Translucent Radiosity: Efficiently Combining Diffuse Inter-Reflection and Subsurface Scattering

Yu Sheng, *Member, IEEE* , Yulong Shi, Lili Wang, and Srinivasa G. Narasimhan, *Member, IEEE*

Abstract—It is hard to efficiently model the light transport in scenes with translucent objects for interactive applications. The inter-reflection between objects and their environments and the subsurface scattering through the materials intertwine to produce visual effects like color bleeding, light glows, and soft shading. Monte-Carlo based approaches have demonstrated impressive results but are computationally expensive, and faster approaches model either only inter-reflection or only subsurface scattering. In this paper, we present a simple analytic model that combines diffuse inter-reflection and isotropic subsurface scattering. Our approach extends the classical work in radiosity by including a subsurface scattering matrix that operates in conjunction with the traditional form factor matrix. This subsurface scattering matrix can be constructed using analytic, measurement-based or simulation-based models and can capture both homogeneous and heterogeneous translucencies. Using a fast iterative solution to radiosity, we demonstrate scene relighting and dynamically varying object translucencies at near interactive rates.

Index Terms—Global illumination, radiosity, subsurface scattering, inter-reflection

1 Introduction

Accurate rendering of translucent materials such as leaves, flowers, marble, wax, and skin can greatly enhance the realism of synthetic images. The interactions of light within translucent objects and between the objects and their environments produce pleasing visual effects like color bleeding (Fig. 1), light glows and soft shading. The two main mechanisms of light transport—(a) scattering beneath the surface of the materials and (b) inter-reflection between surface locations in free space—combine in complex, non-linear ways (see Fig. 2), making it challenging to render such scenes quickly and realistically. Just modeling one of them or a simple addition of them cannot capture the complete light transport.

Monte-Carlo ray tracing [1], [2] can be used to faithfully render both inter-reflection and subsurface scattering but is too slow to be practical. Jensen and Buhler [3] model the inter-reflection between the diffuse surface and a translucent object by substituting translucency inter-reflection with diffuse inter-reflection. Computational speedups have been achieved by combining Photon Mapping with analytical models for subsurface scattering to capture interesting global illumination effects, such as inter-scattering and caustics, for translucent materials [4]. However, this method does not support multi-layered objects and heterogeneous materials. Furthermore, none of these speedups are sufficient for interactive applications. Previous precomputed radiance transfer (PRT) methods have achieved interactive rendering and relighting of translucent objects by intensive precomputation, usually taking hours [5], [6], [7], [8]. Unfortunately, all these approaches only model light scattered beneath the object surface, without considering surface to surface light interactions from other parts of the object or the environment. Theoretically, volumetric path tracer can be incorporated with PRT to make it possible to model both light paths. But in practice, such a method not only requires long precomputation time, but also loses the flexibility of changing object materials at runtime.

In this paper, we present a simple analytic model of the light transport between translucent objects and diffuse environment. We assume that the translucent objects are highly scattering and hence the reflected and transmitted light is diffuse. This is valid for a wide range of materials and is commonly used in rendering and acquisition systems [9], [10], [11]. We also assume that any other objects in the scene are diffuse. In this setting, the appearance of the scene can be computed by the radiosity at every surface location. Our model extends the classical work in radiosity [12] by including a subsurface scattering matrix that operates in conjunction with the traditional form factor matrix. The subsurface scattering matrix can be constructed using any analytical model (dipole [9] and multi-pole [13]), simulations [11], or measurements [10]. Hence, our method can render both homogeneous and heterogeneous materials.

Being a radiosity-like approach, the method requires a one-time precomputation (10 min) and storage (1.5 GB for a 40k polygon mesh) of the form factor and subsurface scattering matrices. But once this is done, rendering

- *Y. Sheng and Y. Shi are with Bosch Research and Technology Center North America, 4005 Miranda Ave, Ste 200, Palo Alto, CA 94304. E-mail: shengyu@cs.cmu.edu, shiyl@vrlab.buaa.edu.cn.*
- *L. Wang is with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, No. 37 Xueyuan Rd, Haidian District, PO Box 6863, Beijing 100191, China. E-mail: lily_w@vrlab.buaa.edu.cn.*
- *S.G. Narasimhan is with the Robotics Institute, Carnegie Mellon University, 5000 Forbe Ave, Pittsburgh, PA 15213. E-mail: srinivas@cs.cmu.edu.*

Manuscript received 14 Apr. 2013; revised 23 Sept. 2013; accepted 29 Oct. 2013. Date of publication 7 Nov. 2013; date of current version 29 May 2014. Recommended for acceptance by S.-E. Yoon and G. Meenakshisundaram. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TVCG.2013.256

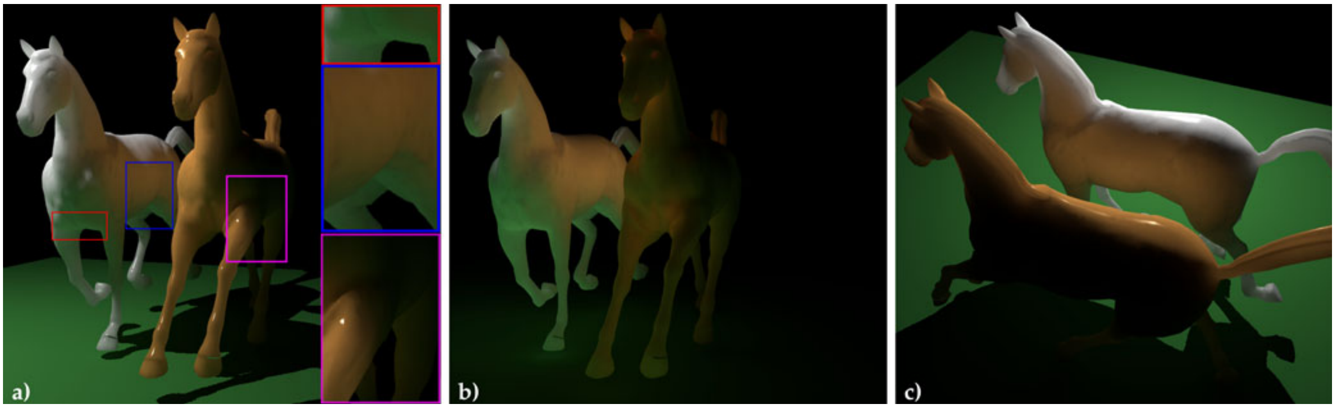

Fig. 1. Inter-reflection and subsurface scattering are closely intertwined for scenes with translucent objects. The main contribution of this work is an analytic model of combining diffuse inter-reflection and subsurface scattering (see Fig. 2). One bounce of Fresnel reflection is added in a separate pass. (a) Two translucent horses (63k polygons) illuminated by a point light source. The three zoomed-in regions show that our method can capture both global illumination effects. (b) The missing light transport component if only subsurface scattering is simulated. (c) The same mesh rendered with a different lighting and viewing position. Our model supports interactive rendering of moving camera, scene relighting, and changing translucencies.

is achieved at interactive rates using our GPU implementation of a standard iterative solver. This allows us to perform fast relighting (5-10 fps) of scenes under different types of light sources (spot-light and environment maps) with arbitrary radiance distributions. Further, our parametrization of the subsurface scattering can be exploited to vary the translucency of objects at near interactive rates (1-4 fps). If we wish to perform only relighting, we can precompute the inverse. Each relighting computation thus requires only one matrix-vector product computation and we achieve real time frame rates. We demonstrate results on objects with complex shapes, including multiple-layered objects like a flower (Fig. 10), and showcase effects like color bleeding, glows, and object-environment interactions (Fig. 1).

2 Background and Related Work

To render a scene with translucent objects accurately, it is important to capture both inter-reflection and subsurface scattering. However, most of the previous studies only focus on one of them and therefore cannot model the complete light transport in scenes such as Fig. 1.

2.1 Radiosity

A classical technique to compute the global illumination for diffuse environments is radiosity [12]. The entire scene is discretized into surface patches, which reflect incoming light evenly to every possible outgoing direction in the surrounding hemisphere. The radiosity on each patch is usually assumed to be constant. An important part of the formulation is the form factor $\mathbf{F}_{ij}$, which represents the fraction of energy leaving one surface patch and directly arriving at another. The radiosity $\mathbf{B}_i$ of each patch can be computed by

$$\mathbf{B}_i = \mathbf{L}_i + \rho_i \sum_{j=1}^{n} \mathbf{B}_j \mathbf{F}_{ij} \tag{1}$$

where, $\mathbf{L}_i$ and ρ_i are the emittance and reflectance of patch i. Written in matrix form, we get

$$\mathbf{B} = \sum_{n=0}^{\infty} (\mathbf{PF})^n \mathbf{L} = (\mathbf{I} - \mathbf{PF})^{-1} \mathbf{L} \tag{2}$$

where, $\mathbf{I}$ is an $n \times n$ identity matrix and $\mathbf{P}$ is the $n \times n$ diagonal albedo matrix. The form factor matrix needs to be precomputed and can be expensive for large scenes. Different strategies have been proposed to accelerate the original algorithm, such as progressive radiosity [14] and hierarchical radiosity [15]. Radiosity performs well for materials with a matte appearance. Specular and glossy materials require ray tracing based approaches to create realistic renderings [16], [17]. Radiosity has also been extended to compute the light transport in the context of participating media by introducing the volume-volume and volume-surface form factors [18]. However, light transport due to inter-reflection and volumetric scattering is tangled together in the formulation. Therefore, with this method, once the properties of participating media are changed, all the form factors need to be recomputed. In contrast, our method models these two light transport separately and supports translucency changes at near-interactive speed. Rushmeier and Torrance [19] have also extended radiosity for translucent surfaces, but only limited to the transmission from the opposite side of the same patch.

