Thirteenth Eurographics Workshop on Rendering (2002)
P. Debevec and S. Gibson (Editors)

Time Dependent Photon Mapping

Mike Cammarano and Henrik Wann Jensen

Department of Computer Science, Stanford University

Abstract

The photon map technique for global illumination does not specifically address animated scenes. In particular, prior work has not considered the problem of temporal sampling (motion blur) while using the photon map. In this paper we examine several approaches for simulating motion blur with the photon map. In particular we show that a distribution of photons in time combined with the standard photon map radiance estimate is incorrect, and we introduce a simple generalization that correctly handles photons distributed in both time and space. Our results demonstrate that this time dependent photon map extension allows fast and correct estimates of motion-blurred illumination including motion-blurred caustics.

1. Introduction

Motion blur is generally understood to be important in image synthesis for high quality animated scenes. Motion blur arises in traditional photography and film-making because real cameras require finite, nonzero exposure times to acquire an image. The resulting images represent the integral of the radiance incident on the image plane over the duration of the exposure. Thus computer graphics systems striving for photorealism — the production of images that appear photographic — must simulate this behavior.

While simulating cameras is one reason to model motion blur, even stronger motivation comes from the need to conform to the limitations of common display devices. Most high quality computer animations are ultimately intended for viewing at 24 or 30 frames per second. However, the human visual system is reported to be sensitive to temporal frequencies up to 60Hz [15]. As a result, film and video frame rates are insufficient when each frame represents an instantaneous sample of the temporal domain. In practice, early computer graphics pioneers found that the artifacts introduced by low rates of temporal sampling were unacceptable. However, by incorporating motion blur, each frame represents the image radiance integrated over time, rather than an instantaneous sample. This filtering eliminates the most perceptually objectionable artifacts like temporal strobing, resulting in the illusion of fluid motion even at the limited frame rate of typical display media. The simulation of motion blur is essential to image quality in fast-moving animated scenes.

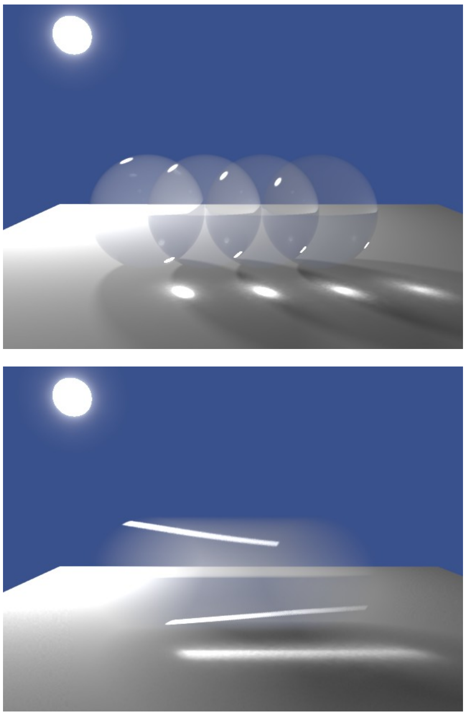


Figure 1: *Glass ball moving above a ground plane. The top image shows the strobing effect of a sequence of still frames, while the bottom image shows the smooth motion blur, including a motion-blurred caustic, obtained by integrating over the exposure interval.*

Most of the research into motion blur can be classified as one of four basic approaches [11]:

1. Analytic methods [5]
2. Temporal supersampling [5] [3]
3. Postprocess blur [13]
4. Geometric substitution [2]

1.1. Temporal supersampling

All four of the approaches characterized above have remained active research areas, and fostered many interesting algorithms. However, only supersampling methods have shown the generality and robustness to accurately handle complicated dynamic scenes. Postprocessing methods have all relied on approximations of scene motion, while analytic and geometric substitution methods have been limited in the geometry and motion they can handle. For robust high-quality rendering, and in particular for simulating global illumination, temporal supersampling is the only viable method.

We can further classify temporal supersampling techniques as belonging to one of two general categories. Accumulation buffer methods render a sequence of complete "in-between" frames and average them together [5] [4], while distribution ray tracing techniques evaluate multiple time samples at each image pixel [3].

The simulation of global illumination and motion blur together has received relatively little attention. Cook et al. [3] demonstrated that distribution ray tracing could be used to simulate global illumination effects including motion blur. Besuievsky and Pueyo [1] simulated motion blur with the radiosity algorithm by computing multiple frames at different times and averaging those using an accumulation buffer. Myszkowski et al. [12] presented an efficient global illumination method for animated scenes in which the result of a photon tracing pass is averaged over several frames. However, their method still computes a solution for a static scene for a given frame, and they do not consider motion blur within individual frames. The only methods that currently are able to simultaneously simulate both motion blur and global illumination are based on Monte Carlo ray tracing. In Monte Carlo ray tracing it is straightforward to simulate motion blur by distributing paths randomly in time — for example, Lafortune [10] demonstrated this using bidirectional path tracing.

Unfortunately, the general Monte Carlo ray tracing algorithms in which motion blur can be trivially simulated require substantial computation to reduce the noise to an acceptable level. For static scenes, there are several algorithms for improving the speed of Monte Carlo ray tracing by caching information about the illumination in the scene. Methods such as irradiance caching [16] and photon mapping [7] are examples of such caching techniques where illumination values are stored for points on the surfaces in the scene (or inside a scattering medium [8]). However, these caching techniques do not work when the objects move within the shutter time used for the frame. If different photon or ray-paths have different times, then the cached illumination values will be distributed in space along the path of moving objects. Notably, in the case of photon mapping this invalidates the assumptions that the photons are located on the surfaces of the objects.

