VisibilityCluster: Average Directional Visibility for Many-Light Rendering

Yu-Ting Wu and Yung-Yu Chuang, *Member, IEEE*

Abstract—This paper proposes the VisibilityCluster algorithm for efficient visibility approximation and representation in many-light rendering. By carefully clustering lights and shading points, we can construct a visibility matrix that exhibits good local structures due to visibility coherence of nearby lights and shading points. Average visibility can be efficiently estimated by exploiting the sparse structure of the matrix and shooting only few shadow rays between clusters. Moreover, we can use the estimated average visibility as a quality measure for visibility estimation, enabling us to locally refine VisibilityClusters with large visibility variance for improving accuracy. We demonstrate that, with the proposed method, visibility can be incorporated into importance sampling at a reasonable cost for the many-light problem, significantly reducing variance in Monte Carlo rendering. In addition, the proposed method can be used to increase realism of local shading by adding directional occlusion effects. Experiments show that the proposed technique outperforms state-of-the-art importance sampling algorithms, and successfully enhances the preview quality for lighting design.

Index Terms—Ray tracing, visibility approximation, importance sampling, the many-light problem

1 Introduction

The commonly employed Monte Carlo ray tracing method tends to suffer from noisy results due to the large variance in stochastic sampling. Importance sampling is an effective strategy for reducing that variance. It requires to efficiently approximate the integral kernel of the rendering equation, a triple product of incident lighting, material properties, and visibility. Among these terms, visibility is often ignored because its estimation demands expensive shadow ray casting or shadow map construction. However, a recent study has reported that in industry-level scenes more than half of the shadow rays (sometimes over 90 percent of them) end up with occlusion [1]. Thus, the neglect of visibility largely limits the effectiveness of importance sampling.

We, therefore, propose an efficient method for estimating average visibility to improve the effectiveness of importance sampling. We focus on rendering with the many-light formulation (i.e., the virtual point light (VPL) method), which has received much attention in recent years. In previous importance sampling approaches, visibility is either completely omitted [2] or sparsely sampled [3] because of its expensive computational cost. The *bidirectional importance sampling* (BIS) approach [2] only approximates lighting and BRDFs without taking visibility into account. It suffers from significant noise when contributions from strong lights or BRDF peaks are occluded. Georgiev et al. [3] later proposed *Importance Caching* (IC) to address this problem. However, their method fails to handle high-frequency variations of geometry and visibility. Fig. 1 shows problems with these approaches.

We propose *VisibilityCluster*, a method for efficient computation and compact representation of the visibility function. The method is based on the observation that visibility terms in the many-light transport matrix exhibit good local structures if shading points and lights are properly clustered. Fig. 2a visualizes the visibility matrix for a scene, in which rows and columns correspond to shading points and lights, respectively, and matrix entries represent their visibility values. It took 4,856 seconds to compute the full matrix of pairwise visibility. After clustering, the reordered matrix exhibits good local structures (Fig. 2b). We call a submatrix formed by a cluster of lights (light cluster (LC)) and a cluster of shading points (shading cluster (SC)) a *VisibilityCluster* and calculate its average visibility (Fig. 2c). The average visibility is particularly useful for the applications that do not require perfect visibility, such as importance sampling. By sampling lights and shading points, we can estimate all average visibility terms between clusters in 14 seconds and with only 3.57 percent error (Fig. 2d). Our method has several advantages:

- It is usually sufficient to estimate the average visibility accurately with few visibility samples.
- It makes the importance function more compact and improves performance for both construction and sampling, which makes it more scalable in terms of number of lights.
- The estimated visibility provides a direct quality measure to visibility estimation and can be used for guiding further refinement of clustering.

The proposed VisibilityCluster method produces visibility information and can be built on top of *BIS* [2] for more effective variance reduction. Including visibility allows us

- *The authors are with the Department of Computer Science and Information Engineering, National Taiwan University, R503, CSIE Building, No. 1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan, R.O.C.*
E-mail: {kevincosner, cyy}@cmlab.csie.ntu.edu.tw.

Manuscript received 29 Sept. 2012; revised 5 Feb. 2012; accepted 8 Feb. 2013; published online 15 Feb. 2013.
Recommended for acceptance by A. Sheffer.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2012-09-0213.
Digital Object Identifier no. 10.1109/TVCG.2013.21.

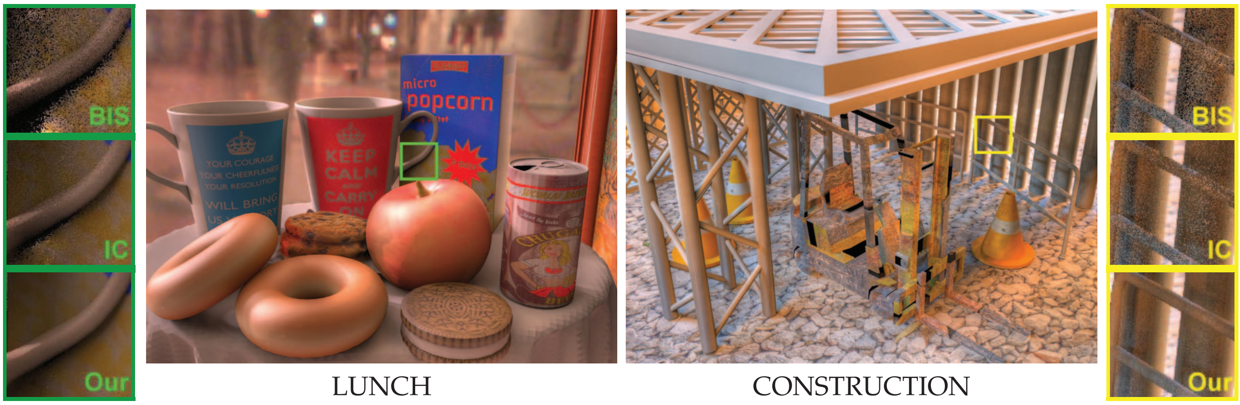


Fig. 1. Examples for limitations of previous importance sampling methods. Previous methods produce significant noise in regions with high-frequency visibility variations due to complex occlusion or fine scene structures, such as the coffee mug handle very close to the popcorn box in LUNCH and the thin handrails in CONSTRUCTION. Insets show detailed comparisons for parts of the scenes rendered with different methods given the same amount of time (200 seconds for LUNCH and 500 seconds for CONSTRUCTION). The insets, from top to bottom, are BIS [2], IC [3], and our approach. Our method provides better noise reduction across the whole image including these difficult areas.

to better sample unoccluded lights. Experiments show that our method achieves superior noise reduction, compared to other state-of-the-art importance sampling techniques when rendering a variety of complex scenes with different illumination conditions. In addition to importance sampling, we demonstrate that VisibilityCluster can be combined with local shading to produce visually pleasing directional occlusion effects. Altogether, the proposed VisibilityCluster algorithm offers a good compromise between performance and quality, and is well suited for lighting previews and high-quality rendering.

2 Related Work

2.1 Importance Sampling

Importance sampling is an effective strategy to reduce variance in Monte Carlo rendering. Its main idea is to sample with a distribution proportional to the integral kernel, which in our case is the triple product of incident lighting, BRDFs, and visibility. Efficient construction of such distributions, however, is challenging because it requires accurate approximation of these terms. Among them, visibility is particularly costly to approximate.

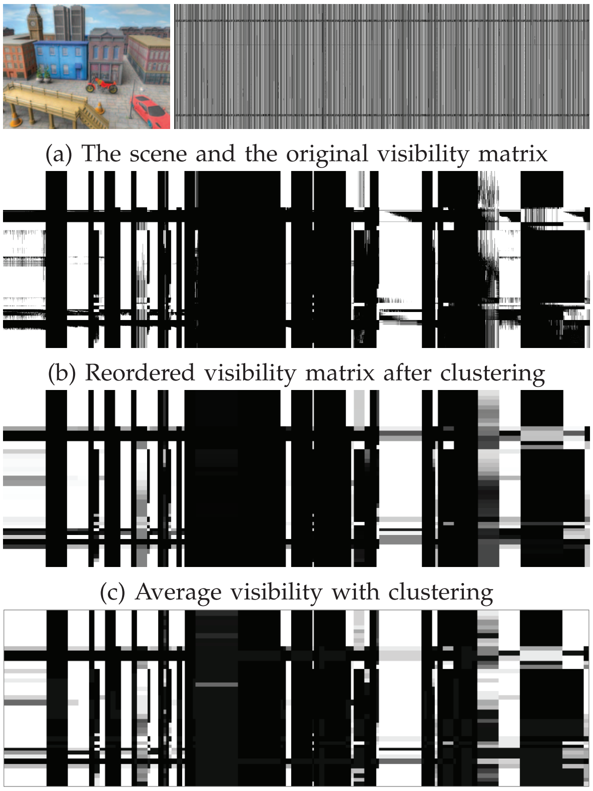
Earlier algorithms only approximate one of these terms, such as illumination [4], [5], [6] or material properties [7]. Recently, several (bidirectional) product importance sampling methods have been proposed. Pioneered by wavelet importance sampling [8], earlier approaches focus on sampling the product of environment lighting and BRDFs [9], [10], [11]. Wang and Åkerlund [2] later proposed a more general approach that can sample both structured and unstructured illumination (such as area lights or indirect lighting) by using the many-light formulation. Other well-known approaches for generating product distributions of lighting and BRDFs are based on the idea of importance resampling [12], [13]. They approximate the product distribution by starting from one distribution and then refining with the other. These approaches can handle both direct and indirect illumination. The major problem of the above methods lies in the oversampling of occluded lights, making them unsuited for scenes with complex visibility.

