The Journal of the Society of Art and Science Vol. 11, No. 4, pp. 138 – 148

適応的な自由行程サンプリングによるプログレッシブフォトンビーム法

鈴木 健太郎 1)(非会員)　Yonghao Yue1)(正会員)　西田 友是 1)(正会員)
1) 東京大学

Progressive Photon Beams with Adaptive Free Path Sampling [1]

Kentaro Suzuki1)　Yonghao Yue1)　Tomoyuki Nishita1)
1) The University of Tokyo
{k_suzuki, yonghao, nis} (at) nis-lab.is.s.u-tokyo.ac.jp

概要
煙，水，炎，煙といった関与媒質の写実的なレンダリングはコンピュータグラフィックスにおける重要な研究テーマの一つである．本論文では, 関与媒質をレンダリングするためのプログレッシブフォトンビーム法に基づいた高速で物理ベースな手法を提案する．提案法では，プログレッシブフォトンビーム法の欠点を，アダプティブフリーパスサンプリングの採用と光の減衰関数推定方法の改善による分散の減少によって解決する．提案法は媒質中における光の単一散乱のみならず多重散乱も取り扱うことができ, 複雑な密度分布を持つ媒質や, 複雑な照明環境も取り扱うことができる．また，提案法により得られる解（レンダリング結果）は厳密解に収束する．我々は, GPU を用いて提案法を実装しており, 解を高速に得ることができる．

Abstract
Photo-realistic rendering of participating media, such as steam, water, fire, and smoke, is an important research topic in the computer graphics field. We present a fast and physically based method for rendering participating media based on the progressive photon beams method. For this, we propose a new version using adaptive free path sampling technique and applying a new estimation method for light transmittance functions based on stratified sampling, which overcomes the deficiencies in the progressive photon beams method. Our method is able to take into account not only single but also multiple scattering of light inside the participating medium. Using our method, participating media with complex density distributions can be handled, as well as complex lighting conditions. Additionally, we can obtain images which are guaranteed to converge to exact solutions. Moreover, our method is implemented on a GPU, enabling fast computation of the solutions.

キーワード
レンダリング，関与媒質，自由行程サンプリング，大域照明

Keywords
Rendering, Participating media, Free path sampling, Global illumination

[1] この論文は NICOGRAPH International 2012 に投稿した論文 [13] を芸術科学会論文誌に投稿するものである．

1 Introduction

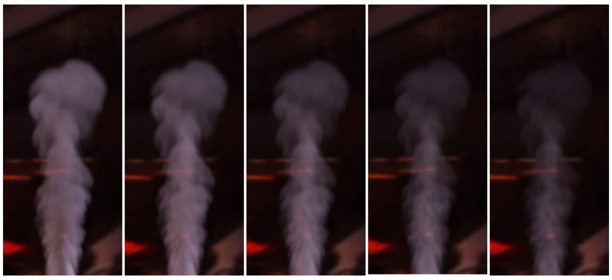


Figure 1: Images of smoke with different density rendered by our method. Each image was rendered in less than 3 minutes.

In the field of computer graphics, photo-realistic rendering of participating media is an important research topic. Examples of participating media are cloud, steam, fire, smoke, etc. Light inside a participating medium will be either absorbed or scattered. The scattered light will go in a direction according to a phase function describing the probability distribution of the scattered direction. These effects are visually important for photo-realistic rendering of participating media, thus need to be taken into account.

Many algorithms have been developed for rendering participating media. Path tracing method and volumetric photon mapping [7] are examples. In particular, the recently proposed progressive photon beams method [6] is a more efficient method than conventional methods for rendering participating media.

The progressive photon beams method is based on a generalization [5] of previous participating rendering methods. In this method, photon beams are emitted from light sources and interact with objects and the participating media. The image is then rendered by estimating the light transported toward the viewpoint from the distribution of photon beams in the scene. However, there is much room for improvement in this method.

First, estimating the scattering locations inside an inhomogeneous medium is slow. The distance between successive scattering locations is called the free path. In the progressive photon beams method, Woodcock tracking [14] is used for this free path sampling. Woodcock tracking is a method for getting statistically unbiased samples of free paths inside an inhomogeneous medium. However it becomes less efficient for more inhomogeneous media. In the progressive photon beams method, rendering inhomogeneous media, such as smoke, cloud, fire, etc. takes a long computation time.

Second, estimating the light transmittance function is also inefficient. The functions are estimated by sampling the free paths many times. If the number of samples is insufficient, the variance becomes large. This variance appears as visual noise in the rendered images which is usually annoying for the viewer, especially, when rendering animations, in which this noise may cause flickering.

In this paper, we present a method for improving the above two deficiencies in the progressive photon beams method. First, to overcome the inefficiency with which it handles inhomogeneous participating media, we utilize the adaptive free path sampling technique [15] instead of Woodcock tracking. Second, to estimate the light transmittance function more efficiently, we present a new sampling technique to reduce the variance inspired by stratified sampling.

With the above improvements, inhomogeneous participating media were rendered an order of magnitude faster compared to using the original progressive photon beams method. Figure 1 shows images rendered using our method.

2 Related Work

We describe previous conventional methods for rendering participating media in Section 2.1, free path sampling in Section 2.2 and estimation of the light transmittance function in Section 2.3.

2.1 Rendering Participating Media

There are and have been many methods used to render participating media. To begin with, participating media were rendered using the method due to T.Nishita using illumination volumes [9]. Improvements to this method led to the rendering of light shafts [10]. In order to obtain more physically correct images, path tracing method based on radiative transfer theory [2] that simulates the rendering equation for participating media [1] were developed. A more efficient method is volumetric photon mapping [7]. These methods have been widely used for rendering participating media.

