Real-Time Spherical Harmonics Based Subsurface Scattering

Anna Tomaszewska and Krzysztof Stefanowski

West Pomeranian University of Technology in Szczecin,
Faculty of Computer Science,
Żołnierska 49, 71-210, Szczecin, Poland
atomaszewska@wi.zut.edu.pl

Abstract. In this paper we present an algorithm for a subsurface scattering simulation (SHSS) based on spherical harmonics. The approach has a physical basis. As we focus on specifying an algorithm suitable for commercial games, it contains simplifications designed for real-time calculations. Spherical harmonics (SH) functions were used to encode the thickness of an object in every possible direction for each vertex of a graphical model. The information about the thickness of an object is the basis for a simulation of light absorption. The starting point of our approach was the Green method [3], where the thickness is calculated by a shadow casting algorithm. In our technique the model volume is encoded by spherical harmonics similarly to the Precomputed Radiance Transfer where the SH are used to encode the values of a transfer function. The quality of our approach is presented in comparison with the other algorithms.

Keywords: subsurface scattering, subjective metrics, 3D models, computer graphics.

1 Introduction

Currently available computer games offer much better quality graphics, including effects that make the illusion of reality. One of these effects may be seen in the real world for semi-translucent objects. For many real world materials such as human skin, wax or ice, light scatters within the object volume (see Fig. 1). Simulation of such effects in computer games requires very high-speed image generation. In the most widely used lighting model simplification, the reflection of reality is impossible. The beam of light at the point of incidence must be taken into consideration. Subsurface scattering can be modelled by the bidirectional subsurface scattering reflectance distribution function (BSSRDF) [10] that expresses the light transport in a translucent material. However, real-time rendering and visualization, where BSSRDFs are used, become complex tasks. It is an especially challenging problem in computer games where the subsurface scattering algorithm coexists with other computations which are performed for scenes consisting of hundreds of objects. The calculations associated with the

A. Campilho and M. Kamel (Eds.): ICIAR 2012, Part I, LNCS 7324, pp. 402–409, 2012.

graphics are so much time consuming that even the fastest graphics cards or processors do not provide a satisfactory run time when the algorithm is not optimal. Existing methods allow simulation of subsurface scattering in real time, but the quality of the images is not satisfactory or their generation time is too long to meet the ever-growing expectations.

Fig. 1. Subsurface scattering effect

In this paper an algorithm for subsurface scattering materials in computer games is presented. We take into consideration the crucial requirements of game environments like real time performance, low memory consumption and the low complexity of the algorithm that must be suitable for the GPU based hardware based implementation.

In the following section we overview previous work connected with rendering subsurface scattering materials. Special attention is paid to real time algorithms. The approach based on spherical harmonics is presented in Sect. 3. The presentation of the method results and performance tests is described in Sect. 4. The paper is concluded in Sect. 5.

2 Previous Work

In real-time applications, such as games or VR where action is very dynamic, there is an overwhelming problem with proper balancing between physical accuracy and interactivity. According to the visualization of subsurface scattering effect, a lot of investigations into such applications were introduced. Jensen et al. [9] introduced a dipole diffuse approximation model for multiple scattering in a surface simulation. To perform this technique in real time, in [1] the hierarchical texture atlas of surface fragments with precomputed radiosity and maps of scattered irradiance was used. However, the dipole model is not fully satisfying for complex shape objects and the composition of heterogeneous scattering materials.

Multiple-scattering in heterogeneous materials was also investigated. In [8] a diffuse equation for translucent materials was introduced. To solve such an equation Wang et al. [13] used a discretized model of the interior of a translucent object. The model is based on a manually built regular grid (poly grid). To simulate a subsurface scattering effect the diffuse equation was solved for every node. However, such a model is not suited to complex geometry, especially with thin features. The latter problem was solved in [14] by an automatically built mesh called QuadGraph. Sloan et al. [12] introduced a precomputed transfer function (PTR) based on spherical harmonics (SH) to produce light transport in a diffuse environment approximation. The results of the approach were implemented in the commercial game Halo3 [5], where traditional light maps were exchanged with incident radiance computed by SH for every point on a surface. The approach produced high quality however it needed to be sped up. The problem of balance between speed and quality of subsurface scattering rendering was discussed in [11]. Normal vector filtering uses information about incident light at neighboring surface points, or the spherical harmonics. The original normals are replaced with new ones producing light scattering in the object. Because of its simplicity the algorithm was very fast but suffered from poor quality.

