Visual Comput (2007) 23: 59–69
DOI 10.1007/s00371-006-0082-7

ORIGINAL ARTICLE

Shih-Ling Keng
Wang-Yeh Lee
Jung-Hong Chuang

An efficient caching-based rendering of translucent materials

Published online: 31 October 2006

S.-L. Keng · W.-Y. Lee (✉) · J.-H. Chuang
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China
{kensl, wylee, jhchuang}@csie.nctu.edu.tw

Abstract This paper presents an efficient caching-based rendering technique for translucent materials. The proposed caching scheme, inspired by the irradiance caching method, is integrated into a hierarchical rendering technique for translucent materials. We propose a split-disk model to determine the cache distribution and derive the subsurface illuminance gradient used for interpolation by reformulating the equation of dipole diffusion approximation as a 3D convolution process. Our experiments show that only a few caches are required to interpolate the entire image, while the visual difference is negligible. The speedup could be achieved up to one order of magnitude.

Keywords Subsurface scattering · BSSRDF · Light transport · Diffusion theory · Realistic image synthesis

1 Introduction

Accurately modeling the behavior of light to produce realistic images is a great challenge in computer graphics. Over the years, many illumination models have been developed for realistic image synthesis, trying to describe the scattering of light from materials. Most of them focused on developing models using BRDFs (bidirectional reflectance distribution functions), which assume that light enters and leaves a material at the same point on the surface. In some cases like metals, this assumption is valid and results in convincing visual appearances. But when accounting for translucent materials that exhibit significant light transport below the surface, BRDF is not sufficient. Light hitting a translucent material does not merely bounce from surfaces. Instead, light beams penetrate below the surface, scatter inside the material, and leave the object at a different point on the surface. This phenomenon is known as subsurface scattering.

Traditionally, subsurface scattering has been approximated as Lambertian diffuse reflection that makes the final images look hard and distinctly computer-generated. In computer graphics, the first model dealing with subsurface scattering was proposed by Hanrahan and Krueger [6]. They proposed an analytic expression for single scattering in a homogeneous, uniformly lit slab. Dorsey et al. [4] later used photon mapping to simulate full subsurface scattering for the rendering of weathered stones. Pharr and Hanrahan [13] proposed the idea of nonlinear scattering equations and demonstrated how the scattering equation could be used to simulate subsurface scattering more efficiently than does a traditional Monte Carlo ray tracing. For highly scattering material, Stam [14] first introduced the diffusion theory to computer graphics and proposed a multigrid method to solve a diffusion equation approximation.

A major breakthrough was recently proposed by Jensen et al. [10] with an analytic BSSRDF (bidirectional surface scattering reflection distribution function) model for

subsurface scattering. Based on this model, Jensen and Buhler [9] dramatically reduced computation time from several minutes to a few seconds. The BSSRDF model was then adopted for interactive rendering with mesh-based objects using various rendering algorithms [1, 3, 7, 8, 11, 12].

Although recent researches have improved the speed of rendering translucent materials to some extent, none of them could be easily integrated into existing renderers. They require either complex rendering algorithms or some specific data structures. In other words, most of them are isolated systems for the pure purpose of experiments. The algorithms used in these rendering systems are not suitable for the movie industry, where some specific renderers must be used and objects are often not mesh-based.

To devise an efficient rendering technique for the film industry, we investigate the effect of subsurface scattering and find that it has distinguishing characteristics just as the effect of indirect lighting in the global illumination. They both tend to change slowly and require a lot of sample points to compute. This inspires us to use the classic irradiance caching technique introduced by Ward et al. [17] as a basis, and to extend it for calculating the subsurface illuminance. Irradiance caching was originally designed for accelerating the computation of indirect illumination in a Monte Carlo ray tracer [16, 17]. It is a method for caching and reusing irradiance values (via interpolation) on Lambertian surfaces. The irradiance caching technique was later extended to accelerate the computation of ambient occlusion in production [2]. In this paper, we show that it is feasible to extend the irradiance caching technique to accelerate the computation of the subsurface illuminance as well.

2 The dipole diffusion approximation

The dipole diffusion approximation, which approximates the volumetric source distribution using a dipole (i.e., two point sources), was originally developed in the medical physics community. Farrell et al. [5] used a single dipole to represent the incident source distribution for the non-invasive determination of tissue optical properties in vivo. Jensen et al. [10] then introduced the dipole diffusion approximation to the computer graphics community for modeling the subsurface light transport.

The dipole diffusion approximation consists of positioning two point sources near the surface to approximate an incoming light. One point source, the positive real light source, is located at a distance z_r beneath the surface, and the other one, the negative virtual light source, is located above the surface at a distance z_v. By using dipole diffusion approximation to solve the diffusion equation, we can get the following expression for the radiant exitance M_{p_i} at surface location p_o due to incident flux $\Phi(p_i)$ at p_i (see [9] for the details of derivation):

$$dM_{p_i}(p_o) = d\Phi(p_i)\frac{\alpha'}{4\pi}\left[z_r(1+\sigma s_r)\frac{e^{-\sigma s_r}}{s_r^3} + z_v(1+\sigma s_v)\frac{e^{-\sigma s_v}}{s_v^3}\right], \quad (1)$$

where $\alpha' = \sigma_s'/\sigma_t'$ is the *reduced albedo*; $\sigma_s' = \sigma_s(1-g)$ is the *reduced scattering coefficient* and g is the *mean cosine of the scattering angle*; $\sigma_t' = \sigma_s' + \sigma_a$ is the *reduced extinction coefficient*; σ_a is the *absorption coefficient*; $\sigma = \sqrt{3\sigma_a\sigma_t'}$ is the *effective transport coefficient*; $s_r = \sqrt{r^2 + z_r^2}$ is the distance from p_o to the positive real light source; $s_v = \sqrt{r^2 + z_v^2}$ is the distance from p_o to the negative virtual light source; $r = \|p_o - p_i\|$ is the distance from p_o to p_i; and $z_r = l_u$ and $z_v = l_u(1 + (4/3)A)$ are the distances from the dipole sources to the surface. The mean-free path $l_u = \frac{1}{\sigma_t'}$ is the average distance at which the light is scattered. Finally, the boundary condition for mismatched interfaces is taken into account by the A term that is computed as $A = (1 + F_{dr})/(1 - F_{dr})$, where the diffuse Fresnel term F_{dr} is rationally approximated from the relative index of refraction η by [10]:

$$F_{dr}(\eta) = -\frac{1.440}{\eta^2} + \frac{0.710}{\eta} + 0.668 + 0.0636\eta.$$

