Eurographics Symposium on Rendering 2003
Per Christensen and Daniel Cohen-Or (Editors)

Interactive Rendering of Translucent Deformable Objects

Tom Mertens[†] Jan Kautz[‡] Philippe Bekaert[†] Hans-Peter Seidel[‡] Frank Van Reeth[†]

Expertise Centre for Digital Media
Limburgs Universitair Centrum
Universitaire Campus, 3950 Diepenbeek, Belgium[†]

Max-Planck-Institut für Informatik[‡]
Saarbrücken, Germany

Abstract

Realistic rendering of materials such as milk, fruits, wax, marble, and so on, requires the simulation of subsurface scattering of light. This paper presents an algorithm for plausible reproduction of subsurface scattering effects. Unlike previously proposed work, our algorithm allows to interactively change lighting, viewpoint, subsurface scattering properties, as well as object geometry.

The key idea of our approach is to use a hierarchical boundary element method to solve the integral describing subsurface scattering when using a recently proposed analytical BSSRDF model. Our approach is inspired by hierarchical radiosity with clustering. The success of our approach is in part due to a semi-analytical integration method that allows to compute needed point-to-patch form-factor like transport coefficients efficiently and accurately where other methods fail.

Our experiments show that high-quality renderings of translucent objects consisting of tens of thousands of polygons can be obtained from scratch in fractions of a second. An incremental update algorithm further speeds up rendering after material or geometry changes.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image GenerationDisplay algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism Color Color, Shading, Shadowing and Texture;

1. Introduction

In our daily life, we are surrounded by many translucent objects, such as milk, marble, wax, skin, paper, and so on. The translucency is caused by light entering the material and scattering inside it. This subsurface scattering diffuses the incident light, making small surface detail look smoother. Furthermore, light may scatter through an object, which lights up thin geometric detail if illuminated from behind. These effects create a distinct look that cannot be achieved with simple surface reflection models. Even if a material does not seem to be very translucent at first sight, at the appropriate scale it might exhibit subsurface scattering effects [12] (see also color plate figure 4).

Subsurface scattering effects can be rendered offline using a wide range of methods proposed for participating media, including finite element methods [25, 1, 26], (bidirectional) path tracing [8, 19], photon mapping [14, 3], and diffusion approximations [29].

A major breakthrough was recently achieved with an analytical model for subsurface scattering [15], eliminating the need for numerical simulation of subsurface light transport in homogeneous highly scattering optically thick materials such as milk and marble. In order to use this model in global illumination algorithms, a hierarchical integration technique was proposed by Jensen et al. [13]. Unfortunately, this technique does not appear to allow interactive rendering. The method proposed by [20] is interactive, but only for rigid objects with fixed, possibly inhomogeneous, subsurface scattering properties. Hao et al.[10] produce similar results with their technique, also for fixed geometry. Our goal is to render deformable, translucent objects at interactive rates under varying lighting and viewing conditions (see color page for examples).

The problem at hand is related to the widely studied problem of real-time rendering with complex surface BRDFs e.g. [11, 16, 21]. These techniques cannot handle subsurface scat-

† {Tom.Mertens,Philippe.Bekaert,Frank.VanReeth}@luc.ac.be
‡ {jnkautz,hpseidel}@mpi-sb.mpg.de

tering because they assume that light is directly reflected and not scattered inside the object. Recent work on precomputed radiance transfer [27] can be extended to handle subsurface scattering, as it has already been demonstrated to work for participating media, but it assumes static models.

We show that our goal can be reached with a hierarchical boundary element method to solve the subsurface scattering integral (4) of §2, in the spirit of hierarchical radiosity with clustering [9, 28, 26, 31]. An outline of the method is given in §3. The success of the method is in part due to an efficient and accurate semi-analytic integration method for the needed point-to-patch form-factor like transport coefficients (§4). Design choices of our implementation are described in §5 and results are presented and discussed in §6.

2. Background

2.1. The Subsurface Reflection Equation

We first introduce the necessary background on subsurface scattering and the dipole source approximation recently brought to the attention for rendering translucent materials by Jensen et.al [15, 13].

The shade of a surface point x_o on a translucent object, observed from a direction ω_o, is computed with the following integral:

$$L^{\rightarrow}(x_o,\omega_o) = \int_{\mathsf{S}} \int_{\Omega^+(x_i)} L^{\leftarrow}(x_i,\omega_i) S(x_i,\omega_i;x_o,\omega_o)(\omega_i \cdot N_i) d\omega_i dx_i \quad (1)$$

$L^{\rightarrow}(x_o,\omega_o)$ and $L^{\leftarrow}(x_i,\omega_i)$ denote exitant/incident radiance respectively. S is the object's surface, $\Omega_+(x_i)$ is the hemisphere above x_i in normal direction N_i, and $S(x_i,\omega_i;x_o,\omega_o)$ is the bidirectional surface scattering distribution function (BSSRDF). In general, the BSSRDF is an eight-dimensional function, expressing what fraction of (differential) light energy entering the object at a location x_i from a direction ω_i leaves the object at a second location x_o into direction ω_o. Because of its high dimensionality, it is infeasible to precompute and store this term, especially since it depends on the object's geometry.

Previous techniques [8, 15] show that subsurface scattering can be modeled adequately using two components: single scattering and multiple scattering. Single scattering accounts for only a small fraction of the outgoing radiance of a highly scattering optically thick translucent material such as marble, milk, ... The dominant term is the multiple scattering, which we will focus on.

Multiple scattering diffuses the incident illumination, such that there is almost no dependence on the incident and outgoing direction anymore. Therefore it can be represented at high accuracy as a four-dimensional function $R_d(x_i,x_o)$, which only depends on the incident and outgoing positions. Additionally accounting for the Fresnel transmittance when light enters and leaves the material, we get the following subsurface scattering reflectance function:

$$S(x_i,\omega_i;x_o,\omega_o) = \frac{1}{\pi} F_t(\eta,\omega_o) R_d(x_i;x_o) F_t(\eta,\omega_i). \quad (2)$$

Substituting this into Equation 1, we get:

$$L^{\rightarrow}(x_o,\omega_o) = \frac{1}{\pi} F_t(\eta,\omega_o) B(x_o) \quad (3)$$

$$B(x_o) = \int_{\mathsf{S}} E(x_i) R_d(x_i,x_o) dx_i \quad (4)$$

$$E(x_i) = \int_{\Omega^+(x_i)} L^{\leftarrow}(x_i,\omega_i) F_t(\eta,\omega_i)(N_i \cdot \omega_i) d\omega_i \quad (5)$$

In order to render translucent objects efficiently, one needs an efficient way to solve this equation at every surface point.

2.2. Dipole Source Approximation

First, we need to decide on a model for the BSSRDF, and thus a model for R_d. This model has to fulfill two main criteria: it should be adaptable to different materials, and should provide an efficient solution of equation (4).

