Rendering of Translucent Objects Based upon PRT

Jin Zhou Zhang Jiawan Sun Jizhou
School of Computer Science and Technology, Tianjin University
Tianjin 30072, P.R.China
hill0731@tju.edu.cn

Abstract

A framework to incorporate subsurface scattering effects is proposed in this paper based on precomputed light transport technology, which can interactively rerender translucent objects. First, the complete BSSRDF model is considered and the two-pass hierarchical technique by [5] in the space of non-linearly approximated transport vectors is applied, which efficiently evaluate transport vectors due to diffuse multiple scattering. Single scattering is also incorporated by considering [16]'s approximation. For a general phase function, a technique from BRDF rendering is extended which using a separable decomposition to allow decoupled precomputation and rendering [8]. Using this approach, interactive frame rates with comparable speed to the state-of-the-art systems is achieved.

1. Introduction

Most illumination models give the bidirectional reflectance distribution function (BRDF). The basic assumption that light enters and exits the object surface at the same point for BRDF models is not always valid. As an example, BRDF is inadequate translucent materials, such as marble, skin, milk, etc. For simulating the appearance of these materials, we need the more general bidirectional surface scattering reflectance distribution function (BSSRDF). To simulate the BSSRDF to a full extent using either radiance transfer, or Monte Carlo simulation, or path tracing will be too slow for interactive rendering. A lot of papers put forward a wide range of proposed methods for describing such material, such as finite element methods [1], path tracing [3][9], photon mapping [2][6] and diffusion approximation [14]. Recently, By assuming homogeneous media, [7] formulated the BSSRDF as the sum of single scattering and a diffuse dipole approximation for multiple scattering. [5] then presented a two-pass hierarchical integration technique to accelerate the computation of the multiple scattering components remarkably. However, previous rendering algorithms handling subsurface scattering do not nearly allow interactive image synthesis. Several recent papers, such as [4][10][16] exploit this property and implement interactive systems for rendering multiple scattering. Our system is similar with them and supporting soft-shadow, directional illumination and reflection.

Precomputed Radiance Transfer (PRT) was first proposed by Sloan et al. in [12]. In PRT, spherical harmonic (SH) function is employed for the decomposition of incident lighting and geometric scattering. It replaces the integral of the rendering equation by a simple set of dot products using orthogonality of spherical harmonics. Rerendering then reduce to the inner product of a light vector, represented in the same SH basis, with the precompiled transport vectors. This enables dynamic movements of lights, using environment mapping and a rigid object, at an interactive speed. Later, [13] included the diffuse multiple scattering component of the BSSRDF in their PRT framework, which was extended by [15] and [11]. Single scattering is simulated by a glossy BRDF, but is not physically based. One challenge of incorporating BSSRDF is to find a compact formula of light transport function that makes precomputation feasible. [13] improved the rendering speed by compressing the matrix of spherical harmonic coefficients using clustered principal component analysis (CPCA) in [13]. Similarly, ours strategy is to compress the computation for every vertex response to each lighting basis, and use non-linear approximation later.

In this paper, the model that we chose to implement sub-surface scattering was the work proposed by [5]. Rapid Hierarchical Rendering was considered which contains both single and diffusing multiple scattering component. In multiple diffusing components, we apply their two-pass hierarchical technique [5] in the space of transport vectors, and used compressed approximation by wavelet basis. In the first pass, in order to sampling the irradiance, we compute

0-7695-2929-1/07 $25.00 © 2007 IEEE
DOI 10.1109/ICIG.2007.87

irradiance transport vectors for each selected point on the surface, which describe irradiance values parameterized on the lighting environment. In the second pass, based upon the pre-computed irradiance samples, a rapid hierarchical evaluation of the diffusion approximation was utilized to integrate the precomputed irradiance transport vectors to compute a per-vertex multiple scattering transport vector. Since single scattering component is the 1st-order source term of diffusion approximations and expensive to fully simulated. We use a similar approximated formula derived from [16] for single scattering. For an isotropic phase function, we evaluate a diffuse single scattering transport vector per vertex and combine it with the multiple scattering components. For a general phase function, we use a technique similar to [16]. The phase function used a separable decomposition [8] and keeps K loworder terms, and each consisting of a purely light-dependent part and a purely view-dependent part. Rendering results are showed in this paper and the quality of the appearance of translucent objects is comparable to the state of the art. This interactive system incorporates all-frequency environment lighting, reflection and subsurface scattering with both single and multiple scattering.

