Dual-Matrix Sampling for Scalable Translucent Material Rendering

Yu-Ting Wu, Tzu-Mao Li, Yu-Hsun Lin, Yung-Yu Chuang, *Member, IEEE,*

Abstract—This paper introduces a scalable algorithm for rendering translucent materials with complex lighting. We represent the light transport with a diffusion approximation by a dual-matrix representation with the *Light-to-Surface* and *Surface-to-Camera* matrices. By exploiting the structures within the matrices, the proposed method can locate surface samples with little contribution by using only subsampled matrices and avoid wasting computation on these samples. The decoupled estimation of irradiance and diffuse BSSRDFs also allows us to have a tight error bound, making the adaptive diffusion approximation more efficient and accurate. Experiments show that our method outperforms previous methods for translucent material rendering, especially in large scenes with massive translucent surfaces shaded by complex illumination.

Index Terms—Importance sampling, translucent materials, ray tracing.

1 Introduction

Translucent materials, such as jade, marble, milk, skin, meat, and leaves, are very common in real world. Correctly modeling and rendering their appearance is very important for realistic rendering. Different from surface reflection, light could enter such materials and scatter within them. Such subsurface scattering gives these materials distinct soft look. However, its simulation with Monte Carlo method is computationally expensive because of the needs for tracing a large number of long complex paths especially for highly scattering materials. The analytical dipole diffusion model [1] and later improved models [2], [3] provide efficient solutions for multiple scattering and have been widely adapted in industry and film production.

For efficiency, these diffusion models are usually evaluated with the two-pass approach proposed by Jensen and Buhler [4]. In the first pass, the irradiance values at a set of selected surface samples are precomputed and stored in a hierarchical structure. These pre-computed irradiance values are then adaptively gathered with the diffusion kernel at camera samples in the second pass. Although possessing the advantage of reusing precomputed irradiance, this approach still suffers from the problem that computation could be largely wasted on the unimportant surface samples which either are occluded by other objects or contribute very little to the final image. As a result, the precomputation of surface irradiance samples in the first pass often becomes the performance bottleneck, especially in a large scene with massive translucent objects shaded by complex illumination. Fig. 1(a) and 1(b) demonstrate two such examples: close-up rendering of a sculpture and a large exhibition with difficult occlusion. Arbree *et al.* [5] proposed an approach to address this efficiency issue. Based on the lightcuts framework [6], [7], their method performs adaptive clustering on light-surface-camera paths. Surface irradiance values are computed on demand by bounding the errors of path clusters. However, because visibility is not taken into account during the error estimation, errors could be over-estimated with their conservative bound when bright lights are occluded. This makes their approach less effective for scenes with backlit translucent objects (Fig. 1(c)) or difficult light paths (Fig. 1(d)). In addition, the time saving by adaptive computation in their method was often overshadowed by the overhead of tree refinement and the low hit-rate of the form factor cache. Finally, their single-pass strategy also makes the reuse of surface irradiance less intuitive and complicates the implementation.

This paper proposes an efficient method for scalable translucent material rendering, possessing the advantage of reusing pre-computed irradiance while largely reducing computation on unimportant surface irradiance samples. Inspired by recent matrix sampling and reconstruction methods [8], [9] (where light transport in the many-light setting is represented by a *Light-to-Camera* matrix), we decompose the light transport tensor (light-surface-camera) for translucent material rendering into two sub-matrices, the *Light-to-Surface* matrix and the *Surface-to-Camera* matrix. Fig. 2 visualizes the partial *Light-to-Surface* and *Surface-to-Camera* matrices of the *ChessGame* scene (Fig. 9). We observe

- *Y.-T. Wu, T.-M. Li, and Y.-Y. Chuang are with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 106. E-mail: {kevincosner|bachi|cyy}@cmlab.csie.ntu.edu.tw*

- *Y.-H. Lin is with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan, 106. E-mail: lymanblue@cmlab.csie.ntu.edu.tw*

(a) Sculpture (b) Exhibition (c) Cathedral (d) Room

Fig. 1: Several scenes inefficient to render with previous methods [4], [5]. In (a) and (b), only a small portion of the translucent surfaces show up in the final image due to a close-up viewing or occlusion. The traditional two-pass method [4] which computes surface irradiance in a view-independent manner could waste computation on unimportant surface samples. The single-pass approach based on lightcuts [5] alleviates this problem but does not work well for cases with backlit objects (c) or difficult light paths (d) as it completely ignores visibility.

that, by clustering properly, the *Light-to-Surface* matrix exhibits a good coherence structure with a low-rank property. The *Surface-to-Camera* matrix is very sparse. For each camera sample, there are few significant terms and these terms often group together. Our algorithm exploits the specific structures of both matrices for estimating the surface irradiance values and diffuse BSSRDFs adaptively. The advantages of our method are twofold. First, it is more efficient. The combined information from both matrices helps predict which surface samples could be less important during the adaptive diffusion gathering step. These samples can be largely approximated and calculated more quickly. Second, it is more accurate. The decoupled estimation of irradiance values and diffuse BSSRDFs allows us to have a tight error bound. The tight bound not only makes the prediction of surface importance more accurate, but also the adaptive diffusion approximation more efficient. Experiments show that our method significantly outperforms previous methods on a set of complex scenes with massive translucent objects and complex lighting.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 gives an overview of our method. Section 4 describes the procedure of dual-matrix sampling. Experiments are described in Section 5. Finally, Section 6 concludes the paper with directions for future work.

2 Related Work

In this section we briefly review previous work on modeling and rendering for subsurface scattering, with an emphasis on methods using diffusion approximation. Methods for translucent material acquisition and editing are out of the scope of this paper and not included here. In addition, we also include the most relevant work on many-light rendering in Section 2.3 since our method is inspired by matrix sampling approaches and adopts the many-lights formulation.

2.1 BSSRDF models for subsurface scattering

Subsurface scattering is usually modeled by the Bidirectional Subsurface Scattering Reflectance Distribution Function (BSSRDF) [10]. To compute the outgoing radiance L_o reflected at a point x_o along direction ω_o, we need to integrate the product of the BSSRDF S and the incoming radiance over the surfaces of translucent objects:

$$L_o(x_o,\omega_o) = \int_A \int_\Omega S(x_o,\omega_o;x_i,\omega_i)L(x_i,\omega_i)(n\cdot\omega_i)d\omega_i dA(x_i), \quad (1)$$

where $L(x_i,\omega_i)$ is the incident radiance at a point x_i along the incoming direction ω_i. Jensen *et al.* [1] split the BSSRDF into a single scattering term S^1 and a multiple scattering term S^d:

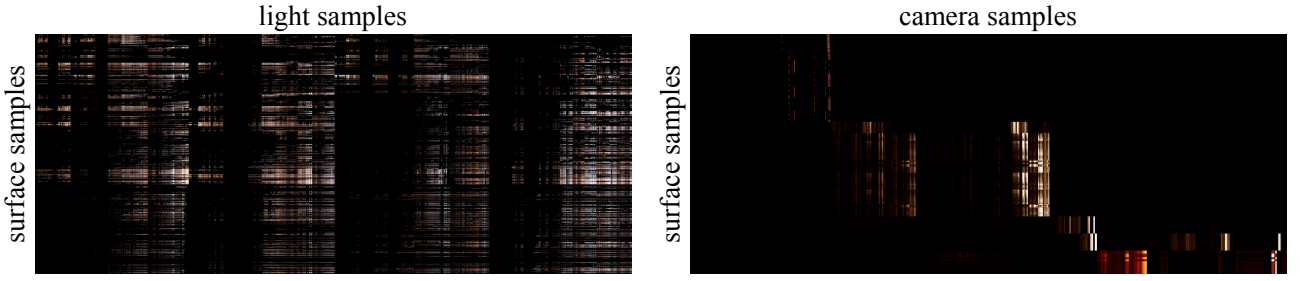
$$S(x_o,\omega_o;x_i,\omega_i) = S^1(x_o,\omega_o;x_i,\omega_i) + S^d(x_o,\omega_o;x_i,\omega_i). \quad (2)$$

The expensive multiple scattering term S^d can be efficiently approximated with the dipole source approximation and solved analytically with a diffuse BSSRDF R_d. The outgoing radiance due to the multiple scattering term is therefore approximated as:

$$L_o^d(x_o,\omega_o) = \frac{1}{\pi}F_t(\eta,\omega_o)\int_A R_d(\|x_o-x_i\|)\cdot \int_\Omega F_t(\eta,\omega_i)L(x_i,\omega_i)(n\cdot\omega_i)d\omega_i dA(x_i), \quad (3)$$

where F_t is the Fresnel transmittance and η is the relative index of refraction.