By using Eq. 1, the subsurface illuminance, which is defined as the light flux per unit area arriving at an inner surface point within materials via subsurface scattering from the nearby surfaces, then could be computed as:

$$\begin{aligned} S(p_o) &= \int_{p_i \in A} d\Phi(p_i)\frac{\alpha'}{4\pi}\left[z_r(1+\sigma s_r)\frac{e^{-\sigma s_r}}{s_r^3} + z_v(1+\sigma s_v)\frac{e^{-\sigma s_v}}{s_v^3}\right] \\ &= \int_{p_i \in A} \hat{E}(p_i)\frac{\alpha'}{4\pi}\left[z_r(1+\sigma s_r)\frac{e^{-\sigma s_r}}{s_r^3} + z_v(1+\sigma s_v)\frac{e^{-\sigma s_v}}{s_v^3}\right] dp_i \\ &= \int_{p_i \in A} \hat{E}(p_i) R_d(p_i, p_o) dp_i, \end{aligned} \quad (2)$$

where $\hat{E}(p_i) = F_{dt}(\eta)E(p_i)$ and $E(p_i)$ is the irradiance at point p_i within the material. The diffuse Fresnel transmittance $F_{dt}(\eta)$ is defined as $F_{dt}(\eta) = 1 - F_{dr}(\eta)$, and $R_d(p_i, p_o)$ is the diffuse BSSRDF defined as the ratio of radiant exitance to incident flux [10].

Finally, since the diffusion approximation already includes a diffuse Fresnel transmittance, the diffuse radiance

L is computed as:

$$L(p_o, \omega) = \frac{F_t(1/\eta, \omega)}{F_{dt}(\eta)} \frac{S(p_o)}{\pi},$$

where F_t is the Fresnel transmittance.

Alternatively, we could omit the Fresnel terms and assume a diffuse radiance:

$$L(p_o, \omega) = \frac{S(p_o)}{\pi}. \tag{3}$$

3 A caching technique for rendering translucent materials

3.1 Dipole diffusion approximation as a convolution process

Recall that the subsurface illuminance function (Eq. 2) is

$$S(p_o) = \int_{p_i \in A} \hat{E}(p_i) R_d(p_i, p_o) dp_i, \tag{4}$$

where $\hat{E}$ is the transmitted irradiance function over the surface and R_d is the diffuse BSSRDF:

$$R_d(p_i, p_o) = \frac{\alpha'}{4\pi} \left[z_r(1+\sigma s_r) \frac{e^{-\sigma s_r}}{s_r^3} + z_v(1+\sigma s_v) \frac{e^{-\sigma s_v}}{s_v^3} \right].$$

In the assumption of semiinfinite plane-parallel medium, R_d becomes a function of only the distance between p_i and p_o. By replacing parameter p_i and p_o with the offset $\|p_i - p_o\|$ and expressing the vector parameter in terms of scalar values, we can rewrite R_d as follows:

$$\begin{aligned} R_d(p_i, p_o) &= R_d(p_{i,x}, p_{i,y}, p_{i,z}, p_{o,x}, p_{o,y}, p_{o,z}) \\ &= R_s(p_{i,x} - p_{o,x}, p_{i,y} - p_{o,y}, p_{i,z} - p_{o,z}), \end{aligned}$$

where

$$\begin{aligned} R_s(x, y, z) = \frac{\alpha'}{4\pi} \Bigg\{ & z_r \left(1 + \sigma \sqrt{z_r^2 + x^2 + y^2 + z^2}\right) \\ & \times \frac{e^{-\sigma \sqrt{z_r^2 + x^2 + y^2 + z^2}}}{(\sqrt{z_r^2 + x^2 + y^2 + z^2})^3} \\ & + z_v \left(1 + \sigma \sqrt{z_v^2 + x^2 + y^2 + z^2}\right) \\ & \times \frac{e^{-\sigma \sqrt{z_v^2 + x^2 + y^2 + z^2}}}{(\sqrt{z_v^2 + x^2 + y^2 + z^2})^3} \Bigg\}. \end{aligned} \tag{5}$$

Note that R_s is a three-dimensional radial function with its value decaying exponentially with the distance.

The original equation (Eq. 4) integrates p_i over the surface A. It seems that the integration is a two-dimensional process. But actually the integration is reformed in three-dimensional space because p_i is a three-dimensional point. So we change the integral domain and rewrite Eq. 4 in a form of three dimensions:

$$\begin{aligned} & S(p_x, p_y, p_z) \\ & \quad = \iiint_{XYZ} \hat{E}(x, y, z) R_d(x, y, z, p_x, p_y, p_z) dx dy dz. \end{aligned}$$

Then replacing R_d, we get:

$$\begin{aligned} & S(p_x, p_y, p_z) \\ & \quad = \iiint_{XYZ} \hat{E}(x, y, z) R_s(x - p_x, y - p_y, z - p_z) dx dy dz. \end{aligned}$$

Because $R_s(x - p_x, y - p_y, z - p_z)$ is a symmetric function, we can change the sign of the parameter:

$$\begin{aligned} & S(p_x, p_y, p_z) \\ & \quad = \iiint_{XYZ} \hat{E}(x, y, z) R_s(p_x - x, p_y - x, p_z - z) dx dy dz. \end{aligned} \tag{6}$$

Obviously, the resultant equation is in the form of a three-dimensional convolution of two functions. Finally we arrive at:

$$S = \hat{E} \otimes R_s,$$

where $\otimes$ denotes the operator of convolution.

