Eurographics Symposium on Rendering (2007)
Jan Kautz and Sumanta Pattanaik (Editors)

# Efficient Rendering of Human Skin

Eugene d'Eon, David Luebke, and Eric Enderton[†]

NVIDIA Corporation

**Abstract**

*Existing offline techniques for modeling subsurface scattering effects in multi-layered translucent materials such as human skin achieve remarkable realism, but require seconds or minutes to generate an image. We demonstrate rendering of multi-layer skin that achieves similar visual quality but runs orders of magnitude faster. We show that sums of Gaussians provide an accurate approximation of translucent layer diffusion profiles, and use this observation to build a novel skin rendering algorithm based on texture space diffusion and translucent shadow maps. Our technique requires a parameterized model but does not otherwise rely on any precomputed information, and thus extends trivially to animated or deforming models. We achieve about 30 frames per second for realistic real-time rendering of deformable human skin under dynamic lighting.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

## 1. Introduction

Accurate rendering of many real-world objects requires modeling subsurface scattering effects to capture translucent appearance. Examples of translucent materials range from milk and ketchup to jade, marble, and plastic. Many important materials consist of multiple translucent layers – notably many organic materials such as plant leaves and skin. Human skin in particular presents a challenging but crucially important rendering problem for photo-realistic graphics. By approximating the physiological layers of skin – epidermis, dermis, and so on – as locally homogeneous thin slabs, researchers have achieved remarkably realistic renderings [DJ05, DJ06]. However, this realism comes at a cost: today's most accurate simulations of multi-layer translucent materials typically require seconds or even minutes to render.

We present a novel and extremely efficient formulation of the multipole technique by Donner and Jensen [DJ05] for light transport through multi-layer translucent materials. Our technique enables real-time rendering of such materials and requires no precomputation. The key idea is to approximate diffusion profiles of thin homogeneous slabs as a linear combination of carefully chosen Gaussian basis functions. This representation greatly accelerates the computation of multi-layer profiles and enables improved algorithms for texture-space diffusion and global scattering via translucent shadow maps. We focus here on the specific application of these ideas to the rendering of human skin.

Most real-time methods capable of rendering translucent materials rely on precomputing light transport among points on the surface. This very general approach can capture a broad range of illumination effects, but requires fixing the geometry of the model at precomputation time (a notable exception: the zonal harmonics approach of Sloan et al. [SLS05] enables limited local deformations). Real-time rendering of translucent objects with no precomputation, necessary for scenes with fully general animation or deformation, has received much less attention. We improve and combine two such approaches: texture-space diffusion [BL03, Gre04], which provides an efficient estimate of local scattering, and translucent shadow maps [DS03], which approximates scattering through thin regions such as ears (Figure 6).

Specifically, we propose several extensions of texture-space diffusion to rapidly and hierarchically evaluate light diffusion within arbitrary deformable manifolds. The use of separable Gaussian kernels accelerates convolution of sur-

[†] {edeon, dluebke, eenderton}@nvidia.com

**Figure 1:** *We project multi-layer diffusion profiles onto a sum-of-Gaussians basis to enable realistic rendering of human skin at 30 frames per second on a modern GPU. From left to right: Albedo (1st) and irradiance (2nd) combine to give subsurface irradiance which is then convolved with each Gaussian basis profile (3rd through 7th) and combined in a final render pass with specular (8th) to produce the final image (9th). Convolutions are performed in off-screen 2D textures but shown here mapped onto the face.*

face irradiance. We use multi-scale texture-space stretching, computed on the fly using per-pixel derivative instructions, to independently correct these convolutions at each level in the hierarchy; this correction eliminates the need for precomputation and affords a highly accurate approximation of diffusion even under considerable deformation.

We then combine translucent shadow maps with texture space diffusion by rendering the depth and UV coordinate of the topmost surface in light space (instead of rendering surface normal and irradiance [DS03]). While convolving irradiance over the surface, we simultaneously convolve depth through the object (stored in the alpha channel) to better estimate the average depth, and exploit the separability of Gaussians again to quickly compute global scattering. Since the entire algorithm is run each frame, our final system can render deformable models, such as the human head shown in Figure 8, with environment lighting and one or more translucent shadow maps at real-time rates.

## 2. Previous Work

The topic of rendering translucent materials has attracted a great deal of attention in recent years, especially since the seminal work of Jensen et al. [JMLH01, JB02] first made subsurface scattering practical in many rendering environments. A particularly rich vein of real-time rendering research has built on precomputed radiance transfer by Sloan et al. [SKS02], which samples light transport on the surface in the domain of illumination and projects the resulting transport vectors onto a low-dimensional frequency-space basis. These approaches are capable of capturing translucent appearance but involve significant precomputation steps that fix the geometry and thus do not permit general animation or deformation of the rendered object.

### 2.1. Scattering in Multi-layered Materials

Donner and Jensen [DJ05] show that accurate rendering of skin requires modeling multi-layered subsurface scattering. They use a three layer skin model with separate properties in each layer and compute combined scattering profiles. This profile computation, discussed further in Section 3, takes several seconds. They render final images in about five minutes using a Monte Carlo renderer. Motivated by the realism of their results and the desire to achieve faster rendering times, we developed a simple but effective mathematical approximation that enables new algorithms for efficiently rendering multi-layer translucent materials. We demonstrate that this approximation is quite accurate for rendering materials like skin (for which single scattering is often ignored and surfaces with extreme curvature are not typically encountered).

### 2.2. Real-Time Subsurface Scattering

A large body of work has developed real-time rendering methods for translucent objects, but most techniques incur significant preprocessing costs by incorporating precomputed light transport (e.g., [SKS02, HBV03, HV04, WTL05, WWD*05]) or building geometry-specific textures (e.g., [CHH03]). We concentrate on methods that avoid offline precomputation and thus allow arbitrary manipulations such as animated or deforming geometry.

Borshukov and Lewis [BL03] present an inexpensive empirical approach to computing diffuse scattering in human skin. They rasterize diffuse irradiance into a 2D texture map that serves as a parameterized domain over the surface. They then approximate subsurface diffusion by convolving this irradiance in image space using a "rapid falloff" ker-

nel whose width varies per color channel to simulate the frequency-dependent mean free path for different spectral bands. Green [Gre04] adopts this technique for real-time rendering with a GPU implementation that exploits the texturing capabilities of current graphics hardware and performs convolution via several separable blur passes. Gosselin et al. [GSM04] use a similar technique for rendering skin, including a hand-painted scalar map controlling convolution over the surface to correct for stretching in the parameterization or increase diffusion in thin areas such as ears.