2. Background

In this section, we will make a brief introduction about BSSRDF formulation by Jensen et al. There are two assumptions made in this paper. The first is that the media on the "inside" of the interface scatters far more light than it observes. In other word, the participating media is homogeneous. The properties are characterized by the absorption coefficient σ_α, and the scattering coefficient σ_s. Here, the extinction coefficient is $\sigma_t = \sigma_\alpha + \sigma_s$ is supposedly very close to equation 1. The second restriction asserts that the approximation is only valid for locally flat surfaces. The following BSSRDF formulation describe subsurface scattering in homogeneous medium:

$$L_o(x_o,\vec{\omega}_o) = \int_A \int_{2\pi} S(x_i,\vec{\omega}_i;x_o,\vec{\omega}_o) L(x_i,\vec{\omega}_i)(\vec{n}_i \bullet \vec{\omega}_i) d\vec{\omega}_i dA(x_i) \quad (1)$$

Here, L_o is the outgoing radiance at point x_o in direction $\vec{\omega}_o$, L is the incident radiance at point x_i in direction $\vec{\omega}_i$, and S is the given BSSRDF. [7] thought that the BSSRDF as the sum of a single scattering term $S^{(1)}$ and a diffuse multiple scattering term S_d:

$$S(x_i,\vec{\omega}_i;x_o,\vec{\omega}_o) = S^{(1)}(x_i,\vec{\omega}_i;x_o,\vec{\omega}_o) + S_d(x_i,\vec{\omega}_i;x_o,\vec{\omega}_o)$$

Approximating S_d into a dipole source model, they derive S_d as:

$$S(x_i,\vec{\omega}_i;x_o,\vec{\omega}_o) = \frac{1}{\pi} F_t(\eta_i,\vec{\omega}_i) R_d(\|x_i - x_o\|) F_t(\eta,\vec{\omega}_o) \quad (2)$$

where η is the relative index of refraction, F_t is the Fresnel transmittance and R_d is the diffuse reflectance computed by:

$$R_d(r) = \frac{\alpha'}{4\pi}\left[z_r\left(\sigma_{tr} + \frac{1}{d_r}\right)\frac{e^{-\sigma_{tr} d_r}}{d_r^2} + z_v\left(\sigma_{tr} + \frac{1}{d_v}\right)\frac{e^{-\sigma_{tr} d_v}}{d_v^2}\right] \quad (3)$$

where $\sigma_s' = (1-g)\sigma_s$ and $\sigma_t' = \sigma_\alpha + \sigma_s'$ are reduced scattering and extinction coefficients, $\alpha' = \sigma_s' / \sigma_t'$ is the reduced albedo, g is the mean cosine of the scattering angle, $\sigma_{tr} = \sqrt{3\sigma_\alpha \sigma_t'}$ is the effective extinction coefficient, $d_r = \sqrt{r^2 + z_r^2}$ and $d_v = \sqrt{r^2 + z_v^2}$ are the distances from illumination point xi to the dipole source, $r = \|x_o - x_i\|$ is the distance between x_i and x_o, and $z_r = 1/\sigma_t'$ and $z_v = z_r(1 + 4A/3)$ are the distances from x_o to the dipole source. Here $A = (1 + F_{dr})/(1 - F_{dr})$, and F_{dr} is a diffuse Fresnel term $F_{dr} = -1.440/\eta^2 + 0.710/\eta + 0.668 + 0.0636\eta$.

This approximation is somehow expensive in implementation, and that is because, for each point we need to compute an integral on the whole surface to compute the contribution of all light rays interacting with the material [5], has proposed a hierarchical method which decompose compute process into two pass. In the first pass, they carried on uniformly sampling on the surface, these sample points are usually organized and stored in octree, with each node storing represented values of all its child nodes. In the second pass, they traverse through the octree with evaluating the node recursively, and use a simple heuristic for estimation of recursive convergence.

