Real-Time Subsurface Scattering: A Comparative Analysis of Approximation Techniques

Bennett Poh Rui Hao
bennettpoh@gmail.com
Technical University of Munich
Munich, Bavaria, Germany

Abstract

Subsurface scattering (SSS) is essential for achieving a more convincing and lifelike appearance of materials like skin, jade, marble, wax, or milk. However, modelling SSS accurately is computationally expensive and impractical for real-time rendering for games and other interactive applications. This paper investigates several real-time approaches to approximate the behaviour of SSS at varying levels of visual fidelity and performance, from cheap approximations to more physically based approaches.

Rather than benchmarking specific engine implementations, the methods are evaluated in terms of their computational complexity, sampling requirements, memory usage, and qualitative visual behaviour. Visual fidelity is analysed with respect to characteristic subsurface effects such as light diffusion, colour bleeding, and edge softness, while performance is discussed through the structure and cost of the underlying algorithms.

The paper concludes with a comparison of the strengths, limitations, and practical use cases of each approach, showing that real-time approximations can achieve visually convincing SSS at a fraction of the cost of physically based methods, while remaining fundamentally distinct from their offline counterparts.

CCS Concepts

• **Computing methodologies** → **Reflectance modelling**; **Rasterization**.

Keywords

Subsurface scattering, Separable Subsurface Scattering, BSSRDF, Christensen–Burley, Real-time rendering, Translucency

ACM Reference Format:
Bennett Poh Rui Hao. 2025. Real-Time Subsurface Scattering: A Comparative Analysis of Approximation Techniques. In *How to make a PIXAR Movie Seminar — Rendering with a Focus on Subsurface Scattering (WS25).* ACM, New York, NY, USA, 6 pages.

1 Introduction

Realistic rendering in computer graphics depends on accurately modelling how light interacts with different materials. While most shading models describe surface reflection, materials such as skin, jade, marble, wax, or milk allow light to penetrate beneath the surface, scattering multiple times before leaving at a nearby point. This is called *subsurface scattering*. The effect is responsible for the soft and natural look of translucent materials and is essential for convincing digital imagery.

How to make a PIXAR Movie (WS25), Munich, Germany
2025.

A widely adopted physically based SSS model was proposed by Jensen et al. [8]. The model estimates light diffusion by sampling irradiance across the surface and storing it in a point cloud. This strategy was later adopted in offline renderers such as Pixar's RenderMan, though its high sampling and integration cost makes it unsuitable for real-time use.

To overcome the limitations of offline diffusion models, various real-time techniques have been developed to reproduce the appearance of SSS more efficiently. This study focusses on three representative approaches: *a simple translucency approximation* [1], *separable subsurface scattering* [10], and *the Christensen–Burley normalised diffusion model* [4], listed from simplest to most physically accurate.

The remainder of this paper is organised as follows: Section 2 introduces the physical principles underlying SSS. Section 3 presents the selected real-time techniques in more detail. Section 4 evaluates the visual fidelity and computational cost of each approach, discussing their respective advantages, limitations, and intended use cases. Section 5 concludes with a summary and discussion.

2 Understanding Subsurface Scattering

Before discussing the real-time techniques in this study, it is important to understand the physical basis of SSS and why it poses a challenge for real-time rendering.

2.1 Light Transport in Dielectrics

When light strikes a surface, part of it is reflected immediately, while another portion penetrates beneath the surface and undergoes absorption and scattering. The relative amount of scattering across different wavelengths determines the material's colour.

All dielectric materials exhibit SSS to some degree. In opaque materials, the scattering distances are extremely short and can be considered negligible for practical rendering purposes, meaning that light exits at the point of entry [9, Sec. 2.4]. For these materials, local shading models such as Lambertian or Oren–Nayar reflection approximate this behaviour well.

In translucent materials, scattering distances become larger than a pixel footprint (Figure 1). This allows light to travel between neighbouring surface regions before re-emerging, producing soft shadows and colour bleeding that cannot be captured by local shading models.

