Separable Subsurface Scattering
Supplementary Materials

Contents This is the supplementary material for the paper *Separable Subsurface Scattering*. It includes:

· The present document.

· A document including the plots of the approximation kernels, compared against the actual kernel.

· The main Supplementary Video.

This document includes the following sections:

- **B Detailed explanation of the Low-Rank Approximation**
- **C Derivation of our Pre-integrated Kernel**
- **D Motivation for the Guided Optimization**
- **E Detailed description of the shader implementation**
- **F Parameters of the profile simulations**
- **G Plots of the simulated diffusion profiles**

We also include additional rendering results: Figure S.2 and S.5 show a comparison of all kernel approximations for human skin, using the head model as well as artificial 'dot' illumination, respectively. An additional example rendering of fruits on a plate is shown in Figure S.4. A second head model rendered from multiple distances is included in Figure S.8, while Figures S.6 and S.7 illustrate the benefits of the proposed jittered sampling approach.

B Low-rank approximations

In order to accurately approximate the diffuse reflectance profile $R_d(x, y)$ with a sum of separable kernels $A(x, y) = \sum_{i=1}^{N} a_i(x)a_i(y)$, an adequate choice for the individual 1D functions a_i is required. d'Eon and colleagues [dLE07] observed that zero-mean Gaussians G are suitable for this task, i.e.,

$$R_d(x, y) \approx A_g(x, y) = \sum_{i=1}^{N} w_i G(x, y; \tau_i), \tag{S.1}$$

where τ_i denote the standard deviations of the respective Gaussians. Due to the separability of the Gaussian kernel, the convolution with A_g can be realized as $2N$ 1D convolutions. During the computation of adequate w_i's and τ_i's, d'Eon et al. employed an $\mathcal{L}_1$ constraint such that $\|A_g\|_1 = \|R_d\|_1$, which guarantees energy conservation of the approximation A_g. Furthermore, the Gaussian kernel is spherically symmetric and thus A_g also exhibits this feature of the original profile.

We found, however, that emphasizing the closeness criterion and forfeiting radial symmetry yields approximations that produce the same visual quality with less convolutions, i.e., with lower complexity (see Figure S.5). Inspired by previous methods for low-rank approximations of reflectance data [LRR04, PvBM*06, KM99], we employ matrix factorization of the *discrete* 2D diffusion profile $R_d \in \mathbb{R}^{m \times m}$. Following from the Eckhart-Young theorem [EY36], a truncation of the singular value decomposition (SVD) of R_d gives the best low-rank approximation with respect to the Frobenius norm. In more detail, given the SVD of the diffusion profile:

$$\begin{aligned} R_d &= U \Sigma V^T, \\ U &= \left(u^{(1)} | \, u^{(2)} | \, \ldots | \, u^{(m)} \right), \\ V &= \left(v^{(1)} | \, v^{(2)} | \, \ldots | \, v^{(m)} \right), \\ \Sigma &= \operatorname{diag}\left(\sigma_1, \sigma_2, \ldots, \sigma_m \right), \end{aligned} \tag{S.2}$$

the exact solution to the approximation problem

$$\begin{aligned} &\min_{A} \|R_d - A\|_F, \\ &\text{subject to } \operatorname{rank}(A) = N, \end{aligned} \tag{S.3}$$

is given by

$$\begin{aligned} &A_s = U \Sigma_N V^T, \\ &\text{where } \Sigma_N = \operatorname{diag}(\sigma_1, \ldots, \sigma_N, 0, \ldots, 0). \end{aligned} \tag{S.4}$$

The Frobenius norm follows the classical definition, i.e., $\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij}^2} = \sqrt{\sum_{i=1}^{m} \sigma_i^2}$ and corresponds to the $\mathcal{L}_2$ norm for continuous 2D functions. Note that Σ_N is a diagonal matrix and the approximation A_s can be written as a sum of separable kernels, i.e., $A_S = \sum_{i=1}^{N} u^{(i)} \, \sigma_i \, {v^{(i)}}^T$. The 1D functions a_i are therefore given by $a_i = \sqrt{\sigma_i} u^{(i)}$, since due to the symmetry of R_d, $u^{(i)} = v^{(i)}$.

While this approach does not preserve spherical symmetry and energy conservation, it provides optimal closeness in the sense of the $\mathcal{L}_2$ norm. For low-rank approximations starting from $N = 3$, the diffusion profile is more faithfully reconstructed, and the violation of both spherical symmetry and the $\mathcal{L}_1$ norm are not perceivable. Energy conservation can be enforced by scaling the approximation A_s according to $\tilde{A}_s = A_s \frac{\|R_d\|_1}{\|A_s\|_1}$. Although the new approximation $\tilde{A}_s$ is not optimal in the $\mathcal{L}_2$ sense, it still provides a perceptionally better match than the Gaussian approximation A_g with the same number of terms. Thus, our SVD-based approximation scheme yields better performance and the same or superior visual quality at the same number of 1D convolutions when compared to the Gaussian approximation (see Figure S.5).

C Derivation of the pre-integrated approximation

Here you can find a more extended version of the derivation of Equ. (4) in the main paper:

$$\begin{aligned} M_e(x,y) &= \iint E(x',y')\, R_d(x-x', y-y')\, dx' dy' \\ &= \iint \left(E_1(x') + E_2(y')\right) R_d(x-x', y-y')\, dx' dy' \\ &= \int E_1(x') \underbrace{\int R_d(x-x', y-y')\, dy'}_{a_p(x-x')}\, dx' \\ &\qquad + \int E_2(y') \underbrace{\int R_d(x-x', y-y')\, dx'}_{a_p(y-y')}\, dy' \\ &= \int E_1(x')\, a_P(x-x') \underbrace{\frac{1}{\|a_p\|_1} \int a_p(y-y')\, dy'}_{=1}\, dx' \\ &\quad + \int E_2(y')\, a_P(y-y') \overbrace{\frac{1}{\|a_p\|_1} \int a_p(x-x')\, dx'}\, dy' \\ &= \iint E(x',y') \frac{1}{\|a_p\|_1} a_p(x-x') a_p(y-y')\, dx' dy' \end{aligned} \tag{S.5}$$

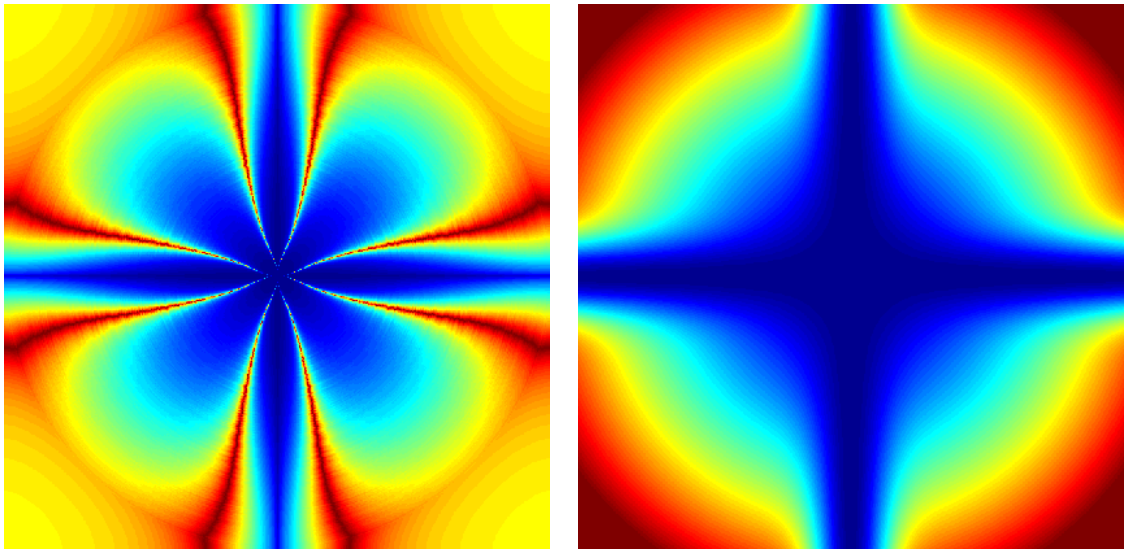
D Motivation for the guided rank-1 approximation

