

Screen-Space Perceptual Rendering of Human Skin

JORGE JIMENEZ

Universidad de Zaragoza

VERONICA SUNDSTEDT

Trinity College Dublin

and

DIEGO GUTIERREZ

Universidad de Zaragoza

We propose a novel skin shader which translates the simulation of subsurface scattering from texture space to a screen-space diffusion approximation. It naturally scales well while maintaining a perceptually plausible result. This technique allows us to ensure real-time performance even when several characters may appear on screen at the same time. The visual realism of the resulting images is validated using a subjective psychophysical preference experiment. Our results show that, independent of distance and light position, the images rendered using our novel shader have as high visual realism as a previously developed physically-based shader.

Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—*Color, shading, shadowing, and texture*

General Terms: Algorithms, Performance, Experimentation, Human Factors

Additional Key Words and Phrases: Real-time skin rendering, psychophysics, perception

ACM Reference Format:

Jimenez, J., Sundstedt, V., and Gutierrez, D. 2009. Screen-Space perceptual rendering of human skin. ACM Trans. Appl. Percept. 6, 4, Article 23 (September 2009), 15 pages.

DOI = 10.1145/1609967.1609970 <http://doi.acm.org/10.1145/1609967.1609970>

1. INTRODUCTION

Lots of materials present a certain degree of translucency, by which light falling onto an object enters its body and scatters within it (a process known as subsurface scattering), before leaving the object at

This research has been partially funded by the Spanish Ministry of Science and Technology (TIN2007-63025) and the Gobierno de Aragón (OTRI 2009/0411). J. Jimenez was funded by a research grant from the Instituto de Investigación en Ingeniería de Aragón.

Authors' address: J. Jimenez, Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Edificio Ada Byron, María de Luna 1, 50018, Zaragoza, Spain; email: jim@unizar.es; V. Sundstedt, Lloyd Institute, Room 0.45, School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland; email: Veronica.Sundstedt@cs.tcd.ie; D. Gutierrez, Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Edificio Ada Byron, María de Luna 1, 50018, Zaragoza, Spain; email: diegog@unizar.es.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2009 ACM 1544-3558/2009/09-ART23 \$10.00

DOI 10.1145/1609967.1609970 <http://doi.acm.org/10.1145/1609967.1609970>

a certain distance from the incidence point. Example objects include paper, tree leaves, soap, a candle, fruit, etc. Human skin is a particularly interesting translucent material. It is made up of multiple translucent layers, which scatter light according to their specific composition [Igarashi et al. 2005]. This creates a very characteristic appearance, to which our visual system seems to be specially well-tuned: Slight errors in its simulation will be picked up easier than, say, errors in the simulation of wax.

Correct depiction of human skin is important in fields such as cinematography and computer graphics. However, while the former can count on the luxury of offline rendering, the latter imposes real-time constraints that make the problem much harder. The main challenge is to compute an approximation of the complex subsurface scattering effects, good enough to be perceptually plausible, but at the same time fast enough to allow for real-time rendering and easy to implement so that it integrates well with existing pipelines. The key is to reduce computational costs while leveraging the limitations of the human visual system.

Several real-time algorithms to simulate skin already exist [Gosselin 2004; d'Eon et al. 2007; Hable et al. 2009; Jimenez and Gutierrez 2008]. Their common key insight is the realization that subsurface scattering mainly amounts to blurring of high-frequency details, which these algorithms perform in texture space. Whilst the results can be quite realistic, they suffer from the fact that they do not scale well; more objects means more textures that need to be processed, and thus performance quickly decays. This is especially problematic in the case of computer games, where a lot of characters may appear on screen simultaneously, but real time is still needed. We believe that this is one of the main issues that is keeping game programmers from rendering truly realistic human skin. The commonly adopted solution is to simply ignore subsurface scattering effects, thus losing realism in the appearance of the skin. Additionally, real-time rendering in a computer game context can become much harder, with issues such as the geometry of the background, depth of field simulation, or motion blur imposing additional time penalties.

To help solve these problems, we propose an algorithm to render perceptually realistic human skin, which translates the simulation of scattering effects from texture to screen space (see Figure 1). We thus reduce the problem of simulating translucency to a postprocess, which has the added advantage of being easy to integrate in any graphics engine. The main risk of this approach is that in screen space we have less information to work with, as opposed to algorithms that work in 3D or texture space. We are interested in finding out how the cognitive process that recognizes human skin as such is affected by this increased inaccuracy. Working within perceptual limits, can we still obtain a model of human skin that is perceived as at least as photorealistic as texture-space approaches?

We note that, while the perception of translucency has been studied before [Fleming and Bülthoff 2005], there has not been any previous research on the perception of the particular characteristics of human skin. We perform a psychophysical evaluation comparing four different approaches: (a) no simulation of subsurface scattering, (b) a state-of-the-art, texture-space algorithm from nVIDIA, (c) a naive mapping of an existing algorithm into screen space, and (d) our novel screen-space algorithm. Results show that, independent of distance and light position, our method performs perceptually on par with the method from nVIDIA (generally accepted as state-of-the-art in real-time skin rendering, and taken here as our ground truth), while being generally faster and scaling much better with multiple characters (see Figure 2).

