# Adaptive Importance Caching for Many-Light Rendering


Hiroshi Yoshida
Wakayama University
Wakayama, Japan
s141064@center.wakayama-u.ac.jp

Kosuke Nabata
Wakayama University
Wakayama, Japan
s141044@center.wakayama-u.ac.jp

Kei Iwasaki
Wakayama University, UEI Research
Wakayama, Japan
iwasaki@sys.wakayama-u.ac.jp

Yoshinori Dobashi
Hokkaido University, JST CREST
Sapporo, Japan
doba@ime.ist.hokudai.ac.jp

Tomoyuki Nishita
UEI Research, Hiroshima Shudo University
Hiroshima, Japan
nishita@shudo-u.ac.jp



## ABSTRACT

Importance sampling of virtual point lights (VPLs) is an efficient method for computing global illumination. The key to importance sampling is to construct the probability function, which is used to sample the VPLs, such that it is proportional to the distribution of contributions from all the VPLs. Importance caching records the contributions of all the VPLs at sparsely distributed cache points on the surfaces and the probability function is calculated by interpolating the cached data. Importance caching, however, distributes cache points randomly, which makes it difficult to obtain probability functions proportional to the contributions of VPLs where the variation in the VPL contribution at nearby cache points is large. This paper proposes an adaptive cache insertion method for VPL sampling. Our method exploits the spatial and directional correlations of shading points and surface normals to enhance the proportionality. The method detects cache points that have large variations in their contribution from VPLs and inserts additional cache points with a small overhead. In equal-time comparisons including cache point generation and rendering, we demonstrate that the images rendered with our method are less noisy compared to importance caching.



## Keywords

Global Illumination, Many-Light Rendering, Importance Sampling, Importance Caching


## 1 INTRODUCTION

Photorealistic rendering has, for many years, been an interesting and challenging topic in the field of computer graphics. It has been widely used in many applications such as movies, games, architectural design, and so on. Indirect illumination plays an important role in enhancing realism. However, efficient rendering with indirect illumination is still a challenging problem due to the high computational cost.

To compute indirect illumination efficiently, Keller introduced an instant radiosity, which approximates the indirect illumination with virtual point lights (VPLs) [KELLER97]. Many-light rendering [DACHSBACHER14], which extends the instant radiosity, has been extensively researched. Many-light rendering approximates both the direct and indirect illumination incident onto each point to be shaded (referred to as *shading points*) with VPLs. Increasing the number of VPLs increases the accuracy of many-light rendering, but at the cost of computational time.

To handle a large number of VPLs efficiently, importance sampling methods [WANG09, GEORGIEV12, WU13] for VPLs that estimate the outgoing radiance of shading points have been proposed. The key component for the importance sampling method is to construct a probability function that is as proportional as possible to the distribution of contributions from all the VPLs. However, constructing a probability function perfectly proportional to the distribution at each shading point is computationally expensive since it requires a large number of visibility tests between the shading point and all VPLs. Importance caching [GEORGIEV12] constructs a probability function by sparsely distributing the cache points on the surfaces of the scene, and recording the contributions of all VPLs. The probability function at each shading point is calculated by interpolating those at nearby cache points. This method, however, distributes cache points randomly and does not consider the variation in contributions from each VPL. This makes it difficult to construct a probability function that is proportional to the distribution of the VPL contributions, leading to an increase in variance.

To address this problem, we propose an adaptive cache insertion method for a many-light rendering framework. Our method detects regions where the distribution of the VPL contributions varies drastically due to the spatial variations of the shading points, the directional variations of the normals to the shading points, and due to the occlusions between the VPLs and the shading points. Additional cache points are inserted into such regions. In addition, while importance caching calculates the interpolated probability function by simple averaging, our method takes into account the spatial correlation between the shading points and the cache points, and the directional correlation between the normals to the shading points and cache points, and uses these to weight the interpolation. Our results demonstrate that, in an equal time comparison, our method provides better performance (i.e. less variance) than importance caching.

## 2 PREVIOUS WORK

Many-light rendering, which is based on the instant radiosity as proposed by Keller [KELLER97], distributes a large number of VPLs in the scene and approximates the incident radiance from the direct and indirect illumination from the VPLs. Increasing the number of VPLs increases the rendering accuracy at the cost of computational time. Since several thousand VPLs are required to obtain plausible results, several methods have been proposed that can handle many VPLs efficiently.

