Interactive Rendering of Translucent Objects

Hendrik P.A. Lensch Michael Goesele Philippe Bekaert Jan Kautz
Marcus A. Magnor Jochen Lang Hans-Peter Seidel
Max-Planck-Institut für Informatik
Saarbrücken, Germany
{lensch,goesele,bekaert,jnkautz,magnor,lang,hpseidel}@mpi-sb.mpg.de

Abstract

This paper presents a rendering method for translucent objects, in which view point and illumination can be modified at interactive rates. In a preprocessing step the impulse response to incoming light impinging at each surface point is computed and stored in two different ways: The local effect on close-by surface points is modeled as a per-texel filter kernel that is applied to a texture map representing the incident illumination. The global response (i.e. light shining through the object) is stored as vertex-to-vertex throughput factors for the triangle mesh of the object. During rendering, the illumination map for the object is computed according to the current lighting situation and then filtered by the precomputed kernels. The illumination map is also used to derive the incident illumination on the vertices which is distributed via the vertex-to-vertex throughput factors to the other vertices. The final image is obtained by combining the local and global response. We demonstrate the performance of our method for several models.

Keywords: *subsurface scattering, interactive rendering, hardware-accelerated rendering*

Contact author:

Hendrik P.A. Lensch
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
Phone: +49-681-9325-428
FAX: +49-681-9325-499
Email: `lensch@mpi-sb.mpg.de`

Interactive Rendering of Translucent Objects

Hendrik P.A. Lensch Michael Goesele Philippe Bekaert Jan Kautz
Marcus A. Magnor Jochen Lang Hans-Peter Seidel
Max-Planck-Institut für Informatik
Saarbrücken, Germany
{lensch,goesele,bekaert,jnkautz,magnor,lang,hpseidel}@mpi-sb.mpg.de

Abstract

This paper presents a rendering method for translucent objects, in which view point and illumination can be modified at interactive rates. In a preprocessing step the impulse response to incoming light impinging at each surface point is computed and stored in two different ways: The local effect on close-by surface points is modeled as a per-texel filter kernel that is applied to a texture map representing the incident illumination. The global response (i.e. light shining through the object) is stored as vertex-to-vertex throughput factors for the triangle mesh of the object. During rendering, the illumination map for the object is computed according to the current lighting situation and then filtered by the precomputed kernels. The illumination map is also used to derive the incident illumination on the vertices which is distributed via the vertex-to-vertex throughput factors to the other vertices. The final image is obtained by combining the local and global response. We demonstrate the performance of our method for several models.

Figure 1. A back lit marble horse sculpture. Left: using a traditional surface based light reflection model. Right: taking into account subsurface scattering of light. Translucency effects are very clear in particular at the ears and the legs. The sculpture measures about 5cm head to tail. This paper presents a rendering method which convincingly reproduces translucency effects as shown in the right image, under dynamic viewing and illumination conditions and at interactive rates.

1 Introduction

On the appropriate scale, the visual appearance of most natural as well as synthetic substances is profoundly affected by light entering the material and being scattered inside [9]. Examples of materials whose macroscopic appearance depends on the contribution from subsurface scattered light include biological tissue (skin, leaves, fruits), certain rocks and minerals (calcit, fluorit, silicates), and many other common substances (snow, wax, paper, certain plastics, rubber, lacquer). Depending on the scale of display, conventional, surface-based reflection functions may only unconvincingly mimic the natural visual impression of such materials (see Figure 1).

Unfortunately, previous rendering algorithms handling subsurface scattering do not nearly allow interactive image synthesis (see Section 2 for an overview). However, light light particles traveling through an optically dense medium undergo frequent scattering events causing severe blurring of incident illumination. The rendering method proposed in this paper takes advantage of this smoothing property of highly scattering media by factoring the light impulse response on the surface of a translucent object into a high frequency local part and a low frequency global part. We show that the impulse response can be precomputed in a short time, stored compactly and processed sufficiently rapidly in order to allow interactive rendering of translucency effects on rigid objects at interactive rates under dynamic viewing and lighting conditions.

2 Previous Work

The algorithm in this paper draws upon previous work in the areas of global illumination, real time local shading with complex BRDFs and database approaches for reflectance.

Usually in global illumination, one assumes that a scattered light particle leaves a hit surface at the location of incidence itself. The relation between the intensity of light scattered at x into an outgoing direction ω_o and the intensity of incident illumination at x received from a direction ω_i is given by the BRDF (bi-directional reflectance distribution function) $f_r(x, \omega_i, \omega_o)$. However, local light scattering is only a valid assumption for a metal surface or for a smooth boundary between non-scattering media. In other cases, a light particle hitting a surface at a first location x_i from direction ω_i may emerge at a different surface location x_o.

This phenomenon can be simulated with a number of algorithms that have been proposed for global illumination in the presence of participating media, including finite element methods [21, 1, 23], path tracing [6], bi-directional path tracing [15], and photon mapping [10, 4], or by a diffusion simulation [26]. Also, the propagation of electromagnetic radiation in scattering media is a well-studied topic outside of computer graphics in fields such as medical imaging, atmosphere and ocean research and neutron transport [9, 25]. Methods for global illumination are often instances of methods used in these other fields. In optically dense media, such methods can be quite expensive, with typical image rendering times in the range from 10 minutes to several hours for static illumination and viewing parameters.

Non-local light scattering at a surface can however also be modelled explicitly, by means of the BSSRDF (bi-directional subsurface scattering reflectance distribution function) $S(x_i, \omega_i; x_o, \omega_o)$. BSSRDF models have been presented for single [6] and multiple [11] subsurface light scattering in homogeneous materials. These models can be used in a ray tracer, much in the same way as traditional BRDF models. The most prominent difference is that they require two surface locations rather than one. However, these BSSRDF models are much more complex than typical BRDF models (see for instance Figure 2), so that rendering times for common image resolutions still are in the order of seconds to minutes per frame.

