

# Variance reduction using interframe coherence for animated scenes

Peng Zhou<sup>1</sup> (✉), Yanyun Chen<sup>1</sup>

© The Author(s) 2015. This article is published with open access at Springerlink.com

**Abstract** In an animated scene, geometry and lighting often change in an unpredictable way. Rendering algorithms based on Monte Carlo methods are usually employed to precisely capture all features of an animated scene. However, Monte Carlo methods typically take a long time to produce a noise-free image. In this paper, we propose a variance reduction technique for Monte Carlo methods which exploits coherence between frames. Firstly, we introduce a dual cone model to measure the incident coherence intersecting camera rays in object space. Secondly, we allocate multiple frame buffers to store image samples from consecutive frames. Finally, the color of a pixel in one frame is computed by borrowing samples from neighboring pixels in current, previous, and subsequent frames. Our experiments show that noise is greatly reduced by our method since the number of effective samples is increased by use of borrowed samples.

**Keywords** physically based rendering; Monte Carlo methods; ray tracing; global illumination; animation

## 1 Introduction

Physically based animation has widespread applications in the movie industry, architectural design, and visualization. In computer graphics, Monte Carlo methods are the most practical way to render animated scenes. However, the convergence of Monte Carlo methods is inherently slow—it is typically a time consuming task for rendering

systems based on pure Monte Carlo methods to produce a noise-free image. Although there are some caching techniques such as irradiance caching and photon mapping which can provide smooth images, they usually suffer from flickering problems because of bias. Despite having been studied for decades, variance reduction is still an open area full of challenges.

In this paper, we propose a novel variance reduction technique for Monte Carlo methods used to render animated scenes. We use a dual cone model to measure the coherence between radiance samples centered at the intersection of the camera ray with the scene. Coherence is a metric used to describe the similarity of radiance samples, which enables a pixel to reuse samples from its neighboring pixels with a suitable weight. Sample reuse happens over a sequence of consecutive frames instead of a single frame. Thus, multiple frame buffers must be allocated to store the samples (instead of color) from each frame. The contribution of our work includes two parts. Firstly, we introduce a scheme for inter-frame sample reuse. Secondly, we provide a criterion to avoid using invalid samples.

## 2 Related work

**Traced global illumination** is mainly used to produce physically based images in an unbiased way for animated scenes with moving objects and dynamic lighting. Refs. [1] and [2] seamlessly integrate photon mapping and bidirectional path tracing into a more robust combined rendering algorithm via multiple importance sampling, overcoming shortcomings of traditional bidirectional path tracing. Ref. [3] proposed gradient-domain rendering for Monte Carlo image synthesis. They

<sup>1</sup> Institute of Software, Chinese Academy of Sciences, Beijing 100190, China. E-mail: P. Zhou, zhoupeng20001@sina.com (✉); Y. Chen, chenyy@ios.ac.cn.

Manuscript received: 2015-10-23; accepted: 2015-10-29

found that estimating image gradients is also possible using standard Monte Carlo algorithms, and furthermore, even without changing the sample distribution, this leads to significant error reduction. Ref. [3] introduced a bidirectional formulation for light transport and a set of weighting strategies to significantly reduce the bias in VPL (virtual point light)-based rendering algorithms. Because of the need to be unbiased, the samples must be used independently, which limits the possibility for reuse of samples. Ref. [4] used statistical dependency between the outputs and inputs of a rendering system to reduce the importance of the sample values affected by Monte Carlo noise when applying an image-space, cross-bilateral filter. This removes only the noise caused by randomness but preserves important scene details. Our method can help to greatly improve the performance of these methods by exploiting redundant information from samples with coherence in object space.