In this paper we extend the photon mapping method to handle motion-blurred global illumination in scenes with moving objects. We derive a time dependent radiance estimate that correctly accounts for photons distributed in both space and time. Our results indicate that this radiance estimate is superior to alternative techniques such as the accumulation buffer, and that it correctly renders motion-blurred illumination effects that the standard photon map radiance estimate cannot handle.

2. Time Dependent Global illumination

In global illumination and realistic image synthesis we are concerned with estimating the radiance through pixels in the image plane. This can be expressed as

$$L_p = \int_{t_s} \int_A L(x', \vec{\omega}, t) s(x', \vec{\omega}, t) g(x')\, dA(x')\, dt, \qquad (1)$$

where L_p is the radiance through pixel p, t_s is the total shutter or frame time, A is the area of the pixel, $g(x')$ is a filter function, the shutter $s(x', \vec{\omega}, t)$ specifies the exposure time for each pixel, and $L(x', \vec{\omega}, t)$ is the radiance through the location x' on the image plane in the direction of the observer at time t. In scenes with only static objects the integral over time can be ignored, and $L(x', \vec{\omega}, t)$ becomes $L(x', \vec{\omega})$. Otherwise, radiance must be integrated in time as well. In the case of Monte Carlo ray tracing this is typically done by tracing rays at different times through the pixel and averaging the radiance returned by each ray. At the first object intersected by a ray the radiance in the direction of the observed is computed using the rendering equation [9]:

$$L(x, \vec{\omega}, t) = L_e(x, \vec{\omega}, t) + \int_\Omega f_r(x, \vec{\omega}', \vec{\omega}, t) L_i(x, \vec{\omega}', t)(\vec{\omega}' \cdot \vec{n})\, d\vec{\omega}'. \qquad (2)$$

Here, $L(x, \vec{\omega}, t)$ is the outgoing radiance at a surface location x with normal $\vec{n}$ in direction $\vec{\omega}$ at time t, f_r is the BRDF, L_e is the emitted radiance, and L_i is the incident radiance. Note that the integration of motion blur only happens at the pixel, and that all rays belonging to the same path have the same time.

3. Global Illumination Using Photon Maps

Before describing how to solve the rendering equation in time using photon maps let us briefly describe the basics of the photon mapping algorithm [7]. Photon mapping is a two-pass algorithm in which the first pass consists of tracing photons from the light sources through the scene and storing these photons in the photon map as they interact with elements in the scene. The second pass is rendering where the photon map is used to provide fast estimates of the reflected radiance, L, of the surfaces in the scene:

$$L(x,\vec{\omega}) \approx \frac{1}{\pi r^2} \sum_{p=1}^{n_p} f_r(x,\vec{\omega}_p,\vec{\omega})\Phi_p(x,\vec{\omega}'). \qquad (3)$$

This radiance estimate uses the n_p nearest photons to the location x, πr^2 is an approximation of the area covered by the photons, and Φ_p is the flux carried by photon p. This estimate assumes that the photons are located on a locally flat surface. As a consequence the estimate is invalid if, for example, the object is moving and the photons are distributed in time along the path of the moving object. Even if the object is static the time information in the photon map is ignored and, as we will show in the following, the standard radiance estimate is incorrect if the path of the incoming ray can intersect one or more moving objects during the time of the exposure interval.

4. Methods for Computing Motion Blur with Photon Maps

As observed in the introduction, we will restrict our attention to supersampling methods. Among such methods, there remain several choices for how to evaluate motion blur in the overall rendering pipeline. We will consider how photon mapping fits into each of these in turn.

As a cautionary example, we will show that some seemingly reasonable sampling strategies can lead to objectionable errors in the resulting images. Also, we will demonstrate that other simple strategies preserve consistency, but at the cost of performing substantial amounts of redundant computation. We will then propose an extension of the photon map radiance estimate into the temporal domain that addresses both problems, maintaining consistency while still allowing for efficient sampling.

4.1. Accumulation buffer

Perhaps the most straightforward method is the “accumulation buffer” approach, which takes the average of a number of in-between frames, each rendered independently at a specific time [4]. In this case, we would recompute the photon map for each partial frame. This approach can compute motion blur to an arbitrary degree of accuracy by using a sufficiently large number of in-between frames; however, the approach is inefficient. To obtain acceptable results free from temporal aliasing in fast-moving areas, a large number of in-between frames may be needed. Portions of a scene unaffected by motion (or for which coarser temporal sampling would suffice) are still rerendered in these in-between frames.

4.2. Distribution ray tracing

A distribution ray tracing renderer can evaluate motion blur by associating a time with each ray. By using adaptive sampling methods, such a renderer can be more efficient than the accumulation buffer approach by directing extra temporal samples only at portions of the image affected by motion. There are several ways in which photon mapping might be incorporated into such a renderer.

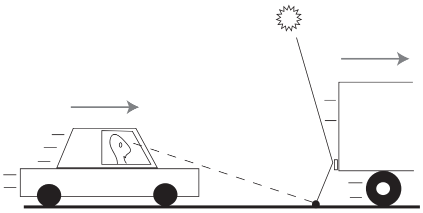


Figure 2: *Sunlight reflected onto the ground from a moving vehicle will appear as a sharp caustic to an observer moving at the same speed - even though both vehicles are moving rapidly relative to the ground beneath. The texture of the road will be motion blurred, but the caustic should not be.*

4.2.1. Do Nothing

The simplest approach is to not worry about temporally distributing the photons at all. We could perform the photon trace at a single time (say, the start of the interval), and use the corresponding map throughout. The resulting photon map does not represent the correct distribution of photons in time and space during the time interval considered. The direct-lighting contribution to the final image will have correct motion blur from the distribution ray tracing, but effects relying on the photon map, such as caustics, will not be evaluated correctly if they are affected by motion.