Motivated by the high quality of the pre-integrated approximation, we tried to achieve a practical optimization framework that allows the user to emphasize either near or far scattering by manipulating only a single parameter, similar to the two-Gaussian approximation, but starting from an accurate fit to an actual diffusion profile. For a default parameter, the output should provide a 'neutral' approximation close to the analytically derived pre-integrated approximation.

Stating the associated optimization problem as

$$\begin{aligned} a_s = \operatorname*{argmin}_{a} &\int_{\mathbb{R}^2} \Gamma(x,y) \big(R_d(x,y) - a(x)a(y)\big)^2 \, dx\, dy \\ &\text{subject to } \|R_d\|_1 = \|a\|_1^2, \end{aligned} \tag{S.6}$$

we derive a separable approximation $A_s(x,y) = a_s(x) a_s(y)$ to the diffuse reflectance profile $R_d(x,y)$. The *guide function* $\Gamma(x,y)$ provides the means to alter the result to provide the desired artistic effect. Since



(a) $\Gamma_p(x, y)$ (b) $\Gamma(x, y; 1.55)$

Figure S.1: Comparison of the guide function that reproduces the pre-integrated approximation (left) and our parametrized version that provides artistic control (right).

an arbitrary guide function would exhibit a huge amount of degrees of freedom, we aim to restrict its shape to a one-dimensional subspace that serves the intended effect of pronouncing either near or far scattering.

As a starting point, we tried to determine the guide function that would reproduce the pre-integrated approximation, i.e., we numerically calculated a guide function $\Gamma_p(x, y)$ such that

$$a_p = \underset{a}{\operatorname{argmin}} \int_{\mathbb{R}^2} \Gamma_p(x, y)\big(R_d(x, y) - a(x)a(y)\big)^2 \, dx \, dy \quad \text{subject to } \|R_d\|_1 = \|a\|_1^2, \tag{S.7}$$

where a_p is the pre-integrated approximation of the previous Section C. As can be seen in Figure S.1a, such a guide function exhibits a rather complicated structure. To reduce the associated complexity down to a single parameter, we empirically defined a parametrized guide function

$$\Gamma(x, y; k) = \big(x^2 + y^2\big)^{k/2} (1 - e^{-bx^2})(1 - e^{-by^2}). \tag{S.8}$$

The radially symmetric polynomial term with exponent k serves as external parameter to either emphasize near or far scattering. A depiction of this function can be found in Figure S.1b for $k = 1.55$. Note that this polynomial term reproduces the behavior of $\Gamma_p(x, y)$ on the diagonals (while ignoring the minor normalization-dependent details in the corners). To reproduce the shape of Γ_p along the coordinate axes, we add 'suppression' functions in the form of $1 - e^{-bx^2}$ and $1 - e^{-by^2}$, which cause the guide function to vanish close to the coordinate axis. We chose $b = 50$ (assuming that $x, y \in [-1, 1]$), and the whole framework is insensitive to moderate changes of this parameter. Varying the parameter k in the range from 0 to 4 allows the user to generate a separable approximation (as a result of Equation S.6) that emphasizes near scattering (towards $k = 0$) or far scattering (towards $k = 4$). Note that no appreciable visual effects are observable beyond $k = 4$. We have used sequential quadratic programming to minimize Equation S.6. To avoid taking on a high-dimensional optimization problem immediately, we solve Equation S.6 for low-resolution versions of the final kernel. Each optimization is then initialized with an interpolation of the solution to the next-lower resolution.

E Shader Implemention Details

As our technique works in screen space, the size of the kernel is a function of the projected surface area in the pixel, which depends on depth and surface orientation. This area is typically specified in world-space units instead of pixels, making the definition of the kernel size more intuitive for artists. When using a discretized kernel, the effect of the surface orientation can be taken into account by using ad-hoc correction factors [JSG09].

In contrast, using the simplified two-Gaussian artist-friendly model A_m allows us to work on real-world distances: we transform the depth of the pixel being evaluated and of each sample to world space,

and then calculate the distance d between them. This distance is used to apply the profile on the fly, yielding more accurate results than using derivatives or ad-hoc correction factors. However, this approach has two problems: (1) we cannot accurately bake the kernel weights by evaluating the area covered by each sample (so we can only evaluate the kernel in the sample position), and (2) the number of slots per sample used by the GPU (as generated by DirectX 11 `fxc`) increases from 16 in a simple ad-hoc correction technique to 28, almost halving the performance, making it useless in production scenarios. The latter problem has two reasons: a) converting from depth and pixel position to world-space requires a few additional ALU instructions; b) evaluating a 2-Gaussian RGB profile requires 6 `exp` instructions, which are extremely costly to execute even on modern GPUs.

We solve these problems by following an approach similar to the one proposed by Mikkelsen [Mik10], splitting the 1D profile application on 3D distances d to 2D distances d_{xy} with an accurate depth d_z correction factor. This transforms the evaluation of the kernel as:

$$\begin{aligned} a_m\left(\sqrt{d_{xy}^2 + d_z^2}\right) &= w\,G\left(\sqrt{d_{xy}^2 + d_z^2}, \tau_{near}\right) + (1-w)\,G\left(\sqrt{d_{xy}^2 + d_z^2}, \tau_{far}\right) \\ &= w\,e^{\frac{-d_z^2}{2\tau_{near}}} G(d_{xy}, \tau_{near}) \;+\; (1-w)\,e^{\frac{-d_z^2}{2\tau_{far}}} G(d_{xy}, \tau_{far}) \\ &\approx e^{\frac{-d_z^2}{2\tau_{max}}} \left(w\,G(d_{xy}, \tau_{near}) \;+\; (1-w)\,G(d_{xy}, \tau_{far})\right) \\ &= e^{\frac{-d_z^2}{2\tau_{max}}}\, a'_m(d_{xy}), \end{aligned}$$

where $\tau_{max} = max(\tau_{near}, \tau_{far})$. Note that we are making the approximation of taking the maximum variance of the Gaussians, which simplifies the profile application to an accurate depth correction, versus the ad-hoc corrections used in e.g. [JSG09]. This allows us to pre-compute accurate weights for $a'_m(d_{xy})$ using area integration, and reduce the number of instructions to 16 by: a) reducing the number of `exp` from 6 to a single one; b) avoiding the conversion to world space by directly working with depths; c) applying typical low-level optimizations [Per14].

