Advanced Computer Graphics - Sub-Surface Scattering

Liam Gibbs

December 2018

1 Introduction

For my advanced topic I chose to do sub-surface scattering, using Jensen's 2001 paper 'A Practical Model for Subsurface Light Transport'. This outlines a method for approximating sub-surface scattering using a bidirectional surface scattering distribution function (BSSRDF). Sub-surface scattering is used to model translucent materials to achieve a more realistic look because in real life most objects are at least slightly translucent, skin being a good example. Otherwise, objects look completely opaque. It accounts for refracted light that reflects once or more times while inside the object before leaving, giving a much smoother appearance. A bidirectional reflectance distribution function (BRDF) is a simplified version of the BSSRDF as it assumes light leaves an object at the same point it enters. For this reason it creates a very rigid look, only capable of modelling opaque materials. Whereas the BSSRDF describes light that can enter at one point and leave at another. The approximation I use allows us to approximate the appearance of various different materials using different co-efficients such as marble, skin, milk, etc. (Jensen et al, 2001) Below is an example of the difference between the use of a BRDF and BSSRDF.

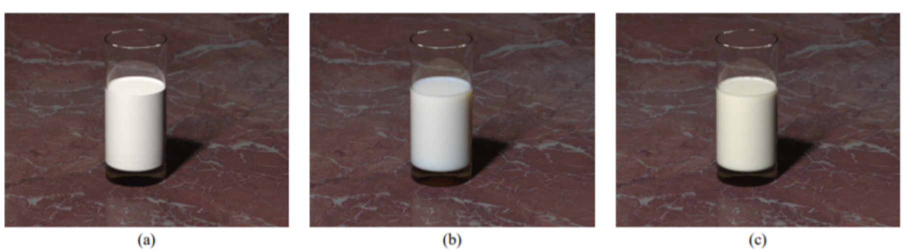

Figure 1: (a) BRDF (b) Skim milk (c) Whole milk (Jensen, 2001)

Jensen et al (2001) consider various different approaches including using photon mapping and solving the full radiative transfer equation but these can be expensive in terms of CPU time and memory. They propose an approximation made up of two parts, a single scattering term and multiple scattering term. The single scattering accounts for light that only scatters once when entering a material before leaving, i.e. the outgoing ray and incoming refracted ray intersect. We need both terms as the multiple scattering term does not account for single scattering.

2 Theory

In this section I'll explain the maths behind the different terms and how they are applied practically. The translucency and general appearance of the object is given by two key values in this approximation. The reduced scattering coefficient, σ_s', and the absorption coefficient, σ_a. There are various example values for different materials given by Jensen et al some of which will be illustrated later in this report. From σ_s' and σ_a we can get the extinction coefficient which will be useful later on. The extinction coefficient is given by $\sigma_t = \sigma_s + \sigma_a$.

2.1 Multiple Scattering

Multiple scattering is calculated using the diffusion approximation, this relies on the fact that highly scattering materials tend to become isotropic. This means the light scatters in all different directions, slowly becoming equal in all directions which essentially makes it blurred, giving a smooth appearance. Most materials have this property, apart from metals which are opaque. The diffusion equation is given below, derived by Jensen et al (2001).

$$D\nabla^2\phi(x) = \sigma_a\phi(x) - Q_0(x) + 3D\vec{\nabla} \cdot \vec{Q}_1(x).$$

In the case of a single isotropic light source it gives the below solution. where

$$\phi(x) = \frac{\Phi}{4\pi D}\frac{e^{-\sigma_{tr}r(x)}}{r(x)},$$

$\phi(x)$ is the radiant energy on that surface, Φ is the power of the light source, r is the distance to the light source and $\sigma_t r$ is the effective transport coefficient. Given by, $\sigma_{tr} = \sqrt{3\sigma_a\sigma_t'}$. This assumes an infinite medium. In the case of a finite medium, such is ours, we need a boundary condition. The net inward diffuse flux at each point should be zero. Jensen et al (2001) derive the boundary condition for when the light travels through two different mediums.

$$\phi(x_s) - 2AD(\vec{n} \cdot \vec{\nabla})\phi(x_s) = 0.$$

Where A is given by,

$$A = \frac{1 + F_{dr}}{1 - F_{dr}}.$$

here F_{dr} is the average diffuse Fresnel reflectance, which is approximated rather than be computed analytically as this would be more expensive,

$$F_{dr} = -\frac{1.440}{\eta^2} + \frac{0.710}{\eta} + 0.668 + 0.0636\eta.$$

And $D = 1/3\sigma'_t$, the diffusion constant.