An accurate method to determine $R_d(x_i;x_o)$ is to use a full simulation. Many different techniques have been proposed (coming from the area of participating media), e.g. [8, 26, 19, 14]. While these techniques are effective, they fulfill none of the above criteria.

We chose to use a recently introduced model for the BSSRDF in homogeneous media as the basis for our work [15]. This model is based on a dipole source approximation for the solution of a diffusion equation [12, 29] that models light transport in densely scattering media. We shall see that it fulfills the above requirements.

The key idea behind the dipole source approximation for $R_d(x_i,x_o)$ [15], is elegant and simple. An incoming ray at position x_i on a homogeneous, planar, and infinitely large and thick medium is converted into a dipole source (i.e. two sources) at the same position. One source of the dipole is placed at a distance z_v above the surface at x_i. The second source is at a distance z_r below the surface at x_i. $R_d(x_i,x_o)$ is then obtained by summing a sort of illumination contribution at x_o, due to the two sources near x_i. The result is a function of the distance r only, between x_i and x_o (see Table 1).

Although the dipole source approximation is only valid for planar surfaces [12, 18], mis-using it for curved surfaces yields highly plausible results too, as has been shown in several complex examples by Jensen et al. [15, 13]. We will use it in the same spirit. The model is also inherently limited to homogeneous materials. The appearance of heterogeneous materials was simulated by means of texture mapping techniques [15, 20] in previous work and this could be done in the work presented here as well.

$$
\begin{aligned}
R_d(x_i,x_o) &= \frac{\alpha'}{4\pi}\left[z_r(1+\sigma s_r)\frac{e^{-\sigma s_r}}{s_r^3} + z_v(1+\sigma s_v)\frac{e^{-\sigma s_v}}{s_v^3}\right] \\
z_r &= 1/\sigma_t', \qquad z_v = z_r + 4AD \\
s_r &= \|x_r - x_o\|, \text{ with } x_r = x_i - z_r \cdot N_i \\
s_v &= \|x_v - x_o\|, \text{ with } x_v = x_i + z_v \cdot N_i \\
A &= \frac{1+F_{dr}}{1-F_{dr}} \\
F_{dr} &= -\frac{1.440}{\eta^2} + \frac{0.710}{\eta} + 0.668 + 0.0636\eta \\
D &= 1/3\sigma_t', \qquad \sigma = \sqrt{3\sigma_a\sigma_t'} \\
\sigma_t' &= \sigma_a + \sigma_s', \qquad \alpha' = \sigma_s'/\sigma_t' \\
\sigma_s' &= \text{reduced scattering coefficient (given)} \\
\sigma_a &= \text{absorption coefficient (given)} \\
\eta &= \text{relative refraction index (given)} \\
F_t(\eta,\omega) &= \text{Fresnel transmittance factor}
\end{aligned}
$$

Table 1: *The dipole source BSSRDF model. This paper presents an efficient strategy for computing integrals of this model, similar to point-to-patch form factors in radiosity, as well as a hierarchical evaluation algorithm allowing interactive rendering speeds.*

3. Outline of the New Method

In this section, we outline a hierarchical boundary element method to solve equation (4) efficiently, in the style of hierarchical radiosity with clustering. Before elaborating the details of the method in next sections, we compare with related work at the end of this section.

3.1. Discretisation of Equation (4)

The additional surface integral (4) required for subsurface scattering, makes the calculation of subsurface scattering considerably more expensive than local light reflection calculations.

If we break the surfaces of our object to be rendered into regions A_k, and if we assume small lighting variation over each region, making $E(x_i) = E_k$ constant for all $x_i \in \mathsf{A}_k$, equation (4) reduces to the following sum:

$$B(x_o) = \sum_k E_k F(\mathsf{A}_k, x_o) \tag{6}$$

$$F(\mathsf{A}_k, x_o) = \int_{\mathsf{A}_k} R_d(x_i, x_o)\, dx_i \tag{7}$$

We compute the factors $F(\mathsf{A}, x_o)$ for the midpoint of each element w.r.t. all other elements. The irradiance E_k is also evaluated at the midpoints. The sum (6) can then be used to evaluate exitant radiance at each elements midpoint and taken for the constant radiance value for the whole element.

The factors $F(\mathsf{A}, x_o)$ are similar to point-to-patch *form factors* in the radiosity method [2]. There, the form factor usually has a purely geometric meaning, whereas our form factor also encodes the material properties. It is generally much smoother, allowing us to treat $E(x_i)$ as constant for large object patches, and even to ignore visibility.

It turns out that for all but the nearest elements, a single-sample estimate of the form factor is sufficiently accurate. The same criterion for selecting a number of samples can be used as in [13]. We present in §4 a semi-analytic integration method for the form factor. It is indispensable for efficiently and accurately handling nearby elements, which would otherwise sometimes require a large number of samples.

3.2. Clustering

Computing subsurface scattered radiance in this simple way is unfortunately a quadratic procedure in terms of the number of elements. This is too costly for interactive image synthesis, except for the simplest models. A log-linear procedure is obtained by grouping distant elements hierarchically in so called clusters, in a similar way as in hierarchical radiosity with clustering [9, 28, 26, 31].

Each cluster represents a collection of faces, along with a compact representation for these faces. This collection of faces is made up of all faces deeper in the hierarchy. A cluster also stores pointers to one or more child-clusters, which themselves store a representation for all their child-faces. The leaf nodes of this hierarchy correspond with the original faces of which the object is composed. We discuss the requirements and options in constructing the cluster hierarchies further in Section 5.1.

Rather than computing the form factor between each pair of face elements, we first create a candidate *link* between the top level cluster containing all the faces of the object and itself. Next, this link is subjected to a refinement oracle. If the oracle decides that the link would not allow sufficiently accurate integration, the candidate link is refined by opening the cluster at the receiver or emitter side. As such, new candidate links result, which are tested in turn. The refinement oracle and strategy we used is described in Section 5.2.

After linking, we compute the irradiance at all leaf nodes (see Section 5.3). The leaf node irradiance is then *pulled up* to the top of the cluster hierarchy by averaging the irradiance on child clusters. We then recursively traverse the cluster hierarchy and *gather* the irradiance at each cluster element from over its links. This involves multiplying the irradiance at the emitter element with the form factor (7) corresponding to the pair of linked elements and adding the product at the receiver cluster. Clusters always receive contributions at their midpoint. Gathering at vertices is possible too, but care needs to be taken with vertices shared by sibling clusters.

