-random points with similar coordinates (a) similarity of first two coordinates. (b) similarity of first four coordinates.

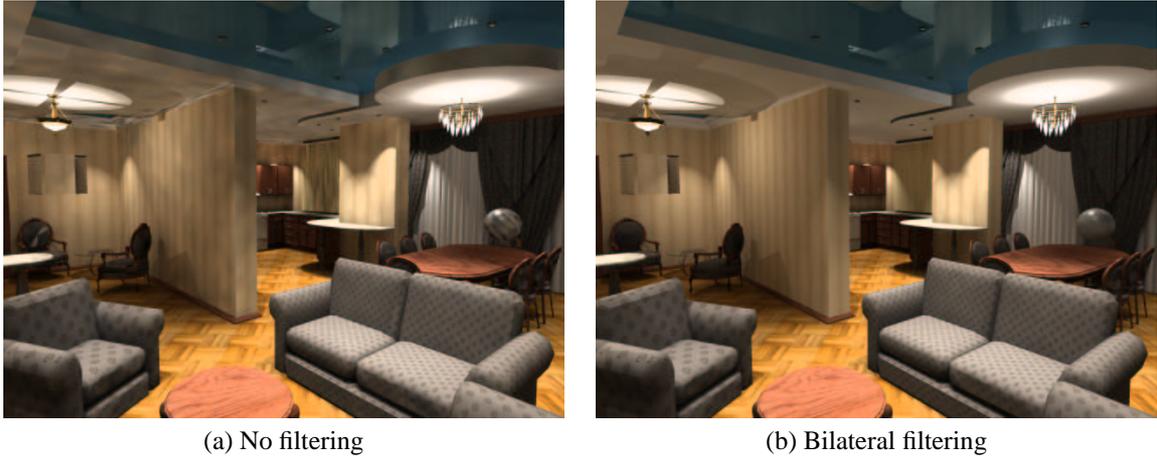By selecting quasi-random points with such an interval

(a) No filtering

(b) Bilateral filtering

Figure 3: Space-time lighting reconstruction for the SALON scene using 622,200 photons per frame.
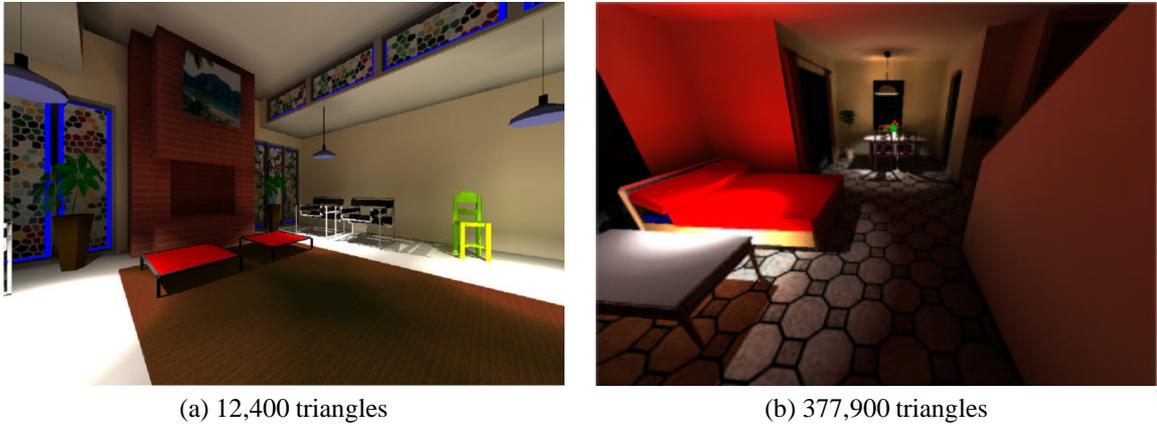

(a) 12,400 triangles

(b) 377,900 triangles

Figure 5: Example frames with global illumination effects obtained using Selective Photon Tracing. On a 1.7 GHz Dual P4 Xeon computer with an NVIDIA GeForce3 64 MB video card the frame rates 8 fps and 1.1 fps have been achieved for (a) and (b), respectively. In order to remove all visible artifacts in lighting distribution resulting from object changes in the scene, 4–8 seconds of the selective photon update computation are usually required.

