

# IMPORTANCE CACHING FOR HOMOGENEOUS PARTICIPATING MEDIA

Ryosuke Enotani  
Wakayama University

Shinya Yasuaki  
Wakayama University

Kei Iwasaki  
Wakayama University/UEI Research

Yoshinori Dobashi  
Hokkaido University/UEI Research

Tomoyuki Nishita  
UEI Research/Hiroshima Shudo University

## ABSTRACT

Many-light rendering methods that simplify complex illumination calculation into a simple sum of the illumination from virtual point lights (VPLs), have attracted attention. Although the accuracy of many-light rendering can be enhanced by using a large number of VPLs, this significantly increases the computation time. To address this problem, we propose a cache-based efficient VPL sampling method to conduct the many-light rendering of participating media. Our method distributes cache points over the scene, calculates the contributions from all VPLs, and constructs a probability function that is perfectly proportional to the contributions from all VPLs. Using a probability function interpolated from those recorded at nearby cache points, our method applies an importance sampling process to the VPLs. An experimental evaluation demonstrates that the proposed method achieves 50 times speedup compared to the method of summing all VPL contributions.

## 1. INTRODUCTION

Rendering homogeneous participating media such as fog, mist, and haze has been a challenging problem in the field of Computer Graphics. To render realistic images of scenes including participating media, both the reflected radiance from surfaces of the scene and the scattered radiance along the viewing ray are computed. In this paper, we refer to the point where the reflected radiance or the scattered radiance is calculated as *shading point*. Multiple scattering of light plays an important role to render realistic images of participating media, and several methods, such as volumetric photon mapping [1] and Metropolis light transport [2], have been proposed. However, simulating multiple scattering of light is computational expensive, and this makes it difficult to render scenes including participating media efficiently.

To render participating media with multiple scattering of light, Raab et al. [3] introduced an instant radiosity method [4]. Many-light rendering, which is the extension of instant radiosity, approximates the multiple scattered light with virtual point lights (VPLs) [5] and variants [6, 7].

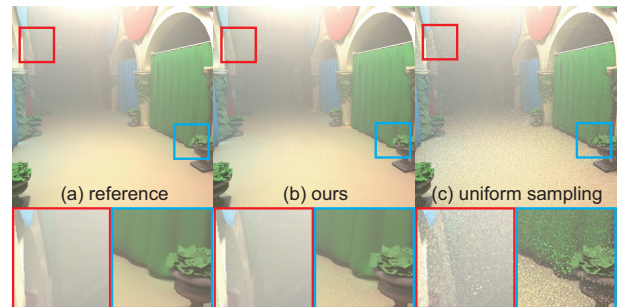


Figure 1: Rendering results of crytek sponza filled with homogeneous participating media. (a) reference image (367.32min) rendered by summing up all VPL contributions. Equal-time comparison between (b) our method (6.90min) and (c) uniform sampling (6.93min). Close-up images for the reference, our method, and uniform sampling are shown in the bottom.

The accuracy of many-light rendering increases with respect to the increase in the number of VPLs. As these methods [3, 5, 6, 7] accumulate contributions from all virtual lights, they also have extensive computation times. To accelerate many-light rendering, acceleration methods by clustering VPLs [8] and variants [9] have been proposed. However, these methods may suffer from errors due to clustering. Rather than clustering the VPLs, importance sampling methods [11, 12] have been proposed. These importance sampling methods, however, do not handle participating media.

To address this problem, we propose an importance sampling method of VPLs using cache for homogeneous participating media with isotropic scattering. Cache points are distributed both on the surfaces of the scene and inside participating media. Then, contributions from all VPLs are recorded at each cache point, and the probability function which is perfectly proportional to the contributions from all VPLs are constructed. The probability function to sample VPLs at each shading point is interpolated by those stored at nearby cache points. As shown in Fig. 1, our results demonstrate that our method can render images an order of magnitude faster than the reference solution, and our method can produce better-quality images than uniform sampling in an

equal-time comparison.

## 2. RELATED WORK

As the rendering of participating media has been extensively researched, we only review previous methods related to both participating media and many-light rendering. For further details on either approach, readers are referred to thorough surveys on participating media [13] and many-light rendering [14].