Beam tracing [3] was the first method that treated light as beams with lengths and thickness. The idea of representing light as beams is used in many methods for rendering participating media. Line space gathering [12] is one such method. In this method, beams are shot from light sources and the intersections between viewing rays from the viewer and the beams are located. At these intersections, the radiance scattered inside media is estimated. Recently, the photon beams method [5] was proposed. This method generalized the previous methods in which photons or beams were shot to estimate the radiance from the media. Later, the photon beams method was improved to give the progressive photon beams method [6] which guarantees convergence to an exact solution. In this paper, we have improved the progressive photon beams method in two areas, firstly. in the free path sampling and secondly, in estimating the light transmittance function. We describe these below.

2.2 Free Path Sampling

The distance between successive scattering locations in a light path is called the free path. Figure 4 illustrates the free path. To calculate the free path, ray marching is typically used. In ray marching, we sample points at a small sampling interval in a scene along the direction of the light ray to detect the location of scattering centers. The accuracy of estimating the free path depends on the sampling interval. In practice this interval cannot be zero, thus, ray marching always implies a statistical bias.

On the other hand, Woodcock tracking [14], which was proposed in nuclear science and introduced to the computer graphics field by Raab [11], enables us to obtain free paths in an unbiased way and render an accurate image. Recently, adaptive free path sampling [15] has been proposed to enable much more efficient computation than Woodcock tracking. In this paper, we examine the efficiency of the adaptive free path sampling method when used with the progressive photon beams method.

2.3 Function Estimation

In the progressive photon beams method, we estimate the light transmittance functions by sampling the free paths. There are many methods used to sample the functions. It is known that for low dimensions, improving the distribution of random samples so that they distribute uniformly would reduce the variance and result in faster convergence. In this paper, we show that the progressive photon beams method can also benefit from this idea by slightly modifying the technique for estimating the light transmittance function, and propose a new sampling technique which helps reduce the noise in the rendered images. In Monte Carlo integration, each pixel on the rendered image is a random variable, therefore it has a variance. In this paper, "noise" is a word represents the variance in the rendered image.

3 The Progressive Photon Beams Method

In this section we describe the progressive photon beams method. For rendering an image, we need to compute the propagation of light emitted from light sources in the scene. The equation used to model this light propagation is called the rendering equation. In this paper, we use a version to render participating media [5].

$$\begin{aligned} L(\mathbf{x} \leftarrow \vec{\omega}) &= T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s) L(\mathbf{x}_s \rightarrow \vec{\omega}) + L_m(\mathbf{x} \leftarrow \vec{\omega}), \\ L_m(\mathbf{x} \leftarrow \vec{\omega}) &= \int_0^s T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t) \sigma_s(\mathbf{x}_t) \\ &\quad (\int_{\Omega_{4\pi}} f_{x_t}(\theta_t) L(\mathbf{x}_t \leftarrow \vec{\omega_t}) d\vec{\omega_t}) dt, \qquad (1) \end{aligned}$$

where $\mathbf{x}_s$ is the nearest location on the surface where a ray from $\mathbf{x}$ to $\vec{\omega}$ intersects the object and $\mathbf{x}_t$ is a location between $\mathbf{x}$ and $\mathbf{x}_s$ with $\mathbf{x}_t = \mathbf{x} + t\vec{\omega}, \mathbf{x}_s = \mathbf{x} + s\vec{\omega}$. s is a parameter between 0 and 1. T_r is the rate of light transmittance between two locations, $L(\mathbf{x} \rightarrow \vec{\omega})$ is the radiance from $\mathbf{x}$ towards $\vec{\omega}$, $L(\mathbf{x} \leftarrow \vec{\omega})$ is the radiance at $\mathbf{x}$ from $\vec{\omega}$, $\sigma_s(\mathbf{x})$ is the scattering coefficient at $\mathbf{x}$, $f(\mathbf{x})$ is a normalized phase function at $\mathbf{x}$, and θ_t is the angle between the incident and outgoing directions at x_t. The phase function $f_x(\theta)$ is the function which determines the ratio of the light scattered at x when the angle is θ. By computing Equation (1), we can render an image taking account of inhomogeneous participating media. However, in general we cannot solve this equation analytically, and need to solve it by numerical calculation, e.g., using Monte Carlo path tracing [8, 1] or volumetric photon mapping [7] derived from photon mapping.

Now, we shall describe the progressive photon beams process in detail. The process used in the progressive photon beams method resembles volumetric photon mapping but where photon beams are used instead of photons. In volumetric photon mapping, we shoot a lot of photons from the light sources, simulate the interactions of photons with objects and the participating medium, and calculate a photon map, which is the distribution of photons in the scene. Once we have the a photon map, we can calculate the radiance to the viewer using this photon information. In the progressive photon beams method, we shoot photon beams from the light sources. Photon beams are represented as beams which have thicknesses and lengths, in contrast to photons which are represented by a geometric point in space. We simulate the transport of the beams in a scene in the same way as volumetric photon mapping, but store photon beams rather than photons. Photon beams can be interpreted as being composed of innumerable photons. In this situation, if the number of shooting times or usage of memory is the same as volumetric photon mapping, the photon density in a scene can be thought of as being higher and so a more efficient estimation can be achieved using photon beams.

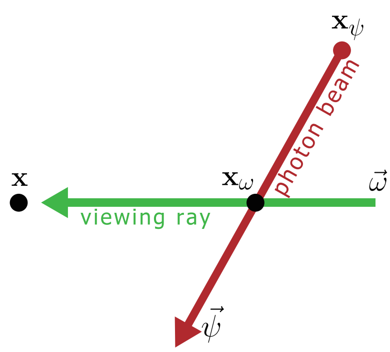


Figure 2: Estimation of radiance with one photon beam. The radius of the beam is r.