2.2 Subsurface Scattering

Subsurface scattering inside translucent materials can be described by an 8D Bi-directional surface scattering distribution function (BSSRDF [20]):

$$s(x_i, \vec{\omega}_i; x_o, \vec{\omega}_o) = \frac{dL_o(x_o, \vec{\omega}_o)}{d\Phi_i(x_i, \vec{\omega}_i)}. \tag{3}$$

Here, $L_o(x_o, \vec{\omega}_o)$ is the outgoing radiance at a surface point x_o in direction $\vec{\omega}_o$, and $\Phi_i(x_i, \vec{\omega}_i)$ is the incoming radiance at a surface point x_i in direction $\vec{\omega}_i$. A brute-force evaluation of this high-dimensional function using Monte-Carlo simulations [2] is very expensive and impractical to use in most applications. Dipole diffusion

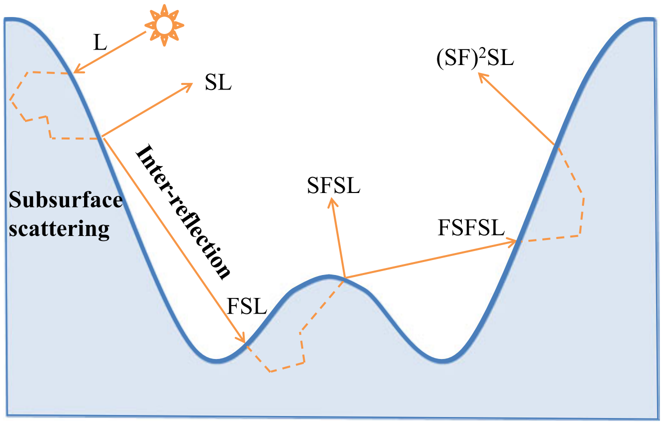

Fig. 2. Light transport in the presence of a non-convex translucent object. Light incident at a surface patch is scattered beneath the surface of the object and exits and re-enters the object at different locations. The propagation of light in free space is termed as inter-reflection and within the object as subsurface scattering. The radiosity of the object is the result of a series of such light interactions and is captured by our model.

model [9] for thick, highly scattering materials is easy to implement, much faster than Monte-Carlo simulations and has been widely used in computer graphics. Later works extended this diffusion model to support multilayer translucent materials [13], heterogeneous translucent materials [11], [21], [22], and anisotropic diffusion [23]. To accelerate the rendering speed of translucent objects, hierarchical methods [3], [24] have been proposed to support large scenes with complex illumination.

For highly scattering materials, the angular dependency of incoming and outgoing directions $\vec{\omega}_i$ and $\vec{\omega}_o$ can be removed [9]. Equation (3) then simplifies to a 4D function that only depends on the incoming and exiting locations. With this assumption, previous work [25], [26] proposed to render translucent objects using radiosity to achieve real-time frame rates. The scene is discretized into N patches. A throughput factor between each pair of patch i and j is used to denote the diffuse subsurface scattering between them. The radiosity of each patch can be computed as the classical radiosity algorithm. Hierarchical extensions of the two methods managed to interactively relight, edit materials, and change geometry for moderately complex translucent objects [27]. However, all these approaches focus only on light scattering beneath the object's surface (dotted lines in Fig. 2) and assume that light does not re-enter the object once it exits the object. Our work builds on these approaches and models all the light interactions shown at Fig. 2.

2.3 Precomputed Radiance Transfer

Realtime global illumination algorithms need to tackle the challenges from the complexity of geometry, lighting, material, and light transport [5]. Instead of addressing all these challenges during the runtime, PRT separates out the lighting and encapsulate all the other factors in a transport operator $\mathbf{T}$ for all the vertices in the scene. For diffuse scenes, the global illumination solution $\mathbf{B}(p)$ at a surface point p can be computed by a vector-vector product with the lighting vector $\mathbf{L}$, at runtime:

$$\mathbf{B}(p) = \mathbf{T}(p) \cdot \mathbf{L}, \tag{4}$$

and thus achieve real time frame rates. In practice, $\mathbf{L}$ and $\mathbf{B}$ are represented with spherical harmonics [5] or wavelet basis [28]. Projecting $\mathbf{L}$ to the basis functions $\mathbf{Y}_j$ ($j = 1, \ldots, N$, N is the number of basis) leads to

$$\mathbf{L}(\omega_i) = \sum_{j=1}^{n} \alpha_j \mathbf{Y}_j(\omega_i) \tag{5}$$

Because light transport is linear under any basis, if the transport operator $\mathbf{T}_j(x)$ for each basis function $\mathbf{Y}_j$ is precomputed, the global illumination results can be computed by a linear combination of $\mathbf{T}_j$ as

$$\mathbf{B}(p) = \sum_{j} \alpha_j \mathbf{T}_j(p) \tag{6}$$

For scenes with glossy objects, the transfer function cannot be simply represented by a vector, but requires a transfer matrix to capture the angular dependency.

This technique has been extended to render translucent objects with both single and multiple scattering components at interactive rates [7] or even real-time speed with only multiple scattering [8]. However, to accurately capture the light propagation between translucent objects and their environment, PRT-based methods require precomputation of global illumination under many (9-25) lighting bases [5], each of which can take several hours if using a Monte-Carlo variant. Furthermore, most PRT methods focus on distant lighting for efficient compression of incident light. Including near-field lighting dramatically increases the precomputation time as even more basis functions are needed. In contrast, our method does not precompute the entire light transfer matrix, but the form factor and subsurface scattering matrices, which takes only a few minutes (versus hours/days for PRT) and works for both distant and near lighting.

3 Analytically Combining Inter-Reflection and Subsurface Scattering

Fig. 2 shows the order of light transport through a non-convex and translucent object. The propagation of light in free space is termed inter-reflection (solid lines) and the propagation within the object is termed subsurface scattering (dotted lines). We assume that light transport alternates only between inter-reflection and subsurface scattering. We are interested in modeling the potentially infinite series of such light interactions (bounces) to compute the final radiosity of the object. Similar to the concept in radiosity, we call the light that reaches the camera with $N-1$ times of inter-reflection the *Nth bounce* of global illumination. Our analytic model can be derived by summing up all the bounces.

3.1 First Bounce

The scene is discretized into n surface patches. The portion of light that is scattered from patch i to patch j is denoted by $\mathbf{S}_{ji}$. If $\mathbf{E}_i$ is the irradiance of patch i, then the light received by patch j is defined as the *local scattering* or as the first bounce:

$$\mathbf{S}_{ji} \cdot \mathbf{E}_i \, . \tag{7}$$

$\mathbf{S}_{ji}$ is called the *scattering factor*, which is the relative amount of the light transported from patch i to j due to subsurface scattering. Note that this term is also called throughput factor [25], [26]. If the areas of the patch i and j are A_i and A_j, $\mathbf{S}_{ji}$ can be written in a manner similar to the form factor as

$$\mathbf{S}_{ji} = \frac{1}{A_i}\int_{A_i}\int_{A_j} s(i,j)dA_j dA_i \tag{8}$$

where, $s(i,j)$ is the simplified BSSRDF function (Equation (3)) that only depends on the incident and outgoing points.

3.2 Second and N- th Bounce

For highly scattering translucent materials, subsurface scattering is dominated by isotropic multiple scattering [9]. With this assumption, the light that exits surface patch j due to the first bounce is equal at any outgoing direction, written as

$$\sum_i \mathbf{S}_{ji}\mathbf{E}_i. \tag{9}$$

The first bounce of light is then transported to other surface patches in the scene through free space. Suppose that $\mathbf{F}_{kj}$ is the form factor from patch j to k, the light received by patch k from patch i is

$$\mathbf{F}_{kj}\sum_i(\mathbf{S}_{ji}\mathbf{E}_i). \tag{10}$$

We obtain the total light received by patch k by adding contributions from all patches j as

$$\sum_j \mathbf{F}_{kj}\sum_i(\mathbf{S}_{ji}\mathbf{E}_i). \tag{11}$$

In matrix form, we thus have

$$\mathbf{FSE} \tag{12}$$

where, $\mathbf{F}$ is the form factor matrix and $\mathbf{S}$ is the subsurface scattering matrix. Both $\mathbf{F}$ and $\mathbf{S}$ are $n \times n$ matrices. Similar to the analysis in Equation (7), the *local scattering*of this bounce is $\mathbf{SFSE}$. This term is the second bounce of light. In general, the Nth bounce of light can be written as

$$(\mathbf{SF})^{N-1}\mathbf{SE}. \tag{13}$$

3.3 Analytic Model

Then, the *entire light transport or radiosity* of the scene is computed as the sum of all bounces of light:

$$\mathbf{B} = \sum_{N=1}^{\infty}(\mathbf{SF})^{N-1}\mathbf{SE} = (\mathbf{I} - \mathbf{SF})^{-1}\mathbf{SE} \tag{14}$$

where, $\mathbf{I}$ is the $n \times n$ identity matrix. The *first bounce* of this equation is the scene appearance if the light transport only contains subsurface scattering. Apparently, if only subsurface scattering is simulated, all the rest of the bounces will be lost. Note that this equation is similar to the classical radiosity equation (Equation (2)) but incorporates the additional subsurface scattering matrix in a simple and elegant manner. In particular, if the entire scene is opaque, $\mathbf{S} = \mathbf{P}$, $\mathbf{PE} = \mathbf{L}$, we obtain Equation (2). Since light propagation in free-space (vacuum) is fundamentally different from propagation through materials, the matrices $\mathbf{F}$ and $\mathbf{S}$ have different characteristics. Treating them as separate entities of light transport allows us to control the translucency of the objects, for instance.