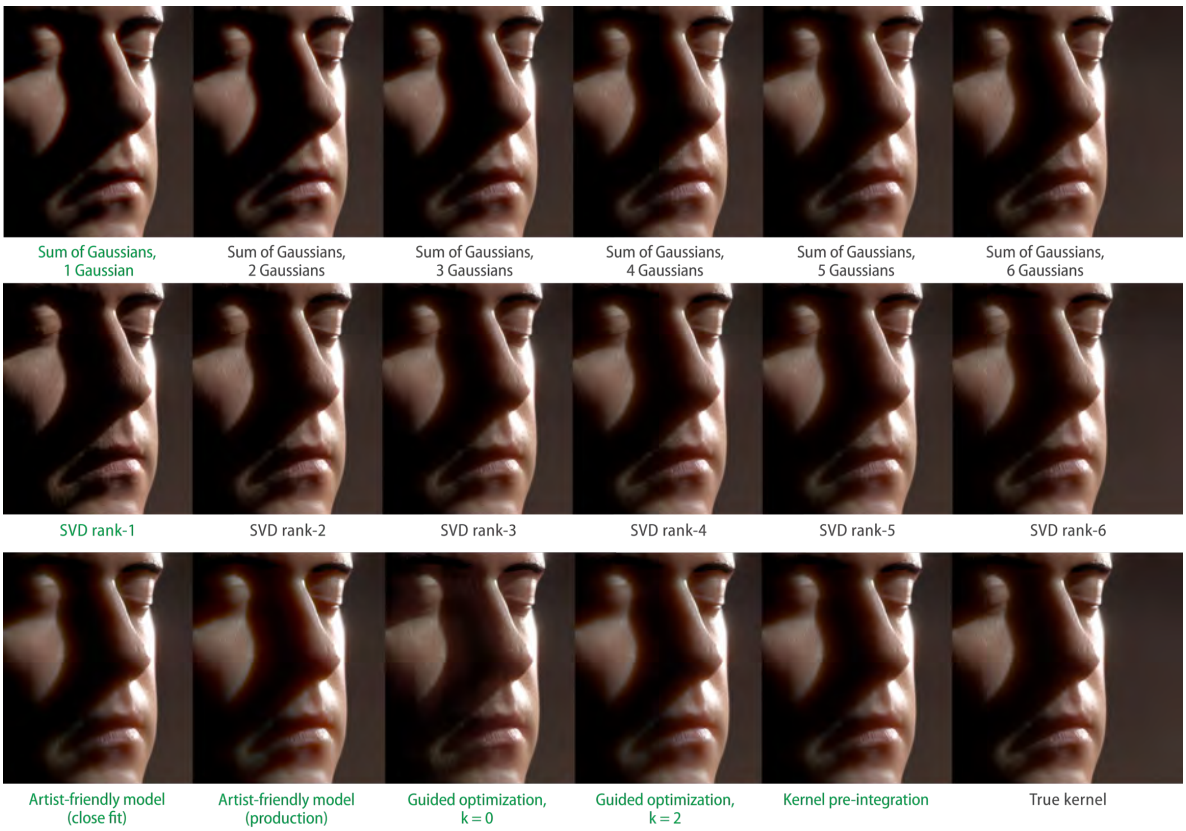


Figure S.2: Comparison of the different techniques proposed with d'Eon's method [dLE07] with six Gaussians, and the ground truth (actual 2D kernel). Note that our separable approximations lead to similar quality results with just two 1D convolutions, as opposed to the twelve needed by the sum of Gaussians approach. Note that all separable rank-1 kernels are highlighted with green.

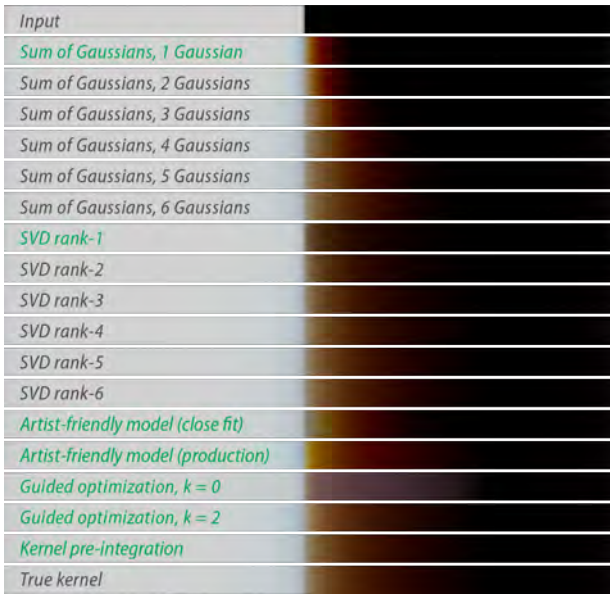


Figure S.3: Our results reveal that SVD-based low-rank approximations scale better with the number of convolutions than the state of the art [dLE07], however, they still yield only a coarse approximation of the true kernel in the separable case (rank-1). Note that all separable rank-1 kernels are highlighted in green. The pre-integrated kernel is exact for axis-aligned functions, such as this example. The guided optimization with $k = 2$ provides a comparably good fit, while the $k = 0$ case captures the fine details near the boundary at the expense of the far-range scattering quality. A manual approximation using our artist-friendly model is illustrated by two approximations, where for the first (close fit), perceptual similarity to the ground truth was the modeling objective, while the second (production) was tweaked by an artist for production purposes. Additional 2D visualizations of the different kernels are provided in the second supplementary document, where we also demonstrate that the radial asymmetry of the SVD-based kernels vanishes rapidly with increasing rank.

Figure S.4: Real-time results for *apple*. The insets show (from top to bottom) input irradiance, d'Eon et al. [dLE07] with 1 Gaussian, our analytic kernel pre-integration technique and the ground truth. Both d'Eon's with one Gaussian and ours are run with the same number of convolutions, thereby yielding similar execution times.

