Real-time Tessellation on GPU

for Finite-Element Methods

Olivier Cotte
McGill University
Department of Electrical and Computer Engineering

Contact Information:
Computational Electromagnetic Lab
Electrical and Computer engineering
McGill University
3480 University Street, Montreal, Quebec
Lab Room 620
Phone: +1 (514) 398-7146
Email: olivier.moussavoucotte@mail.mcgill.ca

Abstract

Modern 3D engines used in real-time applications provide shading to hide the lack of continuity inside surfaces. By using, for example, modulated normals ("*bump mapping*", "*normal mapping*"), textures, "*displacement mapping*", we can simulate smooth surfaces. However, artefacts remain on the outlines and silhouettes. Much of the smoothing of the surface is already produced using shading. Thus, the problem still to be solved is to improve the geometry where the smoothing can not be generated by the shading.

Introduction

1. Produce smooth geometry along the contours so that visual artefacts disappear.
2. Create this geometry with as little as possible operations.
3. The bus between the CPU and GPU is a limitation.
4. Two methods: Phong Tessellation and Curved PN Triangles.

Phong Tessellation

Proposed by Tamy Boubekeur and Marc Alexa at SIGGRAPH Asian 2008 Proceeding. Instead of interpolating the normals as in "Phong Shading", we interpolate the projection of the vertices of a triangle on their tangent plane to define a smooth geometry. This method is an approximation, so simple and effective. Combines well with *Phong shading*. In most cases, the visual results are similar to algorithms using more precise approximations. A quadratic approximation is used for the calculation of the surface, so no points of inflection. Linear approximation for normals. Too many subdivisions can lead to discontinuities.

$$\pi_i(\mathbf{q}) = \mathbf{q} - ((\mathbf{q} - \mathbf{p}_i) \cdot \mathbf{n}_i)\mathbf{n}_i \tag{1}$$

$$\begin{aligned}\mathbf{p}^*(\mathbf{u},\mathbf{v}) = {} & u^2\mathbf{p}_i + v^2\mathbf{p}_j + w^2\mathbf{p}_k + uv(\pi_i(\mathbf{p}_j)) + \pi_j(\mathbf{p}_i) \\ & + vw(\pi_j(\mathbf{p}_k) + \pi_k(\mathbf{p}_j)) + wu(\pi_k(\mathbf{p}_i) + \pi_i(\mathbf{p}_k))\end{aligned} \tag{2}$$

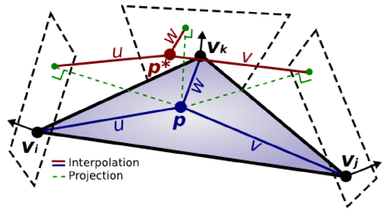

Figure 1: From Boubekeur, T. and Alexa, M. 2008. Phong Tessellation. Proceedings of SIGGRAPH Asia 2008.

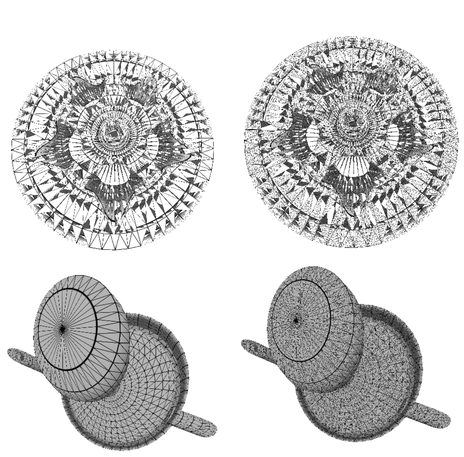

Figure 2: Normal tessellation (left), Phong Tessellation with increased level of details (right)

PN Curved Triangles

We replace the geometry of the triangle by a cubic Bezier patch and quadratically interpolates the normal on each new vertex for shading. Unlike Phong, we use a better approximation (cubic Bezier patch + quadratic interpolation of normals).

1. Better visual quality (smoother surfaces).
2. Improves the quality of the norms for shading.
3. More intensive calculations.

$$\begin{aligned}
b_{300} &= P_1 \\
b_{030} &= P_2 \\
b_{003} &= P_3 \\
b_{210} &= \frac{1}{3}(2P_1 + P_2 - w_{12}N_1) \\
b_{120} &= \frac{1}{3}(2P_2 + P_1 - w_{21}N_2) \\
b_{021} &= \frac{1}{3}(2P_2 + P_3 - w_{23}N_2) \\
b_{012} &= \frac{1}{3}(2P_3 + P_2 - w_{32}N_3) \\
b_{102} &= \frac{1}{3}(2P_3 + P_1 - w_{31}N_3) \\
b_{201} &= \frac{1}{3}(2P_1 + P_3 - w_{13}N_1) \\
b_{111} &= E + \frac{1}{2}(E - V)
\end{aligned}$$

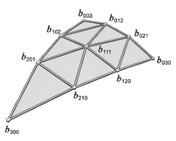

Figure 1: Control points of the PN Triangles.

with $E = \frac{1}{6}(b_{210} + b_{120} + b_{021} + b_{012} + b_{102} + b_{201})$, and $V = \frac{1}{3}(P_1 + P_2 + P_3)$

By defining $v_{ij} = 2\frac{(P_j - P_i)\cdot(N_i + N_j)}{(P_j - P_i)\cdot(P_j - P_i)}$, we have:

$$\begin{aligned}
n_{200} &= N_1 \\
n_{020} &= N_2 \\
n_{002} &= N_3 \\
n_{110} &= \frac{N_1 + N_2 - v_{12}(P_2 - P_1)}{\|N_1 + N_2 - v_{12}(P_2 - P_1)\|} \\
n_{011} &= \frac{N_2 + N_3 - v_{23}(P_3 - P_2)}{\|N_2 + N_3 - v_{23}(P_3 - P_2)\|} \\
n_{011} &= \frac{N_3 + N_1 - v_{31}(P_1 - P_3)}{\|N_3 + N_1 - v_{31}(P_1 - P_3)\|}
\end{aligned}$$

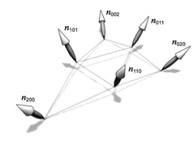

Figure 2: Normal components of the PN Triangles.

Figure 3: From Vlachos A., Peters J., Boyd C., and Mitchell J. 2001. Curved PN Triangles. Symposium on Interactive 3D Graphics 2001..

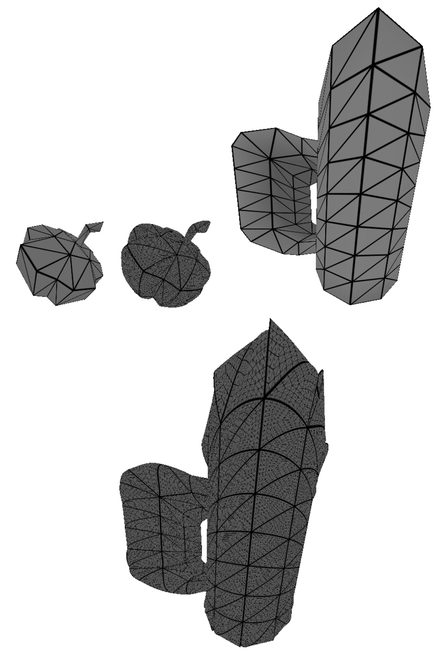

Figure 4: Normal tessellation, PN Curved Triangles Tessellation with increased level of details

Summary

The current *shading* techniques make it possible to simulate smooth surfaces. However, there are often still defects on the contours/silhouettes. Two methods have been investigated that work with any geometry automatically and they are easily implemented in tessellation units.