Given the boundary condition, we can calculate the diffuse BSSRDF, R_d. Jense et al (2001) mention that exact formulas exist to calculate scattering but mostly involve infinite sums or solving partial differential equations. Instead, they explore using a dipole which places two virtual light sources, one positive (above the surface) and one negative (below the surface). These must be placed in such a way to satisfy the boundary condition. The 'real' positive light source is placed below the surface at a distance of $z_r = \frac{1}{\sigma'_t}$ while the 'virtual' negative light source is placed above the surface at a distance of $z_v = \frac{1+\frac{4}{3}A}{\sigma'_t}$. The diffusion component is then derived using the dipole, where d_r is the distance from x, which will be our sampled point, to the real source point and d_v is the distance from x to the virtual source point.

$$\begin{aligned} R_d(r) &= -D\frac{(\vec{n} \cdot \vec{\nabla}\phi(x_s))}{d\Phi_i} \\ &= \frac{\alpha'}{4\pi}\left[(\sigma_{tr}d_r + 1)\frac{e^{-\sigma_{tr}d_r}}{\sigma'_t d_r^3} + z_v\,(\sigma_{tr}d_v + 1)\frac{e^{-\sigma_{tr}d_v}}{\sigma'_t d_v^3}\right]. \end{aligned}$$

Then taking into account Fresnel reflectance we have S_d for our BSSRDF diffusion term. To apply this practically, we sample random points on the surface

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

of the object and calculate the distances between that and the light sources, d_r and d_v. Giving us a value for the BSSRDF diffusion term. We'll use multiple samples to make the image as accurate as possible using a density function to weight each sample, based on how far away the random point is to the outgoing ray's intersection point. It is given as follows $\sigma_{tr}e^{-\sigma_{tr}d}$ where d is the distance. As the distance increases the sample will be given less weight towards the final value (Jensen, 2001).

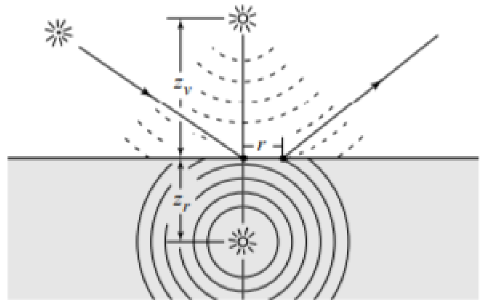

Figure 2: A single dipole source for a diffuse approximation.

2.2 Single Scattering

As stated previously the diffusion approximation does not account for single scattering events so we have to model this separately. Jensen et al (2001) extend Hanrahan and Krueger's (1993) BRDF model that "analytically computes the total first-order scattering from a flat, uniformly lit, homogeneous slab" to work for their BSSRDF model. The outgoing radiance can be computed by integrating along the outgoing ray using the equation below where x_o is the point the ray leaves the material and ω_o is the direction of the outgoing ray. x_i and w_i are the corresponding incident ray terms.

$$\begin{aligned} L_o^{(1)}(x_o, \vec{\omega}_o) &= \sigma_s(x_o) \int_{2\pi} F\, p(\vec{\omega}_i' \cdot \vec{\omega}_o') \int_0^\infty e^{-\sigma_{tc} s} L_i(x_i, \vec{\omega}_i)\, ds\, d\vec{\omega}_i \quad (6) \\ &= \int_A \int_{2\pi} S^{(1)}(x_i, \vec{\omega}_i; x_o, \vec{\omega}_o)\, L_i(x_i, \vec{\omega}_i)\, (\vec{n} \cdot \vec{\omega}_i)\, d\omega_i\, dA(x_i). \end{aligned}$$

Above $F = F_t(\eta, \overrightarrow{\omega_o}) F_t(\eta, \overrightarrow{\omega_i})$ which represents the fresnel transmission of the light entering and leaving the material combined. The fresnel equations represent the transmission and reflectance of light between two different mediums (Lvovsky, 2013). Lvovsky (2013) provides a derivation for these equations and discusses their application.