2. THE DIFFUSION APPROXIMATION

Subsurface scattering is usually described in terms of the Bidirectional Scattering Surface Reflectance Distribution Function (BSSRDF) [Nicodemus et al. 1977]. Jensen et al. [2001] approximate the outgoing radiance L_0 in translucent materials using a dipole diffusion approximation. Thus, the costly



Fig. 1. Top: existing algorithms blur high-frequency details in texture space. The image sequence shows the initial rendered irradiance map and two blurred versions. Bottom: equivalent process in screen space, where the blurring is applied to selected areas directly on the rendered image (see text for details).



Fig. 2. Several heads rendered with our screen-space shader. Our psychophysical experiments show that it ranks perceptually on par with the state-of-the-art NVIDIA shader [d'Eon et al. 2007], while scaling much better as the number of heads increases.

Bidirectional Scattering Surface Reflectance Distribution Function (BSSRDF) becomes

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

where x_i and $\vec{\omega}_i$ are the position and angle of the incident light, x_o and $\vec{\omega}_o$ are the position and angle of the radiated light, F_t is the Fresnel transmittance, and R is the diffusion profile of the material. The model is further simplified in subsequent work by Jensen and Buhler [2002]. Donner and Jensen [2005] extended the dipole model to a multipole approximation, defined as a sum of dipoles. This allowed for better simulation of multilayered materials, such as skin. For each pair of layers, they analyze the convolution of their reflectance and transmittance profiles in frequency space (thus performing faster multiplications instead). They show how the multipole approach provides a better fit with ground-truth Monte Carlo simulations than the simpler dipole approximation. The same authors simplified the highly dimensional parameter space in a subsequent publication [Donner and Jensen 2006]. In Donner et al. [2008], the layered light transport was recently extended to account for heterogeneous scattering of light. However, these techniques usually rely on precomputations and still do not reach real time; thus, they are not suitable for games.

3. PREVIOUS WORK

Relevant related areas include texture-space diffusion approximation for real-time rendering of skin and studies on the perception of translucency.

Texture-space diffusion. Borshukov and Lewis [2003] showed the feasibility of simulating diffusion in texture space. They observed that, once the irradiance map has been rendered to a texture, simply blurring the texture with different kernels yields an approximation of subsurface scattering (as Figure 1, top, shown). Gosselin [2004] and Green [2004] build on this idea and achieve real time, by means of Poisson sampling and a single Gaussian blurring, respectively. The work by d'Eon et al. [2007] and d'Eon and Luebke [2007] made the key observation that the diffusion profiles that the multipole model predicts [Donner and Jensen 2005] could be approximated by a weighted sum of Gaussians. This allows for a representation based on a hierarchy of irradiance diffusion maps which, when combined, closely match the nonseparable global diffusion profile. Texture distortion issues from unfolding the geometrical mesh can affect diffusion between two points: To avoid this, a *stretch map* modulates the Gaussian convolution at each frame. Lately, Jimenez and Gutierrez [2008] and Hable et al. [2009] have built on this work, introducing optimizations into the rendering pipeline. These will be further discussed in Section 4.

Perception of translucency. Translucency is a very important characteristic of certain materials, and easy to pick up by the visual system. Nevertheless, due to the complexity of the light transport involved, it is unlikely that the visual system relies on any inverse optics to detect it [Fleming and Bülthoff 2005]. Koenderink and van Doorn [2001] developed some “rules of thumb” to help explain the perception of translucent materials, and hypothesize that more general laws may be utopic. Fleming and Bülthoff [2005] present a series of psychophysical studies and analyze low-level image cues that affect perceived translucency. A more comprehensive overview can be found in Singh and Anderson [2002]; however, no previous work has focused on the specific characteristics of the perception of human skin.

4. SCREEN-SPACE ALGORITHM

Our rendering algorithm is based on the idea of performing the diffusion approximation in screen space (as opposed to texture space), to streamline the production pipeline and ensure real-time performance even as the number of on-screen characters increases. We are inspired by two works that aimed to optimize texture-space approaches, which we first describe briefly.

Optimizing the pipeline. Current state-of-the-art methods for real-time rendering of human skin are based on the texture-space diffusion approximation discussed earlier. However, rendering irradiance

maps in this way for a given model means that the GPU pipeline cannot be used in the conventional way. This is due to the fact that the vertex shader is not used as usual to transform object coordinates to clip coordinates, but to assign to each vertex a pair of (u, v) coordinates on the unfolded geometrical mesh instead. As a consequence, two optimizations that would otherwise be implicitly performed by the GPU right after the execution of the vertex shader are now lost, namely backface culling and view frustum clipping. Jimenez and Gutierrez [2008] reintroduce those two optimizations in the rendering pipeline proposed in d'Eon et al. [2007]. They perform optimal, per-object modulation of the irradiance map size based on a simple, depth-based method.

Similar in spirit, Hable et al. [2009] also reintroduce backface culling, and propose an additional optimization: Instead of computing the twelve one-dimensional Gaussians to approximate the diffusion profile as in d'Eon et al. [2007], they compute a single bidimensional convolution at thirteen jittered sample points, which account for direct reflection (one point), mid-level scattering (six points), and wide red scattering (six points).

Texture- versus screen-space. Texture-space diffusion has some intrinsic problems that can be easily solved when working in screen space. We outline the most important ones.

