DOI: 10.1111/j.1467-8659.2011.02034.x

COMPUTER GRAPHICS *forum*
Volume 30 (2011), number 8 pp. 2279–2287

Convolution-Based Simulation of Homogeneous Subsurface Scattering

Adolfo Munoz, Jose I. Echevarria, Francisco J. Seron and Diego Gutierrez

Departamento de Informatica e Ingenieria de Sistemas
Universidad de Zaragoza, Spain
joseignacioechevarria@gmail.com, {dolfo, seron, diegog}@unizar.es

Abstract
This paper introduces a new method for simulating homogeneous subsurface light transport in translucent objects. Our approach is based on irradiance convolutions over a multi-layered representation of the volume for light transport, which is general enough to obtain plausible depictions of translucent objects based on the diffusion approximation. We aim at providing an efficient physically based algorithm that can apply arbitrary diffusion profiles to general geometries. We obtain accurate results for a wide range of materials, on par with the hierarchical method by Jensen and Buhler.

Keywords: subsurface scattering, convolution, rendering

ACM CCS: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture; I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms

1 Introduction

Rendering translucent materials is a daunting task due to the complexity of subsurface light transport. In the case of homogeneous materials, the diffusion approximation can be used for reducing the dimensionality of the problem [JMLH01, DJ05], fact that has made the simulation of subsurface scattering practical, both in terms of time and physical accuracy.

Inspired by the work of Donner and Jensen [DJ05] and D'eon *et al.* [dLE07], our approach is based on the observation that the equation defining multiple subsurface scattering in homogeneous materials is similar to the equation that defines convolutions. We therefore model the simulation of subsurface light transport by means of image convolutions. We set up a set of layers uniformly distributed along the volume of the object, parallel to the projection plane of the camera, distribute the incoming irradiance among them and create a set of convolution images that model light transport at each layer and between layers. Discrete Fourier transforms are applied to all the layers, as convolutions become multiplications in frequency space (and therefore are much more efficient to compute).

The results obtained show that this technique is general enough to work both with previously published data [JMLH01, NGD*06, MES*11] and user-defined materials. It compares well with more traditional solutions, while offering an attractive, customizable trade-off between rendering accuracy and computation time.

2 Previous Work

There has been a lot of recent research in the field of computer graphics related to scattering [GJJD09]. One of the most popular techniques for simulating the appearance of translucent materials is the dipole method [JMLH01]. This approach formulates a BSSRDF by combining an exact solution for single scattering with a dipole point source diffusion approximation to simulate multiple scattering. Building on that work, a hierarchical method [JB02] can be used for efficiently rendering of multiple scattering. Furthermore, this method has been mapped to the GPU and extended with an approximation of single scattering, so translucency is simulated at interactive rates [RJD05]. Our work is also based on the diffusion approximation.

Haber *et al.* [HMBVR05] present a very accurate computational solution to subsurface scattering simulation using octree discretization, a multigrid solver and an embedded boundary stencil. These allow to additionally simulate heterogeneous materials and complex geometries. Donner and Jensen [DJ05] show how to extend the dipole method for multi-layered materials. The key idea is to realize that the reflectance and transmittance at each of these thin layers can be computed as convolutions in the frequency domain, assuming that those layers are differentially plain and parallel to the surface of the object; this assumption forces the inclusion of a correction factor according to the surface normal. Although our method also models subsurface scattering as convolutions, we do not require knowing the transmittance profile for the material to be simulated. Instead, we distribute a set of virtual layers (parallel to the plane of the image) that represent two-dimensional slices of the light transport model, based on the diffusion profile.

A number of interactive and real-time approaches exploit consumer-level graphics hardware, at the cost of restricting somewhat the range of applicable diffusion profiles. Lensch *et al.* [LGB*02] store the incident illumination into a texture, which is then used for simulating local scattering by convolving it with the material profile. The global response is calculated via projection of the light texture over the mesh vertices and calculating the light transport between them through a vertex-to-vertex factor matrix (a concept also used in other techniques [HV04]). Finally, both responses are combined. Compared to them, our convolutions take into account both the local and global responses.

Dachsbacher and Stamminger [DS03] show how the global appearance of translucency can be simulated by means of filtering shadow maps, whereas Mertens *et al.* [MKB*03] focus on local behaviour, importance sampling a texture representing the profile. Chang *et al.* [CLH*08] build on top of both works, achieving interesting visual results both at local and global level. However, the diffusion profiles that can be simulated using these techniques are limited: some profiles would require a big number of texture samples for their local behaviour, and therefore maintaining real-time may not be possible. Other techniques extend the idea of translucent shadow maps by modelling the irradiance in cube maps, simulating deformable objects in real-time [BC06, GCZ*08, BJG09]. D'Eon *et al.* [dLE07] define a diffusion profile as a linear combination of zero-mean gaussian functions, to filter incident light and simulate subsurface scattering in texture space. This technique works extremely well for human skin. All these techniques suggest that filtering the incoming irradiance is a good strategy to simulate subsurface scattering. Other recent works take a screen-space approach [JSG09, JWSG], which inspired our approach of defining the irradiance layers parallel to the projection plane of the camera.