This can be applied practically by taking samples along the outgoing ray to determine the single scattering term (Monte-Carlo integration). The below equation describes how to calculate the single scattering value for a certain sampled point. We choose the random distance using $s_o' = \log(\epsilon)/\sigma_t(x_o)$ where

$$L_o^{(1)}(x_o, \vec{\omega}_o) = \frac{\sigma_s(x_o) F p(\vec{\omega}_i \cdot \vec{\omega}_o)}{\sigma_{tc}} e^{-s_i' \sigma_t(x_i)} e^{-s_o' \sigma_t(x_o)} L_i(x_i, \vec{\omega}_i).$$

ϵ is a uniformly distributed random number between 0 and 1. We take as much samples as necessary and then take an average. s'_i is the refracted distance the ray travels through the material and can be determined using Snell's law which states that the ratio of the sines of the angle of incidence and refraction is equivalent to the ratio of indices of refraction, η.

$$s'_i = s_i \frac{|\vec{\omega}_i \cdot \vec{n}_i|}{\sqrt{1 - \left(\frac{1}{\eta}\right)^2 (1 - |\vec{\omega}_i \cdot \vec{n}(x_i)|^2)}} .$$

2.3 Fresnel Term

The terms are scaled by the Fresnel term as the amount of refraction is dependent on the angle of incidence. As it increases refraction will generally decrease while reflection increases. There are two types of Fresnel equations which refer to different types of polarization, P and S. P describes the polarization of an electric field where the electric field vector is in the plane of incidence. S describes it where the electric field vector is perpendicular to the plane of incidence. When light is unpolarized they can be used equally, so in our case we take the average of the two for our approximation. (Lvovsky, 2013)

$$r_P = \frac{(n_1/\mu_1)\cos\theta_t - (n_2/\mu_2)\cos\theta_i}{(n_1/\mu_1)\cos\theta_t + (n_2/\mu_2)\cos\theta_i}$$

$$r_S = \frac{(n_1/\mu_1)\cos\theta_i - (n_2/\mu_2)\cos\theta_t}{(n_1/\mu_1)\cos\theta_i + (n_2/\mu_2)\cos\theta_t}$$

The Fresnel reflectance for P and S polarization is given above, from the derivation provided by Lvovsky (2013). The transmission is found by taking the reflectance away from one. As the reflected and refracted light will always add to the total incoming light.

3 Implementation

To start with I realised I needed to refactor some of my original code so I could add the sub-surface scattering in a way that made sense. This involved adding a scene class which held references to all the objects and the light. The scene would then be given a ray to trace and figure out what the colour of that particular pixel should be, including any shadow calculations, etc. First I would figure out the object with the closest intersection to the ray and process the pixel for that point. I then calculate the single scattering contribution if the object is translucent.

3.1 Single Scattering

Firstly, when calculating the single scattering term I determine the maximum distance the sampled point can go into the object (i.e. where it intersects at the other side). This caused a lot of issues while developing, I would often get triangle artefacts in my image, seen below. The problem was that some

Figure 3: Rendered image with artefacts caused by mesh

models had holes or gaps in them, mostly in the bottom. So when calculating the maximum distance it would return the maximum float value. This would give me an inconsistent image as it would be sampled outside of the object. To resolve this, initially I set the maximum distance as the intersection with the bounding box of the object where the ray passes through. Though this did improve the problem there were still clear dark patches in the image and inconsistencies. In the end the only other solution was to get a complete model with no gaps which wasn't too difficult in the end. I then iterate through the samples, randomly sampling a point along the refracted outgoing ray using the method described in section 2.2. I also made sure the point wasn't too close to the intersection point to avoid any rounding areas where it might be seen as on the surface of the object. In the case where there is a problem with the model I still use the bounding box intersect instead. To get the ray of incidence's intersection point and normal I cast a ray from the sample point to the light. In the case where it does not intersect with the object (presumably because of an issue with the model), I skip that sample. This may still produce artefacts but it is an extreme edge case which shouldn't really occur. To help debug this I've added in a warning print statement to indicate there maybe an issue with that sample. The rest of the single scattering implementation follows the theory given by Jensen et al (2001).