Single scattering is computed by the following equation in which the BSSRDF single scattering term $S^{(1)}$ is implicitly defined:

$$L_o^{(1)}(x_o,\vec{\omega}_o)=\sigma_s\int_{2\pi}\int_0^{\infty}Fp(\vec{\omega}_i^{'},\vec{\omega}_o^{'})e^{-\sigma_t(s_i^{'}+s)}L(x_i,\vec{\omega}_i)dsd\vec{\omega}_i \quad (4)$$

where $\vec{\omega}_i^{'}$ and $\vec{\omega}_o^{'}$ are the refracted incoming and outgoing directions, $F=F_t(\eta,\vec{\omega}_i^{'})\bullet F_t(\eta,\vec{\omega}_o^{'})$ is the combined Fresnel transmittance, $s_i^{'}$ and s are the scattering path lengths along $\vec{\omega}_i^{'}$ and $\vec{\omega}_o^{'}$, and p is a normalized phase function. When sampling the illumination, it is usually difficult to estimate $s_i^{'}$ accurately, and a good approximation of $s_i^{'}$ can be found by:

$$s_i^{'}=s_i\frac{|\vec{\omega}_i\bullet\vec{n}_i|}{\sqrt{1-\left(\frac{1}{\eta}\right)^2\left(1-|\vec{\omega}_i\bullet\vec{n}_i|^2\right)}} \quad (5)$$

where s_i is the observed path length unless the incident ray is refracted. The single scattering component (equation 4) is derived from previous work by [3] and is computed by Monte Carlo integration along $\vec{\omega}_o^{'}$. Fig 1(a) shows the scenario.

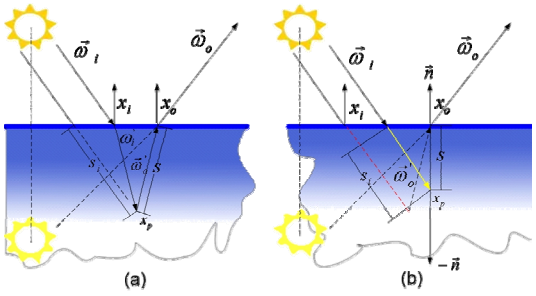


Fig 1. Single scattering

(a) Single scattering is the result of integral along the refracted outgoing ray. The approximation of observed path length is by equation 5. (b) The yellow solid line shows the approximated observed path length, and the red dotted line is the true

3. Algorithms and Implementation

3.1. Multiple Scattering

The multiple scattering components in our implementation are derived from [16]. They assumed illumination is the product of incident light and visibility:

$$L(x_i,\vec{\omega}_i)=L(\vec{\omega}_i)V(x_i,\vec{\omega}_i) \quad (6)$$

where V is the visibility. They rearranged diffusing component to:

$$L_d(x_o,\vec{\omega}_o)=\frac{1}{\pi}F_t(\eta,\vec{\omega}_o)\sum_j T_d(x_o,\vec{\omega}_i)L(\vec{\omega}_i)$$

$$T_d(x_o,\vec{\omega}_i)=\int_A R_d(\|x_i-x_o\|)E(x_i,\vec{\omega}_i)dA(x_i) \quad (7)$$

$$E(x_i,\vec{\omega}_i)=F_t(\eta,\vec{\omega}_i)V(x_i,\vec{\omega}_i)(\vec{n}_i\bullet\vec{\omega}_i) \quad (8)$$

where T_d is the irradiance transport function to be evaluated, E is the irradiance transport function at point of illumination x_i which describes light transport from the environment to x_i, and the diffuse reflectance R_d (defined in equation 7) predicts the light transport from x_i to x_o, similar to form factors in radiosity methods.

Since spherical harmonics, wavelets and L are formulated in orthonormal basis, it can apply efficiently compressing, storage and rendering. Like [11], we use a high-resolution environment cubemap to represent a all-frequency, time-varying illumination, and approximate the environment map in a wavelet basis, keeping only the largest terms(non-linear approximation), and obtain further compression by encoding the light transport matrix sparsely but accurately in the same basis.