3.2 Derivation of the subsurface illuminance gradient

Given the subsurface illuminance function (Eq. 4), it is not clear how to calculate the gradient of the subsurface illuminance. With the reformulated convolution form in Eq. 6, the gradient could be derived straightforwardly. Recall that the derivative of a convolution function is

$$\frac{d}{dx}(f \otimes g) = \frac{df}{dx} \otimes g = f \otimes \frac{dg}{dx}.$$

The gradient of the subsurface illuminance is then derived as follows:

$$\begin{aligned} \nabla S &= \left(\frac{\partial S}{\partial x}, \frac{\partial S}{\partial y}, \frac{\partial S}{\partial z} \right) \\ &= \left(\frac{\partial}{\partial x}(\hat{E} \otimes R_s), \frac{\partial}{\partial y}(\hat{E} \otimes R_s), \frac{\partial}{\partial z}(\hat{E} \otimes R_s) \right) \\ &= \left(\frac{\partial \hat{E}}{\partial x} \otimes R_s, \frac{\partial \hat{E}}{\partial y} \otimes R_s, \frac{\partial \hat{E}}{\partial z} \otimes R_s \right) \end{aligned} \tag{7}$$

or

$$= \left(\hat{E} \otimes \frac{\partial R_s}{\partial x}, \hat{E} \otimes \frac{\partial R_s}{\partial y}, \hat{E} \otimes \frac{\partial R_s}{\partial z} \right). \tag{8}$$

As shown in Eqs. 7 and 8, once we get either $\nabla \hat{E}$ or ∇R_s, we could calculate the gradient of subsurface illuminance. Unfortunately, since $\nabla \hat{E}$ does not have an analytic form, it is infeasible to calculate $\nabla \hat{E}$ in practice. Therefore, in our implementation, we choose Eq. 8 to calculate the gradient of subsurface illuminance. To better clarify the following formulation, we change notation, replacing z_r and z_v with h_r and h_v, respectively. Note that

$$\frac{\partial R_s(x, y, z)}{\partial x} = \frac{\alpha'}{4\pi} x \left[h_r \frac{-e^{-\sigma s_r}}{s_r^4} \left(\sigma^2 s_r + 3\sigma + \frac{3}{s_r} \right) + h_v \frac{-e^{-\sigma s_v}}{s_v^4} \left(\sigma^2 s_v + 3\sigma + \frac{3}{s_v} \right) \right],$$

$$\frac{\partial R_s(x, y, z)}{\partial y} = \frac{\alpha'}{4\pi} y \left[h_r \frac{-e^{-\sigma s_r}}{s_r^4} \left(\sigma^2 s_r + 3\sigma + \frac{3}{s_r} \right) + h_v \frac{-e^{-\sigma s_v}}{s_v^4} \left(\sigma^2 s_v + 3\sigma + \frac{3}{s_v} \right) \right],$$

$$\frac{\partial R_s(x, y, z)}{\partial z} = \frac{\alpha'}{4\pi} z \left[h_r \frac{-e^{-\sigma s_r}}{s_r^4} \left(\sigma^2 s_r + 3\sigma + \frac{3}{s_r} \right) + h_v \frac{-e^{-\sigma s_v}}{s_v^4} \left(\sigma^2 s_v + 3\sigma + \frac{3}{s_v} \right) \right],$$

$$s_r = \sqrt{x^2 + y^2 + z^2 + h_r^2},$$

and

$$s_v = \sqrt{x^2 + y^2 + z^2 + h_v^2}.$$

Thus the problem is reduced to the evaluation of the integrals of convolution. In the case of partial derivative of S with respect to x, this results in

$$\frac{\partial S(p_x, p_y, p_z)}{\partial x} = \iiint_{XYZ} \hat{E}(x, y, z) \frac{\partial R_s(p_x - x, p_y - y, p_z - z)}{\partial x} dx dy dz. \tag{9}$$

Although the integrals still do not have an analytic solution, by exploiting the properties of ∇R (see Fig. 1), we could use some integration techniques such as Monte Carlo and quadrature methods to get a good approximation.

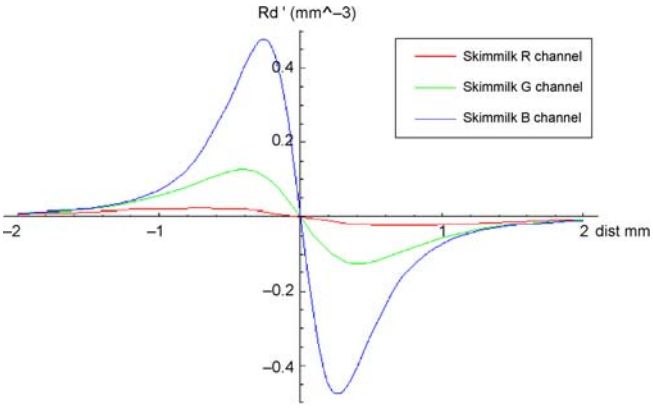


Fig. 1. The graph of $\partial R_s / \partial x$

3.3 Applying the gradient to interpolation

Once we calculate the gradient of subsurface illuminance, we could use the gradient to interpolate the subsurface illuminance more accurately. We use the same weighted average as proposed in [16] to interpolate the subsurface illuminance value:

$$S(p) = \frac{\sum_{k \in C} w_k(p)[S_k + (p - p_k) \cdot \nabla S_k]}{\sum_{k \in C} w_k(p)}, \tag{10}$$

where p is the position of the point to be computed; p_k is the position of cache k; $w_k(p)$ is the weight of cache k with respect to p; C is the set of valid caches such that $\{cache\ k : w_k(p) > 1/a\}$; S_k is the computed subsurface illuminance of cache k; ∇S_k is the computed gradient of subsurface illuminance of cache k; and a is a user-specified error bound.

The next problem is how to determine the spacing of samples, i.e., how to determine the weight of each sample or how to estimate the error of each sample. The simplest and theoretically most accurate solution is directly using the inner product of the offset $(p - p_k)$ and ∇S_k derived in last section as our error estimate of cache k (assuming that the error due to interpolation is proportional to the estimated directional change rate of subsurface illuminance from p_k to p), i.e.,

$$\begin{aligned} \varepsilon \propto S'_p &= \lim_{h \to 0} \frac{\Delta S}{h} \\ &= \Delta p_x \frac{\partial S}{\partial x} + \Delta p_y \frac{\partial S}{\partial y} + \Delta p_z \frac{\partial S}{\partial z} \\ &= \Delta p \cdot \nabla S, \end{aligned}$$

where S'_p is the directional derivative of S in direction p.