Some approaches are built on top of product importance sampling and try to reduce visibility variance. Ghosh and Heidrich [14] reduced noise in partially occluded regions by using Metropolis-Hastings mutations. Rousselle et al. [15] included a very conservative visibility term by approximating scenes with inner spheres. The approach is not suitable for fine geometry and cannot handle partially occluded areas. Clarberg and Akenine-Möller [16] used control variate to reduce variance. They reformulated the rendering equation with an interpolated visibility term. Although with some success, without changing the distribution for sampling, their method is less effective. Finally, the table-driven adaptive importance sampling proposed by Cline et al. [17] improves sample quality by reusing importance functions from neighbor pixels in the previous rendering process. Their method suffers from large variance near the boundary, where no shading information is available.

Georgiev et al. [3] recently proposed IC to exploit coherence in the reflection integrand. Full lighting contributions including visibility are evaluated and cached at sparse locations. Several types of distributions, ranging from aggressive to conservative, are built at these locations and reused by nearby shading points with a multiple importance sampling scheme. The major limitation of their method is that the quality of distributions decays quickly in the presence of high-frequency variations of materials, geometry, and visibility. In addition, as the number of lights grows, the memory for storing distributions and the computation overhead for generating distributions become big concerns.

2.2 Many-Light Rendering

The seminal work, *Instant Radiosity*, proposed by Keller [18] laid the foundation of using VPLs to represent complex illumination. Direct and indirect illumination can be handled with a united framework. However, high-quality rendering requires a large number of VPLs and accumulating all lights is often impractical. *Lightcuts* was proposed to reduce the number of shading operations by adaptive clustering [19], [20], [21]. For each point, a shading cut is used to keep the expected error under a threshold. Visibility is not included when estimating the error bound. Hašan et



(a) The scene and the original visibility matrix

(b) Reordered visibility matrix after clustering

(c) Average visibility with clustering

(d) Approximated average visibility with VisibilityCluster

Fig. 2. Visualization of the visibility matrix for TOWN. The rows and columns correspond to shading points and lights, respectively. Each matrix entry represents the visibility between the corresponding shading point and light. In the original order of shading points and lights, the visibility matrix (a) has little coherence. After reordering, the visibility matrix (b) exhibits blocky structures due to visibility coherence. We call a submatrix formed by an LC and SC a VisibilityCluster. By averaging the visibility terms within each VisibilityCluster, we obtain the average visibility matrix (c). It took 4,856 seconds to compute pairwise visibility between each light and shading point, and construct the average visibility matrix. The brightness indicates the visibility ratio between a pair of clusters. Our VisibilityCluster algorithm efficiently approximates the average visibility matrix by only casting few shadow rays for each VisibilityCluster. The approximated average visibility matrix (d) with VisibilityCluster was estimated with only 14 seconds and 3.57 percent error to the average visibility matrix (c).

al. [22] interpreted the many-light problem using a matrix form. They exploited the low-rank property of the matrix for efficient rendering by sampling both rows (shading points) and columns (lights). Hašan et al. [23] and Davidovič et al. [24] later extended their method for handling glossy materials. Ou and Pellacini [25] further observed that matrices for local surfaces tend to have very low ranks. They computed per-slice LCs for more efficient approximation. Our method, similarly to theirs, exploits the local structure of the many-light matrix. However, we specifically focus on the visibility term and use average visibility to exploit the coherence explicitly, instead of using sparse visibility sampling, like they did. Another important difference is that their method uses a fixed set of representative lights to shade the scene, while our approach adopts a Monte Carlo estimator at each shading point with the full set of VPLs for sampling.

There are also VPL methods focusing on real-time applications [26], [27], [28]. They often ignore occlusion when computing indirect lighting. To take account of visibility, Ritschel et al. [29] proposed the *Imperfect Shadow Map* to accelerate the construction of shadow maps for VPLs. Dong et al. [30] clustered VPLs into large area lights and determined visibility using soft shadowing techniques. Bashford-Rogers et al. [31] used visibility grids to approximate geometry and accelerate multibounce indirect illumination. For efficiency, these methods pursue visually pleasing results rather than physically accurate ones. They are validated by a perceptual study [32] in the context of interactive VPL-based rendering.

2.3 Visibility Algorithms

Several methods have been proposed to reduce the number of shadow tests. Agrawala et al. [33] exploited image-space coherence to efficiently render soft shadows from area lights. Ben-Artzi et al. [34] further extended the idea to reduce shadow tests for environment lighting. Shirley et al. [35] accelerated rendering by only sampling the visibility of strong lights. These methods suffer from artifacts when most strong lights are occluded. Hart et al. [36] reduced the cost of visibility sampling by identifying the occluded region of a light source. They stored occluders in a map and reprojected them onto light sources for culling out the occluded part. This algorithm observes good performance for simple scenes, but does not scale well with geometry and light complexity.

There are also algorithms for approximating visibility using dedicated graphics hardware or point-based representations. Stewart and Karkanis [37] used item buffer and graph relaxation to construct approximated visibility maps. Dutré et al. [38] represented geometry as point clouds and used them for visibility estimation. These methods can only achieve limited success for complex scenes.

The *Local Illumination Environment* (LIE) [39] intends to reduce both the frequency and cost of shadow computation. A LIE represents a part of the scene (an octree cell in their implementation), combined with lists of lights that are fully visible, fully occluded, and partially visible during rendering. Occluders for partially visible lights are also recorded. Contributions from fully visible lights are gathered without tracing shadow rays. Visibility tests for partially visible lights are accelerated by only testing with the recorded occluders. LIE also does not scale well when the number of lights and triangles grows. It takes a long time to assort the lights, and significant amount of memory is required to store the occluders.

Donikian et al. [40] used an adaptive scheme to iteratively accumulate visibility information. They combine uniform, block-level, and pixel-level probability density functions with different weights at different iterations for light sampling. Their method is very robust; however, it has an expensive start-up cost because uniform sampling is used in early stages. Recently Ramamoorthi et al. [41] investigated the impact of sampling patterns on the quality of soft shadows. They offer guidelines for the strategy of sampling area lights with different shapes.

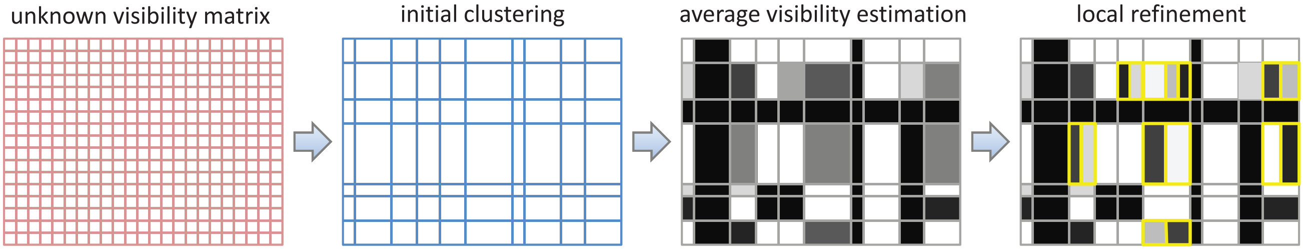


Fig. 3. Construction of VisibilityClusters. We first form initial VisibilityClusters by clustering the lights and shading points based on their geometric properties independently. Next, we estimate the average visibility of each submatrix (for a pair of LC and SC) by sampling few shadow rays. Finally, we locally refine the VisibilityClusters with large visibility variance (highlighted in yellow) by splitting LCs.

2.4 Directional Occlusion

Our method can be used for enhancing local shading with directional occlusion. In this context, Ritschel et al. [42] proposed a screen space method for interactive applications. It produces visually pleasing results but fails to account for occluders outside the view frustum. The practical filtering method proposed by Egan et al. [43], on the other hand, is designed for high-quality rendering. Our method strikes a balance between performance and quality, and also offers a good solution for lighting previews.

3 Background

In this section, we outline the many-light rendering problem which the VisibilityCluster algorithm aids solving. Prior work has shown that indirect illumination, environment lighting, and many direct light sources can all be approximated by many VPLs [18], [19], [22]. Assume that all sources of illumination are represented by a set of point lights, S. For a shading point x_o, its reflected radiance L_o along the viewing direction ω can be computed by summing contributions from all lights in S:

$$L_o(x_o, \omega) = \sum_{x_i \in S} L_i(x_i) f_r(x_i \rightarrow x_o, \omega) V(x_i) G(x_i) \cos(\theta_i), \quad (1)$$

where x_i is a point light in S, L_i is its incident radiance, f_r is the surface BRDF of x_o, V is the binary visibility function between x_o and x_i, and $G(x_i)$ is the subtended solid angle at x_o by x_i. By combining $G(x_i)$ and $\cos(\theta_i)$ into BRDF and hiding x_o and ω for a fixed shading point, the equation can be simplified as

$$L_o = \sum_{x_i \in S} L_i(x_i) f_r(x_i) V(x_i). \quad (2)$$

As the number of lights increases, directly summing the contributions of all lights becomes impractical. Monte Carlo method and importance sampling are often employed to efficiently estimate (2) with a small number of samples N:

$$L_o \approx \frac{1}{N} \sum_{s=1}^{N} \frac{L_i(x_s) f_r(x_s) V(x_s)}{p(x_s)}, \quad (3)$$

where p is the probability mass function (importance function) used to draw samples. Variance can be effectively reduced by choosing p that approximates the product of lighting, BRDF, and visibility. As it is very expensive to even obtain approximation of visibility, most previous work assumes uniform visibility and only approximates the product $L_i f_r$ for p. Unfortunately, the constant visibility assumption often fails for complex scenes with nonnegligible visibility variations. The proposed VisibilityCluster method can estimate visibility more efficiently. Therefore, it becomes feasible to incorporate visibility into p for more effective variance reduction. We call it triple-product importance sampling because the product $L_i f_r V$ is approximated.