The other group of algorithms for subsurface scattering approximation utilizes shadow maps. A traditional shadow map was used in RenderMan [7] in an offline context. An extension of the idea for multiple scattering in real time was proposed by Dachsbacher et al. in [2], where filtering of the shadow map depends on pixel depth and its normal value. In [3] estimating the distance covered by light under the surface of the material is based on a shadow casting algorithm. This technique is used practically in real time rendering. However, it suffers from artifacts, especially visible in the form of sharp changes of light intensity between different models parts. The approach was improved by blurring the depth map giving better results, but it still differs from a real effect.

In the subsequent section we present a real-time approach combining the advantages of the presented algorithms, which makes it suitable for computer games with a good quality visual effect.

3 Spherical Harmonics Subsurface Scattering

The problem between speeding up and quality in computer games and the VR is still ongoing. Therefore, we present a very simple algorithm with good quality based on the spherical harmonics. Our technique is based on the concept used in Green algorithm [3]. To thoroughly reproduce the effect in all visible points from the object in every frame, the technique uses depth maps for object thickness calculation. This causes all edges of the object to be clearly visible. In comparison with the objects in the real world it is improper, as the light in a real object is not only attenuated, but also scattered. Therefore all details and sharp edges should be blurred, and should depend on the object thickness and material characteristics.

In presented approach (SHSS) the object thickness at each vertex model in every possible direction is encoded by SH coefficients. The algorithm is divided into two main phases: preprocessing and visualization. First, the object thickness is encoded by SH coefficients, then the factors are used to reproduce the approximate thickness of the object. (see Fig. 2). The SH coefficients describe a level of similarity of the approximated function to the source function created as the spherical depth map with a ray-tracing algorithm. Next, by processing every vertex separately with a GPU, the selected part with a similar sphere depth map according to the light source direction is defined. Therefore, instead of computing the depth by tracing a ray from the vertex, the reconstructed spherical depth map computed for the pixel is used in a Phong model.

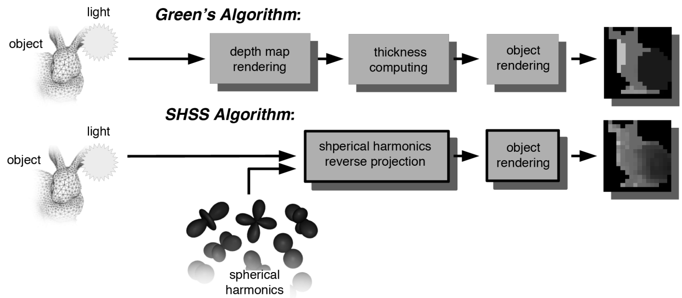

Fig. 2. Comparison of algorithms: Green [3] and SHSS

The first phase is very important as it makes all the calculations that need to be made just the once, to be immediately transmitted to the next stage, or saved in a file together with the 3D model for later use. A preprocessing step consists of three substeps: 1) sampling the distance from the viewpoint to each node of the 3D model, 2) calculating the SH values for full field angles in the spherical system and 3) SH projecting in order to evaluate the similarity between the distance and the spherical band harmonics.

The spherical harmonic coefficients based on SH functions [4] are computed according to Equation 1 respectively. To ensure the simplicity of the calculations for rigid object transformations, instead of the SH coefficients, the light is converted to the local system.

In the next step the proper values of SH are chosen. It enables the elimination of unnecessary large amounts of data in the graphics memory. Therefore, only the function values for the current object position are sent to memory, per frame. The visualization task is to render the subsurface scattering for semi-translucent objects with recalculated data in real time. To restore object thickness for light scattering in the light equation, SH projection is used (Equation 2). Finally, the colour of the object is computed based on the light model (Equation 3).

$$c_l^m = \int_S f(s)\, y_l^m(s)\, ds \tag{1}$$

where: c - SH coefficient, l - band number, $m \in < -l, l >$, f - original function, y - spherical harmonics function, s - sphere sample.

$$\tilde{f}(s) = \sum_{l=0}^{n-1} \sum_{m=-l}^{l} c_l^m y_l^m (s) = \sum_{i=0}^{n^2} c_i y_i (s) \quad (2)$$

where: $\tilde{f}$ - thickness function.

$$I = \overbrace{I_a \cdot k_{am}}^{ambient} + \underbrace{I_p \cdot k_{di} \cdot (\boldsymbol{N} \circ \boldsymbol{L})}_{diffuse} + \overbrace{I_p \cdot k_{sp} \cdot cos^n \alpha}^{specular} + \underbrace{I_p \cdot k_{sc} \cdot (1-d)}_{scatter} \quad (3)$$

where: I - the total outgoing intensity, I_a - intensity of ambient light, I_p - the intensity of light scatter, k_{am} - reflectance of ambient light, k_{di} - coefficient of diffuse reflection, k_{sp} - the directional reflectance factor, k_{sc} - damping coefficient of light in the building, d - thickness of the object toward the light, N - normal vector surface, L - vector toward the light, α - the angle between the direction of the calculated reflectance and the direction of specular reflection, n - coefficient of surface roughness.