The above techniques approximate subsurface scattering with ad hoc parameters not directly based on the properties of a single- or multi-layered translucent material. The real-time techniques use a single Gaussian kernel, which does not model physically-based diffusion through real materials and leads to an unrealistic, waxy look. The rapid-falloff kernel [BL03] improves on this but can not be evaluated separably and relies on an artist to adjust the kernel parameters by eye. Still, the results improve substantially over renderings with no subsurface effect at all, and the technique lends itself to implementation on modern GPUs, where texture convolutions can be carried out efficiently each frame to eliminate the need for a precomputation step. We build heavily on this *texture-space diffusion* approach, using our sum-of-Gaussians framework to improve the diffusion accuracy and to extend the approach to the multi-layer model [DJ05].

Dachsbacher and Stamminger [DS03] describe *translucent shadow maps*, which extend traditional shadow maps to render global scattering effects. At each shadow map pixel, which represents a sampled point on the surface facing the light, they capture not only depth (and hence 3D position) but also irradiance and surface normal. Irradiance is convolved in light space using texture mip-mapping hardware, and rendered pixels of shadowed surfaces can integrate illumination at the appropriate scale by looking up into the preconvolved light-space texture. Section 5 describes how we extend and improve translucent shadow maps to achieve global scattering effects not captured by the texture-space diffusion approach.

Mertens et al. [MKB*05] present a promising approach similar in spirit to translucent shadow maps. Instead of rendering irradiance in light space and filtering the resulting texture using mipmapping, they render an irradiance texture from the camera view and perform importance sampling rather than explicitly filtering the irradiance. They achieve interactive frame rates when accelerated by graphics hardware, but their approach has some limitations. The image-space importance sampling can create noise and does not account for information which is not visible in a given frame. Their system also misses global scatter through thin regions, and requires a large number of render passes to accurately capture the diffusion of complex layered materials like skin.

## 3. Background

### 3.1. Dipole Scattering

Light transport in translucent materials is governed by the Bidirectional Scattering Surface Reflectance Distribution Function (BSSRDF). The BSSRDF, $S$, gives the proportion of light incident at position $x_i$ from direction $\vec{\omega}_i$ that radiates out from position $x_o$ in direction $\vec{\omega}_o$. Total outgoing radiance $L_o$ is then

$$L_o(x_o,\vec{\omega}_o) = \int_A \int_{2\pi} S(x_i,\vec{\omega}_i;x_o,\vec{\omega}_o) L_i(x_i,\vec{\omega}_i)(\vec{n}\cdot\vec{\omega}_i) d\omega_i dA(x_i). \tag{1}$$

Jensen et al. [JMLH01] introduce a dipole diffusion approximation allowing efficient simulation of highly scattering materials. This reduces $S$ to depend only on the scattering properties of the material, the Fresnel terms at $x_o$ and $x_i$, and the distance between $x_o$ and $x_i$,

$$S_d(x_i,\vec{\omega}_i;x_o,\vec{\omega}_o) = \frac{1}{\pi} F_t(x_i,\vec{\omega}_i) R(||x_i - x_o||_2) F_t(x_o,\vec{\omega}_o), \tag{2}$$

where $F_t$ is the Fresnel transmittance and $R(r)$ is the diffusion profile of the material.

Using dipoles, Jensen et al. derive simple analytic diffusion profiles assuming a flat, homogeneous, semi-infinite dielectric material. Directly applying the same profiles to non-flat surfaces works well, in most cases.

### 3.2. Multi-layered materials

Recent work by Donner and Jensen [DJ05] extends diffusion theory to account for multiple thin layers (removing the semi-infinite requirement). This is essential for accurate rendering of many natural materials, such as skin.

For a single thin planar slab within a multi-layered material, reflectance profiles $R(r)$ and transmittance profiles $T(r)$ are computed using a multipole (a sum of a number of dipoles). These profiles describe the reflected and transmitted response to an infinitesimal focused beam of light illuminating the slab. Different profiles are computed for each wavelength of light being treated. Rough surfaces may also be accounted for by replacing the Fresnel terms with a diffuse transmission function, $\rho_{dt}$ (see [DJ05]).

Donner and Jensen then show how to compute the reflectance and transmittance profiles for two slabs placed together by analyzing the convolution of one slab's diffusion profiles by the other's as light bounces between them. Since (1) with (2) is a 2D surface convolution of irradiance (scaled by Fresnel terms) with a radially symmetric convolution kernel $R(r)$, associativity of convolution allows precomputation of the net kernel that describes the diffusion by the combined slabs. For instance, the net diffusion profile of direct transmission through two slabs without inter-slab reflection is the convolution $T_1^+(r) * T_2^+(r)$ where $T_1^+(r)$ and $T_2^+(r)$ are the forward transmittance profiles for the two slabs. Accounting

for inter-slab reflections results in

$$T_{12}^{+} = T_1^{+} * T_2^{+} + T_1^{+} * R_2^{+} * R_1^{-} * T_2^{+} + T_1^{+} * R_2^{+} * R_1^{-} * R_2^{+} * R_1^{-} * T_2^{+} + \ldots. \quad (3)$$

Here $+$ and $-$ superscripts denote reflectance and transmittance profiles in a given direction. For instance, the second term in (3) accounts for light transmitting through slab 1, reflecting off slab 2, then reflecting off slab 1 going backward$(-)$, and then transmitting through slab 2 going forward$(+)$. In the case of varying indices of refraction among slabs, and in the case of 3 or more slabs, $R_i^{+}(r) \neq R_i^{-}(r)$ in general, and care must be taken to distinguish between them (similarly for $T(r)$ profiles). Donner and Jensen transform each 2D radial profile to frequency space where convolutions become multiplications