To handle a large number of VPLs efficiently, several methods that cluster VPLs have been proposed. Walter et al. proposed a hierarchical representation of the VPLs called Lightcuts [WALTER05]. Hasan et al. proposed the matrix row-column sampling (MRCS) method that samples a small number of VPLs that give a good approximation to the contributions from all the VPLs [HASAN07]. Ou et al. proposed the Lightslice method, which extends the MRCS method by clustering the shading points and applying the MRCS method to each cluster to improve the accuracy [OU11]. Although these clustering methods can efficiently render realistic images by approximating the contributions of the VPLs in each cluster by a representative VPL from each cluster, the rendered images suffer from errors due to VPL clustering.

Importance sampling methods for VPLs have also been proposed. The contribution of a VPL is the product of the incident radiance, the bidirectional reflectance distribution function (BRDF), a geometry term, and the visibility function. By constructing probability functions proportional to the contribution from VPLs and sampling the VPL according to this probability function, the outgoing radiance can be estimated with high accuracy and small variance. Wang and Akerlund proposed a bidirectional importance sampling method for many-light rendering [WANG09]. This method, however, does not take into account the visibility function, resulting in high variance where the incident light is occluded. Wu et al. proposed the VisibilityCluster algorithm, which clusters shading points and the VPLs [WU13]. The visibility function is approximated by the average values of the estimated visibilities between the clusters of shading points and the VPLs. Although this method can render realistic images efficiently, it can fail to sample VPLs with large contributions since estimates of the average values of the visibilities are done by random sampling.

Cache-based methods that exploit correlation to increase the rendering efficiency have been proposed. Ward et al. proposed irradiance caching, which accelerates the indirect illumination calculation by interpolating the incident illumination stored at cache points [WARD88]. Radiance caching methods [KRIVANEK05, KRIVANEK06] store radiance instead of irradiance to efficiently render glossy materials. Visibility caching stores visibility information to accelerate the direct illumination computation [CLARBERG08] . The work that is most relevant to our method is importance caching [GEORGIEV12]. This method randomly distributes cache points, called importance records, in the scene and records the contributions of all the VPLs as shown in Fig. 1(a). Then a probability function perfectly proportional to the contributions of the VPLs at each cache points is calculated. At each shading point, the probability function is calculated by interpolating the contributions stored at nearby cache points, and a small number of VPLs are sampled to estimate the outgoing radiance. Although it can render plausible images efficiently, this method has several drawbacks. Firstly, cache points are distributed randomly. If the VPL contributions stored at the cache points vary drastically, the interpolated probability function may not be proportional to the contributions of the VPLs. Secondly, the interpolated probability function is simply an average of those recorded at nearby cache points, which does not account for the correlation between the shading and cache points. To address this problem, we propose an adaptive cache insertion method for many-light rendering. Our method distributes cache points taking into account variations in the VPL contributions. In addition, our method interpolates the probability function by weighted averages of the probability functions stored at nearby cache points taking into account the correlations between the shading and cache points.