- It requires special measures to bring back typical GPU optimizations (backface culling and viewport clipping), and to compute an irradiance map proportional to the size of the subject on the screen. In screen space, these optimizations are implicit.
- Each subject to be rendered requires her own irradiance map (thus forcing as many render passes as subjects). In image space, *all subjects* are processed at the same time.
- The irradiance map forces the transformation of the model vertices twice: during the irradiance map calculation, and to transform the final geometry at the end. In screen space, only this second transformation is required.
- Modern GPUs can perform an early-Z rejection operation to avoid overdraw and useless execution of pixel shaders on certain pixels. In texture space, it is unclear how to leverage this and optimize the convolution processes according to the final visibility in the image. In screen space, a depth pass can simply discard them before sending them to the pixel shader.
- Depending on the surface orientation, several pixels in texture space may end up mapped on the same pixel in screen space, thus wasting convolution calculations.¹ This situation is avoided altogether by working directly in screen space; as our tests will show, the errors introduced by this simplification are not picked up by the visual system.
- Adjacent points in 3D world space may not be adjacent in texture space. Obviously, this will introduce errors in texture-space diffusion that are naturally avoided in screen space.

Apart from fixing these problems, our psychophysical evaluation (Section 5) shows that errors introduced by working in screen space are mostly unnoticed by a human observer.

As we will see in Section 7, our screen-space algorithm has some limitations.

- Adjacent points in 2D screen space may not be adjacent in 3D world space. This produces artifacts in the form of small haloes around some areas of the model such as the ears or nose.
- It fails to simulate the light transmitted through high-curvature features, since we lack lighting information from the back of the objects.
- It requires the usage of additional textures in order to store the specular channel and object's matte.

¹Modulating the size of the irradiance map may reduce the problem somewhat, but will not work if the viewpoint (and thus the surface orientation) changes in a dynamic environment.

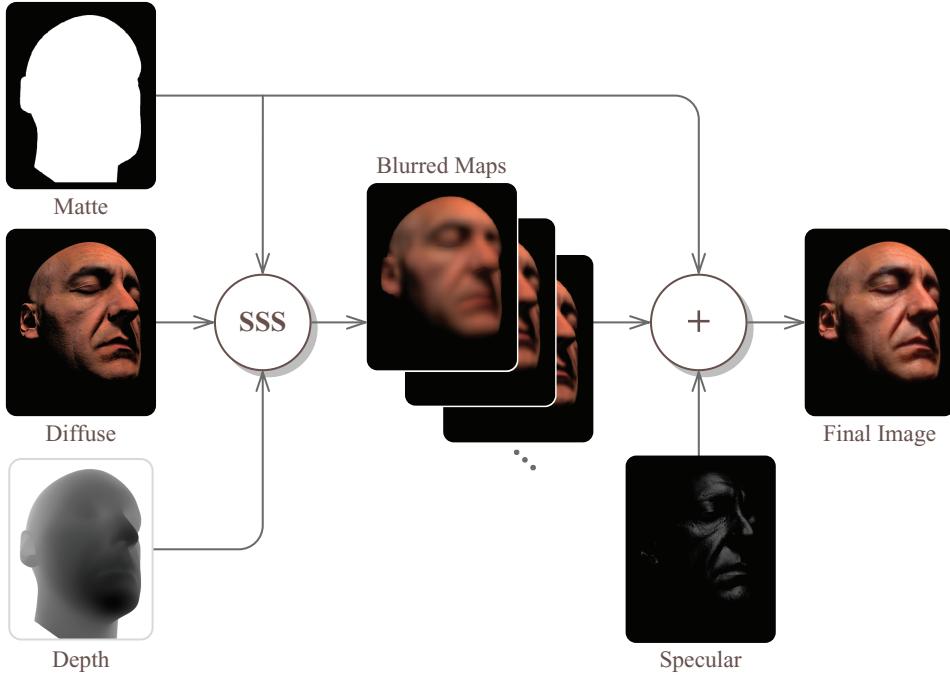


Fig. 3. Overview of our screen-space algorithm.

Screen-space diffusion algorithm. Our algorithm follows the idea of optimizing the pipeline [Jimenez and Gutierrez 2008; Hable et al. 2009], and further simplifies the texture-space diffusion approximation by working in screen space. Recall that all previous real-time algorithms are based on the idea of convolving a diffusion profile over the irradiance map for each model. In contrast, our algorithm takes as input a rendered image with no subsurface scattering simulation, plus the corresponding depth and matte of the object. Since the subsurface scattering effect should only be applied to the diffuse component of the illumination, not affecting specular highlights, we take advantage of the multiple render targets capability of modern GPUs, and store the diffuse and specular components separately. Depth is linearized following [Gillham 2006]. We then apply our diffusion profiles directly on the rendered diffuse image, as opposed to the irradiance map stored as a texture. Figure 3 shows an overview of our algorithm.

We have applied the convolution kernel in two different ways: using a single bidimensional convolution with jittered samples (as proposed in Hable et al. [2009]) and using the six one-dimensional Gaussians described in d'Eon et al. [2007], combining them with a weighted sum in a second pass. For both cases, we have run a psychophysical evaluation to validate the results. For the specular component we have used the Kelemen/Szirmay-Kalos model [Kelemen and Szirmay-Kalos 2001]; additionally, we apply a bloom filter similar to the one used by our reference texture-space algorithm [d'Eon and Luebke 2007], which follows Pharr and Humphreys's recommendation [2004].