N_g, that the similarity of coordinates $j = 0$ and $j = 1$ is assured, photons are emitted coherently inside a pyramid with its apex at light source position. However, those photons do not reflect coherently (Figure 4a). By additionally increasing N_g to obtain a similarity of coordinates $j = 2$ and $j = 3$, photons are selected so that they reflect coherently inside a more complex bounding shape, as shown in Figure 4b.

The lighting computation algorithm considers a pool of photons of fixed size Z for the whole interactive session. The photons are traced during the initialization stage. Then the paths which become invalid due to changes in the scene are selectively updated. This is performed by tracing photons for the previous scene configuration with negative energy and tracing photons for the new scene configuration with positive energy. To detect the invalid photon paths, pilot photons, which constitute 5–10% of Z, are emitted in the scene. For each pilot photon i which

hits a dynamic object and therefore requires updating, the remaining photons in the pool Z with similar paths in the scene can be found immediately by adding the offsets mN_g to i, where m are integers such that $i + mN_g < Z$ (refer to Equation 1).

The photon update computations are performed iteratively. Each iteration consists of reshooting one coherent photon group. The order in which the photon groups are updated is decided using an inexpensive energy- and perception-based criterion whose goal is to minimize the perceivability of outdated illumination.

The frame rate in an interactive session using the SPT algorithm is mostly determined by the OpenGL performance. For rendering with shadows from multiple light sources with goniometric diagrams, frame rates ranging from 1.1 to 8 fps. are reported (refer to Figure 5). Indirect illumination is updated incrementally and the result of each update is immediately delivered to the user. Most

lighting artifacts created by outdated illumination are removed in the first 4–8 seconds after the scene change. If photons are updated in a random order at least 2–4 times longer computations are needed to obtain images of similar quality. Better performances can be expected for more complex scenes, or when user modifications have more localized consequences. The unique feature of SPT is that while the temporal coherence in lighting computations is strongly exploited, no additional data structures storing photon paths are required.

In the case study examples (refer to Figure 5) we considered the coherence of photons only up to their second bounce. This proved to be sufficient to efficiently update the illumination for the studied scenes. When such a coherence for a higher number of bounces is required, the number of coherent photon groups N_g increases exponentially. This results in a proportional increase of the photon pool size Z. Interactive update of such an increased pool of photons might not be feasible on a single PC-class computer. The major problem with our algorithm is lack of adaptability of the mesh used to reconstruct the indirect illumination in the scene. Also, only point light sources are explicitly handled in our hardware-supported algorithm for the direct illumination computation.

Selective Photon Tracing can be used in the context of offline global illumination computation as well. In particular, this technique seems to be attractive in all those applications which require local reinforcement of computations based on some importance criteria. An example of such an application is the efficient rendering of high quality caustics, which usually requires a huge number of photons. After identifying some paths of caustic photons, more coherent particles can easily be generated using this approach. The drawback of many existing photon based methods is that too many particles are sent into well illuminated scene regions with a simple illumination distribution, and too few photons reach remote scene regions. This deficiency could easily be corrected using the SPT approach, by skipping the tracing of redundant photons and properly scaling the energy of the remaining photons.

3 High-Quality Rendering Using an Irradiance Cache

In the rendering of production quality animation, global illumination computations are usually performed using two-pass methods. In the first (preprocessing) pass, the lighting distribution over scene surfaces is sparsely computed using radiosity [Lischinski et al. 1993; Smits 1994; Christensen et al. 1997], or photon mapping [Jensen 2001; Christensen 2002] methods. In the second (ren-

dering) pass a more exact global illumination computation is performed on a per pixel basis using the results obtained in the first pass. The algorithm of choice for this final rendering of high quality images is the so-called *final gathering* [Reichert 1992; Lischinski et al. 1993; Smits 1994; Christensen et al. 1997]. Usually the direct lighting is computed for a surface region covered by a given pixel, and the indirect lighting is obtained through the integration of incoming radiances, which is very costly. Those costs can be reduced by using the *irradiance cache* data structures [Ward et al. 1988; Ward and Heckbert 1992] to store irradiance samples sparsely in the object space. Within this method irradiance samples are lazily computed and sparsely cached in object space for a given camera position (a view-dependent process). The indirect illumination is interpolated for each pixel based on those cached irradiance values, which is significantly faster than the final gathering computation for each pixel. The irradiance cache technique efficiently removes shading artifacts which are very difficult to avoid if the indirect lighting is directly reconstructed based on a radiosity mesh or a photon map. However, the price to be paid for this high quality lighting reconstruction is long computation times, which are mostly caused by the irradiance integration that is performed for each cache location in the scene.

