Direct Ray Tracing of Phong Tessellation

Shinji Ogaki*
Square Enix Co., Ltd.

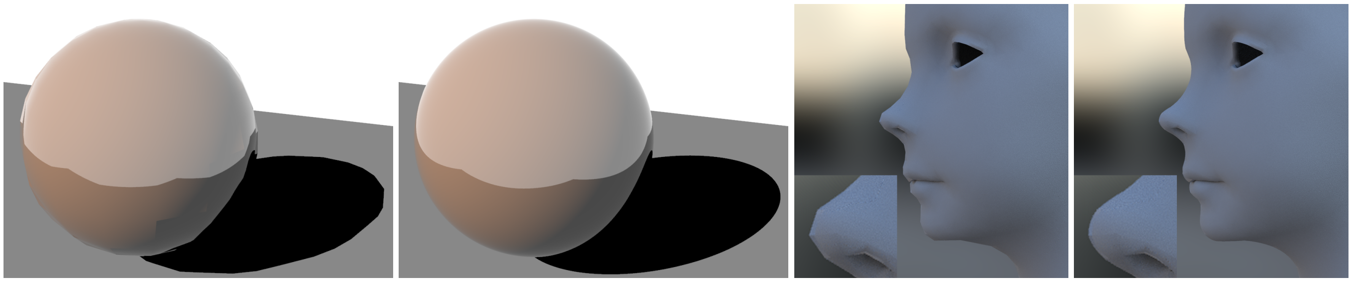

Figure 1: *From left to right: a low polygon sphere (1024x720 pixels, 3 seconds); a low polygon sphere rendered with our method (1024x720 pixels, 5 seconds); a low polygon head model (800x800 pixels, 8 seconds); a low polygon head model rendered with our method (800x800 pixels, 11 seconds). Note that the silhouettes and shadow boundaries are smooth in our method.*

Abstract

There are two major ways of calculating ray and parametric surface intersections in rendering. The first is through the use of micropolygons, and the second is to use parametric surfaces such as NURBS surface together with numerical methods such as Newton Raphson. Both methods are computationally expensive and complicated to implement. In this paper, we introduce a direct ray tracing method for Phong Tessellation. Our method gives analytic solutions that can be readily derived by hand and enables rendering smooth surfaces in a computationally inexpensive yet robust way.

1 Introduction

There are mainly two ways of handling parametric surfaces. One approach is to perform intersection tests with tessellated polygons. The other method is to directly calculate intersect points.

Many efficient ray tracing methods for tessellated polygons have been proposed, such as [Smits et al. 2000]. State-of-the-art methods can handle motion blur of micropolygons (see for example [Hou et al. 2010]). To achieve high performance, tessellated micropolygons must be cached and rays have to be traced in a coherent way so that cache performance is maximized. Additionally, in order to select the appropriate tessellation level, vital for achieving a high quality result, ray differentials [Igehy 1999] must be used.

Direct ray tracing of NURBS surface has been intensively studied. There exist a number of effective methods such as [Abert et al. 2006]. However, they are computationally expensive since normally they depend on numerical methods including Newton Raphson. Furthermore, additional memory is required to store initial values. Besides NURBS surface, there are a variety of parametric surfaces: Phong Tessellation [Boubekeur and Alexa 2008], and T-Spline surface [Sederberg et al. 2003] to name a few. Phong Tessellation has a particularly simple form and produces good results.

In this paper, we focus on Phong Tessellation and present a direct ray tracing method for it. Our method gives analytic solutions by using a pencil of curves. This enables us to render smooth surfaces in a simple and robust way.

*e-mail: ogaki@square-enix.com

2 Phong Tessellation

Firstly, we briefly review Phong Tessellation. Let (u, v, w) be a barycentric coordinate and $\boldsymbol{p_i}$ a vertex. Phong Tessellation is given as

$$p_\alpha(u, v) = (1-\alpha)p(u, v) + \alpha p^*(u, v), \tag{1}$$

where

$$\begin{aligned} p(u,v) &= (u,v,w)(\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_3})^T \\ p^*(u,v) &= (u,v,w)\begin{pmatrix} \pi_1(p(u,v)) \\ \pi_2(p(u,v)) \\ \pi_3(p(u,v)) \end{pmatrix}, \end{aligned}$$

and $w = 1-(u+v)$. Letting $\boldsymbol{n_i}$ be a vertex normal vector, the function π_i is given as

$$\pi_i(\boldsymbol{q}) = \boldsymbol{q} - ((\boldsymbol{q}-\boldsymbol{p_i})^T\boldsymbol{n_i})\boldsymbol{n_i}.$$

The factor α is used for controlling the shape. See [Boubekeur and Alexa 2008] for more detail .