To compute T_d efficiently, we apply the two-pass hierarchical technique by [5]. Specifically, in the first pass we compute irradiance transport functions E for a set of uniformly distributed surface sample points. We rasterize a high-resolution visibility hemi-cube at each x and then rendered raw transport matrix. Then we carry on a wavelet transform on these transport functions, with keeping only a bounded number of the largest wavelet coefficients for approximation then dither and quantize the matrix elements to 6, 7 or 8 bits, depending on the desired level of fidelity, and discard all zero coefficients. In the second pass, we use an octree by clustering and summing the irradiance transport vectors computed in the first pass. To reduce memory consumption, we always apply wavelet non-linear compression at each node, and truncate the remaining. For each vertex we traverse the octree and

evaluate transport vector T_d by accumulating clustered irradiance transport vectors at each node.

3.2. Single scattering

For highly translucent materials, multiple scattering components dominates, while single scattering also play an important role. Besides we face a challenge for single scattering: the integration path(the refracted outgoing direction) is unknown at precomputation time, and difficulty of decoupling rendering from precomputation because a general phase function depends on both incident and view directions. Therefore, we considered the solution proposed by [16] which is simple but faster: We arbitrarily substitute the view-dependent integration path with a path along the opposite of normal where outgoing radiance ray lies. By doing this, we have approximated the view-dependent integration path with a view independent path known at precomputation time (see Fig 1(b)). This approximation changes the observed incident path length s_i. Then, we approximate the phase function using a separable decomposition technique introduced by [8], which has been used for BRDF rendering. We can now handle phase functions in the same way as [15] and [11]. The final equations for single scattering are:

$$L_o^{(1)}(x_o,\vec{\omega}_o) = F_t(\eta,\vec{\omega}_o)\sum_k h_k(\vec{\omega}_o')\sum_j T_k^{(1)}(x_o,\vec{\omega}_j)L(\vec{\omega}_j)$$

$$T_k^{(1)}(x_o,\vec{\omega}_o) = \int_0^\infty \sigma_s T_p(x_p,\vec{\omega}_i)e^{-\sigma_t s}ds \quad (9)$$

$$T_p(x_p,\vec{\omega}_i) = g_k(\vec{\omega}_i')F_t(\eta,\vec{\omega}_i)e^{-\sigma_t s_i'}V(x_i,\vec{\omega}_i) \quad (10)$$

where $T_k^{(1)}$ is the k-th single scattering transport function, corresponding to the k-th phase function term. x_p is the sample point at distance x along the integration path, and T_p is the illumination transport function at x_p describing the amount of light transported to x_p from the environment but attenuated along the incident path. Evaluation of $T_k^{(1)}$ can employ Monte Carlo techniques to integrate T_p over the sample path. For importance sampling, we pick a random distance along the sample path by $s(\xi) = -\ln(1-\lambda\xi)/|\sigma_t|$, where $\xi \in [0,1]$ is a uniformly distributed random number, $|\sigma_t|$ is the luminance of σ_t, $\lambda = 1 - e^{-|\sigma_t| d_m}$ is the normalization term, and dm is the integration upper limit, which is the maximum distance along the path. In practice, since it takes many random samples to eliminate noise, we instead use a fixed number of N deterministic sample distances by taking $s = s(\xi_i)$, where $\xi_i = (i-0.5)/N$ and i = 1… N.

3.3. PRT

In general, there are two computing units on our experiment machine: CPU and GPU. In order to compute efficiently, our policy is to carry on illumination related computation on GPU and other on CPU. First, we discretized direct light on a $8\times32\times32$ cubemap. Irradiance and illumination transport functions are rendered into the card. For multiple scattering, we first evaluate irradiance transport functions for a set of evenly distributed surface sample points. At each point, we gather and store the visibility information, which will be applied to generate the transport function according to equation 8. We then download the data on the card to the CPU to perform the non-linear wavelet approximation. When irradiance sampling is completed, we apply the hierarchical integration to compute transport vectors T_d by the diffuse approximation.

We first compute transport matrix T in the raw lighting basis of $8\times16\times16$ cubemap, then perform a SVD on the tabulated data, and storing them as cubemap textures. Each term contains a pair of light map and view map. We apply the $K = 6$ lightmaps in precomputation as indicated by equation 10 and produce K transport functions per vertex. Finally, we download data to the CPU to perform the final step of non-linear wavelet approximation. GPU takes over most vector related computing work, which provides us a performance improvement.