In the following sections we extend the concept of irradiance cache for dynamic environments to improve the rendering performance and reduce the temporal aliasing [Tawara et al. 2002; Tawara et al. 2004]. In our approach we use a two pass photon mapping algorithm [Jensen 2001], which we extend to make the global illumination computation more efficient for such dynamic environments.

3.1 Static and Dynamic Irradiance Cache

We focus on exploiting the temporal coherence of photons to speedup the costly irradiance cache computation and to improve the quality of indirect lighting reconstruction. We introduce the notion of *static irradiance cache*, which is computed once for an animation segment. For the static irradiance cache computation we remove all dynamic objects (i.e., objects changing their position, shape, or light reflectance properties as a function of time) from the scene and we trace the so-called *static photons*.

The illumination component reconstructed from the static irradiance cache is perfectly coherent in the temporal domain and results in flicker-free animations. However, the dynamic illumination component caused by animated objects must also be considered. For this purpose

the *dynamic photons* which interact with dynamic objects are computed for each frame and are stored in a separate photon map. The map may store photons with negative energy [Jensen 2001], which are needed to compensate for occlusions of the static parts of the scene by dynamic objects (for example, in the regions of indirect shadows cast by dynamic objects).

Figures 6 illustrate the illumination reconstruction using our technique. Figure 6a shows the final animation frame whose lighting was composed from the direct illumination and specular reflection (Figure 6b) and the diffuse indirect illumination (Figure 6c). The dynamic component of the indirect lighting is reconstructed at two levels of accuracy depending on the influence of dynamic objects on local scene illumination. In regions with higher influence, the *dynamic irradiance cache* is recomputed, which leads to a better accuracy of reconstructed dynamic lighting (refer to Figure 6e). In the remaining scene regions as shown in Figure 6f the illumination stored in the static irradiance cache (shown in Figure 6d) is corrected by adding its dynamic component reconstructed from the dynamic photon map using density estimation. A direct visualization of the dynamic illumination component is not shown because its values are rather small for the most parts of the scene and are negative in the regions occluded by dynamic objects. We blend lighting reconstructed using those two different methods (Figures 6e and 6f) to assure smooth transition of the resulting lighting.

As a result significant speedup of the computation was achieved by localizing in the scene space costly recomputation of the irradiance cache. Also, the temporal aliasing was reduced by introducing the concept of static irradiance cache which can be reused across many subsequent frames during which the scene lighting conditions do not change significantly.

In the following section we propose an algorithm in which the irradiance cache locations are not only re-used, but also the cache values are selectively updated following changes in dynamic environments.

3.2 Selective Irradiance Cache Update

The key idea of our selective cache update algorithm is inspired by the observation that many gathering rays are shot for each frame and their sampled values are similar between the neighboring frames. For the efficiency and correctness of the final gathering, the integration of irradiance is performed over a stratified domain. In our algorithm, the incident radiance, the distance to the nearest intersection point and the refreshing probability for each stratified sampling direction at the location of an irradiance cache, are stored in a hard disk during the gath-

ering stage. To select the updating directions for each cache, the cumulative distribution function (*CDF*) of the refreshing probability is built and a modified random permutation algorithm is used. The directions selected are updated by recasting the gathering rays. Although any arbitrary probabilistic distribution could be used, in this work we update the CDF based on the sample age (i.e. the number of frames since the sample was computed)