4.2.2. Distribute photons in time, but ignore their times in the reconstruction

We can distribute photons in time in the same manner we distribute rays: by associating a time with each photon and tracing it through the scene accordingly. This gives us a correct distribution of photons in space and time. The photons can then be stored in the standard photon map structure, without further regard for the associated times. In this approach, the photon map stores the projection of the correct distribution of photons in space and time (4 dimensions) into a distribution in space (3 dimensions). This might seem reasonable, but it leads to incorrect results in many cases.

The problem is that the standard photon map radiance estimate at a point in space will give the average intensity of illumination over the entire time period during which photons were traced. Instead, as we observed in section 2, we should compute the radiance at the time associated with the sample ray.

As a simple test scene for illustrating inconsistency, we will consider the scene shown in Figure 2. An observer is driving in a car behind a truck. Both vehicles are driving at the same speed and as a consequence the truck appears focused to the observer while other elements of the scene are motion blurred. Consider a caustic on the road generated by sunlight reflecting of the back of the truck. This caustic will move along with the truck and it will therefore appear sharp and focused at any given instant in time to the observer. However, if this caustic is simulated simply by distributing photons in time then the photons will be covering a larger area of the road corresponding to the distance moved by the truck within the shutter time. Therefore, the radiance estimate will predict a large blurred caustic whereas the correct caustic should be sharp and focused at all times.

4.2.3. Estimate radiance directly from photons distributed in time as well as space

A key element in the photon mapping algorithm is the ability to estimate radiance effectively from an irregular distribution of photons. This is done using filtered density estimation over the photons in a spatial neighborhood. It is consistent with the philosophy of the photon map to extend this approach to temporal sampling as well as spatial. The idea is to distribute the photons continuously in time, and retain this temporal information as an extra dimension in the photon map. In the following sections we will show how this temporal information can be used to derive a simple extension to the photon map that effectively enables filtering of photons in both time and space.

5. Building a Time Dependent Photon Map

For the purpose of estimating density in time as well as space we include time information in the photon map by adding an additional `float time` element to each photon. This increases the photon size from 20 bytes [6] to 24 bytes and results in the following photon representation:

```
struct photon {
  float x,y,z;       // position
  float time;        // photon time
  char q[4];         // energy (rgbe)
  char phi, theta;   // incident direction
  short flag;        // flag used in kdtree
}
```

If space is a concern then it would be possible to add the time to the photon structure presented by Jensen [6] by using the available bits in the photon flag to represent an 8-bit quantized time (this compressed representation of the time would be sufficient except in scenes with extreme motion blur).

To build the time dependent photon map we perform photon tracing in time by associating a random time with each photon path. At the light source we pick a uniformly distributed random time in the given shutter interval. This time is then used for the entire photon path, and the time is stored with all the photons generated along the path. Otherwise the photon tracing is exactly the same as for the standard photon map.

An important difference from standard photon tracing is that we propagate and store radiant energy instead of radiant flux (the energy is compressed in the same way as the flux in the standard photon map). The radiant energy, Q_e, of each emitted photon is:

$$Q_e = \frac{\Phi_l t_s}{n_e}, \tag{4}$$

where Φ_l is the power of the light source emitting the photon, t_s is the shutter or frame time, and n_e is the number of emitted photons.

6. A Time Dependent Radiance Estimate

In this section we will describe how the time dependent photon map enables us to compute radiance for a given surface location at a given time.

The reflected radiance, L_r, from a surface is given by:

$$L_r(x,\vec{\omega},t) = \int_\Omega f_r(x,\vec{\omega}',\vec{\omega},t) L_i(x,\vec{\omega}',t)(\vec{\omega}' \cdot \vec{n})\, d\vec{\omega}'. \tag{5}$$

Here, $L(x,\vec{\omega},t)$ is the radiance at surface location x in direction $\vec{\omega}$ at time t, f_r is the BRDF, L_i is the incident radiance, and $\vec{n}$ is the surface normal at x (Note that x depends implicitly on the time, since the intersection of a ray with a moving object depends on the time associated with the ray). The standard photon map radiance estimate applies the photon map to this equation using the relationship between radiance and flux, and similarly we will use the relationship between radiance and radiant energy, Q:

$$L_r(x,\vec{\omega},t) = \frac{d^3 Q(x,\vec{\omega},t)}{(\vec{n} \cdot \vec{\omega})\, d\vec{\omega}\, dA\, dt}. \tag{6}$$

Substituting this expression in equation 5 we obtain:

$$\begin{aligned} L_r(x,\vec{\omega},t) &= \int_\Omega f_r(x,\vec{\omega}',\vec{\omega},t) \frac{d^3 Q(x,\vec{\omega}',t)}{(\vec{n} \cdot \vec{\omega}')\, d\vec{\omega}'\, dA\, dt} (\vec{\omega}' \cdot \vec{n})\, d\vec{\omega}' \\ &= \int_\Omega f_r(x,\vec{\omega}',\vec{\omega},t) \frac{d^3 Q(x,\vec{\omega}',t)}{dA\, dt}. \end{aligned} \tag{7}$$

This equation shows that we have to estimate the density of the radiant energy in both space and time. To use the time dependent photon map, we will make a series of assumptions similar to those for the standard photon map. Specifically, we will assume that the nearest photons in the photon map in space and in time represent the incident flux at x at time t. This assumption enables us to approximate the integral by a sum over the radiant energy, Q_p, of the n_p nearest photons:

$$L_r(x,\vec{\omega},t) \approx \sum_{p=1}^{n_p} f_r(x,\vec{\omega}_p,\vec{\omega},t) \frac{Q_p}{\Delta A \Delta t}. \tag{8}$$

In the standard photon map it is assumed that the photons are located on a locally flat surface, and the area ΔA is approximated by πr^2 where r is the smallest possible radius of a sphere centered at x containing the n_p nearest photons. This is equivalent to a nearest neighbor density estimate [14].