2.2 The BSSRDF

SSS can be mathematically described by a *Bidirectional Surface Scattering Reflectance Distribution Function* (BSSRDF):

$$S(x_i, \omega_i; x_o, \omega_o) = C\, F_t(x_i, \omega_i)\, R(|x_o - x_i|)\, F_t(x_o, \omega_o),$$

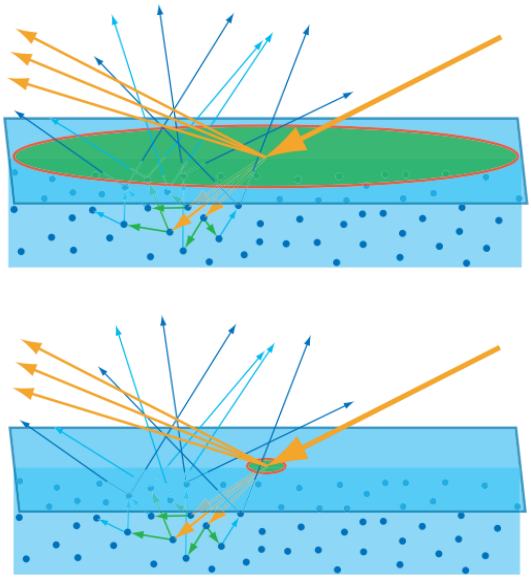

Figure 1: Illustration of SSS scale relative to a pixel. The green circle represents the shading area for a single pixel. Top: Local scattering allows local shading models. Down: Scattering exceeds pixel footprint, requiring SSS methods. *Images from Hoffman [7].*

where $R(r)$ is a radially symmetric *diffuse reflectance profile* that models the spatial redistribution beneath the surface as a function of surface distance r. The Fresnel transmission terms F_t describe the angle-dependent amount of light that enters and exits the surface, while C ensures energy normalisation.

Unlike the BRDF, which assumes light enters and exits the surface at the same point, the BSSRDF, accounts for light entering at one point and exiting at a different point. This can be formally defined as:

$$L_o(\mathbf{x}_o, \boldsymbol{\omega}_o) = \int_A \int_{2\pi} S(\mathbf{x}_i, \boldsymbol{\omega}_i; \mathbf{x}_o, \boldsymbol{\omega}_o)\, L_i(\mathbf{x}_i, \boldsymbol{\omega}_i)\, (\mathbf{n}_i \cdot \boldsymbol{\omega}_i)\, d\boldsymbol{\omega}_i\, dA(\mathbf{x}_i).$$

This equation tells us that the outgoing radiance at a point $\mathbf{x}_o$ and direction $\boldsymbol{\omega}_o$ is given by integrating contributions from all incoming surface points and directions.

While physically accurate, this formulation is unwieldy even for offline rendering, so subsurface transport is approximated using diffusion models that assume light becomes nearly isotropic after multiple scattering events [9, Sec. 2.1]. These diffusion-based approximations form the basis of most practical SSS models used in both offline and real-time rendering.

3 Real-Time Subsurface Scattering Techniques

Despite the impressive results that offline techniques deliver, such approaches are still too expensive for real-time applications like video games. Consequently, practical rendering pipelines must adopt additional simplifications of the BSSRDF. In this section, we explore several popular real-time techniques that trade physical accuracy for speed while still retaining the visual fidelities that players expect.

3.1 Simple Translucency Approximation (STA)

In GDC 2011, Barré-Brisebois and Marc Bouchard from EA presented a cheap and quick approximation of translucency that produced a convincing SSS look [1]. Rather than working with additional depth or texture maps that require significant computation, they stripped SSS down to the minimum visual features that actually sell the effect. There was therefore no need for the solution to reflect the entire BSSRDF accurately because humans do not perceive most of that complexity.