**Cached global illumination** is mainly used to produce smoothed physically based images for static scenes by caching and interpolating results of lighting computations. Ref. [5] observed that indirect irradiance typically varies smoothly. A small number of sample points is sufficient to calculate and cache the indirect diffuse irradiance while keeping the error under a given threshold. Ref. [6] introduced second order gradients (Hessians) as a principled error bound for the irradiance; this approach significantly outperforms the classic split-sphere heuristic. Ref. [7] first introduced photon mapping, in which a spatial structure called a photon map is used to store and estimate indirect lighting at different locations within the scene. Ref. [8] introduced progressive photon mapping where the global illumination solution becomes increasingly accurate and can be visualized to provide progressive feedback. Ref. [9] introduced a new formulation of progressive photon mapping based on a probabilistic approach. Ref. [10] proposed reversed photon mapping which spreads camera importance into the scene and evaluates photon contributions by lookup in the importance map. The interpolating step required inevitably introduces bias into the image and the cache has to be reused excessively to pay off. The filtering technique employed by these methods cannot completely

suppress the errors, so that the images for different frames often have inconsistent color, making the animation flicker. Our method does not suffer from a flickering problem, as we use a dual cone model to give each sample a reasonable reuse weight and rate.

**Interactive ray tracing** focuses on optimizing acceleration structure such as grids, *kd*-trees, and BVHs (bounding volume hierarchy). Ref. [11] proposed interactive frustum-bounded packet tracing on a uniform grid. Tracing performance can be accelerated by a factor of 10 by incrementally computing the overlap of the frustum with a slice of grid cells. Ref. [12] proposed a method to efficiently extend refitting for animated scenes with tree rotations. Ref. [13] proposed a ray-space hierarchy method, which can handle truly dynamic scenes without the need to rebuild or update the scene hierarchy. Ref. [14] proposed a method to selectively restructure the BVH for ray tracing dynamic scenes. These methods optimize a different part of the rendering pipeline from our method, so they play a complementary role to our method in the rendering of animated scenes.

### 3 Measuring incident coherence using a dual cone

The light transport at a point in object space is described by the rendering equation:

$$L_o(x, \omega_o) = \int_{\Omega} L_i(x, \omega_i) f(x, \omega_o, \omega_i) V(x, \omega_i) \cos \theta d\omega \quad (1)$$

where  $L_o$  is the outgoing radiance leaving shading point  $x$  in direction  $\omega_o$ ,  $L_i$  is the incident radiance arriving at  $x$  from direction  $\omega_i$ ,  $f$  is the bidirectional reflectance distribution function (BRDF), and  $V$  is the visibility between the light source and shading point.

The goal of the dual cone model is to provide a tool for measuring the similarity of lighting conditions for different locations. The model assumes that all light sources have non-zero projected area on the illuminated surface and have constant intensity. For a given light source, lighting coherence between two points can be measured as the size of the solid angle through which the light source is visible from both locations. Precisely computing the solid angle for shared visibility can be quite expensive because the projected shape on the unit sphere can be very

complex. To make the computation practical, in the dual cone model the projected shape of the visible part of the light source on the unit sphere is idealised as a disk that subtends the same solid angle as the original shape.

Given a location  $x$ , the primary cone is defined by three factors: (i) the direction  $\omega$  to the light source, (ii) the solid angle  $A$  subtended by the light source, and (iii) the distance  $R$  to the light source, which we call the light source distance. The geometry is shown in Fig. 1. The secondary cone is constructed for any other location  $x + \Delta x$  for whose similarity of lighting conditions we wish to compare with location  $x$ . Note that the secondary cone may not be placed exactly at  $x + \Delta x$ . The two cones have exactly the same height and size, and are parallel to each other. The metric of similarity is given by the ratio of the area of the overlapping region to the surface area at the base of the cone. To be specific, the metric is given by  $C = S_{\text{overlap}}/S_{\text{disk}}$ , where  $S_{\text{overlap}}$  is the overlap area (the yellow region in Fig. 1) and  $S_{\text{disk}}$  is the area of the disk at the cone base.  $S_{\text{overlap}}$  consists of two symmetrical bows, each of which can be computed by subtracting the area of inner triangle  $S_{\text{triangle}}$  (the

blue region in Fig. 1) from the sector:

$$\begin{aligned} S_{\text{overlap}} &= 2(S_{\text{sector}} - S_{\text{triangle}}) \\ S_{\text{sector}} &= \pi r^2 \alpha / 2\pi \\ S_{\text{triangle}} &= \frac{d}{2} \sqrt{r^2 - \left(\frac{d}{2}\right)^2} \\ \alpha &= 2 \arccos \frac{d}{2r} \end{aligned}$$