Figure 7 depicts the values of incident radiance samples over the hemisphere for a selected cache location. The samples are captured in the middle of an animation sequence in order to check whether errors in their value do not accumulate as a function of time. As can be seen the directional distribution of samples is very similar for the frame-by-frame computation and our method.
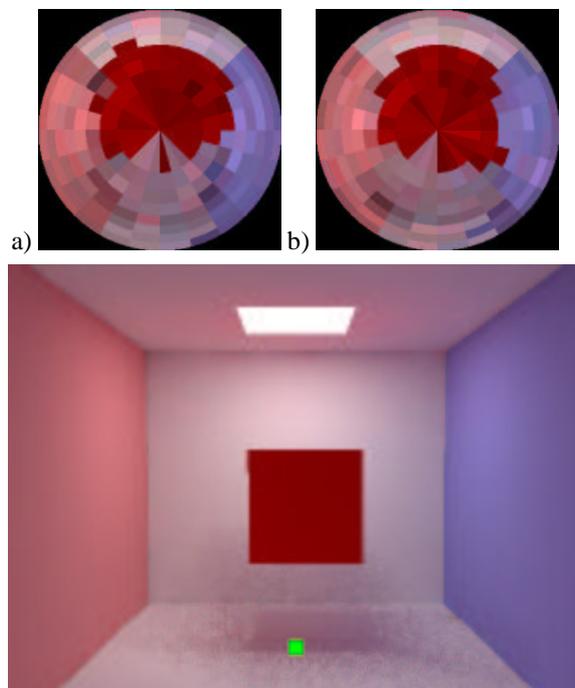

Figure 7: Distribution of incoming radiance samples over the hemisphere for a selected cache location on the floor (shown as the green dot in the bottom image): a) the frame-by-frame computation and b) our method (10% of samples is refreshed for each frame according to the aging criterion).

Figure 8 shows an example of a complex scene in which both the motion of light (sun position) and objects (rotating fan) are simultaneously considered. In this test scene indirect lighting changes quickly and those changes affect a large portion of the scene, which is a difficult case for our algorithm. The average rendering speedup for our selective cache update technique amounts to five times per frame with respect to the frame-by-frame irradiance cache approach. We expect that in most practical cases the indirect lighting changes are more moderate, and the
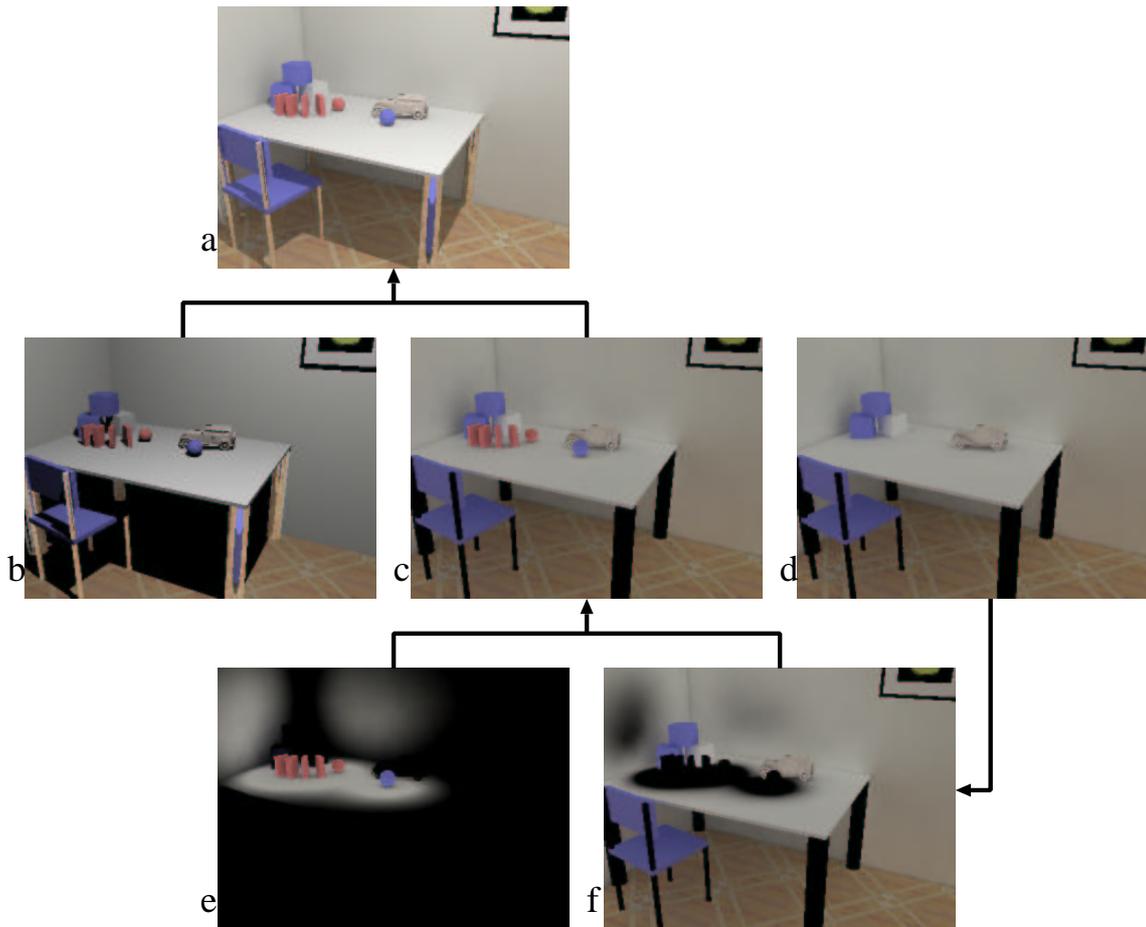
Figure 6: Processing flow in the computation of global illumination for an animation frame: (a) the final frame, (b) direct lighting, (c) indirect lighting, (d) static indirect lighting computed using the static irradiance cache, (e) dynamic indirect lighting computed using the dynamic irradiance cache, and (f) the dynamic indirect lighting computed through the photon density estimation and summed with the static indirect lighting which is shown in (d).