Real-time rendering of objects with complex BRDFs has been done with a variety of techniques [2, 8, 12, 13]. These techniques assume point light sources or distant illumination (environment maps) and usually do not allow spatial variation of the BRDF. None of these techniques can be applied to subsurface scattering for translucent objects, since the influence on incident light is not local anymore. Recent work on interactive global illumination of objects [24], including self-shadowing and interreflections, can proba-

$$
\begin{aligned}
S(x_i, \omega_i; x_o, \omega_o) &= \frac{1}{\pi} F_t(\eta, \omega_i) R_d(x_i, x_o) F_t(\eta, \omega_o) \\
R_d(x_i, x_o) &= \frac{\alpha'}{4\pi} \left[z_r (1 + \sigma_{tr} d_r) \frac{e^{-\sigma_{tr} d_r}}{d_r^3} \right. \\
&\qquad \left. + \; z_v (1 + \sigma_{tr} d_v) \frac{e^{-\sigma_{tr} d_v}}{d_v^3} \right] \\
z_r &= 1/\sigma_t' \\
z_v &= z_r + 4AD \\
d_r &= \|x_r - x_o\|, \text{ with } x_r = x_i - z_r \cdot N_i \\
d_v &= \|x_v - x_o\|, \text{ with } x_v = x_i + z_v \cdot N_i \\
A &= \frac{1 + F_{dr}}{1 - F_{dr}} \\
F_{dr} &= -\frac{1.440}{\eta^2} + \frac{0.710}{\eta} + 0.668 + 0.0636\eta \\
D &= 1/3\sigma_t' \\
\sigma_{tr} &= \sqrt{3\sigma_a \sigma_t'} \\
\sigma_t' &= \sigma_a + \sigma_s' \quad ; \quad \alpha' = \sigma_s'/\sigma_t' \\
\sigma_s' &= \text{reduced scattering coefficient (given)} \\
\sigma_a &= \text{absorption coefficient (given)} \\
\eta &= \text{relative refraction index (given)} \\
F_t(\eta, \omega) &= \text{Fresnel transmittance factor} \\
x_i, x_o &= \text{in- and out-scattering location (given)} \\
\omega_i, \omega_o &= \text{in- and out-scattering direction (given)} \\
N_i &= \text{surface normal at } x_i \text{ (given)}
\end{aligned}
$$

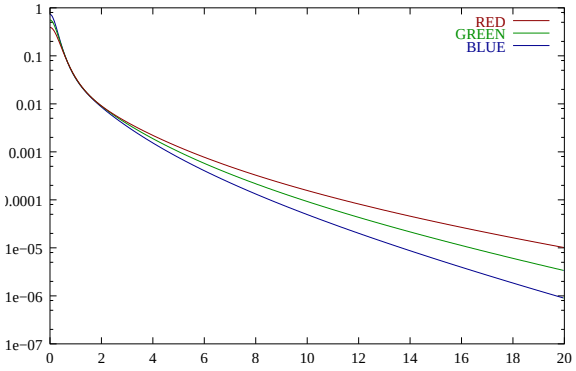

Figure 2. The BSSRDF model (from [11]) used in this paper. Constants for some materials are also found in [11]. The graphs show the diffuse reflectance due to subsurface scattering $\mathbf{R_d}$ for a measured sample of marble with $\sigma_\mathbf{s}'$=(2.19,2.62,3.00)$^{(*)}$, $\sigma_\mathbf{a}$=(0.0021,0.0041,0.0071)$^{(*)}$ and η=1.5. $\mathbf{R_d}(\mathbf{r})$ indicates the radiosity at a distance r in a plane, due to unit incident power at the origin. Subsurface scattering is significant up to a distance of several millimeters in marble. The graphs also explain the strong colour filtering effects observed at larger distances.
$^{(*)}$ RGB colour triplet

bly be extended to subsurface scattering. But this method assumes low-frequency distant illumination, whereas our method allows high-frequency localized illumination.

Image-based techniques like light fields [16, 5] or surface light field [18, 28] represent the appearance of objects such that they can be interactively displayed for different views. The outgoing radiance is recorded and stored in a sort of database which then can be efficiently queried for assembling new views. Light fields can represent the outgoing radiance of an object which exhibits subsurface scattering under fixed illumination. Relighting of the object requires to additionally record the dependency on the incident illumination. Reflection fields [3] parameterize the incident illumination by its direction only. Although different illumination can be simulated by use of different environment maps, it is not possible e.g. to cast a shadow line onto the object. The directional dependency is not sufficient to represent local variation of the illumination on the object's surface.

Our approach for the representation of translucent objects takes the spatial variation of incident illumination into account while the directional dependency is not stored explicitly. This approach will be motivated in the following section.

3 Background and Motivation

In order to compute the shade of a translucent object at a surface point x_o, observed from a direction ω_o, the following integral needs to be solved:

$$L^{\rightarrow}(x_o, \omega_o) = \int_S \int_{\Omega_+(x_i)} L^{\leftarrow}(x_i, \omega_i) S(x_i, \omega_i; x_o, \omega_o) d\omega_i dx_i.$$

S denotes the surface of the object and $\Omega_+(x_i)$ is the hemisphere of directions on the outside of the surface at x_i. Note that the BSSRDF, which represents the outgoing radiance at x_o into direction ω_o due to incident radiance at x_i coming from direction ω_i, is an eight-dimensional function, so that naive precomputation and storage approaches are not feasible in all practical cases.

Previous subsurface scattering studies [6, 11] however reveal that:

- subsurface scattering can be accurately modelled as a sum of a single scattering term and a multiple scattering term;
- single scattering acounts for at most a few percent of the outgoing radiance in materials with high scattering albedo, like marble, milk, skin, etc. . . — we will ignore single scattering in this work;
- multiple scattering diffuses incident illumination: any relation between directions of incidence and exitance is lost.