Wang *et al.* [WeCPW*08] use PCA analysis to estimate the diffusion profiles by a linear combination of basis functions, which in turn allows to pre-compute multiple scattering for interactive editing and relighting of translucent materials. Munoz *et al.* [MES*11] approximate the BSSRDF from single images of objects made of optically thick, homogeneous materials. They extract geometry and lighting information from the input image, and use piecewise-constant basis functions to derive a plausible diffusion profile.

This paper builds on top of the work by Echevarria *et al.* [EMSG10] by

- Improving the visual accuracy of the renders by modifying the way in which light is distributed to (and combined from) the virtual layers.
- Developing several optimizations over the original algorithm, that avoid computations that would be unperceivable on the final render (less convolutions at less resolution).
- Building a set of test cases for demonstrating the behavior of the algorithm.
- Showing a greater variety and quality of results.

3 The Diffusion Approximation

Our algorithm is based on the diffusion approximation [JMLH01], which defines multiple subsurface scattering for homogeneous materials as

$$L_{\mathrm{m}}(\vec{x}_0, \vec{\omega}_0) = \frac{1}{\pi} F_{\mathrm{t}}\left(\frac{n_{\mathrm{ob}}}{n_{\mathrm{med}}}, \vec{\omega}_0\right) \int_A R_{\mathrm{d}}(\|\vec{x}_0 - \vec{x}_i\|) E(\vec{x}_i) \mathrm{d}A(\vec{x}_i), \tag{1}$$

where L_{m} is the exitant radiance, F_{t} is the Fresnel transmission term, A defines the surface area of the object, $\vec{x}_i$ and $\vec{x}_0$ define the incident and exitant point of light respectively, $\vec{\omega}_i$ and $\vec{\omega}_0$ define the incident and outgoing light directions and n_{ob} and n_{med} define the indices of refraction of the object and the medium. R_{d} is a one-dimensional function called the *diffusion profile* that defines the properties of the material regarding subsurface scattering. Several models for this function can be found in different works, such as the dipole model [JMLH01], which we choose in this work in order to compare our results with results from physically captured materials. E defines the irradiance, that can be computed as follows:

$$E(\vec{x}_i) = \int_\Omega F_{\mathrm{t}}\left(\frac{n_{\mathrm{ob}}}{n_{\mathrm{med}}}, \vec{\omega}_i\right) L(\vec{x}_i, \vec{\omega}_i)(\vec{n}_i \cdot \vec{\omega}_i) d\vec{\omega}_i \tag{2}$$