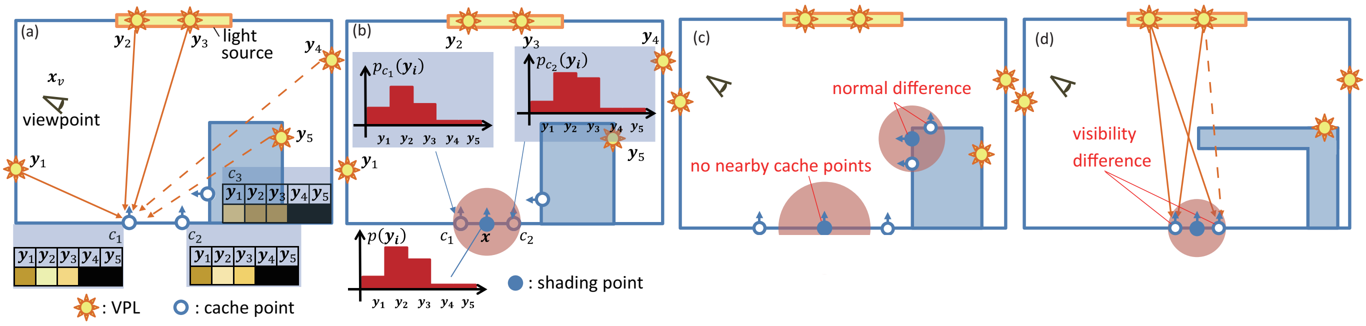


Figure 1: Importance caching (a) records contributions of all VPLs at cache points $c_j$. Tables under cache points represent VPL contributions. (b) Probability function $p$ at shading point is calculated by averaging those at nearby cache points. Each graph shows the probability function, and the probability functions at cache points are calculated by normalizing VPL contributions. (c) Inefficient cases of importance caching, lack of nearby cache points and lack of cache points with similar normals. (d) Contributions of VPLs at nearby cache points differ due to occlusions.

## 3 IMPORTANCE CACHING

Importance caching [GEORGIEV12] samples VPLs based on a probability function calculated by interpolation between those stored at cache points. The outgoing radiance $L_o(\mathbf{x}, \mathbf{x}_v)$ at shading point $\mathbf{x}$ towards the viewpoint $\mathbf{x}_v$ is estimated by the following equation:

$$L_o(\mathbf{x}, \mathbf{x}_v) = \frac{1}{N} \sum_{n=1}^{N} \frac{L(\mathbf{y}_n, \mathbf{x}) f_r(\mathbf{y}_n, \mathbf{x}, \mathbf{x}_v) G(\mathbf{x}, \mathbf{y}_n) V(\mathbf{x}, \mathbf{y}_n)}{p(\mathbf{y}_n)}, \tag{1}$$

where $N$ is the number of sampled VPLs, $\mathbf{y}_n$ is the $n$-th VPL, and $L$, $f_r$, $G$, and $V$ are the radiance, BRDF, the geometry, and the visibility terms, respectively (please refer to the many-light rendering survey paper [DACHSBACHER14] for more details). The contribution of the VPL is the product of $L$, $f_r$, $G$, and $V$. The probability function $p$ for sampling the VPLs is expected to be proportional to the distributions of the VPL contributions. However, constructing a probability function perfectly proportional to the distribution of the VPL contributions is computationally expensive since it requires evaluation of all the VPL contributions.

To address this problem, importance caching randomly distributes a small number of cache points in the scene. At each cache point, the contributions from all the VPLs are calculated. The probability function that is perfectly proportional to the distribution of the VPL contributions is calculated by normalizing the distribution. The contribution from the VPLs to the shading point seems to be correlated with those stored at nearby cache points. By exploiting the correlation of the contributions, the probability function $p$ at each shading point is obtained by interpolating those at nearby cache points. However, when geometrical information (e.g. the normal) or the VPL contribution between a shading point and a cache point has a small correlation as shown in Figs. 1(c)(d), the proportionality of the interpolated probability function decreases.

## 4 PROPOSED METHOD

Instead of random sampling, our method distributes cache points taking into account the geometrical information of the shading points and the distribution of the VPL contributions. Fig. 2 shows an overview of our method.

### 4.1 Generating Initial Cache Points

The contributions from a VPL to two shading points $\mathbf{x}_i, \mathbf{x}_j$ have large correlation when the positions $\mathbf{x}_i, \mathbf{x}_j$ and the normals $\mathbf{n}_i, \mathbf{n}_j$ to the shading points are similar. By exploiting this, our method first clusters the shading points based on the positions and the normals, employing the clustering method described in [OU11]. The shading points are represented by 6-dimensional points consisting of the positions and the normals. Firstly, the positions of the shading points are normalized into $[-1, 1]^3$, which is equal to the range of the normals. The bounding box of the 6-dimensional points is calculated, and then recursively subdivided until the number of 6-dimensional points or the size of bounding box is smaller than the thresholds. The bounding box is split along its longest axis. After the subdivision is terminated, one shading point is randomly sampled from each cluster and is used as the cache point. At each cache point, the contributions from all the VPLs are calculated and a cumulative distribution function is constructed.

### 4.2 Adaptive Insertion of Cache Points

The initial cache points are distributed according to the similarity of the shading points, but not considering the contributions of VPLs. For example, as shown in Fig. 1(d), the contributions of VPLs can differ at nearby cache points due to occlusions, leading to the interpolated probability function having reduced proportionality and increased variance.