As a result, subsurface scattering in highly scattering materials can be represented to an accuracy of a few percent by a four-dimensional diffuse subsurface reflectance function $R_d(x_i, x_o)$, which relates scattered radiosity at a point x_o with differential incident flux at x_i:

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

$$B(x_o) = \int_S E(x_i) R_d(x_i, x_o) dx_i \quad (2)$$

$$E(x_i) = \int_{\Omega_+(x_i)} L^{\leftarrow}(x_i, \omega_i) F_t(\eta, \omega_i) \mid N_i \cdot \omega_i \mid d\omega_i \quad (3)$$

The Fresnel transmittance factors F_t indicate what fraction of the flux or radiosity is transmitted at a surface boundary. The Fresnel factor in (3) indicates what fraction of incident light enters the translucent object. In (1), it models what fraction of light coming from underneath re-appears in the environment. The remainder re-enters the object, for instance due to total internal reflection if the object has a higher refraction index than its surrounding. A fast approximation of Fresnel factors has been proposed in [22]. The factor $1/\pi$ in (1) converts radiosity into exitant radiance.

The diffuse sub-surface scattering reflectance R_d in (2) plays a somewhat similar role as the radiosity integral kernel $G(x, y)$ in the radiosity integral Equation

$$B(x) = B^e(x) + \rho(x) \int_S G(x, y) B(y) dy \quad (4)$$

$$G(x, y) = \frac{\mid N_x \cdot \omega_{xy} \mid\mid N_y \cdot \omega_{xy} \mid}{\pi \|x - y\|^2} \mathrm{vis}(x, y).$$

where $B(x)$ denotes the radiosity at x, $B^e(x)$ the self-emitted radiosity, $\rho(x)$ the reflectivity, ω_{xy} the direction of a line connecting x and y and $\mathrm{vis}(x, y)$ is the visibility predicate. Factors like $G(x, y)$ in radiosity and $R_d(x_i, x_o)$ in our case, are usually called (differential) *throughput factors*.

The main idea of this paper is to discretise equation (2), much in the same way as the the radiosity integral equation (4) is discretised in Galerkin radiosity methods [7, 29]. The throughput factors that result from discretisation of the radiosity equation are better known as *form factors*. Note however that form factors in radiosity encode purely geometric information about a scene to be rendered and that they do not directly allow to re-render a scene under dynamic lighting conditions. The subsurface scattering reflectance $R_d(x_i, x_o)$ encodes besides geometric information also the volumetric material properties anywhere in the object relevant for light transport from x_i to x_o: it is the Greens function (impulse-reponse, global reflectance distribution function [14]) of the volumetric rendering equation inside the object. In radiosity, the Greens function does in general not result in practical relighting algorithms due to its high storage cost. The primary goal of this paper is to

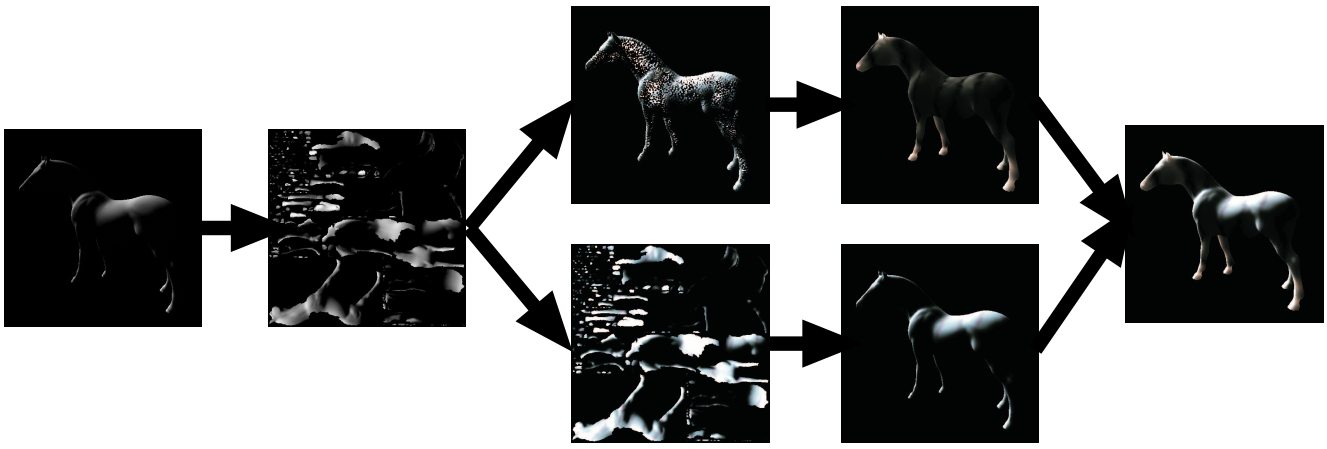

Figure 3. Work flow during rendering: first, incident illumination is computed and projected into the texture atlas. The resulting illumination map is processed in two ways. The global response (upper branch) is computed by projecting the illumination to the mesh vertices and multiplying the vertex irradiance vector with a vertex-to-vertex throughput factor matrix. The local response (lower branch) is computed by filtering the incident illumination map with spatially varying 7×7 texel-to-texel filter kernels. Finally the global and the local response are combined.

demonstrate that explicit representation of the Greens function however *is* practical for dynamic relighting of translucent objects. This is because R_d is in general a much smoother function than the Greens function for radiosity.

In this paper, we will use the diffuse sub-surface scattering reflectance model from [11] (see figure 2). This model has been derived for scattering at a planar boundary surface between homogeneous materials. It is in principle not valid for curved surfaces and neither for heterogeneous materials, although using it in such cases often yields plausible results. This paper however focusses on the feasibility of storing and using the Greens function for interactive rendering of translucent materials. A proper treatment of curved surfaces and heterogeneous materials requires a more sophisticated preprocessing than shown here, and is the topic of future work. However, it does not affect the rendering algorithm itself proposed in this paper.