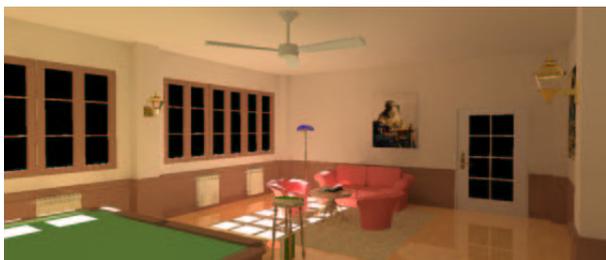performance of our algorithm will be even better.

Figure 8: Selected frame for the ROOM animation sequence.

4 Spatio-Temporal Bi-Directional Path Tracing

In the previous sections we discussed density estimation algorithms in which spatio-temporal processing was performed in the object space. In this section we consider a view-dependent algorithm called *bi-directional path tracing* (BPT) [Lafortune 1996; Veach 1997] which we extend to handle dynamic environments [Havran et al. 2003]. In this algorithm, the bookkeeping of global illumination samples is organized in the image space. We consider the BPT algorithm within a more general rendering framework for computing multiple frames at once by exploiting the coherence between image samples (pixels) in the temporal domain. For each sample representing a given point in the scene we update its view-dependent components for each frame and we add its contribution to pixels identified through the compensation of camera and object motion.

In this section we describe in more details two major challenges that we face in our framework: The visibility and global illumination computation in dynamic environments. Also, we discuss some standard rendering tasks such as shading and motion blur.

The *visibility* computation in our rendering framework is based on a *multi-frame visibility data structure* (MFVDS). The MFVDS provides for a given ray the visibility information for all frames considered at once. A *kd*-tree data structure is used to implement the MFVDS. Static objects are stored in a global *kd*-tree. Dynamic objects are instantiated for every frame. The instantiated objects are processed using a hierarchical clustering algorithm to separate them in space. For each cluster of instantiated objects a separate *kd*-tree is constructed and is inserted in a global *kd*-tree, which is refined in spatial regions where dynamic *kd*-trees have been inserted. During processing of visibility queries the performance of the MFVDS is improved by intensive caching of intermediate results.

A single visibility query using our data structures provides a ray intersection information for a given frame and marks those frames for which this information becomes invalid due to dynamic changes in the scene. This allows us to avoid redundant computation for each frame if the ray does not hit any instantiated object. In regions not populated by animated objects the same ray traversal cost is achieved as for completely static scenes.

The global illumination computation in our framework is based on the BPT algorithm [Lafortune 1996; Veach 1997] and uses the MFVDS to query visibility for all considered frames at once. Each bi-directional estimate of a given pixel color is reused for several frames before and after the one it was originally computed for. To reuse these estimates, the BRDF values at the first hit point of the eye path needs to be recomputed to take into account the new viewpoint. The corresponding estimates are then added to the pixel through which this hit point can be seen for the considered frame. Since it involves only the evaluation of direct visibility from the viewpoint and a few BRDF recomputations, reusing a sample is much faster than recomputing a new one from scratch. Reusing samples for several frames also makes the noise inherent to stochastic methods fixed in object space, which enhances the quality of the resulting animations.