Unfortunately, this will lead to bias in the calculation. Since the gradient is a very local property, areas that just happen to have small subsurface illuminance gradients would be sampled at low density, even though there

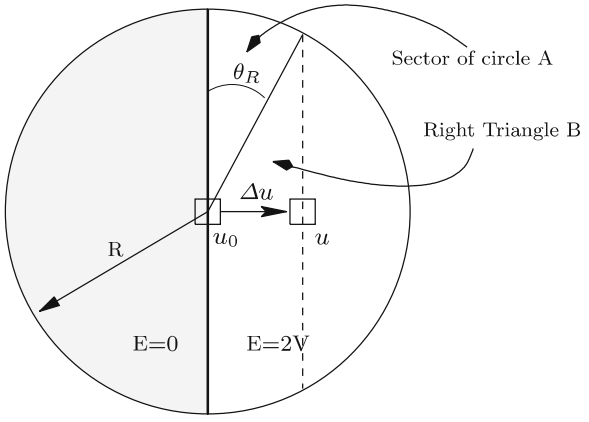


Fig. 2. The split-disk model. A surface element is located at the center of a half-dark disk

could still be sudden changes in the subsurface illuminance value due to nearby surfaces. A possible solution is to use some approximation models to capture the largest expected gradient in determining the sample density so that we do not miss anything relevant.

To estimate the largest expected gradient, we introduce a split-disk model analogous to the split-sphere model proposed by Ward et al. [17]. The split-disk model, based on the assumption that the geometry is locally flat, relates the subsurface illuminance gradient to the variance V of the irradiance values within nearby surfaces. It assumes that a surface element is located at the center of a disk that approximates nearby surfaces (see Fig. 2). The radius of the disk, R, is heuristically determined according to the material scattering property. Half of the disk is totally bright with constant irradiance E and the other half is totally dark with constant irradiance of zero. Because the variance of the irradiance values within the disk is V, we can conclude that $E = 2V$. The split disk has the largest expected gradient possible for surfaces with variance V.

An approximate bound to the change rate of subsurface illuminance in the split disk, ε, is just given by the first order Taylor expansion of the function S of one variable:

$$\varepsilon(u) \leq \left| (u - u_o) \frac{\partial S}{\partial u} \right|,$$

where u_o is the center of the disk and u is some other point on the disk. Note u_o and u are both one-dimensional values because we only care about the distance between two points. To derive

$$\frac{\partial S}{\partial u} = \lim_{h \to 0} \frac{S(u+h) - S(u)}{h} = \lim_{h \to 0} \frac{\Delta S}{h},$$

we firstly consider a surface element moving from u_o to u, the change of S could be computed as twice an integral over sector of circle A plus an integral over right triangle B (see Fig. 2):

$$\Delta S = 2(\Delta S_A + \Delta S_B),$$

where

$$\Delta S_A = \int_0^{\theta_R} \int_0^{R} E(r) R_d(r) r dr d\theta$$

and

$$\Delta S_B = \frac{1}{2} \int_0^{\sqrt{R^2 - \Delta u^2}} \int_0^{\Delta u} E(x, y) R_d(\sqrt{x^2 + y^2}) dx dy.$$

Unfortunately, we cannot find an analytic solution of the integral describing subsurface scattering over right triangle ΔS_B. Inspired by [12], where Mertens et al. derive a semianalytic integration method to solve the integral describing subsurface scattering over an arbitrary triangle, we can approximate ΔS by an integral over four sectors of a circle:

$$\Delta S \approx 4 \Delta S_A.$$

The derivation of ΔS is as follows:

$$\Delta S \approx 2V \frac{\alpha'}{\pi} \arcsin\left(\frac{\Delta u}{R}\right) \left(e^{-\sigma z_r} - \frac{z_r}{R_r} e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v} e^{-\sigma R_v} \right),$$

where $R_r = \sqrt{R^2 + z_r^2}$ and $R_v = \sqrt{R^2 + z_v^2}$. Since $\frac{\partial S}{\partial u} = \lim_{h \to 0}(\frac{\Delta S}{h})$, we get

$$\frac{\partial S}{\partial u} \approx V \frac{2\alpha'}{\pi R} \left(e^{-\sigma z_r} - \frac{z_r}{R_r} e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v} e^{-\sigma R_v} \right).$$

And the error estimate $\varepsilon_k(p)$ of cache k with respect to p can be computed as

$$\varepsilon_k(p) = |p - p_k| V_k \frac{2\alpha'}{\pi R} \times \left(e^{-\sigma z_r} - \frac{z_r}{R_r} e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v} e^{-\sigma R_v} \right),$$

where V_k is the variance of irradiance values within the disk of cache k. As in [17], the inverse of the error estimate

$$w_k(p) = \frac{1}{\varepsilon_k(p)} = \frac{1}{|p - p_k| V_k \frac{2\alpha'}{\pi R} (e^{-\sigma z_r} - \frac{z_r}{R_r} e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v} e^{-\sigma R_v})} \quad (11)$$

is then used as our weight. Substituting Eq. 11 into Eq. 10, the subsurface illuminance of some point of interest then can be computed by interpolating nearby caches.

3.4 A three-pass technique for rendering translucent materials

To integrate our model into Jensen's hierarchical evaluation method [9] , we use a three-pass approach, in which the first pass consists of computing the irradiance at selected points on the surface, the second pass generates all the necessary cache samples, whose values including subsurface illuminance, gradient of subsurface illuminance, and variance of irradiance over nearby surface, are computed by using the precomputed irradiance values, and the last pass reuses the caches to produce the final image via interpolation.

Note that the second pass in our three-pass approach only generates caches. It does not use caches to interpolate any value. The interpolation using caches is done in the third pass.

Pass 1: Sampling the irradiance

To solve Eq. 4, firstly, we need to sample the irradiance function $E(x)$. There are a number of methods to generate sampling positions on the surface. In [9], Turk's point repulsion algorithm [15] is used to obtain a uniform sampling of points on a polygon mesh. However, a uniform sampling seems irrelevant in their hierarchical approach as each sample point is weighted by the area associated with it. Instead of using the Turk's point repulsion algorithm, which is hard to implement, we use a very simple method to obtain the sampling positions on the polygon mesh. We directly use the centroid of each face as our sample point and assign the area of the face as the area associated with this sample point. If a model is too coarse and results in low-frequency noise in the final image, we subdivide the model until the noise disappears.

