Bi-Quadratic Interpolation of Intensity for Fast Shading of Three Dimensional Objects

*Ekta Walia and †Chandan Singh

Abstract

Researchers in the field of Computer Graphics are often confronted with the trade off between visual realism and computational cost. So far, Phong and Gouraud shading have been treated as well established methods and attempts have been made to improve visual realism or to reduce computational cost or both. These methods use linear interpolation to compute the normals or intensity, respectively, at each point on the surface. However, it has been proved that no surface would yield prper distribution of illumination generated by the traditional Phong shading. Attempts have been made to improve the defects of linear interpolation used in Phong shading. One such attempt is the use of biquadratic normal vector interpolation. In this paper we have prpose d an algorithm to achieve the visual realism of this method and at the same time we have reduced the cost of shading.

Keywords: Phong Shading, Linear Interpolation, Quadratic Interpolation, Biquadratic Interpolation, Bezier Triangle

1. Background:

In increasing order of visual realism, there are three well-known shading methods. The simplest shd ing model for a polygon mesh is Constant Shading or Flat Shading [1]. It uses illumination model once to determine a single intensity value that is then used to shade an entire polygon. This shading method does not prduce the va riations in shade across the polygon that should actually occur for visual realism. However, in this method, no interpolation takes place for variation in shade. This is a very fast method.

Gouraud shading [2], also called intensity interpolation shading or color interpolation shding, elimi nates the intensity discontinuities across the adjacent polygons. Although a fairly fast method of shading, it suffers from Mach Band effect and fails to capture detailed lighting characteristics. All the subsequent works are a variation of such interpolative shading.

Phong Shading [3], also known as normal-vector interpolation shading, interpolates the surface normal vector rather than the interpolation of intensity. The interpolation occurs across a polygon span on a scan line, between starting and ending normals for the span. These normals are interpolated along polygon edges from the vertex normals. These interpolated normals are then used in intensity calculations. Phong shading yields substantial improvements over Gouraud shading when an illumination model with specular reflection is used. With this method, however the cost of shding is increase d since the interpolated normal are used to interpolate intensity over the surface of a polygon.

2. Existing Methods:

2.1 Attempts to improve Visual realism:

Many researchers have contributed a lot to improve the visual realism proposing various interpolation techniques.

Overveld and Wyvill [4] proposed a quadratic normal vector interpolation algorithm to replace the traditional linear interpolation. Their algorithm is an extension of Phong shading in which quadratic interpolation of normals is done. It overcomes the inappropriateness of traditional linear interpolation when a surface approximated by polygons has

* Lecturer, Department of Computer Science, Technical Teachers Training Institute, Chandigarh, India
wekta@yahoo.com

† Professor&Head, Department of Computer Science &Engg, Punjabi University, Patiala, India
chandan@pbi.ac.in

inflection points. However, their algorithm has many defects and is unsuited to standard triangle scan conversion.

Lee and Jen [5] have proposed an algorithm that improves upon the defects and problems of the method given in [4]. They have prpose d a bi-quadratic normal vector interpolation method, which performs biquadratic normal vector intepolation across a t riangle. The results of their algorithm as applied to a zigzag surface are shown in figure 3.

Since Phong shading is based on computation of normals, Thurmer, Grit and Charles [6] and Max [7] have prpose d methods for computing the exact normals.