3 Intersection Test

Since Phong Tessellation defines a quadratic surface, the intersection test can be rearranged into the intersection problem of two quadratic curves as in [Kajiya 1982]. By using a pencil, the intersections of the two curves $F(u, v)$ and $G(u, v)$ can be easily obtained as follows (see more detail, for example, [Hosaka 1992]).

The general form of a quadratic curve is given as

$$\begin{aligned} F(u,v) &= au^2 + bv^2 + c + 2duv + 2eu + 2fv \\ &= (u,v,1)\begin{pmatrix} a & d & e \\ d & b & f \\ e & f & c \end{pmatrix}(u,v,1)^T \\ &= 0. \end{aligned} \tag{2}$$

With this notation, the pencil P is defined as

$$\begin{aligned} P &= xF(u,v) + G(u,v) \\ &= x(u,v,1)M(u,v,1)^T + (u,v,1)N(u,v,1)^T, \end{aligned}$$

where

$$M = \begin{pmatrix} a & d & e \\ d & b & f \\ e & f & c \end{pmatrix}, N = \begin{pmatrix} l & o & p \\ o & m & q \\ p & q & n \end{pmatrix}.$$

If $|xM + N| = 0$, P can be represented as a product of two lines. This is a special case of a hyperbola. (An intuitive explanation is that $u^2/s^2 = v^2/t^2$ can be expressed as $(tu + sv)(tu - sv) = 0$.)

By letting

$$S = xM + N = \begin{pmatrix} A & D & E \\ D & B & F \\ E & F & C \end{pmatrix}$$

and

$$\begin{aligned} L_1 &= \alpha_1 u + \beta_1 v + \gamma_1 && (3) \\ L_2 &= \alpha_2 u + \beta_2 v + \gamma_2, && (4) \end{aligned}$$

we have

$$Au^2 + Bv^2 + C + 2Duv + 2Eu + 2Fv = L_1 L_2. \quad (5)$$

The factorization is done by comparing the coefficients of u^2, v^2, uv, u, v, and 1. (Dividing the both sides of Equation (5) by A or B, whichever is larger, makes the calculation easier.) Let $\{\theta \cap \phi\}$ denote the set of the intersections of θ and ϕ. Clearly $\{F(u,v) \cap G(u,v)\} = \{F(u,v) \cap P\}$. Hence, the set of the intersections is decomposed as $\{F(u,v) \cap G(u,v)\} = \{F(u,v) \cap L_1\} + \{F(u,v) \cap L_2\}$.

Thus, the problem can be simplified by finding the value of x such that $|S| = 0$. Substituting Equation (3) and (4) into Equation (2) gives two quadratic equations. By solving them, we obtain u's and v's. The parameters u, v, and $w = 1 - (u + v)$ must lie between 0 and 1. Otherwise there is no intersection. Note also that the intersection test is terminated if $0 < AB - DD$ since $0 = (u, v, 1)S(u, v, 1)^T$ becomes an ellipsoid whose area is zero.

Finally, we are ready to solve the ray-Phong Tessellation intersection problem. Every ray can be represented as an intersection of two planes, say $Plane_1(x, y, z) = \boldsymbol{N_1}(x, y, z)^T + D_1$ and $Plane_2(x, y, z) = \boldsymbol{N_2}(x, y, z)^T + D_2$. The intersection $\boldsymbol{p}$ lies on the both planes. Therefore we have

$$\begin{aligned} 0 &= \boldsymbol{N_1} \cdot \boldsymbol{p} + D_1 && (6) \\ 0 &= \boldsymbol{N_2} \cdot \boldsymbol{p} + D_2. && (7) \end{aligned}$$

It is obvious that substituting the right-hand side of Equation (1) for $\boldsymbol{p}$ of Equation (6) and (7) gives two quadratic curves. The method described above gives solutions.

4 Shading Normal

For smooth rendering, we use the Phong-interpolated normal $\boldsymbol{p_N}$ which is given as

$$\boldsymbol{p_N} = (u\boldsymbol{n_0} + v\boldsymbol{n_1} + w\boldsymbol{n_2})/|u\boldsymbol{n_0} + v\boldsymbol{n_1} + w\boldsymbol{n_2}|.$$

In general, using $\boldsymbol{p_N}$ produces good results. However, this leads to unpleasant artifacts at grazing angles since $\boldsymbol{p_N}$ is not necessarily equivalent to the surface normal of Phong Tessellation $\boldsymbol{s_N}$. In order to avoid this, we use $\boldsymbol{s_N}$ instead, if the inner product of $\boldsymbol{s_N}$ and the reflection vector $\boldsymbol{r}$ is negative. Letting $\boldsymbol{v}$ be the viewing direction, $\boldsymbol{r}$ is given as

$$\boldsymbol{r} = \boldsymbol{v} - 2\boldsymbol{p_N}(\boldsymbol{v} \cdot \boldsymbol{p_N}).$$