4 VisibilityCluster

Our goal is to construct VisibilityClusters with high visibility coherence: all shading points in an SC have similar visibility functions toward the lights in an LC. A straightforward way is to start with two large clusters containing all lights and shading points, respectively, estimate their visibility variance, and split them into smaller ones if the variance is too large. However, because each iteration of the variance estimation demands expensive shadow testing, the construction cost is not acceptable. To find a good compromise between accuracy and efficiency, we propose a two-pass approach described in this section.

Fig. 3 demonstrates the flowchart of the two-pass method. In the initial clustering stage, we cluster lights and shading points separately according to their geometric properties. This is based on the observation that locally close shading points often behave similarly to a group of lights within a small solid angle. For each pair of SC and LC, we estimate the average visibility of their VisibilityCluster by sampling lights and shading points uniformly within each cluster. Finally, based on the estimated average visibility, local refinement is performed for VisibilityClusters with large visibility variance by splitting the LCs.

4.1 Initial Clustering

In the initial clustering stage, we separately cluster lights and shading points based on their geometric properties and use those to create initial VisibilityClusters.

Light clustering. We cluster lights using a modified IlluminationCut approach [44], [2] for its native support of hierarchical refinement. The original IlluminationCut starts by building a binary tree for all lights based on locations and directions. Each leaf corresponds to an individual point light while each internal node represents a cluster of lights. For each LC L_k, we record its bounding box, the mean $\langle L_k \rangle$, and the variance $\mathrm{Var}(L_k)$ of illumination for all lights within the cluster.

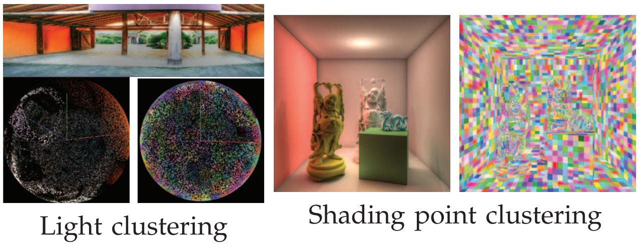


Fig. 4. Light clustering and shading point clustering. Left: The original environment map (top) is represented by 32K VPLs (bottom left) which make up 1,200 clusters (bottom right, lights with the same pseudo color belong to the same cluster). Right: Shading points are grouped into 2,400 clusters.

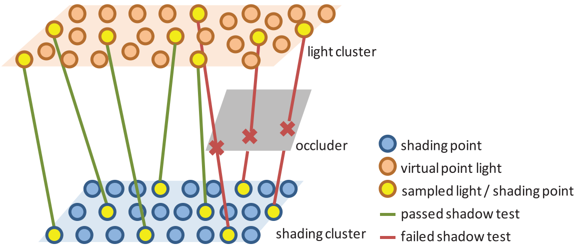


Fig. 5. Average visibility estimation. For an SC and an LC, we sample shading points and lights uniformly and evaluate their visibility. The average visibility between the two clusters is approximated as the ratio of unoccluded shadows rays to all shadow rays. For this particular example, it is $5/8 = 0.625$.

Any cut to the binary tree defines a clustering configuration. In our case, the criterion for selecting a cut is to keep the variance of both luminance and visibility low. Starting from the root, we keep selecting a node and splitting it into its two children. Since at this stage we do not have the results of visibility testing, there is no guarantee that visibility variance will be reduced by splitting nodes. We use a heuristic approach that works well in practice to partition the tree. Assume we want to create $N_c + 1$ clusters. N_c splits are required. For the first $\lfloor k \times N_c \rfloor$ $(k \leq 1)$ splits, we select nodes according to their luminance values. The cluster (node) with the largest luminance variance $\mathrm{Var}(L_k)$ is replaced with its two children in each iteration. For the rest of splits, we focus on splitting clusters with large spatial extents because they potentially have large visibility variations. We select the cluster with the largest axis extent to split until the number of clusters has been reached. In practice, we found that $k = \frac{2}{3}$ works well for all kinds of illumination conditions.

Shading point clustering. Shading points are clustered in a top-down partitioning manner with a kd-tree as in previous methods [20], [25]. First, we construct a tree for all shading points using 6D coordinates including both positions and surface normals. Next, similar to light clustering, we compute a shading cut by iteratively replacing the node with the largest axis extent with its two children, until reaching the target number of SCs. The final shading cut represents the clustering configuration.

We have experimented with other methods, such as k-means clustering, applied to grouping lights and shading points. With the same budget of the cluster size, different clustering methods produce results of similar rendering quality. We adopt the top-down splitting approaches, IlluminationCut and kd-tree, for better efficiency. Fig. 4 gives examples of light and shading point clustering.

4.2 Average Visibility Estimation

After clustering, we estimate the average visibility for each VisibilityCluster. Assume there are N lights in the LC and M shading points in the SC, the true average visibility is obtained by performing $N \times M$ shadow tests and counting the ratio of unoccluded rays to all rays, which is computationally expensive. Fortunately, because of visibility coherence within a VisibilityCluster, the average visibility can be estimated accurately using only a few shadow rays (in our implementation, usually 10-12). The shadow rays are generated by sampling the lights and shading points uniformly in their clusters (Fig. 5). The average visibility of a VisibilityCluster is approximated as the ratio of unoccluded shadow rays to all sampled shadow rays.

4.3 Local Refinement

The estimated average visibility ($\bar{V}$) from the previous step also serves as a quality measure to visibility estimation because it is related to visibility variance. The visibility variance of a VisibilityCluster C_k is calculated as follows:

$$\mathrm{Var}(C_k) = \frac{1}{n}\sum_{i=1}^{n}(V_i - \bar{V})^2, \tag{4}$$

where n is the number of shadow tests and V_i is the individual result of a shadow test. Since $V_i \in \{0, 1\}$, we have $V_i^2 = V_i$. In addition, $\frac{1}{n}\sum_{i=1}^{n} V_i = \bar{V}$. As a result, we have

$$\mathrm{Var}(C_k) = \bar{V} - \bar{V}^2 = \bar{V}(1 - \bar{V}). \tag{5}$$

From the other point of view, if V_i follows a Binomial distribution with the success probability $\bar{V}$, the sample variance is $\bar{V}(1 - \bar{V})$. Thus, by using (5), we can estimate visibility variance from the estimated average visibility. For VisibilityCluster whose visibility variance is larger than a predefined threshold σ_v ($\mathrm{Var}(C_k) \geq \sigma_v$), the current clustering configuration is not good enough and the visibility estimation might be inaccurate. We replace the LC with its two children in the light tree and re-estimate the average visibility for each one. The refinement continues until the visibility estimation for all VisibilityClusters are accurate enough or the maximal depth has been reached. Note that local refinement is performed per SC. Thus, as shown in Fig. 3, SCs (rows in the matrix) could have different light clustering configurations (column configurations). Fig. 6 demonstrates the improvement of visibility coherence with local refinement.

It is worth mentioning that for local refinement we only split LCs but not SCs because of the cost of building importance functions. For importance sampling, for each shading point, we need to determine its SC and calculate the corresponding cumulative distribution function (CDF) for all LCs (Fig. 7d). Splitting SCs generates additional SCs, each requiring its own CDF (Fig. 7e). As a result, the number of SCs grows fast with the refinement depth, making the CDF calculations very expensive. Thus, we choose to split the LCs only (Fig. 7c). In practice, we found that our choice of only splitting LCs has little impact on the

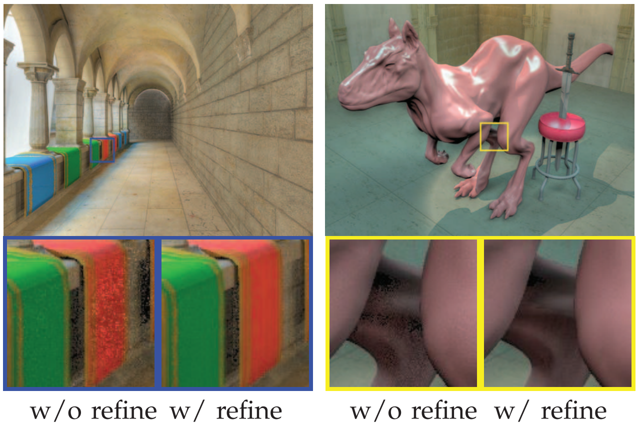


Fig. 6. Equal-time comparisons of rendering using the initial VisibilityClusters (w/o refine) versus the locally refined ones (w/refine). Local refinement improves rendering quality in regions with difficult visibility by reducing visibility variation within a VisibilityCluster.

quality of VisibilityClusters, but makes CDF construction (Fig. 7f) much more efficient and memory-friendly.

4.4 Bias Avoidance

For the case that all shadow tests return occluded, assigning zero visibility dogmatically introduces bias. To avoid bias, an additional test is performed to check whether the LC falls on the other side of the hemisphere of the SC. If so, the average visibility is set to 0; otherwise, it is set to a small value ϵ. We determine ϵ for an SC by estimating its average accessibility p_v (computed by averaging its visibility ratios to all LCs). ϵ is set to $(1 - p_v)^n$, where n is the number of performed shadow tests. This value accounts for the probability of failing n times in shadow tests if the average probability is p_v.