Their analysis identified two dominant cues. First, the light travelling inside the object must respond to the *local material thickness*, since thicker regions absorb and attenuate more light. Second, this internal light must exhibit some form of *angularly dependent spreading*, which gives translucent materials their characteristic soft, glowing appearance.

Estimating thickness per-pixel was solved by a clever re-use of *ambient occlusion* (AO). Standard AO measures how enclosed a surface point is by sampling a hemisphere oriented along the surface normal. Points surrounded by nearby geometry receive higher occlusion and appear darker. By flipping the surface normal inwards, thin regions lying close to the opposite side of the surface appear highly occluded, while thicker regions have more internal space and produce lower occlusion values. The result is then inverted to adhere to artistic conventions, assigning high values to thin regions and lower values to thick ones.

Translucency is then approximated using a view-dependent back-lighting term modulated by local thickness. For a surface point $\mathbf{x}$ with normal $\mathbf{N}$, light direction $\mathbf{L}$ and view direction $\mathbf{V}$, back-lighting is given by:

$$w_{\text{back}}(\mathbf{x}) = \max\big(0,\ \mathbf{V} \cdot (-\mathbf{L})\big).$$

To break uniformity and accentuate silhouettes, the light direction is distorted using the surface normal:

$$\mathbf{L}_T = \frac{\mathbf{L} + d_T\, \mathbf{N}}{\|\mathbf{L} + d_T\, \mathbf{N}\|},$$

and raised to a power p_T, yielding:

$$T_{\text{dir}}(\mathbf{x}) = s_T \left[\max\big(0,\ \mathbf{V} \cdot (-\mathbf{L}_T)\big)\right]^{p_T}.$$

The final translucency term applies thickness $t(\mathbf{x})$, light attenuation $A_L(\mathbf{x})$ and ambient offset a_T to T_{dir}:

$$T(\mathbf{x}) = A_L(\mathbf{x})\, t(\mathbf{x})\, [\, T_{\text{dir}}(\mathbf{x}) + a_T\,],$$

which can then be added to the surface's diffuse contribution to produce the final colour.

3.2 Separable Subsurface Scattering

A more physically based approach is *separable subsurface scattering* (SSSS) [10], which was developed as a solution to poor scalability of an older real-time approach: *texture-space diffusion* [5] that operates on a per-object basis. To overcome this, Jimenez et al. developed a technique that computes SSS in screen-space using a pre-integrated 1D diffusion kernel that reproduces the behaviour of the original 2D profile under practical lighting conditions, reducing the convolution to two cheaper 1D passes.

Figure 2: STA based on thickness-modulated light transmission. Left: reference illustration from Barré-Brisebois and Hill [1]. Right: author's implementation in Unity URP (HLSL).

3.2.1 Mathematical Foundations. The outgoing irradiance due to SSS is expressed as the convolution between the surface irradiance $E(x, y)$ and the diffuse reflectance profile $R_d(x, y)$, which is precomputed offline for each material.

$$M_e(x, y) = (E * R_d)(x, y) = \int_{\mathbb{R}^2} E(x', y')\, R_d(x - x',\ y - y')\, dx' dy'.$$

For homogeneous translucent materials, the R_d is symmetric and can be simplified to $R_d(x, y) = R_d(r)$ with $r = \|(x, y)\|$.

Naturally, this 2D convolution is too expensive for real-time applications. Therefore, a common strategy is to find an approximation A that expresses R_d as a sum of separable functions. If R_d can be expressed as:

$$A(x, y) = \sum_{i=1}^{N} a_i(x)\, a_i(y), \tag{1}$$

the complexity is reduced from $O(n^2)$ to just $O(n)$, since the 2D convolution now simplifies to a sequence of $2N$ 1D convolutions:

$$(E * R_d)(x, y) \approx (E * A)(x, y) = \sum_{i=1}^{N} \big((E * a_i)(x)\ *\ a_i(y)\big).$$

With a suitable approximation of the diffuse reflectance profile, this can drastically improve the performance without sacrificing the visual fidelity of the original scattering behaviour.