As in classical radiosity, the above equation can be efficiently solved by iterative methods, such as Jacobi or Gauss-Seidel. Since the norm of the form factor matrix $\|\mathbf{F}\| < 1$, the radiosity equation is proven to converge if not all the surfaces are perfect reflectors (with albedo equal to 1) [29]. According to the physical meaning of $\mathbf{S}_{ji}$, the sum of all the percentages of light that scatters from patch i to others should be less than 1. That is, for any i, $\sum_j \mathbf{S}_{ji} < 1$, then

$$\|\mathbf{S}\|_1 = \max_{1\leq i\leq n}\sum_j \mathbf{S}_{ji} < 1. \tag{15}$$

Therefore, the spectral radius ρ of matrix $\mathbf{S}$ satisfies $\rho(\mathbf{S}) \leq \|\mathbf{S}\|_1 < 1$. Since we also have $\rho(\mathbf{F}) \leq \|\mathbf{F}\| < 1$, then $\rho(\mathbf{SF}) \leq \rho(\mathbf{S})\rho(\mathbf{F}) < 1$. Therefore, Equation (14) always converges.

4 Computing Subsurface Scattering Matrix

The model in Equation (14) provides a framework for rendering translucent objects and their interactions with diffuse environments. The subsurface scattering matrix $\mathbf{S}$ can be computed using either analytical models, numerical simulations or measurements. We describe this process below.

Let us start with the numerical evaluation of Equation (8) that computes the light throughput between two surface patches. In practice, since radiance decays (roughly) exponentially through materials, the number of samples of $s(i,j)$ used in the integral can be reduced as the distances between patches increase. Further, depending on the desired quality of rendering, a threshold can be set for the distance beyond which subsurface scattering can be safely ignored.

For *homogeneous materials*, the sampling kernel $s(i,j)$ can be approximated by a 1D function $s(r)$ that only depends on the distance r between the sampling points on each surface. In this case, our model can take either analytical formulations such as dipole diffusion [9], or 1D scattering profiles obtained by measurement. A common approach to acquire $s(r)$ is to illuminate a planar slab of the sample material and measure the intensity distribution on both sides of the slab to obtain the reflection and transmission profiles.

However, when applying the dipole, multi-pole models or observed profiles to objects with arbitrary shapes, it is unclear how the reflectance and transmittance components change due to the local curvature and global shape of the object. Premoze et al. [30] proposed using the most probable paths to compute multiple scattering in participating media. We follow an approach that is similar in spirit and approximate the profile as a 1D function of the shortest optical path through the translucent objects between the light incident and exiting points, as shown in Fig. 3a. Note that this is not the only path for light propagation but only a parametrization. For a convex object,

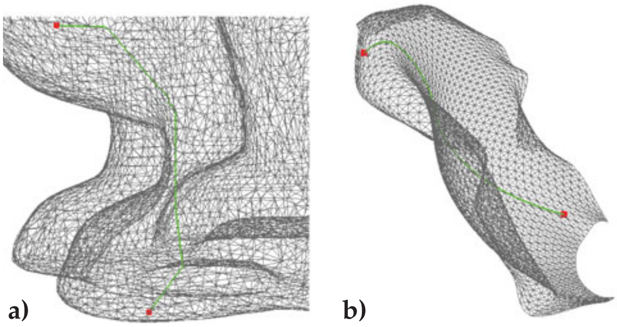

Fig. 3. The green line in the left image shows the optical path for two points on the bunny. The distance of the straight line that connects these two points will greatly over-estimate the scattering factor. The right image shows a geodesic path on a thin rose petal.

this path is simply the euclidean distance. However, for objects with concavities, euclidean distance does not accurately capture how light propagates through the translucent objects. For example, in Fig. 3a, part of the line segment that connects the two red points is outside the object. The light transport between different parts of the object through vacuum should be modeled by inter-reflection and not subsurface scattering. For objects with high curvatures (for example, a long but narrow U-shape object), the difference between the length of actual optical path and euclidean distance can be significant. This matrix of shortest distances needs to be computed only once and stored in memory for a particular scene. The subsurface scattering matrix is then computed by just indexing, based on distances, into the reflection/transmission profiles. Thus, by simply re-computing the 1D profiles, we can *vary the translucency of the scene at near-interactive rates*.

Since there is no constraint (other than energy conservation) the matrix $\mathbf{S}$ can represent an arbitrary *heterogeneous material*. Again, measurements such as those from the DISCO system [10], or numerical evaluations [11], [21], can be incorporated. We present a simple heuristic in the next section to generate plausible renderings by combining the scattering properties of multiple homogeneous materials.

5 Implementation Details

The high performance of our method relies on a carefully-designed implementation for both precomputation and the runtime solver. In this section, we will discuss our implementation in details.

5.1 Form Factor Matrix

We use the hemicube algorithm [31] to compute the form factor F_{ji}. For every polygon, we construct the hemicube over the centroid of the polygon. We then render the cube map of the scene through the hemicube, with each polygon shaded by a unique color. By counting the number of pixels each polygon projected on the cubemap, we can compute the form factors from all the other polygons to this polygon. The space complexity of this matrix is $O(n^2)$, which can be unaffordably huge for large scenes. Fortunately, due to occlusions in the geometry, the form factor matrix is usually sparse. Therefore, we can store it in sparse matrix format and make the radiosity method applicable for our test scenes.

5.2 Scattering Factor

The scattering factor (Equation (8)) can be evaluated by sampling the incident and exiting points. To reduce the computational cost, we simplify the double integral to a single integral by fixing the exiting point to be the centroid. Due to the exponential decay of subsurface scattering, the number of points on each incident polygon is determined by the distance between the centroid of the incident and outgoing polygon. In practice, the number of sampling points on the polygon itself is 100, and reduces exponentially as the distance increases. We only sample at the centroid if the distance is larger than certain threshold. Furthermore, if the distance is larger than a predefined cut-off distance max_d, we set the scattering factor to be zero.

To compute the optical distance between two points on the object surface, we construct a uniform voxel grid that covers the bounding box of the object. A ray-casting and surface-intersection test is performed to determine which voxels are inside the object. For each polygon, we find the nearest voxel grid from the centroid of the polygon. Then, regarding each interior voxel as a node of a large graph, we determine the shortest path (and distances) between the interior voxel-pair by the Dijkstra algorithm [32]. We optimize this process by using the following three strategies: First, if the incident and exiting points share the same grid, we simply compute the euclidean distance. Second, the intermediate distances computed in the algorithm are the shortest distances from the source grid, these distances can be re-used. Therefore, by caching them, we do not need to repetitively compute the distance between each pair of voxel grids. Finally, in each step of the Dijkstra algorithm, we check the distance from the source node to the current node. If the distance is longer than max_d, we stop the algorithm and set the distance to be a number larger than the cut-off distance because there is no need to compute the accurate distance. For thin sheets like the petals of a flower (see Fig. 3b), we assume constant thickness for the flower petals. Then the shortest distance between light entry and exit locations can be approximated by their geodesic distance [33].

As discussed in Section 4, our model can achieve interactive translucency variation by caching the indices to the tabular scattering profiles for a static scene. To ensure a lossless indexing to the scattering profiles, each index contain four fields: reflection and transmission profile indices (16-bit unsigned integer) and offsets in between each reflection/transmission segment (32-bit floating-point number). For each polygon, we only store the indices for all the sampling points shorter than max_d, which can greatly reduce the memory consumption. However, because the size of each index is three times the size of a distance (represented by a 32-bit scalar number), the computation of indexing matrix to the subsurface surface scattering matrix has to be performed on CPU in our current implementation.

5.3 Iterative Solver

We implement an iterative linear equation solver for Equation (14) on GPU with the CUDA SPARSE library.

TABLE 1
Performance Summary of the Five Test Scenes

Scene	# of polys	Precomputation	Memory	Relighting (*fps*)	Change-in-Translucency (*fps*)
Cornell box	24640	4 min+3 min	800MB	14	4
Bowls and bunny	73256	10 min+12 min	2.7GB	3	0.25
Rose	72474	15 min+20 min	2.7GB	3	0.5
Chess room	266307	40 min + 8 min	8.2GB	0.4	0.2
Two horses	63000	9 min + 10 min	2.4GB	5	0.4

The precomputation time includes both form factor and subsurface scattering matrices. Note that the values of the “two horses” scene are listed for each frame, not the whole animation. As shown in previous work [39], hierarchical links can further reduce the storage to less than 200 MB.