The contributions gathered at all clusters are then pushed

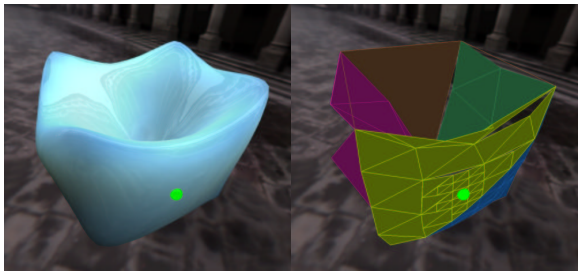

Figure 1: *Part of the cluster hierarchy relevant for calculating subsurface scattered light at the dot. Each cluster element level is depicted with a different shade. The surrounding triangles are directly linked to the leaf containing the dot, whereas the other, larger triangles are connected through one of the leaf's parents. We see that interactions with distant geometry are handled at higher levels. The cracks between the different levels do not cause visible artifacts as long as the refinement criteria are met.*

down the hierarchy, i.e. energy received at higher levels in the hierarchy is distributed to the leaf clusters. The accumulated results at the leaf nodes (the faces of the object) are averaged at the vertices to allow Gouraud shading. More details on the rendering are given in Section 5.4.

As before, a single sample estimate suffices for most form factors. Form factors with nearby face or cluster elements, which would require more than one sample according to Jensen et al.[13], are integrated using the semi-analytical technique of Section 4.

Figure 1 shows the hierarchy of cluster elements, relevant for calculating subsurface scattered light at the indicated spot. Part of the illumination is gathered at a higher level than at the leaf face node.

3.3. Comparison with Previous Work

Jensen's Algorithm Jensen et al. [13] also presented a hierarchical integration technique for (4) that allows to integrate subsurface scattering in global illumination algorithms like photon mapping. First, they cover the surface of the object with a uniform cloud of irradiance sample point locations. These sample points are then sorted into an octree data structure, which also allows to treat distant sample points as a group. Their integration technique is not mesh based and therefore allows a wider class of object descriptions than our method, such as procedurally generated geometry.

The method proposed in this paper aims at hardware accelerated interactive rendering, which is very often done for mesh-based objects of the kind shown throughout this paper. Our method offers a speed advantage in three areas:

- Integration of the distant response: our cluster hierarchy allows to gather the response from distant elements at higher levels, amortizing its cost over many leaf elements.
- Integration of the local response: the semi-analytic form factor integration method of Section 4 is considerably more accurate and cheaper than their sampling approach for uniformly lit nearby areas (see Section 4.2). Only if nearby elements are not approximately uniformly lit, do they need to be broken into smaller pieces. Jensen's sampling approach would correspond to always breaking up nearby elements to a predefined maximum resolution. In our proof-of-concept implementation, only cluster elements are broken up. Adaptive subdivision of face elements still is to be incorporated.
- Hierarchy rebuilding after a material or geometry update: as long as the topology of the object doesn't change, the only effect of geometry or material changes is that links need to be demoted or promoted in the hierarchy [5]. It will be shown in the results section, that our refinement oracle and strategy are cheap enough to allow sufficiently fast relinking. They are even fast enough to re-render from scratch at interactive rates. Re-building the sample octree seems to be a major bottleneck inhibiting interactive updates in Jensen's approach.

The quality offered by both approaches is in principle similar. In Jensen's method, it is controlled by the sample point density. In our approach, it is controlled by the mesh density. Our method may suffer from artifacts due to the use of Gouraud shading, or due to a poor quality mesh. These problems have been well studied in the context of radiosity however and many solutions are available [2].

Both Jensen's method and ours are based on the dipole source approximation BSSRDF model. We mentioned before that this model is only correct for planar boundaries between homogeneous materials, but yields plausible results if mis-used for curved surfaces. A heterogeneous material appearance can be simulated with texture mapping techniques [15, 20].

Hierarchical Radiosity (HR) Radiosity and the subsurface scattering problem share some similarities, which the authors of [20] also noticed. A full surface to surface transfer needs to be computed. Hierarchical methods can relieve the computational load considerably, as HR demonstrates. Our method however differs from HR in the following way:

- Interreflections do not need to be computed. I.e. contributions collected in the hierarchy do not need to be redistributed.
- Whereas HR computes area-to-area transfers, we can afford to just sample area at a single sample point for which our form factor was designed.
- We always assume full visibility, allowing for faster link refinement. Moreover it reduces possible artifacts caused by large contributions collected at higher levels.
- Cluster roots can be linked to themselves and their children (and vice versa). As a matter of fact, these links

represent a very important contribution, i.e. the global response.

Lensch et al. [20] Comparing to this work, our technique is superior in computing the global response since they rely on a matrix multiplication to scatter from one patch to another. In other words, this is a quadratic procedure for which the interactive rendering rate can easily break down in practice.

4. Subsurface Scattering Form Factor

Our rendering algorithm is based on a discretization of the object surfaces into planar polygons A_i that both fulfill certain constraints on size (discussed below) and that can be regarded as homogeneously lit (irradiance E_i). In that case, the outgoing radiosity (4) at any point x_o can be obtained by summing contributions $E_i F(\mathsf{A}_i, x_o)$. The integral $F(\mathsf{A}_i, x_o) := \int_{\mathsf{A}_i} R_d(x_i, x_o) dx_i$ is similar to a point-to-patch form factor in radiosity: it expresses the fraction of light entering an object through the polygon A_i, that gets scattered to the point x_o. We propose here an efficient, custom integration technique for it.

For planar surfaces, R_d is symmetric in x_o and x_i. Hence, the fraction of light that scatters through an object from a patch A_i to a point x_o, is the same as the fraction of light entering the object at x_o that gets scattered into A_i. This is no longer true when using the dipole approximation formula for curved surfaces, due to the fact that the dipole approximation itself is in principle not valid in that case. However, using the dipole approximation (with inverse interpretation) will be shown to result in plausible and attractive images, as demonstrated in previous work.[15]

As mentioned before, $R_d(x_i, x_o)$ is the sum of two terms: the contribution from two sources in a dipole configuration. The terms are similar so that we will focus on the integral of the contribution of a single source point d of the dipole (see Table 1):

$$I = \int_{\mathsf{A}} z(1+\sigma s)\frac{e^{-\sigma s}}{s^3} dx. \qquad (8)$$

s denotes the distance between the point x on the polygon A to the considered dipole source d. We first assume that A is a triangle. We shall see that our result straightforwardly extends to the case of arbitrary planar polygons. Figure 2 and Table 2 illustrate and explain the symbols used in our derivation.

We solve the integral in polar coordinates in A's supporting plane, using the orthogonal projection of d onto this plane, called d', as the pole:

$$I = \int_{\theta_{min}}^{\theta_{max}} \int_{r_{min}(\theta)}^{r_{max}(\theta)} z(1+\sigma s)\frac{e^{-\sigma s}}{s^3} r\, dr\, d\theta \qquad (9)$$

Consider, to start with, the case that one of A's vertices coincides with d' so that $r_{min}(\theta) = 0$ for all polar angles θ. Change of integration variable r to the distance $s = \sqrt{r^2+h^2}$, and substitution of $u = \sigma s$, then yields:

$$\begin{aligned} I &= \int_{\theta_1}^{\theta_2} \int_0^{s(\theta)} z(1+\sigma s)\frac{e^{-\sigma s}}{s^3} s\, ds\, d\theta \\ &= \int_{\theta_1}^{\theta_2} \Big[-\frac{z\sigma}{u} e^{-u}\Big]_{\sigma h}^{\sigma s(\theta)} d\theta \\ &= \frac{z}{h} e^{-\sigma h}(\theta_2 - \theta_1) - z\int_{\theta_1}^{\theta_2} \frac{e^{-\sigma s(\theta)}}{s(\theta)} d\theta. \qquad (10) \end{aligned}$$

$s(\theta)$ denotes the distance from d to points on the triangle edge opposite to d'. Our area integral thus has been reduced to an integral over one triangle edge.