To apply the convolution only in the necessary parts of the image, and thus apply the pixel shader in a selective way, we rely on the matte mask. Since we are working in screen space, the kernel width² must be modulated taking into account the following.

²Jittered sample positions in case of the convolution with jittered samples and standard deviation in case of the convolutions with Gaussians.

- A pixel representing a further away object should use a narrower kernel.
- Greater gradients in the depth map should also use narrower kernels. This is similar in spirit to using stretch maps, without the need to actually calculate them.

We thus multiply the kernel width by the following stretch factors. We have

$$s_x = \frac{\alpha}{d(x, y) + \beta \cdot \text{abs}(\nabla_x d(x, y))}, \quad (2)$$

$$s_y = \frac{\alpha}{d(x, y) + \beta \cdot \text{abs}(\nabla_y d(x, y))}, \quad (3)$$

where $d(x, y)$ is the depth of the pixel in the depth map, α indicates the global subsurface scattering level in the image, and β modulates how this subsurface scattering varies with depth gradient. The operators ∇_x and ∇_y compute the depth gradient, and are implemented on the GPU using the functions ddx and ddy , respectively. Note that increasing depth gradients reduce, the size of the convolution kernel as expected; in practice, this limits the effect of background pixels being convolved with skin pixels, given that in the edge, the gradient is very large and thus the kernel is very narrow. The value of α is influenced by the size of the object in 3D space, the field-of-view used to render the scene, and the viewport size (as these parameters determine the projected size of the object). All the images used in this work have empirically fixed values of $\alpha = 11$ and $\beta = 800$. Figure 4 shows the influence of α and β in the final images. As noted in d’Eon et al. [2007], where a similar stretch factor is used in texture space to modulate the convolutions, filtering is not separable in regions where stretch varies. However, we have not noticed any visual artifacts as result of this stretched one-dimensional Gaussian convolutions.

To illustrate its efficiency, Figure 5 (top), shows a scheme of the necessary steps involved in a generalized texture-space skin shader. The stretch and irradiance maps, the Gaussian convolutions, and the render must be performed once per character. In contrast, our algorithm only requires the rendering step to be performed on a per-character basis; no stretch nor irradiance maps are needed, and convolutions are performed only once directly in screen space (Figure 5, bottom). The final bloom pass, not optimized with our algorithm, is also performed on the final image and thus its computation time does not increase as the number of objects grows. It simulates the point-spread-function of the optical system used to capture (render) the image.

5. PERCEPTUAL VALIDATION

The realism of the rendered images was validated using a subjective psychophysical experiment. The experiment was based on comparing four different shaders, namely: no SSS, TS, SS Jittered, and SS Full (see Table I for a brief summarizing description). The conditions used to render each shader are further described in Section 5.2. The images rendered using the TS shader were always used to compare the other shaders against.

Human subjects were used to evaluate the realism of the shaders using a subjective two-alternative forced-choice (2AFC) preference experiment. The research hypothesis in the experiment was that the images rendered using the no SSS, SS Jittered, and SS Full shaders would produce as high visual realism as the ground-truth TS shader.

5.1 Participants

Sixteen volunteering participants (14 male and 2 female; age range: 23–43) were recruited for the experiment. All of the participants were naïve as to the purpose of the experiment. The subjects had a variety of experience with computer graphics, and all self-reported normal or corrected-to-normal



Fig. 4. The influence of the α and β parameters. Top: fixed $\beta = 800$ and varying α of 0, 5.5, and 11, respectively. Note how the global level of subsurface scattering increases. Bottom: fixed $\alpha = 11$ and varying β of 0, 400, and 800, respectively. Note how the shadow under the nose gets readjusted according to the depth gradient of the underlying geometry.

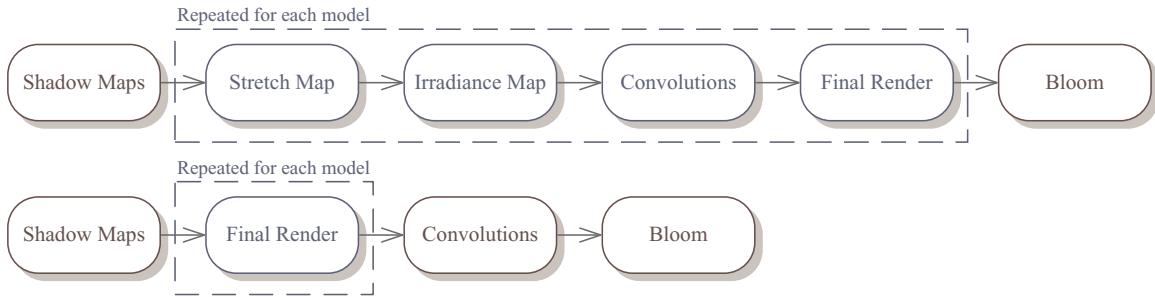


Fig. 5. Top: scheme of the pipeline described in d'Eon et al. [2007]. Bottom: our simplified screen-space strategy. Notice how less steps are performed on a per-character basis.

vision. Seven subjects reported that they did not have any expertise in art (drawing, photography, computer graphics, or digital art, video, etc.), whereas nine participants reported that they did. It was hypothesized that the participants with knowledge in computer graphics would be better at judging which of the images were the most realistic ones in theory.