Raab et al. [3] proposed a rendering method for participating media using instant radiosity [4], and Engelhardt et al. [5] developed a many-light rendering framework for participating media using VPLs. These methods calculate the scattered radiance at each shading point on the viewing ray by summing all VPL contributions, which results in excessive computation times. Novak et al. [6] introduced virtual ray lights (VRLs), which represent virtual light with line segments, to alleviate the singularities that arise from VPL rendering, and extended VRLs to virtual beam lights (VBLs) to further eliminate singularities [7]. As these methods accumulate contributions from all virtual lights, they also have extensive computation times.

To accelerate many-light rendering, Walter et al. [8] proposed the concept of multidimensional lightcuts by clustering the VPLs. Frederickx et al. clustered VRLs to accelerate the rendering of participating media [9] based on Light-Slice [10]. However, these methods may suffer from errors due to clustering.

Thus, rather than clustering the VPLs, importance sampling methods [15, 16] have been proposed. These importance sampling methods, however, cannot handle participating media. To increase the rendering efficiency, several cache-based methods (e.g. [17]) have been developed. The main focus of these approaches is to accelerate the radiance calculations for surfaces.

To render participating media efficiently using a cache, Jarosz et al. [18] proposed a radiance caching method whereby the inscattered radiance is calculated by directly extrapolating the radiances stored at the cache points. Since this method is orthogonal to our method, it would be possible to incorporate the radiance caching method into our method for further acceleration.

Georgiev et al. proposed an importance caching method [11] that samples VPLs based on their contributions at nearby cache points, and Yoshida et al. [12] developed an adaptive importance caching method that inserts additional cache points where there are large variations in VPL contributions. However, none of these methods can handle participating media.

Our method extends the importance caching methods [11, 12] to render homogeneous participating media. Our technical contribution is the generalization of the importance

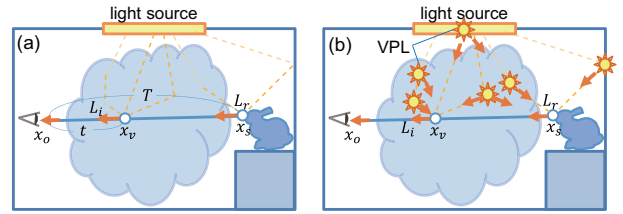


Figure 2: (a) Light transport in participating media. Direct illumination from the light source and indirect illumination (multiple scattered light and multiple reflected light) are incident on shading points  $x_v$  and  $x_s$ . (b) Many-light rendering methods represent direct and indirect illumination with VPLs.

caching method to volumetric scattering. By distributing cache points within participating media and constructing probability distribution functions proportional to the contributions of VPLs, our method can render satisfactory results compared to the reference solution.

## 3. VPL RENDERING OF PARTICIPATING MEDIA

In this section, we first review the rendering equation for participating media, and then describe the radiance calculation using VPLs. In the following, we denote the participating media and the surface in the scene as  $\mathcal{V}$  and  $\mathcal{A}$ , respectively. The radiance  $L(x_o, \omega_o)$  that reaches viewpoint  $x_o$  along viewing direction  $\omega_o$  in a scene including participating media (see Fig. 2) is calculated as:

$$L(x_o, \omega_o) = \tau(x_o, x_s)L_r(x_s, \omega_o) + \int_0^T \tau(x_o, x_v)L_i(x_v, \omega_o)dt, \quad (1)$$

where  $x_s$  is the intersection point between the viewing ray and the surface in the scene,  $\tau(x, x')$  is the transmittance between  $x$  and  $x'$ ,  $L_r(x_s, \omega_o)$  is the reflected radiance at  $x_s \in \mathcal{A}$ ,  $T$  is the distance between  $x_o$  and  $x_s$ ,  $L_i(x_v, \omega_o)$  is the inscattered radiance inside the participating media, and  $x_v = x_o - t\omega_o$ , where  $t$  is the distance from  $x_o$  to  $x_v$ . To compute Eq. (1), we require the reflected radiance  $L_r$  at  $x_s$  and the inscattered radiance  $L_i$  at  $x_v$ . In the following,  $x_s$  is referred to as the *surface shading point*, and  $x_v$  is the *volume shading point*. Both  $L_r$  and  $L_i$  can be calculated in a unified way as:

$$L(x, \omega_o) = \int_{\mathcal{A} \cup \mathcal{V}} L(y, x)f(y, x, x_o)G(x, y)V(x, y)d\mu(y), \quad (2)$$

where  $L(y, x)$  is the radiance from  $y$  to  $x$ ,  $V(x, y)$  is the visibility term that returns 1 if  $x$  and  $y$  are mutually visible and 0 otherwise,  $f$  is the generalized scattering function that

returns the phase function  $f_p$  if  $x \in \mathcal{V}$  or the surface BRDF  $f_r$  if  $x \in \mathcal{A}$ , and  $G(x, y)$  is the generalized geometry term given by  $G(x, y) = \frac{D_x(y)D_y(x)\sigma(x)\tau(x,y)}{\|x-y\|^2}$ , where  $D_x(y)$  returns 1 if  $x$  is in the volume while  $D_x(y)$  returns  $n_x \cdot \omega_{xy}$  if  $x$  is on the surface where  $n_x$  is the normal at  $x$  and  $\omega_{xy}$  is the unit vector from  $x$  to  $y$ ;  $D_y(x)$  is defined in a similar way.  $\sigma(x)$  returns the scattering coefficient  $\sigma_s(x)$  if  $x \in \mathcal{V}$ , and 1 otherwise.  $d\mu(x)$  is the differential measure that represents  $d\mathcal{V}(x)$  if  $x$  is in the volume, or represents  $d\mathcal{A}(x)$  if  $x$  is on the surface. As shown in Fig. 2(b), many-light rendering approximates the radiance  $L(y, x)$  in Eq. (2) using VPLs as the following equation:

$$L(x, \omega_o) \approx \sum_{i=1}^{N_{vpl}} L(y_i, x) f(y_i, x, x_o) G(x, y_i) V(x, y_i), \quad (3)$$

where  $N_{vpl}$  is the number of VPLs and  $y_i$  is the  $i$ -th VPL. To render the participating media faithfully, a large number of VPLs are required. In this case, summing  $L(y_i, x) f(y_i, x, x_o) G(x, y_i) V(x, y_i)$  for all VPLs is computationally expensive. Rather than summing all VPL contributions,  $L(x, \omega_o)$  can be estimated using importance sampling as follows:

$$L(x, \omega_o) \approx \frac{1}{N} \sum_{n=1}^N \frac{L(y_n, x) f(y_n, x, x_o) G(x, y_n) V(x, y_n)}{p(y_n)}, \quad (4)$$

where  $N$  is the number of samples,  $y_n$  is the  $n$ -th sampled VPL, and  $p$  is the probability function. VPL sampling incurs a variance  $\text{Var}[L(x, \omega_o)]$ , which is calculated as:

$$\text{Var}[L(x, \omega_o)] = \frac{1}{N} \sum_{n=1}^N \left( \frac{L \cdot f \cdot G \cdot V}{p} - \bar{L} \right)^2 p, \quad (5)$$

where  $\bar{L}$  is the true value of  $L(x, \omega_o)$  (we have omitted the function arguments for brevity). As shown in Eq. (5), the variance  $\text{Var}[L(x, \omega_o)]$  decreases when  $p$  is close to  $\frac{L \cdot f \cdot G \cdot V}{\bar{L}}$ . However, constructing  $p$  close to  $\frac{L \cdot f \cdot G \cdot V}{\bar{L}}$  is difficult, because computing  $\bar{L}$  involves summing all VPL contributions.

