Directional Dipole Model for Subsurface Scattering

Pan An (A0134556A)

April 28, 2016

Contents

1 Introduction

Jensen et al. proposed a method for rendering the effects of subsurface scattering by using the Bidirectional Surface Scattering Reflectance Distribution Function (BSSRDF). Subsurface scatter- ing is caused when light scatters one to multiple times in a semi-translucent object, and is very important in a variety of different objects, such as milk or jade. Without this effect, rendered objects have a rough and hard computer generated feel to them.

This effect is simulated in a ray tracer by casting samples over the surface of the object and into the object. Original BSSRDF renders translucent materials using Monte Carlo ray tracing is computationally expensive due to a large number of subsurface scattering events.

This report is the implementation of an analytical model for subsurface scattering which captures translucency effects that are present in the reference solutions but remain absent with existing models. The idea is that a ray source corresponds better to the light that refracts through the surface of a translucent material.

2 Related Work

Subsurface scattering(SSS), is an optical physics based machanism describing the process of light penetrating translucent materials. Different models have been proposed in order to produce artifitial images of real life materials. Bidirectional reflectance distribution [4] function was introduced as a simple but efficient model for reflection of light at the surface of objects. Jensen et. al. [3] introduced an improved model: bidirectional subsurface scattering reflection distribution function(BSSRDF).

A list of some physical caracteristics of different materials has been measured in Jensen's work. Gkioulekas et. al. [1] studied on the physical characteristics of daily materials. Gkioulekas et. al. used a series of techniques and algorithms in order to achieve a set of data for daily materials such as wine, milk, coffee etc. Based on their analysis we are able to achieve a better result in multimedia rendering.

3 Platform and Frameworks

My code is running on my personal computer, the specifications is listed:

Hardware	Params	Additional
CPU	i7	4700MQ
GPU	NVIDIA	GTX 780m
RAM	16GB	
DRAM	3GB	
OS	Windows 7	64-bit

Upenn(University of Pennsylvania) has open sourced one of their framework for global illumination and rendering. My project is based on the ray tracing framework of Upenn. Some simplified version of the calculations are also provided for analysis of the algorithm.

4 Theory

Subsurface scattering (SS) is a physical phenomenon that naturally occurs in a wide range of natural materials. A BSSRDF is a function S between two points $\mathbf{x}_i$ and $\mathbf{x}_o$ on the surface of an object that describes the repation between an element of emergence radiance $dL(\mathbf{x}_o, \vec{\omega}_o)$ and an element of incident flux $d\Phi(\mathbf{x}_i, \vec{\omega}_i)$:

$$S(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o, \vec{\omega}_o) = \frac{dL(\mathbf{x}_o, \vec{\omega}_o)}{d\Phi(\mathbf{x}_i, \vec{\omega}_i)}$$

We can then use the BSSRDF in the general formulation of the rendering equation, obtaining:

$$\begin{aligned} L_o(\mathbf{x}_o, \vec{\omega}_o) &= L_e(\mathbf{x}_o, \vec{\omega}_o) + L_r(\mathbf{x}_o, \vec{\omega}_o) \\ &= L_e(\mathbf{x}_o, \vec{\omega}_o) + \int_A \int_{2\pi} S(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o, \vec{\omega}_o) L_i(\mathbf{x}_i, \vec{\omega}_i)(\vec{\omega}_i \cdot \vec{n}_i) d\vec{\omega}_i dA \end{aligned} \tag{1}$$

Normally the BSSRDF term is split into two or more additional terms in order to achieve a better approximation of the real world, accounting for single and multiple scattering. In case of multiple scattering, i.e. when light bounces multiple times inside the material, the radiance becomes largely isotropic, and the whole process can be treated as a diffusion [2].

Directional subsurface scattering

In Jensen's work, based on approximations of the diffusion equation, the BSSRDF S is modeled as two points lights positioned close to $\mathbf{x}_i$, and depended on the distance between the points and the scattering parameters. In the model we are considering for our thesis, proposed by Firsvad et al., we use a dipole of ray sources in order to better approximate the diffusion equation. The derived BSSRDF describes effectively the diffusion on an infinite medium, so some corrections are necessary in order to take into account the boundary conditions.