Later, several models have been proposed for improving the dipole model. Donner and Jensen [11] used the multipole model for rendering multi-layer transparent materials. They also proposed a hybrid model with photon diffusion to account for occlusion and caustics inside the medium [12]. Recently, quantized diffusion (QD) [2] and its analytical version, the



(a) The Light-to-Surface matrix (b) The Surface-to-Camera matrix

Fig. 2: Visualization of the Light-to-Surface and Surface-to-Camera matrices for the *ChessGame* scene in Fig. 9. As the figures show, the Light-to-Surface matrix has a low-rank structure while the Surface-to-Camera matrix is very sparse. Note that only partial matrices are shown and the Surface-to-Camera matrix is transposed for better fitting into this figure.

better dipole [3], have been proposed for improving the accuracy of dipole models. To further improve QD, Yan *et al.* [13] made correction of oblique illumination. Habel *et al.* [14] also proposed a correction factor to correct the overestimation of near-surface sources.

2.2 Subsurface scattering rendering

One of the most popular approaches for rendering with the dipole model is the two-pass method proposed by Jensen and Buhler [4]. In their approach, irradiance is first computed at surface samples and then adaptively gathered with a hierarchical structure at each camera sample. The method robustly generates smooth images by reusing irradiance values from a fixed set of surface samples.

An alternative for integrating over surface is to generate samples in the surroundings of camera samples on demand [15], [16]. This kind of methods do not require additional memory for storing surface samples, however, at the expense of less intuitive reuse of surface irradiance.

Caching-based approaches have also been proposed. Inspired by irradiance caching, Keng *et al.* [17] proposed to cache subsurface scattering in the image space and perform interpolation based on gradients. The multiresolution radiosity caching [18] instead caches irradiance values on vertices of micropolygons, which can be reused by nearby camera samples.

Some approaches combine the dipole approximation with Monte Carlo simulation for more accurate rendering [19], [20], [21]. These methods share the same spirit by splitting the volume into parts, the inner core and the outer shell. The inner part is rendered using the dipole solution while the outer one is simulated using Monte Carlo method.

In the context of interactive translucent materials rendering, lots of approaches have been proposed to exploit the graphics hardware. Radiosity-based methods achieve high frame rates by pre-computing the diffusion transport between vertices or patches [22], [23], [24]. Other approaches avoid pre-computation by first rendering an irradiance map into textures and later sampling or filtering it [25], [26], [27], [28], [29], [30]. d'Eon *et al.* [31] extended translucent shadow maps [25] and texture-space diffusion for rendering human skin. Munoz *et al.* [32] modeled the subsurface scattering as image convolution and solved it using discrete Fourier transform. These approaches, however, either restrict illumination to simple light sources or are not scalable to complex scenes. Recently, Li *et al.* [33] focused on rendering translucent cutouts and presented TransCut. By implementing the algorithm on the GPU, they achieved interactive frame rates for small scenes.

Other methods render translucent materials under complex illumination in real-time using expensive precomputation [34], [35], [36]. Wang *et al.* [37] proposed a system for real-time editing and relighting of homogeneous translucent materials. The approach proposed by Sheng *et al.* [38] is also based on precomputation but focuses on the inter-reflection between translucent objects.

2.3 Scalable many-light rendering

Recently, many-light rendering has received great acclaim due to its flexibility for handling arbitrary illumination. For high-quality rendering, it is common to use hundreds of thousands of virtual point lights (VPLs). Direct evaluation of contributions from all lights is impractical. Lightcuts [6] and multidimensional lightcuts [7] reduce shading cost by adaptively clustering lights. The clustering configuration is selected based on estimated error bounds. One of the major challenges with these methods is that they assume lights are fully visible when estimating errors. They make the assumption because the bound of visibility is unknown or expensive to obtain. Hasan *et al.* [8] dealt with the many-light problem with matrix sampling and reconstruction. They exploited the low-rank property of the light-transport matrix by first rendering a low-resolution image and

then using the lighting contributions to cluster VPLs. The later LightSlice approach [9] combines the best of lightcuts and matrix sampling by locally refining light clusters for a cluster of shading points. Wang *et al.* [39] presented a scheduling system for rendering out-of-core geometry and lights on GPUs. Unfortunately, none of these methods discussed how to handle translucent materials.

With the same goal as our paper, Arbree *et al.* [5] proposed a scalable single-pass method to address the excessive irradaince computation of the two-pass method [4] based on the lightcuts framework. Surface irradiance is computed adaptively until the contribution is smaller than an estimated error bound. However, as the original lightcuts approach, their method becomes less effective when complex occlusion occurs. Another disadvantage of their single-pass strategy is that it requires form-factor caches to reuse surface irradiance samples. It not only complicates the implementation, but also becomes ineffective in complex scenes because the cache hit rate could be extremely low, as reported in Arbree's thesis (e.g. 1%) [40].

3 Algorithm Overview

This paper extends the matrix sampling approach to scalable translucent material rendering. As stated in Section 1, in translucent material rendering, the contribution of a camera sample due to multiple scattering can be calculated by integrating irradiance values of surface samples with distance-dependent diffusion kernels, while the irradiance of a surface sample is calculated by summing contributions from light samples. Thus, in principle, to use the matrix sampling approach, the light transport in our setting can be modeled by a 3D *Light-Surface-Camera* tensor. However, we found that the tensor-based approach consumes a dramatic amount of memory and could be redundant in our application. Thus, we decompose the light transport into two matrices for *Light-to-Surface* and *Surface-to-Camera* light transport respectively.

To use the matrix representation, we first discretize the lights, the surfaces, and the image into point samples. Light samples are generated as in previous many-light approaches [6], [8]. Surface samples are created by using a ray-tracing-based Poisson-disk sampling approach [41]. Camera samples are 3D intersections of the scene and the eye rays corresponding to sub-pixel samples for anti-aliasing.

3.1 Dual-matrix representation

Our light transport representation consists of *Light-to-Surface* and *Surface-to-Camera* matrices (Fig. 2). The *Light-to-Surface* matrix describes the energy transport from light samples (columns) to surface samples (rows). Matrix entries are lighting contributions. By summing along each row, we obtain the irradiance value of each surface sample. That is, by taking row sums, the matrix collapses into an irradiance vector for surface samples. Similarly, the *Surface-to-Camera* matrix describes the energy transport from surface samples (columns) to camera samples (rows). Matrix entries are diffusion contributions. Note that the *Surface-to-Camera* matrix in Fig. 2 is transposed for better fitting into the figure. The pixel colors of the final image can be obtained by multiplying the surface irradiance vector with the *Surface-to-Camera* matrix.

As demonstrated in Fig. 2, because the characteristics of light transport and diffusion transport differ, these two matrices exhibit distinct structures. The *Light-to-Surface* matrix has both global and local structures and the *Surface-to-Camera* matrix tends to be more local. Our algorithm exploits these structures to cluster the surface samples with little contribution, avoiding wasting computation on unimportant surface samples to the final image.

3.2 Flow of the algorithm

Fig. 3 demonstrates the flow of our algorithm. There are two passes. The first pass generates the surface irradiance vector while the second pass computes the final image. Since not every surface sample makes significant contribution to the final image, we do not need to compute a fully accurate surface irradiance vector. We exploit the matrix structure for locating important surface samples. In the first pass, we first coarsely approximate the *Light-to-Surface* matrix and the *Surface-to-Camera* matrix for estimating the irradiance values and diffusion reflectances of surface samples, respectively. Due to the local structures of both matrices, we can accurately predict how much a surface sample would influence the camera samples by sampling the low-resolution matrices. This allows us to selectively reconstruct the *Light-to-Surface* matrix and obtain the surface irradiance vector with sufficient precision. The second pass renders the final image by adaptively reconstructing the *Surface-to-Camera* matrix. Since the surface irradiance values have been obtained in the first pass, the potential error for adaptive diffusion gathering can be bounded more tightly. With a more accurate bound, surface samples can be clustered more properly for each camera sample. This way, the proposed method spends computation wisely on samples with more contributions, making it more scalable for large scenes with many translucent objects and complex illumination.