5 Experiments

We have implemented the proposed VisibilityCluster algorithm on PBRT2 [45] for two applications: triple-product importance sampling and directional occlusion. Experiments were performed on a machine with Intel Xeon 5420 CPU (2.5 GHz) and 32-GB RAM.

Table 1 lists the scene profiles and rendering settings for all test scenes. The number of initial LCs was set according to the illumination complexity. For most scenes, it was set to about 1K as suggested by previous work [2]. High-frequency lighting, as observed in the LUNCH scene (Fig. 10), might require more clusters. Similarly, the number of SCs was set based on the geometry complexity. In our experiments, 6,400 clusters usually gave good results. More accurate visibility estimation using more shadow rays allows fewer lights to be sampled for estimating (3) in the same amount of time. Thus, we need to find a good tradeoff between the accuracy of the visibility estimate and the number of allowed sampled lights. As demonstrated in Fig. 8, 8-12 shadow rays provided a good compromise between accuracy and performance. Finally, for local refinement, we set σ_v to 0.16 (equivalently, the refinement is performed if the estimated average visibility is between 0.2 and 0.8) and the maximal refine depth to 2 for all test scenes in this paper.

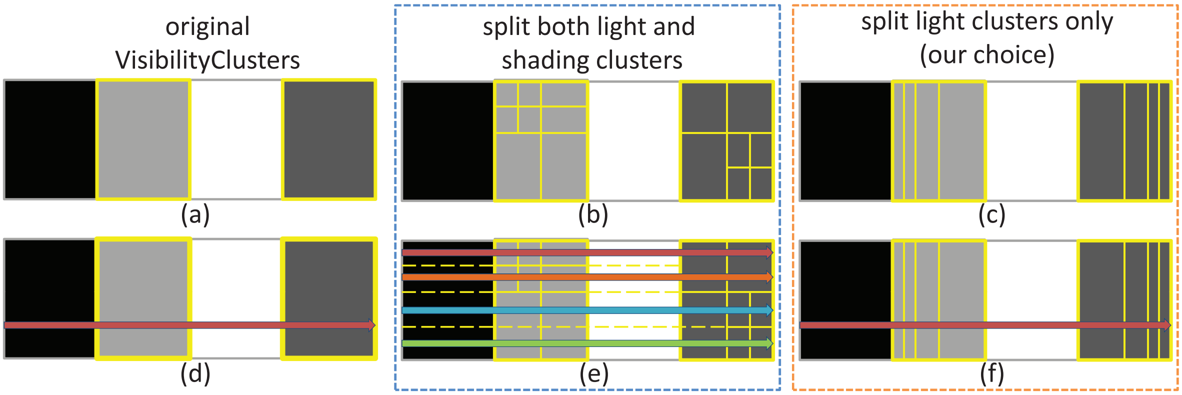

original VisibilityClusters
split both light and shading clusters
split light clusters only (our choice)
(a)
(b)
(c)
(d)
(e)
(f)

Fig. 7. The choice of splitting for local refinement. Part (a) shows the original VisibilityClusters for an SC. We want to refine the second and fourth ones (outlined in yellow). Part (b) is a refinement of both LCs and SCs. Part (c) is a refinement of LCs only. Importance sampling requires calculating a CDF over all LCs, as shown in (d). Splitting SCs as in (b) would result in additional SCs, each with its own CDF (the rows with different colors in (e)). We found splitting LCs only as in (c) produces results of similar quality but is significantly more efficient and memory-friendly for building importance functions (f).

TABLE 1
Scene Profiles and Rendering Settings

Scene	Resolution (#SPP)	Faces	Lights (Type)	LC (Avg.LC)	SC	Time for VC (Ratio)
TOWN	1600 x 1200 (4)	818,047	32K (Env.)	600 (697)	6,400	19.5 sec. (13%)
LUNCH	1600 x 1200 (4)	180,807	32K (Env.)	1,200 (1321)	6,400	24.1 sec. (12%)
KILLEROO	1600 x 1200 (4)	744,140	32K (Env.) + 1 (Spot)	700 (748)	6,400	24.3 sec. (8%)
SPONZA	1600 x 1200 (4)	279,163	8K (Env.) + 70K (Ind.)	1,400 (1464)	6,400	108.5 sec. (18%)
ROOM	1600 x 1200 (8)	462,878	40K (Ind.) + 1(Spot)	800 (914)	6,400	41.7 sec. (14%)
CONSTRUCTION	1600 x 1200 (4)	272,566	65K (Env.)	800 (935)	6,400	58.4 sec. (12%)
HAIRBALL	1600 x 1200 (4)	2,893,834	32K (Env.)	800 (972)	6,400	65.9 sec. (8%)

The columns list the image resolution (the number of samples per pixel is listed in parenthesis), the number of triangles, the number of VPLs and the illumination type (Env. for environment lights, Spot for spot lights and Ind. for indirect illumination), the number of initial LCs and the average number of LCs after local refinement (Avg.LC), the number of SCs, and the time for computing VisibilityCluster (and its ratio to the total rendering time).

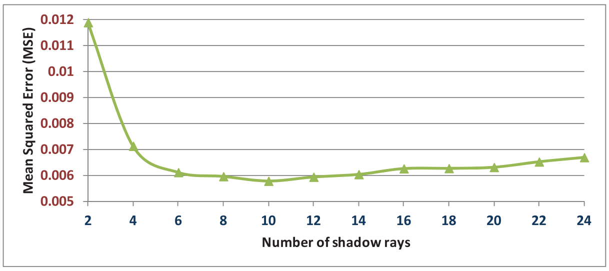


Fig. 8. The experiment to determine the number of shadow rays for the average visibility estimation in the LUNCH scene. To find a good balance between computation time and accuracy, we carefully tune shadow ray and sampled light numbers while keeping the rendering time at 200 seconds. The plot of the number of shadow rays versus MSE indicates that 8-12 shadow rays provide a good compromise.

5.1 Triple-Product Importance Sampling

VisibilityCluster can be easily combined with BIS [2] to offer a more effective triple-product importance sampling solution. We use BIS to obtain bidirectional importance for each LC. Then, the estimated average visibility is multiplied to form the triple-product importance function for sampling:

$$p(x_s) \sim |L_k|\langle L_k\rangle\langle \rho_k\rangle\langle V_k\rangle \cdot \frac{1}{|L_k|} = \langle L_k\rangle\langle \rho_k\rangle\langle V_k\rangle, \qquad (6)$$

where $|L_k|$, $\langle L_k\rangle$, $\langle \rho_k\rangle$, and $\langle V_k\rangle$ are the number of lights, the average luminance, the average BRDF, and the average visibility of the kth LC, respectively. To select a light from the set of VPLs S for shading, we first sample an LC using the triple-product importance $|L_k|\langle L_k\rangle\langle \rho_k\rangle\langle V_k\rangle$ and then uniformly sample a light from the selected LC.

5.1.1 Comparisons

We compared VisibilityCluster with BIS [2] and IC [3], which are state-of-the-art product and triple-product importance sampling methods, respectively. The number of sampled lights for each shading point was carefully set for each method to provide equal-time comparisons, including the time for computing importance records in IC and estimating average visibility in VisibilityClusters. As suggested by the IC authors, the number of importance records was set to 7K and the number of nearest records used by multiple importance sampling was set to 3.

The first two scenes, TOWN (Fig. 9) and LUNCH (Fig. 10), were rendered using direct lighting from environment illumination. TOWN was illuminated by a low-frequency cloudy sky and LUNCH was illuminated by high-frequency indoor lighting. For TOWN, BIS suffers from severe variance on the clock tower (Fig. 9, detail 1). Tall buildings act as occluders to each other and block the sun in the sky. BIS wastes most light samples toward the occluded sun because it has the strongest luminance. IC reduces variances by including the visibility term into the importance function. However, in the narrow passageway with only few importance records (Fig. 9, detail 2), the rendering looks quite noisy because the nearby records provide inaccurate information. Similar situations occurred in LUNCH. The image rendered with BIS is very noisy

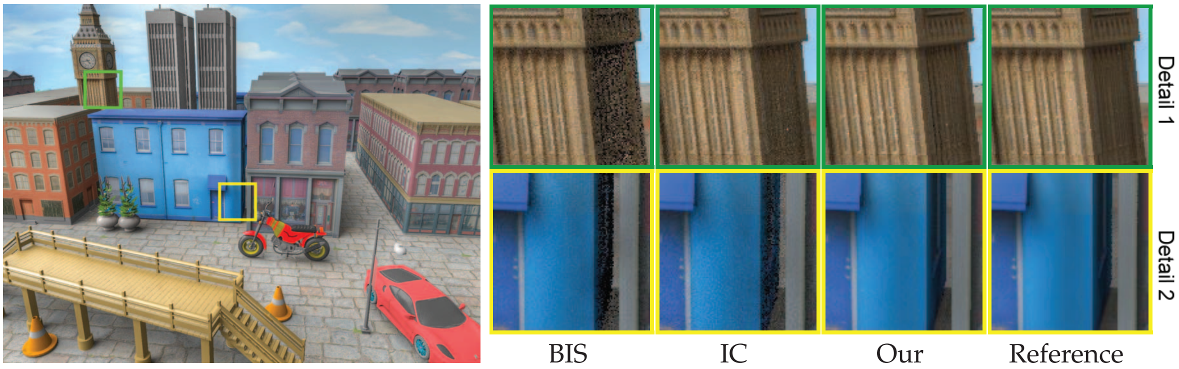


Fig. 9. TOWN (equal-time comparison, 150 seconds). Left: full image rendered by our approach. From column 2 to column 5: detailed images of BIS [2], IC [3], our approach, and the reference. Our method provides better noise reduction, where there is severe occlusion (detail 1) or high-frequency geometry variation (detail 2).

