TECHNICAL REPORT

Subsurface scattering

Liana-Daniela Palcu

-117106643-

-CS6111-

-2017-

ABSTRACT

Nowadays, by looking at photographs or rendered images of translucent objects you can notice that they are very similar or even identical with the real world. This thing is possible due to the research which has been developed in producing efficient and accurate models of subsurface light transport: light enters into object's surface, is scattered around inside the material, and then exits the surface, potentially at a different point from where it entered. Although completely physically accurate simulations of subsurface scattering are out of the reach of current graphics hardware, it is possible to approximate much of the visual appearance of this effect in real-time. You can see in different images how subsurface scattering tends to soften the overall effect of lighting. Light from one area tends to bleed into neighbours areas on the surface, and small surface details become less visible. The farther the light penetrates into the object, the more it is attenuated and diffused. Subsurface scattering is critical for creating materials for all different kinds of surfaces like paper, marble, wax, and most importantly, skin. If there is no subsurface scattering on skin, then it won't look realistic because there is a distinct look that skin has, which is the level of translucency. [6],[7]

This technical report provides a description of the principles of Subsurface Scattering, with references to recent advancements in this area. Also, it describes the rendering techniques used to obtain this effect in Computer Graphics and what are the possible further developments in this research field.

INTRODUCTION

Scattering is a physical process where some forms of radiation are forced to deviate from a straight trajectory by one or more paths due to localized non-uniformities in the medium through which they pass. The reflection of light from most materials consists of two major terms: the **specular** and the **diffuse**.[1] **Specular reflection** represents the reflection off of smooth surfaces such as mirrors or a calm body of water. **Diffuse reflection** is generally considered to result from multiple scattering either from a rough surface such as clothing, paper, and the asphalt roadway or from within a layer near the surface.[5]

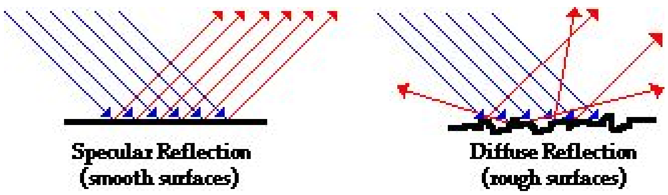

Figure 1 [5]

Subsurface scattering is a mechanism of light transport in which light penetrates the surface of a translucent object, is scattered by interacting with the material, and exits the surface at a different point. The light will generally penetrate the surface and be reflected a number of times at irregular angles inside the material, before passing back out of the material at an angle other the angle it would have if it had been reflected directly off the surface.[1]

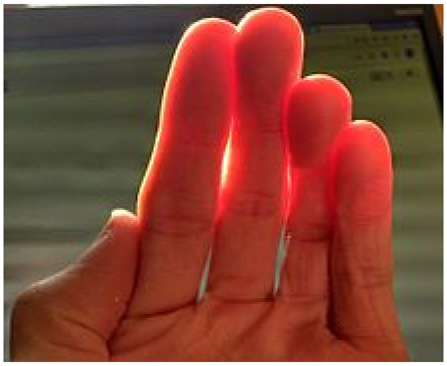

Figure 2

Computer graphics has succeeded to achieve this effect from real life by using different techniques and models presented below. Images are more closer to the reality due to new developments done by researchers in the recent years.

RENDERING TECHNIQUES

Simulating absorption using Depth Maps

One of the most important thing when you talked about subsurface scattering is the farther the light travels through material. The farther it goes, the more it is scattered and absorbed and this fact can simulates very translucent material. For simulating this effect, we need a measure of the distance light has traveled through the material.