4 Dual-matrix Sampling

This section gives details of the proposed method. Section 4.1 describes the first pass by explaining how to obtain a surface irradiance vector, a vector composing of estimated irradiance values of all surface samples, with sufficient precision. Section 4.2 depicts

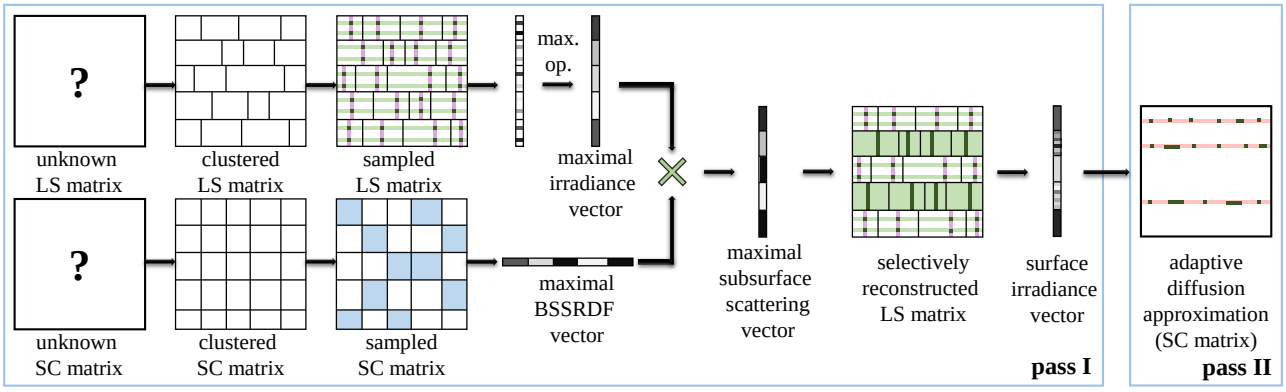


Fig. 3: The flowchart of our algorithm. Our method has two passes. In the first pass, we first estimate the maximum contribution of each surface cluster. This is done efficiently by exploiting the matrix structures in the Light-to-Surface (LS) matrix (for irradiance) and the Surface-to-Camera (SC) matrix. Only dark green entries in the sampled LS matrix are computed. Coupled with the estimated maximum contribution, we then selectively reconstruct the surface irradiance vector as the output of the first pass. In this vector, surface samples with little contributions to the final image are coarsely approximated using only cluster representatives. In the second pass, for rendering the final image, for each camera sample (a row in the SC matrix), we determine the best cluster configuration of surface samples using the adaptive diffusion approximation scheme which is based on a tighter error bound.

the second pass by defining the error bound and explaining how it can be used for adaptive clustering for each camera sample.

4.1 Light-to-Surface matrix reconstruction (the first pass)

The goal of the first pass is to obtain the surface irradiance vector. To avoid overspending computation on unimportant surface samples, we need to identify them without computing them first. As seen in Fig. 2, because a cluster of nearby samples usually have similar behaviors, both transport matrices are low-rank in nature. Thus, their subsampled versions provide very good approximations to the full versions if samples are properly clustered. For clustering surface samples, we first build a bounding volume hierarchy. Clustering is determined by finding a cut in the hierarchy of surface samples. Starting from the root, we repeat replacing the node with the maximum extent with its two children until the desired number of clusters has been reached. For clustering light and camera samples, we use the same method used in LightSlice [9].

The contribution of a surface sample to a camera sample depends both its irradiance value and diffuse BSSRDF value. As stated in the previous paragraph, it would be sufficient to estimate the contribution at the cluster level. As we will introduce in Section 4.2, the maximal contribution of a surface cluster can be estimated by the product of its maximal irradiance and diffuse BSSRDF values. We can estimate the former from the *Light-to-Surface* matrix and the latter from the *Surface-to-Camera* matrix as depicted in the following two paragraphs respectively.

Maximal irradiance vector. The structure of the *Light-to-Surface* matrix is similar to the *Light-to-Camera* matrix in traditional many-light approaches. The only difference is that some surface samples might not contribute much to the final image. Since the matrix has been shown locally low-rank [8], [9], we can obtain an accurate approximation with a small set of representative lights and surface samples. The detailed steps are described as follows: We first randomly select a few surface samples in each surface cluster (light-green rows in the "sampled LS matrix" of Fig. 3). Next, we use LightSlice [9] to select a small set of representative lights for each cluster (purple columns in the "sampled LS matrix" of Fig. 3). The irradiance values of the selected samples are then evaluated using these representative lights. The result is a low-resolution Light-to-Surface matrix with only a small number of surface and light samples calculated (dark-green entries in the "sampled LS matrix" of Fig. 3). By taking row sums, we obtain a reduced surface irradiance vector. Since our goal is to obtain the maximal irradiance for each cluster of surface samples, we take the maximum operator over sampled irradiance values for each cluster to obtain the "maximal irradiance vector" in Fig. 3.

Maximal BSSRDF vector. The structure of the *Surface-to-Camera* matrix is very sparse after properly clustering. This means only a few camera clusters receive contributions from a surface cluster. Since the diffusion kernel depends on the distance between surface and camera samples, camera clusters close to a surface cluster are more likely to receive its contribution. Based on this observation, we estimate the upper bound of diffuse BSSRDF values of a surface cluster

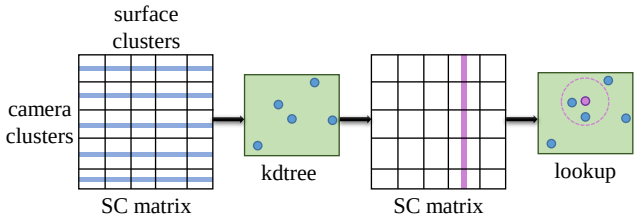


Fig. 4: Algorithm for R_d estimation. We first build a kdtree for the centers of all camera clusters. Then for each surface cluster, we look up its K nearest neighbors in the kdtree. The maximum R_d value of a surface cluster is estimated by taking the maximum R_d values between the surface cluster and the K camera clusters.

from its nearby camera clusters. Fig. 4 illustrates the procedure. We first cluster the camera samples according to their spatial positions. Next, a kd-tree is built for the centers of all camera clusters. For each surface cluster, we find its K nearest camera clusters (K is usually set to 10 in our implementation). For each camera cluster, we estimate the R_d value using the shortest distance between the bounding volumes of the surface and camera clusters. The maximum of these K R_d values is used as the estimation of the maximal R_d value of the surface cluster. The blue blocks in the "sampled SC matrix" of Fig. 3 illustrate the nearest K camera clusters for each surface clusters. By estimating the maximal R_d values for all clusters, we obtain the "maximal BSSRDF vector" in Fig. 3.

Finally, we obtain the estimated maximum contribution of a surface cluster by multiplying its maximum irradiance and R_d values. If the contribution of a surface cluster is too small (smaller than the acceptable error ϵ), we simply average the estimated irradiance values of the selected samples for the surface cluster and use it for all surface samples in the cluster. For other surface clusters, they potentially have significant contributions and need to be evaluated more accurately. Thus, for each surface sample in the cluster, we use LightSlice [9] to compute its irradiance. Through the process, we obtain the surface irradiance vector which has accurate values for important surface samples and rough estimations for unimportant ones. It is worth mentioning that, in our framework, LightSlice can be replaced with any other light importance sampling algorithms [42], [43]. The green blocks in the "selectively reconstructed LS matrix" of Fig. 3 represent surface clusters with significant contributions and the irradiance values of all surface samples within them are estimated using LightSlice. For other clusters, the average irradiance values of selected samples are used as the approximations for all surface samples within them. This way, we obtain the "surface irradiance vector" in Fig. 3 as the output of the first pass.

Please note that our estimation of maximum irradiance is only an approximation and could be smaller than the real maximum since we use sparse samples. However, in practice, we found that it has little influence on the rendered results. It is similar to the argument made by Delves and Mohamed [44], in which they argue that a realistic error estimation is preferred over a pessimistic bound.

4.2 Surface-to-Camera matrix reconstruction (the second pass)

With the surface irradiance vector obtained from pass 1, pass 2 reconstructs the *Surface-to-Camera* matrix and renders the image by adaptively refining a hierarchy of surface samples in a similar way as previous work [4]. We reuse the bounding volume hierarchy of surface samples built in previous pass, where an interior node corresponds to a cluster of surface samples. For a camera sample, to compute its color, we traverse the hierarchy to find the cluster configuration of the surface samples that best approximates the diffusion transport from surface samples to the camera sample based on the following metric for the error bound:

$$\left\| L_C^{true} - L_C^{est} \right\| \leq R_d^{(ub)} \left[\sum_{j \in C} E_j A_j \right], \tag{4}$$

where C is a surface cluster represented by a node in the hierarchy; $R_d^{(ub)}$ is the upper bound of diffusion transport from the camera sample to the surface cluster C determined by computing the shortest distance from the camera sample to the bounding volume of node C; E_j and A_j are respectively the estimated irradiance value and the area of a surface sample j in C.