#### 4. PROPOSED METHOD

The proposed method consists of a preprocessing step and a rendering process. Fig. 3 shows the pipeline of our method, in which a probability function  $p$  is constructed to be as proportional as possible to the VPL contribution  $L \cdot f \cdot G \cdot V$  by using a cache. Our method distributes cache points  $c$  both on the surfaces in the scene and inside participating media. Cache points on the surfaces are referred to as *surface cache points*  $c_s \in \mathcal{A}$ , and those inside volumes are

called *volume cache points*  $c_v \in \mathcal{V}$ . All VPL contributions  $L(y_i, c) f(y_i, c, x_o) G(y_i, c) V(y_i, c)$  are calculated at each cache point  $c$ . Therefore, the constructed probability function is perfectly proportional to the VPL contributions. The probability function at each shading point is then interpolated according to the probability functions stored at nearby cache points, which are expected to be proportional to the VPL contributions, resulting in less variance.

#### 4.1. PREPROCESSING

In the preprocessing stage, our method generates VPLs, shading points, and cache points. VPLs are generated using the method described in a previous survey paper [14]. To generate surface shading points  $x_s$ , our method traces rays from the viewpoint  $x_o$ . The intersection points between the rays and the surfaces of the scene are regarded as surface shading points  $x_s$ . Volume shading points  $x_v$  are located uniformly along each segment  $x_o x_s$  (other sampling techniques [3, 19, 20] can also be used). To generate surface cache points  $c_s$ , our method randomly samples the surface shading points, whereas volume cache points  $c_v$  are randomly sampled from the line segments between the viewpoint and the surface shading points. For each cache point, the VPL contribution  $L(y_i, c) f(y_i, c, x_o) G(c, y_i) V(c, y_i)$  is computed, and the probability function  $p(y_i)$  is calculated as:

$$p(y_i) = \frac{L(y_i, c) f(y_i, c, x_o) G(c, y_i) V(c, y_i)}{\sum_{j=1}^{N_{vpl}} L(y_j, c) f(y_j, c, x_o) G(c, y_j) V(c, y_j)}. \quad (6)$$

#### 4.2. RENDERING

To render a scene including participating media, the reflected radiance  $L_r$  and the integral of the inscattered radiance  $L_i$  along the viewing ray is calculated as shown in Eq. (1). As the integral of the inscattered radiance  $L_i$  cannot be calculated analytically, we use a ray marching method:

$$L(x_o, \omega_o) \approx \tau(x_o, x_s) L_r(x_s, \omega_o) + \sum_{k=1}^K \tau(x_o, x_v^k) L_i(x_v^k, \omega_o) \Delta, \quad (7)$$

where  $x_v^k$  is the  $k$ -th volume shading point along the viewing ray,  $K$  is the number of volume shading points at each viewing ray, and  $\Delta = T/K$ .  $L_r$  and  $L_i$  are estimated using Eq. (4). To estimate  $L_r$  and  $L_i$  with less variance, our method calculates the probability function  $p$  by interpolating the probability functions stored at nearby cache points. To calculate  $p$  at each surface shading point  $x_s$ , only the surface cache points are used, because the VPL contribution at surface cache point  $c_s$  can differ from that at volume cache point  $c_v$ , even if  $c_s$  is located very close to  $c_v$ . This is because the generalized geometry term  $G$  takes into account

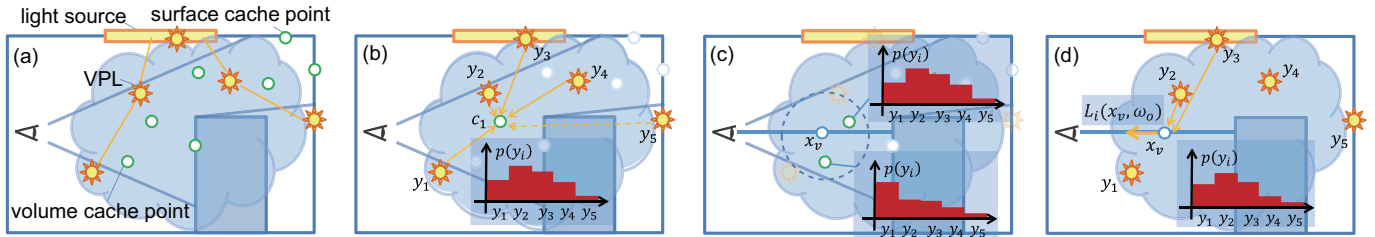


Figure 3: Rendering pipeline of our method. In the preprocessing step, VPLs, shading points, and cache points are generated (a). For each cache point, contributions from all VPLs are normalized to calculate the probability function  $p$  (b). In the rendering process, nearby cache points are searched at each shading point (c). The probability function  $p$  at each shading point is interpolated using the probabilities at nearby cache points. A small number of VPLs are sampled to estimate the inscattered radiance at volume shading point  $x_v$  (d).

the direction of the normal for the surface shading points, whereas the geometry term for the volume shading points does not consider the normal direction. For the same reason, only volume cache points are used to interpolate  $p$  at volume shading points.