2.2 Attempts to improve speed:

Bishop and Weimer [8] used a second order Taylor Series apprximation to simplify the normalization operations and thus reduce the computational cost. Kuijk and Blake[9] interpolated polar angles instead of normal vectors to eliminate normalization operations and thus make the shading faster. In their work, specular term has been approximated by piece-wise quadratic function, which acts directly on angle and not on cosine and therefore is almost as expensive as exponentiation involved in the original Phong model. Another technique to avoid the normalization of interpolated normal vectors is proposed in a work by Mohamed, Szirmay-Kalos and Horvath [10]. They have prp osed spherical interpolation strategy and have proved that by virtue of this method, the interpolated normals will automatically be normalized. But, it requires the use of arc cosine function, which is almost as expensive as normalization. Schlick [11] proposed a simple approximation to specular term. The work by Hast, Barrera and Bengtsson [12] proposes Hybrid shading which applies the use of Gouraud shading for diffuse component and Phong shading for specular highlight component. Overveld and Wyvill [13] used an infinite distance model instead of the expensive finite distance model for computation. This model works by replacing a light source at a finite distance, by a light source at an infinite distance .The normal vectors are adjusted in such a way that the resulting illumination pattern remains the same. But, in this method, the overhead for rotating the normal vectors is expensive and a Mach Band is easily introduced by the look up table. Cho, Neumann and Woo [14] describe a fast test that determines the triangles that have highlights on them. The potentially highlighted triangles are regularly subdivided in four triangles and Gouraud shdin g is applied. This method prdu ces Phong-shaded quality images. Lai and Tai [15] have prpose d highlight track shading i.e. a method to catch the highlight –track for a triangle so that all normal vectors of pixels on this track are calculated accurately. Tai and Hsu [16] suggest a method to identify which polygons should be Phong shaded, Gouraud shaded or Fence shaded (interpolation of normals along the edges and interpolation of color along the scan line). This leads to an improvement over other fast realistic shading methods. Lin et al. [17] propose an efficient and effective approach for fast Phong shading by using local maximal specular estimation in specular term for light at both finite and infinite distance. Seiler [18] has also suggested a simple way to set up coefficients to implement quadratic shading in rendering hardware. Brown [19] prposes the use of Bezier triangles to interpolate specular highlights. He suggests that the convex hull prperty of Bezier triangles allow a simple test that eliminates the need to interpolate specular highlights for 90 percent of pixels.

The important point is that all these methods[8-19] are an attempt to improve speed of linear or quadratic interpolation of normals. However, it has been proved in [4] and [5] that linear and quadratic interpolation is not suitable for every type of surface. This paper presents an algorithm which is applicable to standard triangle scan conversion and is faster than the algorithm in [5] which provides better shading results for all surfaces.

3. Proposed Algorithm:

The proposed method results in visual quality comparable to that of biquadratic normal vector interpolation given in [5] but is faster than it. The algorithm calculates the normals at the six points on a triangle usig a daptive normal vector interpolation given in [5]. These points are three vertices of a triangle and three mid points of the edges of a triangle as shown in Fig. 1. After the normals have been computed, a Bezier triangle is set-up with the help of a method given in [19].

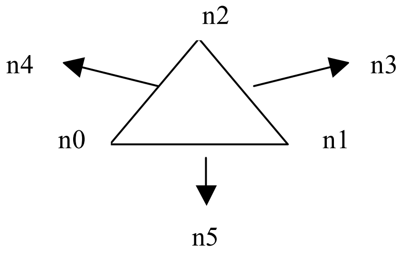


Figure 1: Keypoints where normals are calculated

It defines a bi-quadratic shading function for the underlying planar triangle. The planar nature of these triangles permit rapid calculation of barycentric coordinates that may be usd as interpolants in the bi-quadratic shading function. Although the method proposed in [19] was intended to interpolate specular highlights, the present algorithm uses this method for the interpolation of diffuse as well as specular highlight component.

For the sake of completeness, all the steps of bi-quadratic intensity interpolation algorithm are given here as follows:

1. The normals n0, n1 and n2 at the vertices are determined by averaging the face normals. The normals n3, n4 and n5 are determined using adaptive normal vector interpolation by detecting whether the curve approximated by the polygon edge is having serpentine profile or having an arch-type nature, as mentioned in [5]. If the nature of the curve is serpentine, quadratic interpolation is used to determine the normal at the mid-point of the edge. If the nature of the curve is arch-type, linear interpolation is used to determine the normal at the midpoint of the edge. To determine the nature of the curve, following test is performed:-

If n0 and n1 are the two normals at the two ends of the edge, and E represents the edge vector, then

If (((n0.E)>=0 && (n1.-E)>=0)||((n0.E) <=0 && (n1,-E)<=0)) then
the curve is arch-type
else
the curve is zigzag in nature.