One method of estimating this distance is to use **depth maths.** This technique consists of the following steps: first, the scene is rendered from the point of view of the light, storing the distance from the light to a texture. The resulted image is projected back onto the scene using standard projective texture mapping. In the rendering step, given a point to be shaded, we can look up into this texture to obtain the distance from the light at the point the ray entered the surface. By subtracting this value from the distance from the light to the point at which the ray exited the surface, we obtain an estimate of the distance the light has traveled through the object.[4]

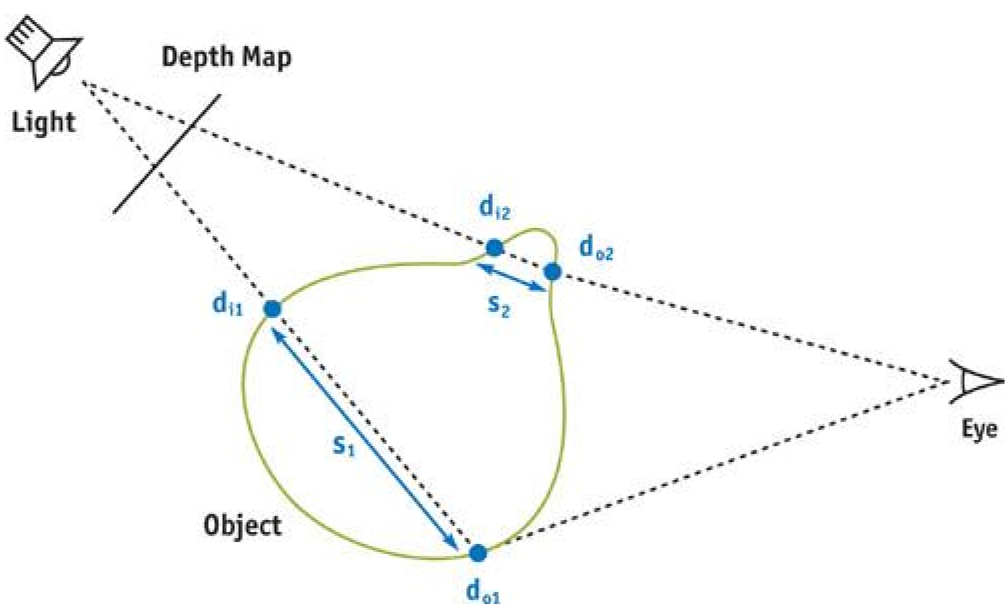

Figure 3

After we obtained a measure of the distance the light has traveled through the material, there are several ways it can be used: the distance can be **mapped** to a color using for example arrays as a data structure. The colour should fall of exponentially with distance. By changing this color map, and combining the effect with other, more traditional lighting models, we can produce images of different materials, such as marble or jade.[4]

Depth Maps has different advantages, but also some drawbacks.

★ **Advantages:**

- The precalculated maps that represents the approximate thickness of the surface at each point can take into account the direction of the incoming light by using depth map which is better for animating models.[4]

★ **Disadvantages:**

- It only works with convex objects: holes within the object are not accounted for correctly. This problem can be solved using **depth peeling**. This method has the advantage of being able to generate correct results for complex images containing intersecting transparent objects.[4]
- Another problem is that it does not simulate the way light is diffused as it passes through object.When the light is behind the object, you will often clearly see features from the back side of the object showing through on the front.[4]

Implementation details

On GeForce FX hardware, when reading from a depth texture, only the most significant eight bits of the depth value are available. This is not sufficient precision. Instead, it can be used either floating-point textures or the pack and unpack instructions from the NVIDIA fragment program extension to store 32-bit float value in a regular eight-bit RGBA texture. Floating-point textures do not currently support filtering, so block artifacts (noticeable distortions of the media) will sometimes be visible where the projected texture will appear larger than it is. If necessary, bilinear filtering can be performed in the shader, at some performance cost.[4]

More accurate simulations can be achieved by knowing the normal, and potentially the surface color, at the point at which the light entered the object. This can be made by rendering additional passes that render the extra information to textures. You can look up in these textures in a similar way to the depth texture. On systems that support multiple render targets, it may possible to collapse the depth, normal, and other passes into a single pass that outputs multiple values.[4]

Figure 4 [4]

More Sophisticated Scattering Models

More sophisticated models attempt to accurately simulate the cumulative effects of scattering within the medium.