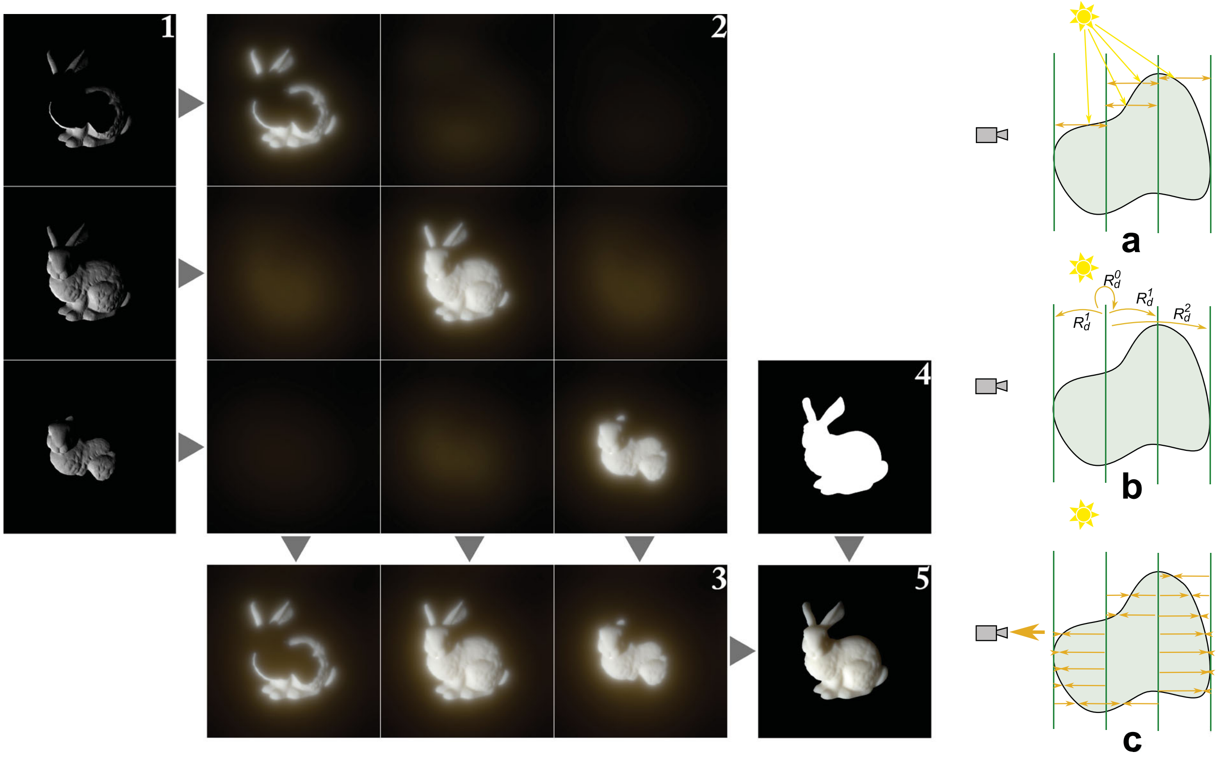


Figure 1: *Overview of our algorithm. We place parallel layers equally spaced along the volume of the object. The irradiance of the surface of the object is distributed among the layers (1). Partial subsurface scattering is simulated by means of convolutions applied to each of the layers (2). Adding these partial results we obtain the radiant emittance maps for each layer (3). Using the object matte as a mask (4), these maps are combined for the final result (5). Figures a– c on the right column show a schematic view of the first three steps, respectively.*

where Ω refers to the whole hemisphere of incident light, L represents incident radiance and $\vec{n}_i$ is the normal of the surface at $\vec{x}_i$. This is computed independently for all colour channels (RGB).

Assuming that the material to be simulated is optically thick, single scattering is negligible compared to multiple scattering [JB02], and therefore Equation (1) approximates subsurface scattering for such materials. In the case of homogeneous materials, this simple assumption will allow us to simulate light transport by means of image convolutions.

4 Our Algorithm

4.1 Overview

Figure 1 shows an overview of our rendering algorithm. At its core lies the observation that the effect of subsurface scattering can be modelled as convolutions [DJ05, dLE07]. We first define a set of equally spaced parallel planes that divide the volume covered by the object. Although our algorithm works for a generic orientation of those planes, the best results are achieved when these are parallel to the projection plane of the camera. From now on, we consider that orientation to be the optimal and therefore the default for our simulations.

We first compute the irradiance across the surface of the object. For each point on the surface, we then store it dividing it between the two closest layers (front and back for each point). Subsurface scattering effects are calculated by considering the diffusion profile of the material and performing convolutions between layers. In our work, we model these convolutions from the diffusion profile R_d for a each layer. Then the final result integrates the contributions from all layers. Our algorithm is sufficiently general that it can also be used with measured scattering data [JMLH01, NGD*06, MES*11] as well as user-defined diffusion profiles, by simply defining absorption and extinction coefficients and using the dipole model [JMLH01].

4.2 Rendering light transport as convolutions

Our aim is to represent Equation (1) as convolutions. The key idea is to notice the similarity between the integrals of

Equation (1) and the definition of convolution:

$$(f * g(x, y)) = \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} f(x, y)g(x - i, y - j)\mathrm{d}i\right)\mathrm{d}j. \tag{3}$$

The integral in Equation (1) contains a product of functions; if we reduce the vectors $\vec{x}_i$ and $\vec{x}_0$ to a two-dimensional space, we can express Equation (1) as

$$L_{\mathrm{m}}(\vec{x}_0, \vec{\omega}_0) = \frac{1}{\pi} F_{\mathrm{t}}\left(\frac{n_{\mathrm{obj}}}{n_{\mathrm{med}}}, \vec{\omega}_0\right)(E * R_{\mathrm{d}})(\vec{x}_0). \tag{4}$$

The extension to three dimensions is performed by defining a set of two-dimensional layers, and by finding the convolutions that define the light transport between them. This is an efficient approach given that all the convolutions can be done in the frequency domain (in which they become multiplications) by applying a Discrete Fourier Transform to both E and R_{d}.

In our coordinate system, x and y are the horizontal and vertical axes of the image, whereas z represents depth, perpendicular to the image plane. We subdivide the scene by defining a series of n_{l} layers equally spaced along the z -axis of the shape of the object, and parallel to the image plane. Each layer is then located at a distance z^{l} given by $z^{\mathrm{l}} = \min(z) + l\Delta z$ where $\Delta z = (\max(z) - \min(z))/(n_{\mathrm{l}} - 1)$ $(l = 0 \ldots n_{\mathrm{l}} - 1)$ and $\min(z)$ and $\max(z)$ represent the limits of the bounding box of the object in the z-axis. A good compromise between rendering time and accuracy is achieved with a number of layers n_{l} varying between 4 and 8. Assuming that all interactions between pairs of layers are due to multiple scattering, we rely on convolutions to compute the contribution between them.