For each sample point, we store the position, the area associated with the point, and a computed irradiance estimate. Since we focus on a caching technique in this paper, we do not use any rendering technique that accounts for global illumination (such as photon mapping and distributed ray racing) to compute the irradiance. We simply sum up irradiance contributions from each light source for evaluating direct illumination on each sample point.

As stated in [9], these irradiance samples should be stored in a hierarchical structure so that by clustering distant samples, we can exploit the exponentially shaped falloff property of R_d. Here we choose an octree as proposed by Jensen et al. [10] in our implementation. Each octree node contains the average position, the average irradiance, and the total areas of its subnodes.

Pass 2: Generating necessary caches

Before we state when and where all the necessary caches should be generated, we present how to compute the values stored in each cache. The values we stored in each cache are subsurface illuminance S, gradient of subsurface illuminance ∇S, and variance V of irradiance over nearby surfaces. All these values are computed by using the precomputed irradiance values (distributed in Pass 1) stored in a hierarchical structure.

The subsurface illuminance S is computed using the rapid hierarchical integration technique proposed by [9]. As for evaluating the gradient of subsurface illuminance ∇S, ideally, we should devise some integration technique exploiting the properties of ∇R_s (Fig. 1) to solve Eq. 9; and the integration technique has to be evaluated very fast, or the cost for computing the gradient will cancel out the gain from the interpolation. Fortunately, directly using the hierarchical integration technique proposed by [9] yields reasonable results and the variance V can be evaluated at the same time.

To determine when and where all the necessary caches should be generated, we firstly use ray casting to find a set of visible points X. For each point x_i in X, we check if there is a previously computed cache at a nearby surface that could be used for interpolation, i.e., any cache k with $w_k(x_i) > 1/a$. If any, we leave x_i to next pass; otherwise, we generate a new cache at point x_i, evaluate $S(x_i)$, $\nabla S(x_i)$, and V associated with the cache. Checking each point x_i in X in a different order such as bottom-up, top-down, and random scan-line order can change the distribution of caches; but the visual difference of interpolated images using these resulting cache distributions would be negligible. Here we choose the bottom-up scan-line order in our implementation.

As stated in [17], each previously computed cache is only valid for interpolation in some finite space. A hierarchical structure is required for efficiently searching nearby valid caches. Here we use the same data structure as proposed by [17], an octree, to store the caches.

Pass 3: Reusing caches to interpolate the image

After we generate all the necessary caches, we use Eq. 10 to calculate the subsurface illuminance of each point x_i in X (which is a set of visible points computed by ray casting) via interpolation. Finally, the radiance of each point x_i is obtained using Eq. 3.

3.5 Discussion

In Ward's original paper [17], though the split-sphere model is only a crude estimate of the gradient magnitude, generating cache and reusing cache were proposed to be done in a single pass. This results in some rather disturbing artifacts due to inaccurate interpolation and extrapola-

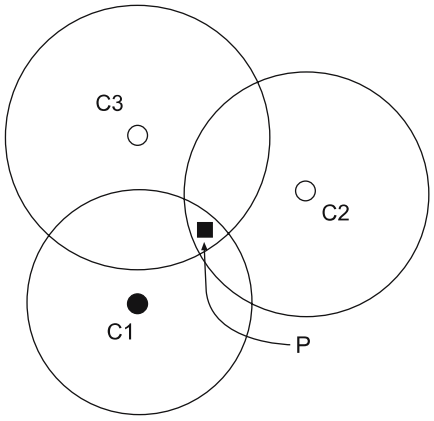


Fig. 3. Inaccurate interpolation due to a single-pass scan-line rendering

tion. Figure 3 shows an example of inaccurate interpolation.

In single-pass scan-line rendering (assuming in bottom-up scan-line order), the four points P, C_1, C_2, and C_3 are examined in the order of C_1, P, C_2, and C_3. Assuming that we generate C_1, C_2, and C_3 as caches; and P is within their valid domains, we can find that when we examine point P, cache C_2 and C_3 have not yet been generated, thus using only one cache C1 to interpolate point P. If we use two pass, i.e., the first pass generates the cache C_1, C_2, and C_3, then the second pass interpolates P using these three caches. That way would result in more accurate interpolation.

To solve this problem, Ward then proposed irradiance gradient [16], which computes actual irradiance gradients, not just a directionless upper bound, to make the interpolation and extrapolation significantly more accurate, thus avoiding separating generating cache and reusing cache into two passes. However, in our translucent caching technique, although we have successfully computed the gradient of subsurface illuminance, we still need a two-pass calculation to get an acceptable image. This is mainly due to the fact that subsurface illuminance gradient is much greater than the irradiance gradient. Figure 4 compares images rendered in one-pass and two-pass approaches.

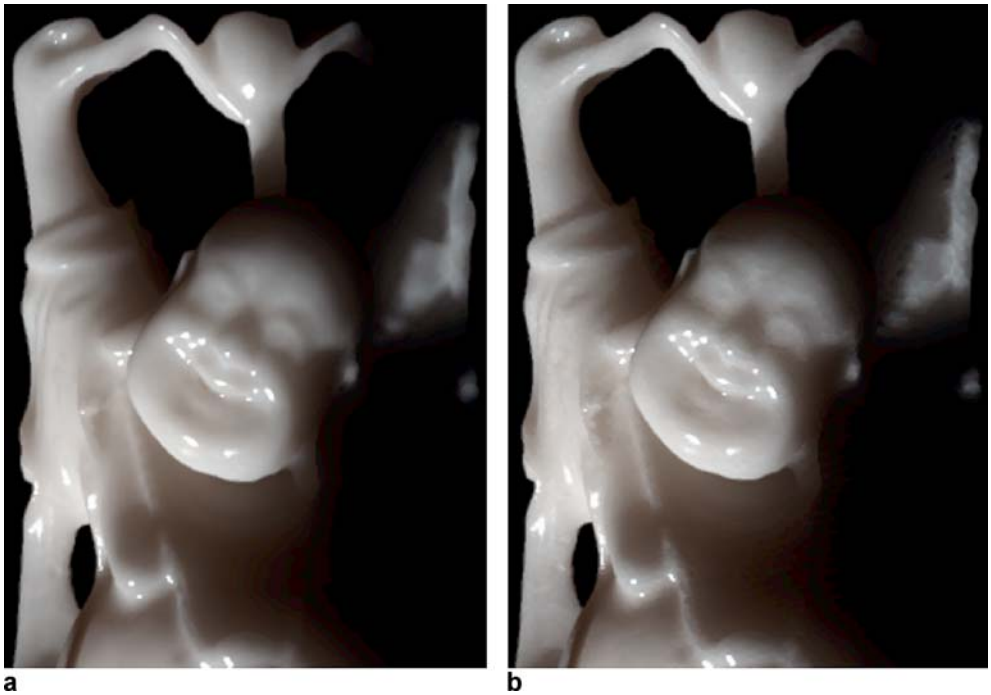
Instead of the split-disk model, we could use other approaches to determine the cache distribution. One alternative is applying a filter to ∇R, i.e., convolving ∇R with some filter function G (e.g., Gaussian function):