The equation below is the rendering equation using a single photon beam. Using this equation, Equation (1) can be solved approximately and it means that we can calculate the effect in the participating media.

$$\begin{aligned} L_m(\mathbf{x} \leftarrow \vec{\omega}) \quad \approx \quad & k_R(r) \sigma_s(\mathbf{x}_\omega) \Phi \\ & T_r(\mathbf{x} \leftrightarrow \mathbf{x}_\omega) T_r(\mathbf{x}_\omega \leftrightarrow \mathbf{x}_\psi) \frac{f(\vec{\omega} \cdot \vec{\psi})}{\sin \theta}, \end{aligned} \qquad (2)$$

where k_R is the kernel function, Φ is the power of the photon beam, and θ is the angle between $\vec{\omega}$ and $\vec{\psi}$. r is the distance between the viewing ray and the photon beams. The kernel function is used to blur the power of single photon beams to obtain more natural images. Figure 2 shows an illustration of this. The red line represents a single photon beam and the green line represents the viewing ray from the viewer at x. The viewing ray and the photon beam are skew lines. x_ω on the viewing ray is the nearest point between the viewing ray and the photon beam.

Progressive photon mapping progressively improves the image. First, we render an image normally. Then we shoot photon beams and render the image again. In this way, we render many images while reducing the radii of the beams and produce the final image by averaging all the rendered images. This progressive process simultaneously reduces both the variance and the bias of the rendered image and produces an exact solution. Reducing the radii according to the equation below guarantees convergence of this algorithm.

$$R_i = R_1 (\prod_{k=1}^{i-1} \frac{k+\alpha}{k}) \frac{1}{i}, \tag{3}$$

where R_i is the radius of the photon beams of the i-th pass, i is the number of the pass and α is a user specified parameter between 0 and 1. We set this parameter to 0.9 and R_1 to 5.0 in this paper. Figure 3 shows images in which the noise is reduced by this progressive processes. Here we give an overview of the progressive photon beams

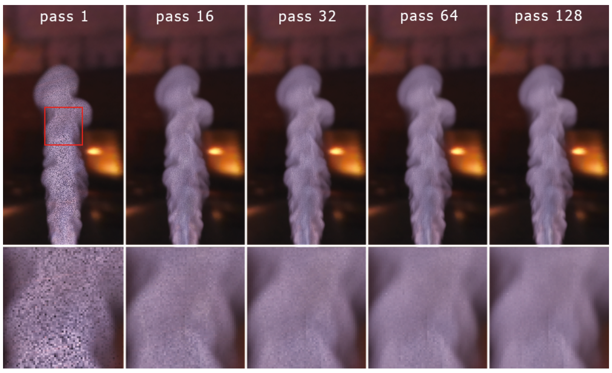


Figure 3: Images of smoke. Each is processed after n times. The resolution is 200x400 and the total rendering time is 130 sec.

method.

1. Shoot photon beams from each light source. Calculate the points of interactions between the beams and the participating medium and shoot new beams from these locations if they are scattered. Beams that are not scattered are absorbed, and no new beams are generated in these cases. To estimate the interaction points, we have to sample the free paths. If the beam intersects an object, reflect the beam according to the Bidirectional Reflectance Distribution Function (BRDF) and shoot the beam again. Store all shot beams.
2. Using the distribution of photon beams calculated in step 1, estimate the radiance and the values of the pixels in the image. Use Equation (2) for this estimation. In the original progressive photon beams method, which is implemented on a GPU, the photon beams are represented as quad polygons and a GLSL fragment shader is applied for this estimation.
3. Reduce the radii of the photon beams according to Equation (3), and repeat step 1. Repeat steps 1 to 3 until the noise and bias in the rendered images becomes sufficiently small, go to step 4.
4. Average the images obtained from step 1 to step 3 to obtain the final image.

In practice, these processes are implemented on GPUs and CPUs. Photon beam shooting and estimating the transmittance functions on a CPU is done in parallel with estimating the radiance on a GPU. This optimization is faster than the version run using only a single CPU.

4 Our Method

Our method acceleralates the progressive photon beams method for rendering inhomogeneous participating media. First, we use the idea of adaptive free path sampling. Second, we change the sampling technique for estimating the light transmittance function along the ray of a viewer.

In this section, we will describe Woodcock tracking for free path sampling (Section 4.1) and adaptive free path sampling to improve the progressive photon beams method (Section 4.2). Then, we describe our new technique for estimating light transmittance functions in detail (Section 4.3).

4.1 Free Path Sampling

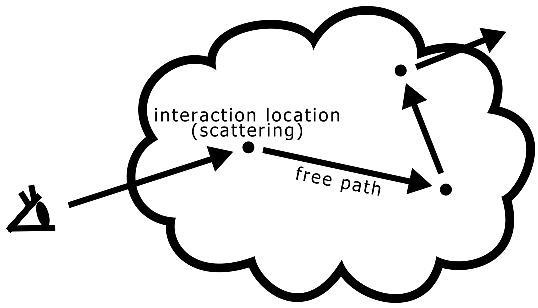


Figure 4: The free path of light from the eye through participating media.

Free path sampling is a core technique in rendering participating media, and it usually determines the overall performance. The method for sampling the free path in inhomogeneous participating media is different to that in homogeneous media. First, we describe the case for homogeneous media.

In homogeneous participating media, the free path can be sampled using the equation below [4].

$$d = -\frac{\log \xi}{\sigma_t}, \tag{4}$$

where d is the length of the free path, ξ is a random number between 0 and 1, and σ_t is the extinction coefficient of the media. By sampling the free path using this equation, the results of various algorithms, such as path tracing and volumetric photon map converge to exact solutions. In inhomogeneous participating media, however, the problem is much more difficult.