The relationship between half cone angle  $\beta$  and the solid angle  $A$  subtended by the basal disk can be computed as

$$\begin{aligned} A &= \int_0^{2\pi} \int_0^\beta \sin \theta d\theta d\phi = 2\pi(1 - \cos \beta) \\ \beta &= \arccos \left(1 - \frac{A}{2\pi}\right) \end{aligned}$$

The overlap ratio  $C$  is finally given by

$$\begin{aligned} C &= \frac{S_{\text{overlap}}}{S_{\text{disk}}} = \frac{2(S_{\text{sector}} - S_{\text{triangle}})}{\pi r^2} \\ &= \frac{2}{\pi} (\arccos g - g\sqrt{1 - g^2}) \end{aligned} \quad (2)$$

where  $g = d/(2r)$ .

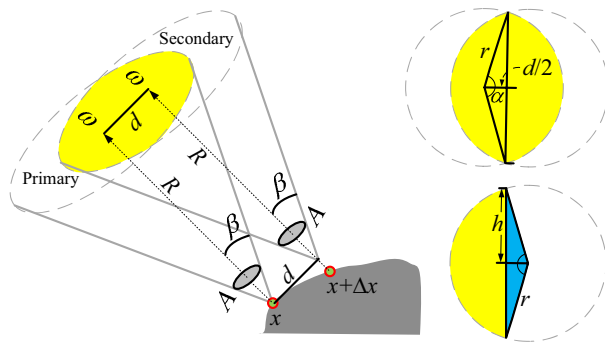
The projected distance  $d$  is given by  $d = \|\Delta x - (\omega \cdot \Delta x)\omega\|$ , and the disk radius is  $r = R \tan \beta$ , which lets us compute  $g$  as

$$g = \frac{\|\Delta x - (\omega \cdot \Delta x)\omega\|}{2R \tan \beta} \quad (3)$$

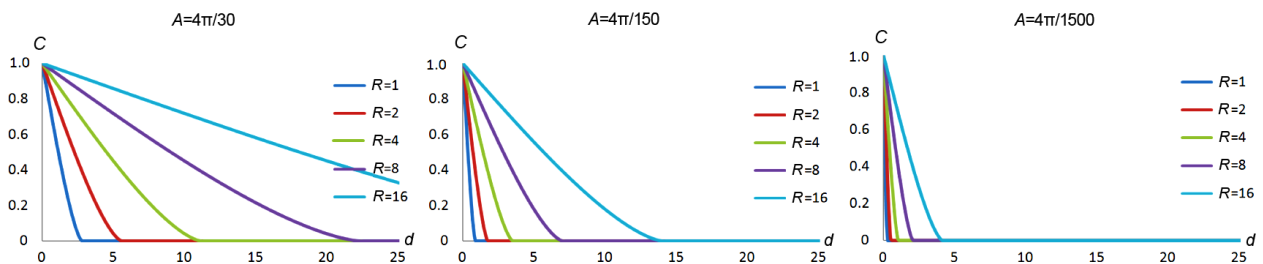
Figure 2 shows the overlap ratio  $C$  for different lighting environments. Clearly,  $C$  increases with the solid angle  $A$  ( $4\pi/30$ ,  $4\pi/150$ , and  $4\pi/1500$ ) subtended by the light source, and the light source distance  $R$  (1, 2, 4, 8, and 16). It decreases with the projected distance  $d$ . This is consistent with our expectation: smaller visible light sources casts harder shadows, and larger light source distances results in softer shadow.

## 4 Setup lighting environment

The dual cone model ignores the impact of the incident light's intensity. When comparing



**Fig. 1** The dual cone model. The primary cone is defined by location  $x$ , direction  $\omega$ , solid angle  $A$ , and light source distance  $R$ . The secondary cone is constructed by moving the primary cone along vector  $d$ , where  $d$  is the projected distance between  $x$  and  $x + \Delta x$  in the plane defined by the base of the primary cone.



**Fig. 2** Overlap ratio for different light source distance  $R$ , solid angle  $A$ , and projected distance  $d$ .

two radiance samples, it assumes that the lighting conditions are constant. The original lighting model must be converted to a group of area lights with low radiance variance, in order to meet the assumption.