4 Results

The effect was implemented in C++. For graphics implementation, OpenGL and GLSL languages were used. GPU: Nvidia, GeForce 8400 GS. In order to reduce the data exchange between RAM and the GPU memory, each model node and SH coefficients are stored in VBO in video memory. This removes the need for each frame of the newly submitted data. SH function value, depending on the light direction, is calculated on the CPU and sent to the GPU per frame.

The main focus of the tests is the quality of images rendered in real-time. For testing, three models: Athena, Armadillo and Dragon ([16]) were used (see Fig. 3). We chose complex models, equivalent to models in modern computer games. To compare the quality of the SHSS algorithm, we conducted experiments based on a pairwise comparison experiment. In order to perform the experiment for every algorithm, graphic models illuminated from three different sites (back, front and left) were prepared. The brightness level of resulting images was normalized to eliminate any influence on the outcome. In the experiment four methods are compared: SHSS, Patro [11], Green [3] and PRT [12].

The algorithms were assessed by computer graphics laymen who had normal or corrected to normal vision. The age varied between 22 and 65. There were 4 male and 6 female observers. To estimate inter-observer variability the subjects repeated each session three times, but neither of the repetitions took place on the same day. The observers were free to adjust the viewing distance to their preference. The illumination in the room was subdued by blackout curtains to minimize the effect of display glare. Images were shown on 50%-gray background. The observers were asked to read a written instruction before every experiment and choose the most realistic image in all displayed pairs . According to [15]

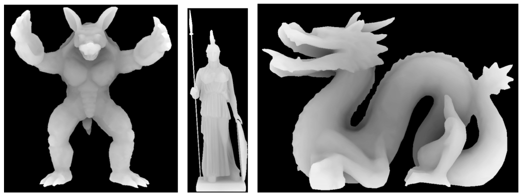

Fig. 3. Left: Athena model (18935 triangles, 37651 vertices). Middle: Armadillo model (129720 vertices, 345944 triangles). Right: Dragon model: (566,098 vertices, 1132830 triangles).

SHSS	PRT	Green	Patro
score = 3.36	score = 2.45	score = 2.25	score = 1.93

Fig. 4. Experimental results. Algorithms ordered from the highest to the lowest score. Armadillo model is lit from left to right.

recommendation, the experiment started with a training session in which observers familiarized themselves with the task and interface. After that session, they asked questions or started the main experiment. To ensure that the observers fully attended the experiment, three random trials were shown at the beginning of the main session without recording the results. The images were displayed in a random order and with a different randomization for each session. Two consecutive trials showing the same scene were avoided if possible.

The measure of the experiment is computed as the number of votes (the number of times one algorithm is preferred to another) assuming that all pairs of algorithms are compared. The result is averaged across the observers and models for every algorithm. The vote count is also equivalent to the position in the ranking. Figure 4 depicts the results. For consistency, we ordered the algorithms according to the average ranking, from the highest to the lowest score.

Figure 5(Top) allows us to analyze the results of the experiment. According to ANOVA, there is a significant difference between the PRT, SHSS and the rest of the algorithms. The SHSS and PRT were assessed in a similar way without a significant difference. SHSS was slightly better. It is worth noticing that the SHSS algorithm is much simpler than the PRT. Although the Green algorithm simulates the absorption of light, it is characterized by sharp changes between brightness levels. Therefore, it is very often evaluated in last position. The Parto

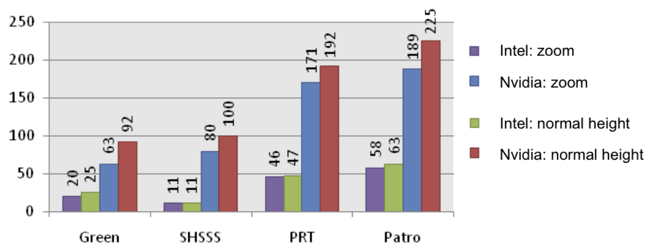

Fig. 5. Top: ANOVA results. Algorithms PRT and SHSS are comparable, with slightly preference to SHSS. SHSS is significantly different from Patro and Green. Bottom: Efficiency comparison for *budda* model.

technique was assessed to be worse than SHSS, because the light distribution is characterized by unnatural shadows visible in the images.