4 Outline

Our method is based on a discrete version of the integral expression in Equation 2 at which we arrive with a Garlekin type approach. In this approach we employ two different sets of basis functions arriving at two different discretization. One set of basis functions are hat functions placed at object vertices in order to model subsurface scattering at large distances (smooth global part). The other set of basis functions are piecewise constant corresponding to the texels in a texture atlas (disucssed below) in the immediate neighborhood of a point, in order to accurately model R_d at small scattering distances (detailled local part). Each of the two discretizations proceed as follows:

1. We fix a set of spatial basis functions $\psi_i(x)$. The basis functions we use are discussed below;

2. We project the irradiance $E(x)$ (equation (3)) onto the chosen basis: the coefficients E_i in $\tilde{E}(x) = \sum_i E_i \psi_i(x) \approx E(x)$, are found by calculating scalar products of $E(x)$ with dual basis functions $\tilde{\psi}_i(x)$:

$$E_i = \int_S E(x)\tilde{\psi}_i(x)dx \tag{5}$$

   The dual basis functions are the unique set of linear combinations of the primary basis functions $\psi(x)$ that fulfill the following orthonormality relations:

$$\int_S \psi_i(x)\tilde{\psi}_j(x)dx = \delta_{ij}.$$

   δ_{ij} denotes Kroneckers delta function (1 if $i = j$, 0 otherwise);

3. Equation (2) is transformed into a matrix-vector multiplication:

$$B_j = \sum_i E_i F_{ij} \tag{6}$$

   with throughput factors

$$F_{ij} = \int_S \int_S \psi_i(x) R_d(x,y) \tilde{\psi}_j(y) dy dx \tag{7}$$

4. The radiosity $B(y)$ at surface position y is reconstructed as

$$B(y) \approx \sum_j B_j \psi_j(y). \tag{8}$$

Equation (1) shows how radiosity is converted into outgoing radiance for a particular direction.

Two discrete representations of the same problem are of course redundant. We apply such a double representation to exploit the advantages of each, however appropriate blending will be necessary for correct results. The work flow for rendering an image is illustrated in figure 3.

Our method has to address the following sub-problems, which are discussed in detail below:

- Preprocessing: generation of a texture atlas for the input model; computation of throughput factors from each texel to a 9×9 texel neighborhood (detailled local response); computation of weights for distributing the illumination in each texture atlas texel to the nearest triangle vertices as well as for reconstructing the illumination from the nearest triangle vertices and computation of vertex-to-vertex throughput factors (smooth global response); computation of factors for blending the local and global response;
- At rendering time: computation of the irradiance in each texture atlas texel (incident illumination map); distribution of the irradiance in each texel to triangle mesh vertex irradiance and application of the precomputed vertex-to-vertex throughput factors in order to obtain the scattered radiosity at each vertex (global response); convolution of the incident illumination map with the precomputed texture filter kernels (local response); blending of local and global responses using the precomputed blending factors.

5 Preprocessing

The preprocessing phase of the proposed algorithm consists of two steps – the generation of a texture atlas for the input model and the calculation of the local and global light distribution for light hitting an object at a single point (Green's functions).

5.1 Geometry Preprocessing

All rendering results presented here are based on triangle models which are reduced to less then 20000 triangles. During simplification we try to obtain round triangles of similar size.

To obtain a 2D parameterization of the object surface, we generate a texture atlas. The atlas is generated by first splitting the mesh of the model into different triangle chunks and orthographically projecting each chunk onto a suitable plane. The angle between the normals of the projected triangles and the plane normal are kept small to avoid distortion and ensure best sampling rate. Starting with a random triangle, we add an adjacent triangle to a chunk if the deviation of the triangle's normal compared to the average normal of the chunk is below some threshold, e.g. 30 degrees. We also add to each chunk a border formed by adjacent triangles. The width of the border is required to be at least 3 texels to provide sufficient support for applying the 7×7 filter kernels to the original chunk. The border triangles may be distorted in order to fulfill this constraint.

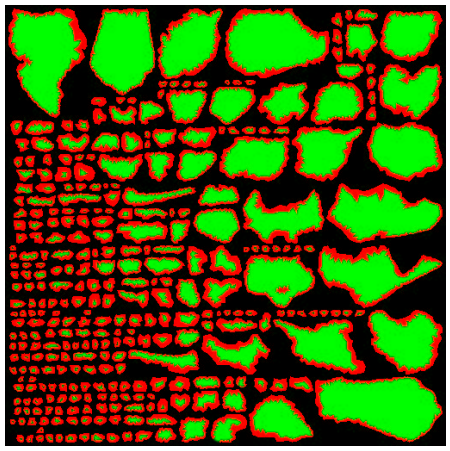

Figure 4. Example of a texture atlas for the bird model. Inner triangles are drawn in green, border triangles are marked in red.

All projected texture chunks are rotated to ensure that the area of their axis-aligned bounding box is minimal [27, 20]. A generic packing algorithm generates a dense packing of the bounding boxes into a square texture of predefined size. The algorithm is able to scale the size of the bounding boxes using a global scaling factor in order to ensure dense packing. Figure 4 shows an example texture atlas for the bird model.

5.2 Global response

Subsurface scattering at larger distances tends to be very smooth and amendable to representation by means of vertex-to-vertex throughput factors using linear interpolation of vertex radiosities.

Linear interpolation of vertex colors is well-known in graphics under the name of Gouraud interpolation. On a triangle mesh, it corresponds to representing a color function by its coefficients w.r.t. the following basis functions:

$$\psi_1^g(x) = \beta_1(x) \quad ; \quad \psi_2^g(x) = \beta_2(x) \quad ; \quad \psi_3^g(x) = \beta_3(x).$$

where $x = \beta_1(x)v_1 + \beta_2(x)v_2 + \beta_3(x)v_3$ with v_1, v_2, v_3 the three vertices of the triangle containing x. The β's are the barycentric coordinates of x in the triangle v_1, v_2, v_3. Note that $\beta_3(x) = 1 - \beta_1(x) - \beta_2(x)$. We associate a basis function ψ_i^g with every vertex i in the triangle mesh. $\psi_i^g(x)$ is zero except for the three vertices in the triangle containing x. When plotted, these basis functions look like "hats" centered at each mesh vertex.