Equation (4) gives the maximal contribution of the surface cluster C to a camera sample. Since the error of estimating a cluster's contribution with its representative sample cannot exceed its maximal contribution, Equation (4) also gives a bound on the error of adaptive approximation. We repeat substituting the node C with its two children if the maximum error is larger than a predefined acceptable error δ until all clusters are fine enough. The color of the camera sample is then estimated by summing contributions from the resulting configuration of surface clusters.

It is worth noting that Equation (4) is only bounded by the maximum diffuse BSSRDF $R_d^{(ub)}$ because we already know (approximately) the irradiance values of surface samples in C. The dual-matrix representation allows us to obtain a tighter error bound than previous approaches as discussed below.

Comparison to the traditional two-pass approach. The two-pass approach proposed by Jensen and Buhler [4] adopts a heuristic error metric based on the approximated maximum solid angle that a surface cluster C covers toward a camera sample x_o:

$$\Delta\omega = \frac{A_C}{\|x_o - P_C\|}, \tag{5}$$

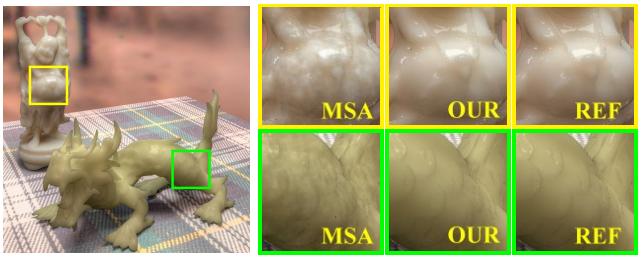


Fig. 5: Equal-time comparison (50 sec.) of the error metrics: maximum solid angle, MSA (Equation (5)), and ours (Equation (4)). Our metric can locate surface samples with larger contributions better, making the adaptive diffusion approximation more efficiently.

where A_C is the total surface area in node C and P_C is the position of the surface representative, computed by averaging the positions of surface samples in the cluster weighted by their irradiance values. For a given camera sample, this error metric spends more effort on refining nearby surface clusters, without considering their irradiance values. When the lighting is not uniform (for example, with strong back lighting), it requires more time to converge.

Fig. 5 shows the images rendered with two metrics, the maximum solid angle approximation (MSA) in Equation (5) and ours. Although the two methods converge to the correct results eventually, under a given time constraint, our method converges faster by locating the surface samples with potentially large errors and refining them first. It has advantages in applications such as lighting previews.

Comparison to the single-pass approach. Arbree *et al.* [5] estimate the error bound of irradiance based on lightcuts [6], [7] and combine it with the bound of diffuse BSSRDF,

$$\left\|L_T^{true} - L_T^{est}\right\| \leq R_d^{(ub)} F^{(ub)} \left[\sum_{i \in C_L} I_i\right] \left[\sum_{j \in C_B} A_j\right], \tag{6}$$

where T is a light-surface-camera triple [5]. Since the irradiance is computed on demand during the adaptive refinement, its value is unknown when estimating the bound and can only be conservatively estimated by the upper bound of the form factor $F^{(ub)}$ multiplied by total intensity from lights in the light cluster C_L, $\sum_{i \in C_L} I_i$, and the total area of the surface cluster C_B, $\sum_{j \in C_B} A_j$. The negligence of visibility is the main weakness of this metric.

5 Results and Discussions

We implemented the proposed algorithm on top of the PBRT2 system [45]. All results were generated on a machine with an Intel Xeon E5-2650 CPU at 2.0 GHz, 128GB of RAM, and using 32 threads.

Scene	# Triangles	# VPLs	# Surface Samples
Sculpture	800,179	18,432	13,026,444
Cathedral	157,740	11,741	6,565,679
Room	503,741	165,912	13,761,152
ChessGame	1,514,008	52,675	11,202,462
Exhibition	7,354,427	186,448	20,512,386
Museum	1,453,519	83,458	52,424,751

TABLE 1: Statistics of the six test scenes. The light samples (VPLs), including both direct and indirect illumination, were obtained by uniform sampling the environment light and tracing photons. The surface samples were uniformly distributed on all translucent surfaces (objects that fully outside the view frustum were culled).

5.1 Tested scenes

For assessing the performance of the proposed method, we tested it on six scenes with different illumination conditions and geometry complexities. The first two scenes, *Sculpture* (Fig. 6) and *Cathedral* (Fig. 7), have relatively simple scene layouts. In the *Sculpture* scene, a marble stone carving is illuminated by a high-frequency environment light and a large area light. The scene demonstrates a case of close-up rendering as only a small portion of the sculpture is within the view frustum. In the *Cathedral* scene, the jade-made lucy model is placed in front of a stained glass window (modeled as an environment map). We used this scene for experimenting with back lighting. The other three scenes are more complex. The *Room* scene (Fig. 8) contains very difficult light paths. The illumination of this scene comes from an environment map and an indoor area light. Lights from the environment map can only come into the room though the door and only their indirect bounces can shade the translucent objects on the desk. The *ChessGame* scene (Fig. 9) contains a large number of translucent objects: a few chess pieces, a vase, a set of teapot and cups, and a candle. Illumination comes from a high-frequency environment light and an area light. The *Exhibition* scene (Fig. 10) contains lots of translucent statues. The scene is illuminated by an environment light and five area lights. Due to the complex layout, only a small portion of translucent surfaces appear in the final image. The last scene, *Museum* (Fig. 11), contains massive translucent surfaces. Many objects in the scene are translucent, and cannot be easily culled out. The scene is illuminated by an environment map and a large area light. It contains difficult light paths because direct lighting coming from the environment map can only contribute through the windows and doors.

All the parameters of translucent materials in these scenes were taken from the measurements reported in previous papers [1], [46]. The translucent materials were rendered with the better dipole model [3]. The first two scenes, *Sculpture* and *Cathedral*, were rendered at a resolution of 500×850 and the other scenes,

Room, *ChessGame*, *Exhibition*, and *Museum*, were rendered at a resolution of 800×600, all with 16 subpixel samples for antialiasing (for efficiency, multiple scattering is computed only once per pixel). Finally, surfaces without subsurface scattering were rendered with the matrix row-column sampling approach [8]. Table 1 summarizes a few statistics of the six tested scenes.

5.2 Comparisons

We conducted both equal-time and equal-quality comparisons on the following approaches:

- Jensen and Buhler [4]: the traditional two-pass algorithm which computes surface irradiance in a brute-force manner and adopts the maximum solid angle approximation (Equation (5)) for adaptive diffusion approximation. We followed the implementation provided by PBRT2 [45]. Matrix sampling approaches [8], [9] were employed for reducing the number of light samples during irradiance computation.
- Arbree et al. [5]: a single-pass approach based on lightcuts. We implemented their method on the top of PBRT2 [45]. The size of the form-factor cache was set to 1,000,000. All other parameters were set according to the authors' suggestion.
- Our method: our dual-matrix sampling method described in section 3 and 4. The number of light samples was set as the same as the traditional two-pass method. For all scenes, we used 4,096 surface clusters and 4,096 camera clusters. In our experiments, we found the number of clusters makes only slight difference to the final results. We fixed the number of sparse surface samples in irradiance estimation to 8, and the number K of nearest neighbors in diffuse BSSRDF estimation to 10.