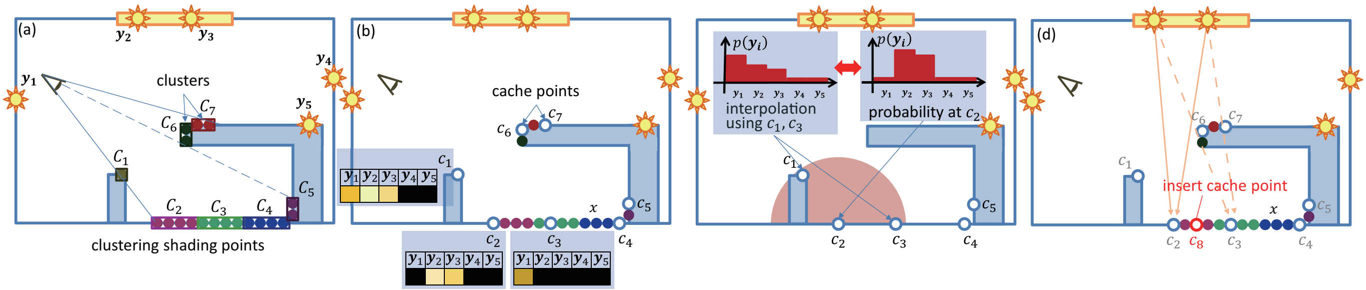


Figure 2: Overview of our method. (a) Clustering shading points based on their positions and normals. (b) Calculate contributions of VPLs at each cache point $c_j$. (c) Calculate sum of differences of $p$ at $c_2$ and interpolated probability function using $c_1$ and $c_3$. (d) Insert new cache point $c_8$ from cluster $C_2$.

To address this problem, our method exploits the fact that each cache point records the contributions of all VPLs and therefore, an ideal probability function perfectly proportional to the contributions of all the VPLs is easily obtained. Our method detects those nearby cache points whose recorded contributions differ due to occlusions, and inserts additional cache points for such regions. If the VPL contributions recorded at cache points near to $c_j$ differ drastically from those at $c_j$ due to occlusions, the interpolated probability function differs from the probability function of $c_j$. Therefore, our method calculates the sum of the differences between the probability function recorded at $c_j$ and interpolated probability function from nearby cache points of $c_j$. If the sum of differences exceeds the threshold $\delta$, an additional cache point is inserted. The threshold $\delta$ is set experimentally in the current implementation.

The VPL contributions at the additional cache point need to have large correlation with those recorded at $c_j$. To correlate VPL contributions between the additional cache point and $c_j$, small variations in the geometrical information and the occlusions are required. However, computing the visibilities between all VPLs and a cache point is computationally expensive, it is difficult to detect the variations in the occlusions with a small overhead. Our method calculates the positions of the additional cache points using the geometrical information of $c_j$. Since the shading points in the cluster $C_j$ corresponding to $c_j$ have similar geometry information, our method samples one shading point randomly from $C_j$. The insertion process for all the cache points is repeated until the number of inserted cache points is smaller than a threshold. The cache points are stored in a kd-tree for fast search of cache points near to each shading point.

Fig. 3 shows the initial cache points (left) and the adaptively inserted cache points (right) of a Cornell box scene. The initial cache points are distributed uniformly on the surfaces of the scene, while the inserted cache points are distributed near the boundaries of shadows, where the visibilities between the VPLs and the cache points change.

## 4.3 Rendering

The outgoing radiance $L_o(\mathbf{x}, \mathbf{x}_v)$ at shading point $\mathbf{x}$ is calculated by sampling VPLs according to the probability function $p$ interpolated from those recorded at a number, M, of nearby cache points. In contrast to the simple average as in importance caching [GEORGIEV12], our method calculates the probability function $p$ using a weighted average that considers the spatial and directional correlations between the shading and cache points. The probability function $p$ is calculated by the following equation:

$$p(\mathbf{y}_n) = \sum_{k=1}^{M} w_k p_k(\mathbf{y}_n), \quad (2)$$

where $M$ is the number of cache points. $M = 3$ works well for our method as proposed in [GEORGIEV12]. $w_k$ and $p_k$ are the weight and probability function for the $k$-th nearest cache point, respectively. The weight $w_k$ is calculated using the formula proposed in [CLARBERG08].