3.2.2 Pre-integrated Kernel. After several failed attempts to find a suitable approximation for the high-rank diffuse reflectance profile, Jimenez et al. shifted their focus from the kernel to the input. They reconsidered the problem from a different angle. Rather than forcing a high-rank kernel, they observed that screen-space irradiance is often very simple, typically consisting of smooth lighting gradients or shadow boundaries that have mere 1D variations. It is therefore possible to assume that irradiance is additively separable, since lighting patterns are merely strips of increasing/decreasing brightness as we move horizontally/vertically. As a result, irradiance can be described as the combination of a function that changes along the x-axis and a function that changes along the y-axis. This motivates the approximation:

$$E(x, y) = E_1(x) + E_2(y),$$

and therefore:

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

where $a_p(x)$ is our pre-integrated 1D kernel of R_d along a coordinate axis that tells us how light scatters horizontally when we sum the diffusion kernel along the vertical direction. Likewise, $a_p(y)$ tells us how light scatters vertically when we sum along the horizontal direction.

Because R_d is radially symmetric, both integrations $a_p(x)$ and $a_p(y)$ produce the same values. Therefore, we only need to store one copy of a_p and reuse it for both horizontal and vertical passes. Another important note is that a_p contains the same total scattering energy as the full 2D diffusion kernel. We can show this by a simple change of integration order. By definition, the pre–integrated kernel is:

$$a_p(x) = \int R_d(x, y)\, dy.$$

The total energy of a_p is its L_1 norm:

$$\|a_p\|_1 = \int a_p(x)\, dx = \int \left(\int R_d(x, y)\, dy \right) dx.$$

We can swap the order of integration (Fubini's theorem) and obtain:

$$\|a_p\|_1 = \int \int R_d(x, y)\, dx\, dy = \|R_d\|_1. \tag{2}$$

Thus, the pre–integrated 1D kernel a_p preserves the total scattering energy of the original 2D diffuse reflectance profile R_d. With that established, we can construct our separable approximation for our diffuse reflectance profile A_p by combining the horizontal and vertical falloff functions and normalising by $\|R_d\|_1$ to ensure that A_p carries the same total energy:

$$A_p(x, y) = \frac{1}{\|R_d\|_1}\, a_p(x)\, a_p(y).$$

3.2.3 Artist-friendly Separable Model. Despite reliably reconstructing the scattering behaviour of a wide range of materials given their diffuse reflectance profile, the pre-integrated kernel offers limited control in an artistic environment, since the diffuse reflectance profile may not match the art direction of the assets. To give artists more creative control over how the SSS looks, Jimenez et al. introduced the *artist-friendly separable model* that can be edited via physically meaningful parameters.

A direct parametrisation of a separable kernel, however, is normally impractical. General separable models like Equation (1) expose numerous low-level parameters: each 1D component of the