The conventional method for free path sampling of inhomogeneous participating media is ray marching. In ray marching, we sample many points at small intervals along the ray in the media and at each point, determine the light transport stochastically according to the extinction coefficient and scattering coefficient at each point. The accuracy of ray marching depends on the interval. Since this cannot be zero, the result is usually biased and the rendered image is inaccurate. Woodcock tracking [14] resolves this accuracy problem. Using this method, a statistically unbiased free path can be sampled and a physically correct image can be obtained. We show this algorithm in Algorithm 1 [15].

Algorithm 1 Woodcock tracking $(\mathbf{x_0}, \vec{\omega}, k_M, d_{min}, d_{max})$

Input: $\mathbf{x_0}, \vec{\omega}$: The ray starting at $\mathbf{x_0}$ in direction $\vec{\omega}$.
k_M : The majorant extinction coefficient.
$(d_{min}, d_{max}]$: The interval of the ray to evaluate.
Output: The free path d to the next scattering event.

$d \leftarrow d_{min} - \frac{\ln(1-rand())}{k_M}$
while $d \le d_{max} \wedge \frac{k(\mathbf{x_0}+d\vec{\omega})}{k_M} < rand()$ **do**
  $d \leftarrow d - \frac{\ln(1-rand())}{k_M}$
end while
return d

Woodcock tracking is an innovative free path sampling method because it has statistically unbiased characteristics compared to ray marching. However the efficiency of this algorithm depends on the majorant extinction coefficient of the media, which is the highest extinction coefficient in the subject space. Since the progression is related to the reciprocal of the majorant extinction coefficient and in the case of inhomogeneous participating media, the computation time may increase drastically even when only a small part of the media has high extinction coefficient. The original progressive photon beams method uses Woodcock tracking and so rendering of inhomogeneous media is slow.

4.2 Adaptive Free Path Sampling

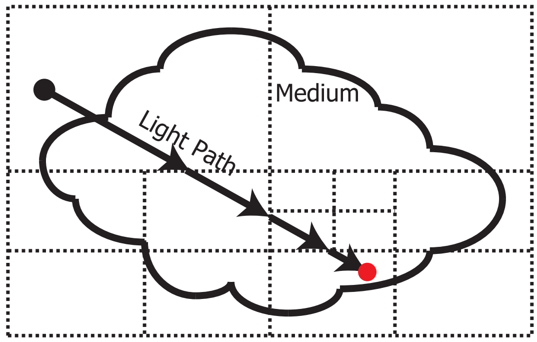


Figure 5: In adaptive free path sampling, Woodcock tracking is processed in each partitioned subregion. The red point is a scattering location.

Adaptive free path sampling [15] is an improved free path sampling method derived from Woodcock tracking. It has been proven that the adaptive free path sampling technique produces a stochastically unbiased estimation of the free path (i.e., accurate in the sense of stochastic sampling). In this method, the subject space is divided into subregions with the density of the participating media in the subregions begin uniform. In the original paper [15], this splitting is done by the largest empty rectangle problem. When sampling the free path in the participating media, we do this by Woodcock tracking in each subregion and advance between the subregions. Figure 5 illustrates this process. Dividing the space reduces the majorant extinction coefficient in each subregion compared to the majorant extinction coefficient in the total space and so the computation time of each subregion is reduced by this division. Thus, the total computing time is also reduced. Compared to Woodcock tracking, the adaptive free path sampling technique is much more efficient for highly inhomogeneous participating media. To implement this algorithm, the data structure for storing the subregions is important. There are algorithms that use various data structures [16]. In this paper we use an octree.

Originally, adaptive free sampling was implemented for path tracing. In this paper, we examine its efficiency when used together with the progressive photon beams method. It is expected that adaptive free path sampling will benefit the progressive photon beams method, because computing the free path is fundamental in this method. Moreover, we modified the adaptive free path sampling technique for a parallel optimized version of the method. In the progressive photon beams method, free path sampling is used mainly for shooting photon beams and estimating transmittance functions. For parallelization, we assigned additional buffers for the random numbers used in adaptive free path sampling and divided the tasks into many small tasks to simultaneously uses as many CPU cores as possible. With this modification, we succeeded in applying adaptive free path sampling to the progressive photon beams method.

4.3 Estimation of the Transmittance Function

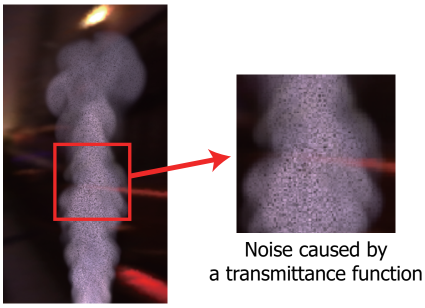


Figure 6: Example of noise caused by estimating the light transmittance function.

Images obtained using the progressive photon beams method converge to an exact solution by the progressive process, but if the number of passes is insufficient or the density of the participating media is too high, the variance, and thus the noise, of the rendered image becomes too large. To reduce this noise, we improve the estimation of the light transmittance function. Figure 6 shows an example of this noise. The main source of the noise is the light transmittance function, which is Tr in Equations (1) and (2). The transmittance function gives the ratio of the transmitted light depending on the distance travelled. In the progressive photon beams method, we can estimate the transmittance function by sampling the free paths along a viewing ray in the participating media. Figure 7 illustrates the sampling method for this estimation. Because all of the rendered image is averaged in the final step, this free path is sampled many times over. Even if the sampling number per pass is small (this lack of samples introduces a large error when estimating the light transmittance function), convergence to an exact solution is guaranteed. This method is called a progressive deep shadow map [6]. However, when the total number of samples is insufficient, using this estimated function produces noise in the rendered image. We have improved this estimation by using a method based on stratified sampling.