Our method calculates the probability function by interpolating the probabilities stored at cache points. However, a naive interpolation method may fail to sample VPLs with large contributions, as shown in Fig. 4(b). In this case, the contribution from  $y_4$  is zero at cache points  $c_1$  and  $c_2$  because of occlusion effects, and thus the probability function  $p(y_4)$  is zero at  $c_1$  and  $c_2$ . This causes the interpolated probability  $p(y_4)$  at  $x_v$  to be zero as well, and so  $y_4$  is never sampled although it has a large contribution. To address this problem, our method employs multiple importance sampling of the interpolated probability functions and uniform sampling. The probability function  $p(y_i)$  for the  $i$ -th VPL  $y_i$  is calculated as:

$$p(y_i) = \frac{\left( \sum_{m=1}^M w_m p(c_m) + w_u p_u \right)}{\sum_{m=1}^M w_m + w_u}, w_u = \frac{\sum_{m=1}^M w_m}{M},$$

where  $M$  is the number of nearby cache points ( $M = 3$  is used in our examples) and  $p_u$  is the probability function of uniform sampling (i.e.,  $p_u = 1/N_{vpl}$ ).  $w_m$  is the weight of cache point  $c_m$ , and  $w_m = \frac{1}{\|c_m - x_v\|^2}$  for volume shading points. The weight  $w_m$  for surface shading points is calculated using the method described in [21].

## 5. RESULTS

Fig. 1 shows comparisons between (a) the reference solution that sums up all VPL contributions, (b) our importance caching method, and (c) uniform sampling of VPLs. The resolution of all images is  $512 \times 512$ . The computation times for Fig. 1 were measured on a PC with an Intel Xeon E5-2600 v2 2.80 GHz Core2 CPU. All computations were performed in parallel using multithreading. The numbers of

VPLs, cache points, and samples are 15k, 30k, and 150, respectively. The computation time of our method (Fig. 1(b)) for preprocessing is 156.9s and that for rendering is 257.1s. The computation time of uniform sampling (Fig. 1(c)) is 415.8s. The root mean square error (RMSE) of our method is 0.0354, while that of uniform sampling is 0.1483. That is, in equal-time comparison, the proposed technique results in less noise than the uniform sampling method. The computation time of reference solution (Fig. 1(a)) is 6.12h. As shown in Fig. 1, our method produces plausible images that are comparable to the reference solution an order of magnitude faster than the reference technique.

Fig. 5 shows equal-time comparisons of our method with the rendering given by uniform sampling and power sampling, which samples VPLs based on the radiance of VPL  $L(y_i, x)$  for Conference scene and Sponza scene. Error images compared with the reference solution, which sums all VPL contributions, are also shown. The computational times for Conference scene are (a) 2.79h, (b) 3.07min, and (c) 3.14min, respectively, and those for Sponza scene are (a) 4.93h, (b) 5.09min, (c) 5.05min, respectively. The computational times for power sampling are omitted since those are almost similar to those of uniform sampling. The numbers of VPLs, cache points, and samples for Conference scene are 10k, 15k, and 100, respectively, and those for Sponza scene are 15k, 30k, and 150, respectively. RMSEs of (b) our method and (c) uniform sampling are 0.0238 and 0.111 in Conference scene, and RMSEs of (b) our method and (c) uniform sampling for Sponza scene are 0.0146 and 0.111, respectively.

As shown in Fig. 5(a) and (b), our method produces less noisy images that are comparable to the reference solution an order of magnitude faster than the reference technique. In addition, an equal-time comparison shows that the proposed technique results in less noise than the other two methods. The chairs under the table in Conference scene and the columns in Sponza scene, show that the images rendered using uniform sampling and power sampling

contain significant noise due to occlusions. The proposed method produces less noisy images, because our approach constructs the probability function by taking into account the visibility between VPLs and cache points.