For a general triangle, the sum of three such expressions is obtained: one for each edge of the triangle. The contribution of back facing edges (furthest away from d') is counted positive, while the contribution of front facing edges is subtracted (see Figure 2).

For any closed polygon, the first terms in (10), called I_1

τ	supporting plane of triangle
d	single source of dipole
d'	d orthogonally projected on τ
d''	d' orthogonally projected on edge
R	$\|d' - d''\|$
S	$\|d - d''\|$
h	$\|d - d'\|$, height of d w.r.t. τ
S^2	$R^2 + h^2$
t	parameter on edge, t=0 indicates d''
t_0, t_1, t_2	midpoint, start and end of edge, respectively
Δt	$t = t_0 + \Delta t$
r, s	$s^2 = r^2 + h^2$, $r^2 = R^2 + t^2$
L	edge length, $L = t_2 - t_1$

Table 2: *Symbols used in the form factor derivation.*

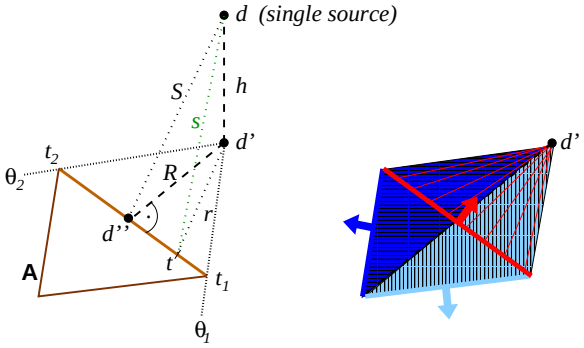

Figure 2: *Left: Depiction of used variables (for one edge). Right: Integration over the triangle is performed by integrating along the edges. The contribution of front facing edges is subtracted from the contribution of back facing edges.*

from now on, cancel if d' is outside the polygon. If d' is inside, their sum equals $2\pi\frac{z}{h}e^{-\sigma h}$.

The second terms, which we call I_2, are more complicated. Change of integration variable from θ to t, the distance along the edge segment measured w.r.t. the orthogonal projection d'' of the dipole source point d onto the edges supporting line, yields:

$$t(\theta) = R\tan\theta \quad \text{and} \quad d\theta = \frac{R\,dt}{R^2+t^2} \tag{11}$$

$$I_2 = z\int_{\theta_1}^{\theta_2} \frac{e^{-\sigma s}}{s}\frac{R\,dt}{R^2+t^2} \tag{12}$$

$$= zR\int_{t_1}^{t_2} \frac{1}{\sqrt{S^2+t^2}}\frac{1}{R^2+t^2}e^{-\sigma\sqrt{S^2+t^2}}dt. \tag{13}$$

$t_1 = R\tan(\theta_1)$ and $t_2 = R\tan(\theta_2)$ denote the distances from the edge endpoints to d''. We did not find an analytic solution to this integral as such, but Taylor series expansion w.r.t. the midpoint of the edge (parameter t_0) of the three main factors in the integrand allows to obtain a good approximation:

$$\frac{1}{\sqrt{S^2+t^2}} = \frac{1}{s_0}\Big(1-\frac{1}{2}\Big[\frac{2t_0\Delta t}{s_0^2}+\big(\frac{\Delta t}{s_0}\big)^2\Big]+\ldots\Big) \tag{14}$$

$$\frac{1}{R^2+t^2} = \frac{1}{r_0^2}\Big(1-\Big[\frac{2t_0\Delta t}{r_0^2}+\big(\frac{\Delta t}{r_0}\big)^2\Big]+\ldots\Big) \tag{15}$$

$$e^{-\sigma s} = e^{-\sigma s_0}\Big(1-\sigma s_0\Big[\frac{2t_0\Delta t}{s_0^2}+\big(\frac{\Delta t}{s_0}\big)^2\Big]+\ldots\Big) \tag{16}$$

Here, $s_0 := s(t_0)$ and $r_0 := r(t_0)$.

The product of the Taylor approximations results in a simple polynomial and its integration is straightforward:

$$I_2 = \frac{zRe^{-\sigma s_0}}{r_0^2 s_0}L\Big(1+\ldots\Big). \tag{17}$$

The higher order terms can be safely ignored by imposing constraints on the edge length $L = t_2 - t_1$, as discussed in the next section.

In short, the contribution of a single edge e due to one of the dipole source points is $I_e = I_1 + I_2$. The form factor of a triangle A with three edges (a,b,c) and the dipole is:

$$\int_{\mathsf{A}} R_d(x_i,x_o)dx_i = \frac{\alpha'}{4\pi}\big[\pm(I_a^r+I_a^v) \pm (I_b^r+I_b^v) \pm (I_c^r+I_c^v)\big], \tag{18}$$

where I^r and I^v denote the contribution from each dipole source point. The sign for a term in the sum is positive if it is a back facing edge for x_o, otherwise it is negative.

4.1. Error Analysis

The error due to ignoring second and higher order terms in the Taylor series expansions (14) to (16) is straightforward to analyze if one makes sure that the first order term is sufficiently smaller than 1. The series expansions converge rapidly in that case.

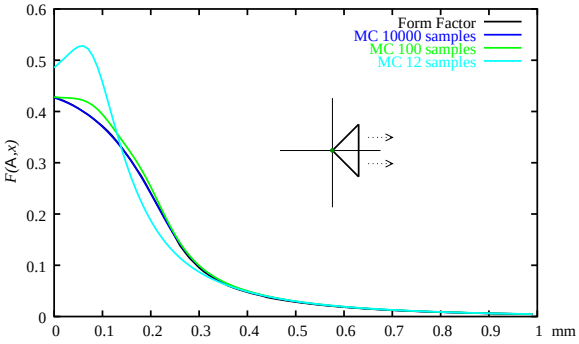

Figure 3: *This graph shows $F(\mathsf{A},x)$ of a standard triangle* A *with the tip at $x = (0..1cm, 0cm)$. The base edge is $L = 1cm$ and the height is $0.5cm$. The x-axis of the graph shows the distance of the triangle's tip to O. The sample point x_o lies also in the origin (planar setting). Skim milk was used as the material. The black curve is evaluated using our method, the colored curves show Monte Carlo evaluations. Our form factor and 10000 Monte Carlo samples result in virtually the same curve.*

Suppose, we take $\frac{2t_0}{s_0}\frac{\Delta t}{s_0} + \big(\frac{\Delta t}{s_0}\big)^2 < \frac{1}{2}$ for the first factor. Inspecting the higher order terms, we then find that the relative error is about 15% maximally. This condition will be satisfied if we ensure that $\frac{|\Delta t|}{s_0} < \frac{1}{5}$.