The surface normal $\boldsymbol{s_N}$can be derived as

$$\boldsymbol{s_N} = \left(\frac{\partial p_\alpha}{\partial u} \times \frac{\partial p_\alpha}{\partial v}\right) \bigg/ \left|\frac{\partial p_\alpha}{\partial u} \times \frac{\partial p_\alpha}{\partial v}\right|.$$

The partial derivatives $\frac{\partial p_\alpha}{\partial u}$ and $\frac{\partial p_\alpha}{\partial v}$ are given as

$$\begin{aligned} \frac{\partial p_\alpha}{\partial u} &= \alpha((2\boldsymbol{p_1} - \boldsymbol{c_3})u + (\boldsymbol{c_1} - \boldsymbol{c_2})v + (\boldsymbol{c_3} - 2\boldsymbol{p_3})w) \\ &+ (1 - \alpha)(\boldsymbol{p_1} - \boldsymbol{p_3}) \\ \frac{\partial p_\alpha}{\partial v} &= \alpha((2\boldsymbol{p_2} - \boldsymbol{c_2})v + (\boldsymbol{c_1} - \boldsymbol{c_3})u + (\boldsymbol{c_2} - 2\boldsymbol{p_3})w) \\ &+ (1 - \alpha)(\boldsymbol{p_2} - \boldsymbol{p_3}), \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{c_1} &= \boldsymbol{p_2} - \boldsymbol{n_1}(\boldsymbol{n_1} \cdot (\boldsymbol{p_2} - \boldsymbol{p_1})) + \boldsymbol{p_1} - \boldsymbol{n_2}(\boldsymbol{n_2} \cdot (\boldsymbol{p_1} - \boldsymbol{p_2})) \\ \boldsymbol{c_2} &= \boldsymbol{p_3} - \boldsymbol{n_2}(\boldsymbol{n_2} \cdot (\boldsymbol{p_3} - \boldsymbol{p_2})) + \boldsymbol{p_2} - \boldsymbol{n_3}(\boldsymbol{n_3} \cdot (\boldsymbol{p_2} - \boldsymbol{p_3})) \\ \boldsymbol{c_3} &= \boldsymbol{p_1} - \boldsymbol{n_3}(\boldsymbol{n_3} \cdot (\boldsymbol{p_1} - \boldsymbol{p_3})) + \boldsymbol{p_3} - \boldsymbol{n_1}(\boldsymbol{n_1} \cdot (\boldsymbol{p_3} - \boldsymbol{p_1})). \end{aligned}$$

5 Conclusion and Future Work

In this paper, we introduced a direct ray tracing method for Phong Tessellation. With this method we are able to render high quality images from low polygon models (see Figure 1) while keeping the renderer architecture unchanged. In our implementation one of the main bottlenecks is the kd-tree construction. We would like to develop an algorithm that accelerates this process. In addition, the comparison with Newton Raphson has to be done to investigate the numerical accuracy of our method.

References

ABERT, O., GEIMER, M., AND MÜLLER, S. 2006. Direct and fast ray tracing of NURBS surfaces. In *Proc. of the IEEE Symposium on Interactive Ray Tracing*, 161–168.

BOUBEKEUR, T., AND ALEXA, M. 2008. Phong tessellation. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers*, ACM, New York, NY, USA, 1–5.

HOSAKA, M. 1992. *Modeling of curves and surfaces in CAD/CAM*. Springer-Verlag New York, Inc., New York, NY, USA.

HOU, Q., QIN, H., LI, W., GUO, B., AND ZHOU, K. 2010. Micropolygon ray tracing with defocus and motion blur. In *SIGGRAPH '10: ACM SIGGRAPH 2010 papers*, ACM, New York, NY, USA, 1–10.

IGEHY, H. 1999. Tracing ray differentials. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 179–186.

KAJIYA, J. T. 1982. Ray tracing parametric patches. In *SIGGRAPH '82: Proceedings of the 9th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 245–254.

SEDERBERG, T. W., ZHENG, J., BAKENOV, A., AND NASRI, A. 2003. T-splines and t-nurccs. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, ACM, New York, NY, USA, 477–484.

SMITS, B., SHIRLEY, P., AND STARK, M. M. 2000. Direct ray tracing of smoothed and displacement mapped triangles. Tech. rep.