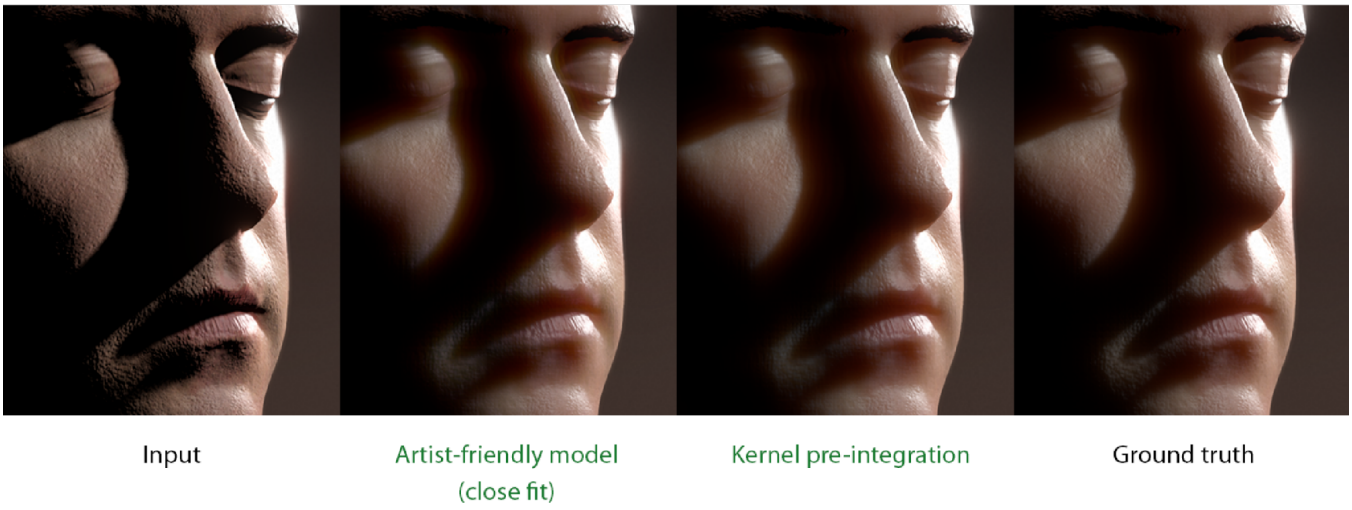

Figure 3: Visual comparison of SSSS approximations under controlled lighting, reproduced from Jimenez et al. [10]. From left to right: input irradiance, artist-friendly separable approximation, kernel pre-integration approach, and Monte Carlo ground truth.

kernel carries its own shape and scale controls, and these parameters interact in non-obvious ways when combined into a 2D diffuse reflectance profile. Adjusting one term can unintentionally alter the behaviour of the others, producing global changes to the scattering that are difficult to predict or art-direct. This makes traditional separable approximations accurate but unwieldy for artists.

To solve this, the artist-friendly separable model collapses the parameter space to a few physically meaningful controls. Jimenez et al. represent the 1D kernel as a weighted blend of a near-range Gaussian and a far-range Gaussian:

$$a_m(r) = wG(r, \sigma_{\text{near}}) + (1 - w)G(r, \sigma_{\text{far}}),$$

where w controls the balance between the short and long-range scattering, while σ_{near} and σ_{far} set the width of the near and far Gaussians respectively. The full separable kernel is simply:

$$A_m(x, y) = a_m(x)a_m(y).$$

This formulation still captures the qualitative structure of the true diffuse reflectance profile: a sharp peak from short scattering paths and a broad tail from long paths. A narrow and a wide Gaussian replicate these two features with far fewer parameters, offering the artists simpler and more intuitive controls over a physically accurate model.

3.3 Christensen–Burley's Normalised Diffusion Model

Another widely used approach is the *Christensen–Burley's normalised diffusion model* (CBNDM), introduced by Per H. Christensen and Brent Burley in 2015 [4]. Found in modern engines like Unity HDRP, Blender and RenderMan, it is often implemented in screen-space rather than volumetrically to achieve better real-time performance. To motivate the model, we first examine the diffuse reflectance profiles obtained through brute-force Monte Carlo simulation, shown in Figure 4. These curves represent the ground-truth behaviour of SSS for various surface albedos.

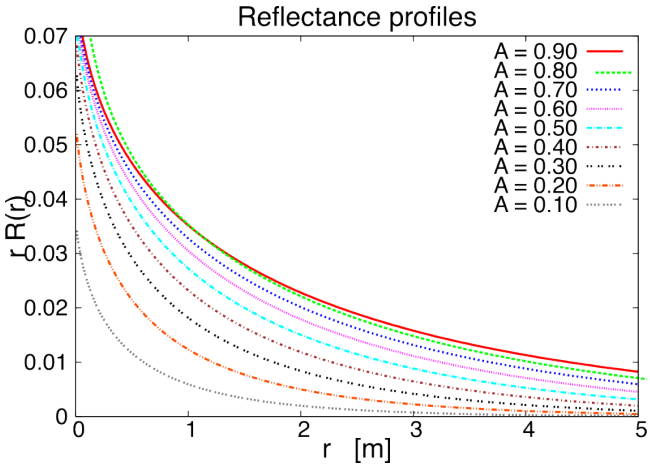

Figure 4: Monte Carlo reference reflectance profiles $rR(r)$ for a range of surface albedos.