$$w_k = \sqrt{1 - |\mathbf{n} \cdot \mathbf{v}|} \cdot \hat{w}(d, \theta), \quad (3)$$

where $\mathbf{n}$ is the normal to shading point $\mathbf{x}$, and $\mathbf{v}$ is the normalized vector from $\mathbf{x}$ to the $k$-th nearest cache point $c_k$. $d$ is the distance between the shading point and the cache point, and $\theta$ is the angle between the normals to the shading and cache points. The weight function $\hat{w}$ is calculated from the unnormalized weight function $w$:

$$w(d, \theta) = \left(1 - \frac{\theta}{\pi}\right)\left(1 - \frac{d'}{1 + \lambda d'}\right), \quad (4)$$

where $d' = d/d_{max}$, $d_{max}$ is the maximum search range, and $\lambda$ is a parameter. The weight function $\hat{w}$ is calculated from $\hat{w} = (w(d, \theta) - w(d_{max}, \theta_{max}))/(1 - w(d_{max}, \theta_{max}))$, where $\theta_{max}$ is the maximum angle between the normals. $\theta_{max} = \pi/6$ and $\lambda = 5$ are used as proposed in [CLARBERG08].

# 5 RESULTS

Figs. 4, 5, and 6 show equal-time comparisons between our method and importance caching [GEORGIEV12].

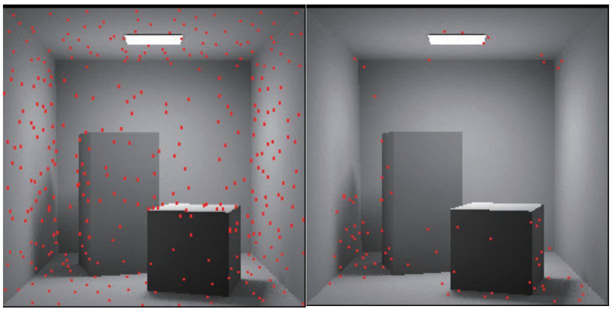

Figure 3: Example of cache points (red points) in the Cornell box scene. Left: initial cache points generated by sampling each cluster of shading points. Right: inserted cache points generated by considering VPL contributions.

The computational times are measured on a desktop PC with an Intel Xeon CPU E3 1270 3.4GHz. All the computations were performed in parallel using multithreading. The image resolutions are $1024 \times 768$. All the computations used in our method, except for cache generation, adaptive insertion, and weight calculation, are the same as those used in importance caching. Cache points are generated randomly in importance caching, and the same numbers of cache points are used in our method as in importance caching. As described in [GEORGIEV12], four sampling strategies are used. Bilateral multiple importance sampling using the $\alpha$-max heuristic is performed for both methods. Our method distributes VPLs in the same way as described in [DACHSBACHER14]. The reference images are rendered by accumulating the contributions from all the VPLs. Table 1 shows the statistics of our results.

Fig. 4 shows an equal-time comparison of a Sibenik scene. The computational times (cache generation/rendering) for our method and importance caching were 36.6s (17.8s for cache generation/18.8s for rendering) and 33.7s (15.2s/18.5s), respectively. Fig. 4(a) shows the result rendered using our method. Figs. 4(b), (c), and (d) show close-ups of the area outlined in red in Fig. 4(a) for the reference image, and the results rendered by our method, and importance caching, respectively. Figs. 4(e), (f), and (g) show close-ups of the area outlined in blue in Fig. 4(a). As shown in these images, our method can render images with less noise compared to importance caching. Importance caching [GEORGIEV12] tends to distribute cache points near the viewpoint. Therefore regions far from the viewpoint (e.g. the windows in Fig. 4(d)) have less cache points, resulting in noisy images. In addition, since importance caching does not take into account occlusions in distributing cache points, the regions where occlusions vary drastically (e.g. Fig. 4(g)) suffer from noise, whereas our method can distribute cache

Table 1: Statistics of results. $N_T$, $N$, and $N_c$ are the number of triangles, VPLs, and cache points, respectively.

| Scene | $N_T$ | $N$ | $N_c$ |
|---|---|---|---|
| Sibenik (Fig. 4) | 75,284 | 7,785 | 2,950 |
| Sponza (Fig, 5) | 66,450 | 6,479 | 1,728 |
| Conference (Fig. 6) | 331,179 | 5,133 | 2,478 |

points for such regions, resulting in less noise as shown in Fig. 4(f).