Table I. Names and Description of the Four Shaders Used in Our Psychophysical Experiments

Name	Description
No SSS	Rendered without subsurface scattering
TS	Texture-space simulation [d'Eon et al. 2007]
SS Jittered	Screen-space adaptation of [Hable et al. 2009], with jittered samples
SS Full	Our screen-space algorithm



Fig. 6. Example stimuli used in our psychophysical tests: zoom, near, and far images with light at 0° , for the case of TS vs. SS Full. In this figure, the TS version appears always on the left, and our shader on the right for comparison purposes.

5.2 Stimuli

The head model used in the experiment was the head obtained from XYZRGB.³ This model was used to create stimuli images for the experiment using the four shaders. Each participant viewed 72 trials in total. The number of images rendered for each shader was twelve (1 shader \times 3 camera distances \times 4 light angles). The three camera configurations used were at 28, 50, and 132 distance units, which in the article are referred to zoom, near, and far (for reference, the head is 27 distance units tall).

Four different lighting configurations were used for each camera distance. These had a fixed elevation angle θ of 45° and a varying azimuth angle ϕ with values 0° (just in front of the head), 60° , 110° , and 150° (almost behind of the head). Examples of the different configurations used in the experiment can be seen in Figure 6.

The 72 trials were displayed in random order for each participant. Counterbalancing was used to avoid any order bias. This meant that each participant saw each comparison pair twice. In half the trials the TS shader was displayed first and in half the trials it was displayed second.

All stimuli images were displayed on a monitor with a 1650×988 resolution. The lighting was dimmed throughout the experiment. The participants were seated on an adjustable chair, with their eye level approximately level with the center of the screen, at a viewing distance of approximately 60 cm.

5.3 Procedure

After filling in a consent form and questionnaire the participants were given a sheet of instructions on the procedure of the particular task they were to perform. The task description given to the participants is shown in the Appendix. The participants were asked to perform a 2AFC task that assessed the realism of the skin rendering. A *no reference* condition [Sundstedt et al. 2007] was used in which the participants were asked to discriminate which of two consecutively displayed images (TS and either No SSS, SS Jittered, or SS Full) looked most like real human skin.

³<http://www.xyzrgb.com/>



Fig. 7. Zoom-in configuration with lighting in an angle of 110°. Although the Chi-square technique showed a significant difference for this configuration, the images are perceptually very similar.

Since the aim was to find out which shader looked most like real skin, a reference condition was not used, which would have displayed the TS image at the same time as a TS and other shader image. In a real application, such as a computer game, the shader would not be directly compared with a physically-based shader.

Upon viewing each trial the participants wrote down their responses (1 or 2) in a table depending on which image they thought looked most like real skin. They controlled the loading of the next trial themselves by using the space bar. A 50% grey image was shown between the images in the pair with a duration of two seconds. This image was used to let the participants know that a new stimuli was being displayed.

The participants could watch each experiment image for as long as they liked, but were encouraged to spend around 10 seconds looking at each image. A previous pilot study had also suggested that the rougher appearance of the No SSS shader output could make the person look older. Due to this the participants were also instructed that the person in each image had the same age.

6. RESULTS

Figure 8 shows the overall results of the experiment for the different light angles and camera distances, respectively. The results are normalized in the y-axis. In each group of conditions, a result of 1 indicates that the TS shader was always chosen over the second option, while a result of 0.5 is the unbiased ideal. This is the statistically expected result in the absence of a preference or bias towards one shader, and indicates that no differences between the TS and the other shader images were perceived.

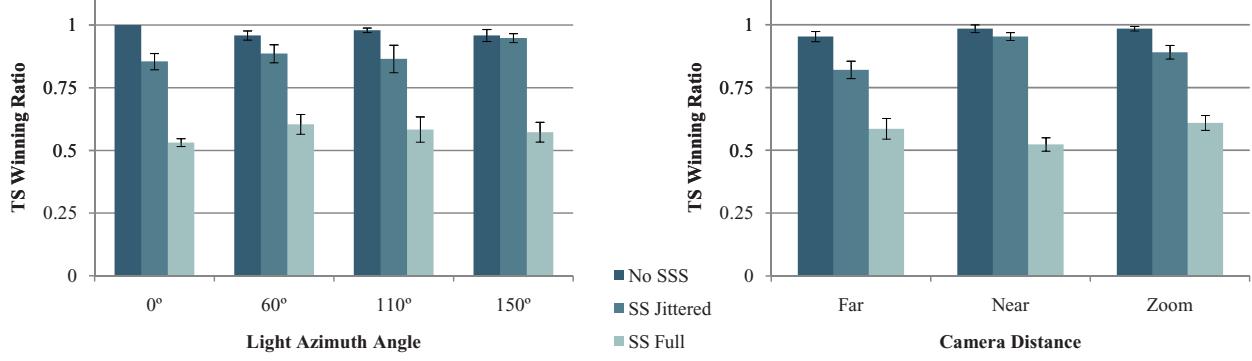


Fig. 8. Results of our psychophysical experiments for varying light position (left) and distance to the camera (right). The shaders No SSS, SS Jittered, and SS Full are compared against TS. Our SS Full shader outperformed the other two. Although TS was chosen over SS Full slightly above chance, they consistently showed no statistical difference ($p > 0.05$), while SS Full is faster and scales better.