A sufficient condition for fast convergence of the second factor is $\frac{|\Delta t|}{r_0} < \frac{1}{5}$. This condition is stronger than the previous one since $r_0 \le s_0 = \sqrt{r_0^2+h^2}$.

For the last factor we need to take $|\Delta t|\sigma < \frac{1}{5}$. With $L = 2|\Delta t|$, we have the following conditions:

$$\frac{L}{s_0} < \frac{L}{r_0} < \frac{2}{5} \quad \text{and} \quad L < \frac{2}{5\sigma}. \tag{19}$$

If these conditions are satisfied, the relative error on the integrand of I_2 is maximally roughly 50%. This apparently high error can only occur near the end-points of the edges. On most part of the edges, the error is much lower and so is the error on the integrals.

Edges not satisfying the above constraints are recursively subdivided until the conditions hold. The error decreases exponentially with subdivision.

Even when ignoring second and higher order terms, first order terms still appear in the result (17) for I_2. Taking into account the same conditions above, it turns out that also the contribution from the first order terms are bounded and can be safely ignored.

The condition $L < \frac{2}{5\sigma}$ is a global one. For more translucent materials this condition becomes less important. We noticed no significant impact on quality when ignoring this condition.

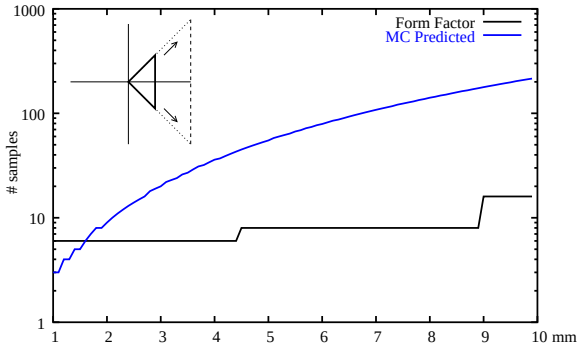

Figure 4: *This graph compares the number of samples we have to take vs. Jensen and Buhler's method [13] to evaluate the BSSRDF over an incrementally scaled standard triangle. We chose $x_o = O$. The tip of the triangles lies at the origin as well. The x-axis of the graph shows the distance from the triangle's base edge to origin. In this example we chose the material coefficients for marble. Although Jensen and Buhler's method needs less samples for small triangles, their predicted number of samples is too small to achieve acceptable accuracy, see Figure 3.*

Figure 3 compares the results of our method with Monte Carlo integration. It shows that our method indeed produces accurate results with the above conditions. Using only 12 samples as done in [13], produces rather low accuracy for this setting.

4.2. Discussion

Our integration strategy has several distinct properties. First, it turns out that many of the edge contributions in a polygon mesh will cancel out. Indeed, when integrating over neighboring planar triangles, shared edges are iterated over twice, once per sharing triangle, and in opposite direction. If the triangles receive the same irradiance, the terms $\Gamma := I_e^r + I_e^v$ for each such edge cancel out. Only the contribution of outer edges remains. For the same reason, the form factor for an arbitrary planar polygon can be obtained by simply summing the appropriately signed contributions Γ for all polygon edges.

Our integration strategy is partly based on analytic integration, and partly on numerical approximation through Taylor series expansion. The level of accuracy can be controlled by loosening or tightening the refinement conditions (Equation 19).

In figure 4, we show the number of samples that we need to take for different triangles. In this case, a sample corresponds to computing I_2 for a line segment (there can be multiple line segments for an edge, due to subdivision in order to meet the conditions). We compare this to the number of samples needed by Jensen and Buhler [13]. The efficiency of our integration strategy is evident.

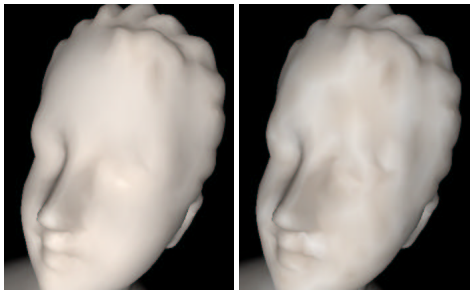

Figure 5: *Comparison of visual quality with the form factor procedure and point sampling for marble at scale 20cm. Low frequency noise is clearly visible with point sampling.*

For distant clusters, the form factor will be sufficiently small to be computed with a single point sample at the midpoint. As figure 4 indicates, there is nothing to gain by using the form factor in these cases. Due to the steeply descending nature of R_d as distance increases, we see that only for nearby clusters the form factor is appropriate.

By calculating form factors directly with an object mesh, we avoid the need to generate and keep track of a dense set of uniformly distributed surface point samples. This set requires costly operations to be performed each time the object geometry is changed. Furthermore, our integration method does not suffer from high-frequency noise like Monte Carlo methods [15]. We compared an obvious alternative to our form factor algorithm, i.e. a form factor using simple uniform sampling. The amount of samples is chosen according to the point sample distribution criteria used by Jensen et al. [13]. It turns out that our form factor procedure is superior for both quality and performance. For moderately translucent materials, R_d is quite steep and more samples are needed in the integration. While making a material less translucent we noticed a low frequency noise and a serious frame rate drop. See table 5 and figure 5 for details.

Our form factor algorithm assumes polygons with sufficiently constant irradiance. For sampling from environment maps and point/spot lights, this assumption is acceptable since lighting does not tend to change dramatically. Care must be taken in situations where sharp shadow borders occur, although for highly scattering media this becomes less of a problem due to the diffusing nature of multiple scattering.

5. Implementation

We will now discuss several implementation-related issues of our approach.

5.1. Mesh Hierarchy Construction

Our algorithm is given a hierarchical structure such as a multi-resolution mesh, a face cluster hierarchy or a subdivision surface, but needs to ensure that finer clusters are fully embedded in coarser levels. For instance, approximating subdivision schemes and progressive meshes cannot be applied here.

5.1.1. Face Cluster Hierarchy

Face cluster hierarchies have been used in different applications, but they are best known from face cluster radiosity [31]. Such a hierarchy is built by merging neighboring clusters until a single cluster is reached containing all original faces.

We build our hierarchy bottom-up with an algorithm similar to [6]. Unlike their work, our measure only tries to cluster faces as compactly as possible, ignoring surface curvature. Instead we make sure during linking, i.e. at run-time where geometry and curvature may change, that only near-planar clusters are chosen, since our form factor is correct for planar clusters only. To this end we take a simple curvature measure into account during linking in order to reduce the error for the assumption of planar clusters: $\frac{A'}{A}$, where A' and A are the total projected area and total area of the cluster triangles, respectively. The form factor from a cluster is then approximated by projecting all its triangles onto the plane defined by the cluster's midpoint and averaged normal, and working with the resulting polygon.