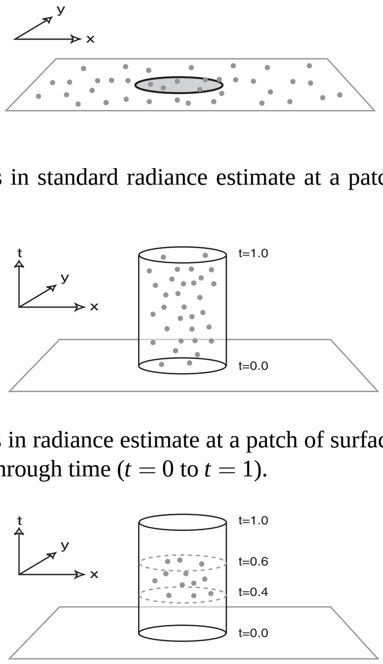
Essentially, we are estimating the incident radiance based on the photons that hit a small disc-shaped patch of surface centered about x, as in Figure 3a. For photons distributed in time, we can visualize the space-time distribution of photons by considering a patch of surface extruded into the time dimension, as in Figure 3b. We are considering a 4-dimensional analogue to a cylinder, in which a sphere in the 3 spatial dimensions is extruded through time from $t = 0$ to $t = 1$. For a basic visualization, however, it is convenient to omit the spatial dimension orthogonal to the surface and simply depict the surface patch extruded through time. The standard photon map radiance estimate, which ignores temporal distribution of photons, can be thought of as implicitly using a cylinder with maximum extent in time, as in Figure 3b.

Note that this standard estimate blurs illumination over the entire exposure time of the frame. This is entirely appropriate if there is no change in visibility along the eye path during the exposure time, since the pixel values ultimately computed are integrated over the full exposure time. If the path from the eye can potentially intersect any moving objects, however, then it is important to determine the incident illumination at the time associated with the ray, rather than blurring over the entire exposure time. In practice, our system tests whether an eye path has intersected the bounding boxes of moving objects. If not, then the standard radiance estimate can be used, ignoring any time information in the photons. When moving objects are intersected, however, we must use a revised density estimate that accounts for time dependence by restricting the photons in the estimate to be nearby in time as well as space.

This restriction to temporally neighboring photons can be represented as taking a cylindrical slice of space-time over a narrower range of times, as shown in Figure 3c. Such a cylindrical slice represents a compact neighborhood of space-time, local in all 4 dimensions. More details on the selection of neighboring photons will be described in the following section.

Let Δt be the time spanned by the nearest photons and let r be the radius of the smallest sphere centered at x enclosing the n_p photons. We can then approximate equation 8 as

$$L_r(x,\vec{\omega},t) \approx \frac{1}{\pi r^2 \Delta t} \sum_{p=1}^{n_p} f_r(x,\vec{\omega}_p,\vec{\omega},t) Q_p. \tag{9}$$



a) Photons in standard radiance estimate at a patch of surface.

b) Photons in radiance estimate at a patch of surface, shown extruded through time ($t = 0$ to $t = 1$).

c) Photons in time-dependent radiance estimate at a patch of surface over a restricted slice of time.

Figure 3: *Space-time diagrams illustrating the photons used in a radiance estimate.*

This is equivalent to a nearest neighbor density estimate in both space and time. For slightly better estimates it can be advantageous to include a smoothing kernel as well. This gives

$$L_r(x,\vec{\omega},t) \approx \frac{1}{\pi r^2 \Delta t} \sum_{p=1}^{n_p} f_r(x,\vec{\omega}_p,\vec{\omega},t) K_1\left(\frac{x_p - x}{r}\right) K_2\left(\frac{t_p - t}{\frac{1}{2}\Delta t}\right) Q_p. \tag{10}$$

where $K_1(y)$ is the smoothing kernel in space, and $K_2(y)$ is the smoothing kernel in time. The time of photon p is t_p, and it is assumed here that the maximum distance in time to a photon is $\frac{1}{2}\Delta t$. See Silverman [14] for an overview of different smoothing kernels and their respective trade-offs between bias (blur) and variance (noise). Specifically, in time one might use a uniform kernel, $K_2(y) = 1$, in which case the estimate is equivalent to the nearest neighbor estimate. Another option is the Epanechnikov kernel, $K_2(y) = \frac{3}{2}(1 - y^2)$ which gives less blur but slightly more noise. Notice that the kernels are different from the corresponding 2D kernels used in the space domain.

This time dependent radiance estimate is a consistent estimator. As the number of photons in the photon map is increased to infinity, we can increase n_p while Δt and ΔA go to zero in the estimates — thus it is possible to get an arbitrarily accurate estimate of the radiance at a given time on any locally flat surface.

Note that the technique in this section could be used to derive a time dependent radiance estimate for participating media. This would require changing the volume radiance estimate of Jensen and Christensen [8] to estimate density in time as well.

7. Locating nearby photons

The time dependent radiance estimate depends on the nearby photons in time and space. In this section we present two strategies for locating these photons. One is based on a simple two-pass extension of the standard photon map, and the second uses a 4D kd-tree.