Table II. Output for the Chi-Square Analysis (df = 1, 0.05 Level of Significance)

Configuration	No SSS		SS Jittered		SS Full	
	chi-value	p-value	chi-value	p-value	chi-value	p-value
TS - Far - 0°	21.125	0.000	21.125	0.000	3.125	0.077
TS - Far - 60°	28.125	0.000	18	0.000	3.125	0.077
TS - Far - 110°	24.5	0.000	15.125	0.000	0.125	0.724
TS - Far - 150°	32	0.000	15.125	0.000	0	1
TS - Near - 0°	32	0.000	28.125	0.000	0	1
TS - Near - 60°	32	0.000	28.125	0.000	0.125	0.724
TS - Near - 110°	24.5	0.000	28.125	0.000	1.125	0.289
TS - Near - 150°	32	0.000	21.125	0.000	0.125	0.724
TS - Zoom - 0°	28.125	0.000	28.125	0.000	0.5	0.480
TS - Zoom - 60°	28.125	0.000	18	0.000	2	0.157
TS - Zoom - 110°	32	0.000	15.125	0.000	4.5	0.034
TS - Zoom - 150°	32	0.000	18	0.000	0.5	0.480

Values in bold indicate no significant difference.

The results were analyzed statistically to determine any significance. To find out whether the number of participants who correctly classified the TS shader images is what would be expected by chance, or if there was really a pattern of preference, we used the Chi-square nonparametric technique. A one-sample Chi-square includes only one dimension, such as the case as in our experiments.

The obtained (TS/No SSS, TS/SS Jittered, and TS/SS Full) frequencies were compared to an expected 16/16 (32 for each comparison) result to ascertain whether this difference would be significant. The Chi-square values were computed and then tested for significance, as shown in Table II.

The obtained values show that when participants compared the No SSS shader with the TS shader they always managed to identify the TS shader correctly ($p < 0.05$). This was also true for the SS Jittered shader under all camera distances and lighting angles. However, for the SS Full shader comparison with the TS shader there was no significant difference ($p > 0.05$) in all conditions apart from when the configuration was zoomed in on the skin and the lighting was in an angle of 110°. This indicates that the SS Full shader can produce images that are as realistic looking as the physically-based TS shader. Although the result showed a significant difference for the angle of 110°, it is believed this may be due to chance, since the images are perceptually very similar (see Figure 7). The use of a few

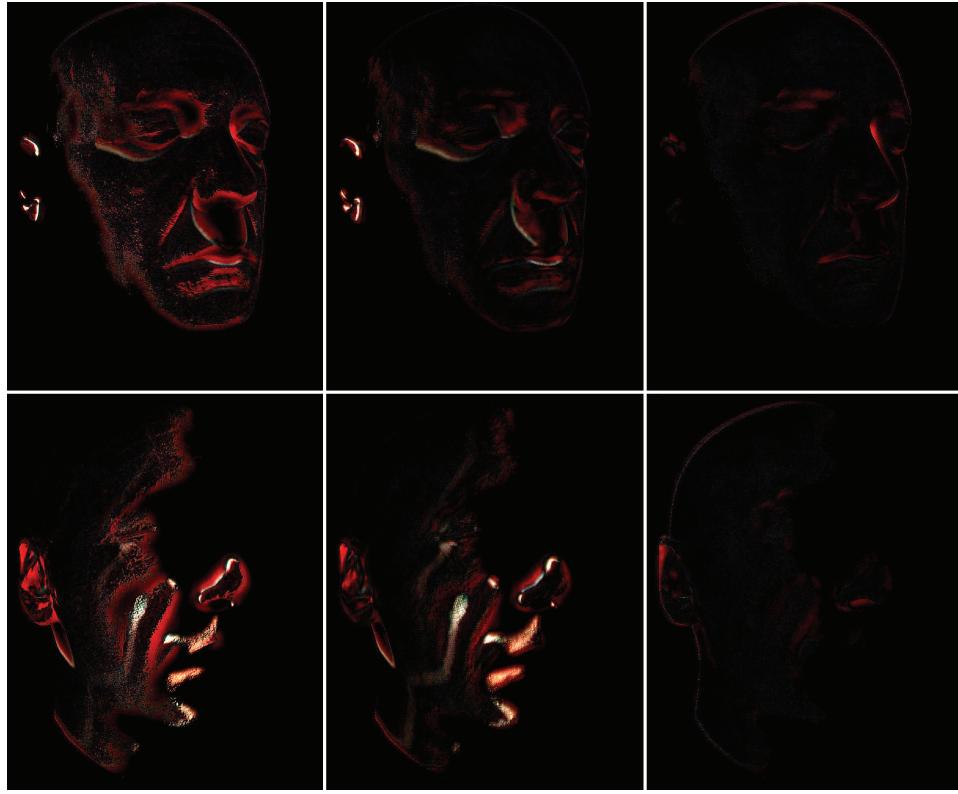


Fig. 9. Differences between the No SSS, SS Jittered, and SS Full shaders (left, middle, and right, respectively) and the TS shader. Top: angle at 0°. Bottom: angle at 110° (contrast enhanced by a factor of 75 for visualization purposes). Our shader most closely matches the TS shader.

additional participants or stimuli repetitions could potentially have clarified this result; nevertheless, the fact that the light angle in combination with camera distance possibly can alter the threshold is also an interesting outcome which warrants future work. Overall the correct selections for the SS Full shader are very much different from the comparisons with the other shaders and the physically-based TS shader.