$$\begin{aligned}\mathcal{T}_{12}^{+} &= \mathcal{T}_1^{+}\mathcal{T}_2^{+} + \mathcal{T}_1^{+}\mathcal{R}_2^{+}\mathcal{R}_1^{-}\mathcal{T}_2^{+} + \mathcal{T}_1^{+}\mathcal{R}_2^{+}\mathcal{R}_1^{-}\mathcal{R}_2^{+}\mathcal{R}_1^{-}\mathcal{T}_2^{+}\ldots \\ &= \mathcal{T}_1^{+}\mathcal{T}_2^{+}(1 + (\mathcal{R}_2^{+}\mathcal{R}_1^{-}) + (\mathcal{R}_2^{+}\mathcal{R}_1^{-})^2 + (\mathcal{R}_2^{+}\mathcal{R}_1^{-})^3 + \ldots) \quad (4)\end{aligned}$$

and they note that the geometric series can be replaced, provided $\mathcal{R}_2^{+}(r)\mathcal{R}_1^{-}(r) < 1$ for all $r$, leaving the frequency-space Kubelka-Munk equations,

$$\mathcal{T}_{12}^{+} = \frac{\mathcal{T}_1^{+}\mathcal{T}_2^{+}}{1 - (\mathcal{R}_2^{+}\mathcal{R}_1^{-})} \quad (5)$$

with similar derivations for $\mathcal{T}_{12}^{-}(r)$, $\mathcal{R}_{12}^{+}(r)$, and $\mathcal{R}_{12}^{-}(r)$.

These four formulae allow combination of any number of slabs (plus an optional semi-infinite bottom layer), treating two at a time. For each layer in a material model, four radial profiles ($T^{+,-}(r)$,$R^{+,-}(r)$) are computed using multipoles. Four 2D FFTs (or four 1D Hankel transforms) transform these profiles (applied radially in two dimensions) to frequency space, and then four formulae $\mathcal{R}_{12}, \mathcal{T}_{12}{}^{+,-}(r)$ are applied to compute four frequency-space profiles of the combined slabs. This is repeated recursively for additional slabs. Finally, two inverse FFTs produce the reflectance and transmittance profiles used for rendering. For a small number of slabs, this process takes a few seconds to compute on a modern CPU. We show how to accelerate this in Section 4.

## 4. Fast Approximate Diffusion Profiles

Our key observation is that dipoles and multipoles are approximated well with sums of a small number of Gaussians. Four Gaussians fit most single slab profiles extremely well and more can be used to increase accuracy.

To fit Gaussians to a diffusion profile $R(r)$, we minimize

$$\int_0^{\infty} r\left(R(r) - \sum_{i=1}^{k} w_i G(v_i, r)\right)^2 dr \quad (6)$$

where both the weights $w_i$ and the variances $v_i$ for $k$ Gaussians are allowed to vary, and the Gaussian of variance $v$ is

$$G(v,r) \coloneqq \frac{1}{2\pi v} e^{-r^2/2v}. \quad (7)$$

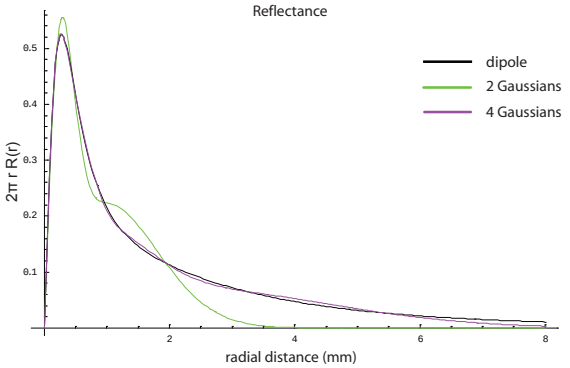


**Figure 2:** *Approximating a dipole with a sum of 2 and 4 Gaussians (for the green wavelength of marble [Jensen et al. 2001]). Profiles are plotted scaled by $2\pi r$, since any error in approximating the dipole is applied radially. Rendering with the 4 Gaussian sum is visually indistinguishable from using the dipole.*

The constant $1/2\pi v$ is chosen such that all Gaussians have unit total diffuse response

$$\int_0^{\infty} 2\pi r G(v,r) dr = 1. \quad (8)$$

Total diffuse reflectance can be matched exactly by restricting the sum of the weights, $w_i$, to be the total predicted by the multipole,

$$\sum_{i=1}^{k} w_i = R_d \coloneqq \int_0^{\infty} 2\pi r R(r) dr. \quad (9)$$

We weight error terms in (6) by $r$ because profiles are applied radially, each value combining light from a circle of circumference proportional to $r$. When rendering, we convolve irradiance with a sum of Gaussians $G_{sum}(r)$ rather than with $R(r)$. If the total error in (6) is small in comparison to the total radial integral of $R(r)$, then the visual error in our approximation will be undetectable in all but a few contrived lighting scenarios. Thus, we propose an error metric of relative RMS power,

$$\frac{\sqrt{\int_0^{\infty} r(R(r) - G_{sum}(r))^2 dr}}{\sqrt{\int_0^{\infty} r(R(r))^2 dr}}. \quad (10)$$

For every set of scattering coefficients given in [JMLH01], each profile was approximated with four Gaussians by using Levenberg-Marquardt optimization to minimize (6), and (10) was computed. The errors in approximation ranged from 1.52% for the blue wavelength of spectralon, to 0.0793% for the blue wavelength of Chicken2. Figure 2 shows how two and four Gaussians approximate a dipole profile for the green wavelength of marble [JMLH01]. Four Gaussians fit the dipole with an error of 1.25% and we found

rendering with the dipole versus the four Gaussian sum to be indistinguishable. Eight Gaussians fit the same dipole with an error of 0.093%.

Sums of Gaussians are advantageous for three main reasons. First, since Gaussians are separable they can be applied as two 1D convolutions in x and then y over a surface. This allows a very fast estimate of the surface irradiance convolution in (1), provided the surface is roughly planar (already assumed in the derivation of the dipole and multipole theories). The convolution by the non-separable kernel $R(r)$ is quickly and accurately estimated as the sum of convolutions by each separable Gaussian term. Second, convolution by a wider Gaussian can be computed from the result of a previous convolution by a narrower Gaussian, considerably faster than from the original irradiance values.