The number of samples per pass is four. At most it was sixteen in the original progressive photon beams method. Increasing this number increases the accuracy with which the function is estimated and reduces the noise faster, however, the progressive photon beams method needs to be implemented on a GPU for high optimization, and the GPU specification limits this sampling number to between four and sixteen. Repeated sampling of the free path to estimate the transmittance function means random sampling for this function. In this paper, we modify this random sampling method to a method derived from stratified sampling. Stratified sampling is a type of sampling

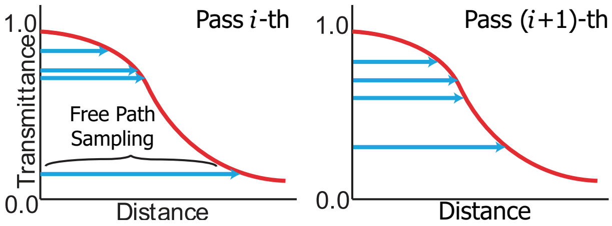


Figure 7: Sampling to estimate the transmittance function in the progressive photon beams method. In each pass, we sample some free paths to estimate the transmittance function. In this figure, four samples are used per pass.

method. In this method, the subject area is split into subareas and random sampling is carried out in each subarea. This method reduces the variance compared to random sampling over the whole area. We developed a new method for reducing the variance in the estimated transmittance function using this idea from stratified sampling.

First, we sample the free paths with N passes at one time and divide these paths into four groups. Next, we choose free paths from each group at random and use these four chosen paths for estimating the light transmittance function. We do not use free paths which have already been selected. Before the N passes have been completed, we use pre-sampled paths for the estimate and after the N passes have been completed, we sample the free paths with N passes again.

Figure 8 shows a concrete example. In this example, we sample eight free paths at one time. In this case, we divide the eight samples into four groups and each group has two samples. After sampling, we use four samples from each group at pass i and another four samples at pass $i+1$. In this way, the variance is less and the accuracy is greater than simple random sampling. This stratified sampling based approach does not increase the total computation time and reduces the variance in estimating the transmittance function. As a result, the noise in the participating media of the rendered image is decreased.

5 Results

We implemented our method in C++ and used Microsoft Visual C++ 2010 for the compiler. For optimization of the progressive photon beams method, we used both GPUs and CPUs. All images were rendered on a machine with a 6-core 3.33GHz Intel Corei7 980 CPU and nVidia GeForce GTX580 GPU. The total main memory was 12 GB and our renderer used between 0.5 and 1.5 GB.

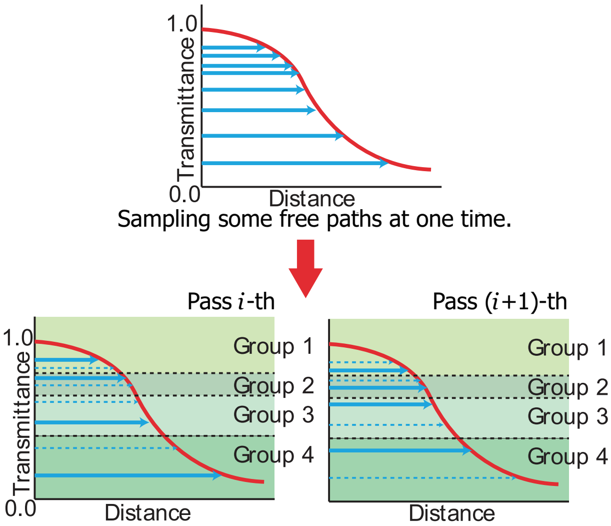


Figure 8: Illustration of sampling to estimate the transmittance function in our method. Sampling paths at one time and resampling of the paths in each pass.

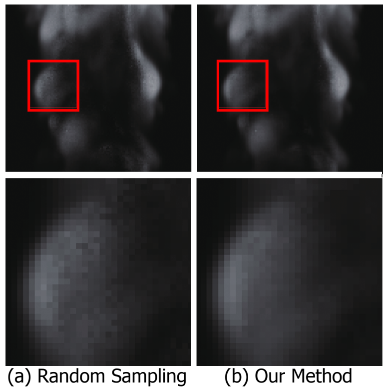


Figure 9: Comparison of the noise reduction between the original progressive photon beams method and our proposed method. (a) is rendered using random sampling and (b) using our method based on stratified sampling. In our method, we sampled free paths with 64 passes at one time. Both images are rendered using 128 passes. The bottom images show detailed views of the parts in red squares in the top images.

Some experiments were carried out. In Section 5.1, the noise achieved by estimating the transmittance functions using our proposed method and new sampling technique is compared with that using the original progressive photon beams method. In Section 5.2, we compare the reduction in noise in some additional situations. In Section 5.3, we show some rendered images using our proposed method and compare the computation time.

5.1 Reduction in Noise by Estimating the Transmittance Function with Adaptive Free Path Sampling

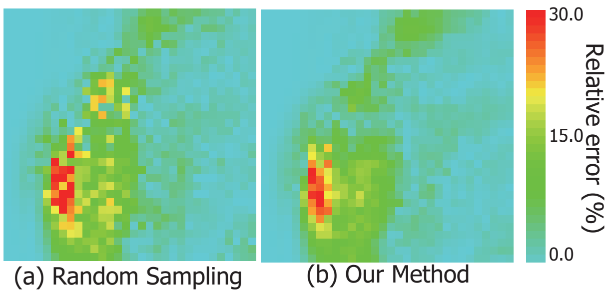


Figure 10: Visualization of the relative error. (a) is using random sampling and (b) is using our method. These images correspond to the regions shown by red squares in Figure 9.