Because the curvatures of the scattering profiles may be completely different for R, G, and B component, we have to store three subsurface scattering matrices, $\mathbf{S}_r$, $\mathbf{S}_g$, and $\mathbf{S}_b$. Our implementation is tailored for a dual-GPU graphics card such as NVIDIA Geforce GTX 590 used in practice. We distribute $\mathbf{F}$ and $\mathbf{S}_r$ on GPU 1, $\mathbf{S}_g$ and $\mathbf{S}_b$ on GPU 2. Before the first iteration, we compute the emittance vector $\mathbf{L} = \mathbf{SE}$. For each iteration, we compute $\mathbf{L}_f = \mathbf{FL}$ with emittance vector $\mathbf{L}$ computed from previous iteration and store the result on GPU 1. Because peer memory access and copy is supported on dual-GPU graphics card since CUDA 4.1 [34], we only need to store $\mathbf{L}_f$ on one GPU. $\mathbf{S}_r\mathbf{L}_f$ and $\mathbf{S}_g\mathbf{L}_f$, $\mathbf{S}_b\mathbf{L}_b$ then can be executed on two GPUs in parallel. All the tasks on the two GPUs must be synchronized at the end of one iteration. For graphics cards with only one GPU, this process has to be performed sequentially. For materials with the same curvature for R, G, and B channels, we can factorize the color out of the subsurface scattering matrix and thus compute only one $\mathbf{S}$ matrix.

For large scenes whose matrix size exceeds the available graphics memory, we implement an out-of-core algorithm to compute sparse matrix vector product by working on chunks of the matrix in serial. To reduce the overhead of data transfer between CPU and GPU, we leverage the capability of overlapping data transfer and device computation for GPU devices with compute capability $>= 1.2$ [35]. We divide the available graphics memory on each GPU to two parts, one part to store the current working set and the other for loading the data from host memory. Once both the computation and data transfer are completed, the working set is set to recent-updated chunk and updates to the depreciated chunk starts simultaneously. In practice, this algorithm can effectively boost the performance by a factor of 1.5 to 3 by hiding the data transfer overhead.

6 Results

In this section, we demonstrate the efficiency and accuracy of our analytic model and algorithm. All our results were generated on a desktop with an Intel Quad-Core i7-2600k CPU and an NVIDIA Geforce GTX 590 graphics card. Our implementation was tested on five scenes. Table 1 lists the number of polygons in each scene, memory, and the precomputation and rendering times. Please see the supplementary video, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2013.256, for more visualizations.

6.1 Scenes

The *Cornell box* is a six-sided enclosure, with a yellow opaque box, and a tall marble block. The scattering profile of the marble block is computed using dipole approximation. The first row of Fig. 6 demonstrates the different components of the light transport. Our method can correctly capture both inter-reflection and subsurface scattering. Note the red and green colors on sides of the translucent block due to the inter-reflection from colored walls.

As described in Section 4, the matrix of shortest distances is the most computationally expensive part in computing the subsurface scattering. For a static scene, it needs to be computed only once and indexed into the 1D BSSRDF profiles to produce the subsurface scattering matrix. Fig. 7 shows renderings when the translucency of the block is changed by varying the 1D BSSRDF profiles.

Fig. 4 demonstrates that our model supports heterogeneous materials. Because we focus on modeling the interaction between subsurface scattering and inter-reflection, a sophisticated implementation of physically-accurate heterogeneous materials is beyond the scope of this work. We instead implement a simple heuristic for heterogeneity that can capture effects on an object with patches of different homogeneous materials. We combine the scattering profiles of each materials according to how far light traverses in that material:

$$\left(S_1(0)\frac{r_1}{r_1+r_2} + S_2(0)\frac{r_2}{r_1+r_2}\right) \times \frac{S_1(r_1)S_2(r_2)}{S_1(0)S_2(0)}, \qquad (16)$$

where, $S_1(r)$ and $S_2(r)$ are homogeneous scattering profiles, $S_1(0)$ and $S_2(0)$ are the values (diffuse albedo) at distance 0; r_1 and r_2 are the distances that light travels in each homogeneous

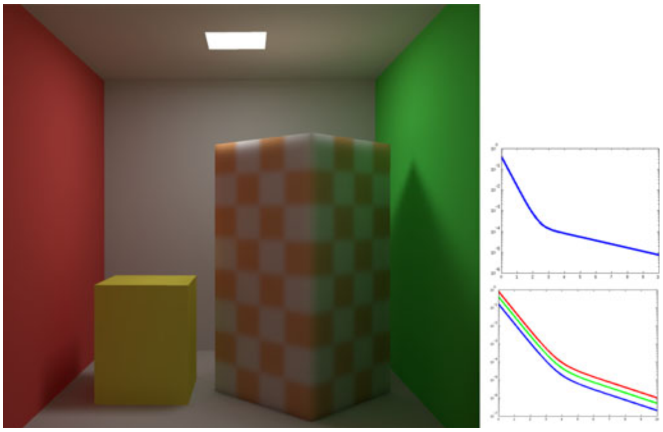

Fig. 4. Our model also supports heterogeneous translucent materials. The left image is the Cornell box scene with the tall block applied by a checkerboard pattern of translucencies. The right images are the scattering profiles of the white and orange squares. Note the color bleeding effects on both sides of the marble block and the orange color on the right part of the ceiling to the reflection from the translucent block.

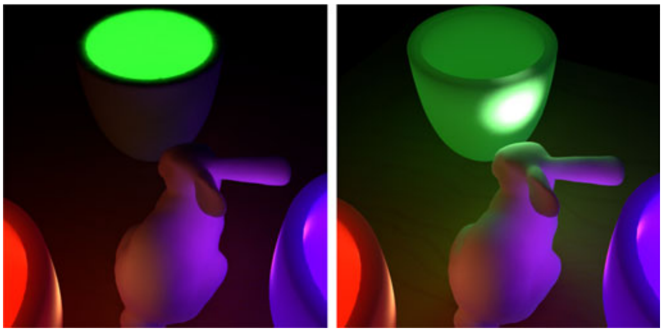

Fig. 5. Renderings of the "bowls and bunny" scene where one bowl is changed from being opaque (left) to being very translucent (right). Notice the color bleeding differences.

block. Although *not physically accurate*, this heuristic can capture plausible spatially varying appearances.

The second scene consists of *three bowls and the Stanford Bunny* on a diffuse table. The bowls and the Bunny are translucent. The scattering profiles of the bowls are based on our measurement described above but we increased translucency and modulated the colors. Different light transport components in Fig. 6 shows the importance of combining inter-reflection and subsurface scattering. Fig. 9 captures the inter-reflection between different translucent objects under different lighting conditions. We add one bounce of Fresnel reflection of translucent objects with a separate rendering pass. Multiple bounces of Fresnel reflection and more complex light paths that involve the mixing of specular and diffuse reflections are not rendered and we will set aside as future work. Fig. 5 shows renderings of this scene with different translucencies. In the left image, the bowl is almost diffuse and the light inside does not spread out. Colors of the other two bowls are clearly visible on the body of the greenish bowl. As the translucency increases, the head of the bunny and the floor receive more light.

The third object is a *rose* whose petals are modeled as having two opposing faces (or sides) with a constant thickness. The number of polygons in the mesh is therefore doubled. We simulate the light entering and exiting the faces on the same side by a reflectance profile and light entering and exiting on different sides by a transmittance profile. For this example, we used parametrized curves that exponentially fall-off with distance as the scattering profiles ($S(r) = a_1 exp(-b_1 r) + a_2 exp(-b_2 r)$). Fig. 10 shows the rendering as illuminated by the St. Peters Basilica environment map and a point light source. Notice how the light propagates through and between the petals. For comparison, the middle image shows the missing component if only subsurface scattering is simulated and the right image shows the component without the first light bounce if only inter-reflection is simulated.

The fourth scene (Fig. 8) is a *chess room*, including ten pieces of translucent chess on a diffuse board and a translucent lamp. The scene is illuminated by two point light sources, one inside the lamp and one floating in the room (rendered as a small white sphere). Due to lack of space, Fig. 8 only shows a part of the room. Please check the complementary video, available online, to see the entire scene. This is the largest scene in all our examples in terms of number of triangles (~270k). The scattering matrix of this scene is extremely sparse because each patch of the chess can only scatter light to a small number of polygons on itself. The entire scattering matrix is ~250 MB. We threshold small fractional numbers to reduce the size of the form factor