For image based lighting, such as an environment map, the image is subdivided into an axis-aligned *kd*-tree by recursively cutting the light intensity in half. The leaves of the *kd*-tree are used as area lights, whose intensities are approximated by the average of all pixels each contains. In the worst case, every pixel will be used as an area light. For extremely smooth images, the entire image may be used as a single area light.

The local light source is usually modeled by a polygon, and in practice, its intensity is specified by a constant value or a texture map. Local lights whose intensity is determined by a texture map can be processed as image based lights as described above. Local lights with constant intensity do not need any change. The approach here demonstrates an alternative for direct lighting, but for small area light sources we still recommend the traditional approach which samples light sources, since it is more efficient than sampling the BSDF (bidirectional scattering distribution function).

#### 4.1 Exploiting intra-frame coherence

For each pixel in an image, camera ray samples from nearby pixels are borrowed. The assumption lying behind this is that nearby pixels have a high probability of representing close by points in the object. However, in some cases this assumption may fail. First, adjacent camera rays may hit two different objects that are far away from each other. Second, two camera rays may hit two surfaces whose orientations are very different. In the first case, we project the pixel block centered at the current pixel into object space and use the projected area as a search region for valid neighboring pixels. To cover the second case, we decide that if the cosine of the angle between the two surface normals is no less than 0.9, then the two pixels can borrow sample from each other.

Equation (4) gives the final contributions to a pixel after borrowing samples in the current frame.

$$L = \frac{1}{N} \sum_{i=1}^N \frac{L(x_i) f(x_i) \cos \theta_i}{p(x_i)}$$

$$\begin{aligned} &= \frac{1}{N} \sum_{i=1}^N \frac{1}{\sum_{k=1}^M C_{ik}} \sum_{j=1}^M \left( C_j \frac{L(x_{ij}) f(x_{ij}) \cos \theta_{ij}}{p(x_{ij})} \right) \\ &= \frac{1}{N} \sum_{i=1}^N \frac{1}{W} \sum_{j=1}^M (C_j \cdot F_{ij}) \end{aligned} \quad (4)$$

In the first line,  $x_i$  is the  $i$ -th path sample for Monte Carlo sampling,  $L(x_i)$  is the incident radiance,  $f(x_i)$  is the BRDF,  $\cos \theta_i$  is a geometric factor, and  $p(x_i)$  is probability density function (PDF). In the second line  $x_{ij}$  is the  $j$ -th neighbor of  $x_i$ .  $L(x_{ij})$ ,  $f(x_{ij})$ ,  $\cos \theta_{ij}$ , and  $p(x_{ij})$  have already been evaluated for  $x_{ij}$  in a previous pass. The final value of  $x_i$  is evaluated by borrowing sampled values from all its neighbors (instead of completely tracing a path).  $C_j$  gives a weight indicating similarity of lighting conditions between  $x_i$  and  $x_{ij}$ .  $\sum_{k=1}^M C_{ik}$  sums the weights from all neighbors of  $x_i$ . The accuracy of the dual cone approach is approximately equivalent to a final gathering process with  $M$  rays per pixel, but only involves the cost of tracing one ray. After eliminating invalid neighbors, the overlap ratio for valid neighbors is computed using Eq. (2).

#### 4.2 Exploiting inter-frame coherence

Since the coherence considered is in object space, camera motion does not change the coherence, so it is possible to re-project samples from a previous frame to current frame. However, the camera ray of one frame may hit a different surface from the ray of another frame, so visibility must be rechecked to validate samples from such previous frames. Furthermore, the object and light can also be in motion, so the visibility of secondary rays associated with valid samples must also be rechecked.

In order to share samples between frames, we render multiple frames at the same time. To store consecutive frames, a frame buffer queue is allocated. Frames are arranged in ascending order. The queue size is left as a parameter for the user. One-sample-per-pixel rendering is performed for each frame in the queue. A sample from one frame can be reused by any frame in the queue. The rendering process is iteratively performed until the required number of samples per pixel is met.

Equation (5) gives the final contributions to a pixel

by borrowing samples from multiple frames.

$$\begin{aligned} L &= \frac{1}{N} \sum_{i=1}^N \frac{1}{W} \sum_{j=1}^M (C_j \cdot F_{ij}) \\ &= \frac{1}{N} \sum_{i=1}^N \frac{1}{W} \sum_{b=1}^B \sum_{j=1}^{M_b} (C_{bj} \cdot F_{bij}) \end{aligned} \quad (5)$$

where  $M = \sum_{b=1}^B M_b$ . In this modified version of Eq. (4),  $B$  is the size of the frame buffer queue, and  $M_b$  is the number of samples borrowed from frame  $b$ .