Third, once a dipole or multipole profile is given as a sum of Gaussians, the convolutions required to combine two slabs (3) become very inexpensive, requiring no Fourier transforms. The radial 2D convolution of any two Gaussians is another Gaussian

$$G(v_1) * G(v_2) = G(v_1 + v_2) \tag{11}$$

The 2D convolution of two radial profiles, each of which is approximated as a sum of Gaussians, is

$$\sum_{i=1}^{k_1} w_i G(v_i, r) * \sum_{j=1}^{k_2} w'_j G(v'_j, r) = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} w_i w'_j G(v_i, r) * G(v'_j, r). \tag{12}$$

The result is a new sum of $k_1 k_2$ Gaussians. Although Eq. (3) contains an infinity of convolutions and additions, in practice, the summation of powers of $R_2^+ R_1^-$ can be terminated quite quickly with negligible error. Tracking the total diffuse response of the summation and comparing to the value predicted by the Kubelka-Munk equations allows early termination. For example, in computing the profile for combined transmission by two slabs (3), we find $n$ s.t.

$$\frac{T_{1d}^+ T_{2d}^+}{1 - (R_{2d}^+ R_{1d}^-)} - T_{1d}^+ T_{2d}^+ (\sum_{i=0}^{n} (R_{2d}^+ R_{1d}^-)^i) < \varepsilon \tag{13}$$

where the $d$ subscript refers to total diffuse response of each profile, and $\varepsilon$ is the maximum tolerated energy loss in terminating the infinite series of interslab reflections at $n$. The total diffuse reponse of any profile represented as a sum of Gaussians is simply the sum of the weights.

Equation (12) roughly squares the number of Gaussian terms after each multiply, causing the number of operations to accumulate quite quickly. However, fitting all initial slab profiles to powers of a single Gaussian of narrow variance $v$ causes any two Gaussians to convolve back into the same set. That is, we approximate each $R(r)$ and $T(r)$ profile as a linear combination of

$$\{G(v), G(v) * G(v), G(v) * G(v) * G(v), ...\} \tag{14}$$

which by (11) is equivalent to

$$\{G(v), G(2v), G(3v), ...\}. \tag{15}$$

Convolutions and additions of scattering profiles become analogous to polynomial multiplications and additions over Gaussians. Thus, we have reduced a lengthy process involving 2D texture additions, multiplications, divisions, FFTs and inverse FFTs to no more than a small number of polynomial operations. While we focus on real-time rendering applications in this paper, we note the technique could be used for efficient offline rendering of spatially varying multi-layered materials, accelerating what was a several-second profile computation cost per pixel by orders of magnitude with negligible error.

We initially used Levenberg-Marquardt optimization to solve (6), but later found closed formulae that, empirically, give a close fit to any dipole over a wide range of material parameters, using 8 Gaussians. See Appendix A for details.

## 5. Rendering

We use the sum-of-Gaussians formulation of diffuse scattering profiles to enable real-time rendering of multi-layer translucent materials, specifically human skin, under dynamic all-frequency lighting. In this section we describe our extensions to the texture-space diffusion and translucent shadow map algorithms discussed in Section 2. We do require a parameterized mesh. For topologically simple models such as human faces and bodies, an artist can easily create such a parameterization with existing tools. Since realistic rendering and animation of human characters often require a parameterized surface anyway, e.g. to store normal maps or ambient occlusion, we do not consider this overly restrictive for our application. We discuss potential strategies to deal with texture charts in Section 7. As is common in recent skin rendering systems [DJ05, DJ06, WMP*06], we assume single-scattering is negligible.

### 5.1. Extending texture-space diffusion

Texture-space diffusion performs the irradiance convolution in (1) by rasterizing irradiance into a texture, computing image filtering operations on that texture, and texture mapping the resulting image back onto the 3D mesh. By expressing the non-separable diffusion profile as a sum of Gaussians, we can separably and hierarchically evaluate the diffusion of irradiance much more efficiently than directly evaluating the 2D convolution. This produces a series of convolution textures whose weighted sum approximates the convolution of irradiance by the original non-separable diffusion profile. Figure 1 illustrates the individual textures generated to render a face and the final image that is created from them.

We first rasterize irradiance into an off-screen texture. Irradiance is attenuated by a Fresnel term for each light (or for rough surfaces, by $\rho_{dt}$, see [DJ05]). Following

Green [Gre04], we use a vertex shader that moves each mesh vertex to its texture coordinates and a fragment shader that computes lighting and Fresnel terms for each light source.

We fit a sum of Gaussians to the skin diffusion profiles, as described in Section 4. This is done once per material, or once per frame if the skin model is to be adjusted interactively by the user. One convolution of the irradiance texture is computed for each Gaussian used in the profile fit. Each convolution is computed separably in two passes, using a temporary buffer for intermediate results. All convolution textures are retained for use in the final render pass.

We use a seven-tap-gather Gaussian kernel with variance $v = 1$ for all convolution passes. Different Gaussians are computed by linearly scaling the spacing of the seven taps about the center tap. This assumes that the first Gaussian computed is narrow enough so that this discrete sampling does not significantly undersample the irradiance. The minimum mean free path of all slabs in the material, $\ell_{min}$, is an appropriate threshold [JB02] and thus the first Gaussian used to fit the diffusion profiles should have a variance $v_1$ no greater than $\ell_{min}^2$. Each successive Gaussian should have variance $v_i$ no more than $4v_{i-1}$ (standard deviation no more than doubles), to avoid undersampling when the same 7-tap filter is used with a wider spacing. We have found that the sets of Gaussians obtained by approximating to dipoles and multipoles are well within this spacing requirement, and using a 7-tap filter at all stages does not introduce any noticeable aliasing artifacts. We found six Gaussians sufficient to capture the appearance of a three-layer skin model. The potentially dense sets of Gaussians obtained from the fast analytic fit and polynomial convolution operations can be reduced for real-time rendering by approximating each Gaussian term as a linear combination of the two nearest Gaussians in a sparser set.

Any texture parameterization of a non-trivial mesh will exhibit texture distortion, or *stretch*. Since diffusion between two points on the surface should depend on their Euclidean distance, significant stretch in the parameterization can distort the irradiance diffusion computation (Figure 3). We correct for this effect by computing texture stretch dynamically each frame and using the measured stretch to modulate the filter support at each pixel during the Gaussian blur convolution steps to better approximate diffusion across the surface (Figure 4). Stretch information directly scales the spread of the separable U,V filter samples in texture space. To keep small-scale features from adversely affecting large-scale convolution, we convolve stretch simultaneously with irradiance (in a separate 2-channel texture). This multi-scale set of stretch values allows each Gaussian convolution to consider a local average of stretching over an appropriate width. Strictly speaking, filtering is not separable in regions where stretch varies, nor in regions where the U and V directions are non-orthogonal, but we have not seen visual artifacts from this.