5.1.2. Subdivision

Another method to obtain a hierarchy with the desired properties is surface subdivision. Since the finer clusters have to be fully embedded in the next coarser level, we chose to use subdivision based on 4-to-1 splits.

Butterfly. Any subdivision scheme to be used must be interpolatory to fulfill our hierarchy criteria. For example, the butterfly algorithm [4, 32] can be used to generate hierarchies fulfilling this criteria. An example can be seen in Figure 1.

Shrink-Wrapped. Using a similar algorithm to [17], we create hierarchies for arbitrary closed meshes. We first project the mesh from its center onto a sphere, and relax it using the umbrella operator. We then take a coarse base mesh, whose vertices are shared with the original mesh, and recursively split. Whenever a new vertex is introduced, we project it onto the sphere, find in which triangle from the original mesh it lies, and use the interpolated coordinates from the original mesh as the new coordinates for the new vertex.

5.1.3. Discussion

The hierarchy generated by subdivision is more efficient than the face clustering method, since every cluster is represented by exactly one triangle. But it imposes also more restrictions on the mesh. The butterfly method can only produce simple shapes from a low polygon count base mesh. Although the shrink-wrapping can work on arbitrary meshes, the approach often leads to unevenly tessellated meshes. Overall, it is a compromise between efficiency and quality.

5.2. Refinement Oracle and Strategy

We use Jensen's maximum solid angle criteria [13], as it is cheap to evaluate and practical:

- When the maximum deviation of the solid angle ω from R's children's midpoints to E's collection of faces exceeds a certain threshold ε_1, split and move the link down the hierarchy at R. Repeat, until the deviation is below ε_1.
- Before connecting a link to an emitter cluster E, check ω from R's midpoint to E's faces. If it is above a threshold ε_2, split and move the link down at E. Repeat, until it is below ε_2.

More advanced linking criteria which reflect the magnitude of the actual form factor better, may be devised in order to reduce link count.

5.3. Irradiance Sampling

Irradiance is computed at each leaf cluster (triangle) at runtime for environment maps and point light sources. In the latter case, shadows can be computed by employing a variant of shadow mapping [30]. In the former case, we sample irradiance from an environment based on its spherical harmonics coefficients. Nine coefficient suffice since the incoming radiance undergoes a cosine-weighted integration over the hemisphere and is thus bandlimited [24]. Shadows however cannot be taken into account, unless we assume rigid objects.[27] The resulting irradiance value is obtained with a simple dot product. Prefiltered environment [7] maps can also be used, but we did not implement this alternative.

Note that irradiance is only sampled at leaf clusters; higher clusters *pull* the averaged irradiance from their children.

5.4. Rendering

Our rendering algorithm provides diffuse exitant radiance at each vertex, which is passed to a set of hardware shaders. The final rendering includes reflection mapping or phong shading. We added a Fresnel factor, as seen in equation 3, which is also used for rendering reflections from environment maps.

Everything is computed in high dynamic range at floating point precision. A simple gamma curve per pixel is applied to the final shading for the purpose of tone mapping. This enhances the global response which is typically much lower that the local response, and adds a lot to the realism.

5.5. Interactivity

Our algorithm operates at different levels of interactivity:

1. **Render Mode**. Shading is re-evaluated under varying lighting conditions. Irradiance is recomputed at the cluster leaf nodes and distributed along the links and down the hierarchy to the receiver leaf nodes, where they are used for rendering.
2. **Incremental Mode**. Lighting, material and geometry can be altered. The algorithm incrementally updates the links' form factors which are connected to altered clusters. In case the material changes, every link is tagged to be updated. We use a simple demotion/promotion procedure [5] to maintain consistency in the hierarchy: links which do not meet the conditions are moved one level up or down. Interactivity can be controlled by bounding the number of form factor computations per frame.
3. **On-the-fly Mode**. Each frame we traverse the hierarchy to perform a full evaluation from scratch. This mode avoids the overhead of keeping track of links, resulting in quicker updates. Memory requirements are also reduced significantly. However, the frame rate is of course lower and may compromise interactivity on slower systems.

6. Results

We implemented our algorithm in C++ on a dual Intel Xeon 2.4Ghz 2Gb RAM configuration with an ATI Radeon 9700 graphics board.

Table 3 shows that our algorithm behaves roughly log-linear in the number of triangles.

T	L	$\frac{L}{T}$	on-the-fly	incr.	render
2016	89K	44.1	20	8	5
8160	388K	47.5	47	23	13
32736	1744K	53.3	188	68	43
131736	6905K	52.7	722	351	199

Table 3: *Model complexity versus link count and running time (ms). Looking at the links-per-triangle ratio, we notice that there is roughly a log-linear correlation between the number of triangles T and the number of links L. The experiment was done for a marble butterfly subdivision model scaled at 2.5cm.*

In table 4 we illustrate the performance of our approach with different models. We see that the number of links is mesh-dependent. In this experiment we choose very conservative thresholds for the refinement oracle such that the quality of the solution is guaranteed for every situation.

Table 5 illustrates the efficiency of the form factor compared to point sampling. We see that it behaves more robustly in rendering time when the local response gains importance.

model	type	T	L	on-the-fly
horse	FC	8764	992K	97
bust	FC	10518	1258K	102
elk	FC	11384	1743K	165
candle	S	4474	435K	41
cube	S	16380	1572K	154

Table 4: *Overview of performance with different models. Timings for* on-the-fly *mode are in ms. Material was marble scaled at 10cm. The hierarchy types are indicated by: (F)ace (C)lustering or (S)ubdivision.*

scale (m)	form factor (ms)	point sampling (ms)
.05	160	162
.1	160	168
.2	161	222
.4	169	501
.8	340	1778

Table 5: *Comparison of the form factor procedure and point sampling on marble. The object's bounding box is scaled at different sizes. The duration of a full* on-the-fly *evaluation is measured.*

We will now discuss some results depicted in figure 6. All screenshots were taken interactively in *on-the-fly* mode, at a frame rate ranging from 3 to 15fps, roughly. Except for figure 6.5, which is rendered in *incremental* mode at 7.5fps.

In figure 6.1-2-3 a horse is rendered with three different materials: skim milk, ketchup and candle wax, respectively. Notice the color shifts across the model in figure 6.1. The scattering of light is very obvious for the thin geometric features in figure 6.3 when the model is lit from behind.

In figure 6.4, the scale of whole milk changes from 1.0m, 10cm, 5cm, 2cm, 1cm, to 5mm, respectively. The hierarchy for the tweety model here was generated using shrink wrapping. In the leftmost image, the shading is practically the same as with simple diffuse reflection. The awkward appearance in regions where normals are orthogonal to the direction of incoming light, is due to (gamma curve) tone mapping.