7.1. A two-pass approach

The implementation of the photon map can be modified slightly to take the time dependence into account. As previously stated we need to quickly locate the nearest photons in both space and time. A natural approach that follows our derivation of the radiance estimate is to simply use the standard 3D photon map kd-tree to identify the photons nearest in space, and then in a second pass further restrict this set to the photons nearby in time. For our implementation, we determine the spatially nearby photons using the standard kd-tree search, and then use a quicksort partitioning algorithm to efficiently select the nearest of those photons in time. We use a randomized quicksort partition to avoid the poor worst case behavior of quicksort in the presence of nearly sorted elements. This is generally a good algorithmic practice, and it is particularly relevant here since photon times will tend to be correlated with position in a motion blurred caustic.

The number of photons used in the time density estimate is a user-defined fraction of the photons located in the first pass (we have found 50% to work well for our test-scenes). This means that we need to locate more photons than the standard photon map in order to have enough photons to get a smooth density estimate. Note that the maximum time difference should go to zero in the limit in order to make the final estimate consistent.

An important detail is that the photons searched over can be trivially clipped to only those that are within certain maximum allowed distances in space and time. The standard photon map routinely restricts the search for k-nearest neighbors to a maximum distance, basing the estimate on fewer than k photons rather than using photons that are unreasonably far away. This is essential to maintain the quality of the estimate in dimly lit regions where photons are sparse, and also speeds the search for nearest neighbors since only photons within this range need be considered. We similarly restrict the maximum allowed difference in time for photons in the time-dependent estimate, thereby obtaining the same benefits: avoiding errors in the estimate due to sparsely distributed photons, and speeding the search for nearest neighbors.

The two-pass approach is straightforward to incorporate into an existing photon map implementation. Our results indicate that the performance is quite good, adding little performance penalty to the standard radiance estimate. In addition the method naturally adapts to local variations in the space and time photon density.

7.2. Locating photons using a 4D kd-tree

An alternative to the two-pass approach is extending the standard 3d photon map kd-tree into a 4d kd-tree with time as the extra dimension. The 4d kd-tree is potentially faster since it makes it possible to directly locate the nearest photons in both space and time.

In order to locate the nearest photons we need to specify a distance metric. In 3d this is trivially given as the distance to the photon, but in 4d the answer is not as simple since space and time do not share common units or a natural distance metric. Given one photon that is 2 units away in distance and 1 unit away in time, and a second photon that is 1 unit away in distance and 2 units away in time, there is no clear answer to which is the nearer neighbor. We can address this problem of specifying a distance metric over space and time by providing the user with a parameter κ to control the relative weights of spatial versus temporal distance. By simple extension of the Euclidean distance metric for space, we can then define a global "space-time distance":

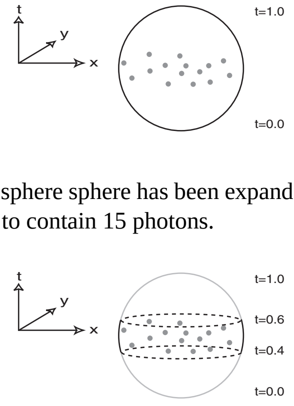
$$d = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2 + \kappa \Delta t^2}.$$

or simply:

$$d = \sqrt{r^2 + \kappa \Delta t^2}.$$

This resolves the immediate problem, but introduces an extra user control with an unintuitive effect on image quality. Also, such an explicitly specified distance metric would apply throughout the scene, whereas our earlier implementation adaptively varies the relative sizes of r and Δt depending on the local distribution of photons.

Given the user-specified distance metric, we can locate neighboring photons by expanding a 4D (hyper-)sphere until it contains k photons. Figure 4a depicts a sphere that has been expanded until just large enough to contain 15 photons. Since we are no longer using a space-time cylinder to locate nearest photons, it is no longer appropriate to substitute $\pi r^2 \Delta t$ for $\Delta A \Delta t$ as in equation 9. Instead, we will need to use the volume of the space-time sphere over which we have located the photons. Since we continue to assume that we are at a small patch of a locally flat surface, the volume under consideration is that of the 3D sphere in x, y, and t, which has volume $\frac{4\pi d^3}{3\kappa}$ (where κ is again the scale factor



a) The hyper-sphere sphere has been expanded until it is just large enough to contain 15 photons.

b) The sphere does not accurately represent the volume occupied by this set of photons. Here, the photons are concentrated in time, and it is more appropriate to use a sphere clipped by the minimum and maximum time values.

Figure 4: *Using a 4D hypersphere to locate photons.*

relating temporal and spatial distance). However, no matter what value of κ is chosen to relate spatial and temporal distance, there can still be regions in the scene where photons are not equally well distributed in space and time. In the examples of Figure 4, the photons are more concentrated in time. Consequently, the 4D bounding sphere does not accurately represent the volume occupied by the photons, and using the volume of the sphere as $\Delta A \Delta t$ in the radiance estimate introduces an excessive amount of temporal blur. This is analogous to the problem with excessive blur near edge discontinuities in the standard photon map [6], but is more problematic since it affects every time-dependent radiance estimate. One fairly straightforward fix is to clip the sphere against the minimum and maximum times represented by the photons, as shown in Figure 4b. It is possible to derive a simple closed form expression for the volume of such a clipped sphere, and substituting this volume for $\Delta A \Delta t$ in equation 9 eliminates the unnecessary temporal blur.