Figure 9 shows an example of the effect of our noise reduction method for estimating transmittance functions based on stratified sampling. When the computation time is the same, using our method (Figure 9 (b)) results in an image with reduced noise compared to using random sampling (Figure 9 (a)). Figure 10 is visualizes the relative error compared to a reference image of the detailed views in Figure 9. Figure 10 (a) has a larger error than Figure 10 (b). The graph in Figure 11 shows the relative error in the area within the red square with the increasing number of passes. This graph shows that when using our technique, there is less error for the same number of passes.

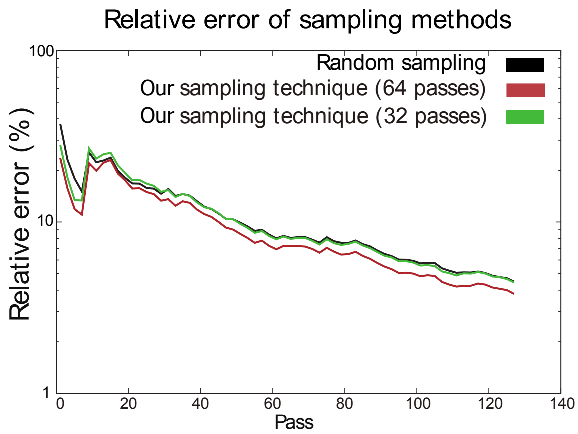


Figure 11: Graph of relative error. The horizontal axis shows the number of progressive passes and the vertical axis shows relative error in the area indicated by the red squares in Figure 9. The red and green lines show the relative errors when using our stratified sampling technique in which 64 and 32 passes are sampled at one time.

5.2 Adaptive Free Path Sampling and the New Transmittance Function Estimation

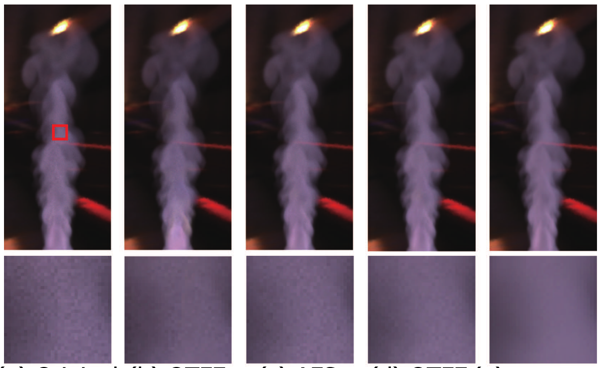


Figure 12: Comparison between images with different rendering methods. All images were rendered in 30 seconds. (a) was rendered by the original progressive photon beams method. (b) was rendered by the progressive photon beams method with our new transmittance function estimation(OTFE). (c) was rendered with adaptive free path sampling (AFS). (d) was rendered with OTFE and AFS. The images at the bottom correspond to the regions in red squares in the top images.

In this section, we compare the results from different rendering methods. These are, using only the original progressive photon beams method(PPB), PPB with our new transmittance function estimation(OTFE), PPB with adaptive free path sampling(AFS), and PPB with both new methods(OTFE+AFS). We rendered the low density smoke images in the same time, 30 seconds. Figure 12 shows the results. The relative error of PPB, AFS and OTFE is between 1.8% and 7.6% but the relative error of OTFE+AFS is 1.4% and so using OTFE+AFS is the best quality (The relative errors are with respect to the reference image).

5.3 Rendered Images

We have used progressive photon mapping to render all surfaces because we wanted to guarantee that the rendered images converged to exact solutions. Progressive photon mapping is compatible with our proposed method based on the progressive photon beams method.

Table 1 shows the computation time and the speedup for the images in Figures 13, 14, 15 and 16. All the rendered images are of the same quality. The total computation time is, of course, important but in the progressive photon beams method, the free path sampling causes a bottleneck, since it takes between 70% and 95% of the total computation time. In this paper, we have mainly improved this bottleneck due to the free path sampling, so we compare the free path sampling time in the rows labeled FPS in Table 1, in order to clearly show the effect of our improvements.

Table 1: Rendering times for different scenes. Total is the total computation time and FPS is the time for free path sampling. PPB shows the computation time in seconds for the progressive photon beams method and OM shows for our method (PPB with our new transmittance function estimation and adaptive free path sampling). Ratio shows the speedup when using our method.

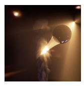
		PPB [sec]	OM [sec]	Ratio
	Total	280	125	2.2
	FPS	220	65	**3.4**
Fig. 13 (a)	Other	60	60	1.0
	Total	1718	131	13.1
	FPS	1659	72	**23.0**
Fig. 13 (b)	Other	59	59	1.0
	Total	1707	501	3.4
	FPS	1447	239	**6.1**
Fig. 14	Other	260	262	0.99
	Total	219	127	1.7
	FPS	159	67	**2.4**
Fig. 15	Other	60	60	1.0
	Total	2243	262	8.6
	FPS	2161	181	**11.9**
Fig. 16	Other	82	81	1.0

Figure 13 is an image of smoke with lower and higher density. The total number of progressive passes is 128 and the number of emitted photon beams is 100,000 per pass. Adaptive free path sampling is suitable for inhomogeneous media. If the density is high and the distribution is nonuniform, it is faster than using Woodcock tracking. When the density distribution is nearly uniform and adaptive free path sampling becomes less efficient, it still outperforms Woodcock tracking in our experimental scenes. In this experiment, low density smoke is almost uniformly low density, and so the speedup is lower; 2.2 in total and 3.4 for free path sampling. On the other hand, with high density smoke the speedup is greater, 13.1 in total and 23.0 for free path sampling.