Approximation

The general idea of directional dipole method is to integrate Equation 1 numerically. In order to do this, we need to make some assumptions. Given

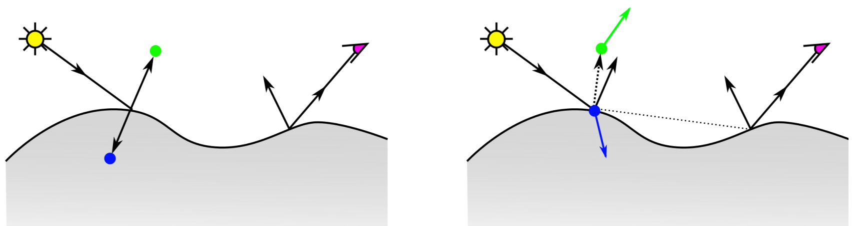

Figure 1: Standard dipole (on the left) versus directional dipole (on the right).

an emergence point $\mathbf{x}_o$. The diffusive part of the proposed directional dipole BSSRDF is as following:

$$S_d(\mathbf{x}_i, \vec{\omega_i}; \mathbf{x}_o) = S'_d(\mathbf{x}_0 - \mathbf{x}_i, \vec{\omega_{12}}, d_r) - S'_d(\mathbf{x}_0 - \mathbf{x}_v, \vec{\omega_v} d_r)$$

where S'_d is the directional version of diffusive approximation.

5 Implementation

The implementation of this new method is based on the hardware that has been described before. I used an open source framework provided by Upenn as the basic structure for displaying, calculation and error controlling.

In order to build a platform for BSSRDF, I implemented some basic functionalities of ray tracing. As my hardware power was limited, I simplified a couple things in order to test faster.

5.1 Basic Functions

Basic ray tracer model is necessary for rendering multiple objects. This GPU based path tracer with global illumination and anti-alising can render diffuse, perfect specular objects, transparent and subsurface scattering materials. Depth of Field was also implemented with some twist.

However since I did not have time to implement mesh geometry, I was not able to produce shapes other than cube and sphere.

5.2 Subsurface Scattering

Both BSSRDF and directional dipole method were implemented. If you want to test, simply change the "BSSRDF" value of a material, then new an object into one of the .txt files in the scenes folder.

Basic BSSRDF was implemented with a fast logarithmic distribution function. Directional dipole, however, was implemented with a Holsen sequence generator, which is not very time efficient. A simple BSSRDF implementation:

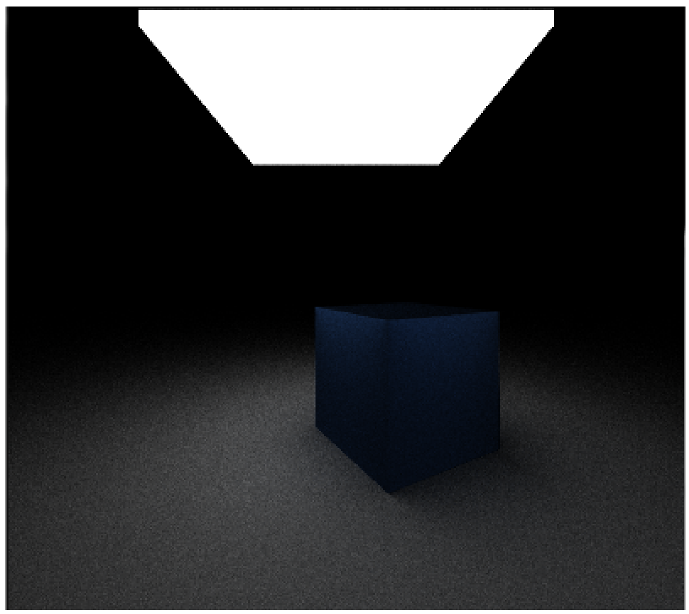

Figure 2: Left: BSSRDF. Right: Directional Dipole

Below is an example of comparison between BSSRDF and directional dipole method:

6 Results and Proposal for Further Research

The rendering process showed a good result of both BSSRDF and Directional Dipole methods. As we can see that the light around the object of BSSRDF is near average, and the result of directional dipole shows some difference. Figure. 5 shows the relationship between the input light angle and the diffusion result.