Fig. 6 shows comparisons between our importance caching method and the importance sampling method that uses the probability function proportional to  $L \cdot f \cdot G$ . That is, the importance sampling method calculates the VPL contribution except for the visibility  $V$  at each shading point for all the VPLs. Then the probability function which is perfectly proportional to  $LfG$  is calculated and used to sample VPLs. Error images compared with the reference solution, which sums all VPL contributions, are also shown. The computation times were measured on a PC with an Intel Xeon E5-2697 2.70 GHz Core2 CPU. In Fig. 6, the number of VPLs is 100k, and the numbers of samples for our method (Fig. 6(b)) and the importance sampling based on  $LfG$  (Fig. 6(d)) are 128 and 16, respectively. The computation times of our method and the importance sampling method are 674s and 1807s, respectively. As shown in Fig. 6(c) and (e), our method can render less noise images compared to the importance sampling method.

### 5.1. LIMITATION AND DISCUSSION

Since our method is based on many-light rendering, our method inherits from limitations of VPLs. Similar to the traditional VPL rendering methods, our method generates diffuse VPLs that limit the generalized scattering function to diffuse reflection and isotropic phase function.

Our method requires a large number of cache points compared to importance caching [11] that distributes cache points on surfaces of the scene. This results in a large computational time for preprocess. However, the total computational time including preprocess and rendering process achieves an order of magnitude faster than that of the reference solution.

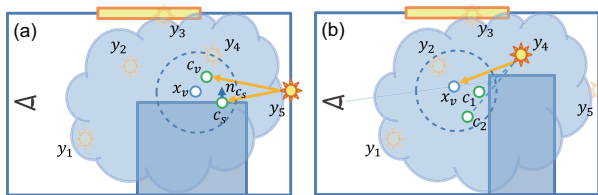


Figure 4: (a) Small correlation of VPL contributions at  $c_s$  and  $c_v$  due to  $G$ . The contribution from  $y_5$  at  $c_s$  is small, as the incident direction  $\omega_{c_s y_5}$  is almost perpendicular to the normal  $n_{c_s}$ , whereas  $G$  at  $c_v$  is irrelevant to the incident direction. (b) The contribution made by  $y_4$  to  $c_1$  and  $c_2$  differs significantly from that at  $x_v$  because of occlusions.

## 6. CONCLUSIONS

We have presented a technique that applies importance sampling to VPLs for the many-light rendering of homogeneous participating media. Using cache points, our method constructs a probability function that is proportional to the VPL contributions. This makes it possible to render plausible images of participating media by sampling a small number of VPLs, resulting in speedup of an order of magnitude compared to the reference solution.

In future work, we will apply our method to heterogeneous participating media with anisotropic scattering.

## ACKNOWLEDGEMENTS

This work was partially supported by JSPS KAKENHI Grant Number JP15H05924, JP16K12571.