Figure 14 is a smoke scene with a glass sphere. The image resolution is 512x512 and the total number of progressive passes is 256 and the number of emitted beams

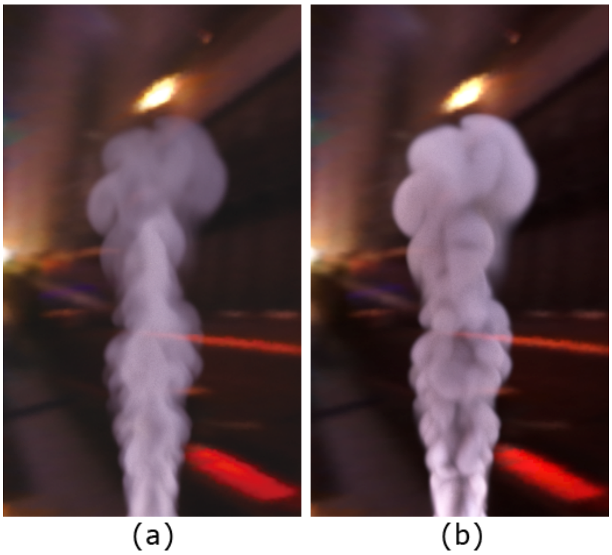


Figure 13: (a) Smoke with lower density. (b) Smoke with higher density.

is 200,000 per pass. In this scene we place a glass sphere, and the scene is filled by a homogeneous participating medium and contains inhomogeneous smoke. This scene is similar to Figure 13, and so for this scene our method is very efficient. The speedup of the total time is 3.4 and that for the free path sampling is 6.1.

Figures 15 and 16 are cloud images. The total number of pass is 128 and the number of emitted beams is 100,000 per pass. Since the cloud density is nonuniform, rendering clouds is suitable for adaptive free path sampling and our adaptive method. Figures 15 and 16 show a simple fractal cloud and a cirrocumulus. In the fractal cloud scene, the speedup for free path sampling is 2.4 and in the cirrocumulus scene the speedup of the total time is 8.6 and that of the free path sampling is 11.9. So, both these scenes are rendered much faster.

Figures 17 and 18 are examples of images that can be rendered efficiently using our method. Figure 17 is a scene in which we locate glass objects above a lambert surface as a floor. Light sources are located above these objects and their colors are red, green and blue. This scene is filled by gas generated with a random noise function. We can see caustics not only on the participating medium but also on the floor. The total rendering time is 500 minutes, and the participating medium was rendered in less than 5 minutes. Figure 18 is a cloud scene with shafts of light passing through gaps in the clouds. The total rendering time is 200 seconds.

6 Conclusion and Future Work

In this paper we overcame the deficiencies in the progressive photon beams method. In our method, we used parallelized adaptive free path sampling and improved the

Figure 14: Smoke with a glass sphere. The image resolution is 512x512.

Figure 15: Single fractal cloud. The image resolution is 512x256

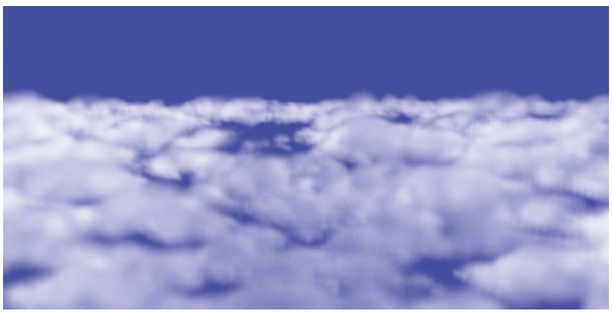


Figure 16: A cirrocumulus. The image resolution is 512x256

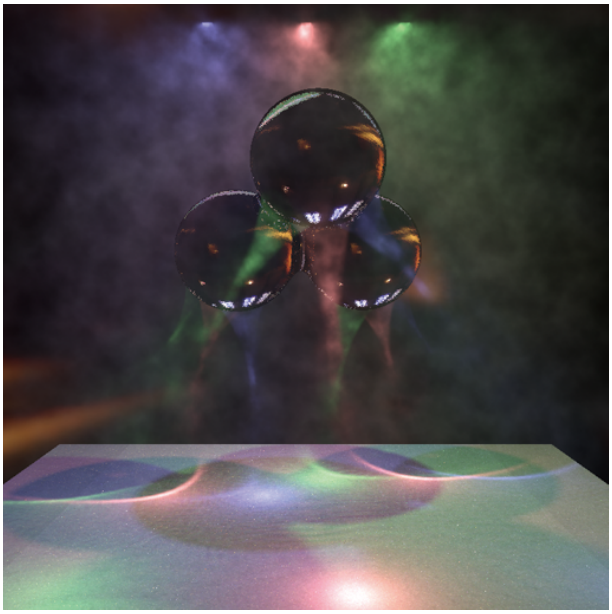


Figure 17: Glass objects and floor.

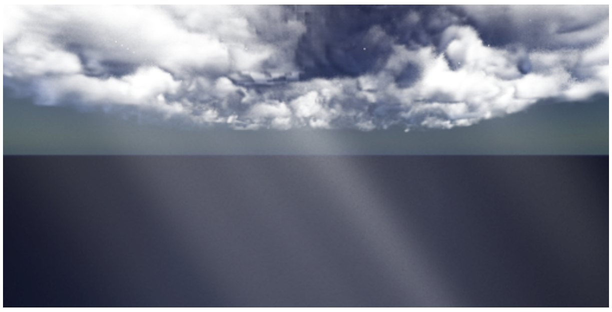


Figure 18: Shafts of light passing through gaps in the clouds.