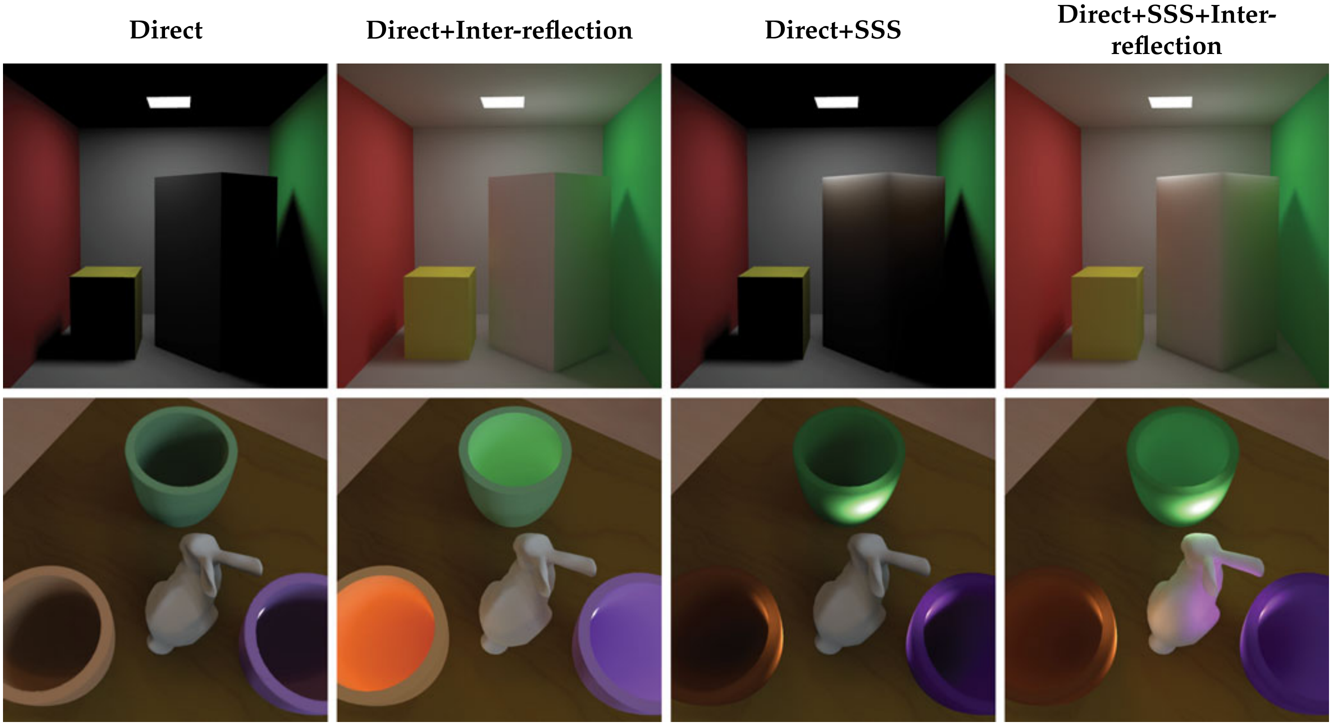

Fig. 6. Renderings of the Cornell box scene and the "bowls and bunny" scene to show different components of light transport. From left to right, each image show the direct illumination, only diffuse inter-reflection, only subsurface scattering scattering, and the results with both global illumination effects. In the Cornell box scene, our method can correctly capture the color bleeding on the translucent block due to the inter-reflection. In the "bowls and bunny" scene, only our method can faithfully model the green, orange, and magenta colors on the bunny's body.

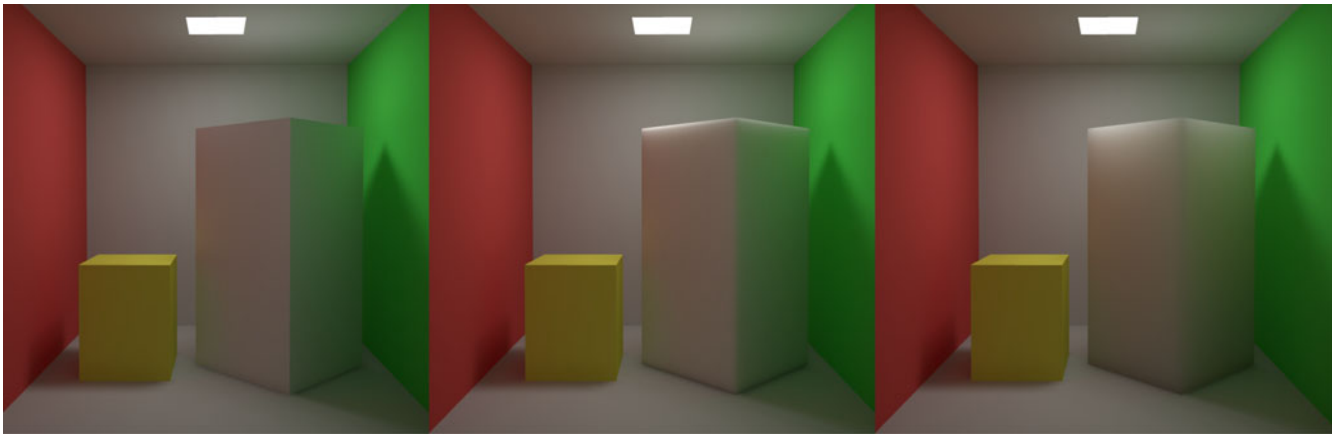

Fig. 7. The Cornell box scene with change in translucency. From left to right, the translucency of the objects increases. Note the color bleeding effects on the translucent block get less visible as the translucency increases.

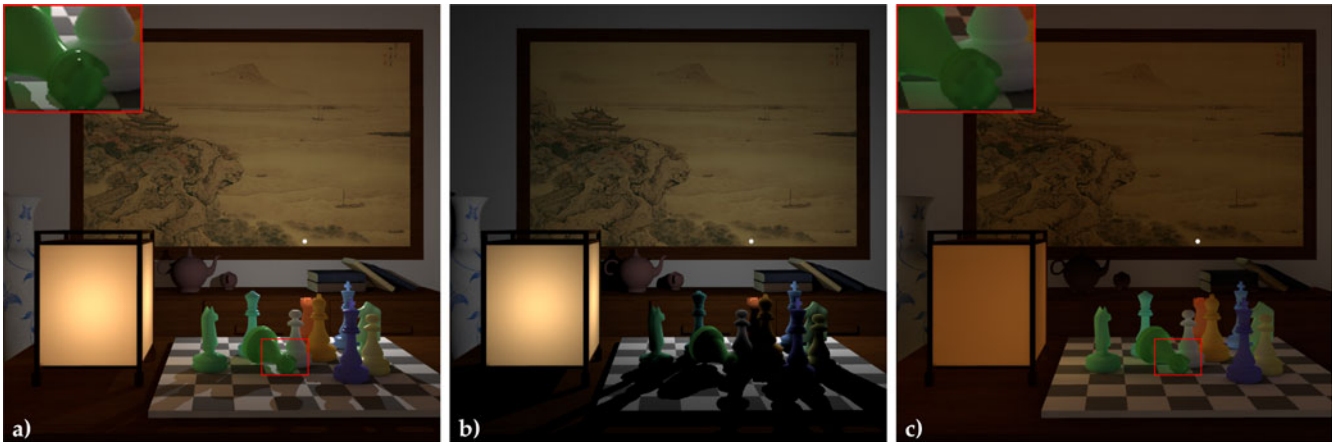

Fig. 8. Rendering of the “chess room” scene under two point light sources, one being placed inside the lamp and one floating in the room (rendered as a small white sphere). The lamp and the 10 chess pieces are translucent. Everything else in the scene is diffuse. (a) The rendering produced by our algorithm. Notice the green and magenta colors on the white chess piece (highlighted region), which both contributes to the indirect lighting from the green and purple chess pieces. The chess board also receives yellow indirect light from the lamp. (b) The shadow on the board is completely dark if only subsurface scattering is simulated. (c) The missing global illumination of image (b).

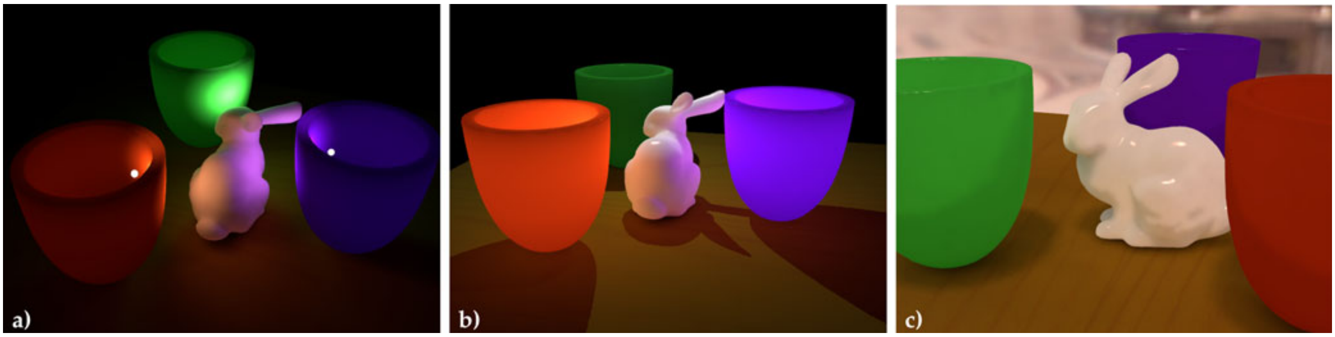

Fig. 9. Rendering of the “bowls and bunny” scene under different lighting conditions. (a) Light scattered through three colored bowls illuminates different parts of the bunny. The scene is illuminated by three spot light sources, with one located inside each bowl. (b) The same scene is illuminated by three point light sources, one above the green bowl and two inside the other two bowls. (c) The same scene illuminated by environment lighting. We add one-bounce specular reflection on translucent objects for (b) and (c). Multiple bounces are not simulated.

matrix to 8 GB, which unfortunately cannot be completely fit into the graphics card. We use the out-of-core algorithm described in Section 5.3 to compute **FL**. Although the relighting speed of this scene is much slower than other examples, we can still render at near-interactive speed, achieving 0.4 fps.