We interactively added a bump to a subdivided cube in figure figure 6.5. The resulting model is lit with a spot light, and enhanced with shadow mapping. Notice how the shadow cast by the bump is 'leaking' further on the adjacent plane. For that same cube we applied a Perlin noise [23] distortion (figure 6.5): 3D noise sampled at each vertex and perturbs its position along the normal. Each frame we increment the offset to the noise's sampling position, resulting in a full deformation on 16K triangles. This situation is rendered inter-

actively at roughly 4-5 frames per second, including irradiance sampling (spot light and shadows).

Figure 6.7a depicts how a simple subdivision shape is used to simulate the appearance of a candle. It is lit from inside by a moving and flickering point light source. When we apply a simple 'twist' deformation, the model gets thinner, resulting in more light passing through from the inside (figure 6.7b). As this is a very simple model of nearly 5K triangles, this experiment runs in real-time (15fps).

We show a translucent toy elk to show that models are not restricted to be of genus 0. The spherical wheel has been dented interactively, resulting in more scattered light passing through.

Figure 6.9a shows a visualization of the clusters used for shading a triangle on the ear on the left. Interactions at the different levels in the hierarchy are colored uniquely. Right next to it (6.9b) we show the actual rendering.

The appearance of a marble bust in figure 6.10 has been enriched with Fresnel reflection from an environment map.

7. Conclusion

7.1. Summary

We have presented an efficient technique for rendering translucent deformable objects. It allows to interactively change the object's geometry as well as the subsurface scattering parameters.

To this end we introduce a novel boundary element method similar to hierarchical radiosity. Our approach differs from existing work by using a mesh hierarchy. It has been proven to be highly suitable for deforming geometry and changing material properties. The integration over the whole surface can be handled at different levels in the hierarchy, thus reducing the complexity for the integration from quadratic to log-linear. The derivation of a semi-analytical form factor for the BSSRDF introduced by Jensen et al. [15] is given, for the purpose of handling the local response very efficiently. It computes the total contribution of a triangle scattering light onto a point. Our experiments show that it is an improvement over existing point sampling schemes for both performance and accuracy.

We achieve interactive rendering rates for complex objects consisting of tens of thousands of polygons. A full evaluation of the shading can be obtained from scratch in fractions of a second. An incremental update routine assists in increasing the degree of interactivity after material or geometry changes.

7.2. Future Work

The current implementation works on a fixed mesh with a fixed hierarchy. It is worthwhile to explore adaptive meshing techniques to handle situations where irradiance is not constant over a triangle, such as sharp shadow borders. This is a common problem in radiosity as has been studied thoroughly [2]. We would like to extend our technique to handle the very local response for fine surface detail such as displacement or bump maps, which it currently can only handle with finely tesselated meshes. Currently our implementation does not handle topology changes. An efficient algorithm could be devised to update the hierarchy at runtime in this case, which seems feasible in combination with *on-the-fly* evaluation.

Better oracle functions may lead to more efficient linking, reducing overall rendering cost. Alternatives which take the magnitude of the form factor into account are suitable candidates.

Interpolation methods employing higher-order basis functions will remove artifacts caused by Gouraud shading, and may reduce the need of many links. Also, the idea of introducing higher order basis functions during hierarchical integration, as has been done for HR [2], seems promising.

Our technique is currently restricted to homogeneous materials, a limitation imposed by using the dipole source BSSRDF model. We would like to e.g. allow for impurities in the material casting volumetric shadows inside the material (e.g. marble). The single scattering term is currently ignored. We would like to add a single scattering term to the renderings, which can be done with a GPU-based implementation of the Hanrahan-Krueger model [8], as demonstrated by NVIDIA [22]. It would be interesting to investigate what the accuracy is of the underlying BSSRDF model (i.e. using the dipole diffusion approximation), for curved surfaces and arbitrary geometry as it is only valid for semi-infinite planar media.

Acknowledgements

The authors would like to thank Jens Vorsatz for his valuable help with the hierarchical meshes.
This work is partly financed by the European Regional Development Fund. The first author was also supported by a Marie Curie Doctoral Fellowship.

References

1. Philippe Blasi, Bertrand Le Saëc, and Christophe Schlick. A Rendering Algorithm for Discrete Volume Density Objects. *Computer Graphics Forum*, 12(3):201–210, 1993.

2. Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Cambridge, MA, 1993.

3. Julie Dorsey, Alan Edelman, Justin Legakis, Henrik Wann Jensen, and Hans Køhling Pedersen. Modeling and Rendering of Weathered Stone. In *Proceedings of SIGGRAPH 99*, pages 225–234, 1999.

4. Nira Dyn, David Levin, and John A. Gregory. A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. *ACM Transactions on Graphics*, 9(2):160–169, April 1990.

5. David A. Forsyth, Chien Yang, and Kim Teo. Efficient Radiosity in Dynamic Environments. In *Fifth Eurographics Workshop on Rendering*, pages 313–323, June 1994.

6. Michael Garland, Andrew Willmott, and Paul S. Heckbert. Hierarchical Face Clustering on Polygonal Surfaces. In *2001 ACM Symposium on Interactive 3D Graphics*, pages 49–58, March 2001.

7. Ned Greene. Applications of world projections. In *Graphics Interface 86*, pages 108–Ű114, May 1986.

8. Pat Hanrahan and Wolfgang Krueger. Reflection from layered surfaces due to subsurface scattering. In *Proceedings of SIGGRAPH 93*, pages 165–174, 1993.

9. Pat Hanrahan, David Salzman, and Larry Aupperle. A Rapid Hierarchical Radiosity Algorithm. In *Computer Graphics (Proceedings of SIGGRAPH 91)*, volume 25, pages 197–206, July 1991.

10. Xuejun Hao, Thomas Baby, and Amitabh Varshney. Interactive Subsurface Scattering for Translucent Meshes. In *Proceedings 2003 ACM Symposium on Interactive 3D Graphics*, page to appear, april 2003.

11. Wolfgang Heidrich and Hans-Peter Seidel. Realistic, Hardware-accelerated Shading and Lighting. In *Proceedings of SIGGRAPH 99*, pages 171–178, August 1999.

12. Akira Ishimaru. *"Wave Propagation and Scattering in Random Media"*, volume 1. Academic Press, 1978.

13. Henrik Wann Jensen and Juan Buhler. A Rapid Hierarchical Rendering Technique for Translucent Materials. *ACM Transactions on Graphics*, 21(3):576–581, July 2002.

14. Henrik Wann Jensen and Per H. Christensen. Efficient Simulation of Light Transport in Scenes With Participating Media Using Photon Maps. In *Proceedings of SIGGRAPH 98*, pages 311–320, 1998.

15. Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A Practical Model for Subsurface Light Transport. In *Proceedings of SIGGRAPH 2001*, pages 511–518, August 2001.

16. Jan Kautz and Michael D. McCool. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In Dani Lischinski and Greg Ward Larson, editors, *Tenth Eurographics Rendering Workshop 1999*, pages 281–292, June 1999.

17. Leif P. Kobbelt, Jens Vorsatz, Ulf Labsik, and Hans-Peter Seidel. A Shrink Wrapping Approach to Remeshing Polygonal Surfaces. *Computer Graphics Forum*, 18(3):119–130, September 1999.