$$F_x = \frac{\partial R_s(x, y, z)}{\partial x} \otimes G,$$
$$F_y = \frac{\partial R_s(x, y, z)}{\partial y} \otimes G,$$
$$F_z = \frac{\partial R_s(x, y, z)}{\partial z} \otimes G.$$

Then by using the resultant functions, we can derive the average subsurface illuminance gradient:

$$\nabla S^* = (\hat{E} \otimes F_x, \hat{E} \otimes F_y, \hat{E} \otimes F_z).$$

This approach has advantages of capturing the average subsurface illuminance gradient more accurately and com-



a **b**

Fig. 4. **a** Two-pass approach. **b** One-pass approach

puting by using the same hierarchical integration technique for S and ∇S. Unfortunately, we did not find any filter function that could produce analytic solutions of function F. Thus we leave it as our future work.

4 Results

In this section we present several results from our implementation of the rendering technique. All the images are rendered by a Monte Carlo ray tracer at the resolution of 1024×1024 pixels. Our timings are recorded on a PC with an AMD Athlon XP 1800+ (1.53 GHz) processor and 512 MB main memory.

To validate our algorithm, we have implemented Jensen's hierarchical rendering technique [9] and compared the images generated by Jensen's hierarchical rendering technique with ours using four different models. Figures 5 and 6 demonstrate the visual comparisons of the rendered images of Dragon with material Skimmilk and Buddha with material Marble [10]. Our approach gives almost the same visual appearance while achieving about one order of magnitude speedup.

Table 1 illustrates the performance and timing statistics of our approach with different models. The first row (No. irr. samples) is the number of samples for sampling irradiance; second row (No. caches) is the number of total caches; third row (No. hit pixels) is the number of total hit pixels; Ratio is the ratio of number of caches to number hit pixels; RMS is the root-mean-square error with respect to the averaged RGB value of each pixel; T1 and T2 are the rendering times used in Jensen's approach and ours, respectively. The time for sampling the irradiance and computing the specular term is not taken into account. Note the ratio of total caches to total hit pixels is about 2%. Almost 98% pixels could be calculated via interpolation. While we only use such small amounts of caches, we still get very good visual appearances with RMS smaller than 0.01. The speedup ratio is dependent on the average cost for computing subsurface illuminance of each cache. The more the computation of subsurface illuminance costs (e.g., for better image quality), the higher speedup we get. Typically, it varies from 5 to 15.

Figures 8 and 9 shows the distribution of the caches, the visualization of the subsurface illuminance gradient, and the visualization of the variance V of irradiance with model Dragon and model Igea. The world coordinate of the gradient is mapped to the RGB channel in the image. The brightness of a pixel corresponds to the magnitude of the gradient. Figure 7a shows examples of Teapot with material Marble and Fig. 7b shows Igea with material Skin1 [10] generated by our proposed rendering technique.

5 Conclusion

We present an efficient caching technique for rendering translucent materials. Our approach is efficient for pro-

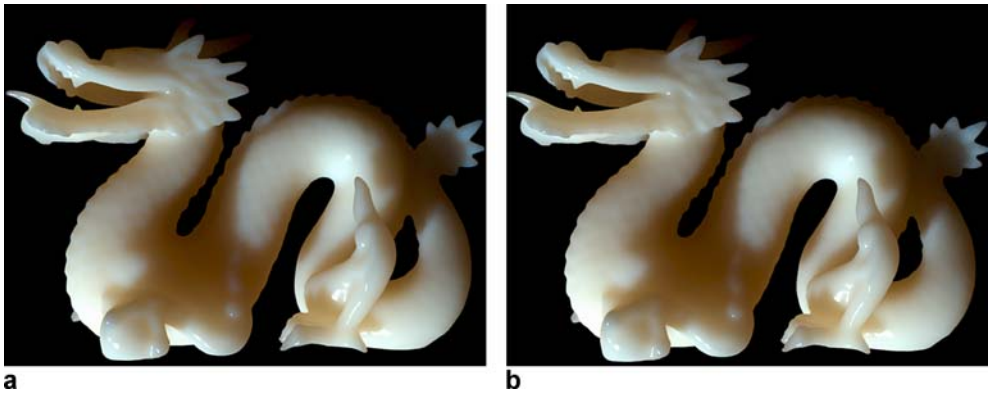


Fig. 5a,b. Visual comparison of model Dragon with material Skimmilk. **a** Image with Jensen's hierachical rendering technique. **b** Image with the proposed method

Table 1. Overview of performance with different models

Model	Buddha	Dragon	Igea	Teapot
No. irr. samples	293 232	202 520	268 686	261 632
No. caches	7698	9454	11 654	7646
No. hit pixels	342 601	446 151	491 830	315 883
Ratio	2.25%	2.12%	2.37%	2.42%
RMS	0.0068	0.0061	0.0046	0.0049
T1 (sec.)	25.80	30.86	35.80	16.75
T2 (sec.)	2.34	2.70	2.91	1.88
Speedup	11.03	11.43	12.30	8.91