Christensen and Burley proposed a simple closed-form approximation to these profiles, with the sum of two exponential terms divided by the radial distance:

$$R(r) = \frac{e^{-r/d} + e^{-r/3d}}{8\pi dr}. \tag{3}$$

The parameter d is the diffusion scale parameter that controls the width and falloff of the diffusion curve. A key property of this expression for $R(r)$ is that it is *normalised*, meaning that its total

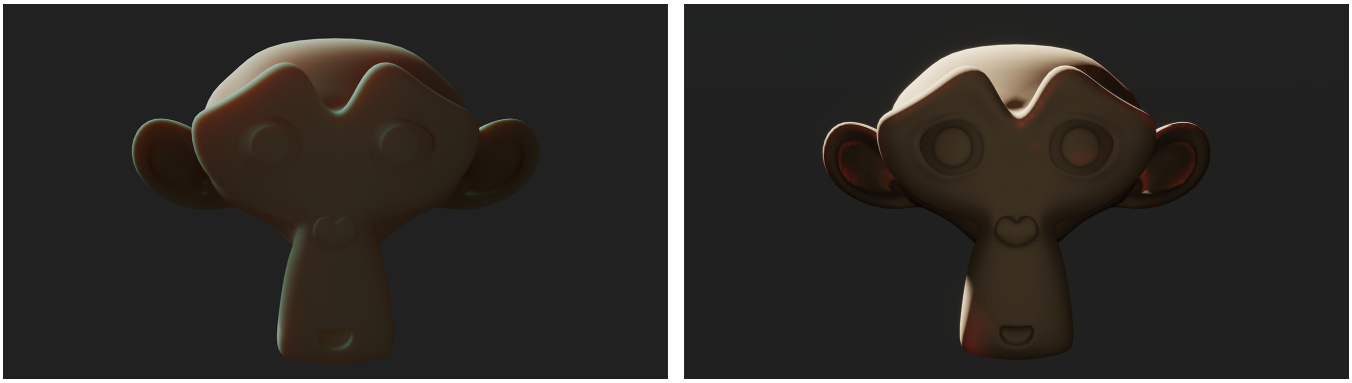

Figure 5: Screen-space SSS using CBNDM in real-time engines. Left: Blender EEVEE. Right: Unity HDRP. While both implementations are based on the same diffusion formulation, Unity HDRP [6] follows an explicitly performance-oriented design optimised for real-time frame rates, whereas Blender EEVEE [2] employs a less aggressively constrained implementation that favours visual quality within its real-time renderer.

diffuse reflectance integrates to one:

$$\int_0^\infty R(r)\, 2\pi r \, \mathrm{d}r = 1 \qquad d > 0.$$

This normalised form captures only the *shape* of the subsurface diffusion profile, independent of the material's actual reflectance.

Following Jensen and Buhler [8], as adopted in Christensen and Burley [4, Sec. 2.2], the diffuse surface albedo of a material is defined as the total scattered energy:

$$A = \int_0^\infty R_{\text{phys}}(r)\, 2\pi r \, \mathrm{d}r. \tag{4}$$

Since $R(r)$ is already normalised, the physical reflectance profile is simply obtained by scaling it with the desired albedo:

$$R_{\text{phys}}(r) = A\, R(r).$$

This guarantees that the integrated reflectance of the material is exactly A, while the parameter d controls how that energy is spatially distributed across the surface. For details on the derivation and selection of d, the reader is referred to their work [4].