We will need to project the irradiance $E(x)$ to such a basis: $E(x) \approx \sum_i E_i^g \psi_i^g(x)$. The coefficients E_i^g are given by scalar products (5) with the dual basis functions $\tilde{\psi}_i^g$ of the ψ_i^g. These dual functions are also zero except for the three vertices of the triangle containing x. The three non-zero values are

$$\tilde{\psi}_\nu^g(x) = \frac{3}{A_\nu}(4\beta_\nu(x) - 1)$$

where A_ν is the sum of the areas of the triangles sharing vertex ν, ν being a vertex of the triangle containing x.

The throughput factors (7) are approximated in the following way, which requires to evaluate the diffuse subsurface scattering reflectance R_d only once for each pair of mesh vertices (v_i, v_j):

$$F_{ij} \approx R_d(v_i, v_j) \cdot \int_S \psi_i(x)dx \cdot \int_S \tilde{\psi}_j(y)dy = \frac{A_i}{3} R_d(v_i, v_j).$$

The matrix-vector product (6) then results in the scattered radiosity B_j^g at the mesh vertices v_j. The global radiosity $B^g(y)$ for intermediate surface points y is found by linear interpolation: $B^g(y) = \sum_j B_j^g \psi_j(y) = \sum_\nu B_\nu^g \beta_\nu(y)$ where the latter sum is over the three vertices of the triangle containing y.

5.3 Local response

Subsurface scattering reflectance is however quite large at small distances (a range of up to about 2mm for marble, see Figure 2), so that detail in incident illumination such as sharp shadow boundaries will be preserved. For this reason, a more accurate representation will be required for the throughput in the immediate neighborhood of a point where light enters the translucent object. We model this by means of 9×9 texel-to-texel throughput filter kernels centered at each non-empty texel of the texture atlas.

Mathematically, this corresponds with projecting to a piecewise constant basis functions $\psi_{(u,v)}^l$. The basis functions are 1 on the part $S(u, v)$ of the model surface projected in a single texture atlas texel (u, v) and they are 0 everywhere else. There is one such basis function per non-empty texture atlas texel. The dual basis functions in $\tilde{\psi}_{(u,v)}^l$ are piecewise constant in the same way, except that they take a value $1/A(u, v)$ instead of 1 on $S(u, v)$. $A(u, v)$ is the area of $S(u, v)$ and is computed as a side result of texture atlas generation.

By equation (5), the irradiance coefficients $E^l(u, v)$ correspond to the average irradiance on $S(u, v)$. We will approximate them by the value at the center point in the texel. The texel-to-texel throughput factor (7) between texel (u, v) and (s, t) is approximated as $K_{(u,v)}(s, t) = A(u, v)R_d(x_c(u, v), x_c(s, t))$ with R_d being evaluated at the surface points x_c corresponding to the center of the texels. These texel-to-texel throughput factors can be viewed as non-constant irradiance texture filter kernels. Equation (6) then corresponds with a convolution of the irradiance texture. The convolved (blurred) texture shows the locally scattered radiosity $B^l(y)$.

5.4 Blending Local and Global Response

The global and the local response cannot be simply added to obtain the correct result. In the regions of direct illumination, both contributions will add up to approximately twice the correct result. However, the radiosity B^g calculated in §5.2 will have the largest interpolation error near the point of light incidence while B^l (§5.3) returns the more accurate response for points close to direct illumination (see Figure 5). B^l is actually only available for those points. Our choice is to keep the texel accurate filter kernel for the local radiosity since it represents the best local response of our model. Thus, we somehow have to reduce the influence of the low-frequency part at small scattering distances and must ensure smooth blending between the local and global response where the influence of the local response ends.

The global radiosity B^g due to direct illumination corresponds to the diagonal of the form factor matrix $\mathbf{F}$. The diagonal entries are set to zero yielding $\mathbf{F}^0$. To obtain smooth blending we introduce a new radiosity vector B_i^d which is directly derived from the illumination map in a way described below. Using this new radiosity vector, the combined radiosity response will be obtained as

$$\begin{aligned} B(x) &= B^l(x) + B^d(x) + B^{g_0}(x) \\ B_j^{g_0} &= \sum_i E_i^g F_{ij}^0 \end{aligned} \tag{9}$$

For each texel (u, v) of the illumination map, we have to determine its optimal contribution $w_i(u, v)$ to the direct radiosity B_i^d of the three vertices v_i of the enclosing triangle. Our approach is to minimize the difference between the global radiosity and the *correct* radiosity for each texel (s, t) on the boundary Γ of its filter kernel $K_{(u,v)}(s, t)$. The *correct* radiosity at Γ is found by calculating a larger filter kernel $K_{(u,v)}^{9\times9}(s, t)$. Notice that the influence of the 7×7 kernel on Γ of the 9×9 kernel is exactly zero. Stated math-

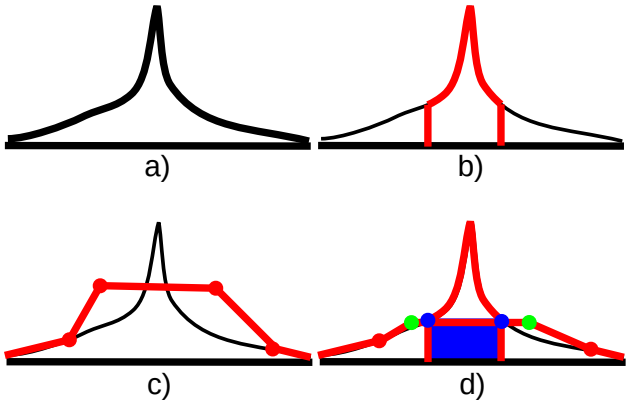

Figure 5. a) Ideal impulse response. b) Local response modeled by the filtering kernel (red) c) Linear interpolation of the global response resulting from distributing the irradiance and evaluating the form factor matrix $\mathbf{F}$. d) Optimized global and local response: The diagonal of $\mathbf{F}^0$ is set to zero, the weights for B_i^d (green dots) are optimized to interpolate the boundary of the filter kernel (blue dots), the blue area is subtracted from the filter kernel.

ematically, the problem is to find $w_\nu(u,v)$ so that

$$\sum_{(s,t)\in\Gamma} \Big[K^{9\times 9}_{(u,v)}(s,t)E(u,v) - B^{g_0}(x_c(s,t)) - E(u,v)\cdot\sum_{\nu=1}^{3}\psi^g_\nu(x(u,v))w_\nu(u,v)\Big]^2$$

is minimal. $x_c(s,t)$ is the surface point corresponding to the center of texel (s,t) and $B^{g_0}(x) = \sum_\nu \psi^g_\nu(x)B^{g_0}_\nu$. The sum is over the vertices ν of the triangle containing x, and $B^{g_0}_\nu$ is given in (9).