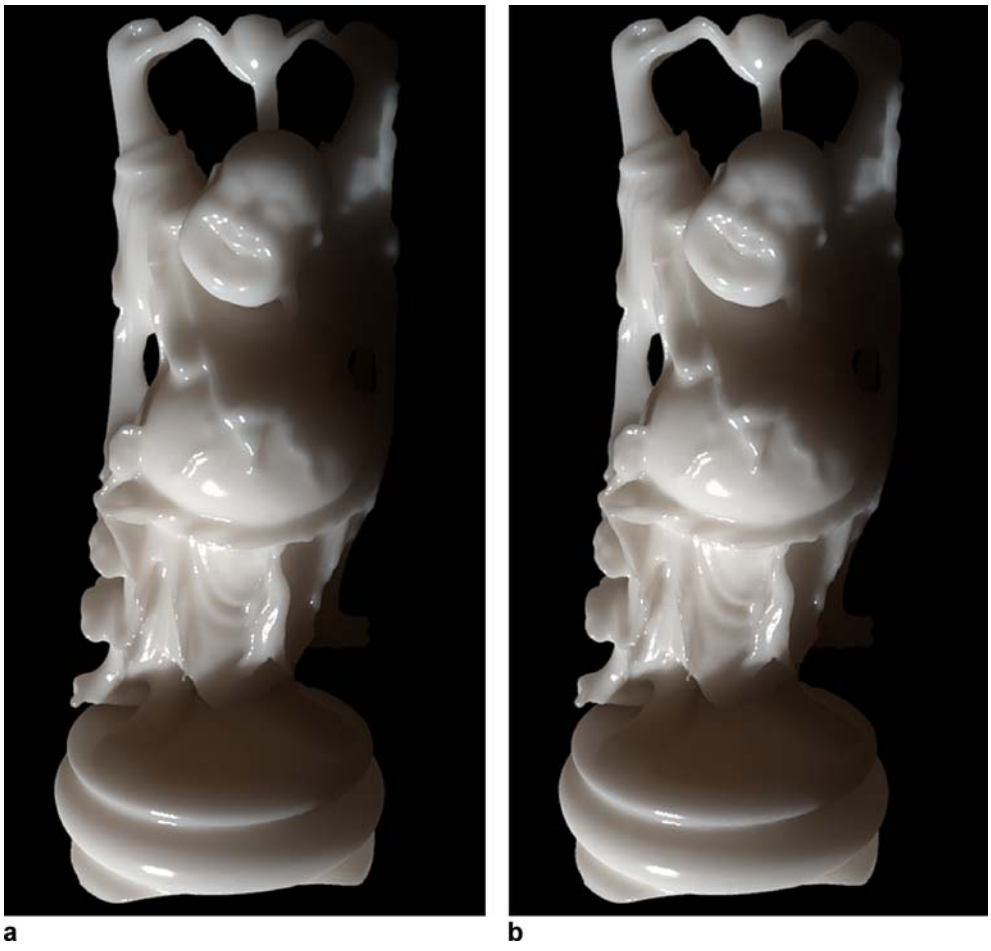


Fig. 6a,b. Visual comparison of model Buddha with material Marble. **a** Image with Jensen's hierachical rendering technique. **b** Image with the proposed method

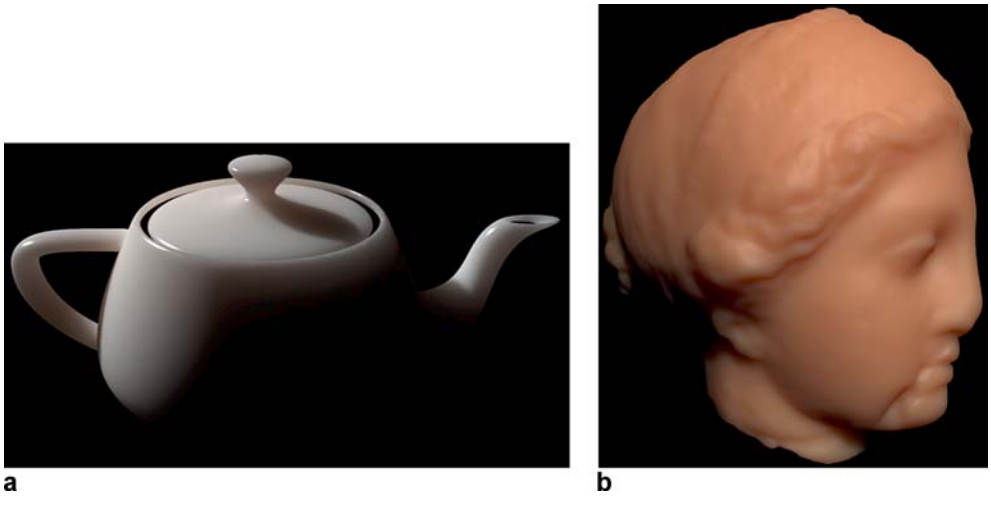


Fig. 7. a Teapot with material Marble. **b** Igea with material Skin1

ducing high-quality images with high resolution and is particularly useful in animations. It also integrates seamlessly with Monte Carlo ray tracing, scan-line rendering, and global illumination methods. Our results demonstrate that speedup could be achieved up to one order of magnitude compared to the hierarchical rendering technique proposed by Jensen and Buhler [9] with negligible visual difference in the final images. The success of our approach is mainly due to the caching technique using the gradient of subsurface illuminance.

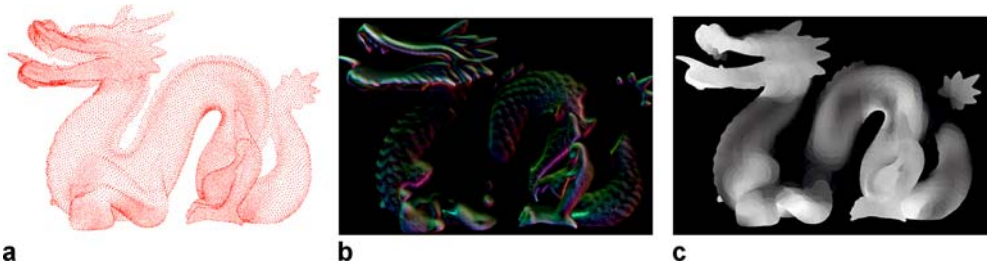


Fig. 8a–c. Visualization of cache distribution, subsurface illuminance gradient, and variance of irradiance with model Dragon. **a** Cache distribution. **b** Subsurface illumination gradient. **c** Variance of irradiance