Creating the convolution kernel simply involves sampling a fixed number of neighbouring pixels around a centre pixel p, arranged in a spiral defined by the golden angle. For each neighbour, the distance r from p is computed and used in Equation (3) to evaluate its contribution weight in the kernel. Convolution is then performed to calculate the final colour of pixel p.

4 Comparison of Visual Fidelities and Computational Costs

Having introduced the real-time approaches, we now compare them in terms of visual fidelity and computational cost. To enable consistent and fair comparison, we evaluate the models' visual fidelities based on a set of criteria derived from the characteristic visual effects of SSS. These criteria reflect a model's ability to reproduce such aspects of SSS independent of specific implementations. Similarly, the performance of each model will be analysed qualitatively through algorithmic structure rather than engine-specific benchmarks.

Not all referenced works explicitly evaluate each of the selected visual criteria. However, since the visual behaviour of SSS models is largely determined by their underlying assumptions and mathematical formulation, these characteristics can be inferred from the structure of each approach. The analysis therefore focusses on the effects that each model is capable of representing by construction, rather than on specific reported renderings.

Visual Fidelity. This comparison considers the following visual criteria: (1) non-local light diffusion beneath the surface, (2) distance-dependent colour bleeding, (3) physically plausible redistribution of energy between surface reflection and subsurface transport. Together, these criteria capture the most visually salient effects associated with SSS and provide a basis for qualitative comparison of the presented techniques.

Non-local light diffusion beneath the surface is the defining characteristic of SSS, describing the lateral redistribution of incident light over distance. STA does not model this behaviour explicitly, as the approach evaluates lighting locally at each surface point without accounting for neighbouring irradiance. However, it approximates this behaviour by artificially adding light when the surface faces away from the light source. In contrast, SSSS and the CBNDM introduce non-local diffusion by redistributing surface irradiance across neighbouring pixels: the former applies successive one-dimensional convolutions in screen space, while the latter samples neighbouring pixels arranged in a spiral pattern around each surface point and accumulates their contributions according to a radially symmetric diffusion profile.

Distance-dependent colour bleeding refers to the gradual change in surface colour caused by wavelength-dependent attenuation as it travels subsurface over increasing distance. Although not explicitly discussed in the papers for each model, this effect can theoretically be reproduced by performing the convolution with each colour channel if a kernel for each colour R, G and B in the diffusion profile is provided. Similarly, even though d is defined as a scalar value in the CBNDM, the model is evaluated independently per colour channel in practice, with channel-specific diffusion distances derived from material parameters (see RenderMan and Blender

documentation [3, 11]). STA does not model this behaviour in any way.

For a model to be energy conserving, it has to be able to consistently allocate the energy from the incoming light between surface reflection and subsurface transport. This means that an increase in surface reflection must correspond to a decrease in SSS. STA does not conserve energy, as it simply adds transmitted light on top of existing shading. In SSSS, the pre-integrated kernel conserves the total energy of the diffusion reflectance profile, as shown in Equation (2). However, this conservation applies only within the convolution step itself. As a result, neither the pre-integrated nor the artist-friendly approaches are energy conserving. The CBNDM is energy conserving as shown in Equation (4), where the model redistributes a fixed amount of energy rather than introducing additional illumination.

Computational Cost. Let m be the number of shaded pixels and v the number of vertices of the mesh.

STA incurs a constant cost per pixel, as it is evaluated locally without any neighbourhood sampling. This yields a cost of $O(m)$ if the computation is done in the fragment shader and $O(v)$ if it is done in the vertex shader.

Since SSSS and the CBNDM are screen-space approaches, we let m_{screen} be the total number of pixels on screen.

SSSS performs two 1D convolutions in both the horizontal and vertical axes, incurring a cost of $O(n) + O(n) = O(2n) = O(n)$ per pixel for n samples. In total, this approach has a complexity of $O(m_{\text{screen}}n)$.

Similarly, CBNDM evaluates a non-separable diffusion profile in a single pass by sampling a neighbourhood with n samples, also giving $O(m_{\text{screen}}n)$.