Directly composing the rendering by the radiosity of each patch causes noticeable artifacts for high-frequency shadows. Instead, we factorize the final radiosity into direct and global illumination and replace the direct illumination on diffuse objects with the direct illumination generated by the shadow mapping algorithm [36], [37]. The shadows on

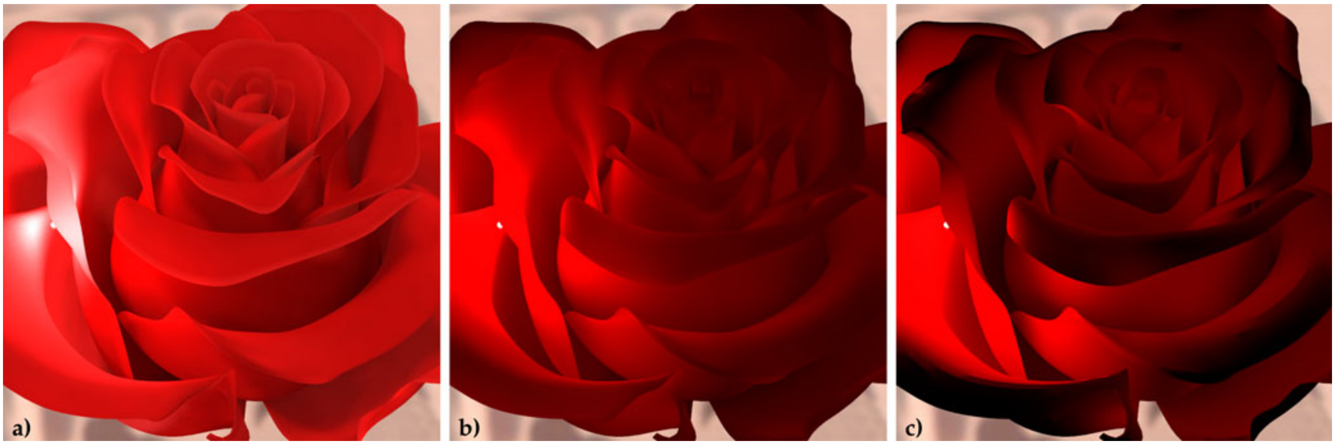

Fig. 10. (a) A rose model (72k polygons) rendered with the lighting from St. Peter's Basilica environment map (debevec.org) and a point light source between the petals. (b) The missing light transport component if only subsurface scattering is simulated. (c) The global illumination if the rose was opaque and only diffuse inter-reflection (classical radiosity) is simulated.

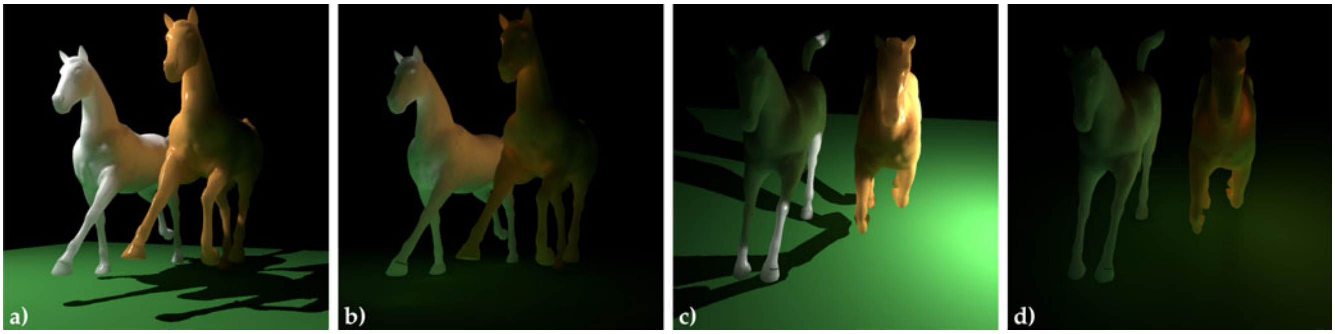

Fig. 11. (a) and (c): Two renderings of the "two horses" animation. Notice the green and brown colors on the white horses due to inter-reflection. (b) and (d): The missing light transport components if only subsurface scattering is simulated.

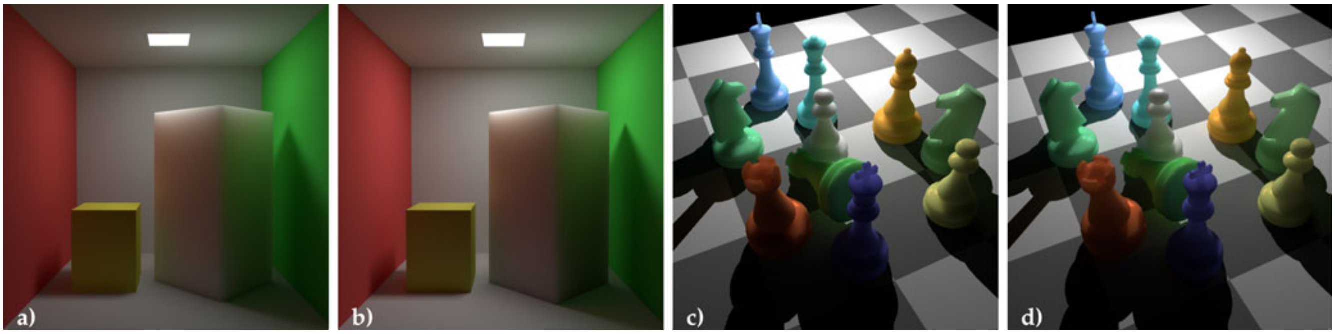

Fig. 12. Comparisons between the renderings generated by our algorithm (a and c) and the reference images rendered by volumetric path tracing (b and d). In the first scene, the red and green colors on the translucent blocks in our rendering is very similar to the reference image. In the second scene, our algorithm can correctly capture the green colors on the white chess piece and the board as shown in the path tracing image. However, The per-patch radiosity solution introduces noticeable artifacts to the shadow behind translucent blocks in the Cornell Box scene. In the second scene, although we adopt a hybrid rendering method [36], [37] to remove the artifacts on diffuse surfaces for point light sources, the shadow artifacts on translucent objects are still visible.

translucent objects are low-frequency, and thus require no special handling.

The last scene is an animation with *two horses* galloping on a green floor, illuminated by a point light source. The entire animation contains 48 meshes (downloaded from MIT deformation database[1]). For each mesh, we precompute the form factor and subsurface scattering matrices. Once the two matrices are loaded from hard drive, the computation is performed as other scenes. Figs. 1 and 11 show renderings of different meshes produced by our algorithm. Please see the accompanied video, available online, for the full animation.

6.2 Validation

Radiosity is a well validated method and our model is a physics-based extension. Iterative methods are commonly used to solve the radiosity equation, extensively studied

[1] http://people.csail.mit.edu/sumner/research/deftransfer/data.html.

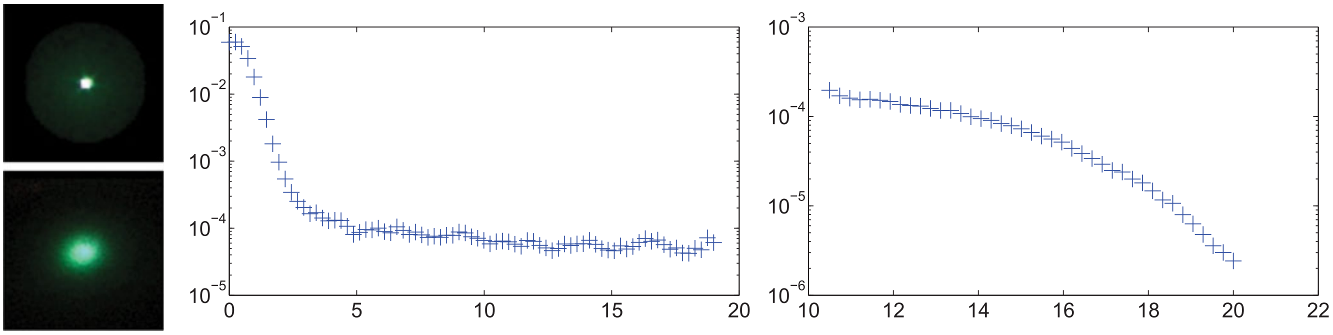

Fig. 13. [Left] Photos of the reflection (top) and transmission (bottom) profiles. [middle] and [right] are the reflection and transmission profiles computed from the photos. The unit of the *X*-axis is millimeter.

for many years. In practice, for our test scenes, it takes 5-10 iterations to converge with a residual error of 0.5 percent. We compare our rendering results with images rendered by a physically based rendering engine, Mitsuba [38] on two scenes (Fig. 12). Mitsuba is a versatile toolkit that consists of a wide range of global illumination algorithms, in which we choose the extended volumetric path tracer to generate our reference images. We set the maximum path depth [38] as 20 and the number of samples per pixel to be 4,096. Our first scene is the traditional five-sided Cornell box scene, similar to our first example. The second scene is a part of our *chess room* scene, including only the chess pieces and the board, illuminated by a point light source. For the translucent objects in both scenes, we use the Henyey-Greenstein phase function. The overall appearance of our renderings matches the path tracer, which suggests the accuracy of our algorithm. However, because our algorithm does not include single scattering term, the intensity of our algorithm in thin regions, such as the corner of the translucent block, is not as bright as the reference image.

To further verify the accuracy of our model, we compared our renderings of a *concave wax bowl* with photographs (Fig. 14). The interior of the bowl is illuminated by a strong spot light (flashlight). The resulting brightness observed on the outside surface shows that the bowl is highly translucent. We setup a projector-camera system to measure the reflection and transmission profiles. We carefully calibrated the geometric relationship between the projector and the camera. The color and brightness of the projector were calibrated by a Gretag Macbeth ColorChecker. We measured the radially symmetric BSSRDF profiles by capturing the light distribution on the bowl while illuminating only one pixel of the projector (Fig. 13). Then the profiles are computed by averaging the intensities of the pixels with the same distance to the incident point.