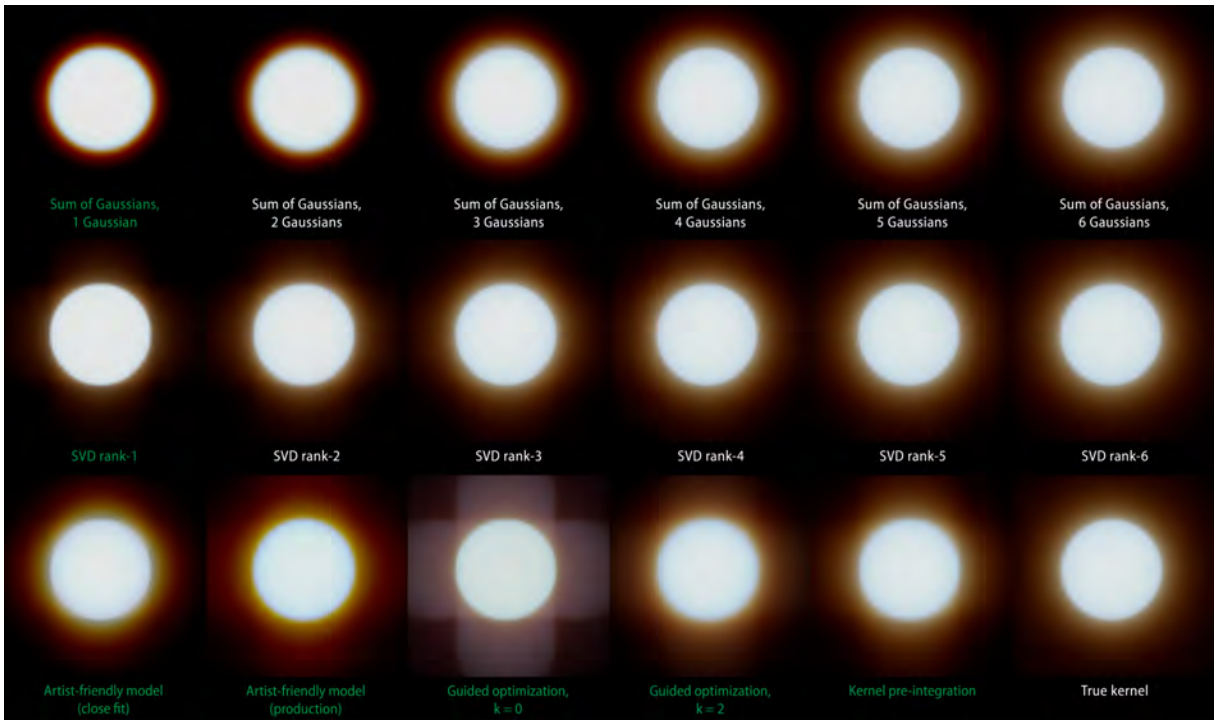
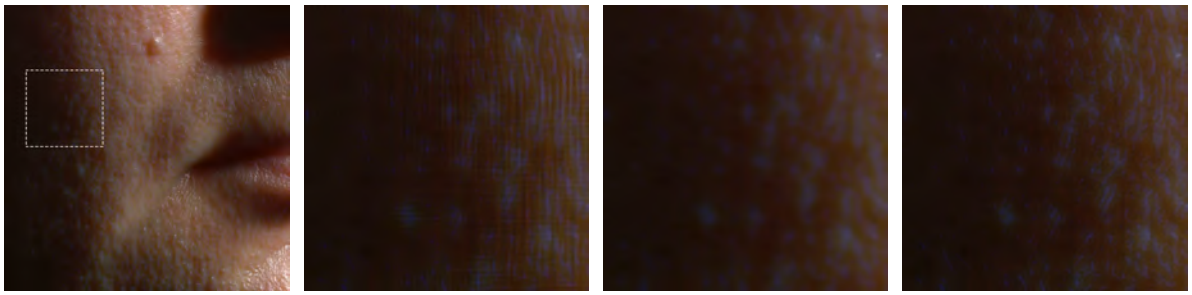


Figure S.5: Additional results reveal the inherent shape and quality of the separable and low-rank approximations on a white disk irradiance signal. Please note that, although the radial asymmetry of our separable kernels and the gradient-color differences of the manual approximations are noticeable in case of the artificial 'dot' illumination, these artifacts are less noticeable in case of our practical rendering examples.



(a) (b) 13 (Non-Jitt.) (c) 65 (Non-Jitt.) (d) 13 (Jittered)

Figure S.6: In harsh lighting conditions, extreme close-ups may reveal artifacts even in the presence of importance sampling. (a) Initial image; (b) Importance sampling with 13 samples shows banding artifacts (please zoom in in the digital version for a better view); (d) Up to 65 samples are needed to eliminate visible banding; (c) Our jittering approach also eliminates banding while keeping the sample count low (this image is best viewed in the digital version).

(a) Without jittered sampling (b) With jittered sampling (33%) (c) Ground truth

Figure S.7: This figure illustrates that our jittered sampling scheme is able to remove banding artifacts stemming from the radial asymmetry of our separable kernels. The images represent our manual approximation (close fit) of human skin, which shows visible artifacts if no jittering is used (a), but is able to approximate the ground truth (b) in a visually plausible way if 33% of the samples are jittered (c).

Figure S.8: Typical cases found in games, for which the technique is designed.

F Profile Simulation (MCML)

The diffuse reflectance profiles were simulated using MCML [WJZ95]. The material parameters used for our simulations were taken from an earlier work [JMLH01]. For each material, the RGB channels were simulated separately. Aside from the material properties, each channel used the same parameters specified as follows:

No. Photons: 10^7

Grid spacing: the minimum mean free path divided by 20 ($d_r = \frac{min(MFP_{rgb})}{20}$).

No. of grid elements: 32 times the maximum mean free path divided by d_r ($n_r = \lceil \frac{32 * max(MFP_{rgb})}{d_r} \rceil$).

Thickness: 10^7 cm (quasi-infinite).

The mean free path is computed as $\frac{1}{\sigma_t'} = \frac{1}{\sigma_a + \sigma_s'}$. As we model isotropic scattering ($g = 0$), the reduced scattering coefficient is trivial, i.e., $\sigma_s' = \sigma_s$. For the use with MCML, all parameters are also converted to cm, cm^{-1} or cm^{-2} respectively. MCML simulates cylindrically symmetric tissue models, and outputs the diffuse reflectance profile as a 1D function, $R_d(r)$. Detailed material parameters, plots of the simulated profiles as well as the derived radially symmetric 2D profiles are included in Section G.

G Simulation Parameters and Results

This section includes the properties of each measured material, simulated 1D profile plots and derived 2D profiles for each material. This gives a detailed description of the *input* to our method. Due to their extensive size, the *output* of our approximation methods to these inputs can be found in a separate supplementay material. The 1D profiles shown in the first two subplots are weighted by $2\pi r$ because the magnitude of the profile at a certain r value represents the reflectance on a complete circle of the radially symmetric 2D profile, whose area increases as r is increasing. It is worth noting that although all three channels were simulated up to the same maximum radius, the 1D profile plot may falsely suggest otherwise. In the case of the logarithmic scale plots, all profile values near machine precision are omitted by the plot function. In the MFP plots, all channel radii are scaled by the their respective MFP values, and therefore each channel has different support on the x axis. Additionally, all MFP plots show the unweighted profiles, and the 2D plots visualize only the most significant center region.

G.1 Apple

Apple	σ_a (mm^{-1}) [RGB]	σ_s (mm^{-1}) [RGB]	η	g	Thickness (cm)
Layer 1	[0.0030, 0.0034, 0.0460]	[2.2900, 2.3900, 1.9700]	1.3000	0.0000	10^7