**Figure 3:** *Here we project a regular pattern of illumination onto a face model and diffuse it by convolving with several Gaussian kernels (top and bottom). Performing this convolution in texture space exhibits distortions due to texture stretch (left). Using a stretch-correction texture to locally correct kernel widths greatly reduces the distortion (right). The remaining distortion proves visually negligible when rendering a textured, illuminated skin model.*

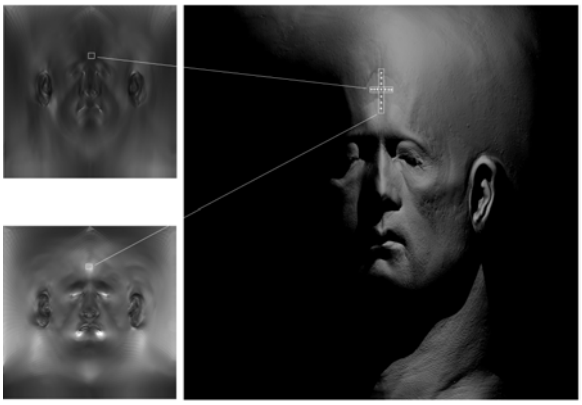

**Figure 4:** ***Top Left:*** *Horizontal Stretch correction texture* ***Bottom Left:*** *Vertical Stretch correction texture.* ***Right:*** *The stretch correction values linearly scale the separable convolution kernels for each location in texture space.*

Figure 3 compares convolution of a uniform pattern of irradiance projected over a face, with and without texture-space stretch correction. The following simple fragment shader computes directional stretch in U and V at each location in texture space.

```
float2 computeStretchMap( float3 worldCoord : TEXCOORD0 )
{
  float3 derivu = ddx( worldCoord );
  float3 derivv = ddy( worldCoord );
  float stretchU = 1.0 / length( derivu );
  float stretchV = 1.0 / length( derivv );
  return float2( stretchU, stretchV );
}
```

### 5.2. Extending Translucent Shadow Maps

Texture-space diffusion captures highly diffuse local scattering typical in skin, but can miss global scattering between regions that are close in Euclidean space but far in texture space, such as the thin parts of the ear (Figure 6). We modify translucent shadow maps (TSMs) [DS03] to account for such regions. TSMs render the depth, irradiance, and surface normal, storing these quantities for the surface nearest the light (the "light facing surface") at every pixel of the shadow map. We instead render and store the (u,v) coordinates and depth of the light facing surface. This allows every shadowed surface location to compute distance through the object toward the light, and to access convolved versions of irradiance on the light facing surface by looking up into the same irradiance textures used for local scattering.

Figure 5 illustrates this process in more detail. At any shadowed location C, the TSM provides the distance $m$ and the UV coordinates of point A on the light-facing surface. We want to estimate the scattered light exiting at C, which is the convolution of irradiance at each light-facing point by the profile $R$ through the thickness of the object. We instead compute this convolution at B, since we can do this very efficiently. (For low angles $\theta$, B will be close to C, and for high angles, the Fresnel term will reduce the contribution in any case.) The convolution kernel is

$$R(\sqrt{r^2+d^2}) = \sum_{i=1}^{k} w_i G(v_i, \sqrt{r^2+d^2}) = \sum_{i=1}^{k} w_i e^{-d^2/v_i} G(v_i, r) \tag{16}$$

using $d = m\cos(\theta)$ as the thickness of the object. Because the Gaussians are separable in the third dimension as well, global transmittance is the weighted sum of $k$ texture lookups, each weighted by $w_i e^{-d^2/v_i}$, where $i$ is the index corresponding to the irradiance texture convolved by $G(v_i, r)$. A single 2D convolution of irradiance on the light facing surface with $R(r)$ could not be re-used to compute transmittance in this fashion because $R$ is not a separable kernel, but by expressing $R(r)$ as a sum of Gaussians, the convolution textures can be used for both local scattering at A and global transmittance at B, and their weighted sum accurately matches convolution by the original profile.

Depths, $m$, computed by the shadow map are corrected by $cos(\theta)$, since a diffusion approximation is being used and the most direct thickness is more applicable. Surface normals at A and C are compared and the distance correction is only applied when the surfaces are oppositely facing. We interpolate between the two using: $lerp(m, mcos(\theta), max(0, -N_A \cdot N_C))$.

Note that the derivation illustrated in Figure 5 assumes a planar surface with constant thickness $d$. This is similar to the dipole and multipole theories, which were derived in a plane-parallel setting but are directly applied to curved surfaces. Of course, this assumption does not hold for 3D models in general, but we have found (like the dipole and multipole theories) that directly applying the technique to curved surfaces works well in practice.

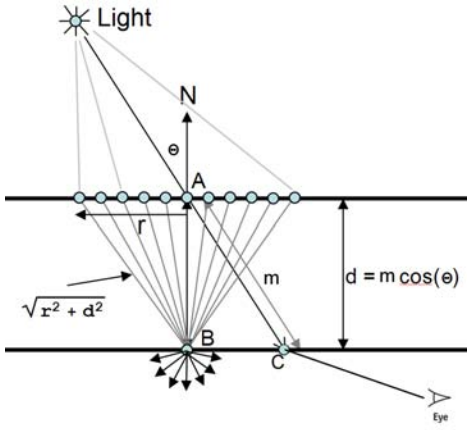


**Figure 5:** *Global scattering through thin regions is computing using a modified translucent shadow map (TSM). The (u,v) coordinate stored in the TSM allows shadowed regions, C, to access the convolved irradiance textures at the illuminated point A on the light-facing surface. Depth through the surface m can be corrected with a cosine term.*

High-frequency changes in depth through the surface can create unwanted high-frequency artifacts in the final image. To mitigate these effects, we convolve depth simultaneously with irradiance by storing the depth in the alpha channel before convolving the irradiance textures. Computation of forward transmittance for each Gaussian $i$ then uses the convolved depth from irradiance texture $i-1$. The intuitive justification is that convolving depth accounts for a number of paths through the surface when computing $d$, and the wider Gaussians consider a wider average of $d$.