The reflection and transmission profiles are then smoothed and used to compute the renderings in Fig. 14. The colors of the source and intensity mapping (gamma) were done manually to match the photographs. Visual comparisons show that our renderings are very similar to the photos. Because we do not have an accurate mesh (with small bumps on the bowl), and spatial-varying scattering profiles, subtle differences in local shading can be observed between our renderings and the photos.

6.3 Comparison with Heuristic Methods

Our model is designed precisely to avoid different types of heuristics, for example, substituting translucency inter-reflection with diffuse inter-reflection [3]. Depending on how diffuse inter-reflection plays its role, this may mean either $(\mathbf{I}-\mathbf{F})^{-1}\,\mathbf{SE}$ or $\mathbf{SE}+(\mathbf{I}-\mathbf{F})^{-1}\,\mathbf{E}$ with our notation. The first heuristic method only models one bounce of translucency inter-reflection and later bounces are substituted with diffuse inter-reflection, while the second one completely considers diffuse inter-reflection and composites the final rendering by adding scattering and diffuse inter-reflection. Both methods cause noticeable errors for multiple layers of translucent objects or objects with different diffuse and scattering colors. Fig. 15 shows comparisons between our method and these two heuristics. The scene is illuminated by three spot-lights, one inside each bowl, and we set the diffuse colors to match the colors of the scattering profiles. Only our method can faithfully

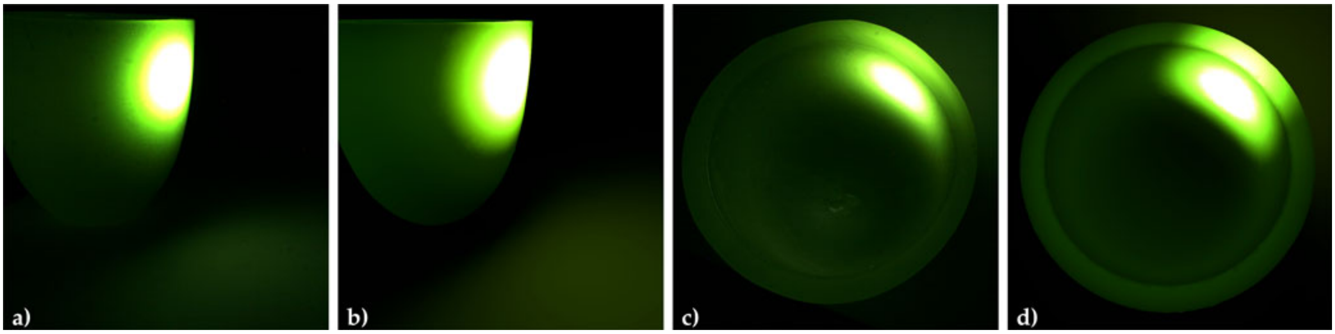

Fig. 14. Comparisons between our renderings (b and d) and photographs (a and c) for a translucent wax bowl.

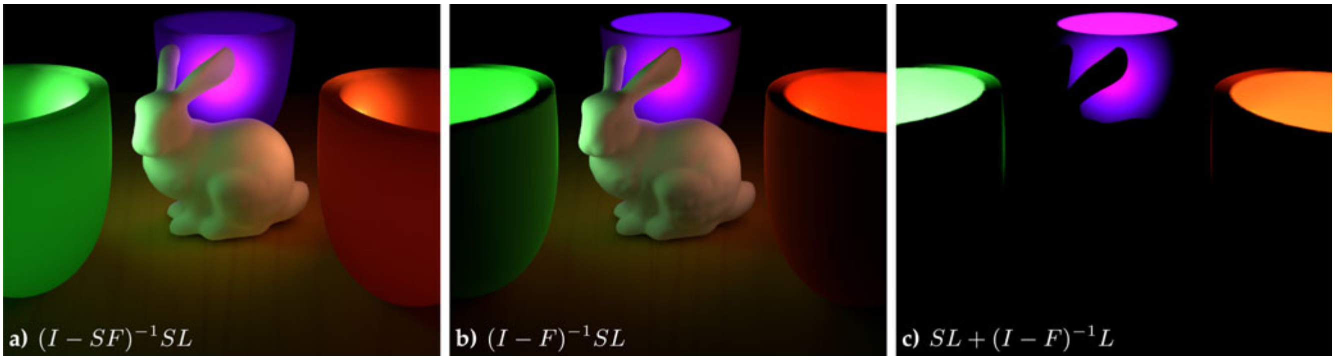

Fig. 15. Comparisons between our model and two heuristic methods. (a) Our model can correctly capture the intertwined light transport of inter-reflection and subsurface scattering. (b) The first heuristic method can only capture one bounce of translucent inter-reflection, and therefore the outside of the bowls and the bunny are much darker than the correct solution. The geometry detail on the bunny is more obvious than our method. (c) The second heuristic method uses diffuse inter-reflection to replace translucent inter-reflection and therefore light cannot scatter outside the bowls at all.

capture all the light transport within the bowls and between the bowls and the bunny.

6.4 Performance

For simultaneous relighting and translucency change, we use an iterative linear system solver (Section 5.3), typically converging within 5-10 iterations. Our CUDA solver achieves a reasonable performance boost, with 5-10 fps for a scene with 40 k polygons, while the CPU implementation runs at 1-2 fps. For the same scene, the precomputation of form factor and subsurface scattering matrices takes about 10 minutes. Since our algorithm is a scene-space method, changing viewpoint is real-time for free. If only relighting is considered, precomputing matrix $\mathbf{T} = (\mathbf{I} - \mathbf{SF})^{-1}\mathbf{S}$ can improve the relighting speed to 50-80 fps, achieving real-time rates. Comparing the performance with two related works in interactive rendering of translucent objects, our relighting speed is faster than relighting translucent objects with PRT [7], and comparable to that reported in Wang et al.'s work [8]. However, our method models the light interactions between translucent objects and the environment while the two methods focus on rendering individual translucent objects. In addition, the precomputation time of our method is significantly shorter than that of these two methods (minutes versus hours).

The downside of any radiosity-based method is the memory consumption. Fortunately, due to occlusions in the geometry and locality of subsurface scattering, both $\mathbf{F}$ and $\mathbf{S}$ are sparse. We further reduce the memory consumption by thresholding small fractional numbers without introducing visible difference. Previous work of face cluster radiosity [39] shows that it is able to consume only 120 MB memory for a 2.7 million polygonal mesh by adopting volume clustering, therefore similar hierarchical links can further reduce our storage.

7 Limitations

Our method builds on physics to analytically model the inter-reflection and subsurface scattering between translucent objects and the environment. However, our approach shares the shortcomings of classical radiosity techniques as compared to ray-tracing methods. As a scene-based approach, our method achieves interactive rendering speed when the lighting or camera changes. However, this merit comes with the penalty of one time of precomputation and memory storage, which can be expensive for large scenes. Furthermore, the per-patch lighting solution restricts the spatial variations of both translucent materials and diffuse colors. This problem also extensively exists in PRT, whose lighting solution is computed per-vertex. In addition, our model cannot accurately simulate translucent objects for which single scattering dominates the subsurface scattering due to the assumption of view-independence. An interesting future work is to mitigate the requirement and thus enable our method to render more types of translucent materials.

8 Conclusions

We presented an analytic model to combine two specific forms of light transport—diffuse inter-reflection and isotropic subsurface scattering. Our approach is simple, easy to implement, extends classical radiosity with little additional cost and can capture interesting effects due to homogeneous and heterogeneous translucency, relighting translucent scenes and controlling object translucency, at near interactive rates. Further speed-ups can be achieved by using hierarchical and adaptive implementations [15], [39], which we will leave as future work. We are also interested in extending the model to interactively render deformable translucent objects, probably using data-driven approaches [40].

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. They also thank Shuo Wang and Changyu Diao for creating some of the models. Yu Sheng and Srinivasa G. Narasimhan were funded in parts by US National Science Foundation (NSF) CAREER award IIS-0643628, NSF grant IIS-0964562, and ONR grant N00014-11-1-0295. Yulong Shi and Lili Wang were supported in parts by NSFC grants 61272349 and 61190121.

References

[1] H.W. Jensen, J. Legakis, and J. Dorsey, "Rendering of Wet Materials," *Proc. Eurographics Rendering Workshop*, June 1999.

[2] M. Pharr and P.M. Hanrahan, "Monte Carlo Evaluation of Non-Linear Scattering Equations for Subsurface Reflection," *Proc. ACM SIGGRAPH '00*, pp. 75-84, July 2000.

[3] H.W. Jensen and J. Buhler, "A Rapid Hierarchical Rendering Technique for Translucent Materials," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 576-581, July 2002.

[4] C. Donner and H.W. Jensen, "Rendering Translucent Materials Using Photon Diffusion," *Proc. 18th Eurographics Workshop Rendering (Rendering Techniques '07)*, pp. 243-252, June 2007.

[5] P.-P. Sloan, J. Kautz, and J. Snyder, "Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 527-536, July 2002.

[6] P.-P. Sloan, J. Hall, J. Hart, and J. Snyder, "Clustered Principal Components for Precomputed Radiance Transfer," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 382-391, July 2003.

[7] R. Wang, J. Tran, and D. Luebke, "All-Frequency Interactive Relighting of Translucent Objects with Single and Multiple Scattering," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 1202-1207, Aug. 2005.