One model is the **single scattering approximation**, which assumes that light bounces only once within the material. By stepping along the refracted ray into the material, one can estimate how many photons would be scattered toward the camera. Phase functions are used to describe the distribution of directions in which light is scattered when it hits a particle.It is also important to take into account the Fresnel effect at the entry and exit points. Fresnel effect represent the amount of reflectance you see on a surface which depends on the viewing angle.   This means that reflection seems to increase with distance. Reflections are strong in the distance, we see more clearer in the foreground that we do in the far distance.[4]

Another model, the **diffusion approximation**, simulates the effect of multiple scattering for highly scattering media, such as skin.[4]

Dipole approximation .Closed - form solution of diffusion can be obtained by placing two virtual point sources in and outside of the medium.[1]

Texture-Space Diffusion

Making images to look less computer-generated is one of the result of the subsurface scattering. The most obvious visual signs of this is a general **blurring of the effects of lightning.** 3D artists often emulate this phenomenon in screen space by performing Gaussian blurs of their renders in Adobe Photoshop and then adding a small amount of the blurred image back on top of the original.[4]

Diffusion can be simulated in texture space by following this steps: The mesh of the object is unwrapped with a vertex program that uses the **UV texture** coordinates as the screen position of the vertex. The program simply remaps the [0,1] range of the texture coordinates to the [-1,-1] range of normalized device coordinates. The object must have a good UV mapping this means that each point on the texture must map to only one point of the object, with no overlaps. By lighting this unwrapped mesh in the normal way, it results a 2D image representing the lighting of the object. Then this image can be processed and reapplied to it the 3D model like a normal texture.[4]

This technique is useful for other applications, because it decouples the shading complexity from the screen resolution: shading is performed only for each texel in the texture map, rather than for every pixel on the object. Points that are close in world space will map to points that are also close in texture space, if the UV parameterization of the surface is relatively uniform.[4]

Light diffusion can be simulated in image space by blurring the light map texture. To do this, it is used all the usual GPU image - processing tricks, such as **separable filters** or exploiting **bilinear filtering hardware**. Rendering the lighting to a relatively low - resolution texture already provides a certain amount of blurring. Figure 5 shows an unwrapped head mesh and the results of blurring the light map texture.[4]

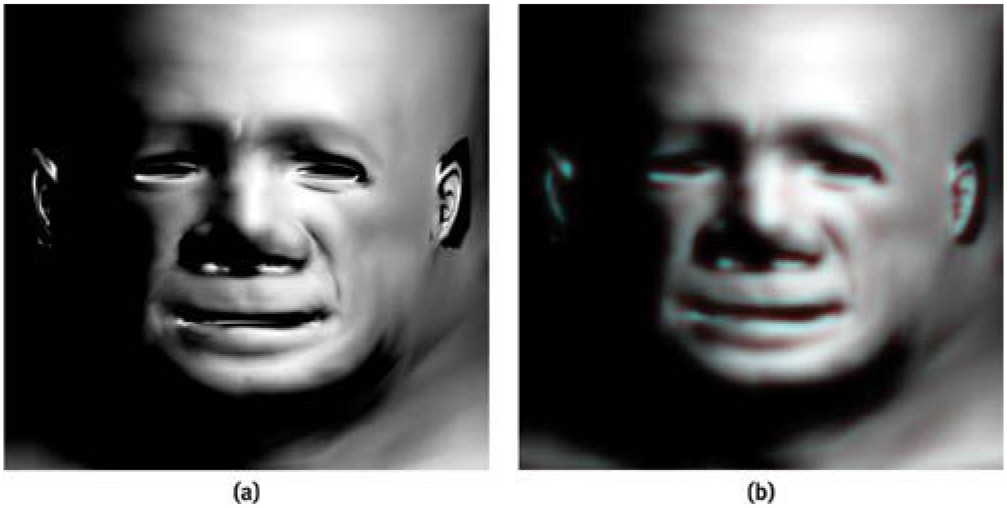

Figure 5 [4]

A diffuse color map can also be included in the light map texture: then details from color map will also be diffused. If shadows are included in the texture, the blurring process will result in soft shadows.