Our approach is similar in spirit to Donner and Jensen's [DJ05] in that both methods use convolutions to simulate subsurface light transport. However, there are several key differences:

- Donner and Jensen's layers represent different material properties of the object being rendered, while our layers are simply a convenient way to represent two-dimensional slices of the irradiance along the object. In Donner and Jensen's method, these layers are defined in object-space, parallel to its surface, which requires a reparametrization of the reflectance and/or transmittance profiles according to the surface normal when the surface is in fact not planar. Our layers are independent of the object's geometry, and are defined in image-space, set at equidistant intervals and parallel to the camera's projection plane. The material properties taken into account by Donner and Jensen's layers are reflectance and transmittance profiles, which vary from layer to layer. Our work accounts for an homogeneous diffusion profile.
- At each of these layers, we define its corresponding *irradiance map* E_i^{l}. The resolution of these maps should be the resolution of the final image, although, depending on the properties of the diffusion profile and on the geometry of the object, lower resolutions (even four times smaller) can be used without any perceivable effect on the subsurface light transport, therefore reducing rendering times.
- Instead of attempting to model and render a multi-layered material, our work aims to improve the efficiency of rendering a homogeneous material without limitations nor assumptions about the diffusion profile.

Light is sampled in image space, both from the front and from the back of the object. The irradiance at each sample is computed from Equation (2) and equally distributed between the corresponding pixels of the two closest layers (half of the irradiance to each of them).

We generate n_{l} convolution maps R_{d}^e, each one representing the effect of the incoming irradiance at a specific layer, on the outgoing radiance at a layer at distance $e = m\Delta Z$ with $m = 0, ..., n_{\mathrm{l}} - 1$. These convolution maps are generated as $R_{\mathrm{d}}^e(x, y) = R_{\mathrm{d}}(\psi)$ where

$$\psi = \sqrt{\left(k_x\left(x - \frac{w}{2}\right)\right)^2 + \left(k_y\left(y - \frac{h}{2}\right)\right)^2 + (e\Delta z)^2}, \tag{5}$$

where w and h represent the width and height of the downsampled irradiance maps, whereas k_x and k_y are scale factors that relate the size of the irradiance maps with the size of the geometry of the object. The *radiant emittance map* at each layer (M_{l}) is then obtained by convolving it with each R_{d}^e yielding

$$M_{\mathrm{l}}(x, y) = \sum_{\forall i, e=|i-l|} E_i(x, y)R_d^e(x, y). \tag{6}$$

For efficiency reasons, this convolution is computed using the Discrete Fast Fourier Transform implemented in the FFTW library [FJ05].

Had lower resolution layers been employed for efficiency, the radiant emittance maps M_{l} are now resized to the image resolution. The final outgoing radiance (the resulting image) at any point x_0 is computed by using Equation (1) as follows:

$$L_{\mathrm{m}}(\vec{x}_0, \omega_0) = \frac{1}{\pi} F_{\mathrm{t}}(\frac{n_{\mathrm{ob}}}{n_{\mathrm{med}}}, \omega_0)M(x_0), \tag{7}$$

where the radiant emittance $M(x_0)$ is computed from linear interpolation between the corresponding pixels of the closest resized radiant emittance maps from the two closest layers (Figure 1). Specular highlights are included afterwards by simply using the Phong model, to enhance the perception of translucency [FB05].

5 Optimizations

Although the full light transport simulation is only achieved by applying the previously presented algorithm, there are several optimizations that can be devised to increase efficiency.

One of the optimizations is based on the fact that the diffusion profile is monotonically decreasing, and thus the influence between layers also decreases with distance. For instance, a large, optically thick object would present no practical contribution from the front layer to the back layer. This can be seen in Figure 1, where most of the energy is stored in the three layers along the diagonal. We can leverage this observation by detecting the threshold distance d_t from which the diffusion profile presents a negligible contribution and then avoid computing convolutions between layers further apart than such distance. More specifically, given a threshold t defining the percentage of energy that is considered as negligible, we find d_t as follows:

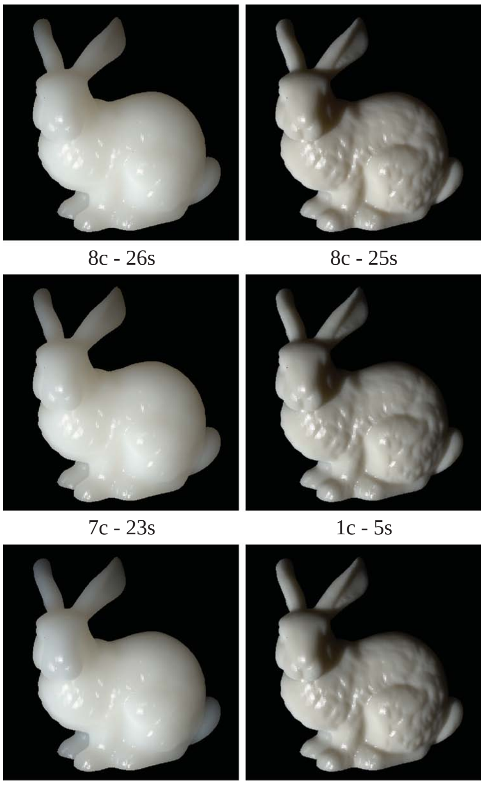
$$t\,R_\mathrm{d}(0) = R_\mathrm{d}(d_\mathrm{t}) \Rightarrow d_\mathrm{t} = R_\mathrm{d}^{-1}\,(t\,R_\mathrm{d}(0))\,. \tag{8}$$

All the contributions from layers i to layer l where $|z_i - z_\mathrm{l}| > d_\mathrm{t}$ can then be ignored in Equation (6). However, since computing the inverse of the diffusion profile might not be possible, in practice we ignore all layers i contributing to layer l where $tR_\mathrm{d}(0) > R_\mathrm{d}(|z_i - z_\mathrm{l}|)$. This optimization reduces the number of convolutions needed to reach the final solution. As the bottleneck of the algorithm is the computation of the corresponding Fourier transforms for the convolutions, the speed-up is noticeable, up to five times faster.