3.4. Rendering

First, we perform a fast wavelet transform on sample of the environment map on $8 \times 32 \times 32$ cubemap. We use a 2D Haar transform here as well because Haar is an orthonormal basis. And we choose a subset of wavelet basis lights to use for the current frame for non-linear approximation. We have implemented the dot products of the light vector with

the transport vectors entirely on the CPU. Since, graphics hardware is not clearly general enough to support our algorithm efficiently.

The multiple scattering component are modulated for Fresnel transmittance to determine the final vertex colors. And we incorporate K-term phase function approximation into texture lookups of the refracted view direction. Finally a simple scene was set up: the model with direct illumination stands on a glazed plane. Specular reflection was added to the model so that it looks more realistic. We use a pre-convolved environment map to add a specular component to the final rendering.

4. Results and Discussion

We experiment our technique for several models. The precomputation time, precomputation size and rerendering speed are listed in Table 1.

For Utah teapot (see Fig 2). The main parameters we choose are σ_s = [0.71, 0.79, 1.00] and σ_α = [0.03, 0.38, 0.06]. The result are compared with the one generated by [5] under similar viewing and lighting conditions. Our approach gives similar effects with smooth speed 7.9 fps, comparable to 7.5 fps reported in [5]. The computation of the pre-computed integral with 300 light sources for all vertices takes about 52 minutes for the teapot model. Throughout our experiments, we use smaller cubemap resolution of $8 \times 32 \times 32$ and keep 128 (2%) wavelet coefficients. Subsurface scattering tends to blur texture and illumination, and then consequently soften the overall look of an object. Therefore compared with the BRDF, the low-frequency BSSRDF transport functions will be enough for computing.

For smurf model (see Fig 3), we applied a perlin style marble texture during the final rendering. In (a), smurf standing on a plane is rendered under back directional light. Note that the translucent feature around ears and fingers. In addition, reflection added reinforces realistic feeling. Form (b) to (c), smurf is rendered without and with consideration of all-frequency shadow. Note that soft shadow boundaries in (c). In (d), we choose lightmap illumination to rerender smurf and obtain an interesting scene. See that it is brighter while with shadows ignored.

5. References

[1] P. Blasi, B.L. Saec, and C. Schlick, "A rendering algorithm for discrete volume density objects", *Computer Graphics Forum 12-3,* 1993, pp. 201-210.

[2] J. Dorsey, A. Edelman, H.W. Jensen, J. Legakis, and H.K. Pedersen, "Modeling and rendering of weathered stone", *In Proc. of SIGGRAPH 1999,* 1999, pp. 225-234.

[3] P. Hanrahan and W. Krueger, "Reflection from layered surfaces due to subsurface scattering", *In Proc. of SIGGRAPH 1993,* 1993, pp. 165-174.

[4] X. Hao and A. Varshney, "Real-time rendering of translucent meshes", *ACM Trans. Graph. 23, 2,* 2004, pp. 120-142.

[5] H.W. Jensen and J. Buhler, "A rapid hierarchical rendering technique for translucent materials", *ACM Trans. Graph. 21, 3,* 2002, pp. 576-581.

[6] H.W. Jensen and P.H. Christensen, "Efficient simulation of light transport in sciences with participating media using photon maps", *In Proc. of SIGGRAPH 1998,* 1998, pp. 311-320.

[7] H.W. Jensen, S.R. Marschner, M. Levoy, and P. Hanrahan, "A practical model for subsurface light transport", *In Proc. of SIGGRAPH 2001,* 2001, pp. 511-518.

[8] J. Kautz and M.D. Mccool, "Interactive rendering with arbitrary brdfs using separable approximations", *In Proc. of the 10th Eurographics Rendering Workshop,* 1999, pp. 281-292.

[9] E.P. Lafortune and Y.D. Willems, "Rendering participating media with bidirectional path tracing", *In Proc. of the 7th Eurographics Rendering Workshop,* 1996, pp. 91-100.