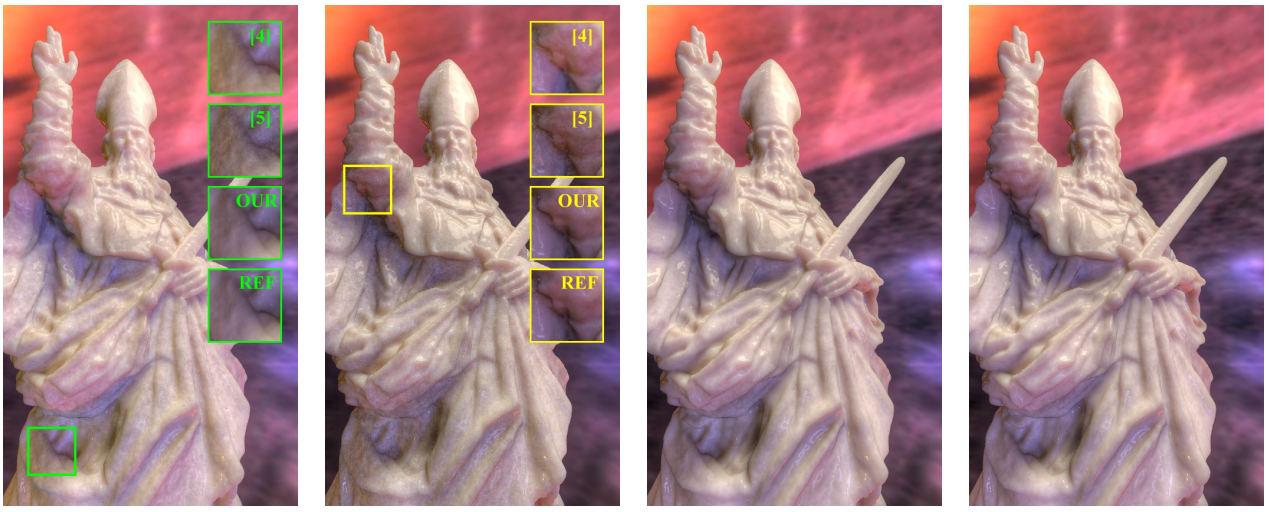
Fig. 6 to 10 show the equal-time comparisons of the six scenes with the three methods in comparisons. In this set of comparisons, we first measured the time spent by Jensen and Buhler's method [4]. The refining threshold in the adaptive diffusion approximation (Equation (5)) was set to 0.05 (the default value used in PBRT2's implementation). The number of light samples in the irradiance computation was set as follows: 50 for *Sculpture* and *Cathedral*, 200 for *ChessGame*, and 400 for *Room*, *Exhibition*, and *Museum*. Then, we carefully adjusted parameters for Arbree *et al.*'s approach [5] and our method to match the same time budgets taken by Jensen and Buhler's method [4]. Table 2 lists the detailed rendering setting and timing of each step of our method.

In the *Sculpture* scene (Fig. 6), because only a small fraction of surface samples (about 10%) locate within the view frustum, the traditional two-pass approach [4] which computes surface irradiance in a brute-force manner becomes very ineffective. The single-pass method based on lightcuts [5] avoids the unnecessary computation by evaluating surface irradiance on demand and only incurring computation for the ones surrounding camera samples. Although their method improves performance in this example, it is not as effective as our method because the surface irradiance cannot be reused directly in their method. Our method possesses both advantages of traditional two-pass method on reusing surface irradiance and the single-pass method on reducing unnecessary irradiance computation, 88% in this example, thus achieving superior performance.

The *Cathedral* (Fig. 7) and the *Room* (Fig. 8) scenes demonstrate two examples in which the lightcut-based method does not work well. In the *Cathedral* scene, the jade lucy is illuminated from the back side and many surface samples at the front side have no contribution. Because the error metric used by subsurface lightcuts [5] does not include a visibility term, it spends a large amount of time on the samples at the front side for their larger diffuse BSSRDF values. A similar situation occurs in the *Room* scene, where the occluded illumination from the environment light makes the refinement of light-surface-camera triples ineffective. The RMSE values listed in the caption show Arbree *et al.*'s approach [5] produces much higher errors than the other two methods. Compared to Jensen and Buhler's method [4], in these two scenes, our method saves 70% and 32% irradiance computation, respectively. The reduced computation allows our method to concentrate on important surface samples by gathering more lights for them. As a result, our method greatly reduces the color shift due to inaccurate surface irradiance values produced by their method. This phenomena is especially obvious in scenes with complex lighting, such as the *Room* (Fig. 8(a)) and *Exhibition* (Fig. 10(a)) scenes. Moreover, our error metric for adaptive diffusion approximation is also more accurate and improves the overall performance.

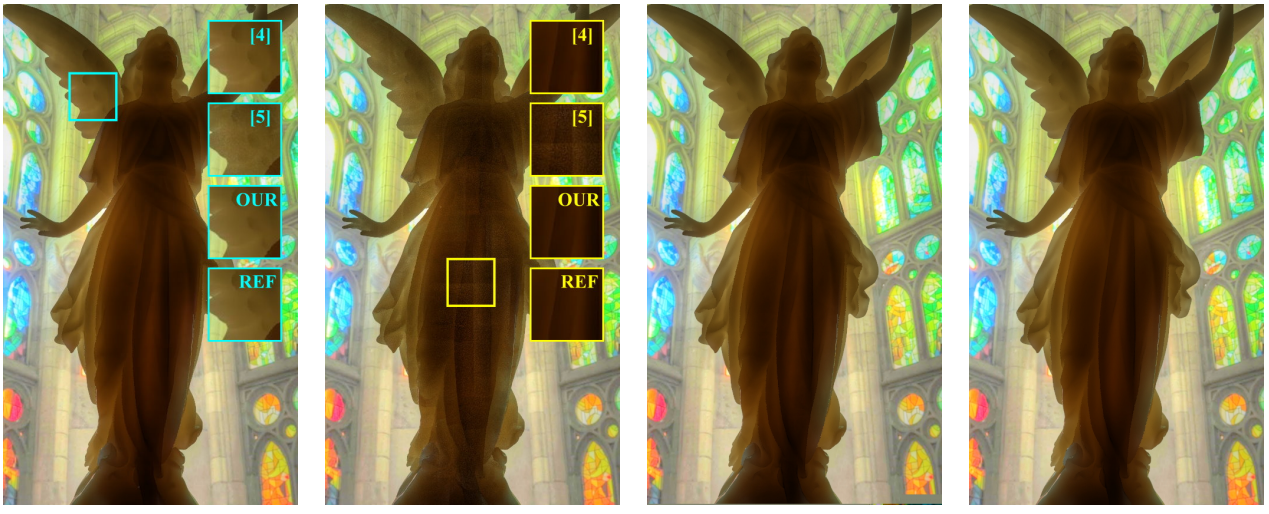
In the *ChessGame* scene (Fig. 9), although the translucent objects form a relatively complex layout, the occlusion is not severe and almost all translucent surfaces locate within the view frustum. In this particular example, subsurface lightcuts [5] and our method can only save about 45% irradiance computation. Although only with the modest saving on irradiance computation, the proposed error metric for adaptive diffusion approximation makes our method more efficient than Jensen and Buhler's method [4] because it can better capture the high-frequency changes in the illumination.

In the *Exhibition* scene (Fig. 10), because a lot of surface samples have very small contributions to the final image due to occlusion, our method saves 70% irradiance computation compared to Jensen and Buhler's method [4]. It is worth noting that although subsurface lightcuts [5] also saves irradiance com-



(a) Jensen and Buhler [4] (b) Arbree et al. [5] (c) Our method (d) Reference

Fig. 6: Equal-time comparison of the *Sculpture* scene (30 sec.). The RMSE values of images (a) to (c) are 0.003270, 0.001661, and 0.000587, respectively.



(a) Jensen and Buhler [4] (b) Arbree et al. [5] (c) Our method (d) Reference

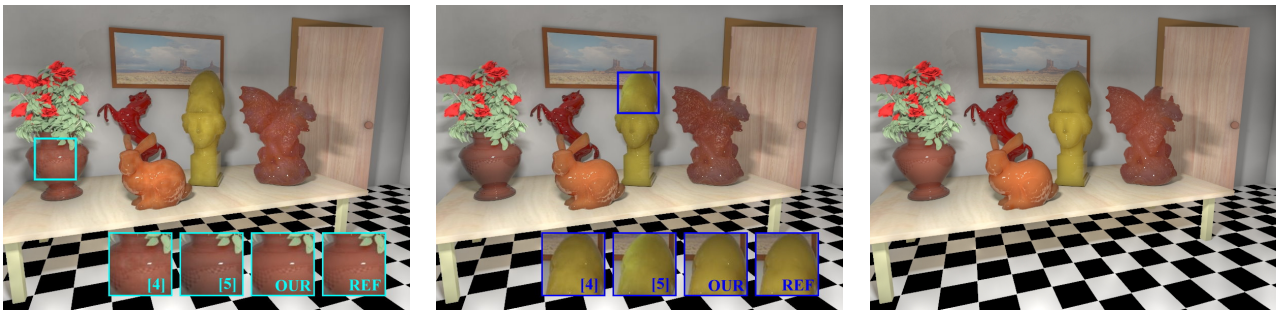
Fig. 7: Equal-time comparison of the *Cathedral* scene (30 sec.). The RMSE values of images (a) to (c) are 0.000107, 0.000240, and 0.000082, respectively.

putation, the benefit is often overshadowed by the overhead of tree refinement and the low hit-rate of the form factor cache.