18. Jan J. Koenderink and Andrea J. van Doorn. Shading in the Case of Translucent Objects. In *Human Vision and Electronic Imaging VI*, pages 312–320. SPIE, 2001.

19. Eric P. Lafortune and Yves D. Willems. Rendering Participating Media with Bidirectional Path Tracing. In *Eurographics Rendering Workshop 1996*, pages 91–100, 1996.

20. Hendrik P. A. Lensch, Michael Goesele, Philippe Bekaert, Jan Kautz, Marcus A. Magnor, Jochen Lang, and Hans-Peter Seidel. Interactive Rendering of Translucent Objects. In *Proceedings of Pacific Graphics 2002*, pages 214–224, October 2002.

21. David McAllister, Anselmo Lastra, and Wolfgang Heidrich. Efficient Rendering of Spatial Bi-directional Reflectance Distribution Functions. In *Proceedings of Graphics Hardware 2002*, pages 79–88, September 2002.

22. Nvidia home page. `http://www.nvidia.com`.

23. Ken Perlin. An image synthesizer. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 287–296. ACM Press, 1985.

24. Ravi Ramamoorthi and Pat Hanrahan. An Efficient Representation for Irradiance Environment Maps. In *Proceedings of SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 497–500, August 2001.

25. Holly E. Rushmeier and Kenneth E. Torrance. Extending the Radiosity Method to Include Specularly Reflecting and Translucent Materials. *ACM Transactions on Graphics*, 9(1):1–27, 1990.

26. François X. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):240–254, 1995.

27. Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. In *Proceedings of SIGGRAPH 2002*, pages 527–536, July 2002.

28. Brian Smits, James Arvo, and Donald Greenberg. A Clustering Algorithm for Radiosity in Complex Environments. *Computer Graphics*, 28(Annual Conference Series):435–442, July 1994.

29. Jos Stam. Multiple scattering as a diffusion process. In *Eurographics Rendering Workshop 1995*, pages 41–50, 1995.

30. Lance Williams. Casting curved shadows on curved surfaces. In *Computer Graphics (SIGGRAPH '78 Proceedings)*, volume 12, pages 270–274, August 1978.

31. Andrew Willmott, Paul S. Heckbert, and Michael Garland. Face Cluster Radiosity. In *Eurographics Workshop on Rendering 1999*, pages 293–304, June 1999.

32. Denis Zorin, Peter Schröder, and Wim Sweldens. Interpolating Subdivision for Meshes with Arbitrary Topology. In *Proceedings of SIGGRAPH 96*, pages 189–192, August 1996.

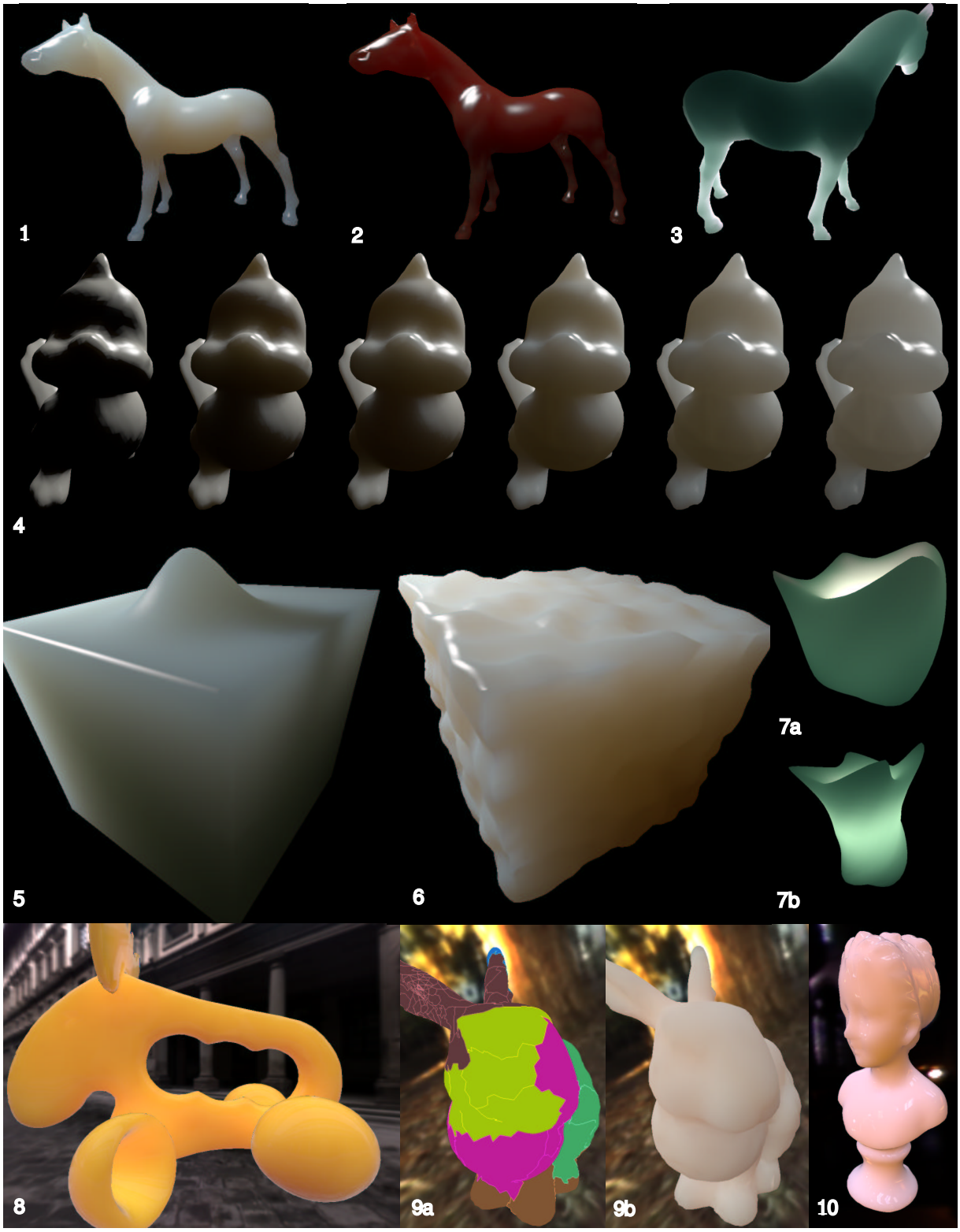

Figure 6: *Interactive rendering results.* ***1-2-3****: Horse model with different materials.* ***4****: Changing the scale of subsurface scattering for whole milk.* ***5****: Tesselated cube with bump casting shadow.* ***6****: Deformation on skim milk cube using Perlin noise.* ***7****: Twist-deformation on candle.* ***8****: Example of a genus 1 model.* ***9****: Visualization of the traversed clusters in the hierarchy for shading the bunny's ear.* ***10****: Marble bust rendered in a high dynamic range environment.*