We have tested the 4D kd-tree method, where the nearest photons are located within a 4D sphere according to a space-time distance metric (based on a user defined parameter κ relating spatial and temporal distance). Our implementation of the radiance estimate for this method is using the clipped sphere volume as described above. The implementation yielded results roughly comparable in quality to those obtained with the two-pass method described above. Overall, however, we found that the complications associated with the 4D kd-tree made it considerably less elegant and appealing than the two-pass method. As a consequence we use the two-pass method for all the examples presented in this paper.

8. Results

We have implemented the time dependent radiance estimate in a Monte Carlo ray tracer that supports both photon mapping and motion blur. The motion blur is rendered by adaptively sampling ray paths with different times through each pixel until the contrast is below a given threshold. All our examples use the two-pass method for locating the nearest photons. The user specified fraction is 50% for all examples, and we use 50 to 100 photons in the combined time-space density estimate, which means that we have to locate 100-200 spatially neighboring photons in the initial kd-tree search. The kd-tree uses the standard Euclidean distance metric to locate the nearest photons (i.e. the photons are located within a sphere around the point of interest).

Our first test scene is shown in Figure 7. It is composed of a diffuse cube lit from above through a glass sphere. In Figure 8, the cube is shown moving downward, causing it to be motion blurred. We rendered this scene using 4 different methods. Figure 8a shows the result of a path tracing simulation where 10,000 paths randomly distributed in time have been used to compute the radiance through each pixel (this “reference” image is still quite noisy even with 10,000 paths per pixel). Figure 8b shows the result of averaging together 20 images of static scenes at equally spaced times in the shutter interval. Figure 8c shows the result of distributing photons in time and using the standard photon map radiance estimate, and finally Figure 8d shows the result of the new time dependent radiance estimate applied to the temporally sampled photon map. Note that the standard radiance estimate significantly underestimates the intensity of the caustic in this example. The three photon map methods all use a total of 900,000 caustic photons. To provide a fair comparison, the accumulation buffer rendering was performed by dividing the caustic photons evenly among each of the 20 in-between frames using 45,000 photons for each frame.

The scene of Figure 8 helps illustrate why we must consider the local distribution of photons in time to obtain a correct radiance estimate. The photons stored near a given point in this scene represent radiant energy concentrated in a very narrow window of time — the time when the point lies on the surface of the moving cube. The standard photon map radiance estimate fails to take this into account, and averages that concentrated radiant energy over the entire exposure interval — even times when the surface is nowhere near. Since the photon map radiance contribution is only evaluated at surfaces, the remaining energy is lost from the estimate.

In contrast, the time dependent radiance estimate correctly renders the caustic. This is particularly noticeable in the graphs comparing the intensity along a line through the caustic with the result of the path tracing reference solution. Note that both the accumulation buffer and the time dependent radiance estimate work well for this scene. In the timing comparison in Table 1 we can see that the new time dependent estimate is slightly faster than the accumulation buffer. The difference is not substantial since this scene is dominated by motion blur, and as a result the adaptive sampling of motion blur is less beneficial here.

Figure 9 shows a scene in which the spot from a laser pointer (aligned alongside the camera) is reflected off the back of a moving truck. (For added clarity, a still frame of this scene is shown in Figure 5). This causes a motion blurred stripe of laser light to be reflected on the ground. However, in the back of the truck, we can see a reflection of this spot on the ground. The interesting point about this scene is that while the laser spot on the ground should be blurred into a stripe by the truck's motion, its reflection in the back of the truck should always appear as a sharply focused spot! The diagram in Figure 6 should help make this point clear. Notice that the observer is static in this scene unlike the earlier example illustrated in Figure 2.

All versions of the truck scene were rendered using a total of 40,000 caustic photons. The comparison of rendering techniques in Figure 9 shows how the accumulation buffer method (a) correctly captures the sharp spot reflected in the back of the truck, but even with 9 in-between frames, it cannot smoothly reproduce the laser stripe on the ground. Applying the standard radiance estimate to photons distributed in time (b) smoothly renders the blurred laser stripe on the ground, but incorrectly blurs the reflection seen in the back of the truck. Our proposed time dependent photon map (c) correctly renders both effects. Furthermore, as the timings in Table 1 show, the proposed method has performance comparable to that of the standard radiance estimate, and substantially faster than the accumulation buffer method. The reason for the better timings is that the caustic and motion blur occupy a relatively small portion of an image — this is often the case in rendered images. Consequently, this scene demonstrates the importance of using efficient, adaptive sampling techniques, which only do extra work (e.g. for caustics or motion blur) in the parts of an image that require it.

9. Conclusions

In this paper we have addressed the problem of using photon mapping in scenes with motion blur. We have derived a simple time dependent radiance estimate. The new insight is that temporal sampling can be simulated by estimating the photon density in time as well as space. The time dependent radiance estimate provides smooth and visually pleasing approximations of radiance in moving scenes, without the distinctive strobing/banding artifacts of the accumulation buffer. It is consistent, and will converge to an arbitrarily accurate solution simply by using a sufficient number of photons in the map. Most importantly, the method is efficient. Its performance is at least as good as accumulation buffer methods, and for typical scenes it can be substantially faster. It imposes a small penalty over the standard photon map ra-