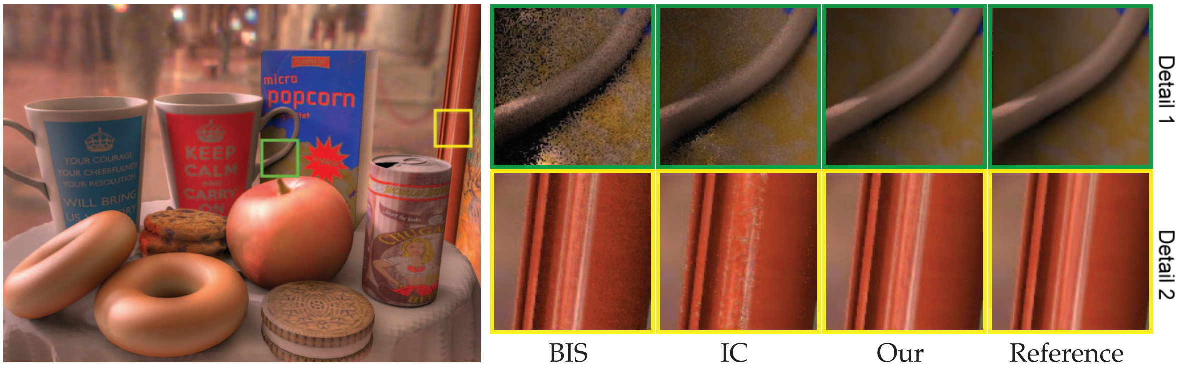


Fig. 10. LUNCH (equal-time comparison, 200 seconds). Left: full image rendered by our approach. From column 2 to column 5: detailed images of BIS [2], IC [3], our approach, and the reference. BIS produces severe noise due to occlusion of strong lights (detail 1). IC cannot capture well high-frequency BRDFs and fine geometry details due to its sparse records (detail 2).

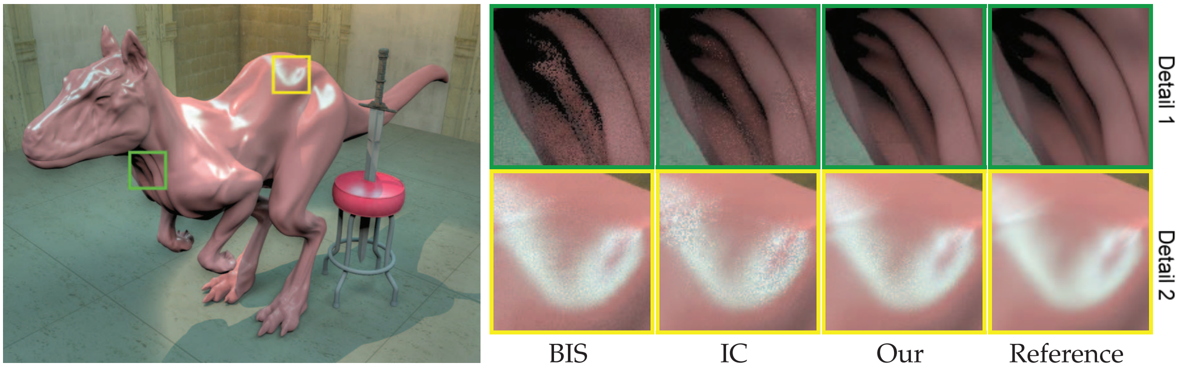


Fig. 11. KILLEROO (equal-time comparison, 300 seconds). Left: full image rendered by our approach. From column 2 to column 5: detailed images of BIS [2], IC [3], our approach, and the reference. BIS fails in regions with complex visibility variation (detail 1). IC is inefficient for glossy materials (detail 2). Our method is more effective for both cases.

since bright lights are occluded by the oil painting (Fig. 10, detail 1). IC outperforms BIS in most areas, but still observes significant noise in areas with large visibility variation, for example, the partially occluded regions on the table (Fig. 16). Its sparse records also cannot capture the fine geometry variation on the painting's frame (Fig. 10, detail 2). Our approach alleviates these problems by properly clustering lights and shading points to better exploit the coherence in visibility.

The third scene, KILLEROO (Fig. 11), was illuminated by a bright spot light and a skylight environment map. The scene demonstrates a very difficult case because most illumination is occluded by the surrounding walls. It also contains glossy materials. BIS works well on the killeroo's lower back (detail 2, variance due to BRDF variation) but poorly on the neck (detail 1, variance due to visibility variation). IC performs in the opposite way. The highly glossy BRDF and complex curvature on the back invalidate the distributions stored at sparse importance records. Our method is much less sensitive to geometric and material variations and significantly reduces noise on both the back and neck.

The next two scenes, SPONZA and ROOM, were designed to experiment with indirect lighting. SPONZA (Fig. 12) was rendered with an environment map (approximated with 8K lights) and 70K indirect VPLs. In this scene, only a small number of VPLs are visible to any shading point. BIS is ineffective because it cannot locate those visible lights that make contributions to the shading point. In this scene, our method achieves great improvement compared to BIS, but is slightly less effective than IC because we only use cluster-level importance. Lights within a cluster are still sampled uniformly, rather than by their importance. On the other hand, IC computes individual contributions for each light at every importance record. Thus, when rendering flat regions without much visibility variation, IC generates less noise than VsibilityCluster (detail 1). However, IC renders the end of the alley less effectively (detail 2), where it lacks importance records. For the ROOM scene (Fig. 13), IC produces noises on the boundary of the two dragons, where the complex geometry invalidates the importance records (detail 1). Our method, in contrast, can better cope with glossy materials and geometry complexity.

The last two scenes, CONSTRUCTION (Fig. 14) and HAIRBALL (Fig. 15), demonstrate situations with complex illumination and geometry. In CONSTRUCTION, there are lots of surrounding lattices, handrails, and trestles, resulting in large visibility variations. The lights coming from the right side are blocked by a group of pillars and shine through the gaps to a bump mapped ground. For this complex scene, our method produces noticeable noise but still outperforms the other two methods by a margin. In HAIRBALL, the complex self-occlusion makes BIS and IC very ineffective. It is worth noting that IC does not work better than BIS in this scene because of the highly complex geometry.

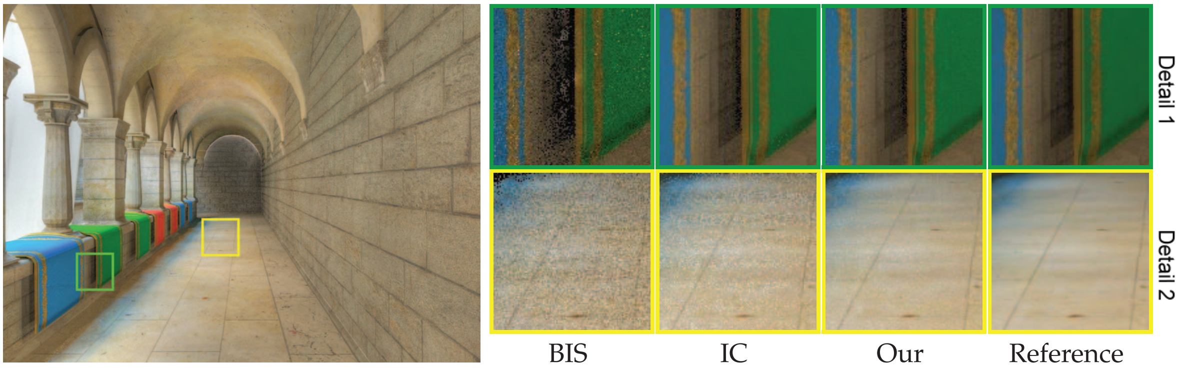


Fig. 12. SPONZA (equal-time comparison, 600 seconds). Left: full image rendered by our approach. From column 2 to column 5: detailed images of BIS [2], IC [3], our approach, and the reference. IC achieves the best noise reduction in flat regions (detail 1), but is less effective for regions that are far away from the camera (detail 2) due to the insufficient number of records. Our method reduces noise consistently across the whole image.

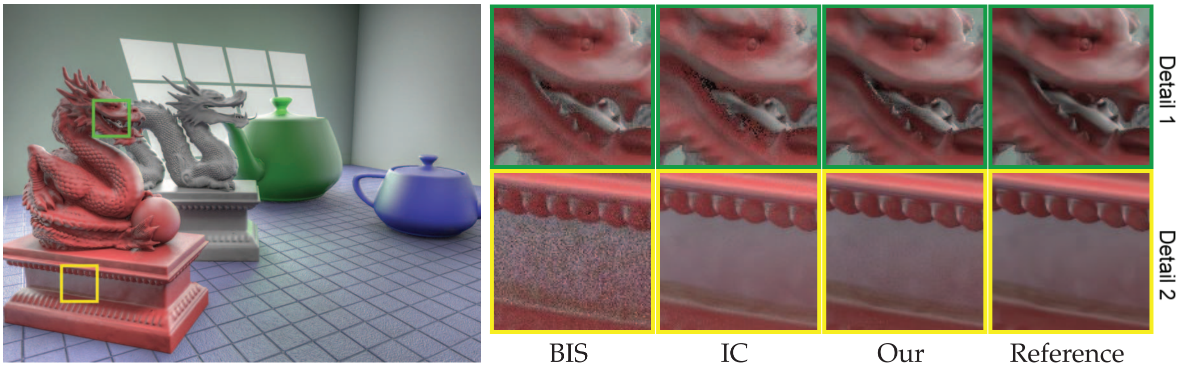


Fig. 13. ROOM (equal-time comparison, 300 seconds). Left: full image rendered by our approach. From column 2 to column 5: detailed images of BIS [2], IC [3], our approach, and the reference. Because of the fine geometry details of dragons, IC's importance records are less accurate when used by nearby shading points without records (detail 1). Our method exploits geometry coherence within SCs and handles scenes with fine geometry details better.