Shaders add rich appearance of objects and can be efficiently computed in our animation framework. We split our shading functions into view-independent and view-dependent components, where the former is computed only once for a given sample point and the latter is recomputed for each frame. It is worth noting that in our BPT technique we need to recompute the view-dependent component only for sample points that are hit

by primary rays, while for the remaining path segments shading results are just reused.

Motion blur is an important visual effect in high quality animation rendering, which is extremely easy to obtain in our framework. A fragment of the motion-compensation trajectory traversed by a given BPT sample within the camera shutter opening time is computed and projected to each frame. The sample radiance contributes to each pixel traversed by the projected trajectory roughly in proportion to the length of traversed path. In practice, we use the 2DDA algorithm for piecewise-linear approximation of this trajectory, and linear interpolation of sample radiance between a pair of frames, which leads to very good visual results (refer to Figure 10).

We use *disc caching*, instead of storing in memory all frames, which makes the resolution of our animations limited only by the disc capacity. Since the computation for subsequent frames is localized in coherent regions in the image space located along the motion-compensation trajectory, we achieve a pretty high cache-hit ratio. As a consequence, disk accesses only increase the total computation time of our method by about 10% on average.

The main advantage of our framework is a significant speedup of animation rendering, which is usually over one order of magnitude in respect to traditional frame-by-frame rendering, while the obtained quality is always much higher due to a significant reduction of flickering. Many standard tasks in rendering such as shading, texturing, and motion-blur can be efficiently performed in our rendering architecture.
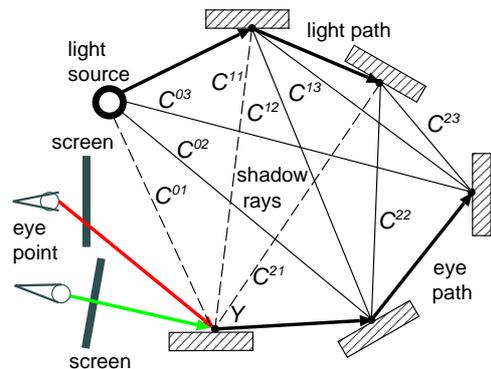

Figure 9: Bidirectional estimations for a given pixel can be used for several camera positions. Only direct visibility and the BRDF corresponding to connections C_{01}, C_{11}, and C_{21} need to be recomputed.

Figure 10: A sample animation frame computed using space-time bi-directional path tracing featuring also motion blur.

5 Conclusions

We presented a number of global illumination and rendering solutions which exploit temporal coherence between subsequent animation frames. Our solutions led to significant improvements of the computation performance and animation quality (through the suppression of temporal aliasing). A practical question arises: which presented technique should be chosen for a given application?

The spatio-temporal photon density estimation technique (Section 2.1) seems to be the most suitable for practical animation systems, where the rendering speed is is the key factor even at the expense of lower accuracy in the lighting computation. This method does not require storing photon hit positions between the subsequent frames. Photon counters for each mesh element are sufficient, which reduces significantly the memory requirements. We implemented this technique as a plug-in for the 3D StudioMax system and our experiences with the exploitation of this technique in the animation production are very good.

The selective photon tracing (Section 2.2) is suitable in particular for interactive rendering applications. Also, it leads to more accurate lighting computation than the previous technique because, in practice, invalid photons do not contribute to the simulation result. However, this method requires photon storage.

The rendering techniques using irradiance cache (Section 3) are suitable for all applications in which the quality requirements are very high. The photon mapping technique based on static and dynamic photons (Section 3.1) is suitable in particular for those applications in which scene changes do not affect significantly lighting distribution (a vast majority of photons can be classi-

fied as static). Selective update of irradiance cache (Section 3.2) is suitable for any two-pass global illumination solution including radiosity, density estimation, and photon mapping and leads to significant reduction of the computation cost.

The rendering framework proposed in Section 4 is suitable for any view-dependent global illumination algorithm and we demonstrated its efficiency for bi-directional path tracing. The framework allows for exploiting temporal coherence in visibility, lighting and shading computations, and motion blur. We hope that in future a similar framework can be used in the professional production of high-quality animations.