### 4.3 The rendering framework

The dual cone model rendering process has two passes.

1) Sampling pass. For each pixel in each frame, we use a single gathering ray to sample radiance for each pixel, and record the result to an image cache. The record includes: radiance/direction/length of sampling ray, solid angle subtended by visible light, position and surface normal at the camera ray intersection.

2) Rendering pass. For each pixel in each frame, records from  $N \times N$  pixel blocks centered at the current pixel position in every frame are used to determine the final color. First, we trace a ray for the current pixel to find an intersection  $i$ . Then we validate the borrowed record, by checking normal difference, distance, and visibility. Next, we compute the overlap ratio  $C$  according to Eq. (2) to weight the lighting similarity between intersection  $i$

---

#### Algorithm - Pass 1

```

1  for each frame  $a$  in frame buffer queue do
2  for each pixel  $i$  in frame  $a$  do
3    Trace one camera ray from  $i$  into scene
4    Record the position and normal of intersection
5    Trace one ray based on BSDF
6    if ray missed, evaluate radiance using
7      chosen directional light, record solid angle
8    else ray is occluded
9      set radiance to zero, record solid angle
10   Record radiance, ray direction and ray length
11 end for

```

---

#### Algorithm - Pass 2

```

1  for each frame  $a$  in frame buffer queue do
2  for each pixel  $i$  in frame  $a$  do
3    Trace one camera ray to find intersection
4    for each frame  $b$  in frame buffer queue do
5      for each pixel  $j$  in  $N \times N$  block centered at  $i$  do
6        Get record from pixel  $j$  in frame  $b$ 
7        Check normal, distance and visibility
8        Ignore invalid record
9        Compute  $C$  for  $i$  and  $j$  using Eq. (2)
10       Update pixel  $i$  using Eq. (5)
11     end for
12   Save contribution of pixel  $i$  in frame  $a$ 
13 end for
14 end for

```

---

and the intersection cached in the record of the  $j$ -th neighbor. Finally we sum the weighted radiance values from records according to Eq. (5) to give the final pixel color.

### 4.4 Progressive refinement

The two-pass rendering described above is essentially an approximated version of final gathering with one sample per pixel, while the number of gathering rays is determined by the number of available pixel records. It is a special case of Eq. (5) where  $N = 1$ .

---

#### Algorithm - Progressive refinement

```

1  Subdivide environment into  $N$  directional lights
2  Build light tree
3  Allocate frame buffer queue of size  $B$ 
4  for every  $B$  frames in animation
5    Clear frame buffer queue
6    for  $i = 1$  to  $n$  do
7      Run Pass 1
8      Run Pass 2
9      Accumulate image samples from Pass 2
10     into frame buffer
11   end for

```

---

We can simply run the rendering algorithm multiple times, and accumulate image samples from each iteration into the frame buffer. The result in the frame buffer will gradually converge to the correct value by the nature of the Monte Carlo process. The frame sequence in the animation is split into frame blocks with  $B$  frames. Samples are shared among all frames in the same block, and coherence between blocks is ignored. The total memory footprint is  $B$  frame buffers with some extra bytes per pixel.

## 5 Results and discussion

We first applied our method to render a scene showing the main stadium for the 29th Olympic Games, consisting of 114,000 polygons. The rendered result is shown in Fig. 3. The richness of lighting effects is correctly captured with only 32 samples per pixel (SPP). This result shows the robustness of our method for a scene with complex geometry.

We then applied our method to render a working loader which consists of 59,000 polygons, as shown in Fig. 4. In this scene, the camera and objects are both in motion. By sharing samples between multiple consecutive frames, our method produces a smooth image with only 32 samples per pixel. Plausible shadows are ensured by rechecking visibility.