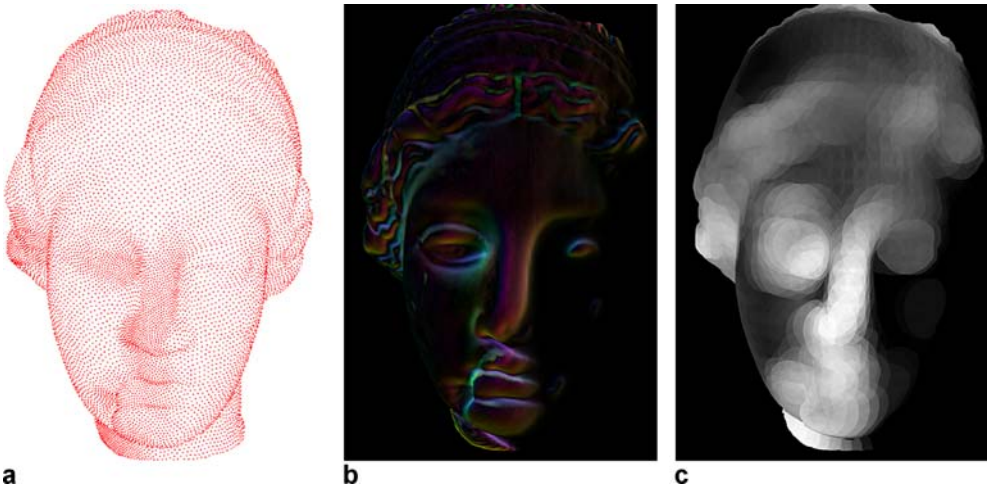


Fig. 9a–c. Visualization of cache distribution, subsurface illuminance gradient, and variance of irradiance with model Igea. **a** Cache distribution. **b** Subsurface illumination gradient. **c** Variance of irradiance

Further improvements include exploring more complex models to determine the cache distribution and devising a reasonable model to determine the radius of split-disk and the upper bound of valid domain of each cache, which are set heuristically in our implementation. Finally, it would also be useful to investigate the accuracy of the dipole diffusion approximation in the presence of complex geometries and to find the solution methods for the heterogeneous materials.

References

1. Carr, N.A., Hall, J.D., Hart, J.C.: Gpu algorithms for radiosity and subsurface scattering. In: Graphics Hardware 2003, pp. 51–59 (2003)
2. Chirstensen, P.H.: Global illumination and all that. SIGGRAPH 2003 Course Note 9 (2003)
3. Dachsbacher, C., Stamminger, M.: Translucent shadow maps. In: Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering, pp. 197–201 (2003)
4. Dorsey, J., Edelman, A., Legakis, J., Jensen, H.W., Pedersen, H.K.: Modeling and rendering of weathered stone. In: Proceedings of SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series, pp. 225–234 (1999)
5. Farell, T.J., Patterson, M.S., Wilson, B.: A diffusion theory model of spatially resolved, steady-state diffuse reflectance for the noninvasive determination of tissue optical properties in vivo. Med. Phys. **19**, 879–888 (1992)
6. Hanrahan, P., Krueger, W.: Reflection from layered surfaces due to subsurface scattering. In: Proceedings of SIGGRAPH 93, Computer Graphics Proceedings, Annual Conference Series, pp. 165–174 (1993)
7. Hao, X., Baby, T., Varshney, A.: Interactive subsurface scattering for translucent meshes. In: 2003 ACM Symposium on Interactive 3D Graphics, pp. 75–82 (2003)
8. Hao, X., Varshney, A.: Real-time rendering of translucent meshes. ACM Trans. Graphics **23**(2), 120–142 (2004)
9. Jensen, H.W., Buhler, J.: A rapid hierarchical rendering technique for translucent materials. ACM Trans. Graphics **21**(3), 576–581 (2002)
10. Jensen, H.W., Marschner, S.R., Levoy, M., Hanrahan, P.: A practical model for subsurface light transport. In: Proceedings of ACM SIGGRAPH 2001, Computer Graphics Proceedings, Annual Conference Series, pp. 511–518 (2001)
11. Lensch, H.P.A., Goesele, M., Bekaert, P., Kautz, J., Magnor, M.A., Lang, J., Seidel, H.P.: Interactive rendering of translucent objects. Comput. Graphics Forum **22**(2), 195–206 (2003)
12. Mertens, T., Kautz, J., Bekaert, P., Seidel, H.P., Reeth, F.V.: Interactive rendering of translucent deformable objects. In: Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering, pp. 130–140 (2003)
13. Pharr, M., Hanrahan, P.M.: Monte Carlo evaluation of nonlinear scattering equations for subsurface reflection. In: Proceedings of ACM SIGGRAPH 2000, Computer Graphics Proceedings, Annual Conference Series, pp. 75–84 (2000)
14. Stam, J.: Multiple scattering as a diffusion process. In: Eurographics Rendering Workshop 1995, pp. 41–50 (1995)

15. Turk, G.: Retiling polygonal surfaces. In: Proceedings of ACM SIGGRAPH 92, Computer Graphics Proceedings, Annual Conference Series, vol. 26 (1992)
16. Ward, G.J., Heckbert, P.: Irradiance Gradients. In: Third Eurographics Workshop on Rendering, pp. 85–98 (1992)
17. Ward, G.J., Rubinstein, F.M., Clear, R.D.: A ray tracing solution for diffuse interreflection. In: Proceedings of ACM SIGGRAPH 88, Computer Graphics Proceedings, Annual Conference Series, vol. 22 (1988)

SHIH-LING KENG received his B.S. and M.S. degrees in computer science from National Chiao Tung University, Taiwan, R.O.C., in 2002 and 2004, respectively. He currently works at CyberLink Corp. as a software engineer. His research interests include realistic image synthesis, real-time rendering, and image processing.

WANG-YEH LEE received his B.S. and M.S. degrees in computer science from National Chiao Tung University, Taiwan, R.O.C., in 1996 and 1997, respectively. He is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering at National Chiao Tung University, Taiwan. His research interests include realistic image synthesis, computer animation, and data visualization. He is a member of the ACM, ACM SIGGRAPH, Eurographics, and IEEE Computer Society.

JUNG-HONG CHUANG is a professor of the Computer Science Department at National Chiao Tung University, Taiwan, ROC. He received his B.S. degree in applied mathematics from National Chiao Tung University, Taiwan, in 1978, and M.S. and Ph.D. degrees in Computer Science from Purdue University in 1987 and 1990, respectively. Current research topics include level-of-detail modeling, mesh parameterization, hybrid rendering, global illumination, and collision detection.