Figure S.9: Apple - Scattering properties

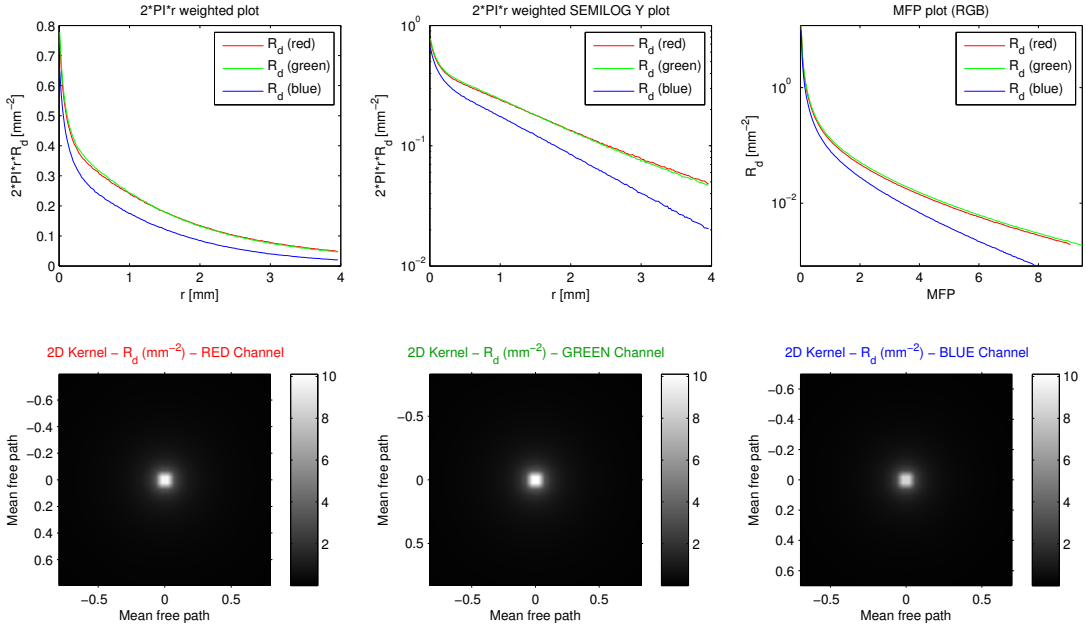


Figure S.10: Apple - Simulation

G.2 Chicken1

Chicken1	σ_a (mm^{-1}) [RGB]	σ_s (mm^{-1}) [RGB]	η	g	Thickness (cm)
Layer 1	[0.0150, 0.0770, 0.1900]	[0.1500, 0.2100, 0.3800]	1.3000	0.0000	10^7

Figure S.11: Chicken1 - Scattering properties

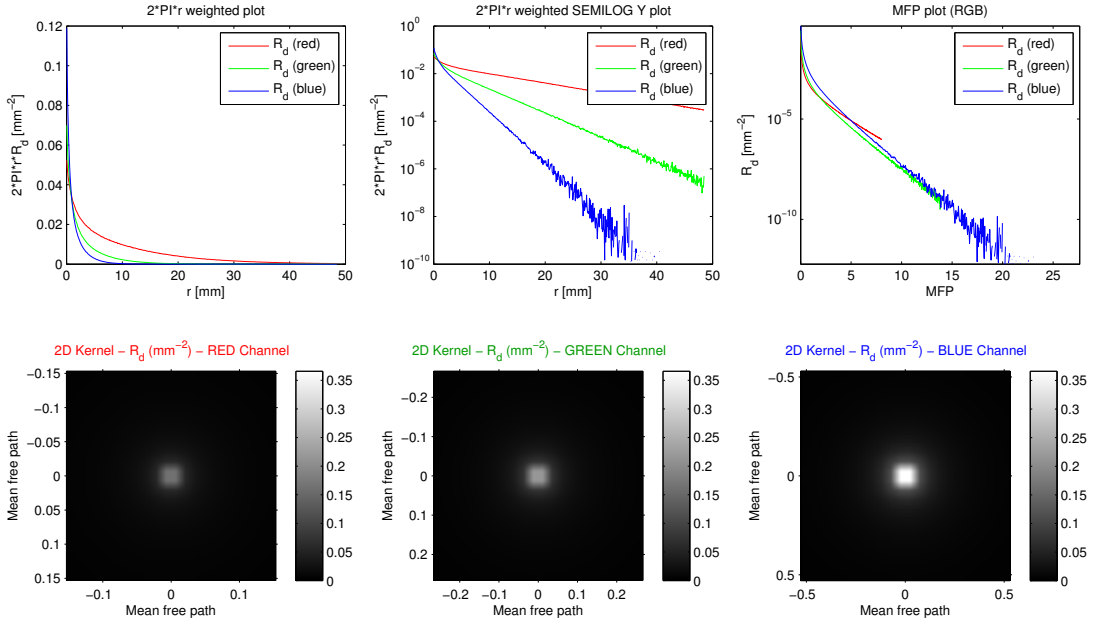


Figure S.12: Chicken1 - Simulation

G.3 Chicken2

Chicken2	σ_a (mm^{-1}) [RGB]	σ_s (mm^{-1}) [RGB]	η	g	Thickness (cm)
Layer 1	[0.0180, 0.0880, 0.2000]	[0.1900, 0.2500, 0.3200]	1.3000	0.0000	10^7

Figure S.13: Chicken2 - Scattering properties

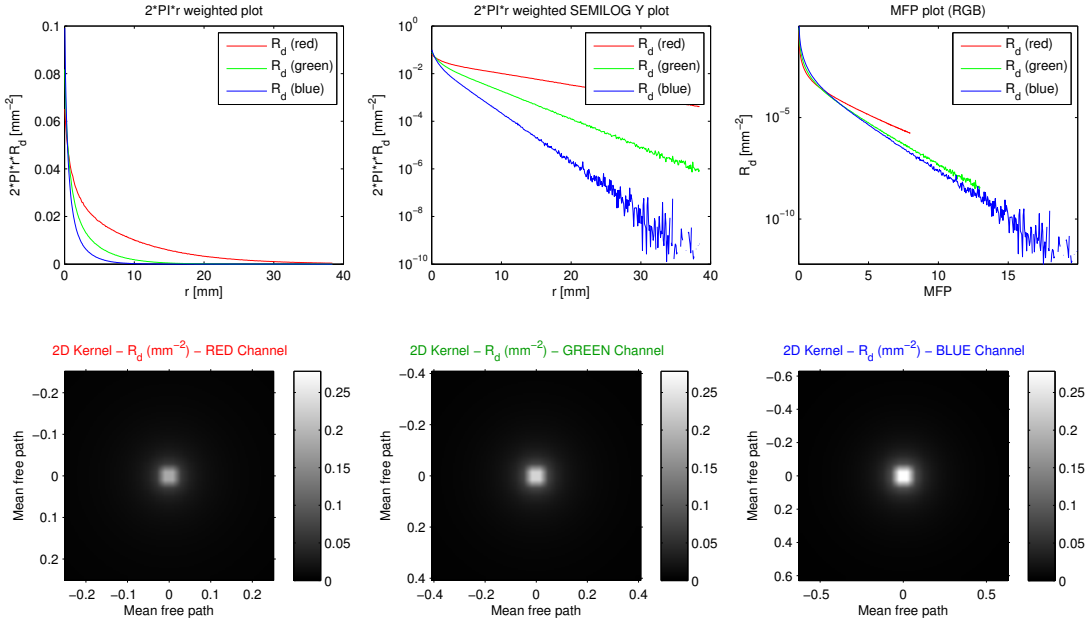


Figure S.14: Chicken2 - Simulation

G.4 Cream

Cream	σ_a (mm^{-1}) [RGB]	σ_s (mm^{-1}) [RGB]	η	g	Thickness (cm)
Layer 1	[0.0002, 0.0028, 0.0163]	[7.3800, 5.4700, 3.1500]	1.3000	0.0000	10^7

