HIGHLIGHT-TRACK SHADING

Yueh-Yi Lai

Wen-Kai Tai
Assistant Professor

Department of Computer Science and Information Engineering
National Dong-Hwa University, Hualien, Taiwan, R.O.C.
Email: wkdai@mail.ndhu.edu.tw;

Keywords: shading, rendering, rasterization.

Abstract

In Computer Graphics, Phong shading is essential and sufficient for producing realistic images, but it suffers from the computation cost. In this paper, we propose a highlight-track shading approach to shade a polygon along the polygon edges and the highlight-track. This approach has the efficiency of Gouraud shading and the effectiveness of Phong shading.

1. Introduction

For years, high quality and real time computer generated images have been required. In spite of the increasing computation power, we could not have real time realistic images for complex scenes. So far, most computer graphics and virtual reality applications [4] are still applying Gouraud shading [5]. Intensity interpolation is very simple and efficient, and also appropriate for hardware implementation. But it comes with some drawbacks, such as missing highlights for specular surfaces. To provide better visual effects, the Phong shading [8] is applied. Instead of intensity interpolation, it takes more computation costs for normal interpolation. Therefore, an intermediate approach providing a solution with the efficiency of Gouraud shading and the effectiveness of Phong shading is significant. In this paper, we propose a shading method that takes the some computation time as Gouraud and presents the visual effects as Phong's.

Duff [3] developed a method by combining the interpolation and reflection equation and using forward differences to reduce the computation per pixel to three additions, one division, and one square root. Bass [1] encoded the surface normal and looked up the intensity of reflectivity. Graphical objects can be quickly displayed. However, quantizing the surface normals produces artifacts of square paths in the rendered image. Bishop[2] provided a more excellent approximation of Phong shading by using the Taylor series expansion. Bishop's approach for Phong shading reduced the amount of computation per pixel to only two additions for the diffuse component, and five additions and one memory access for the whole Phong reflection model. The Highlight Shading [9] recursively subdivides the triangle till the differences between the Normal Shading and the Color Shading color values are small enough for all the vertices of a subtriangle. Then, the triangle is filled using the color shading method. Obviously, it takes much time for subdivision. Overveld [6],

[7] suggested to use the infinite-distance model instead of the expensive finite-distance model for the computation of highlights by replacing a light source at a finite distance by a light source at an infinite distance, and at the same time adjusting the normal vectors in such a way that the resulting illumination pattern stays the same. However it costs too much overhead for adjustments.

In order to have smooth and correct shading effects, it is necessary to use accurate normal vectors for illumination. In Phong shading, it takes about 60% of the entire shading time on normal vectors computation. Eventually, only parts of normal vectors are not supposed to be negligent. These normal vectors are highlight related. We hereby propose a method to catch the highlight-track for a triangle. All normal vectors of pixels on this track are calculated accurately. Then, for in-between pixels in the same scan-line the intensity interpolation is applied.

In the following, we present the basic idea of highlight-track shading in section 2. We demonstrate the experimental results and compare the performance with other shading methods in section 3. Finally, we discuss the proposed method and future works for further study.

2. Highlight-Track Shading

To shade a triangle, $\boldsymbol{T_{ABC}}$, we determine the highlight-track of $\boldsymbol{T_{ABC}}$ and then apply Fence shading to the highlight-track and edges of $\boldsymbol{T_{ABC}}$. Let $\boldsymbol{N_A}$, $\boldsymbol{N_B}$, and $\boldsymbol{N_C}$ be normal vectors in corresponding to vertices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ of triangle $\boldsymbol{T_{ABC}}$ respectively, and $\boldsymbol{H}$ be a halfway vector of the underlying environment. Regarding $\overline{AB}$ as the bottom edge of $\boldsymbol{T_{ABC}}$, there exists a point, N_H, such that $\overline{HN_H}$ is the shortest distance from $\boldsymbol{H}$ to line $\overleftrightarrow{N_A N_B}$. That is to say that calculating the specular term for each point in the line $\overleftrightarrow{N_A N_B}$, $\boldsymbol{H} \cdot \boldsymbol{N_H}$ is with the maximal value, as shown in Figure 1. We know that every scan-line is parallel with each other. It implies that the pixel with maximal specular value corresponding to every scan-line is certainly located in this line $\overleftrightarrow{HN_H}$. Therefore, the intersection line segment, $\overline{H_{AC} H_{BC}}$, between line $\overleftrightarrow{HN_H}$ and triangle $\boldsymbol{T_{ABC}}$ is the highlight-track inside triangle $\boldsymbol{T_{ABC}}$ (see Figure 2). When triangle $\boldsymbol{T_{ABC}}$ is to be rendered, Fence shading is not only applied to the edges of $\boldsymbol{T_{ABC}}$ but also to the highlight-track. Therefore, our method catches all highlights that occur inside polygons to be rendered and has little expense for calculating the highlight-track.

We present the efficiency and effectiveness of the highlight-track shading method by comparing with Gouraud, Fence, Fast Phong, and Phong shading in the following section.