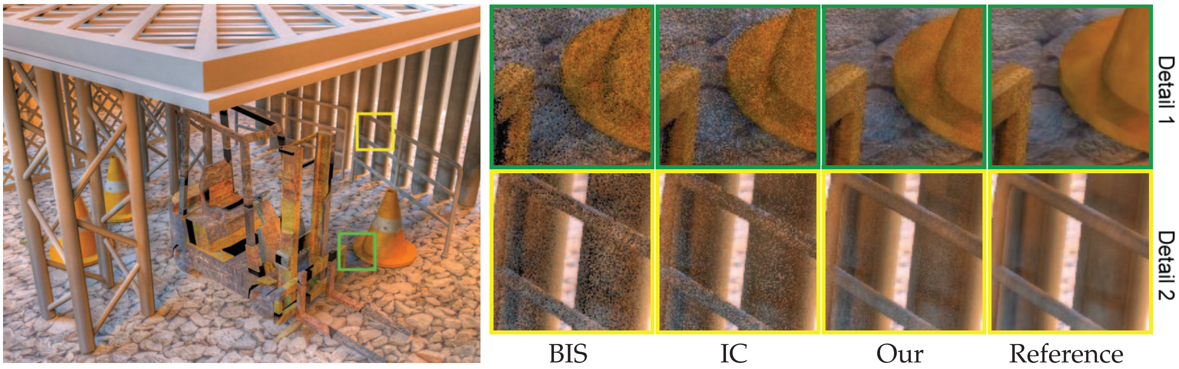


Fig. 14. CONSTRUCTION (equal-time comparison, 500 seconds). Left: full image rendered by our approach. From column 2 to column 5: detailed images of BIS [2], IC [3], our approach, and the reference. The grating patterns produced by pillars and the thin geometry such as lattices, handrails, and trestles result in high-frequency visibility variation, making BIS and IC very inefficient. Our method performs much better in this difficult scene.

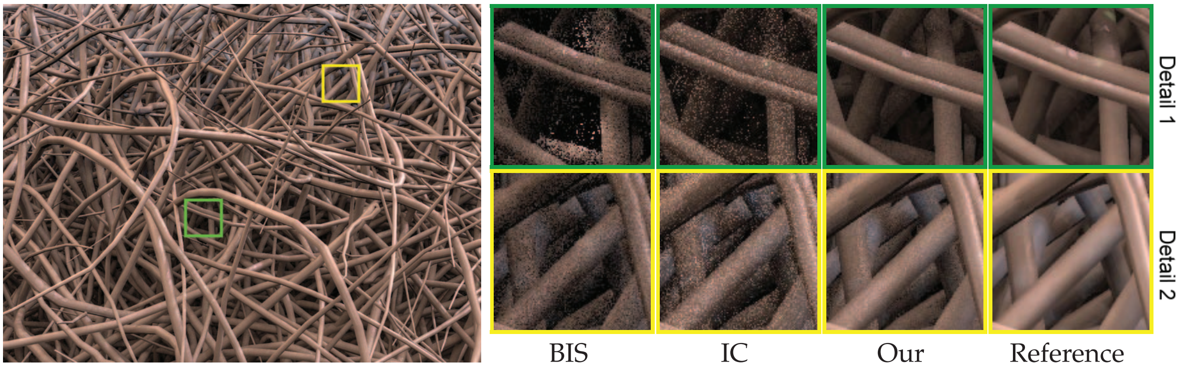


Fig. 15. HAIRBALL (equal-time comparison, 800 seconds). Left: full image rendered by our approach. From column 2 to column 5: detailed images of BIS [2], IC [3], our approach, and the reference. The complex geometry of hairball results in large visibility variation, causing relatively large noise in all methods. However, because VisibilityClusters are computed and refined for each SC, the proposed algorithm still achieves superior noise reduction compared to BIS and IC.

Fig. 16 visualizes errors, and lists the mean squared error (MSE) for all compared methods and scenes. In most scenes (excepts for SPONZA), our method has significantly lower MSE than BIS and IC. IC is slightly better than our method for SPONZA because the scene contains mostly flat surfaces.

5.1.2 Discussions

BIS generally converges fast in unoccluded regions with a few samples, but becomes inefficient where there is occlusion. The worst case occurs when the directions of strong lights or BRDF peaks are occluded. In this circumstance, BIS becomes very inefficient because most samples are wasted on directions without contributions.

IC greatly alleviates variance in areas with smooth surfaces. However, for more complex surfaces and glossy materials, the uniform distribution of importance records in the image plane could miss fine details or scene features, resulting in less effective variance reduction in these areas. Finally, it does not scale well to the number of VPLs. Since the full contributions need to be evaluated and stored for every importance record, the time of drawing samples and the memory requirements quickly grow as the number of VPLs increases.

VisibilityCluster significantly reduces variance for scenes with different illumination conditions and complex

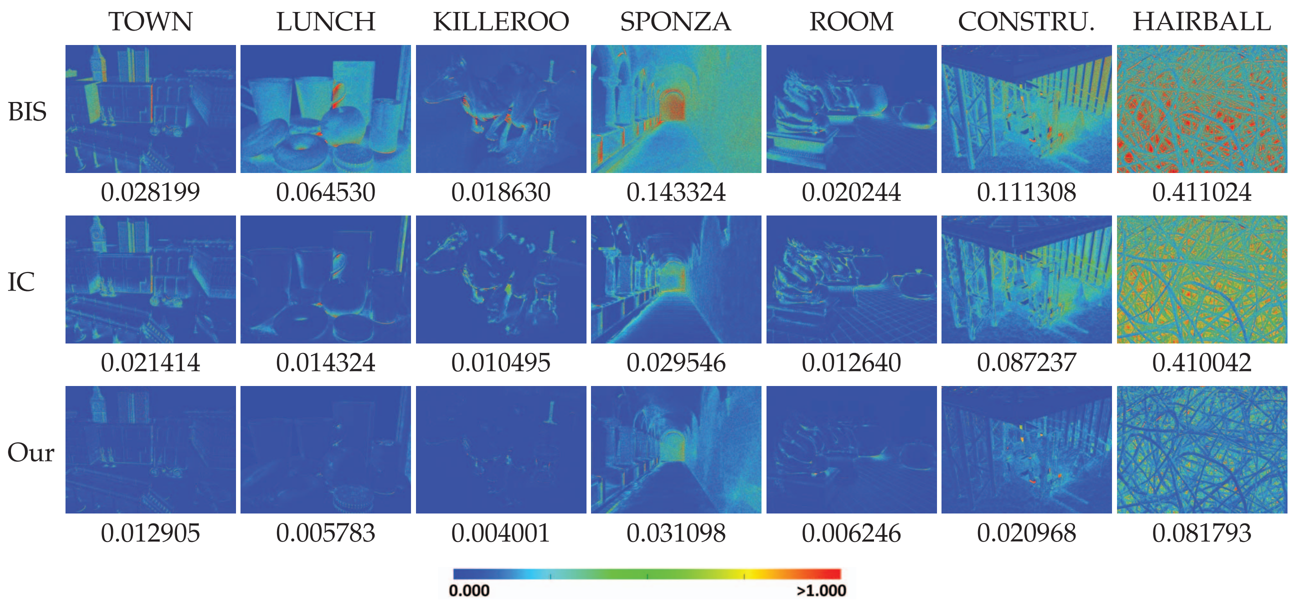


Fig. 16. Visualization of errors. Rows from top to bottom: BIS [2], IC [3], and our approach. The value under each image is its MSE.

geometry. Its scalability to the geometric complexity is improved by grouping similar shading points; similarly, clustering lights and only estimating average visibility improve the scalability in the number of VPLs.

5.1.3 Limitations

The SPONZA scene (Fig. 12) reveals a limitation of our method. In this scene, very few lights are visible for a shading point, making our cluster-level importance sampling less effective. The same problem also appears inside the forklift in the CONSTRUCTION scene (Fig. 16). Another limitation is that in very simple scenes with little occlusion, the overhead of computing VisibilityCluster might not justify its gain.

5.1.4 Memory Usage

The memory usage required for VisibilityCluster is $N_{SC} \times N_{LC} \times 4$ bytes, where N_{SC} and N_{LC} are the number of SCs and LCs, respectively. For 6,400 SCs and 1,200 LCs (the average case in this paper), it uses about 30 MB. The construction of the shading tree and the light tree requires additional memory, but it can be released once the clusters are determined. Compared to IC, our memory requirement is significantly smaller.