To avoid double contribution from both local and global scattering, the global scattering terms are interpolated to 0 as the (u,v) coordinate of the point being rendered, C, approaches the (u,v) coordinate of the light facing surface at A (a separate interpolation is used for each Gaussian starting when the two locations are a distance $6\sqrt{v_i}$ apart).

### 5.3. Texturing

Following Weyrich et al. [WMP*06], we treat the diffuse color map as an infinitesimal, highly absorptive layer that absorbs light once as it enters the surface and once as it leaves. We render with the diffusion profile $R(r) = \sum_{i=1}^{k} w_i G(v_i, r)$ where $w_i$ are spectral weights (we render in RGB, so each $w_i$ is a triple). The $w_i$ are re-normalized to white so that the diffuse color map provides the final skin tone. Two absorptions of $\sqrt{diffuseColor}$ give the desired skin tone.

**Figure 6:** *Previous texture-space diffusion techniques do not capture global scattering through thin regions like ears (Left). Rendering with our modification to translucent shadow maps (Center) creates much of the same look as achieved using Monte Carlo rendering techniques(Right). The two real-time images (Left and Center, 19fps) are computed using the spectral BSSRDF model and parameters listed in [DJ06]. Extra softness is visible in the right image from use of an area light source. The right image is courtesy of Craig Donner and Henrik Wann Jensen.*

### 5.4. Summary of rendering algorithm

We perform the following passes each frame (where textures are in **bold**):

Render irradiance texture:
  Render Transluscent Shadow Map (TSM)
  Render any other shadow maps
  **Irrad** = sum of diffuse light from point, spot, environment lights
  **Irrad** *= Fresnel term $\rho_{dt}(x_i)$
  **Irrad** *= sqrt(**diffuse color**)
  **Irrad**.alpha = distance through surface, from the TSM
Render Stretch texture:
  Simple fragment program with derivative instructions
  **Stretch** = two channel texture of U stretch, V stretch
// *Gaussian basis is G($v_1$), G($v_2$), ..., G($v_k$), $v_0$=0*
For i = 1 to k, compute **Irrad** convolved by G($v_i$):
  Blur **Stretch** by $v_i - v_{i-1}$ in U direction, compensated by **Stretch**.u
  Blur **Irrad** by $v_i - v_{i-1}$ in U direction, compensated by **Stretch**.u
  Blur **Stretch** by $v_i - v_{i-1}$ in V direction, compensated by **Stretch**.v
  Blur **Irrad** by $v_i - v_{i-1}$ in V direction, compensated by **Stretch**.v
  // *Now **Stretch** and **Irrad** are blurred by $v_i$ in both U and V*
  **IrradBasis**[i] = **Irrad**
Render final image:
  Render mesh in 3D, reading from textures
  **Image** = 0
  For i = 1 to k :
    **Image** += $w_i$ * **IrradBasis**[i]
    $d$ = average distance through surface, **IrradBasis**[i-1].alpha
    // fadeOut turns off global scattering when it's too close to local scattering
    $w_i' = w_i e^{-d^2/v_i} * fadeOut_i$
    **Image** += $w_i'$ * **IrradBasis**[i] at the (u,v) from the TSM
  **Image** *= Fresnel term $\rho_{dt}(x_o)$
  **Image** *= sqrt(**diffuse color**)
  **Image** += sum of specular term of lights

## 6. Results

We have applied our efficient subsurface scattering techniques to high resolution scan datasets of two human heads. The head depicted in Figure 7 was chosen for comparison to Donner and Jensen [DJ05]. Using standard tools (UVLayout by Headus, Melody by NVIDIA) we reduced the original 10 million triangle model to 110,000 triangles, parametrized it with a UV map, and extracted 4K×4K color and normal maps from the full resolution mesh. For all images except Figure 6, we use Levenberg-Marquardt fitting to select 6 Gaussian basis functions for the reflectance profiles for the 3 layer skin model in Donner and Jensen [DJ05]. The final pixel shader combines five convolved irradiance inputs with a local non-scattered irradiance calculation (which represents the narrowest of the 6 Gaussians in our basis). The scattering achieved in each color channel appears to accurately model the appearance of real skin. Figure 6 (center) shows a closeup of the forward scattering through the ear, which is achieved with the modified translucent shadow map. In Figure 10, we apply a simple procedural wave deformation to demonstrate our technique's realtime nature and its applicability to animating or deforming models.

| | resolution | | | no TSM | | TSM | |
|---|---|---|---|---|---|---|---|
| | final render | irradiance | shadow maps | env only | 1 light | 1 light | 2 lights |
| Full-Res | 2500x1500* | 2048x2048 | 2048x2048 | 60 fps | 44 fps | 31 fps | 23 fps |
| Low-Res | 1024x1024 | 1024x1024 | 1024x1024 | 115 fps | 103 fps | 75 fps | 61 fps |

*downsampled afterwards to display resolution

**Table 1:** *Summary of performance results for Figure 8. Renders include environment lighting plus 0, 1, or 2 point light sources. Note that no precomputation is necessary.*

Some additional details: we pre-compute $\rho_{dt}$ for various roughness values $m$ of the Specular BRDF by Kelemen and Szirmay-Kalos [KS01] (using the Beckmann $P_{\vec{H}}$ term and Schlick's Fresnel approximation) and store it in a low resolution 2D texture. We apply a low resolution map which varies $\rho_s$ and $m$ over the face according to the survey by Weyrich et al. [WMP*06].

We store irradiance into a 2048×2048 texture initially; large scale convolutions use smaller textures for efficiency. The TSM and traditional shadow map used for point light shadows are both 2048×2048. We render at 2500×1500 and downsample to desktop resolution in a final pass to reduce specular aliasing. Table 1 summarizes our results. We achieve 23-60 fps depending on the use of point light sources, environment lighting, and translucent shadow maps. We also include results from a "low res" run in which all maps and render targets (except normal and color) were reduced to 1024×1024, at frame rates of 61-115 fps. All results were gathered on a AMD Athlon FX-55 with an NVIDIA Geforce 8800 GPU.