Figure S.15: Cream - Scattering properties

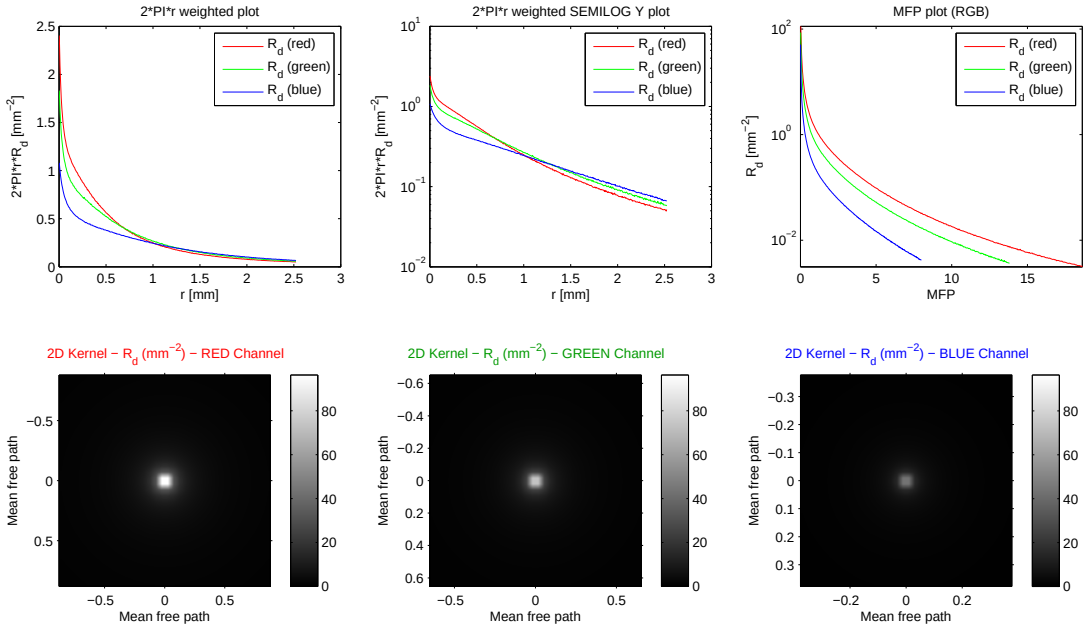


Figure S.16: Cream - Simulation

G.5 Ketchup

Ketchup	σ_a (mm^{-1}) [RGB]	σ_s (mm^{-1}) [RGB]	η	g	Thickness (cm)
Layer 1	[0.0610, 0.9700, 1.4500]	[0.1800, 0.0700, 0.0300]	1.3000	0.0000	10^7

Figure S.17: Ketchup - Scattering properties

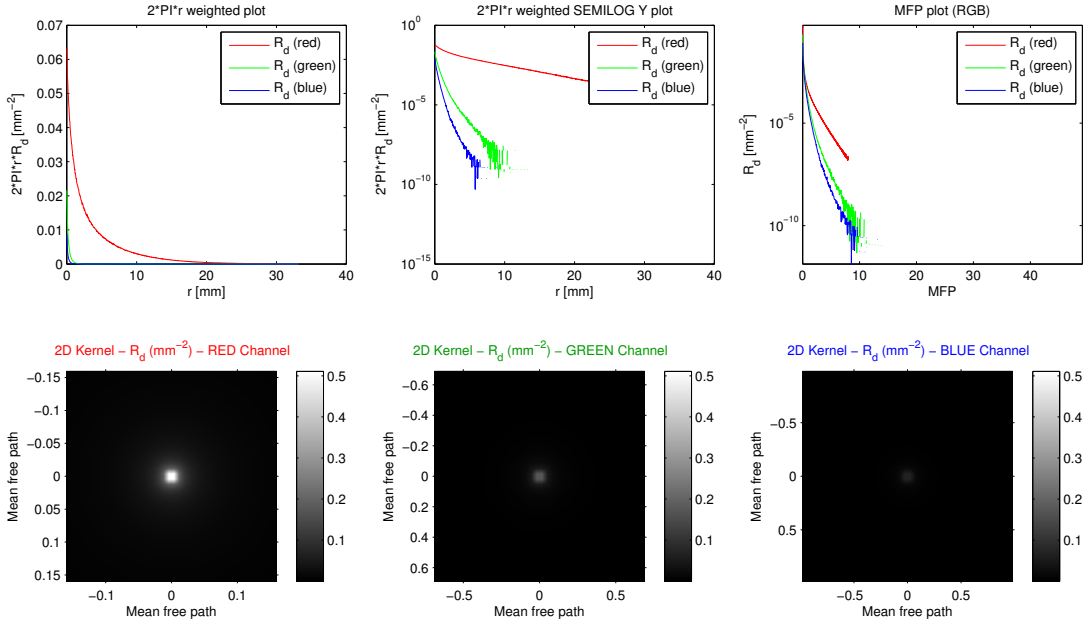


Figure S.18: Ketchup - Simulation

G.6 Marble

Marble	σ_a (mm^{-1}) [RGB]	σ_s (mm^{-1}) [RGB]	η	g	Thickness (cm)
Layer 1	[0.0021, 0.0041, 0.0071]	[2.1900, 2.6200, 3.0000]	1.5000	0.0000	10^7

Figure S.19: Marble - Scattering properties

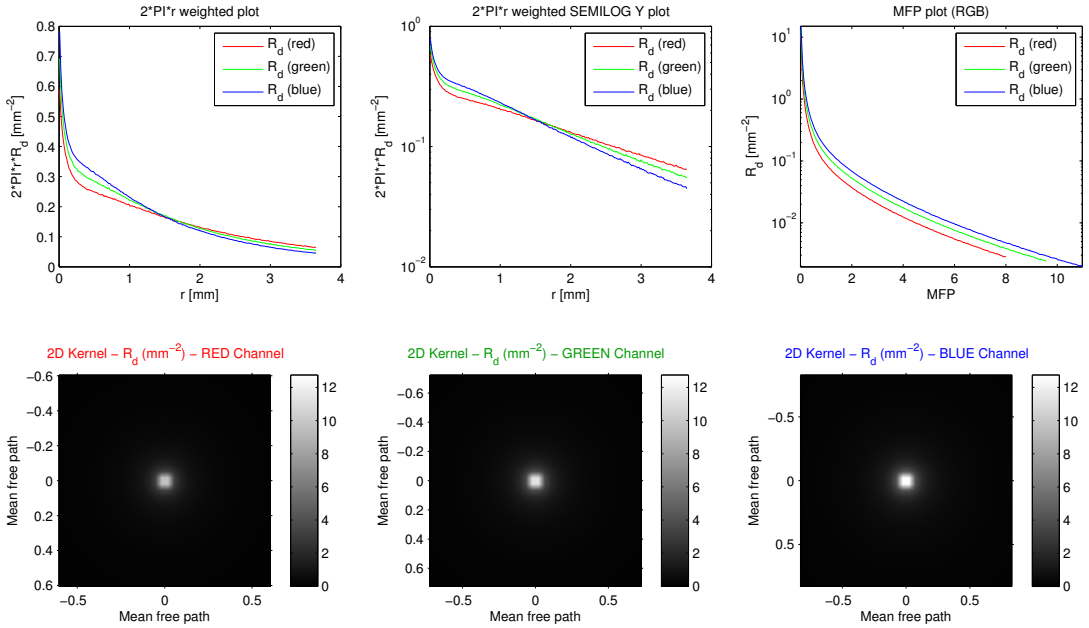


Figure S.20: Marble - Simulation

G.7 Potato

Potato	σ_a (mm^{-1}) [RGB]	σ_s (mm^{-1}) [RGB]	η	g	Thickness (cm)
Layer 1	[0.0024, 0.0090, 0.1200]	[0.6800, 0.7000, 0.5500]	1.3000	0.0000	10^7