Fig. 5 shows an equal-time comparison of a Sponza scene. As shown in Figs. 5(b) to (g), our method can render less noisy images especially for regions (e.g. arches and pillars) where the visibilities between the VPLs and the shading points change. The computational times (cache generation/rendering) for our method and importance caching were 24.7s (7.5s/17.2s) and 23.3s (5.9s/17.3s), respectively.

Fig. 6 shows an equal-time comparison of a Conference scene. Figs. 6(b)(d), (c)(f), and (d)(g) show close-ups of the reference image, the results rendered by our method, and importance caching, respectively. In Fig. 6(d), a large variance due to the occlusion due to the table appears in the chair, whereas our method (Fig. 6(c)) renders an image comparable to the reference image shown in Fig. 6(b). The computational times for our method and importance caching were 18.7s (6.5s/12.2s) and 18.2s (6.4s/11.9s), respectively.

Figs. 7, 8, and 9 show visualizations of the root-mean-square-error (RMSE) between each method and reference images rendered by summing all the VPL contributions. The color bar shows the false color. For the Sibenik scene, the RMSE for our method is 0.0486773 while that for importance caching is 0.0662813. As shown in Fig. 7, our method can render less noisy images especially near windows and pillars. In the Sponza scene, the RMSE for our method is 0.146164 while that of importance caching is 0.207015. Since it is difficult for importance caching to distribute cache points inside the scene, large variance can appear as shown in Fig. 8(b), while our method can lessen this as shown in Fig. 8(a). In the Conference scene (Fig. 9), the RMSE for our method and importance caching are 0.0294903 and 0.032524, respectively. As shown in Fig. 9, by inserting additional cache points, our method reduces the variance near chairs occluded by the table.

Since, with our method, new cache points are added in regions where the variations in the VPL contributions are large, for the same number of cache points, the cache points are distributed more sparsely in other regions compared to importance caching, resulting in a slightly increased variance (e.g. near the floor in Fig. 8). However, our method can reduce the RMSE for the overall scene as shown in Figs. 7, 8, and 9.

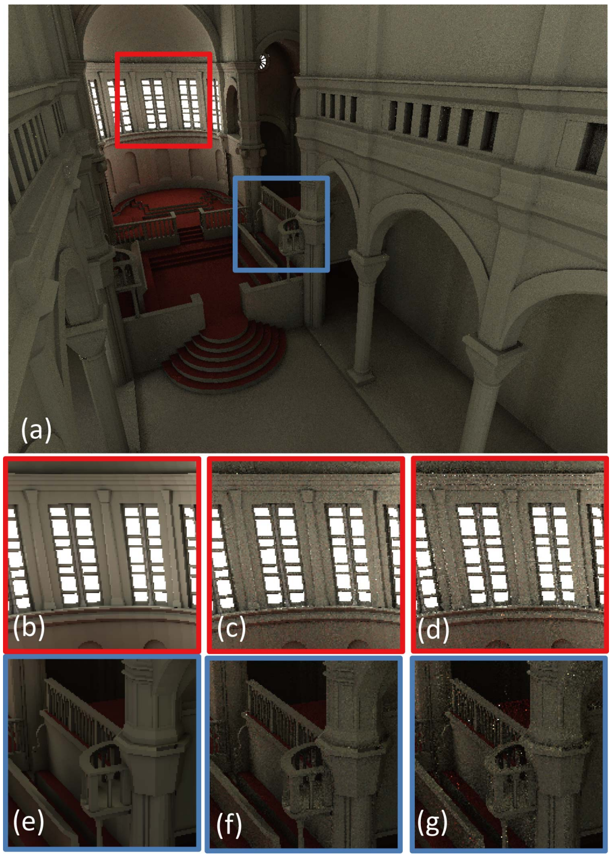


Figure 4: Sibenik scene. (a) rendering result of our method. (b)(c)(d) close-up images of reference, our method, and importance caching, respectively. (e)(f)(g) close-up images of reference, our method, and importance caching. Our method can render less noisy image in equal time rendering compared to importance caching.

