Monte Carlo Subsurface Scattering

Gregory White

May 5, 2011

1 Motivation

Subsurface scattering is a technique that has been of great interest to me for some time now. As a 3D lighting artist, the technique has played a huge part in the quality of my renders. My goal with this project was to gain a deeper understanding of the technique so that I may use it in a more intelligent and efficient manner within my art.

Figure 1: Example of subsurface scattered light; left to right: skim milk, whole milk, white paint

2 Algorithm

After reading the Pharr/ Hanrahan paper, I had a much better understanding of Monte Carlo subsurface scattering. However, the math behind this technique was out of my league. So my algorithm is an artist/programmers abstraction of the method described in this paper. I sought to extract the key elements of the algorithm while leaving out the complexity that at this point is a little beyond my scope. My algorithm fits into the parallel raytracing function that computes the shadow and reflection rays. After shadows are calculated, if the ray hits the sphere, the ray is extended by a fixed amount into the sphere. The color is calculated at that position within the sphere. Color is determined by looking at the depth to which the ray penetrates the sphere. The ray is then shifted by a random unit vector and extended again by the fixed amount. The color at each position is accumulate until the loop terminates. The loop terminates either after a predetermined amount of bounces or if the ray exits the sphere. Upon termination, the color is divided by the number of bounces and added to the answer. The algorithm is slower than the typical raytracing solution, but definitely not as slow as expected.

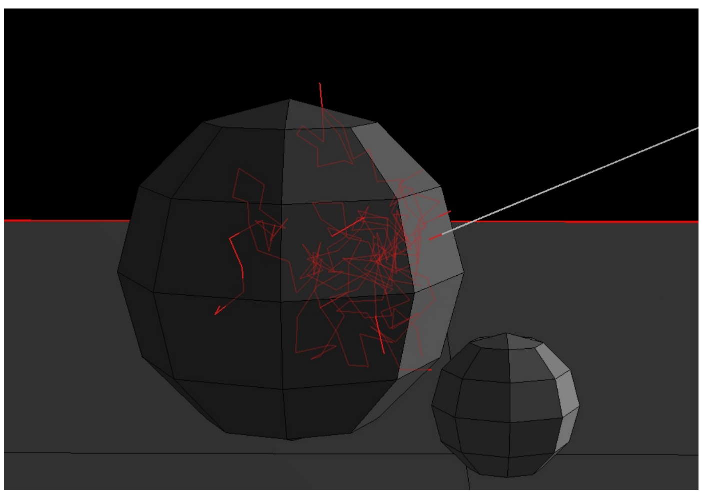

Figure 2: Visualization of scattered Rays;

This algorithm, as previously stated, is an abstraction of the true Monte Carlo Subsurface Scattering. My algorithm randomly decides ray direction and extension length. In the Monte Carlo solution, the ray follows a specific set of rules that describe each possible situation the ray could encounter. Each of these situations is represented by its own equation. These equations can be combined to produce the correct result.

3 Results

My result is by no means perfect. However, it does produce a result akin to accurate examples. The images below are iterations of my attempt at milk. The noise present within the images is a con of my solution. I also had problems with a thin dark rim around my sphere. Despite it actually looking pretty interesting, it is not a desirable result. I can remove it by decreasing my initial ray extension to a very small number but this also decreases the overall effect of the subsurface scattering. My solution also does not handle color very well. I converted RGB to HSV to vary color according to depth but it didnt work quite as planned.

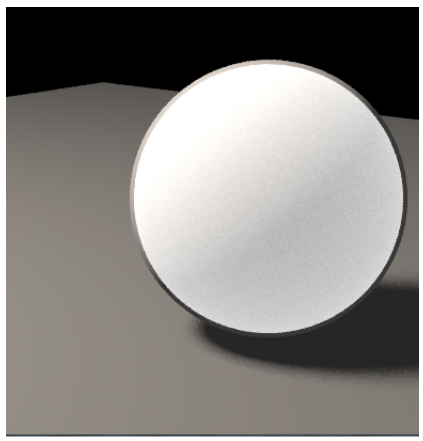

Figure 3: Attempt to recreate SSS effect; best result

4 Conclusion

Overall, my solution produces an interesting and semi-accurate result. It is not nearly at the level of the solutions by Pharr/Hanrahan or Jensen and also not quite at the level of industry programs such as Maya or 3DS Max. However, implementing it has definitely expanded my knowledge of the technique and that is my ultimate goal.