Figure S.21: Potato - Scattering properties

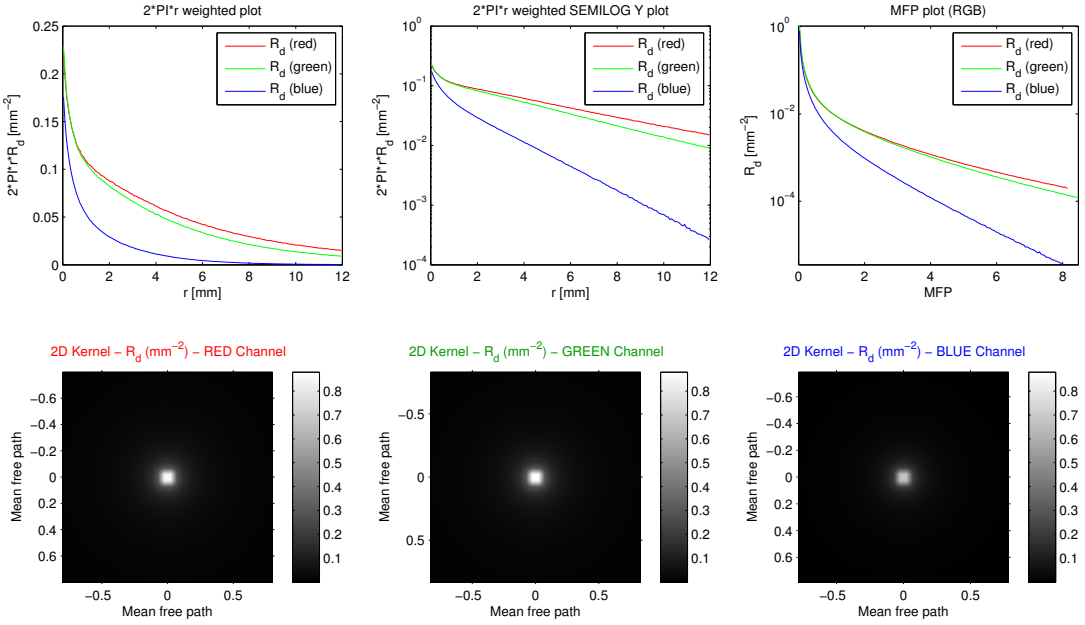


Figure S.22: Potato - Simulation

G.8 Skimmilk

Skimmilk	σ_a (mm^{-1}) [RGB]	σ_s (mm^{-1}) [RGB]	η	g	Thickness (cm)
Layer 1	[0.0014, 0.0025, 0.0142]	[0.7000, 1.2200, 1.9000]	1.3000	0.0000	10^7

Figure S.23: Skimmilk - Scattering properties

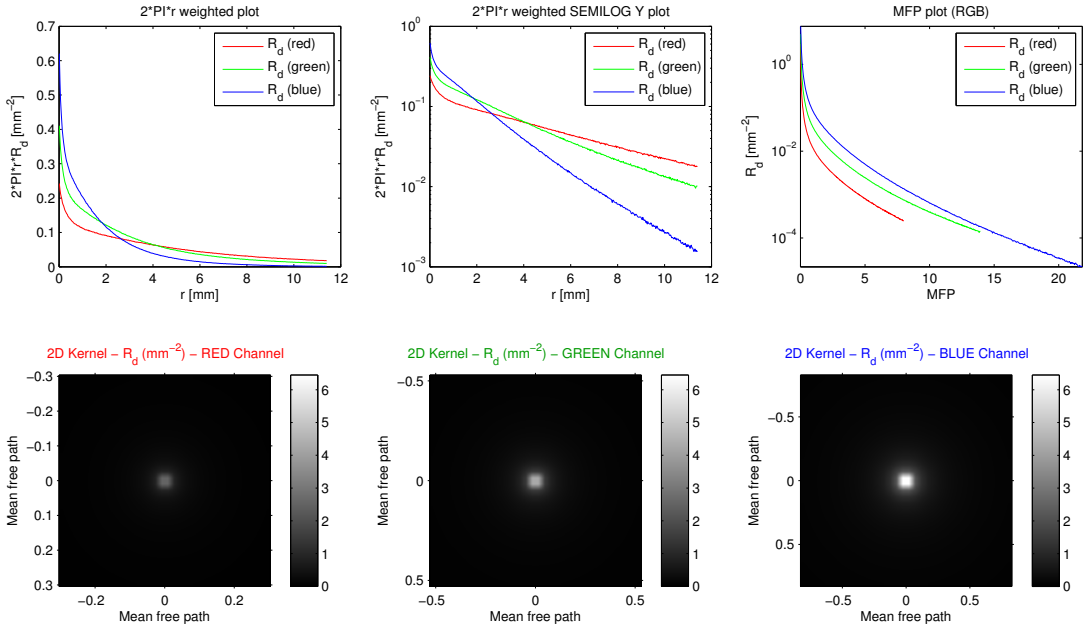


Figure S.24: Skimmilk - Simulation

G.9 Skin1

Skin1	σ_a (mm^{-1}) [RGB]	σ_s (mm^{-1}) [RGB]	η	g	Thickness (cm)
Layer 1	[0.0320, 0.1700, 0.4800]	[0.7400, 0.8800, 1.0100]	1.3000	0.0000	10^7

Figure S.25: Skin1 - Scattering properties

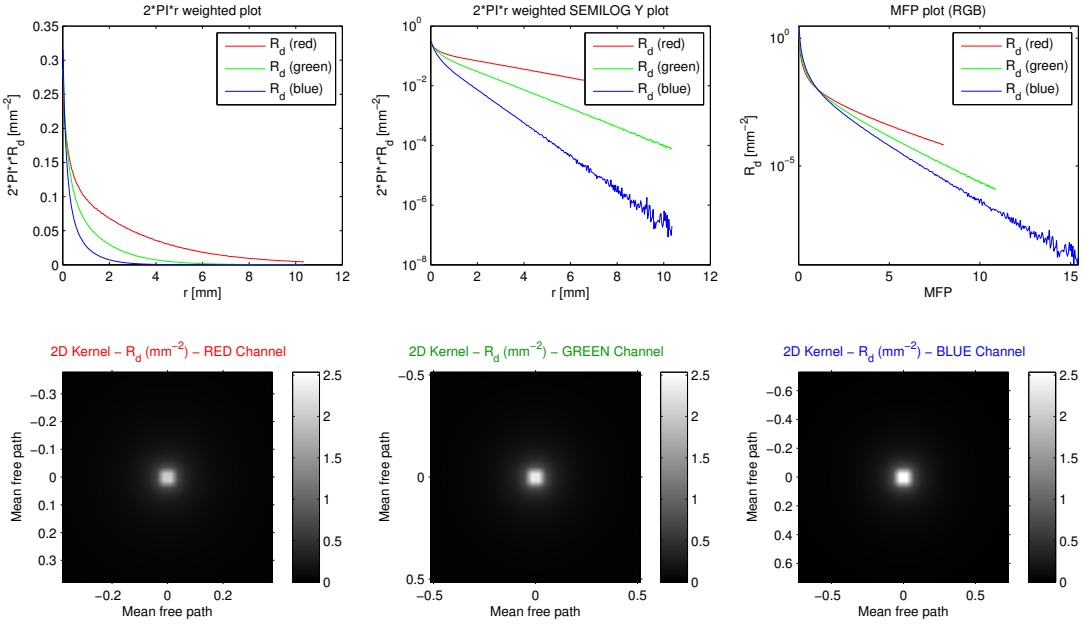


Figure S.26: Skin1 - Simulation

G.10 Skin2

Skin2	σ_a (mm^{-1}) [RGB]	σ_s (mm^{-1}) [RGB]	η	g	Thickness (cm)
Layer 1	[0.0130, 0.0700, 0.1450]	[1.0900, 1.5900, 1.7900]	1.3000	0.0000	10^7