Notice, however, that the number of convolutions required is dependent not only on the chosen threshold t but on the optical thickness of the material. Very optically thick materials present less contributions between layers than optically thinner ones, and as a consequence for the same threshold t the former materials require less convolutions to converge than the latter. Figure 2 shows this. For threshold $t = 0.1\%$, a small bunny requires seven convolutions per layer, whereas the largest one requires just a single convolution per layer (the optimal case). In both cases, the results are almost identical to the full solution. However, if the threshold is increased to $t = 5\%$ the small bunny requires just two convolutions per layer, although the difference starts being noticeable. Our tests show that $t = 0.1\%$ is a practical threshold for this optimization, giving a good compromise between average speed-up and image quality.

Another practical optimization is the downsampling of the resolution of each of the layers with respect to the resolution of the final image. This has already been introduced in Section 4. However, the effect of this optimization greatly depends on the properties of the material (more translucent materials can be rendered with lower resolution layers). Figure 3 shows the case of wholemilk, where the resolution can be as low as 25% of the original size before artefacts start to be noticeable.

Notice how the first optimization proposed works better for optically thick materials, whereas the second one gives better results for more translucent ones. This makes combining both techniques a good strategy to speed up all kinds of homogeneous materials while preserving quality. Finally, another relevant optimization consists of applying Equation (6) just to layers for which the final emittance is seen from the camera. Different geometries result into a different number of layers being ignored, reducing the computation time to about half on average.



8c - 26s 8c - 25s

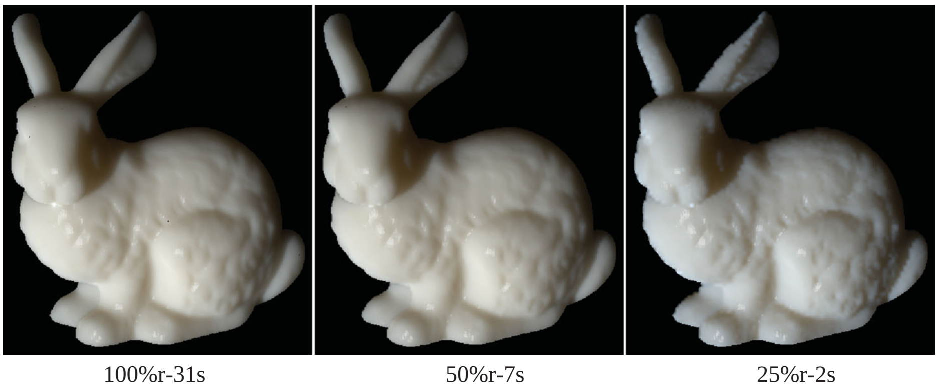
7c - 23s 1c - 5s

2c - 7s 1c - 5s

Figure 2: *Efficiency and visual test for the threshold optimization, using a Stanford bunny made of wholemilk [JMLH01]. Eight layers were used at 800x800, without downsampling. Left column presents bunnies that are ten times smaller than the ones on the right column. From top to botton: threshold t = 0% (no optimization), threshold t = 0.1% (small optimization) and threshold t = 5% (large optimization). Below each image, we show the number of convolutions per layer c and the rendering time in seconds s.*

6 Results and Discussion

We have presented an algorithm for the efficient rendering of subsurface scattering in the case of homogeneous optically thick materials, which provides an attractive tradeoff between accuracy and speed. By means of the proposed



100%r-31s

50%r-7s

25%r-2s

Figure 3: *Efficiency and visual test for different downsampled resolutions, using a Stanford bunny made of wholemilk [JMLH01]. Eight layers were used at 800×800. From left to right: full resolution, half resolution and a quarter resolution (represented by r). Simulation time indicated in seconds s.*