Optimising both approaches with a stencil mask to restrict evaluation to relevant pixels reduces the cost to $O(mn)$.

5 Conclusion

This paper compared several real-time SSS techniques based on their visual fidelity and computational characteristics, focussing on how different approximations reproduce key perceptual effects of subsurface light transport. While simple translucency provides a low-cost approximation suitable for thin or back-lit materials or visually insignificant assets, more advanced diffusion-based approaches offer increasingly accurate representations of non-local scattering and colour bleeding at higher computational cost. In particular, SSSS and the CBNDM illustrate the trade-off between performance and physical plausibility that underpins most real-time SSS implementations. Overall, the analysis highlights that convincing SSS can be achieved in real-time rendering through carefully chosen approximations, with the appropriate model depending on the desired balance between visual quality and performance constraints.

References

[1] Colin Barré-Brisebois. 2011. Approximating Translucency for a Fast, Cheap, and Convincing Subsurface Scattering Look (GDC 2011). https://colinbarrebrisebois.com/2011/03/07/gdc-2011-approximating-translucency-for-a-fast-cheap-and-convincing-subsurface-scattering-look. Accessed: 2025-10-30.

[2] Blender Foundation. 2024. EEVEE Subsurface Scattering Implementation. https://projects.blender.org/blender/blender/src/branch/main/source/blender/draw/engines/eevee/eevee_subsurface.cc. Source code accessed January 2026.

[3] Blender Foundation. 2024. *Subsurface Scattering*. Blender Foundation. https://docs.blender.org/manual/en/latest/render/shader_nodes/shader/sss.html

[4] Per H. Christensen and Brent Burley. 2015. *Approximate Reflectance Profiles for Efficient Subsurface Scattering*. Technical Report Pixar Technical Memo 15-04. Pixar Animation Studios. https://graphics.pixar.com/library/ApproxBSSRDF/paper.pdf

[5] Eugene d'Eon, David Luebke, and Eric Enderton. 2007. Efficient Rendering of Human Skin. In *Proceedings of the Eurographics Symposium on Rendering (EGSR)*. 147–157. https://eugenedeon.com/pdfs/efficientskin.pdf

[6] Evgenii Golubev. 2018. Efficient Screen Space Subsurface Scattering. https://advances.realtimerendering.com/s2018/Efficient%20screen%20space%20subsurface%20scattering%20Siggraph%202018.pdf. SIGGRAPH 2018 Course: Advances in Real-Time Rendering.

[7] Naty Hoffman. 2013. Physics and Math of Shading. In *ACM SIGGRAPH 2013 Courses: Physically Based Shading in Theory and Practice*. ACM, Anaheim, CA, USA. Course slides available at https://blog.selfshadow.com/publications/s2013-shading-course/hoffman/s2013_pbs_physics_math_slides.pdf.

[8] Henrik Wann Jensen and Juan Buhler. 2001. A Rapid Hierarchical Rendering Technique for Translucent Materials. In *Proceedings of SIGGRAPH*. ACM, 511–518. https://graphics.stanford.edu/papers/fast_bssrdf/fast_bssrdf.pdf

[9] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. 2001. A Practical Model for Subsurface Light Transport. In *Proceedings of SIGGRAPH*. ACM, 511–518. http://www.graphics.stanford.edu/papers/bssrdf/bssrdf.pdf

[10] Jorge Jimenez, Karoly Zsolnai, Adrian Jarabo, Christian Freude, Thomas Auzinger, Xian-Chun Wu, Javier von der Pahlen, Michael Wimmer, and Diego Gutierrez. 2015. Separable Subsurface Scattering. https://www.cg.tuwien.ac.at/research/publications/2015/Jimenez_SSS_2015/Jimenez_SSS_2015-paper.pdf

[11] Pixar Animation Studios. 2023. *Subsurface Scattering Parameters*. Pixar. https://rmanwiki-26.pixar.com/space/REN26/19661417/Subsurface+Scattering+Parameters