To achieve a **wider blur**, you can either apply the blur shader several times or write a shader that takes more samples by calculating the sample positions in the fragment program.Figure shows the blurred light map texture applied back onto the 3D head model.[4]

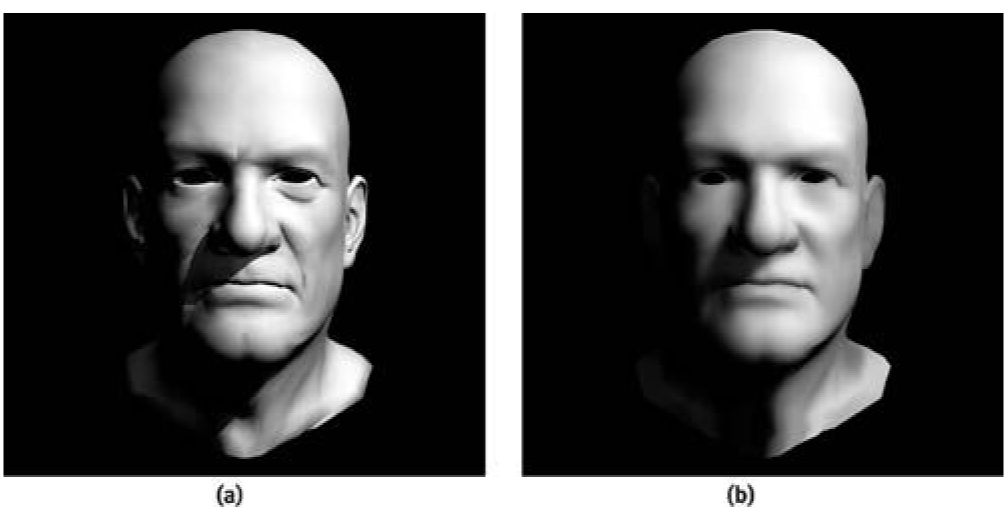

Figure 6 [4]

The final shader blends the diffused lightning texture with the original high-resolution color map to obtain the final effect, as shown in Figure 7. [4]

Figure 7

New advancements

Photon Beam Diffusion

Previously, PRMan has used the popular **dipole diffusion model** for subsurface scattering. However, the dipole diffusion model is overly smooth, and produces subsurface scattering which looks a lot like wax - which is fine if you're rendering a candle but not ideal if you are trying to render e.g human skin.[3]

A new model was developed in the recent years which is more advanced than the dipole model. The new model produces results that look similar to **quantized diffusion** but it is faster to evaluate and more numerically stable. The new model was developed in collaboration with Ralf Habel and Wojciech Jarosz at Disney Research Zurich. Technical details can be found in the paper :**Photon Beam Diffusion: A Hybrid Monte Carlo**

Method for Subsurface Scattering. The model divides the subsurface scattering into an accurate single-scattering term and a diffusion term that represents multi-scattering. Although the new model is more accurate, there is only a modest increase in render time.[3]

The following image shows a reference image of subsurface scattering computed using a brute-force Monte Carlo path tracer. This is an algorithm which integrates over all the illuminance arriving to a single point on the surface of an object. This illuminance is then reduced by a surface reflectance function (BRDF) to determine how much of it will go towards the viewpoint camera.[3]