We also compared our method to image based filtering and irradiance cache methods. All rendered



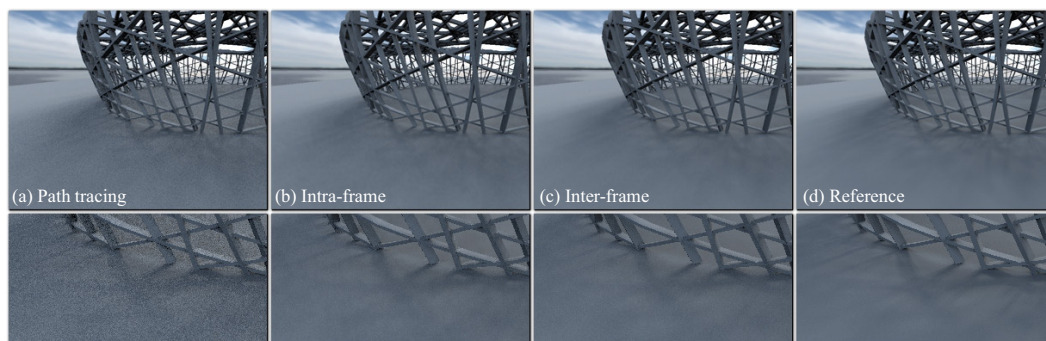
the same scene using 32 samples per pixel. The result is shown in Fig. 5. The image based method can smooth the image by averaging the contribution of nearby pixels, but it also blurs details. Using an irradiance cache smooths the image by reusing irradiance values in object space, but it has low frequency artifacts where the sampling rate is not high enough. Our method borrows and reweighs samples from neighborhoods without introducing blurs and artifacts.

The experiment for direct lighting using an environment map ran on a workstation with two Xeon ®5150 CPUs (dual core, 2.66 GHz) and

3 GB memory. The performance data for all the experimental results are given in Table 1.

## 6 Conclusions

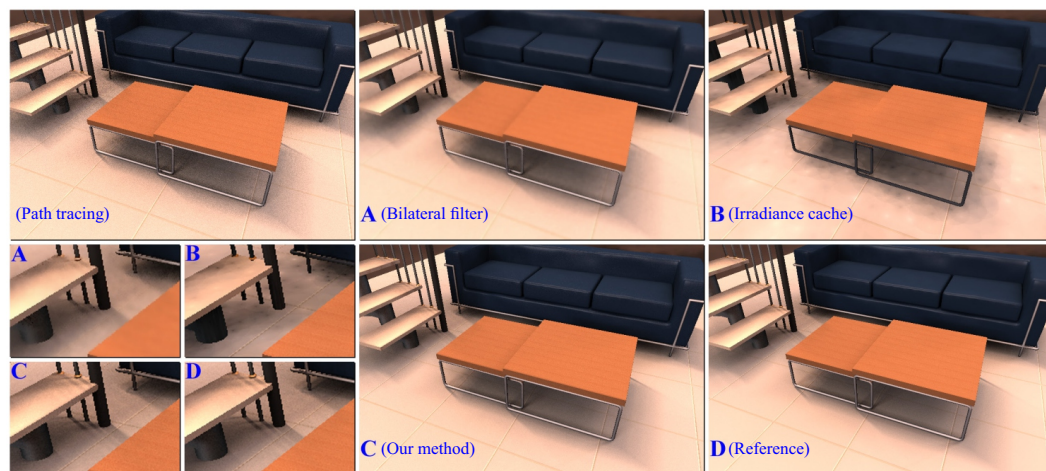
In conclusion, we have proposed a dual cone model to reduce variance for rendering animations. Our method reduces noise by exploiting incident coherence of radiance samples. The samples can be shared not only within one frame, but also between consecutive frames. The sample record must be validated before reuse to ensure correct shadows. To demonstrate its effectivity, we applied our method



**Fig. 3** Rendering results for the steel structure model of a stadium: (a) path tracing result; (b) smoothed result by exploiting intra-frame coherence; (c) smoothed result by exploiting inter-frame coherence; (d) reference image. The bottom line gives close-up detail. Images (a)–(c) were all rendered using 32 samples per pixel.



**Fig. 4** Dynamic scene of a working loader. Four views rendered by our method using 32 samples per pixel.



**Fig. 5** Performance comparison for different methods. Top left: path tracing result. Image A applies bilateral image filtering to the path tracing result. Image B: result using an irradiance cache. Image C: result using our method. Image D: reference.