After correcting the global response, we also have to change the filter kernels. The interpolated values of the global response have to be subtracted from each kernel, corresponding to the blue area in Figure 5.

This optimization has to be done for every texel. It is performed as a preprocessing step and takes just a few minutes. The irradiance at each texel is now distributed to two different vectors: to E^g_i using the dual basis functions $\tilde{\psi}^g$ and to the B^d_i using the weights w described in this section.

6 Rendering

After preprocessing, the rendering is straightforward. In order to render a translucent object interactively, we first compute an illumination map and then split the computation into two branches. The first one derives the irradiance at each vertex from the illumination map and computes the smooth global response. The second branch evaluates the local response by filtering the illumination map. Both branches can be executed in parallel on a dual processor machine. Finally, the global and the local response are combined.

6.1 Computing the Illumination

For an illuminated object, we need to convert its illumination from object space into texture space since the precomputed filter works in texture space. Furthermore, we have to integrate over the illumination map in order to compute the irradiance at the vertices. For the conversion to texture space we use the parameterization of the object given by the texture atlas.

The illumination map can easily be created by rendering the object by not using its 3D vertex positions but its 2D texture coordinates from the texture atlas. This flattens the object according to the atlas and the result is a texture containing the illumination. Some care has to be taken that the lighting is correctly computed even though the geometry is projected into 2D. We do this by computing the lighting in a vertex shader [17] using the original 3D vertex position and normal. Furthermore we include a Fresnel term in the lighting calculations for which we use Schlick's approximation [22] which can be computed in the vertex shader as well. The rendered illumination map is then read back from the frame buffer and stored. In Figure 3a) and b) the illumination on the object and the corresponding illumination map derived using the texture atlas are shown.

Once the irradiance at each texel (u,v) is computed, we can integrate it to obtain the irradiance for each vertex. In order to distribute the texel irradiance correctly to vertex irradiance, we follow Equation 5. The vertex irradiance E^g_i is given as

$$E^g_i = \sum_{(u,v)} \tilde{\psi}^g_i(u,v)E(u,v)A(u,v), \qquad (10)$$

the sum over all texels in the illumination map times the value at the current texel of the dual basis function corresponding to the vertex, times the area $A(u,v)$ of the model surface $S(u,v)$ covered in the texel. As a result, the illumination at each texel is distributed to exactly three different vertices. The weights $\tilde{\psi}^g_i(u,v)A(u,v)$ are precomputed into an image of the same resolution as the texture atlas.

The same distribution mechanism is also applied to obtain the second radiosity vector B^d_i (§5.4). This time, the weights $w_i(u,v)$ are used instead of the dual basis function. Distributing the illumination map to two vectors instead of just one does not significantly influence rendering performance.

6.2 Low Frequency Reconstruction

Given the irradiance at the vertices, the low frequency or global response is calculated with the throughput factors of Section 5.2. The resulting radiosity B_i^g at the vertices based on the transfer functions matrix $\mathbf{F}$ is then found by

$$B_j^g = \sum_i E_i^g F_{ij}. \tag{11}$$

As previously discussed, the radiosity at a particular point x on a triangle is interpolated using the barycentric basis $\psi_\nu^g(x)$ with respect to the vertices of the triangle.

$$B^g(x) = \sum_{\nu=1}^{3} \psi_\nu^g(x) B_\nu^g \tag{12}$$

Depending on the size of the model and the scattering parameters, the entries in the matrix may drop to very small values. In these cases a full $N \times N$ matrix times vector multiplication may be more costly than ignoring form factors below 10^{-5} using just a sparse matrix. In our experiments the overhead of representing a sparse matrix paid off if 40 percent of the form factors could be ignored.

If desired, surface appearance detail can be added by means of a surfacce texture T_ρ which modulates the radiosity. T_ρ represents the overall reflectance at each texel. Since the form factor matrix $\mathbf{F}$ already computes the radiosity at the vertices correctly, we have to ensure that those values are not changed by the texture. Therefore we devide the vertex radiosity by its corresponding texture value prior to multiplication with the texture:

$$B_i^T = B_i^g / T_\rho(v_i) \tag{13}$$

The complete low-frequency response is then given by

$$B^g(x) = T_\rho(x) \sum_{\nu=1}^{3} \psi_\nu^g(x) B_j^T. \tag{14}$$

6.3 Local Response

According to the factorization described in Section 3, the local response to a light impulse impinging on the surface at point x is represented by a 7×7 filter kernel $K_{(u,v)}(s,t)$ in texture space centered at the corresponding texel (u,v) . Each point or texel may possess a different filter kernel. The resulting radiosity due to local response is thus computed by a kind of convolution for each texel:

$$B^l(x) = K_{(u,v)}(s,t) \otimes E(s,t) = \sum_{(s,t)\in 7\times 7} K_{(u,v)}(s,t) E(s,t) \tag{15}$$

Since the filter kernels for each texel are different, we currently implement this step in software. After the convolution, the filtered illumination map is reloaded as texture and applied during the final composition. In future work, we plan to map also the convolution to graphics hardware. On some systems convolution with a single kernel is already available as an OpenGL extension. One could think of performing a principle component analysis on the kernels, use hardware filtering for principle kernels and finally blend the results. Another way would be using 7×7 weighting textures which are offset by the correct position in the filter kernel and then multiplied with the illumination map, adding up the 49 contributions in the frame buffer.

6.4 Combining Local and Global Response

Local and global response are combined in one hardware-accelerated rendering step using multi-texturing. Register combiners are set up in such a way the the vertex radiosity is multiplied with the surface texture T_ρ and at the same time the filtered illumination map corresponding to the local response is added.