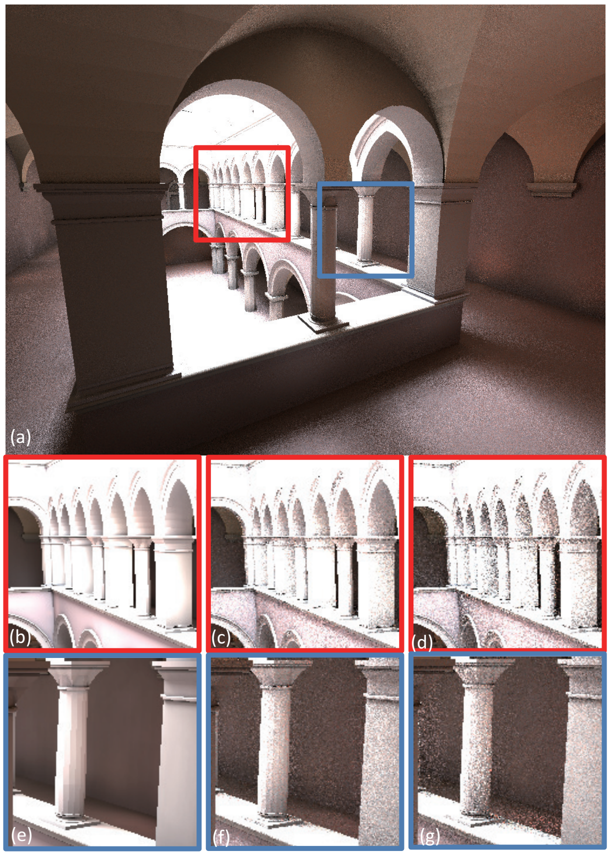


Figure 5: Sponza scene. (a) rendering result of our method. (b)(c)(d) close-up images of reference, our method, and importance caching, respectively. (e)(f)(g) close-up images of reference, our method, and importance caching. Our method can render less noisy images, especially near arches and pillars in equal time rendering.

To inspect the effectiveness of the adaptive insertion of cache points and the weighting function that considers the spatial and directional correlations, our method renders the Sibenik scene using adaptively inserted cache points and uniform weights used in [GEORGIEV12]. The RMSE in this case is 0.0516844, while that with random cache points and uniform weights is 0.0662813. As shown in this experiment, adaptive cache insertion contributes to the improvements most.

## 6 CONCLUSIONS AND FUTURE WORK

We have proposed an adaptive cache insertion method for importance caching. Our method clusters the shading points and selects cache points from clusters and exploits the spatial and directional correlations between shading points and cache points. Our method detects cache points whose VPL contributions differ from those of nearby cache points and inserts further cache points, resulting in reduced variance compared to that obtained in equal-time rendering using the original importance caching method.

For future work, we plan to accelerate our method using VPL clustering. Moreover, we propose to distribute cache points taking into account the scene saliency.

## ACKNOWLEDGEMENTS

This research was partially supported by JST, CREST.

## 7 REFERENCES

[KELLER97] A. Keller, Instant Radiosity, Proc. of SIGGRAPH '97, pp. 49-56, 1997.

[DACHSBACHER14] C. Dachsbacher, J. Krivanek, M. Hasan, A. Arbree, B. Walter, J. Novak, Scalable Realistic Rendering with Many-Light Methods, Computer Graphics Forum, Vol. 33, No. 1, pp. 88-104, 2014.

[WANG09] R. Wang, O. Akerlund, Bidirectional Importance Sampling for Unstructured Direct Illumination, Computer Graphics Forum, Vol. 28, No. 2, pp. 269-278, 2009.