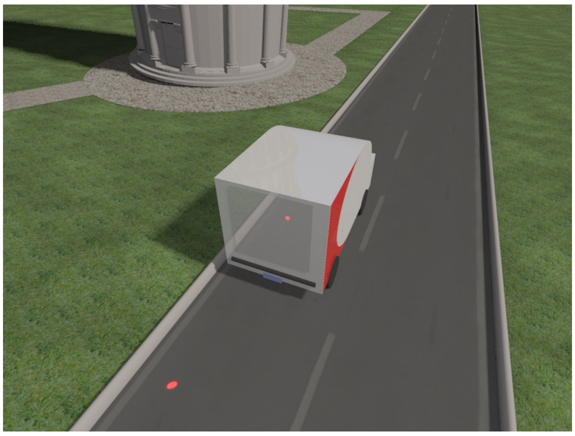


Figure 5: *A shiny truck parked by the tower of Pisa. A red laser aligned with the camera shines a spot on the back of the truck, which reflects the spot onto the ground. This spot on the ground can be seen again in the reflection in the back of the truck.*

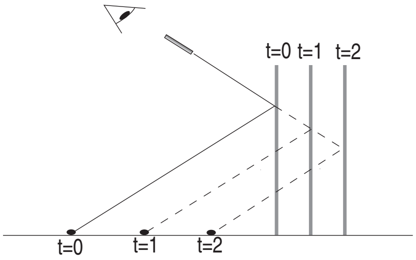


Figure 6: *Diagram of view along line of sight of a laser pointer shining on a moving planar reflector. The spot cast on the ground gets blurred into a stripe by the mirror's motion, but its reflection in the mirror, as seen along the line of sight illustrated, is a focused spot.*

Method	Consistent?	Render times†	
		Fig. 8	Fig. 9
Path tracing	Yes	9+ hrs.	n/a
Accumulation buffer	Yes	47sec	316sec‡
Standard radiance estimate on photons distributed in time	**No**	37sec	74sec
Time dependent radiance estimate	Yes	43sec	72sec

† Render times for photon map techniques include the time for the initial photon tracing.

‡ Note that the accumulation buffer technique would require many more in-between frames (and consequently longer render time) to correctly reproduce the blurred caustic on the ground.

Table 1: *Performance comparison of the techniques.*

diance estimate, but has the considerable advantage of giving correct results.

We have devoted a significant amount of space in section 4 to considering various flawed or inefficient methods for computing motion blur in the presence of photon maps. This is not without good reason. Both photon mapping and motion blur are widely used techniques, and there are a number of rendering systems that support both. However, in the absence of any prior published research on how to correctly evaluate the photon map for time-varying scenes, these existing implementations almost surely use one of these naive methods by default. Consequently, we have felt it worthwhile to examine the shortcomings of these approaches, which can reasonably be expected to underlie any rendering system that supports both photon maps and motion blur without having carefully considered the interrelationship between the two.

One promising opportunity for future work is to extend our technique to take advantage of inter-frame redundancy over an animated sequence, similar to Myszkowski et al. [12].

10. Acknowledgements

Thanks to the reviewers and to Maryann Simmons, Steve Marschner, and Pat Hanrahan for helpful comments. This research was funded in part by the National Science Foundation Information Technology Research grant (IIS-0085864). Mike Cammarano is supported by a 3Com Corporation Stanford Graduate Fellowship and a National Science Foundation Fellowship.

References

1. Gonzalo Besuievsky and Xavier Pueyo. A Motion Blur Method for Animated Radiosity Environments. *Winter School of Computer Graphics, 1998*. 1998.
2. Edwin Catmull. An analytic visible surface algorithm for independent pixel processing. *Computer Graphics (SIGGRAPH 1984 Proceedings)*, **18**:109-115, 1984.
3. Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed Ray Tracing. *ACM Computer Graphics (Proc. of SIGGRAPH '84)*, **18**:137–145, 1984.
4. Paul E. Haeberli and Kurt Akeley. The accumulation buffer: Hardware support for high-quality rendering. *Proceedings of SIGGRAPH 1990*, 309–318, 1990.
5. Jonathan Korein and Norman Badler. Temporal Anti-Aliasing in Computer Generated Animation. *ACM Computer Graphics (Proc. of SIGGRAPH 1983)*, **17**:377–388, 1983.
6. Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*, A K Peters, 2001.
7. Henrik Wann Jensen. Global Illumination Using Photon Maps. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 21–30, New York, NY, 1996. Springer-Verlag/Wien.
8. Henrik Wann Jensen and Per H. Christensen. Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps. *Proceedings of SIGGRAPH 1998*, 311–320, 1998.
9. James T. Kajiya. The Rendering Equation. *ACM Computer Graphics (Proc. of SIGGRAPH '86)*, **20**:143–150, 1986.
10. Eric P. Lafortune. *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*. Ph.d. thesis, Katholieke University, Leuven, Belgium 1996.
11. Ryan Meredith-Jones. Point Sampling Algorithms for Simulating Motion Blur. *Master's Thesis*, University of Toronto, 2000.
12. Karol Myszkowski, Takehiro Tawara, Hiroyuki Akamine and Hans-Peter Seidel. Perception-Guided Global Illumination Solution for Animation Rendering. *Proceedings of SIGGRAPH 2001*.
13. Michael Potmesil and Indranil Chakravarty. Modeling Motion Blur in Computer Generated Images. *ACM Computer Graphics (Proc. of SIGGRAPH 1983)*, **17**:389–399, 1983.
14. B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability 26. Chapman & Hall/CRC. 1998.
15. Brian A. Wandell. *Foundations of Vision*, Sinauer Associates, 1995.
16. Greg Ward, Francis Rubinstein and Robert Clear. A Ray Tracing Solution to Diffuse Interreflection. *Proceedings of SIGGRAPH 1988*, 85–92, 1988.