7. DISCUSSION AND CONCLUSIONS

We have presented a novel real-time shader which can be used for efficient realistic rendering of human skin. We have demonstrated how screen-space subsurface scattering produces results on par with current state-of-the-art algorithms, both objectively (see Figure 9) and subjectively: A psychophysical experiment was conducted which showed that the images rendered using this new shader can produce stimuli that have as high visual realism as a previously developed physically-based shader.

Our screen-space algorithm has two main limitations. Firstly, as Figure 10 shows, it fails to reproduce light transmitted through high-curvature, thin features such as the ears or nose, due to the fact that in screen space we have no information about incident illumination from behind. However, this did not seem to affect its overall perceptual performance. Secondly, under certain configurations of lights and camera, small haloes may appear when using our screen-space algorithm, for instance, scattering from the nose incorrectly bleeding onto the cheek. These are usually small and masked by the visual system

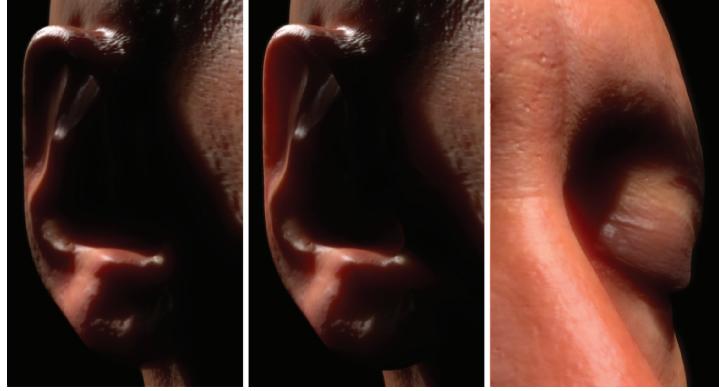


Fig. 10. Limitations of our screen-space shader: It fails to capture transmission of light from behind the object (left, using our shader, and middle, using d'Eon et al. [2007]). It can also create small haloes in certain areas under specific combinations of lights and camera position, such as scattering from the nose falling onto the eyelid (right). However, our tests show that these do not usually hamper perception of skin.

Table III. Performance (frames per second) of the Skin Shaders Used in Our Psychophysical Experiment

Nr. of Heads	No SSS	SS Jittered	SS Full	TS
1	51	42	29	26
3	41	32	24	13
5	40	33	25	9

(see Figure 10).

Our psychophysical evaluation suggests that the two most important perceptual characteristics of translucency in human skin that a shader should reproduce are the general softening of the high-frequency features and the overall reddish glow. These seem to be more important than transmission of light in certain areas, as our psychophysical tests suggest, even in the extreme case shown in Figure 9.

The absence of the reddish glow truly hampers perception of human skin: We noted that the SS Jittered shader correctly softens the overall appearance but sometimes fails to capture most of this reddish glow. Interestingly, it consistently obtained a lower overall ranking. In contrast, the absence of high-frequency feature softening tends to be taken as a sign of age: This insight could potentially guide some age-dependant future skin shaders.

The results also show that there was no statistical difference in the number of error selections by participants with experience in computer graphics and participants reporting no experience ($p > 0.05$). This indicates that although participants with computer graphics experience have been shown to have easier detect errors in rendering [Mastoropoulou et al. 2005], even people with no such experience are as good at detecting the realism of rendered skin. This is probably due to the fact that humans have evolved to detect anomalies in the real world and are used to see other humans in daily life.

Table III shows performance (in frames per second) of the four shaders used in the psychophysical experiment. The data have been obtained with a single light source, a shadow map resolution of 512x512, 4x MSAA, 2988 triangles per head, rendered at 1280x720 (which is becoming a standard resolution for most console games). The size of the irradiance map for the TS shader was 1024x1024. All the images were rendered on a machine with a Core 2 Duo at 2GHz and a GeForce 8600M GS.

Increasing the number of heads from one to five, our SS Full algorithm only drops by 13%, compared to 65%, 21%, and 22% for the TS, SS Jittered, and No SSS, respectively. Although the SS Jittered and No

SSS shaders yield more frames per second, they do so at the cost of lowering perceived visual quality, as our experiments have shown. Thus, our screen-space approach provides a good balance between speed, scalability, and perceptual realism.

For future research it would be necessary to study how the shader would work on different skin types, varying parameters such as race or age. It would also be interesting to study how the different shaders would perform in more complex, dynamic environments and contexts such as computer games; further shader simplifications may be possible in those cases. A more targeted evaluation of the perceptual importance of individual skin features (including general softening, reddish glow, or transmittance) would be interesting, to help establish a solid foundation about what makes skin really look like skin. Further tests could also possibly compare the results of a shader with gold-standards in the form of real pictures, as opposed to comparing against other existing shaders.