7 Results

We report the performance of our system during tests with four different models (see color page). We perfomed one test with a horse model made of homogeneous white marble (parameters taken from [11]), and a second test on the same horse model but with the white marble augmented by dark veins produced by Perlin noise [19]. We rendered a bust model with skim milk, and we applied a completely synthetic material to a bird model.

The preprocessing time for all models are around 1-2 minutes for calculating the filter kernels and around 2-8 minutes for computing the vertex-to-vertex form factors. The filter kernels are computed for a 512×512 texture atlas. In the computation of the global part only the vertex-to-vertex form factors above the threshold of 10^{-5} are considered. The third column in Table 1 shows how the number of form factors above this threshold.

All models can be rendered and relit at interactive frame rates. Table 1 lists how much of the rendering time is spent on each individual step of the rendering procedure. The timings were collected on a dual 1.7 GHz Xeon PC with 1GB of RAM, using a GeForce3 graphics card. The local and the global response are computed in two parallel threads such that the total time is less than the sum of the individual tasks. Table 1 shows that the number of form factors during the computation of the global response has a major influence on the rendering time.

For each model we currently use a resolution of 512×512 to render the illumination map which is also the resolution of all other maps. In the future we hope to be able to improve the resolution by perfoming the convolution in hardware.

model	# vertices	#form factors	fps	illummap	local	global	display	total
horse	10000	16441460	2.3	29	149	33	371	431
horse textured	10000	12409116	2.7	29	145	302	33	364
bust	8574	4946764	5	24	147	144	28	199
bird	4000	1750862	5.6	16	139	86	26	180

Table 1. *This table lists the number of vertices of each model, the number of relevant vertex-to-vertex form factors, the achieved frame rate and the timings for rendering the illumination map, computing the local response by filtering, distributing the illumination to the vertices, performing vector/matrix multiplication for the global response and the overall time. All timings are given in milliseconds. Note that the total timings are such that all four models can be rendered and relit at interactive frame rates.*

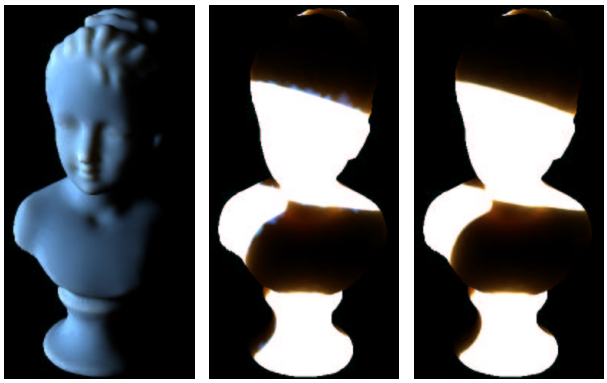

Figure 6. Optimizing the blending: A checkerboard is projected on a bust (left). Triangles in the hair region crossing the border lead to artifacts (middle). The optimized blending yields much better results (right).

Figure 8 displays how the final result is composed of the local and global response. Notice that the color filtering effect for light transport to distant points can be observed at the legs. This effect is actually visible in all models.

The effect of optimizing the blending between the local and global part is demonstrated in Figure 6. Note that the combined result without the optimization is much to bright and exhibits severe artifacts at illumination discontinuities. Those have almost perfectly been removed by the optimization.

8 Conclusion and Future Work

We have developed a system for the interactive rendering of translucent objects. The basic idea is to represent the impulse-reponse as a high frequency local part and a low-frequency global part. The paper shows that storage is feasible, and interactive rendering under dynamic viewing and illumination conditions is achieved.

In the future we would like to show the applicability of the presented method to heterogeneous materials. More complicated algorithms have to be applied during the preprocessing to determine the form factor matrix and the local filter kernels while the rendering procedure will be exactly the same.

It is worth investigating which additional parts of the rendering could be computed directly on the graphics board. This may affect the filtering for the local response, or the computation of the vertex irradiances.

Acknowledgements

We would like to thank Jens Vorsatz, Christian Roessl and Kolja Kähler for providing and simplifying the geometric models.

References

[1] P. Blasi, B. L. Saëc, and C. Schlick. A rendering algorithm for discrete volume density objects. *Computer Graphics Forum*, 12(3):201–210, 1993.

[2] B. Cabral, M. Olano, and P. Nemec. Reflection Space Image Based Rendering. In *Proc. SIGGRAPH*, pages 165–170, August 1999.

[3] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the Reflectance Field of a Human Face. In *Proc. SIGGRAPH*, pages 145–156, July 2000. ISBN 1-58113-208-5.

[4] J. Dorsey, A. Edelman, J. Legakis, H. W. Jensen, and H. K. Pedersen. Modeling and rendering of weathered stone. In *Proceedings of SIGGRAPH 99*, pages 225–234, 1999.

[5] S. Gortler, R. Grzeszczuk, R. Szelinski, and M. Cohen. The Lumigraph. In *Proc. SIGGRAPH*, pages 43–54, Aug. 1996.

[6] P. Hanrahan and W. Krueger. Reflection from layered surfaces due to subsurface scattering. In *Proceedings of SIGGRAPH 93*, pages 165–174, 1993.

[7] P. S. Heckbert and J. Winget. Finite element methods for global illumination. Technical Report UCB/CSD 91/643, Computer Science Division (EECS), University of California, Berkeley, California, USA, July 1991.

[8] W. Heidrich and H. Seidel. Realistic, Hardware-accelerated Shading and Lighting. In *Proc. SIGGRAPH*, pages 171–178, Aug. 1999.

[9] A. Ishimaru. *"Wave Propagation and Scattering in Random Media"*, volume 1. Academic Press, 1978.

[10] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of SIGGRAPH 98*, pages 311–320, 1998.

[11] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. A Practical Model for Subsurface Light Transport. In *Proc. SIGGRAPH*, pages 511–518, August 2001.