We asked the subjects what guided them during the assessment. The participants paid attention to a variety of factors. Mainly, attention was paid to the sharpness of images, visible degree of details, visible contours, shadows, light distribution inside the objects and the intensity of light piercing. To demonstrate the SHSS algorithm efficiency, the measurements were made for different camera settings for every model and for all the SH levels. Then the results were averaged out. For the evaluation of the computational time, 20 measurements in total were performed. For all the models a frame size was fixed to 1280 x 800 pixels and we received average 160 FPS for a model. Efficiency comparison between algorithms is depicted in Figure 5(Bottom). The SHSS algorithm seems to be the most suitable for real-time applications.

5 Conclusions and Future Work

In this paper, we presented an SHSS algorithm for real-time rendering of subsurface scattering in materials. Instead of using a shadow casting algorithm to calculate the thickness of the object, spherical harmonics functions were used to convert thickness for each vertex model. Therefore it works correctly with a concave object, which was improperly rendered in the Green algorithm. The SHSS approach is simple and uses no complicated equations. The cost of computation with multiple lights depends linearly on their quantities.

It was determined that very high-quality rendering effect by light behind the object could be obtained by using 25 coefficients while the light in front of the object needs only 4 coefficients for one vertex. This makes the algorithm very fast, allowing the application to be used in games and VR. The method produces good quality images in real-time, comparable or better than other known algorithms.

Acknowledgements. This work was supported by the Polish Ministry of Science and Higher Education through the grant no. N N516 193537.

References

1. Carr, N.A., Hall, J.D., Hart, J.C.: GPU algorithms for radiosity and subsurface scattering. In: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware (HWWS 2003), San Diego, California, USA, pp. 51–59 (2003)
2. Dachsbacher, C., Stamminger, M.: Translucent shadow maps. In: EGRW 2003: Proceedings of the 14th Eurographics Workshop on Rendering, pp. 197–201 (2003)
3. Green, S.: Real-Time Approximations to Subsurface Scattering. In: GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics, pp. 263–278. Addison-Wesley (2004)
4. Green, R.: Spherical harmonic lighting: The gritty details (2003)
5. Chen, H., Wallace, B., Tchou, C., Cook, D., Wang, X., Hu, Y.: Lighting and material of Halo 3. In: ACM SIGGRAPH 2008 Classes, pp. 1–22 (2008)
6. Hasan, M., Fuchs, M., Matusik, W., Pfister, H., Rusinkiewicz, S.: Physical Reproduction of Materials with Specified Subsurface Scattering. ACM Transactions on Graphics (SIGGRAPH 2010) 29(3) (2010)
7. Hery, C.: Implementing a skin bssrdf. In: SIGGRAPH Courses, pp. 73–88 (2003)
8. Ishimura, A.: Wave Propagation and Scattering in Random Media, vol. 1(2). Academic Press (1978)
9. Jensen, H.W., Marschner, S.R., Levoy, M., Hanrahan, P.: A practical model for subsurface light transport. ACM Transactions on Graphics (SIGGRAPH 2001), 511–518 (2001)
10. Nicodemus, F.E., Richmond, J.C., Hsia, J.J., Ginsberg, I.W., Limperis, T.: Geometrical Considerations and Nomenclature for Reflectance. National Bureau of Standards, US (1977)
11. Patro, R.: Real-Time Approximate Subsurface Scattering. In: PG 2007: Proceedings of the Pacific Conference on Computer Graphics and Application, pp. 403–406 (2007)
12. Sloan, P.P., Hall, J., Hart, J., Snyder, J.: Clustered principal components for precomputed radiance transfer. ACM Transactions on Graphics 22, 382–391 (2003)
13. Wang, J., Zhao, S., Tong, X., Lin, S., Lin, Z., Dong, Y., Guo, B., Shum, H.-Y.: Modeling and rendering of heterogeneous translucent materials using the diffusion equation. ACM Trans. Graph (SIGGRAPH 2008) 27(1), 1–18 (2008)
14. Wang, Y., Wang, J., Holzschuch, N., Subr, K., Yong, J.-H., Guo, B.: Real-time Rendering of Heterogeneous Translucent Objects with Arbitrary Shapes. Computer Graphics Forum (Eurographics 2010) 29(2) (2010)
15. ITU-R.Rec.BT.500-11, Metodology for the Subjective Assessment of the Quality for Television Pictures (2002)
16. The stanford 3D Scanning Repository, `http://graphics.stanford.edu/data/3Dscanrep/`