Figure S.27: Skin2 - Scattering properties

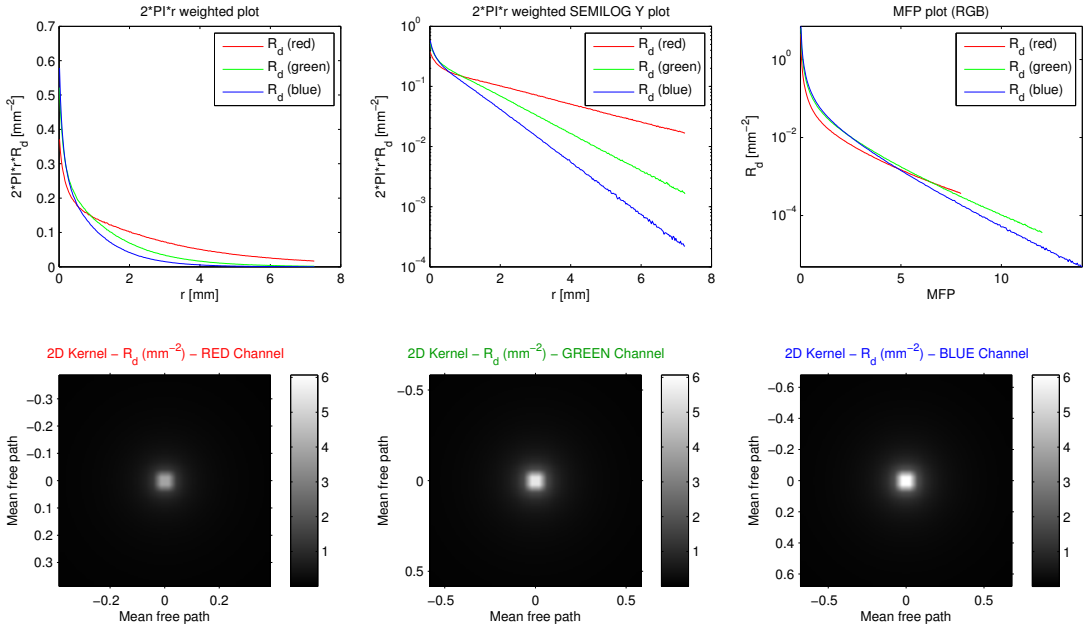


Figure S.28: Skin2 - Simulation

G.11 Wholemilk

Wholemilk	σ_a (mm^{-1}) [RGB]	σ_s (mm^{-1}) [RGB]	η	g	Thickness (cm)
Layer 1	[0.0011, 0.0024, 0.0140]	[2.5500, 3.2100, 3.7700]	1.3000	0.0000	10^7

Figure S.29: Wholemilk - Scattering properties

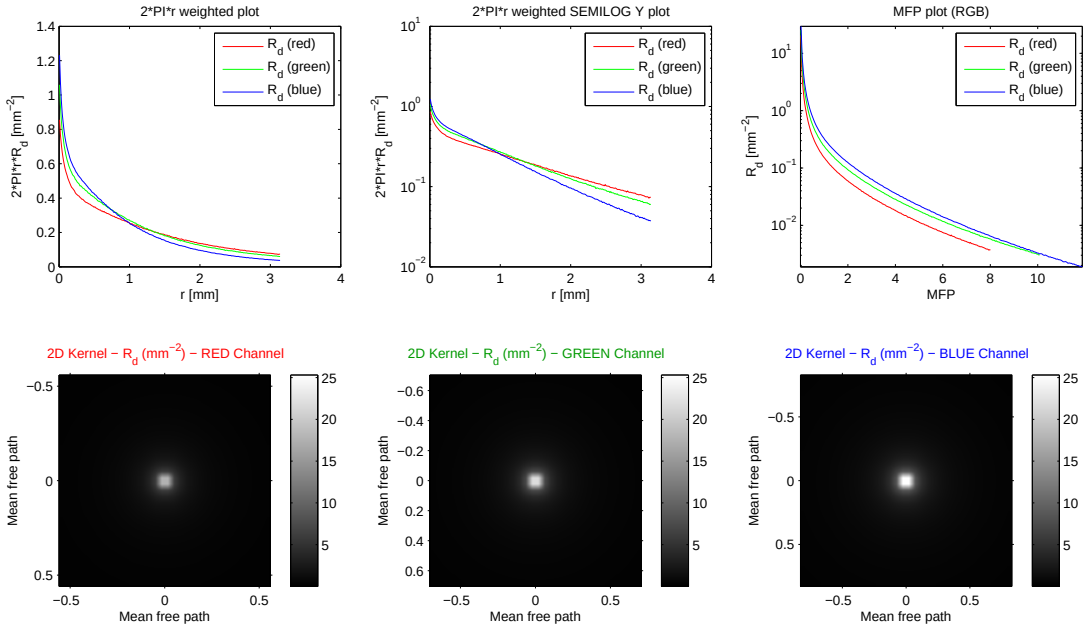


Figure S.30: Wholemilk - Simulation

References

[dLE07] d'Eon E., Luebke D., Enderton E.: Efficient rendering of human skin. In *Proceedings of Eurographics Symposium on Rendering* (2007), pp. 147–157.

[EY36] Eckart C., Young G.: The approximation of one matrix by another of lower rank. *Psychometrika 1*, 3 (1936), 211–218.

[JMLH01] Jensen H., Marschner S., Levoy M., Hanrahan P.: A practical model for subsurface light transport. In *Proceedings of ACM SIGGRAPH 2001* (2001), 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), 1–15.

[KM99] Kautz J., McCool M. D.: Interactive rendering with arbitrary brdfs using separable approximations. In *Rendering Techniques 99*. Springer, 1999, pp. 247–260.

[LRR04] Lawrence J., Rusinkiewicz S., Ramamoorthi R.: Efficient brdf importance sampling using a factored representation. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 496–505.

[Mik10] Mikkelsen M.: *Skin Rendering by Pseudo-Separable Cross Bilateral Filtering*. Tech. rep., Naughty Dog Inc., August 2010.

[Per14] Persson E.: Low-level shader optimization for next-gen and DX11. In *Game Developers Conference* (2014).

[PvBM*06] Peers P., vom Berge K., Matusik W., Ramamoorthi R., Lawrence J., Rusinkiewicz S., Dutré P.: A compact factored representation of heterogeneous subsurface scattering. In *ACM Transactions on Graphics (TOG)* (2006), vol. 25, ACM, pp. 746–753.

[WJZ95] Wang L., Jacques S. L., Zheng L.: Mcml – monte carlo modeling of light transport in multi-layered tissues. *Computer methods and programs in biomedicine 47*, 2 (1995), 131–146.