However during the rendering process, we can see that the result is quite noisy. We can see that some of the details in the final image are missing, but the result is good enough to trick the observer's eye. Regarding the performance, preliminary results give a rendering time of 0.37 seconds per

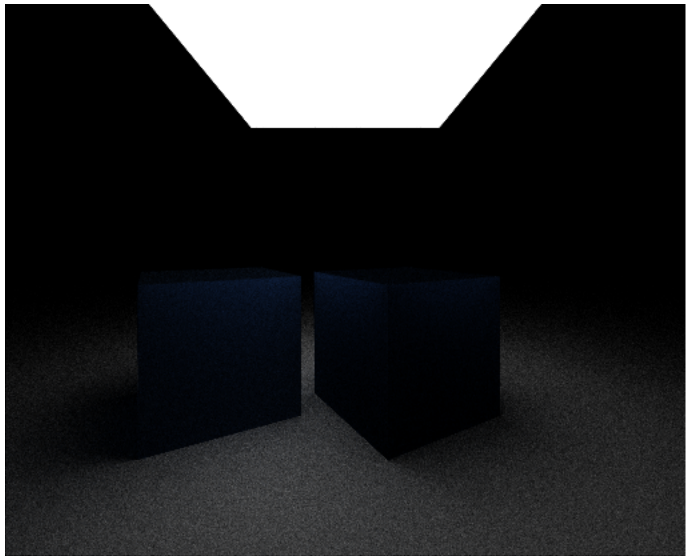

Figure 3: Left: BSSRDF. Right: Directional Dipole

frame on a low-end modern GPU. Directional method with different shapes are also tested as shown in Figure. 4:

Further steps should be focused on a more generalized version of directional dipole for mesh geometry objects. Also during the rendering process directional dipole method was implemented with a Holsen Sequence, which is not O(1) in this case. A better way to get exponential sampling is by utilizing a combination of number theory and Taylor expansion, which is explained in the presentation.

7 Acknowledgements

I want to thank professor Yin for offering to help with the hardware. The platform that I used to develop is provided by CIS565 GPU programming course. Part of my report material comes from DTU github repository, will be adding license information if needed. And special thanks to professor WengFai Wong, for the guidance of low level optimizations.

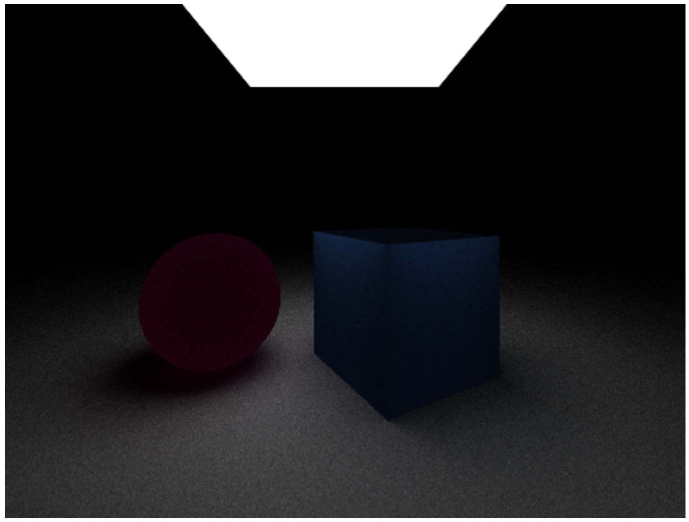

Figure 4: Radiance difference with different angles.

References

[1] Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries: Supplementary material.

[2] Akira Ishimaru. *Wave propagation and scattering in random media.* IEEE, 1997.

[3] Henrik Wann Jensen, Stephen R Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518. ACM, 2001.

[4] FE Nicodemus, JC Richmond, JJ Hsia, IW Ginsberg, and T Limperis. Geometric considerations and nomenclature for reflectance, volume 161 of monograph. *National Bureau of Standards (US)*, 1977.

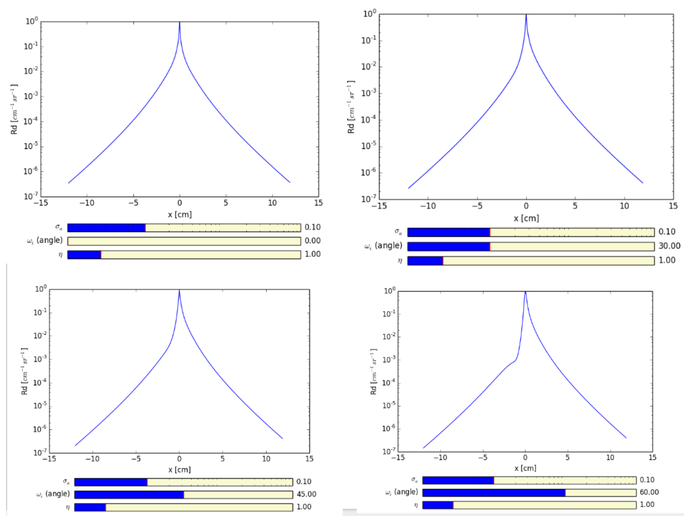

Figure 5: Radiance difference with different angles.