References

CHRISTENSEN, P., LISCHINSKI, D., STOLLNITZ, E., AND SALESIN, D. 1997. Clustering for glossy global illumination. *ACM Transactions on Graphics 16*, 1, 3–33.

CHRISTENSEN, P. 2002. Photon Mapping Tricks. In *Siggraph 2002, Course Notes No. 43*, 93–121.

DAMEZ, C., DMITRIEV, K., AND MYSZKOWSKI, K. 2003. State of the art in global illumination for interactive applications and high-quality animations. *Computer Graphics Forum 22*, 1, 55–77.

DMITRIEV, K., BRABEC, S., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2002. Interactive Global Illumination Using Selective Photon Tracing. In *Proceedings of the 13th Eurographics Workshop on Rendering*, 25–36.

HALTON, J. 1960. On the Efficiency of Certain Quasi-random Sequences of Points in Evaluating Multi-Dimensional Integrals. *Numerische Mathematik*, 2, 84–90.

HAVRAN, V., DAMEZ, C., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2003. An efficient spatio-temporal architecture for animation rendering. In *Proceedings of Eurographics Symposium on Rendering 2003*, ACM, 106–117.

HECKBERT, P. 1990. Adaptive Radiosity Textures for Bidirectional Ray Tracing. In *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, 145–154.

JENSEN, H. 2001. *Realistic Image Synthesis Using Photon Mapping*. AK, Peters.

KELLER, A. 1996. Quasi-Monte Carlo Radiosity. In *Proceedings of the 7th Eurographics Workshop on Rendering*, 101–110.

LAFORTUNE, E. 1996. *Mathematical Models and Monte Carlo Algorithms*. PhD thesis, Katholieke Universiteit Leuven.

LISCHINSKI, D., TAMPIERI, F., AND GREENBERG, D. 1993. Combining Hierarchical Radiosity and Discontinuity Meshing. In *Computer Graphics (ACM SIGGRAPH '93 Proceedings)*, 199–208.

MYSZKOWSKI, K., TAWARA, T., AKAMINE, H., AND SEIDEL, H.-P. 2001. Perception-Guided Global Illumination Solution for Animation Rendering. In *Proceedings of ACM SIGGRAPH 2001*, 221–230.

NIEDERREITER, H. 1992. *Random Number Generation and Quasi-Monte Carlo Methods*. Chapter 4, SIAM, Pennsylvania.

REICHERT, M. 1992. *A Two-Pass Radiosity Method to Transmitting and Specularly Reflecting Surfaces*. M.Sc. thesis, Cornell University.

SHIRLEY, P., WADE, B., HUBBARD, P., ZARESKI, D., WALTER, B., AND GREENBERG, D. 1995. Global Illumination via Density Estimation. In *Proceedings of the 6th Eurographics Workshop on Rendering*, 219–230.

SMITS, B. 1994. *Efficient Hierarchical Radiosity in Complex Environments*. Ph.D. thesis, Cornell University.

TAWARA, T., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2002. Localizing the final gathering for dynamic scenes using the photon map. In *Vision Modeling and Visualization 2002*, 69–76.

TAWARA, T., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2004. Exploiting temporal coherence in final gathering for dynamic scenes. In *Computer Graphics International (CGI 2004)*, IEEE Computer Society.

VEACH, E. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University.

VOLEVICH, V., MYSZKOWSKI, K., KHODULEV, A., AND KOPYLOV, E. 2000. Using the Visible Differences Predictor to Improve Performance of Progressive Global Illumination Computations. *ACM Transactions on Graphics 19*, 2, 122–161.

WALTER, B. 1998. *Density Estimation Techniques for Global Illumination*. Ph.D. thesis, Cornell University.

WARD, G., AND HECKBERT, P. 1992. Irradiance Gradients. In *Proceedings of the 3rd Eurographics Workshop on Rendering*, 85–98.

WARD, G., RUBINSTEIN, F., AND CLEAR, R. 1988. A Ray Tracing Solution for Diffuse Interreflection. In *Computer Graphics (ACM SIGGRAPH '88 Proceedings)*, 85–92.

WEBER, M., MILCH, M., MYSZKOWSKI, K., DMITRIEV, K., ROKITA, P., AND SEIDEL, H.-P. 2004. Spatio-temporal photon density estimation using bilateral filtering. In *Computer Graphics International (CGI 2004)*, IEEE Computer Society.