Figure 19: Single fractal cloud at sunset.

estimation of the light transmittance functions. Through these improvements, we succeeded in developing a new version of the progressive photon beams method that can efficiently handle not only homogeneous media but also inhomogeneous media.

The user specified parameter (α in Equation (3)) used in the progressive photon beams method and progressive photon mapping was left constant in this paper, but it could be determined adaptively. This parameter is related to the speed of convergence of rendering, and so if it could be determined depending on the scene, the speed of convergence would be faster than our method. In particular, the parameters in the progressive photon beams method have not yet been well researched, so there is room for optimization here.

References

[1] J. Arvo. Transfer equations in global illumination. In *Global Illumination, SIGGRAPH '93 Course Notes*, 1993.

[2] S. Chandrasekhar. *Radiative Transfer*. Dover Publications, 1960.

[3] P. S. Heckbert and P. Hanrahan. Beam tracing polygonal objects. *SIGGRAPH Comput. Graph. (Proc. of SIGGRAPH '84)*, 18(3):119–127, 1984.

[4] W. J. Henrik. *Realistic Image Synthesis Using Photon Mapping, 2nd Edition*. A K Peters/CRC Press, 2nd revised edition edition, 2001.

[5] W. Jarosz, D. Nowrouzezahrai, I. Sadeghi, and H. W. Jensen. A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Trans. Graph.*, 30(1):5:1–5:19, 2011.

[6] W. Jarosz, D. Nowrouzezahrai, R. Thomas, P. Sloan, and M. Zwicker. Progressive photon beams. *ACM Trans. Graph. (Proc. of SIGGRAPH ASIA 2011)*, 30(6):181:1–181:12, 2011.

[7] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scences with participating media using photon maps. In *Proc. of SIGGRAPH '98*, SIGGRAPH '98, pages 311–320. ACM, 1998.

[8] J. T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph. (Proc. of SIGGRAPH '86)*, 20(4):143–150, 1986.

[9] T. Nishita, Y. Miyawaki, and E. Nakamae. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. *SIGGRAPH Comput. Graph. (Proc. of SIGGRAPH '87)*, 21(4):303–310, 1987.

[10] T. Nishita and E. Nakamae. Method of displaying optical effects within water using accumulation buffer. SIGGRAPH '94, pages 373–379, 1994.

[11] M. Raab, D. Seibert, and A. Keller. Unbiased global illumination with participating media. In *Monte Carlo and Quasi-Monte Carlo Methods*, pages 591–605, 2006.

[12] X. Sun, K. Zhou, S. Lin, and B. Guo. Line space gathering for single scattering in large scenes. *ACM Trans. Graph. (Proc. of SIGGRAPH 2010)*, 29(4):54:1–54:8, 2010.

[13] K. Suzuki, Y. Yue, and T. Nishita. Progressive photon beams with adaptive free path sampling. In *NICOGRAPH International 2012*, pages 83–93, 2012.

[14] E. Woodcock, T. Murphy, P. Hemmings, and T. Longworth. Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. In *Conference on the Application of Computing Methods to Reactor Problems (ANL-7050)*, pages 557–579, 1965.

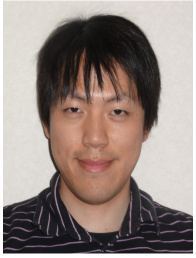
[15] T. Yue, K. Iwasaki, B. Chen, Y. Dobashi, and T. Nishita. Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Trans. Graph. (Proc. of SIGGRAPH ASIA 2010)*, 29(6):177:1–177:8, 2010.

[16] T. Yue, K. Iwasaki, B. Chen, Y. Dobashi, and T. Nishita. Toward optimal space partitioning for unbiased, adaptive free path sampling of inhomogeneous participating media. *Comput. Graph. Forum (Proc. of Pacific Graphics 2011)*, 30(7):1911–1919, 2011.

Kentaro Suzuki

Kentaro Suzuki is a graduate student at the Department of Computer Science, Graduate School of Information Science and Technology, the University of Tokyo, Japan, since 2012. His research interests center in computer graphics, including global illumination. He received his B.S. degrees in Information Science from the University of Tokyo, Japan, in 2012.

Yonghao Yue



Yonghao Yue is an assistant professor at the Department of Complexity Science and Engineering, Graduate School of Frontier Sciences, the University of Tokyo, Japan, since 2011. He is also an assistant professor at the Department of Information Science, School of Science, the University of Tokyo, Japan, since 2012. His research interests center in computer graphics, including realistic image synthesis, global illumination, Monte Carlo techniques and physically-based simulations. He received his B.S., M.S. and Ph.D. degrees in Computer Science from the University of Tokyo, Japan, in 2005, 2007 and 2011, respectively. He is a member of ACM, IIEEJ, IPSJ and ITE.

Tomoyuki Nishita

Tomoyuki Nishita is a professor in the Department of Complexity Science and Engineering (also in the Department of Information Science) at the University of Tokyo, Japan since 1998. He received his BE, ME and Ph.D in Engineering in 1971, 1973, and 1985, respectively, from Hiroshima University. He taught at Fukuyama University from 1979 to 1998 He was an associate researcher in the Engineering Computer Graphics Laboratory at Brigham Young University from 1988 to 1989. His research interests center in CG including lighting/shading models (radiosity), natural phenomena, real-time rendering, geometric modeling, and non-photorealistic rendering. He is one of the pioneers of radiosity method. Dr. Nishita received Research Award on Computer Graphics from Information Processing Society of Japan in 1987, and also received Steaven A. Coons award from ACM SIGGRAPH in 2005. He was a president of The Institute of Image Electronics Engineers of Japan in 2009-2010.