[12] J. Kautz and M. D. McCool. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In D. Lischinski and G. W. Larson, editors, *Tenth Eurographics Rendering Workshop 1999*, pages 281–292, June 1999.

[13] J. Kautz and H.-P. Seidel. Towards Interactive Bump Mapping with Anisotropic Shift-Variant BRDFs. In *Eurographics/SIGGRAPH Hardware Workshop*, pages 51–58, August 2000.

[14] E. P. Lafortune and Y. D. Willems. A theoretical framework for physically based rendering. *Computer Graphics Forum*, 13(2):97–107, 1994.

[15] E. P. Lafortune and Y. D. Willems. Rendering participating media with bidirectional path tracing. In *Eurographics Rendering Workshop 1996*, pages 91–100, 1996.

[16] M. Levoy and P. Hanrahan. Light Field Rendering. In *Proc. SIGGRAPH*, pages 31–42, Aug. 1996.

[17] E. Lindholm, M. J. Kilgard, and H. Moreton. A user-programmable vertex engine. In *Proceedings of ACM SIGGRAPH 2001*, pages 149–158, August 2001.

[18] G. Miller, S. Rubin, and D. Ponceleon. Lazy decompression of surface light fields for precomputed global illumination. In *9th Eurographics Workshop on Rendering*, pages 281–292, June 1998.

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

[20] H. Pirzadeh. Computational geometry with the rotating calipers. Master's thesis, School of Computer Science, McGill University, Montreal, Quebec, Canada, November 1999.

[21] H. E. Rushmeier and K. E. Torrance. Extending the radiosity method to include specularly reflecting and translucent materials. *ACM Transactions on Graphics*, 9(1):1–27, 1990.

[22] C. Schlick. An Inexpensive BDRF Model for Physically based Rendering. In *Eurographics '94*, pages 149–162, Sept. 1994.

[23] F. 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.

[24] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. In *Proc. SIGGRAPH*, July 2002 (to appear).

[25] J. Spanier and E. Gelbard. *Monte Carlo principles and neutron transport problems*. Addison-Wesley, 1969.

[26] J. Stam. Multiple scattering as a diffusion process. In *Eurographics Rendering Workshop 1995*, pages 41–50, 1995.

[27] G. T. Toussaint. Solving geometric problems with the rotating calipers. In *Proceedings of IEEE MELECON'83, Athens, Greece*, May 1983.

[28] D. Wood, D. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle. Surface Light Fields for 3D Photography. In *Proc. SIGGRAPH*, pages 287–296, July 2000.

[29] H. R. Zatz. Galerkin radiosity: A higher order solution method for global illumination. In *Proceedings of SIGGRAPH 93*, pages 213–220, 1993.

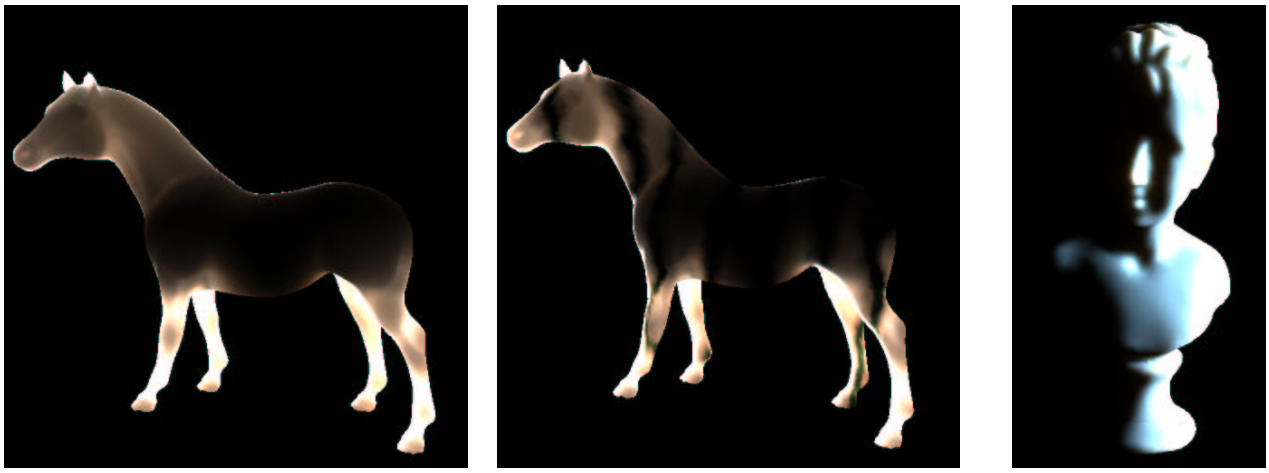

Figure 7. The horse model with uniform marble and with added veins. The structures in the head and leg areas are clearly noticable. The bust on the right is rendered as skim milk.

Figure 8. The textured horse model lit from the top left, parallel to the horse. The left image shows the local response, the middle the global response and the right image the combined result. Notice that different areas are illuminated due to local and global response.

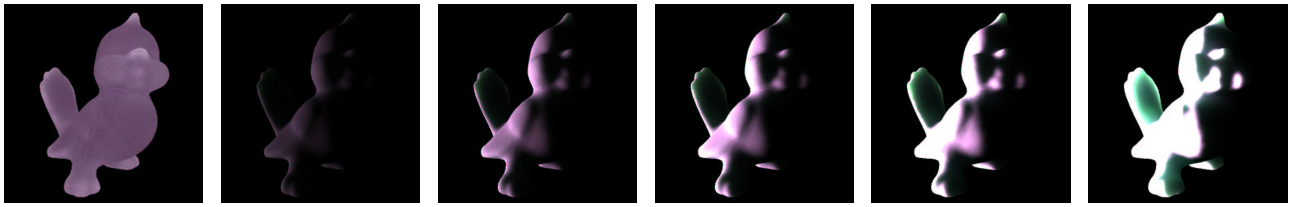

Figure 9. The bird model with an artificial material under uniform illumination (left) and illuminated by a white slide of increasing intensity (left-to-right: 1 uniform, 1, 3, 5, 10, 50). Note the color shift from magenta to green.