Finally, the *Museum* scene (Fig. 11) demonstrates a challenging example to all methods. The scene contains a huge number of surface samples distributed over the whole scene and the shading produced by the fine structures of the windows requires lots of light samples for reconstruction. The scene also contains difficult light paths because direct lighting from the environment map can only contribute to the image through the small windows and doors. Compared with Jensen and Buhler's method [4], 68% of the surface samples are saved by our method. By using the saved time for more light samples, our method greatly reduces the artifacts appeared in other methods.

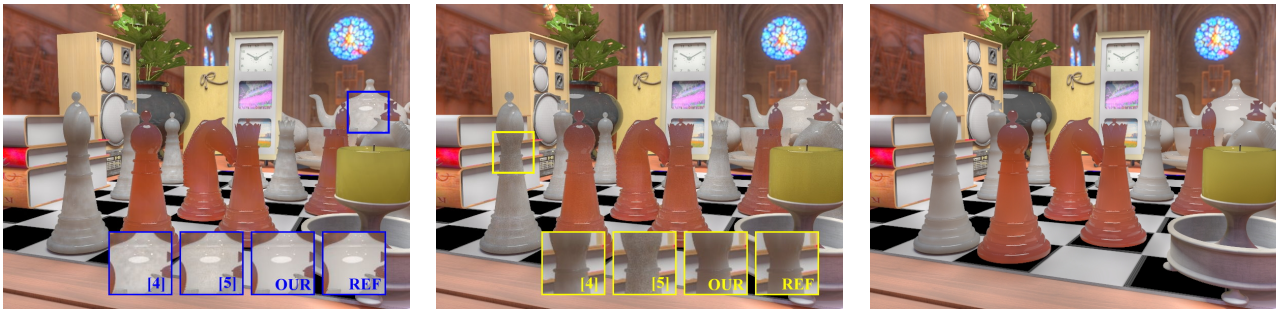
Another interesting experiment is the comparison between the proposed method with and without selective Light-to-Surface matrix reconstruction. We conducted equal-quality comparisons and list the results in Table 2. As expected, the selective Light-to-Surface matrix reconstruction improves the efficiency and the speedups are between 1.2x to 5.7x for these scenes. Nevertheless, because of the more accurate error metric in adaptive diffusion approximation, our method without the selective Light-to-Surface matrix reconstruction is still more efficient than the original two-pass approach (please refer to Fig. 12).

We also conducted equal-quality comparisons for these three methods. Fig. 12 shows the time-



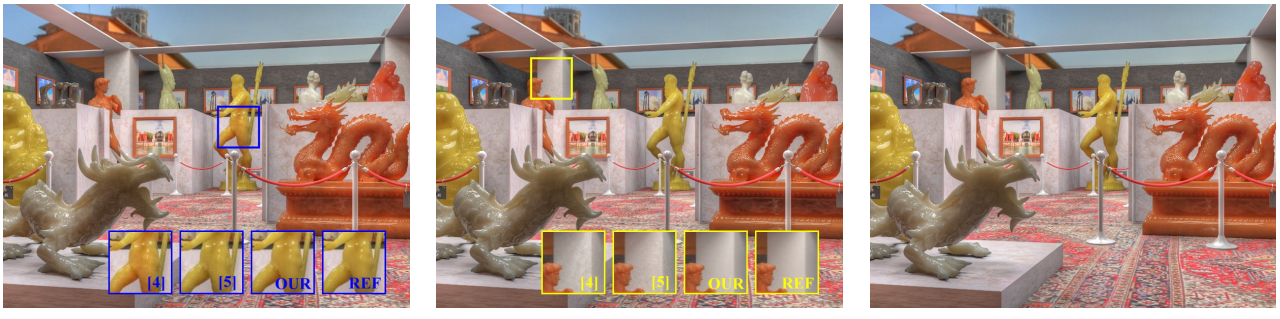
(a) Jensen and Buhler [4] (b) Arbree et al. [5] (c) Our method

Fig. 8: Equal-time comparison of the *Room* scene (300 sec.). The RMSE values of images (a) to (c) are 0.000347, 0.001420, and 0.000114, respectively.



(a) Jensen and Buhler [4] (b) Arbree et al. [5] (c) Our method

Fig. 9: Equal-time comparison of the *ChessGame* scene (110 sec.). The RMSE values of images (a) to (c) are 0.003472, 0.005018, and 0.001196, respectively.



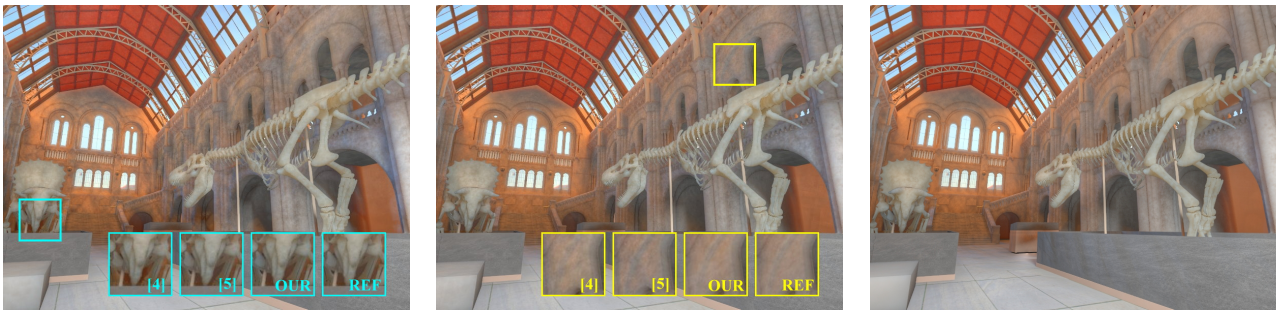
(a) Jensen and Buhler [4] (b) Arbree et al. [5] (c) Our method

Fig. 10: Equal-time comparison of the *Exhibition* scene (300 sec.). The RMSE values of images (a) to (c) are 0.011949, 0.014364, and 0.007185, respectively.

RMSE plots for *Sculpture*, *Cathedral*, *ChessGame*, and *Exhibition*. For the two-pass methods (including Jensen and Buhler's approach [4] and our method), there are two ways to reduce error: increasing light samples or lowering the threshold for the adaptive diffusion approximation. In this experiment, we started with a small number of light samples (50 for *Sculpture* and *Cathedral*, and 100 for *ChessGame*, and *Exhibition*). For a fixed number of light samples, we gradually lowered the threshold of the adaptive diffusion approximation. The error decreased along with the threshold quickly at the beginning but got stuck after some threshold due to inaccuracy of surface irradiance. When the error got stuck, we increased the number of light samples to continue lowering the error. It is why the plots for the two-pass methods look piecewise at times. As the figures reveal, in most cases, the two-pass methods are more efficient because of reusing surface irradiance. Our method outperforms the other approaches consistently in all scenes. It takes much less time to achieve a specific error.

6 Conclusion and Future Work

In this paper we propose a scalable algorithm for rendering translucent materials. Our method is based



(a) Jensen and Buhler [4] (b) Arbree et al. [5] (c) Our method

Fig. 11: Equal-time comparison of the *Museum* scene (2900 sec.). The RMSE values of images (a) to (c) are 0.001267, 0.001880, and 0.000697, respectively.

Scene	Setting			Timing				
	Num. LC	Num. SC	Num. CC	LS smp.	SC smp.	Irr. Computation	Rendering	w/o Selective
Sculpture (30 s.)	200	4096	4096	0.6 s.	0.2 s.	24.3 s.	4.6 s.	170.1 s.
Cathedral (30 s.)	190	4096	4096	0.5 s.	0.2 s.	26.8 s.	1.6 s.	81.3 s.
Room (300 s.)	570	4096	4096	1.2 s.	0.2 s.	286.2 s.	4.8 s.	368.6 s.
ChessGame (110 s.)	360	4096	4096	0.6 s.	0.2 s.	103.2 s.	6.8 s.	180.5 s.
Exhibition (300 s.)	950	4096	4096	1.6 s.	0.2 s.	290.7 s.	6.8 s.	619.0 s.
Museum (2900 s.)	1200	4096	4096	6.1 s.	0.2 s.	2881.2 s.	9.9 s.	8267.3 s.