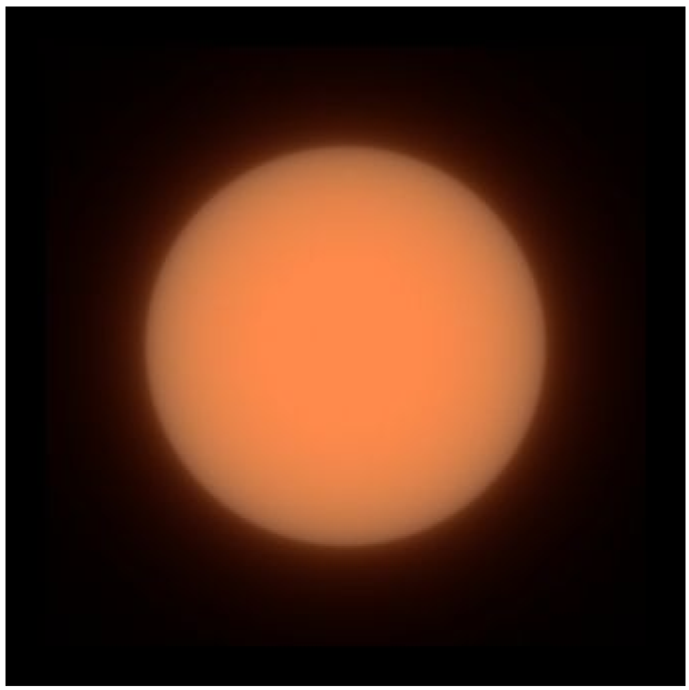

Figure 8.[3]

This look is what you are aiming for when rendering subsurface scattering- but it is much too slow to compute this way, so we use faster approximations as described below.

In the approximations, we often separate out **single scattering** from **multiple scattering**. To make this distinction clearer, in Figure 9 it is shown this separation for Monte Carlo reference image: the top third of the image shows single scattering, the middle third shows multiple scattering (up to hundreds of bounces), and the bottom third shows all scattering (single plus multiple). Tracing takes several hours of computation time even for this very simple scene (a square illuminated by a disk light source).[3]

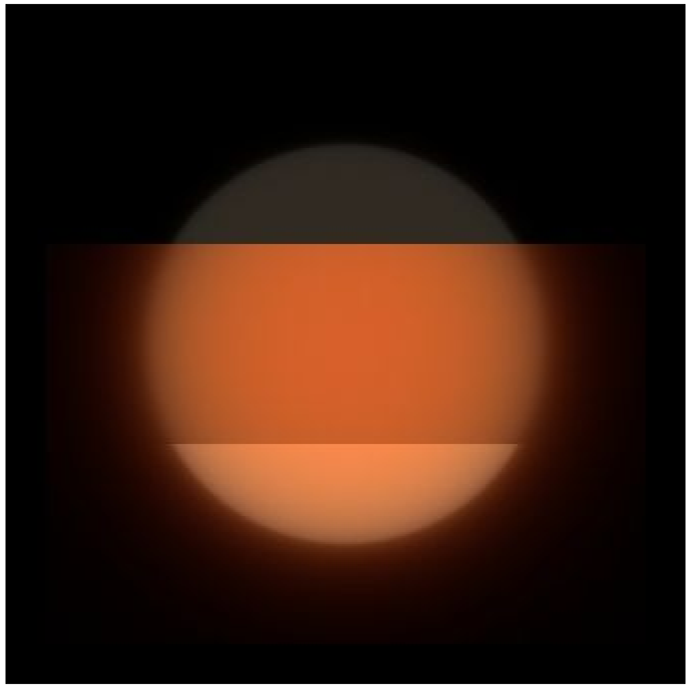

Figure 9.[3]

The image below shows six different approximations of the reference solution discussed above. The left column shows the old approximations, and the right column the new ones.