[8] R. Wang, 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," *The Visual Computer*, vol. 24, no. 7, pp. 565-575, July 2008.

[9] H.W. Jensen, S.R. Marschner, M. Levoy, and P. Hanrahan, "A Practical Model for Subsurface Light Transport," *Proc. ACM SIGGRAPH*, pp. 511-518, Aug. 2001.

[10] M. Goesele, H.P.A. Lensch, J. Lang, C. Fuchs, and H.-P. Seidel, "Disco: Acquisition of Translucent Objects," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 835-844, Aug. 2004.

[11] J. Wang, S. Zhao, X. Tong, S. Lin, Z. Lin, Y. Dong, B. Guo, and H.-Y. Shum, "Modeling and Rendering of Heterogeneous Translucent Materials Using the Diffusion Equation," *ACM Trans. Graphics*, vol. 27, no. 1, pp. 9:1-9:18, Mar. 2008.

[12] C. Goral, K. Torrance, D. Greenberg, and B. Battaile, "Modelling the Interaction of Light between Diffuse Surfaces," *Proc. ACM SIGGRAPH*, pp. 213-222, July 1984.

[13] C. Donner and H.W. Jensen, "Light Diffusion in Multi-Layered Translucent Materials," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 1032-1039, Aug. 2005.

[14] M. Cohen, S.E. Chen, J. Wallace, and D. Greenberg, "A Progressive Refinement Approach to Fast Radiosity Image Generation," *Proc. ACM SIGGRAPH*, pp. 75-84, Aug. 1988.

[15] P. Hanrahan, D. Salzman, and L. Aupperle, "A Rapid Hierarchical Radiosity Algorithm," *Proc. ACM SIGGRAPH*, July 1991.

[16] J. Wallace, M. Cohen, and D. Greenberg, "A Two-Pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods," *Proc. ACM SIGGRAPH*, vol. 21, no. 4, pp. 311-320, July 1987.

[17] P.H. Christensen, D. Lischinski, E.J. Stollnitz, and D.H. Salesin, "Clustering for Glossy Global Illumination," *ACM Trans. Graphics*, vol. 16, no. 1, pp. 3-33, Jan. 1997.

[18] H. Rushmeier and K. Torrance, "The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium," *Proc. ACM SIGGRAPH '87*, pp. 293-302, July 1987.

[19] H.E. Rushmeier and K.E. Torrance, "Extending the Radiosity Method to Include Specularly Reflecting and Translucent Materials," *ACM Trans. Graphics*, vol. 9, no. 1, pp. 1-27, Jan. 1990.

[20] F.E. Nicodemus, J.C. Richmond, J.J. Hsia, I.W. Ginsberg, and T. Limperis, "Geometrical Considerations and Nomenclature for Reflectance," *Science and Technology*, vol. 160, 1977.

[21] A. Arbree, B. Walter, and K. Bala, "Heterogeneous Subsurface Scattering Using the Finite Element Method," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 7, pp. 956-969, July 2011.

[22] Y. Wang, J. Wang, N. Holzschuch, K. Subr, J.-H. Yong, and B. Guo, "Real-Time Rendering of Heterogeneous Translucent Objects with Arbitrary Shapes," *Computer Graphics Forum*, vol. 29, pp. 497-506, Apr. 2010.

[23] W. Jakob, A. Arbree, J.T. Moon, K. Bala, and S. Marschner, "A Radiative Transfer Framework for Rendering Materials with Anisotropic Structure," *ACM Trans. Graphics*, vol. 29, no. 4, pp. 53:1-53:13, July 2010.

[24] A. Arbree, B. Walter, and K. Bala, "Single-Pass Scalable Subsurface Rendering with Lightcuts," *Computer Graphics Forum*, vol. 27, no. 2, pp. 507-516, 2008.

[25] H.P.A. Lensch, M. Goesele, P. Bekaert, J. Kautz, M.A. Magnor, J. Lang, and H.-P. Seidel, "Interactive Rendering of Translucent Objects," *Proc. 10th Pacific Conf. Computer Graphics and Applications*, pp. 214-224, Oct. 2002.

[26] N.A. Carr, J.D. Hall, and J.C. Hart, "GPU Algorithms for Radiosity and Subsurface Scattering," *Proc. ACM SIGGRAPH/EUROGRAPHICS Conf. Graphics Hardware*, pp. 51-59, July 2003.

[27] T. Mertens, J. Kautz, P. Bekaert, H.-P. Seidel, and F.V. Reeth, "Interactive Rendering of Translucent Deformable Objects," *Proc. Eurographics Symp. Rendering: 14th Eurographics Workshop Rendering*, pp. 130-140, June 2003.

[28] R. Ng, R. Ramamoorthi, and P. Hanrahan, "All-Frequency Shadows Using Non-Linear Wavelet Lighting Approximation," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 376-381, July 2003.

[29] M. Cohen and J. Wallace, *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Inc., 1993.

[30] S. Premoze, M. Ashikhmin, J. Tessendorf, R. Ramamoorthi, and S. Nayar, "Practical Rendering of Multiple Scattering Effects in Participating Media," *Proc. Eurographics Symp. Rendering*, 2004.

[31] M. Cohen and D. Greenberg, "The Hemi-Cube: A Radiosity Solution for Complex Environments," *ACM Computer Graphics*, vol. 19, no. 3, pp. 31-40, Aug. 1985.

[32] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, pp. 269-271, 1959.

[33] J.S.B. Mitchell, D.M. Mount, and C.H. Papadimitriou, "The Discrete Geodesic Problem," *SIAM J. Computing*, vol. 16, pp. 647-668, Aug. 1987.

[34] "CUDA Programming Guide 4.1," NVIDIA, https://developer.nvidia.com/cuda-toolkit-41-archive, 2011.

[35] "CUDA C Best Practice Guide 4.1," NVIDIA, https://developer.nvidia.com/cuda-toolkit-41-archive, 2011.

[36] Y. Sheng, T.C. Yapo, C. Young, and B. Cutler, "A Spatially Augmented Reality Sketching Interface for Architectural Daylighting Design," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 1, pp. 38-50, Jan. 2011.

[37] Y. Sheng, T. Yapo, C. Young, and B. Cutler, "Virtual Heliodon: Spatially Augmented Reality for Architectural Daylighting Design," *Proc. IEEE Virtual Reality Conf.*, 2009.

[38] W. Jakob, "Mitsuba Renderer," http://www.mitsuba-renderer.org, 2010.

[39] A. Willmott, P. Heckbert, and M. Garland, "Face Cluster Radiosity," *Proc. Eurographics Rendering Workshop*, June 1999.

[40] M. Stanton, Y. Sheng, M. Wicke, F. Perazzi, A. Yuen, S. Narasimhan, and A. Treuille, "Non-Polynomial Galerkin Projection on Deforming Meshes," *ACM Trans. Graphics*, vol. 32, no. 4, pp. 86:1-86:14, July 2013.

Yu Sheng received the PhD degree from Rensselaer Polytechnic Institute in 2011 on the "interactive daylighting visualization in spatially augmented reality environments" and was a postdoctoral fellow at the Robotics Institute, Carnegie Mellon University, from 2011 to 2013. He is a research scientist in Bosch Research and Technology Center, North America. His works received the Best Paper Runner-up Prize at ACM I3D 2013 and Best Paper Nomination at Eurographics 2010. He is a member of the IEEE.

Yulong Shi received the bachelor's degree in computer science from China University of Geoscience, Wuhan, China in 2010 and the master's degree in computer science from China State Key Laboratory of Virtual Reality Technology and System of Beihang University, Beijing, China in 2013.

Lili Wang received the PhD degree from the Beihang University, Beijing, China, where she was an assistant professor and an associate professor at School of Computer Science and Engineering from 2005 and 2007. She is also a researcher at the State Key Laboratory of Virtual Reality Technology and Systems. Her interests include the real-time rendering, realistic rendering, global illumination, soft shadow, and texture synthesis.

Srinivasa G. Narasimhan received the PhD degree from Columbia University in February 2004. He is an associate professor in the Robotics Institute at Carnegie Mellon University. His group focuses on novel techniques for imaging, illumination and light transport to enable applications in vision, graphics, robotics and medical imaging. His works have received several awards: FORD URP Award (2013), Best Paper Runner up Prize (ACM I3D 2013), Best Paper Honorable Mention Award (IEEE ICCP 2012), Best Paper Award (IEEE PROCAMS 2009), the Okawa Research Grant (2009), the NSF CAREER Award (2007), Adobe Best Paper Award (IEEE Workshop on Physics based methods in computer vision, ICCV 2007) and IEEE Best Paper Honorable Mention Award (IEEE CVPR 2000). His research has been covered in popular press including NY Times, BBC, PC magazine, and IEEE Spectrum and is highlighted by NSF and NAE. He is the coinventor of smart headlights for seeing through rain and snow which made several top 10 lists of promising technologies including Car and Driver and Edmunds. He is also the coinventor of Aqualux 3D display, Assorted-pixels and Motion-aware cameras and low-power outdoor “Kinect.” He cochaired the International Symposium on Volumetric Scattering in Vision and Graphics in 2007, the IEEE Workshop on Projector-Camera Systems (PROCAMS) in 2010, and the IEEE International Conference on Computational Photography (ICCP) in 2011, is coediting a special journal issue on Computational Photography, and serves on the editorial board of the *International Journal of Computer Vision* and as an area chair of several conferences (CVPR, ICCV, ECCV). He is a member of the IEEE.