3. Experiments Results

To demonstrate the effectiveness and efficiency of the proposed approach, we compare results with Gouraud, Phong, Fast Phong and Fence shading. Objects in the scene are modeled in a way that highlights occur inside polygons and rendered on a PC with Pentium !!! 500MHz and 256MB SDRAM.

As shown in Figure 3, Figure 4, Figure 5, Figure 6, and Figure 7, this scene consisting of one teapot, four cups, and one tea tray is with 1536 polygons. In this coarse resolution model, we intend to point out that the highlight effects in some polygons are not properly rendered by Gouraud (see Figure 3) and Fence shading (see

Figure 4).

In Table 1, we know that the proposed highlight-track shading method eventually takes a little more computation time than Fence and Gouraud shading. It is merely slower about 1.0615 times of Gouraud shading. However, it drastically reduces the computation cost for demonstrating Phong's visual effects. It speeds up about 2.5 times than Phong shading. Although, Fast Phong shading, as shown in Figure 5, also presents the visual effects of highlight, the computation overhead for coefficients in Taylor series costs a lot. Consequently, the proposed highlight-track shading method is as efficient as Gouraud shading and effective as Phong shading.

4. Conclusions

In this paper, we propose an approach for highlight tracking. Computer generated images with visual effects of Phong shading now take a little more computation than Gouraud shading. On the demand of real time rendering system with Phong shading, our method is essential and suitable.

To extend our approach, we will consider environments with positional light sources and the finite distance viewpoint, the light vector ***L*** and halfway vector ***H*** are at finite distance, and provide more smoothness intensity propagation along scan-lines instead of linear interpolation. This is so we can present even more impressive rendered images.

5. Reference

[1] Bass, Daniel H., "*Using the Video Lookup Table for Reflectivity Calculations: Specific Techniques and Graphic Results*", Computer Graphics and Image Processing, No. 17, 1981, pp. 249-261.

[2] Bishop, Gary and Wiemer, David M., "*Fast Phong Shading*", SIGGRAPH 86, pp. 103-106.

[3] Duff, T., "*Smoothly Shaded Renderings of Polyhedral Objects on Raster Displays*", ACM Computer Graphics, Vol. 13, No. 2, 1979, pp. 270-275.

[4] Foley, J. D., vanDam, A., Feiner, S., and Hughes, J., *Computer Graphics: Principles and Practices*, second edition, Addison-Wesley, 1993.

[5] Gouraud, H., "*Continuous Shading of Curved Surfaces*", IEEE Trans-actions on Computers, Vol. 20, No. 6, 1997, pp. 623-628.

[6] Overveld, van C.W.A.C.M, "*Phong Normal Interpolation Revisited*", ACM Transactions on Graphics, Vol. 16, No. 4, October 1997, pp. 397-419.

[7] Overveld, van C.W.A.M., and Wyvill, Brian, "*Hi-speed, Hi-fi, Hi-lights: A Fast Algorithm for the Specular Term in the Phong Illumination Model*", Journal of Graphics Tools, Vol. 1, No. 2, 1996, pp. 25-30.

[8] Phong, B. T., "*Illumination for Computer Generated Pictures*", Communications of the ACM, Vol. 18, No. 60, 1975, pp. 311-317.

[9] Popsel, J. and Hornung, Ch. "*Highlight Shading: Lighting and Shading in a PHIGS+ /PEX- Environment*", Computer and Graphics, Vol. 14, No. 1, 1990, pp. 55-64.

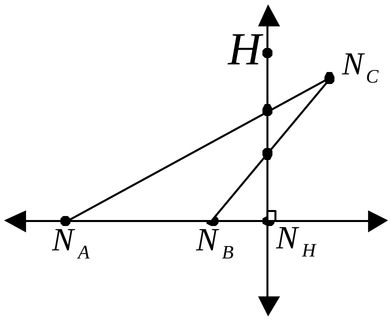

Figure 1. The specular term for each point in line $\overleftrightarrow{N_A N_B}$, $H\ ?N_H$ is with the maximal value.

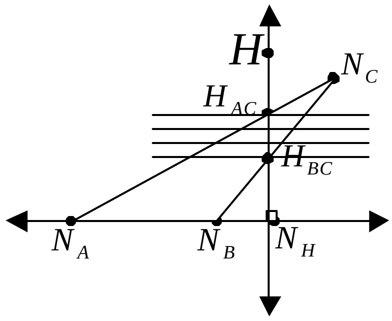

Figure 2. $\overline{H_{AC} H_{BC}}$ is the highlight-track inside triangle T_{ABC}.

	CPU Time(sec.)	Gouraud Ratio	Phong Ratio
Gouraud	2.1310	1.0000	0.3680
Phong	5.7908	2.7174	1.0000
Fast Phong	2.8804	1.3517	0.4974
Highlight-track	2.2620	1.0615	0.3906
Fence	2.1410	1.0047	0.3697

Table 1. The rendering time of Gouraud, Phong, Fast Phong, Fence, and Highlight-track shadings for the scene, 1536 polygons, consisting of one teapot, four cups, and one tea tray.

Figure 3. Gouraud shading

Figure 4. Fence shading

Figure 5. Fast Phong shading

Figure 6. Phong shading

Figure 7. Highlight-track shading