In summary, our psychophysical experiments show that the obvious loss of physical accuracy from our screen-space approach goes mainly undetected by humans. Based on our findings, we have hypothesized what the most important perceptual aspects of human skin are. We hope that these findings motivate further research on real-time, perceptually accurate rendering.

APPENDIX

For the participants the task description was the following:

This test is about selecting one image in a set of two images (72 pairs in total). You will be shown the two images consecutively with a grey image being displayed for 2 seconds between them. A trial nr (1–72) will separate each trial.

Your task is to choose the image which you think look most realistic (i.e., most like real human skin). You can view the images for an unlimited time, but we recommend that you spend around 10 seconds before making your selection. You should also keep in mind that the person in each image has the same age (around 45 years old).

If anything is unclear please ask any questions you might have before the study starts.

ACKNOWLEDGMENTS

We would like to thank the members of the GV2 group at Trinity College, Dublin and all of the participants in the experiments. We would also like to thank C. Vogel, the anonymous reviewers, and XYZRGB Inc. for the high-quality head scan.

REFERENCES

- BORSHUKOV, G. AND LEWIS, J. P. 2003. Realistic human face rendering for “The Matrix Reloaded”. In *ACM SIGGRAPH Sketches and Applications*.
- D’EON, E. AND LUEBKE, D. 2007. Advanced techniques for realistic real-time skin rendering. In *GPU Gems 3*. H. Nguyen, Ed. Addison Wesley, Chapter 14.
- D’EON, E., LUEBKE, D., AND ENDERTON, E. 2007. Efficient rendering of human skin. In *Proceedings of Eurographics Symposium on Rendering*.
- DONNER, C. AND JENSEN, H. W. 2005. Light diffusion in multi-layered translucent materials. *ACM Trans. Graph.* 24, 3, 1032–1039.
- DONNER, C. AND JENSEN, H. W. 2006. A spectral BSSRDF for shading human skin. In *Proceedings of Eurographics Symposium on Rendering*.
- DONNER, C., WEYRICH, T., D’EON, E., RAMAMOORTHI, R., AND RUSINKIEWICZ, S. 2008. A layered, heterogeneous reflectance model for acquiring and rendering human skin. *ACM Trans. Graph.* 27, 5.
- FLEMING, R. W. AND BÜLTHOFF, H. H. 2005. Low-Level image cues in the perception of translucent materials. *ACM Trans. Appl. Percep.* 2, 3, 346–382.

- GILLHAM, D. 2006. Real-time depth-of-field implemented with a postprocessing-only technique. In *Shader X5*, W. Engel, Ed. Charles River Media, Chapter 3.1, 163–175.
- GOSSELIN, D. 2004. Real-time skin rendering. In *Proceedings of the Game Developers Conference*.
- GREEN, S. 2004. Real-time approximations to subsurface scattering. In *GPU Gems*, R. Fernando, Ed. Addison-Wesley, Chapter 16, 263–278.
- HABLE, J., BORSHUKOV, G., AND HEJL, J. 2009. Fast skin shading. In *Shader X7*, W. Engel, Ed. Charles River Media, Chapter 2.4, 161–173.
- IGARASHI, T., NISHINO, K., AND NAYAR, S. K. 2005. The appearance of human skin. Tech. rep., Columbia University.
- JENSEN, H. W. AND BUHLER, J. 2002. A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph.* 21, 3, 576–581.
- JENSEN, H. W., MARSCHNER, S. R., LEVOY, M., AND HANRAHAN, P. 2001. A practical model for subsurface light transport. In *Proceedings of ACM SIGGRAPH*. 511–518.
- JIMENEZ, J. AND GUTIERREZ, D. 2008. Faster rendering of human skin. In *Proceedings of the CEIG*. 21–28.
- KELEMEN, C. AND SZIRMAY-KALOS, L. 2001. A microfacet based coupled specular-matte BRDF model with importance sampling. In *Proceedings of the Eurographics Short Presentations*.
- KOENDERINK, J. J. AND VAN DOORN, A. J. 2001. Shading in the case of translucent objects. *Proceedings of the SPIE*. vol. 4299, 312–320.
- MASTOROPOULOU, G., DEBATTISTA, K., CHALMERS, A., AND TROSCIANKO, T. 2005. The influence of sound effects on the perceived smoothness of rendered animations. In *Proceedings of the 2nd Symposium on Applied Perception in Graphics and Visualization (APGV'05)*. 9–15.
- NICODEMUS, F. E., RICHMOND, J. C., HSIA, J. J., GINSBERG, I. W., AND LIMPERIS, T. 1977. Geometrical considerations and nomenclature for reflectance. National Bureau of Standards.
- PHARR, M. AND HUMPHREYS, G. 2004. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, Chapter 8.4.2, 382–386.
- SINGH, M. AND ANDERSON, B. L. 2002. Perceptual assignment of opacity to translucent surfaces: The role of image blur. *Perception* 31, 5, 531–552.
- SUNDSTEDT, V., GUTIERREZ, D., ANSON, O., BANTERLE, F., AND CHALMERS, A. 2007. Perceptual rendering of participating media. *ACM Trans. Appl. Percep.* 4, 3.

Received July 2009; accepted August 2009