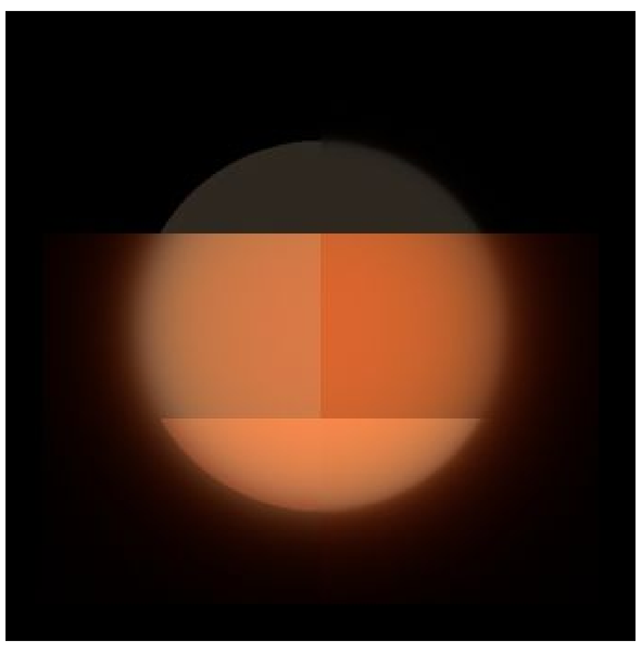

Figure 10.[3]

The **top row** shows two representations of **single scattering**. On the left is a matte material with ideal diffuse reflection (a BRDF: pure surface reflection, no subsurface scattering). On the right is the new single scattering term; it is an exact match of the Monte Carlo single scattering reference.The **middle row** shows two diffusion approximations of multiple scattering. On the left is the classic dipole diffusion model. It is too smooth and has the wrong hue. On the right hand is the new diffusion model (beam diffusion) which has the correct smoothness and hue.The **bottom** row shows two approximations of all subsurface scattering (single - plus multi scattering). On the left is a combination of the dipole with a weighted diffuse term. There is no way to combine these to obtain an edge with the right smoothness nor the right fall-of. On the right is the new single-scattering plus the new diffusion model. The new scattering model are significantly closer to the reference image than the other approximations.[3]

Single scattering - Our subsurface scattering formulation assumes that the light goes straight into the surface. From there, the light can scatter in any direction.For single-scattering it is considered only those scattering events that scatter the particle toward the surface.[3]

Diffusion approximation of **multi scattering** - In real life, particles may be scattered many, many times inside a volume before they are scattered back out of the surface. Previous approximations of this multi-scattering assumed a semiifinite plane and used a dipole. Multipoles are used to better approximate the multi scattering near volume edges and due to multiple layers of material with different scattering properties. The quantized diffusion method uses a sum over dipoles to improve the accuracy of the subsurface scattering on a semiinfinite plane - particularly to better model the correct sharpness and falloff.[3]

The trajectory of the particle is considered straight into the object as an extended light source. Multi scattering is computed by integrating along this light source, computing a dipole for each integration sample point. Hence the term **“beam diffusion”**.

Due to various optimizations and the use of table lookups, **beam diffusion** (with single scattering) is typically only around 15 percent slower than dipole diffusion. But in practice , many people have found that they are needed to compute two dipole diffusions: one to simulate single - scattering and one to simulate multi-scattering. If comparing two dipole diffusion computations with one beam diffusion with single scattering, **the latter is clearly faster.[3]**

Choice of technology

One of the **technologies** used to simulate various aspects of subsurface scattering is **Pixar's RenderMan(PRMan)**. The images below illustrate a simple example of subsurface scattering with PRMan. The scene consists of a teapot made of a uniform, diffuse material. The teapot is illuminated by two spherical area lights (casting ray-traced shadows), one very bright light behind the teapot and another light to the upper left of the teapot. There is no subsurface scattering in the first image, so the parts of the teapot that are not directly illuminated are completely back. In the second image, the material of the teapot has subsurface scattering with a relatively short mean path length, so the light can penetrate this parts, such as the handle, knob, and spout, but very little light makes it through the teapot body. In the third image, the mean path length is longer, so more light can penetrate the material and even the teapot body is brighter.[2]

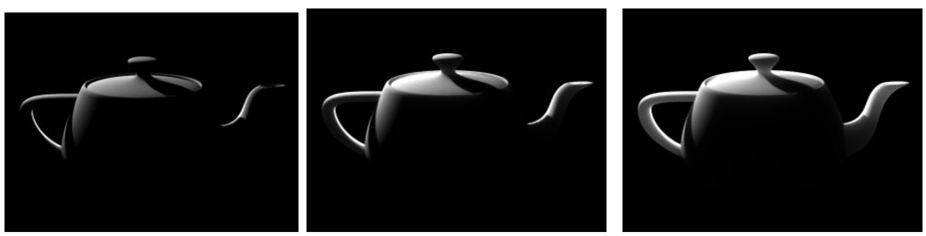

Figure 11.