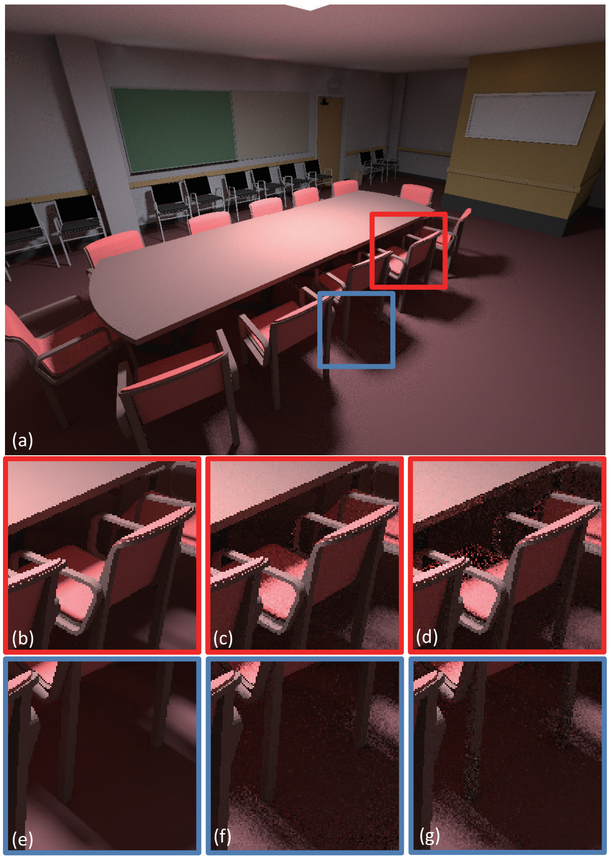


Figure 6: Conference scene. (a) rendering result of our method. (b)(c)(d) close-up images of reference, our method, and importance caching, respectively. (e)(f)(g) close-up images of reference, our method, and importance caching. Our method can render less noise images, especially for chairs in equal time rendering.

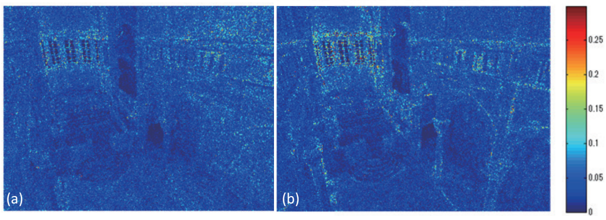


Figure 7: Error images of Sibenik scene. (a) our method and (b) importance caching. RMSEs of our method and importance caching are 0.0486773 and 0.0662813, respectively.

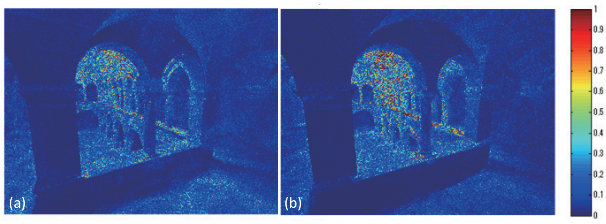


Figure 8: Error images of Sponza scene. (a) our method and (b) importance caching. RMSEs of our method and importance caching are 0.146164 and 0.207015, respectively.

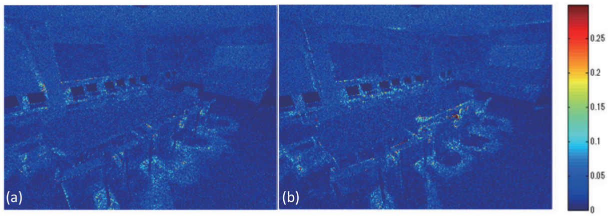


Figure 9: Error images of Conference scene. (a) our method and (b) importance caching. RMSEs of our method and importance caching are 0.0294903 and 0.032524, respectively.


[WU13] 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.

[GEORGIEV12] I. Georgiev, J. Krivanek, S. Popov, P. Slusallek, Importance Caching for Complex Illumination, Computer Graphics Forum, Vol. 31, No. 2, pp. 701-710, 2012.

[WALTER05] B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian, D. Greenberg, Lightcuts: A Scalable Approach to Illumination, ACM Transactions on Graphics, Vol. 24, No. 3, pp. 1098-1107, 2005.

[HASAN07] M. Hasan, F. Pellacini, K. Bala, Matrix Row-Column Sampling for the Many-Light Problem, ACM Transactions on Graphics, Vol. 26, No. 3, pp. 26:1-26:10, 2007.

[OU11] 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.

[CLARBERG08] P. Clarberg, T. Akenine-Moller, Exploiting Visibility Correlation in Direct Illumination, Computer Graphics Forum, Vol. 27, No. 4, pp. 1125-1136, 2008.

[WARD88] G.J. Ward, F.M. Rubinstein, R.D. Clear, A Ray Tracing Solution for Diffuse Interreflection, ACM SIGGRAPH Computer Graphics, Vol. 22, No. 4, pp. 85-92, 1988.

[KRIVANEK05] J. Krivanek, 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.

[KRIVANEK06] J. Krivanek, K. Bouatouch, S. Pattanaik, J. Zara, Making Radiance and Irradiance Caching Practical:Adaptive Caching and Neighbor Clamping, Proc. of Eurographics Symposium on Rendering, pp. 127-138, 2006.