3.2 Multiple Scattering

After I calculate the single scattering contribution, I then calculate the multiple scattering term. This was simpler than the single scattering as the calculation was more straight-forward. I start by iterating through each sample. First finding a random point on the same face as the intersection of the object, using the density function, $\sigma_{tr} e^{-\sigma_{tr} d}$, I calculate the weight of that sample. I take the sample randomly from any point on the bunny. Ideally I would have liked to implement the random sampling so it would take a radius and it would find a point on the bunny within that radius. This is very expensive though as it involves randomly sampling until you find one within range, so I decided against it as the results from any random point were quite strong anyway. I then calculate the dipole source positions, placing the virtual source $1/\sigma_t' + 4AD$ above the surface and the real source $1/\sigma_t'$ below the surface to satisfy the boundary condition. I then calculate the multiple scattering contribution using the diffuse approximation equation.

3.3 Other features

I also implemented a bounding box to speed up the rendering time, to implement this only required a minimum and maximum point to define the box. It was simple to find the points, when I loaded in the mesh I simply take note of the lowest and highest coordinates and then pass them to the bounding box. We assume the box is aligned with our world axis, making it easier to calculate an intersection. I also attempted interpolating normals at an intersection point, I did it successfully but found that it gave strange results when using it in conjunction with sub-surface scattering. It seems the approximation needs a flat surface to work properly which Jensen et al (2001) do mention, that the geometry should be locally flat, whereas interpolating the normals gives the illusion of a smooth curved surface. For these reasons I decided against using it as it gave the object a very flat look, the code is still there but is unused.

Figure 4: Sub-surface scattering when interpolating normals.

I also implemented shadows in conjunction with my sub-surface scattering by doing a simple shadow check for each sample in single and multiple scattering. The problem with this is it is very expensive to do for every sample on every raytrace, especially when there is hundreds or even thousands of samples. For this reason I put a flag in to turn shadows on/off to allow for quicker run times. An extension of this would be to shoot multiple sample shadow rays when doing the shadow test for soft shadowing which fades the shadow off instead of a having a sharp edge. Again, this would be even more expensive to compute.

4 Results

A rendering of the end result can be seen below, shadows are omitted here due to time constraints but this will be seen in the final image submission.

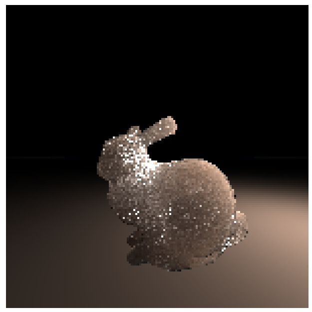

Figure 5: A rendering of a marble bunny.

The material coefficients used here are for marble. As you can see there is a clear smoothing effect on the bunny with the light fading off towards the back. You can also see the transluceny in the ears and head as these parts of the bunny are brighter than others with multiple scattering taking greater effect here. This is because these parts of the are thinner than other parts of the bunny so naturally they are brighter. In the end it was a successful implementation of sub-surface scattering minus a few limitations. Given more time I would have liked to implement it in parallel and added effects such as soft shadows for greater realism. A limitation of the raytracer is that it needs a good number of samples to avoid noisy artefacts, though this is often the case with a number of raytracer features which results in large computation times. Although, generally this approximation is much quicker and is supposedly just as accurate as other methods (Jensen et al, 2001).

5 Bibliography

1. Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan, 'A Practical Model for Subsurface Light Transport', 2001

2. Alexander I. Lvovsky, 'Fresnel Equations', 2013

3. P. Hanrahan and W. Krueger, 'Reflection from layered surfaces due to subsurface scattering', In ACM Computer Graphics (SIGGRAPH'93), pages 165–174, August 1993.