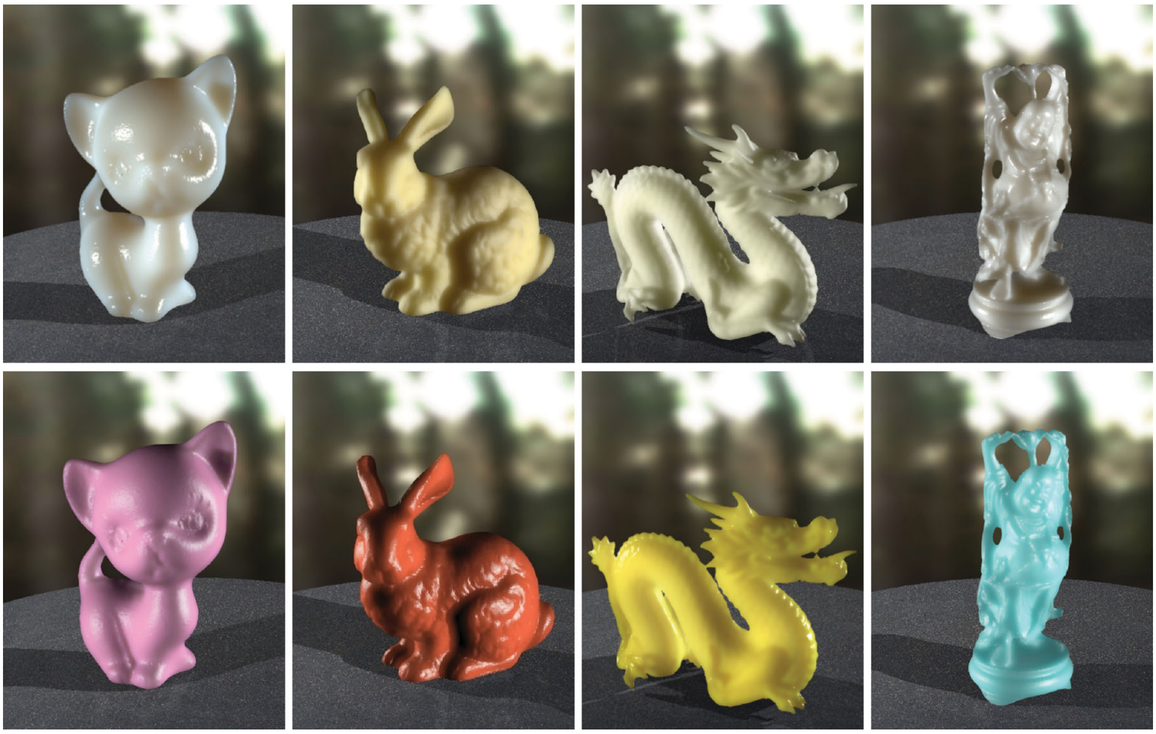
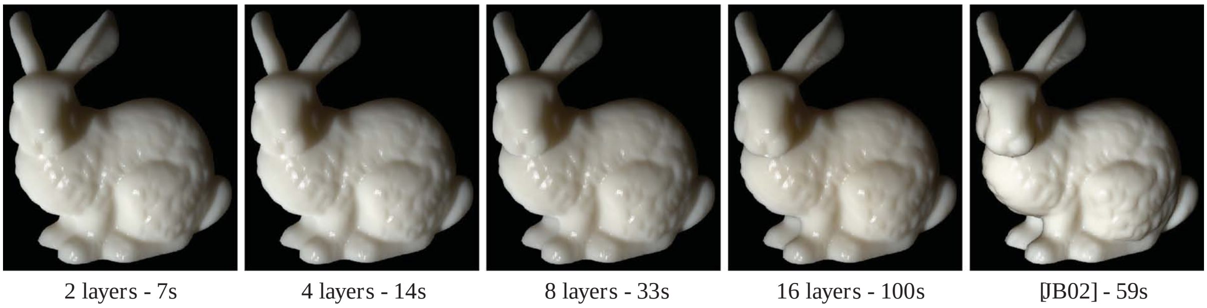


Figure 4: *Results of our algorithm. Top row: skimmilk, potato, apple and wholemilk from [JMLH01]. Bottom row: Wax, ketchup, orange juice and soap from [MES*11].*

optimizations, this trade-off can be customized for different scenarios. Our technique is versatile enough to model a wide variety of materials. Figure 4 shows different diffusion profiles applied to different geometries. We have applied the photographic tone-mapping operator [RSSF02] to all the images for display purposes.

Our algorithm takes an average of seven seconds for the simulation of light transport, at 800 × 600, without any optimizations and using four layers, on an Intel Core i7 @ 2.66 GHz with 6 GB of RAM. By applying all the optimizations (0.1% threshold, reducing layer resolution to a half) our algorithm runs in around 2 s. We have integrated



2 layers - 7s | 4 layers - 14s | 8 layers - 33s | 16 layers - 100s | [JB02] - 59s

Figure 5: *Test of our algorithm with respect to the number of layers (at 800 × 800 resolution), rendering a Stanford bunny made of wholemilk. From left to right: 2 layers, 4 layers, 8 layers, 16 layers and ground truth (rendered using the hierarchical algorithm [JB02]), with the corresponding rendering times in seconds. No optimizations were applied.*