[10] T. Mertens, J. Kautz, P. Bekaert, H.P. Seidelz, and F.V. Reeth, "Interactive rendering of translucent deformable objects", *In Proc. of the 14th Eurographics Symposium on Rendering,* 2003, pp. 130-140.

[11] R. Ng, R. Ramamoorthi, and P. Hanrahan, "All-frequency shadows using non-linear wavelet lighting approximation", *ACM Trans. Graph. 22, 3,* 2003, pp. 3760-381.

[12] P.Sloan, J. Kautz, and J. Snyder, "Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments", *In ACM Trans. Graph., vol. 21,* 2002, pp. 527-536.

[13] P. Sloan, J. Hall, J. Hart, and J. Snyder, "Clustered principal components for precomputed radiance transfer", *ACM Trans. Graph.,* 2003, pp. 382-391.

[14] J. Stam, "Multiple scattering as a diffusion process", *In Proc. of the 6th Eurographics Rendering Workshop,* 1995, pp. 41-50.

[15] R.Wang, J. Tran, and D. Luebke, "All-Frequency Relighting of Non-Diffuse Objects using Separable BRDF

Approximation", *In Proc. of the 15th Eurographics Symposium on Rendering,* 2004, pp. 345-354.

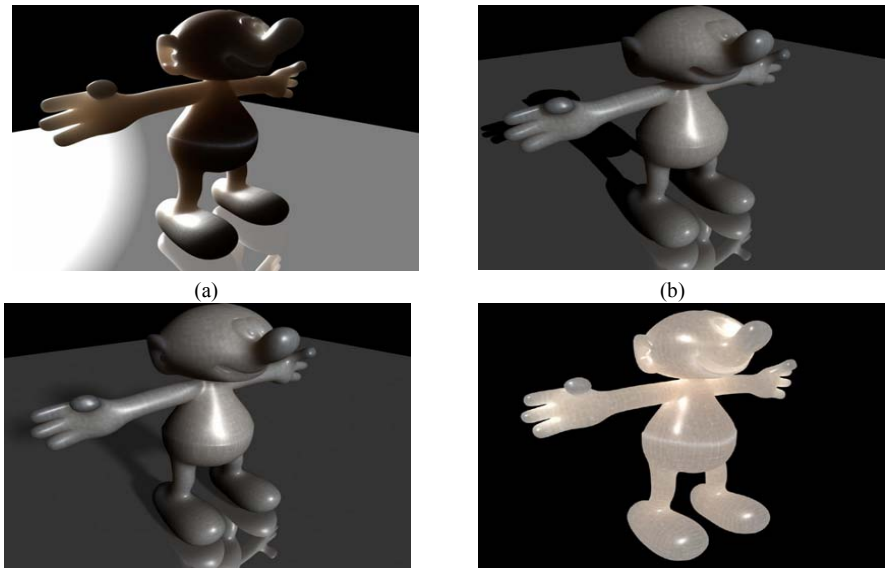
[16] R Wang, J. Tran, and D. Luebke, "All-Frequency Interactive Relighting of Translucent Objects with Single and Multiple Scattering", *In Proc. of SIGGRAPH 2005, ACM Trans. Graphics 24(3),* 2005.

Table 1: Rendering results by our technique

Model Name	Number of Vertices	Number of Triangles	Precomputation time	Precomputation size	Frame rate (fps)	
					with scattering	without scattering
Teapot	150,510	292,168	52 min	131MB	7.9	11.2
Smurf	372,543	310,321	121 min	243MB	4.1	6.3

(a) (b)

Fig 2 Result for Utah teapot model.
(a) A Utah teapot with scattering was rendered by our system. (b) is a image Image from paper [5].



(a) (b)

(c) (d)

Fig 3. A Perlin stylemarble texture is applied during the final rendering.
In (a), smurf standing on a plane is rendered under back directional light. Note that the translucent feature around ears and figures. In addition, reflection added reinforces realistic feeling. Form (b) to (c) smurf is rendered without and with consideration of all-frenquncy shadow. Note that soft shadow boundaries in (c). In (d), we choose different lightmap to rerender smurf and obtain an interesting scene. See that it is brighter while with shadows ignored.