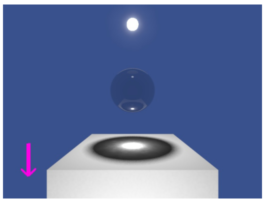
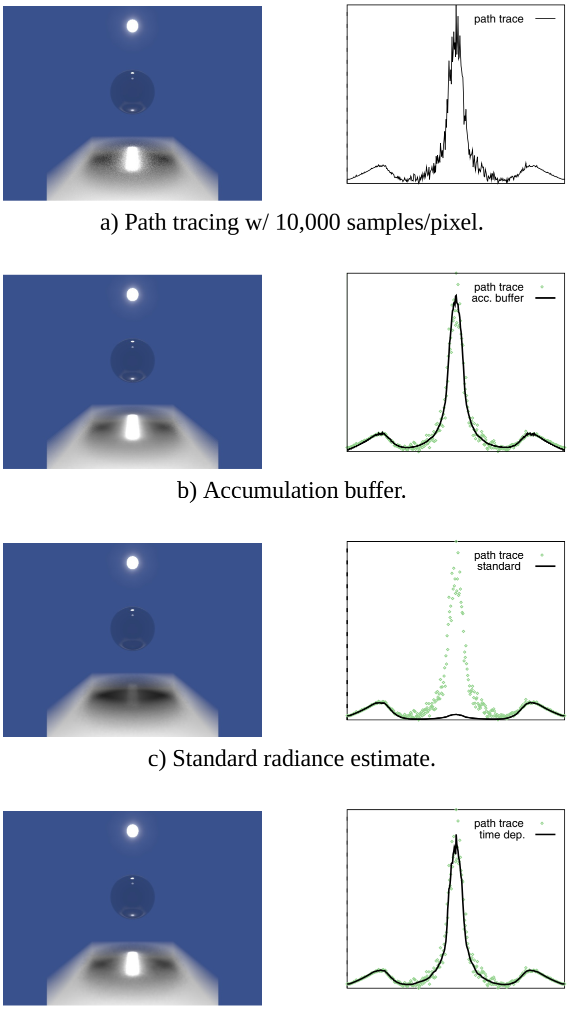


Figure 7: *Still frame showing the setup for the cube scene. The arrow indicates the direction the cube will move in Figure 8.*



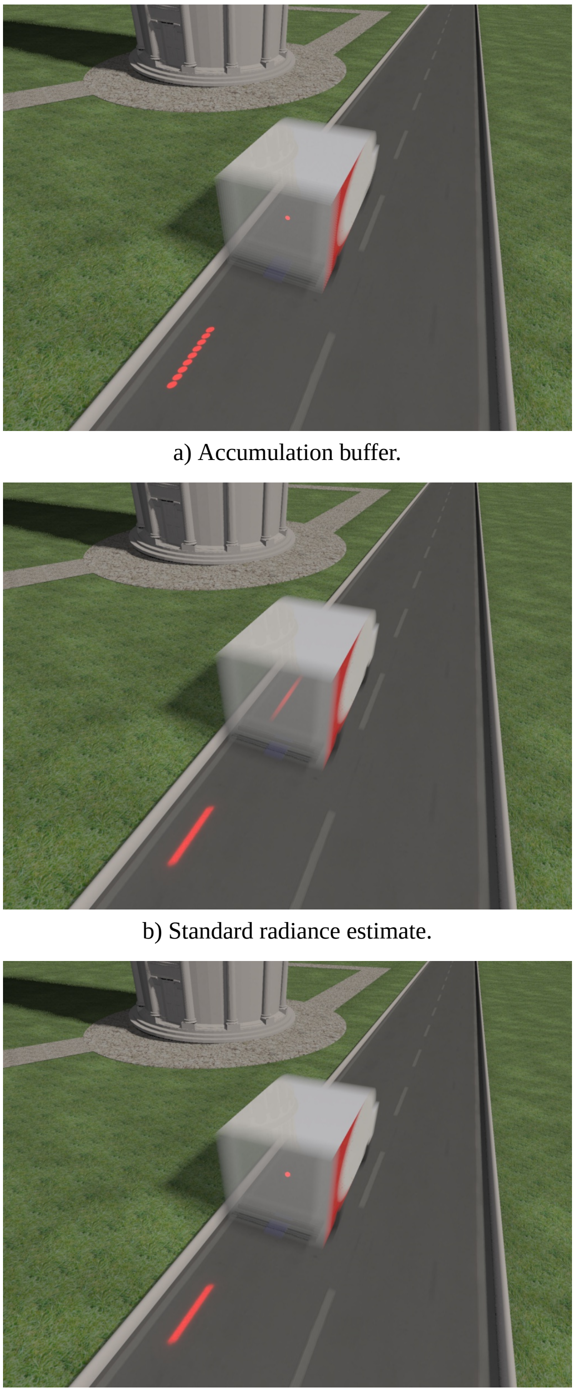
a) Path tracing w/ 10,000 samples/pixel.

b) Accumulation buffer.

c) Standard radiance estimate.

d) Time dependent radiance estimate.

Figure 8: *Motion blurred images of the moving cube scene rendered using various techniques. All 4 images were rendered at 640x480. To illustrate the variance associated with the techniques, the graphs show pixel values for each image along a horizontal line through the caustic. For comparison, the graph for each photon map method also shows pixel values from the path traced "reference" image - which is still very noisy even with 10,000 samples per pixel.*



a) Accumulation buffer.

b) Standard radiance estimate.

c) Time dependent radiance estimate.

Figure 9: *Moving truck scene, also depicted without motion in Figure 5. The laser spot reflected onto the ground is blurred into a stripe by the truck's motion. However, the reflection of this spot seen in the back of the truck should not be blurred, as illustrated in Figure 6.*