TABLE 2: The detailed statistics of our method for rendering the images shown in Fig. 6(c), Fig. 7(c), Fig. 8(c), Fig. 9(c), Fig. 10(c), and Fig. 11(c). For the rendering setting, we list the number of light clusters (Num. LC), the number of surface clusters (Num. SC), and the number of camera clusters (Num. CC). We also list the time spent by each step, including LS matrix sampling (LS smp.), SC matrix sampling (SC smp.), irradiance computation (Irr. Computation), and adaptive diffusion approximation (Rendering). The last column lists the time required for the proposed method without selective Light-Surface matrix reconstruction to reach the same quality for comparisons.

on the dual-matrix representation, which represents the light transport due to multiple scattering by two matrices: *Light-to-Surface* and *Surface-to-Camera*. These two matrices have distinct structures because of different characteristics of energy transport for translucent materials. We exploit such structures to avoid computing unnecessary surface irradiance values that will not be used in the adaptive diffusion approximation. Our method is substantially faster because it not only reduces irradiance pre-computation but also provides a more accurate error estimation for the adaptive gathering.

In the future, we would like to extend our method to deal with heterogeneous translucent materials. In this case, the bound of diffuse BSSRDF cannot be easily determined by the shortest distance of a camera sample to a surface cluster. We would like to explore methods for addressing this problem. It is also interesting to combine our method with the image-space caching approaches [17], or the local BSSRDF sampling approaches [16].

Acknowledgments

The authors would like to thank the creators of the models and textures used in this paper: Lucy in Fig. 7, bunny in Fig. 8, David statue, happy buddha, and xyz rgb dragon in Fig. 10, via the Stanford 3D Scanning Repository; the vase in Fig. 8, the candle, clock, table, teapot and cups, books, bag, and speaker in Fig. 9, and the buddha in Fig. 10, downloaded from archive3d.net; the vase in Fig. 9 and the paintings in Fig. 10 from cgrealm.org; SanktKilian in Fig. 6 from forum.jotero.com; the gargoyle in Fig. 8 via the AIM@SHAPE repository; Atenea in Fig. 8 from 3dcadbrowser.com; the horse in Fig. 8 from crazy3dfree.com; the rope barrier in Fig. 10 from sharecg; the chinese dragons in Fig. 10 from newcger.com; the statues in Fig. 10 from 3dmodelfree.com; the Persian carpet in Fig. 10 from cgtextures.com; environment maps from Light Probe Image Gallery and Dan Meyer; the chesses and chessboard in Fig. 9 were modeled by Wojciech Jarosz; the walls, door, and table in Fig. 8 were made by Miika Aittala, Samuli Laine, and Jaakko Lehtinen. the natural history museum scene in Fig. 11 from the lighting challenges on 3dRenderer.com.

References

[1] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan, "A practical model for subsurface light transport," in *Proc. 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH 01)*, 2001, pp. 511–518.

[2] E. D'Eon and G. Irving, "A quantized-diffusion model for rendering translucent materials," *ACM Trans. Graph. (Proc. SIGGRAPH 11)*, vol. 30, no. 4, pp. 56:1–56:14, Jul. 2011.

[3] E. D'Eon, "A better dipole," Tech. Rep., 2012.

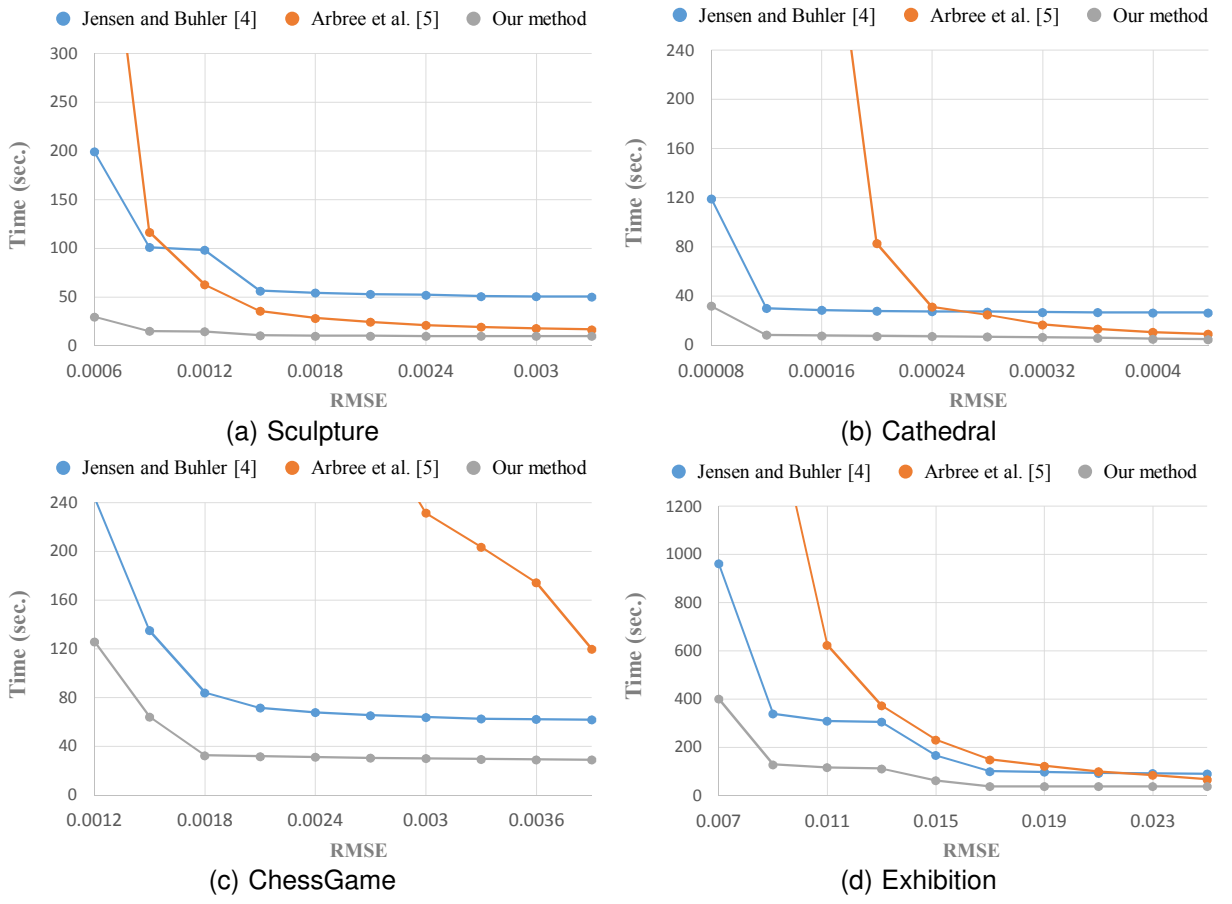


Fig. 12: Equal-quality comparisons of Jensen and Buhler [4], Arbree et al. [5], and our method.

[4] H. W. Jensen and J. Buhler, "A rapid hierarchical rendering technique for translucent materials," *ACM Trans. Graph. (Proc. SIGGRAPH 02)*, vol. 21, no. 3, pp. 576–581, Jul. 2002.

[5] A. Arbree, B. Walter, and K. Bala, "Single-pass scalable subsurface rendering with lightcuts," *Computer Graphics Forum (Proc. Eurographics 08)*, pp. 507–516, 2008.

[6] B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian, and D. P. Greenberg, "Lightcuts: a scalable approach to illumination," *ACM Trans. Graph. (Proc. SIGGRAPH 05)*, vol. 24, no. 3, pp. 1098–1107, Jul. 2005.

[7] B. Walter, A. Arbree, K. Bala, and D. P. Greenberg, "Multidimensional lightcuts," *ACM Trans. Graph. (Proc. SIGGRAPH 06)*, vol. 25, no. 3, pp. 1081–1088, Jul. 2006.

[8] M. Hašan, F. Pellacini, and K. Bala, "Matrix row-column sampling for the many-light problem," *ACM Trans. Graph. (Proc. SIGGRAPH 07)*, vol. 26, no. 3, Jul. 2007.

[9] J. Ou and F. Pellacini, "LightSlice: matrix slice sampling for the many-lights problem," *ACM Trans. Graph. (Proc. SIGGRAPH Asia 11)*, vol. 30, no. 6, pp. 179:1–179:8, Dec. 2011.

[10] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis, "Radiometry," 1992, ch. Geometrical considerations and nomenclature for reflectance, pp. 94–145.

[11] C. Donner and H. W. Jensen, "Light diffusion in multi-layered translucent materials," *ACM Trans. Graph. (Proc. SIGGRAPH 05)*, vol. 24, no. 3, pp. 1032–1039, Jul. 2005.

[12] C. Donner and H. W. Jensen, "Rendering translucent materials using photon diffusion," in *Proc. 18th Eurographics conference on Rendering Techniques (EGSR 07)*, 2007, pp. 243–251.

[13] L.-Q. Yan, Y. Zhou, K. Xu, and R. Wang, "Accurate translucent material rendering under spherical gaussian lights," *Comp. Graph. Forum*, vol. 31, no. 7pt2, pp. 2267–2276, Sep. 2012.