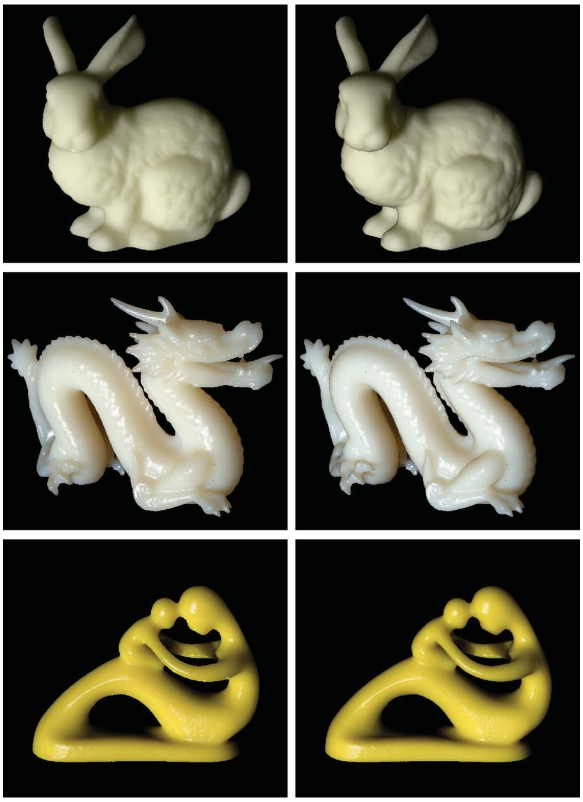


Figure 6: *Comparison of the results. Left column: renders from our algorithm. Right column: renders from Jensen and Buhler's algorithm [JB02]. From top to bottom: apple and marble [JMLH01] and orange juice [MES*11].*

our algorithm into a ray-tracer, that takes an average of 14 s to compute all the intersections and shadows. However, this could be optimized by using a rasterizer and porting it to GPU.

In Figure 5, we test the output of our algorithm with respect to the number of layers used. Notice that, even though the accuracy and the perception of the geometrical details increase with the number of layers (see for instance the chin and the front feet, with more accurate shading as the number of layers increases), good solutions are obtained with a very low number of layers.

The results of our technique are visually comparable to previous physically based subsurface rendering algorithms. Figure 6 shows different results compared to ground truth renderings. There are several previous techniques that are based on the diffusion approximation for simulating subsurface scattering, which could be potentially used as ground truth for our tests. Some of these techniques are based on the local high-frequency behaviour of the material [MKB*03], on a global low-frequency representation [DS03] or on a combination of both [LGB*02, HV04, CLH*08]. Others have been designed specifically to simulate specific materials, such as human skin [dLE07, JSG09, JWSG]. To provide fair comparisons with our results, we thus use the method of Jensen and Buhler [JB02] which, similar to our approach, is able to deal with generic diffusion profiles.

7 Limitations and Future Work

Current limitations of our method are mainly related to rendering times. In our CPU implementation, Fourier transforms are still relatively costly. As we have seen, this can be ameliorated by using lower resolutions for the irradiance layer maps. However, this down-sampling affects the final appearance of the render and may become a problem with fast-decaying diffusion profiles, where the softening effects of our approach could turn into a visible loss of detail. Using layers with variable resolution and distribution depending on the geometry could be studied. Recent FFT GPU implementations [GLD*08] could also be used to speed up computations; while achieving real-time would still be a challenge, performance gains could reach interactive frame rates. We have also offered several other optimizations strategies which maintain visual accuracy in the results, while significantly reducing rendering times.

Our results are restricted to optically thick, homogeneous materials. It would be interesting to extend our algorithm to heterogeneous and optically thin materials. For the latter, single scattering would need to be taken into account.

8 Conclusions

Our algorithm is general enough to work with a wide range of diffusion profiles, either measured or defined by the user. Even though we approximate light transport by convolutions, it succeeds at reproducing the appearance of translucent materials, producing results that are visually on-par with previous state-of-the-art techniques. We believe it presents an attractive, customizable balance between speed and quality. Although our current implementation runs on the CPU, its image-based approach makes it a good candidate to efficiently run on the GPU. We hope it can inspire future real-time algorithms, as well as image editing techniques.

Acknowledgements

This research has been funded by a Marie Curie grant from the Seventh Framework Programme (grant agreement no.: 251415), the Spanish Ministry of Science and Technology (TIN2010-21543) and the Gobierno de Aragón (projects OTRI 2009/0411 and CTPP05/09). Jose I. Echevarria was additionally funded by a research grant from the Instituto de Investigación en Ingeniería de Aragón.

References

[BC06] Banterle F., Chalmers A.: A fast translucency appearance model for real-time applications. In *Proceedings of the 22nd Spring Conference on Computer Graphics* (Castá-Papiernicka, Slovakia, 2006), ACM Press, New York.

[BJG09] Benmounah, N., Jolivet, V., Ghazanfarpour, D.: Real-time rendering of deformable translucent objects. *EG UK Theory and Practice of Computer Graphics*. IEEE, 2009.

[CLH*08] Chang C.-W., Lin W.-C., Ho T.-C., Huang T.-S., Chuang J.-H.: Real-time translucent rendering using gpu-based texture space importance sampling. *Computer Graphics Forum 27*, 2 (2008), 517–526.

[DJ05] Donner C., Jensen H. W.: Light diffusion in multi-layered translucent materials. *ACM Transactions on Graphics 24*, 3 (SIGGRAPH) (2005), 1032–1039.