2. Set-up the quadratic Bezier triangle as mentiond in [19]. The six control points on the surface are shown in Fig. 2. Barycentric coodinates can be determined from the screen space coodinates of a point p using the screen space areas $a_0,a_1,a_2,$ of the triangles $pP_{020}P_{002}$, $pP_{002}P_{200}$ and $pP_{200}P_{020}$ respectively. The barycentric coodinates are evaluated as:

$$\mathbf{sum} = a_0+a_1+a_2$$
$$b_0 = a_0/\mathbf{sum}$$
$$b_1 = a_1/\mathbf{sum}$$
$$b_2 = a_2/\mathbf{sum}$$

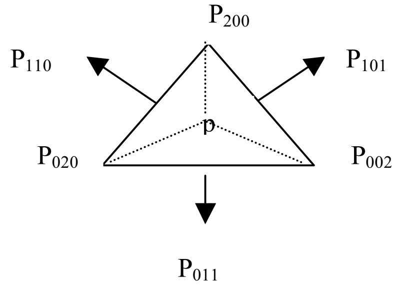


Figure 2: Control points on the Bezier triangle

3. These barycentric coordinates are employed to perform quadratic interpolation for determining the diffuse and the specular components of the intensity at a given point p. The cosines of an angle (hence forth referred as cosine) between vectors **N** and **H** and between vectors **N** and **L** for the highlight as well as diffuse component respectively, are determined usig the six normals com puted at three vertices and at three midpoints of the edges of the triangle. These cosines namely c_{020}, $c_{002,}$ c_{200}, $c_{011,}$ c_{101}, c_{110} are used to create six control points (for vertices and midpoints of the edges), using the expression of the type:

$$C_{011} = 4c_{011}-C_{020}-C_{002}$$

where C_{020} , C_{002} are vertex control points and have same value as the value of cosine c_{020}, c_{002} at the vertices. c_{011} is the cosine at the midpoint of the edge between P_{020} and P_{002}.

4. Cosines calculated for the diffuse component as well as specular component can be placed in the following equation to obtain their respective values at a point p, whose barycentric coodinates are b 0, b1, b2.

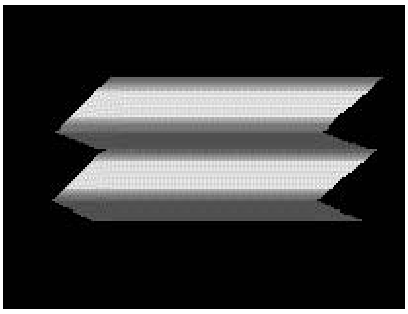
$$c(b_0,b_1,b_2)= C_{200}b_0^2+C_{020}b_1^2+C_{002}b_2^2+C_{110}b_0b_1+C_{011}b_1b_2+C_{101}b_2b_0 \qquad \text{(Equation 1)}$$

where C_{200},C_{020},C_{002} ,C_{011}, C_{110},C_{101} are control points generated from the cosines at the six keypoints of the surface.

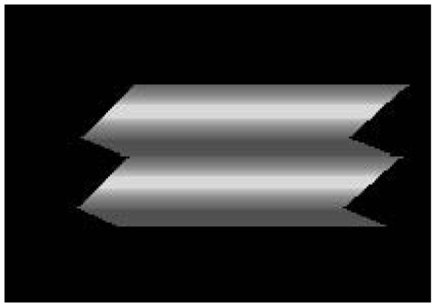
5. Convex Hull prperty of quadratic bezier triangle can be used to limit the calculation of specular highlights to those triangles on which highlight is predicted to occur. For this pupose, limits can be defined so that specular highlights will occur only for the cosines lying in a narrow range,say, *Clow* and *Chigh*, where

Chigh = cos(0) = 1 and Clow = cos(∅) where ∅ is the largest angle for which a specular reflection is visible. If any one of the control points is greater than *Clow*, then a specular highlight might occur on the triangle. In this case, it is necessary to perform quadratic interpolation of the specular cosines calculated at the six points of the triangle. However, if all of the control points are less than *Clow*, then no specular highlight will occur in the triangle and therefore, it is not necessary to perform quadratic interpolation of highlight cosines.

6. Table lookup can then be performed to determine the color corresponding to the intensity levels computed at each pixel.

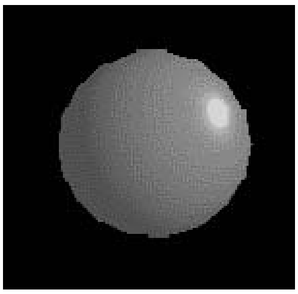


Result of biquadratic normal interpolation

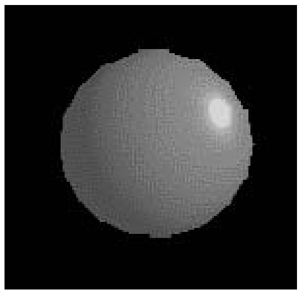


Result of fast proposed algorithm

Figure 3: Results of Zigzag surface(tessellated with 8 triangles) shaded with proposed algorithm and biquadratic normal interpolation, respectively



Result of biquadratic normal vector interpolation



Result of fast proposed algorithm

Figure 4: Results of a sphere(tessellated with 136 triangles) shaded with proposed algorithm and biquadratic normal interpolation, respectively

4. Visual Results and Comparison of Computation:

The above algorithm has been implemented in C++.