- [1] H.W. Jensen, P.H. Christensen, Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps, Proc. of SIGGRAPH'98, pp. 311-320, 1998.
- [2] M. Pauly, T. Kollig, A. Keller, Metropolis Light Transport for Participating Media, Proc. of Eurographics Workshop on Rendering 2000, pp. 11-20, 2000.
- [3] M. Raab, D. Seibert, A. Keller, Unbiased Global Illumination with Participating Media, Monte Carlo and Quasi Monte Carlo Methods 2006, pp. 591-606, 2008.
- [4] A. Keller, Instant Radiosity, Proc. of SIGGRAPH '97, pp. 49-56, 1997.
- [5] T. Engelhardt, J. Novak, T-W. Schmidt, C. Dachsbacher, Approximate Bias Compensation for Rendering Scenes with Heterogeneous Participating Media, Computer Graphics Forum, Vol. 31, No. 7, pp. 2145-2154, 2012.
- [6] J. Novak, D. Nowrouzezahrai, C. Dachsbacher, W. Jarosz, Virtual Ray Lights for Rendering Scenes with Participating Media, ACM Transactions on Graphics, Vol. 31, No. 4, pp. 60:1-60:11, 2012.
- [7] J. Novak, D. Nowrouzezahrai, C. Dachsbacher, W. Jarosz, Progressive Virtual Beam Lights, Computer Graphics Forum, Vol. 31, No. 4, pp. 1407-1413, 2012.
- [8] B. Walter, A. Arbre, K. Bala, D.P. Greenberg, Multidimensional Lightcuts, ACM Transactions on Graphics, Vol. 25, No. 3, pp. 1081-1088, 2006.
- [9] R. Frederickx, P. Bartels, Ph. Dutre, Adaptive LightSlice for Virtual Ray Lights, Eurographics Short Paper, pp. 61-64, 2015.
- [10] J. Ou, F. Pellacini, LightSlice: Matrix Slice Sampling for the Many-Lights Problem, ACM Transactions on Graphics, Vol. 30, No. 6, pp. 179:1-179:8, 2011.
- [11] I. Georgiev, J. Krivánek, S. Popov, P. Slusallek, Importance Caching for Complex Illumination, Computer Graphics Forum, Vol. 31, No. 2, pp. 701-710, 2012.
- [12] H. Yoshida, K. Nabata, K. Iwasaki, Y. Dobashi, T. Nishita, Adaptive Importance Caching for Many-Light Rendering, Journal of WSCG, Vol. 23, No. 1, pp. 65-72, 2015.
- [13] E. Cerezo, F. Perez-Cazorla, X. Pueyo, F. Seron, F. Sillion, A Survey on Participating Media Rendering Techniques, The Visual Computer, Vol. 21, pp. 303-328, 2005.