[14] R. Habel, P. H. Christensen, and W. Jarosz, "Photon beam diffusion: A hybrid monte carlo method for subsurface scattering," in *Proc. 24th Eurographics conference on Rendering Techniques (EGSR 13)*, 2013.

[15] B. Walter, P. Khungurn, and K. Bala, "Bidirectional lightcuts," *ACM Trans. Graph. (Proc. SIGGRAPH 12)*, vol. 31, no. 4, pp. 59:1–59:11, Jul. 2012.

[16] A. King, C. Kulla, A. Conty, and M. Fajardo, "BSSRDF importance sampling," in *ACM SIGGRAPH 2013 Talks*, 2013, pp. 48:1–48:1.

[17] S.-L. Keng, W.-Y. Lee, and J.-H. Chuang, "An efficient caching-based rendering of translucent materials," *Vis. Comput.*, vol. 23, no. 1, pp. 59–69, Dec. 2006.

[18] P. H. Christensen, G. Harker, J. Shade, B. Schubert, and D. Batali, "Multiresolution radiosity caching for global illumination in movies," in *ACM SIGGRAPH 2012 Talks*, 2012, pp. 47:1–47:1.

[19] Y. Chen, X. Tong, J. Wang, S. Lin, B. Guo, and H.-Y. Shum, "Shell texture functions," *ACM Trans. Graph. (Proc. SIGGRAPH 04)*, vol. 23, no. 3, pp. 343–353, Aug. 2004.

[20] H. Li, F. Pellacini, and K. E. Torrance, "A hybrid monte carlo method for accurate and efficient subsurface scattering," in *Proc. Sixteenth Eurographics conference on Rendering Techniques (EGSR 05)*, 2005, pp. 283–290.

[21] X. Tong, J. Wang, S. Lin, B. Guo, and H.-Y. Shum, "Modeling and rendering of quasi-homogeneous materials," *ACM Trans. Graph. (Proc. SIGGRAPH 05)*, vol. 24, no. 3, pp. 1054–1061, Jul. 2005.

[22] H. P. A. Lensch, M. Goesele, P. Bekaert, J. Kautz, M. A. Magnor, J. Lang, and H.-P. Seidel, "Interactive rendering of translucent objects," in *Proc. 10th Pacific Conference on Computer Graphics and Applications (PG 02)*, 2002.

[23] N. A. Carr, J. D. Hall, and J. C. Hart, "GPU algorithms for radiosity and subsurface scattering," in *Proc. of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, 2003, pp. 51–59.

[24] X. Hao and A. Varshney, "Real-time rendering of translucent

meshes," *ACM Trans. Graph.*, vol. 23, no. 2, pp. 120–142, Apr. 2004.
[25] C. Dachsbacher and M. Stamminger, "Translucent shadow maps," in *Proc. 14th Eurographics workshop on Rendering (EGRW 03)*, 2003, pp. 197–201.
[26] F. Banterle and A. Chalmers, "A fast translucency appearance model for real-time applications," in *Proc. Spring Conference on Computer Graphics (SCCG 2006)*, April 2006.
[27] Y. Gong, W. Chen, L. Zhang, Y. Zeng, and Q. Peng, "GPU-based rendering for deformable translucent objects," *Vis. Comput.*, vol. 24, no. 2, pp. 95–103, Jan. 2008.
[28] T. Mertens, J. Kautz, P. Bekaert, F. Van Reeth, and H.-P. Seidel, "Efficient rendering of local subsurface scattering," in *Proc. 11th Pacific Conference on Computer Graphics and Applications (PG 03)*, 2003.
[29] C.-W. Chang, W.-C. Lin, T.-C. Ho, T.-S. Huang, and J.-H. Chuang, "Real-time translucent rendering using gpu-based texture space importance sampling," *Computer Graphics Forum (Proc. Eurographics 08)*, pp. 517–526, 2008.
[30] M. A. Shah, J. Konttinen, and S. Pattanaik, "Image-space subsurface scattering for interactive rendering of deformable translucent objects," *IEEE Comput. Graph. Appl.*, vol. 29, no. 1, pp. 66–78, Jan. 2009.
[31] E. d'Eon, D. Luebke, and E. Enderton, "Efficient rendering of human skin," in *Proc. 18th Eurographics Conference on Rendering Techniques (EGSR 07)*, 2007, pp. 147–157.
[32] A. Muñoz, J. I. Echevarria, F. J. Serón, and D. Gutierrez, "Convolution-based simulation of homogeneous subsurface scattering," *Comput. Graph. Forum*, vol. 30, no. 8, pp. 2279–2287, 2011.
[33] D. Li, X. Sun, Z. Ren, S. Lin, Y. Tong, B. Guo, and K. Zhou, "Transcut: Interactive rendering of translucent cutouts," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 3, pp. 484–494, 2013.
[34] P.-P. Sloan, J. Hall, J. Hart, and J. Snyder, "Clustered principal components for precomputed radiance transfer," *ACM Trans. Graph. (Proc. SIGGRAPH 03)*, vol. 22, no. 3, pp. 382–391, Jul. 2003.
[35] R. Wang, J. Tran, and D. Luebke, "All-frequency interactive relighting of translucent objects with single and multiple scattering," *ACM Trans. Graph. (Proc. SIGGRAPH 05)*, vol. 24, no. 3, pp. 1202–1207, Jul. 2005.
[36] P.-P. Sloan, B. Luna, and J. Snyder, "Local, deformable precomputed radiance transfer," *ACM Trans. Graph. (Proc. SIGGRAPH 05)*, vol. 24, no. 3, pp. 1216–1224, Jul. 2005.
[37] R. Wang-erb, E. Cheslack-Postava, R. Wang, D. Luebke, Q. Chen, W. Hua, Q. Peng, and H. Bao, "Real-time editing and relighting of homogeneous translucent materials," *Vis. Comput.*, vol. 24, no. 7, pp. 565–575, Jul. 2008.
[38] Y. Sheng, Y. Shi, L. Wang, and S. G. Narasimhan, "A practical analytic model for the radiosity of translucent scenes," in *Proc. symposium on Interactive 3D graphics*, 2013.
[39] R. Wang, Y. Huo, Y. Yuan, K. Zhou, W. Hua, and H. Bao, "GPU-based out-of-core many-lights rendering," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 210:1–210:10, Nov. 2013.
[40] A. Arbree, "Scalable and heterogeneous rendering of subsurface scattering materials," Ph.D. dissertation, Cornell University, Oct. 2009.
[41] J. Lehtinen, M. Zwicker, E. Turquin, J. Kontkanen, F. Durand, F. X. Sillion, and T. Aila, "A meshless hierarchical representation for light transport," *ACM Trans. Graph. (Proc. SIGGRAPH 08)*, vol. 27, no. 3, pp. 37:1–37:9, Aug. 2008.
[42] I. Georgiev, J. Křivánek, S. Popov, and P. Slusallek, "Importance caching for complex illumination," *Computer Graphics Forum (Proc. Eurographics 12)*, vol. 31, no. 2, 2012.
[43] Y.-T. Wu and 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, Sep. 2013.
[44] L. M. Delves and J. L. Mohamed, Eds., *Computational methods for integral equations*. New York, NY, USA: Cambridge University Press, 1986.
[45] M. Pharr and G. Humphreys, *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2010.
[46] S. G. Narasimhan, M. Gupta, C. Donner, R. Ramamoorthi, S. K. Nayar, and H. W. Jensen, "Acquiring scattering properties of participating media by dilution," *ACM Trans. Graph. (Proc. SIGGRAPH 06)*, vol. 25, no. 3, pp. 1003–1012, Jul. 2006.

Yu-Ting Wu received his B.S. and M.S. from National Chiao Tung University, Hsinchu, Taiwan, in 2007 and 2009 respectively, Ph.D. degree from National Taiwan University, Taipei, Taiwan, in 2014, all in Computer Science. His research interests include computer graphics and visual effects.

Tzu-Mao Li received his B.S. and M.S. degree from National Taiwan University in 2011 and 2013, respectively. He is currently a Ph.D. student at Massachusetts Institute of Technology. His research interests mainly lie in physically-based rendering, in particular light transport simulation and sampling and reconstruction.

Yu-Hsun Lin received his Ph.D. degree from the Graduate Institute of Networking and Multimedia (GINM) in National Taiwan University (NTU), 2014.

Yung-Yu Chuang received his B.S. and M.S. from National Taiwan University in 1993 and 1995 respectively, and the Ph.D. from 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.