Another program which use Subsurface Scattering is **Blender**. It calculates SSS in two steps:

At **first** the irradiance, or brightness, of the surface is calculated, from the front side of the object as well as from its back side. This is pretty much the same as in a normal render. Ambient Occlusion, Radiosity, the type of diffuse Shader, the light color, etc. are taken into account.[6]

In the **second** step, the final image is rendered, but now the SSS shader replaces the diffuse shader. Instead of the lamps, the calculated lightmap is used. The brightness of a surface point is the calculated "Average" of the brightness of its surrounding points. Depending on your settings the whole surface may be taken into account, and it is a bit more complicated than simply calculating the average.[6]

Figure 12

Possible Future Work

- ★ One possible extension to the depth map technique would be to render additional depth passes to account for denser objects within the object, such as bones within a body. The problem is that it is wanted volumetric effects using a surface-based representation. Volume rendering does not have this restriction, and it can produce much more accurate rendering of objects whose density varies.[4]
- ★ Another possible extension to this technique would be to provide several color maps each representing the color of a different layer of skin. For example, you might provide one map for the surface color and another for the veins and capillaries underneath the skin.[4]
- ★ Another technique is one which handles arbitrary polygonal objects by first adding up the distance of all the back-facing surfaces and then subtracting the distances of all the front-facing surfaces. His application computes distances in screen space for volumetric fog effects, but it could be extended to more general situations.[4]
- ★ An interesting area of future research is combining the depth-map and texture-space techniques to obtain the best of both worlds.[4]

CONCLUSION

The effects of subsurface scattering are an important factor in producing convincing images of skin and other translucent materials.By using several different approximations, it can be achieved much of the look of subsurface scattering today in real time. As graphics hardware becomes more powerful, increasingly accurate models of subsurface light transport will be possible.[4]

Figure 13

Figure 14

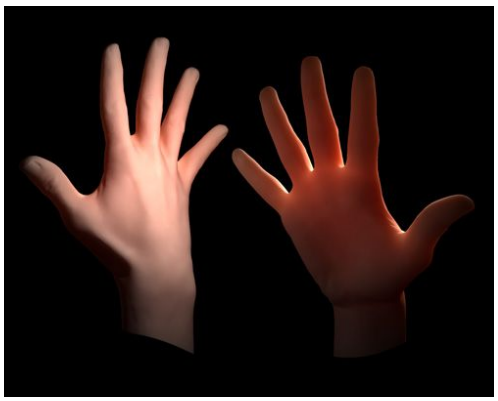

Figure 15

REFERENCES

[1] Wikipedia. 2017. WikipediA: the Free Encyclopedia. Retrieved from https://www.wikipedia.org/.

[2] “Documentation.” *Renderman Documentation*, weber.itn.liu.se/~stegu/pixar/RenderMan_20/subsurface.html.

[3]“Documentation.” *Renderman Documentation*, weber.itn.liu.se/~stegu/pixar/RenderMan_20/improvedSubsurface.html.

[4]"GPU Gems." *NVIDIA Developer*,
developer.nvidia.com/gpugems/GPUGems/gpugems_ch16.html.

[5]"Specular vs. Diffuse Reflection." *The Physics Classroom*,
www.physicsclassroom.com/class/refln/Lesson-1/Specular-vs-Diffuse-Reflection.

[6]"Subsurface Scattering¶." *Subsurface Scattering — Blender Manual*,
docs.blender.org/manual/en/dev/render/blender_render/materials/properties/subsurface_scattering.html.

[7]"Understanding Subsurface Scattering - Capturing the Appearance of Translucent Materials." *Pluralsight*, 7 June 2014,
www.pluralsight.com/blog/film-games/understanding-subsurface-scattering-capturing-appearance-translucent-materials.