5.2 Directional Occlusion

VisibilityCluster can also be used to approximate directional occlusion for lighting previews. A row in the VisibilityCluster matrix (for a shading point) gives the average visibility of each LC with respect to the shading point. It can be used as an approximation of directional occlusion. In this setting, we approximate the reflected radiance in (2):

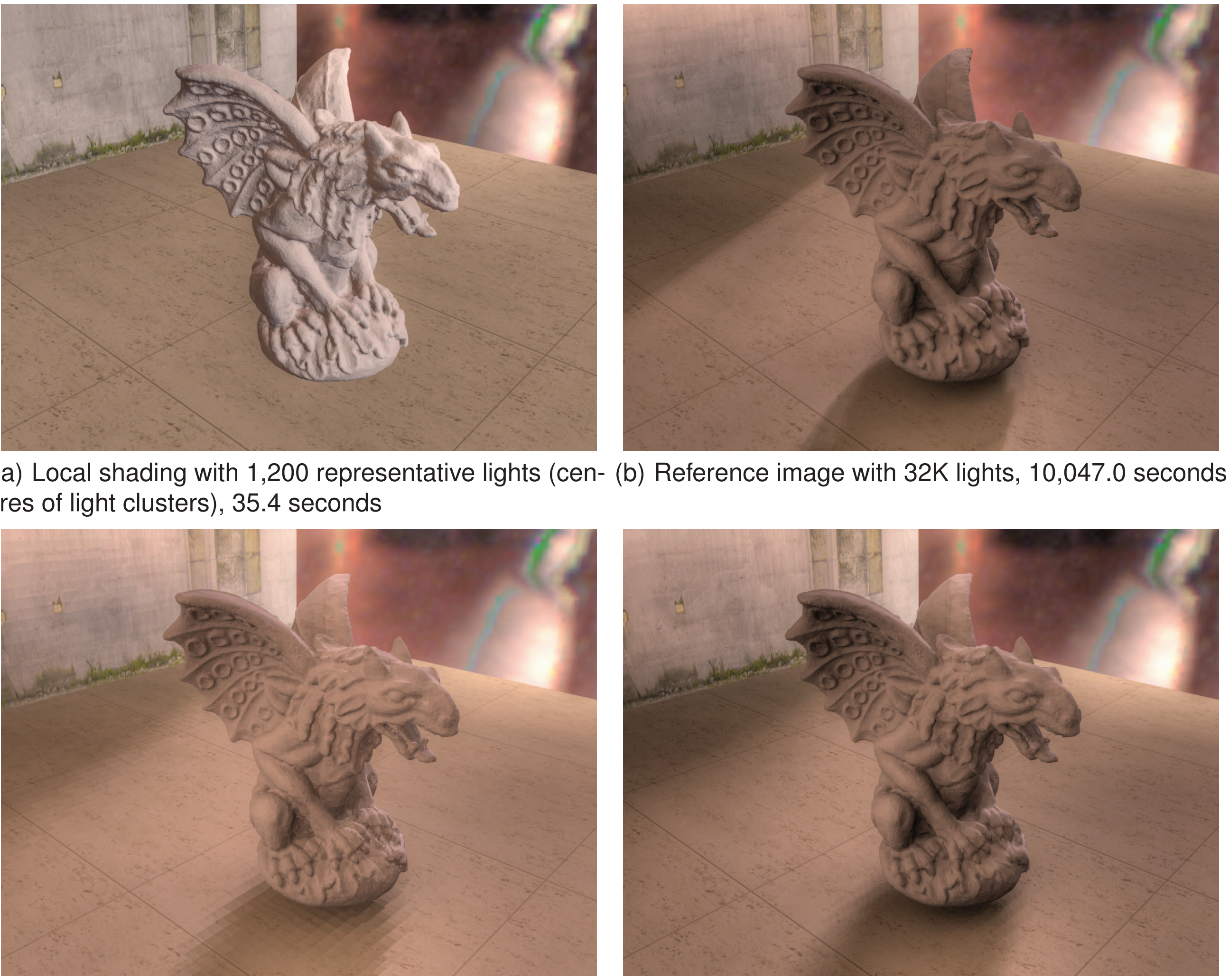
$$L_o \approx \sum_{L_s} \tilde{L}_i(L_s)\bar{f}_r(L_s)\bar{V}(L_s), \tag{7}$$

where L_s is a LC, $\tilde{L}_i(L_s)$ is its total illuminance, $\bar{f}_r(L_s)$ is the average BRDF from the LC to the shading point, and $\bar{V}(L_s)$ is the average visibility between the LC and the shading point. Our method can be used for estimating $\bar{V}$.

Fig. 17 presents an example for enhancing local shading using VisibilityCluster. Fig. 17a gives the result of local shading without considering visibility to lights. It looks quite different from the reference (Fig. 17b) in both, shadows and tone. By incorporating the approximated visibility supplied by VisibilityCluster, Fig. 17c significantly improves the shadows and tone. However, since average visibility is computed per VisibilityCluster, directly using it for $\bar{V}$ could result in blocking artifacts (Fig. 17c). The problem can be alleviated by blending visibility terms of nearby VisibilityClusters. A kd-tree is built for the centers of all SCs. To determine $\bar{V}(L_s)$ for a shading point, we look up the n nearest SCs based on geometric properties in the kd-tree, and linearly blend their $\bar{V}(L_s)$ values using the reciprocals of distances as weights. We used $n = 3$ in our implementation. As Fig. 17d shows, although mild blocking artifacts are still visible, the directional shadows on the floor look better, and surface details on the statue are also greatly improved. With very little cost, by adding directional occlusion, VisibilityCluster significantly enhances the quality of local shading.

6 Conclusion and Future Work

This paper proposes VisibilityCluster, an efficient estimation and compact representation for averaged shading point visibility. We observe that by properly clustering shading points and lights, visibility entries in the many-light transport matrix exhibit a block-like structure and can be efficiently approximated by sampling. The submatrix formed by a cluster of shading points and a cluster of lights is called a VisibilityCluster. We demonstrate that the average visibility of a VisibilityCluster can be faithfully estimated with only few shadow tests. We also use the average visibility as an estimation of quality to guide further refinement of clustering for improving accuracy. The result is an efficient algorithm for estimating visibility



(a) Local shading with 1,200 representative lights (centres of light clusters), 35.4 seconds

(b) Reference image with 32K lights, 10,047.0 seconds

(c) Approximated directional occlusion with 1,200 light clusters using VisibilityCluster directly, 47.8 seconds

(d) Approximated directional occlusion with 1,200 light clusters using interpolated VisibilityCluster, 59.5 seconds

Fig. 17. The STATUE scene (220K triangles) illuminated by 32K lights (clustered into 1,200 LCs) and rendered with a $1{,}600 \times 1{,}200$ resolution (4 spp). (a) Local shading with 1,200 representative lights (centers of LCs) without considering visibility. (b) Reference image that accumulates contributions of all 32K lights with visibility included. (c), (d) Lighting previews using VisibilityCluster visibility and interpolated VisibilityCluster visibility for directional occlusion. Directly using VisibilityCluster results in blocking artifacts (c), which are alleviated by linearly combining visibility terms of nearby VisibilityClusters (d).

that can be used in applications such as importance sampling and directional occlusion.

One interesting future direction is to extend our method to the temporal domain for rendering animations. We are planning to investigate possible exploitation of temporal coherence to reduce per-frame cost. Another possible direction is to deal with glossy materials. It is interesting to employ advanced VPL methods that focus on glossiness [23], [24]. Finally, we would also like to combine our method with some recently proposed approaches for further speeding up visibility estimation, for example, using RTSAH [46] to accelerate shadow tests.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was partly supported by grants NTU102R7609-5 and NSC101-2628-E-002-031-MY3. They also want to thank the creators of the models and textures used in this paper: motorbike (Chris34), Ferrari (RenderHere), street lamps (jotijoti), buildings, street cones (Adrian3dartist), western buildings (Tippy), Big Bens (Skipper25), roundtable (dexterdoodle), popcorn box (Norse Graphics), donuts, sandwich cookie (Bluegenie2), cookies (slatcher), mugs (Skip1871), apple (ozzykitten666), beer can (ancestorsrelic), barstool (fusebulb), sword (offrench), forklift (Tim Fullwood), handrail (aewakened), trestle (birdy), lattice (MarcCCTx), downloaded from ShareCG.com; elevated walk (dogbite1066) from ShareAEC.com; plants, painting (hawk), rock texture (leo) from cgrealm.org; dragons from newcger.com; ruined wall from elfpix.com; Crytek Sponza (Marko Dabrovic and Crytek GmbH), hairball (Samuli Laine and Tero Karras) at NVIDIA Research, gargoyle (INRIA) via the AIM@SHAPE repository; kitchen environment map (Paul Debevec) from Light Probe Image Gallery and other environment maps by Dan Meyer. Finally, they would like to thank Tzu-Mao Li and Ken-Yi Lee for the discussion in the early stage of this paper and Dominik Seifert for paper proofreading.

REFERENCES

[1] S. Boulos, "A Summary of Octree Ray Traversal Algorithms," *Ray Tracing News,* vol. 23, no. 1, July 2010.

[2] R. Wang and O. Åkerlund, "Bidirectional Importance Sampling for Unstructured Direct Illumination," *Computer Graphics Forum,* vol. 28, no. 2, pp. 269-278, Apr. 2009.

[3] I. Georgiev, J. Křivánek, S. Popov, and P. Slusallek, "Importance Caching for Complex Illumination," *Computer Graphics Forum,* vol. 31, no. 2, pp. 701-710, May 2012.

[4] S. Agarwal, R. Ramamoorthi, S. Belongie, and H.W. Jensen, "Structured Importance Sampling of Environment Maps," *ACM Trans. Graphics,* vol. 22, no. 3, pp. 605-612, July 2003.

[5] T. Kollig and A. Keller, "Efficient Illumination by High Dynamic Range Images," *Proc. Eurographics Rendering Workshop,* pp. 45-50, 2003.

[6] J. Steinhurst and A. Lastra, "Global Importance Sampling of Glossy Surfaces Using the Photon Map," *Proc. IEEE Symp. Interactive Ray Tracing,* pp. 133-138, 2006.