**Table 1** Performance data

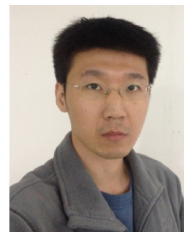
Scene	Triangles	Resolution	SPP	Pass 1	Pass 2	Total time
Indoor (Fig. 5)	19,000	720 × 480	32	11.72 s	56.80 s	70.54 s
Loader (Fig. 4)	59,000	720 × 480	32	13.93 s	53.38 s	68.99 s
Stadium (Fig. 3)	114,000	720 × 480	32	50.43 s	61.04 s	119.64 s
		1024 × 576	32	75.48 s	87.01 s	171.36 s

to animated scenes with both a dynamic camera and objects. Our experimental results show that our method can produce much smoother images than the Monte Carlo method, using very limited samples and memory footprint. In the future, we will combine our work with image based noise reduction methods such as the one proposed by Ref. [4].

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

## References

- [1] Georgiev, I.; Křivánek, J.; Davidovič, T.; Slusallek, P. Light transport simulation with vertex connection and merging. *ACM Transactions on Graphics* Vol. 31, No. 6, Article No. 192, 2012.
- [2] Hachisuka, T.; Pantaleoni, J.; Jensen, H. W. A path space extension for robust light transport simulation. *ACM Transactions on Graphics* Vol. 31, No. 6, Article No. 191, 2012.
- [3] Walter, B.; Khungurn, P.; Bala, K. Bidirectional lightcuts. *ACM Transactions on Graphics* Vol. 31, No. 4, Article No. 59, 2012.
- [4] Sen, P.; Darabi, S. On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Transactions on Graphics* Vol. 31, No. 3, Article No. 18, 2012.
- [5] Ward, G. J.; Rubinstein, F. M.; Clear, R. D. A ray tracing solution for diffuse interreflection. In: *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, 85–92, 1988.
- [6] Schwarzhaupt, J.; Jensen, H. W.; Jarosz, W. Practical Hessian-based error control for irradiance caching. *ACM Transactions on Graphics* Vol. 31, No. 6, Article No. 193, 2012.
- [7] Jensen, H. W. Global illumination using photon maps. In: *Proceedings of the Eurographics Workshop on Rendering Techniques*, 21–30, 1996.
- [8] Hachisuka, T.; Ogaki, S.; Jensen, H. W. Progressive photon mapping. *ACM Transactions on Graphics* Vol. 27, No. 5, Article No. 130, 2008.
- [9] Knaus, C.; Zwicker, M. Progressive photon mapping: A probabilistic approach. *ACM Transactions on Graphics* Vol. 30, No. 3, Article No. 25, 2011.
- [10] Havran, V.; Herzog, R.; Seidel, H.-P. Fast final gathering via reverse photon mapping. *Computer Graphics Forum* Vol. 24, No. 3, 323–332, 2005.
- [11] Wald, I.; Ize, T.; Kensler, A.; Knoll, A.; Parker, S. G. Ray tracing animated scenes using coherent grid traversal. *ACM Transactions on Graphics* Vol. 25, No. 3, 485–493, 2006.
- [12] Kopta, D.; Ize, T.; Spjut, J.; Brunvand, E.; Davis, A.; Kensler, A. Fast, effective BVH updates for animated scenes. In: *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 197–204, 2012.
- [13] Roger, D.; Assarsson, U.; Holzschuch, N. Whitted ray-tracing for dynamic scenes using a ray-space hierarchy on the GPU. In: *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, 99–110, 2007.
- [14] Yoon, S.-E.; Curtis, S.; Manocha, D. Ray tracing dynamic scenes using selective restructuring. In: *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, 73–84, 2007.



system.

**Peng Zhou** is a postdoctoral fellow in the Institute of Software, Chinese Academy of Sciences. He received his Ph.D. degree in computer science from Shandong University in 2012. His research interests include physically based rendering, ray tracing acceleration structure, and creating rendering



**Yanyun Chen** is a professor in the Institute of Software, Chinese Academy of Sciences. He received his Ph.D. degree in January 2000. His research interests include photo-realistic rendering, non-photo-realistic rendering, computer animation, virtual reality, and textures.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.