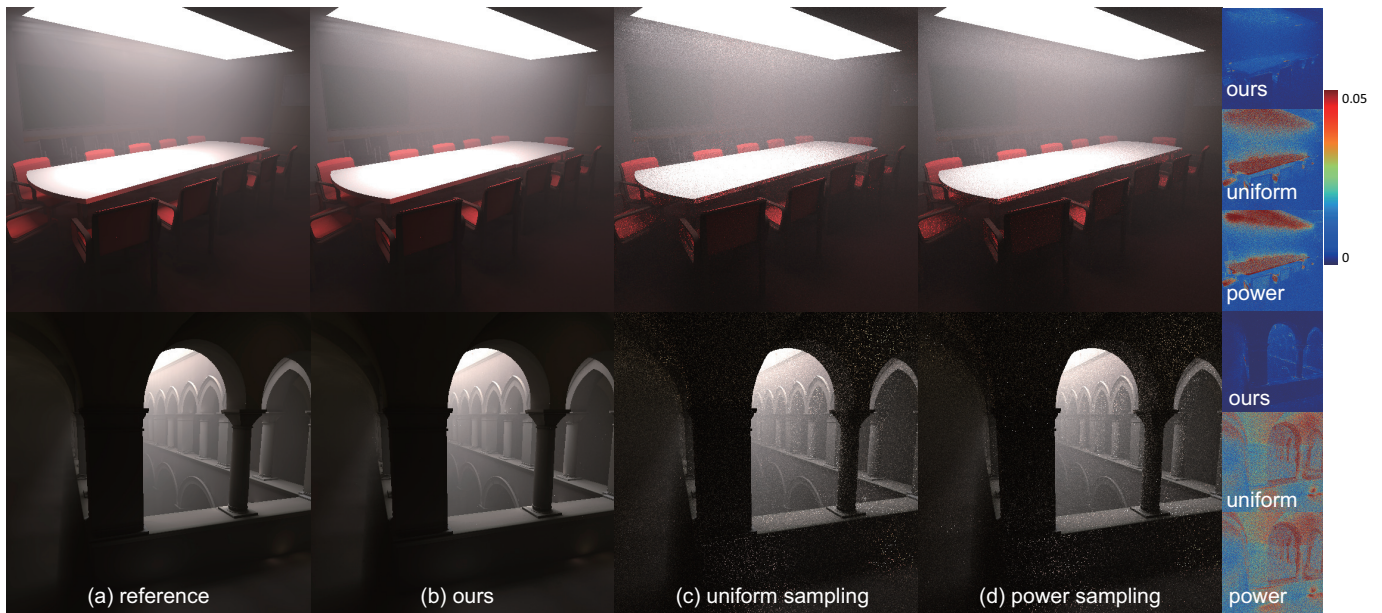


Figure 5: Rendering results of Conference scene (top row) and Sponza scene (bottom row). (a) Reference image rendered by summing all VPL contributions. Equal-time comparisons between (b) our method, (c) uniform sampling, and (d) power sampling. Rightmost images show error images for our method, uniform sampling, and power sampling respectively, as compared with the reference image.

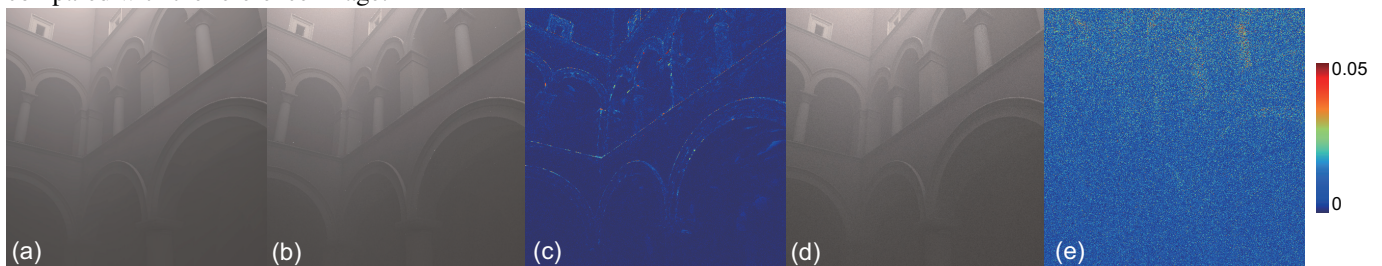


Figure 6: Rendering results of Sponza scene. (a) Reference image rendered by summing all VPL contributions, (b) our method, (c) error image of our method, (d) importance sampling based on the product  $L \cdot f \cdot G$ , and (e) error image of (d). The color bar for (c) and (e) is shown in the rightmost image.

- [14] C. Dachsbacher, J. Křivánek, M. Hašan, A. Arbree, B. Walter, J. Novak, Scalable Realistic Rendering with Many-Light Methods, *Computer Graphics Forum*, Vol. 33, No. 1, pp. 88-104, 2014.
- [15] R. Wang, O. Akerlund, Bidirectional Importance Sampling for Unstructured Direct Illumination, *Computer Graphics Forum*, Vol. 28, No. 2, pp. 269-278, 2009.
- [16] Y.T. Wu, Y.Y. Chuang, VisibilityCluster: Average Directional Visibility for Many-Light Rendering, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 19, No. 9, pp. 1566-1578, 2013.
- [17] J. Křivánek, P. Gautron, S. Pattanaik, K. Bouatouch, Radiance Caching for Efficient Global Illumination Calculation, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 11, No. 5, pp. 550-561, 2005.
- [18] W. Jarosz, C. Donner, M. Zwicker, H.W. Jensen, Radiance Caching for Participating Media, *ACM Transactions on Graphics*, Vol. 27, No. 1, pp. 7:1-7:11, 2008.
- [19] Y. Yue, K. Iwasaki, B.Y. Chen, Y. Dobashi, T. Nishita, Unbiased, Adaptive Stochastic Sampling for Rendering Inhomogeneous Participating Media, *ACM Transactions on Graphics*, Vol. 29, No. 6, pp. 177:1-177:7, 2010.
- [20] J. Novak, A. Selle, W. Jarosz, Residual Ratio Tracking for Estimating Attenuation in Participating Media, *ACM Transactions on Graphics*, Vol. 33, No. 6, pp. 179:1-179:11, 2014.
- [21] P. Clarberg, T. Akenine-Moller, Exploiting Visibility Correlation in Direct Illumination, *Computer Graphics Forum*, Vol. 27, No. 4, pp. 1125-1136, 2008.