[dLE07] D'Eon E., Luebke D., Enderton E.: Efficient rendering of human skin. In *Proceedings of the Rendering Techniques: 18th Eurographics Workshop on Rendering* (Grenoble, France, 2007), Eurographics Association, pp. 147–158.

[DS03] Dachsbacher C., Stamminger M.: Translucent shadow maps. In *Proceedings of the 14th Eurographics Workshop on Rendering* (2003), Eurographics Association, pp. 197–201.

[EMSG10] Echevarria, J. I., Munoz, A., Seron, F. J., Gutierrez, D.: Screen-space rendering of translucent objects. In *Proceedings of the Congreso Español de Informática Gráfica* (Valencia, Spain, 2010).

[FB05] Fleming R. W., Bülthoff H. H.: Low-level image cues in the perception of translucent materials. *ACM Transactions on Applied Perception 2*, 3 (2005), 346–382.

[FJ05] Frigo M., Johnson S. G.: The design and implementation of FFTW3. *Proceedings of the IEEE Special Issue on Program Generation, Optimization, and Platform Adaptation 93*, 2 (2005), 216–231.

[GCZ*08] Gong Y., Chen W., Zhang L., Zeng Y., Peng Q.: GPU-based rendering for deformable translucent objects. *The Visual Computer 24*, 2 (2008),95–103.

[GJJD09] Gutierrez D., Jensen H. W., Jarosz W., Donner C.: Scattering. In *Proceedings of the ACM SIGGRAPH ASIA 2009 Courses* (Yokohama, Japan, 2009), ACM Press, New York, pp. 15:1–15:620.

[GLD*08] Govindaraju N. K., Lloyd B., Dotsenko Y., Smith B., Manferdelli J.: High performance discrete Fourier transforms on graphics processors. In *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing* (Piscataway, NJ, USA, 2008), IEEE Press, pp. 2:1–2:12.

[HMBVR05] Haber T., Mertens T., Bekaert P., Van Reeth F.: A computational approach to simulate subsurface light diffusion in arbitrarily shaped objects. In *Proceedings of Graphics Interface 2005* (Austin, Texas, USA, 2005), Canadian Human-Computer Communications Society, pp. 79–86.

[HV04] Hao X., Varshney A.: Real-time rendering of translucent meshes. *ACM Transactions on Graphics 23*, 2 (2004), 120–142.

[JB02] Jensen H. W., Buhler J.: A rapid hierarchical rendering technique for translucent materials. *ACM Transactions on Graphics 21*, 3 (SIGGRAPH) (2002), 576–581.

[JMLH01] Jensen H. W., Marschner S. R., Levoy M., Hanrahan P.: A practical model for subsurface light transport. In *SIGGRAPH: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (Los Angeles, California, USA, 2001), ACM Press, New York, pp. 511–518.

[JSG09] Jimenez J., Sundstedt V., Gutierrez D.: Screen-space perceptual rendering of human skin. *ACM*

Transactions on Applied Perception 6, 4 (2009). http://doi.acm.org/10.1145/1609967.1609970.

[JWSG] Jimenez J., Whelan D., Sundstedt V., Gutierrez D.: Real-time realistic skin translucency. *IEEE Computer Graphics and Applications 30*, 4 (2010), pp. 32–41. http://dx.doi.org/10.1109/MCG.2010.39.

[LGB*02] Lensch H. P. A., Goesele M., Bekaert P., Kautz J., Magnor M. A., Lang J., Seidel H.-P.: Interactive rendering of translucent objects. In *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications* (2002), IEEE Computer Society, pp. 214–224.

[MES*11] Munoz A., Echevarria J. I., Seron F., Lopez-Moreno J., Glencross M., Gutierrez D.: BSSRDF estimation from single images. *Computer Graphics Forum 30*, 2 (Eurographics) (2011), 455–464.

[MKB*03] Mertens T., Kautz J., Bekaert P., van Reeth F., Seidel H.-P.: Efficient rendering of local subsurface scattering. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications* (Canmore, Alberta, Canada, 2003), IEEE Computer Society, pp. 51–58.

[NGD*06] Narasimhan S. G., Gupta M., Donner C., Ramamoorthi R., Nayar S. K., Jensen H. W.: Acquiring scattering properties of participating media by dilution. *ACM Transansactions on Graphics 25*, 3 (SIGGRAPH) (2006), 1003–1012.

[RJD05] Rui W., John T., David L.: All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Transactions on Graphics 24*, 3 (SIGGRAPH) (2005),1202–1207.

[RSSF02] Reinhard E., Stark M., Shirley P., Ferwerda J.: Photographic tone reproduction for digital images. *ACM Transactions on Graphics 21*, 3 (SIGGRAPH) (2002), 267–276.

[WeCPW*08] Wang-erb R., Cheslack-Postava E., Wang R., Luebke D., Chen Q., Hua W., Peng Q., Bao H.: Real-time editing and relighting of homogeneous translucent materials. *The Visual Computer 24*, 7 (2008), 565–575.