[7] J. Lawrence, S. Rusinkiewicz, and R. Ramamoorthi, "Efficient BRDF Importance Sampling Using a Factored Representation," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 496-505, Aug. 2004.

[8] P. Clarberg, W. Jarosz, T. Akenine-Möller, and H.W. Jensen, "Wavelet Importance Sampling: Efficiently Evaluating Products of Complex Functions," *ACM Trans. Graphics,* vol. 24, no. 3, pp. 1166-1175, July 2005.

[9] D. Cline, P.K. Egbert, J.F. Talbot, and D.L. Cardon, "Two Stage Importance Sampling for Direct Lighting," *Proc. 17th Eurographics Conf. Rendering Techniques (EGSR '06),* pp. 103-113, 2006.

[10] P. Clarberg and T. Akenine-Möller, "Practical Product Importance Sampling for Direct Illumination," *Computer Graphics Forum,* vol. 27, no. 2, pp. 681-690, 2008.

[11] W. Jarosz, N.A. Carr, and H.W. Jensen, "Importance Sampling Spherical Harmonics," *Computer Graphics Forum,* vol. 28, no. 2, pp. 577-586, Apr. 2009.

[12] D. Burke, A. Ghosh, and W. Heidrich, "Bidirectional Importance Sampling for Direct Illumination," *Proc. 16th Eurographics Conf. Rendering Techniques (EGSR '05),* pp. 147-156, 2005.

[13] J.F. Talbot, D. Cline, and P. Egbert, "Importance Resampling for Global Illumination," *Proc. 16th Eurographics Conf. Rendering Techniques (EGSR '05),* pp. 139-146, 2005.

[14] A. Ghosh and W. Heidrich, "Correlated Visibility Sampling for Direct Illumination," *Visual Computer,* vol. 22, nos. 9-11, pp. 693-701, Sept. 2006.

[15] F. Rousselle, P. Clarberg, L. Leblanc, V. Ostromoukhov, and P. Poulin, "Efficient Product Sampling Using Hierarchical Thresholding," *Visual Computer,* vol. 24, nos. 7-9, pp. 465-474, July 2008.

[16] P. Clarberg and T. Akenine-Möller, "Exploiting Visibility Correlation in Direct Illumination," *Computer Graphics Forum,* vol. 27, no. 4, pp. 1125-1136, 2008.

[17] D. Cline, D. Adams, and P. Egbert, "Table-Driven Adaptive Importance Sampling," *Computer Graphics Forum,* vol. 27, no. 4, pp. 1115-1123, 2008.

[18] A. Keller, "Instant Radiosity," *Proc. ACM SIGGRAPH '97,* pp. 49-56, 1997.

[19] B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian, and D.P. Greenberg, "Lightcuts: A Scalable Approach to Illumination," *ACM Trans. Graphics,* vol. 24, no. 3, pp. 1098-1107, July 2005.

[20] B. Walter, A. Arbree, K. Bala, and D.P. Greenberg, "Multidimensional Lightcuts," *ACM Trans. Graphics,* vol. 25, no. 3, pp. 1081-1088, July 2006.

[21] B. Walter, P. Khungurn, and K. Bala, "Bidirectional Lightcuts," *ACM Trans. Graphics,* vol. 31, no. 4, pp. 59:1-59:11, July 2012.

[22] M. Hašan, F. Pellacini, and K. Bala, "Matrix Row-Column Sampling for the Many-Light Problem," *ACM Trans. Graphics,* vol. 26, no. 3, article 26, July 2007.

[23] M. Hašan, J. Křivánek, B. Walter, and K. Bala, "Virtual Spherical Lights for Many-Light Rendering of Glossyc scenes," *ACM Trans. Graphics,* vol. 28, no. 5, pp. 143:1-143:6, Dec. 2009.

[24] T. Davidovič, J. Křivánek, M. Hašan, P. Slusallek, and K. Bala, "Combining Global and Local Virtual Lights for Detailed Glossy Illumination," *ACM Trans. Graphics,* vol. 29, no. 6, pp. 143:1-143:8, Dec. 2010.

[25] J. Ou and F. Pellacini, "LightSlice: Matrix Slice Sampling for the Many-Lights Problem," *ACM Trans. Graphics,* vol. 30, no. 6, pp. 179:1-179:8, Dec. 2011.

[26] C. Dachsbacher and M. Stamminger, "Reflective Shadow Maps," *Proc. Symp. Interactive 3D graphics and games,* pp. 203-231, 2005.

[27] C. Dachsbacher and M. Stamminger, "Splatting Indirect Illumination," *Proc. Symp. Interactive 3D Graphics and Games,* pp. 93-100, 2006.

[28] G. Nichols and C. Wyman, "Multiresolution Splatting for Indirect Illumination," *Proc. Symp. Interactive 3D Graphics and Games,* pp. 83-90, 2009.

[29] T. Ritschel, T. Grosch, M.H. Kim, H.-P. Seidel, C. Dachsbacher, and J. Kautz, "Imperfect Shadow Maps for Efficient Computation of Indirect Illumination," *ACM Trans. Graphics,* vol. 27, no. 5, pp. 129:1-129:8, Dec. 2008.

[30] Z. Dong, T. Grosch, T. Ritschel, J. Kautz, and H.-P. Seidel, "Real-Time Indirect Illumination with Clustered Visibility," *Proc. Vision, Modeling, and Visualization Workshop,* 2009.

[31] T. Bashford-Rogers, K. Debattista, C. Harvey, and A. Chalmers, "Approximate Visibility Grids for Interactive Indirect Illumination," *Proc. Third Int'l Conf. Games and Virtual Worlds for Serious Applications,* pp. 55-62, 2011.

[32] I. Yu, A. Cox, M.H. Kim, T. Ritschel, T. Grosch, C. Dachsbacher, and J. Kautz, "Perceptual Influence of Approximate Visibility in Indirect Illumination," *ACM Trans. Applied Perception,* vol. 6, no. 4, pp. 24:1-24:14, Oct. 2009.

[33] M. Agrawala, R. Ramamoorthi, A. Heirich, and L. Moll, "Efficient Image-Based Methods for Rendering Soft Shadows," *Proc. ACM SIGGRAPH '00,* pp. 375-384, 2000.

[34] A. Ben-Artzi, R. Ramamoorthi, and M. Agrawala, "Efficient Shadows for Sampled Environment Maps," *J. Graphics Tools,* pp. 13-36, 2006.

[35] P. Shirley, C. Wang, and K. Zimmerman, "Monte Carlo Techniques for Direct Lighting Calculations," *ACM Trans. Graphics,* vol. 15, no. 1, pp. 1-36, Jan. 1996.

[36] D. Hart, P. Dutré, and D.P. Greenberg, "Direct Illumination with Lazy Visibility Evaluation," *Proc. ACM SIGGRAPH '99,* pp. 147-154, 1999.

[37] A.J. Stewart and T. Karkanis, "Computing the Approximate Visibility Map, with Applications to Form Factors and Discontinuity Meshing," *Proc. Ninth Eurographics Workshop Rendering,* pp. 57-68, 1998.

[38] P. Dutré, P. Tole, and D.P. Greenberg, "Approximate Visibility for Illumination Computations Using Point Clouds," Technical Report PCG-00-01, Cornell Univ., 2000.

[39] S. Fernandez, K. Bala, and D.P. Greenberg, "Local Illumination Environments for Direct Lighting Acceleration," *Proc. 13th Eurographics Workshop Rendering,* pp. 7-14, 2002.

[40] M. Donikian, B. Walter, K. Bala, S. Fernandez, and D.P. Greenberg, "Accurate Direct Illumination Using Iterative Adaptive Sampling," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 3, pp. 353-364, May 2006.

[41] R. Ramamoorthi, J. Anderson, M. Meyer, and D. Nowrouzezahrai, "A Theory of Monte Carlo Visibility Sampling," *ACM Trans. Graphics,* vol. 31, no. 5, pp. 121:1-121:16, Sept. 2012.

[42] T. Ritschel, T. Grosch, and H.-P. Seidel, "Approximating Dynamic Global Illumination in Image Space," *Proc. Symp. Interactive 3D Graphics and Games,* pp. 75-82, 2009.

[43] K. Egan, F. Durand, and R. Ramamoorthi, "Practical Filtering for Efficient Ray-Traced Directional Occlusion," *ACM Trans. Graphics,* vol. 30, no. 6, pp. 180:1-180:10, Dec. 2011.

[44] E. Cheslack-Postava, R. Wang, O. Akerlund, and F. Pellacini, "Fast, Realistic Lighting and Material Design Using Nonlinear Cut Approximation," *ACM Trans. Graphics,* vol. 27, no. 5, pp. 128:1-128:10, Dec. 2008.

[45] M. Pharr and G. Humphreys, *Physically Based Rendering: From Theory To Implementation,* second ed. Morgan Kaufmann, 2010.

[46] T. Ize and C.D. Hansen, "RTSAH Traversal Order for Occlusion Rays," *Computer Graphics Forum,* vol. 30, no. 2, pp. 297-305, 2011.

Yu-Ting Wu received the BS and MS degrees from National Chiao Tung University, Hsinchu, Taiwan, in 2007 and 2009, respectively, all in computer science, and is currently working toward the PhD degree at the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. His research interests include computer graphics and visual effects.

Yung-Yu Chuang received the BS and MS degrees from National Taiwan University in 1993 and 1995, respectively, and the PhD degree from the University of Washington at Seattle in 2004, all in computer science. He is currently a professor with the Department of Computer Science and Information Engineering, National Taiwan University. His research interests include computational photography, multimedia, computer vision, and rendering. He is a member of the IEEE and the IEEE Computer Society.