## 7. Limitations and Future Work

We have presented a technique for efficient rendering of multi-layered translucent materials. Our approach recognizes that a sum-of-Gaussians basis captures dipole and multipole diffusion profiles very efficiently, and uses that insight to improve and extend algorithms for texture-space diffusion and translucent shadow maps. For human skin, the technique produces real-time results comparable to the state-of-the-art

**Figure 7:** *Real-time render for comparison to offline techniques used in [DJ05]*

**Figure 8:** *Real-time rendering with two point lights, one environment light, 16-bit floating-point buffers for high dynamic range, and two separable bloom passes.*

**Figure 9:** *Our skin rendering algorithm applied to color and normal maps captured live from actors [MHP*07]. Two point lights and an environment light were used here.*

**Figure 10:** *By computing and correcting for stretch each frame, we can realistically render animated or deforming models with no precomputation.*

offline rendering algorithm by Donner and Jensen [DJ05]. Its implementation is particularly efficient on GPUs and enables real-time performance. However, it is not limited to real-time applications and higher quality offline rendering pipelines could benefit equally from accelerating diffusion for skin rendering using these techniques.

Our approach has several limitations that we hope to address in future work. The depth calculation in the translucent shadow maps algorithm can be inaccurate for highly concave objects, or objects with interior structure (e.g. the bones visible through a backlit hand). Translucent shadow maps also inherit the limitations of standard shadow maps, such as selecting an appropriate resolution and depth bias. Furthermore, since we use translucent shadow maps to model the effect of forward scattering through thin regions, we are limited to a small number of point light sources (since each light source requires rendering and storing another shadow map). We cannot model forward scattering directly with environment lighting, although we could conceivably importance-sample the environment light with a few point lights to achieve convincing results.

Requiring a parameterized model poses another limitation. We have argued that this requirement is reasonable for models such as the human faces we show, but it may be burdensome in other applications. Furthermore our technique to correct for stretch would break down for models with complex topology or extreme curvature. To address these problems robustly requires extending our algorithms to support texture charts, such as are typically produced by human artists or automatic parameterization methods. The seams in a texture chart present a challenge; irradiance from different texture regions will be convolved across the seams unless steps are taken to avoid this. Artificially reducing the stretch metric (and thus the convolution kernel widths) to zero near seams mitigates but does not eliminate artifacts. A more robust solution would use multiple overlapping parameterizations such that all points are sufficiently far from any seams in at least one parameterization, along with a partition of unity [PB00] weighting that blends gradually between parameterizations. However, this adds convolution passes and lookups during the final rendering pass (the entire algorithm is duplicated for each set of charts). Another approach would be to perform convolution directly in the 3D Euclidean domain, and thus bypass the difficulties of parameterization altogether.

Although the sum-of-Gaussians approximation should apply equally well to diffuse scattering in any highly scattering multilayer material (such as leaves, fruit, thin coats of paint, etc.), we have not yet demonstrated this. Nor have we performed an exhaustive error analysis, beyond verifying the accuracy of a 4-Gaussian Levenberg-Marquardt fit to dipoles over a representative range (Section 4) and the 8-Gaussian dipole "fast fit". We would like to analyze the powers-of-Gaussians fitting to dipoles and multipoles, and prove formally that the fits can be made arbitrarily accurate as the number of Gaussians increase. Similarly, we would like to perform a formal error analysis of our stretch correction approach, though considering "ground truth" for curved surfaces becomes complicated since the dipole and multipole models themselves assume local planarity.

In future work we plan to implement our efficient analytical fitting of Gaussians directly in the fragment shader, thus fitting profiles at every pixel every frame. This would enable us to support spatially varying (i.e. texture-controlled) diffusion profiles, for example to represent variations in human skin due to freckles, scars, or makeup. It would also enable us to build an efficient and intuitive material editing framework for multilayer materials. In particular we would like to implement real-time fitting to the spectral skin model of Donner and Jensen [DJ06]. Our initial experiments with this approach seem promising (Figure 6).

## 8. Acknowledgements

Many thanks to XYZRGB, for providing two high resolution head scans, and to Paul Debevec and his team [MHP*07] for providing the face capture datasets(Figure 9). Chris Cowan was extremely helpful in preparing all this data for real-time rendering. Thanks to Craig Donner, Henrik Wann Jensen and George Borshukov for answering many questions about their work. The paper was improved in many ways thanks to Larry Gritz, Kevin Bjorke, Sarah Tariq, Shaun Nirenstein and the EGSR reviewers. Very special thanks to actor Doug Jones for permission to use his likeness.

## References

[BL03] BORSHUKOV G., LEWIS J. P.: Realistic human face rendering for "The Matrix Reloaded". In *ACM SIGGRAPH 2003 Sketches & Applications* (2003), ACM Press, p. 1.

[CHH03] CARR N. A., HALL J. D., HART J. C.: GPU algorithms for radiosity and subsurface scattering. In *Graphics Hardware 2003* (July 2003), pp. 51–59.

[DJ05] DONNER C., JENSEN H. W.: Light diffusion in multi-layered translucent materials. *ACM Trans. Graph. 24*, 3 (2005), 1032–1039.

[DJ06] DONNER C., JENSEN H. W.: A spectral BSSRDF for shading human skin. In *Rendering Techniques 2006 (Proceedings of the Eurographics Symposium on Rendering)* (2006), pp. 409–418.

[DS03] DACHSBACHER C., STAMMINGER M.: Translucent shadow maps. In *Rendering Techniques 2003 (Proceedings of the Eurographics Symposium on Rendering)* (2003), pp. 197–201.

[Gre04] GREEN S.: Real-time approximations to subsurface scattering. In *GPU Gems*, Fernando R., (Ed.). Addison Wesley, Mar. 2004, ch. 16, pp. 263–278.

[GSM04] GOSSELIN D., SANDER P. V., MITCHELL J. L.: Real-time texture-space skin rendering. In *ShaderX3: Advanced Rendering Techniques in DirectX and OpenGL*, Engel W., (Ed.). Charles River Media, Cambridge, MA, 2004.

[HBV03] HAO X., BABY T., VARSHNEY A.: Interactive subsurface scattering for translucent meshes. In *ACM Symposium on Interactive 3D graphics* (2003), pp. 75–82.

[HV04] HAO X., VARSHNEY A.: Real-time rendering of translucent meshes. *ACM Trans. Graph. 23*, 2 (2004), 120–142.

[JB02] JENSEN H. W., BUHLER J.: A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph. 21*, 3 (2002), 576–581.

[JMLH01] JENSEN H. W., MARSCHNER S. R., LEVOY M., HANRAHAN P.: A practical model for subsurface light transport. In *Proceedings of ACM SIGGRAPH 2001* (2001), pp. 511–518.

[KS01] KELEMEN C., SZIRMAY-KALOS L.: A microfacet based coupled specular-matte BRDF model with importance sampling. In *Eurographics Short Papers* (2001), pp. 25–34.

[MHP*07] MA W.-C., HAWKINS T., PEERS P., CHABERT C.-F., WEISS M., DEBEVEC P.: Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. *Rendering Techniques 2007 (Proceedings of the Eurographics Symposium on Rendering)* (2007).

[MKB*05] MERTENS T., KAUTZ J., BEKAERT P., REETH F. V., SEIDEL H.-P.: Efficient rendering of local subsurface scattering. *Computer Graphics Forum 24*, 1 (Mar. 2005), 41–50.

[PB00] PIPONI D., BORSHUKOV G.: Seamless texture mapping of subdivision surfaces by model pelting and texture blending. In *Proceedings of SIGGRAPH 2000* (2000), pp. 471–478.

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *ACM Trans. Graph.* (2002), vol. 21, pp. 527–536.

[SLS05] SLOAN P.-P., LUNA B., SNYDER J.: Local, deformable precomputed radiance transfer. *ACM Trans. Graph. 24*, 3 (2005), 1216–1224.

[WMP*06] WEYRICH T., MATUSIK W., PFISTER H., BICKEL B., DONNER C., TU C., MCANDLESS J., LEE J., NGAN A., JENSEN H. W., GROSS M.: Analysis of human faces using a measurement-based skin reflectance model. *ACM Trans. Graph. 25*, 3 (2006), 1013–1024.

[WTL05] WANG R., TRAN J., LUEBKE D.: All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Trans. Graph. 24*, 3 (2005), 1202–1207.

[WWD*05] WANG L., WANG W., DORSEY J., YANG X., GUO B., SHUM H.-Y.: Real-time rendering of plant leaves. *ACM Trans. Graph. 24*, 3 (Aug. 2005), 712–719.

**Appendix A:** An Empirical Dipole Fit

Brief analysis revealed a pattern to the way four Gaussians optimize to fit a given pole equation, which allows fast analytic fitting of four Gaussians to any pole equation with no Levenberg-Marquardt minimization code or convergence concerns. Dipoles and multipoles are the sum of two or more pole functions which depend on r in the following manner

$$P(r,\sigma_{tr},z) = \frac{\left(1+\sqrt{z^2+r^2}\sigma_{\mathrm{tr}}\right)e^{-\sqrt{z^2+r^2}\sigma_{\mathrm{tr}}}}{\left(z^2+r^2\right)^{\frac{3}{2}}}, \qquad (17)$$

which has power $P_d(\sigma_{tr},z) = \int_0^\infty P(r,\sigma_{tr},z)rdr = e^{-\sigma_{tr}|z|}/|z|$. Finding a single Gaussian with equal power and equal x-intercept gives $w_0 G(v_0,r)$, with $w_0(\sigma_{tr},z) = 2\pi P_d(\sigma_{tr},z)$ and $v_0 = P_d(\sigma_{tr},z)/P(0,\sigma_{tr},z)$. This provides a quick analytic single Gaussian fit to any pole, however it is not sufficiently accurate for rendering. Further analysis of four-Gaussian fitting revealed the spread of Gaussian variances was invariably close to $\{0.2304v_0, 0.7225v_0, 2.6569v_0, 13.6v_0\}$ for a wide range of $\sigma_{tr}$ and $z$. With these variances, the four weights are $w_i = C_i(|\sigma_{tr}z|)P_d(\sigma_{tr},z)$, where $C_i$ are four curves which have fairly simple analytic approximations. Fitting each pole independently with 4 Gaussians, we can quickly find 8 Gaussians that fit any dipole with minimal error. An error analysis of this fit is shown in Figure 11.

$$C_1(x) = -0.3454 + \frac{1.2422}{x} - 1.2422\frac{e^{-0.7452x}}{x}$$

$$C_2(x) = 5.34071\left(\frac{\left(1-e^{-0.716x}\right)(0.9029x-0.7401)}{x} + 0.90295e^{-0.7165x}\right)$$

$$C_3(x) = 10.2416\left(\frac{\left(1-e^{-1.0961x}\right)(0.2420x+0.1804)}{x} - 0.00244e^{-1.0961x}(0.2041x+0.1752)\right)$$

$$C_4(x) = 23.185\left(\frac{\left(1-e^{-1.433x}\right)(0.0505-0.0395x)}{x} + 0.04425e^{-1.433x}(0.09187x+0.06034)\right).$$

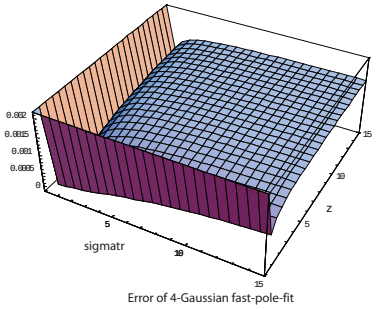


**Figure 11:** *The error metric (10) for the fast-empirical pole fit is computed for a single pole equation with $z$ and $\sigma_{tr}$ in the range [.001,15]. Errors are less than 0.2% for a wide range of scattering parameters.*

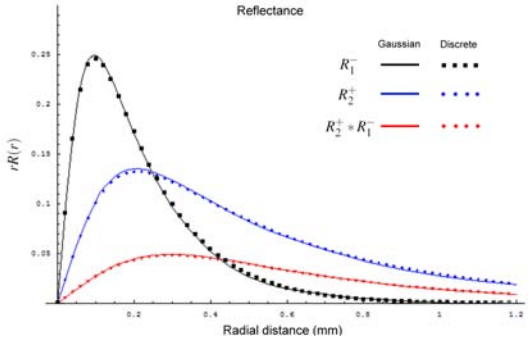


**Figure 12:** *We compare reflectance profiles for the two-layer spectral skin model ( [DJ06], Caucasian skin parameters, 500nm) computed in two ways. The discrete plots show the analytic multipole and dipole for the epidermis ($R_1$) and dermis ($R_2$), and their convolution via a Hankel transform. The continuous curves show the fast empirical fit for the epidermis and dermis, and the fast polynomial-of-Gaussians convolution of the two. Computed in 0.009 seconds, the polynomial convolution is significantly more efficient than discretizing the multipoles and using a Hankel transform.*