The results of this algorithm when executed for shading a surface with a zigzag profile (tessellated with 8 triangles) and a sphere (tessellated with 136 triangles) are shown in Fig. 3 and 4. The quality of shading is almost the same as the result obtained from biquadratic normal vector interpolation algorithm in [5].

Setup Cost:

Since the proposed algorithm differs from the algorithm in [5] after step .2, we compare the computation requirement from step 2 onwards.

The cost of setting up Bezier triangle and computing cosines is only 42 additions and 42 multiplications whereas the cost of setting up the coefficients for biquadratic normal interpolation of [5] is 99 multiplications, 71 additions and 15 divisions.

Cost Per-Pixel:

For our algorithm, the per-pixel operations include 18 additions, 1 division and 18 multiplications for triangles without highlights whereas for triangles with highlights the per-pixel operations include 23 additions,2 divisions and 30 multiplications.

However, for the algorithm in [5], the per-pixel cost is always 38 multiplications, 25 additions and 1 inverse square root operation for both the cases.

Since highlight calculations are required for only 10 percent of the pixels, the average per-pixel cost of our algorithm is 19.2 multiplications, 18.5 additions and 1.1 divisions.

The above comparison has been made for the case when viewer and light source are at infinite distance.

In case of viewer and light source being at finite distance, the cost of algorithm in [5] will be very high because for this case L and V will have to be reconstructed and normalized everytime prior to constructing H. No forward differencing has been considered for both the algorithms. The number of per-pixel operations can further be reduced with forward difference consideration.

Thus, the proposed algorithm is faster than the algorithm in [5] giving the same visual quality.

5. Conclusion:

This paper proposes a fast algorithm in the category of non-linear interpolation techniques for shading of three-dimensional objects. This algorithm has produced results which are comparable with the results of quadratic and bi-quadratic interpolation techniques and at the same time, it is very fast as compared to these techniques.

6. References:

[1] J. Foley, A. Van Dam, S. Feiner and J. Hughes, “Computer Graphics Principles and Practice Second Edition in C”, Addison Wesley, pp.734-735, 1996

[2] H. Gouraud, “Continuous Shading of Curved Surfaces”, IEEE Transactions on Computers Vol 20, No. 6, pp. 623-629, June 1971

[3] B.T.Phong, “Illumination for Computer Generated Pictures”, Communications of the ACM, Vol.18, No. 6, June1975

[4] C.W.A.M. Van Overveld and B. Wyvill, “Phong Normal Interpolation Revisited”, ACM Transactions on Graphics, Vol. 16, No. 4, pp. 397-419,1997

[5] Yuan-Chung Lee and Chein-Wei Jen, “Improved Quadratic Normal Vector Interpolation for Realistic Shading”, Visual Computer, Vol. 17, pp. 337-352,2001

[6] Thürmer, Grit, and A. Wuthrich Charles, "Computing Vertex Normals from Polygonal Facets," Journal of graphics tools, vol. 3, no. 1, pp. 43-46, 1998.

[7] Nelson Max, “Weights for Computing Vertex Normals from Facet Normals” Journal of Graphic Tools, Vol.4, No. 2, pp. 1-6,1999

[8] G. Bishop and D. M. Weimer, ”Fast Phong Shading”, ACM SIGGRAPH, Vol 20, No. 4, pp.103-106,1986

[9] A.A.M. Kuijk and E. H. Blake, ”Fast Phong Shading via Angular Interpolation”, Computer Graphics Forum 8, pp. 315-324, 1989
[10] Abbas Ali Mohamed, Laszlo Szirmay-Kalos and Tamas Horvath, “Hardware Implementation of Phong Shading using Spherical Interpolation”, Machine Graphics and Vision, Vol. 9, No. 4, pp. 825-804, 2001.
[11] Christope Schlick, "A Fast alternative to Phong's Specular Model", Graphics Gems, Vol 4, pp 363-366, Eds Paul Heckbert, Academic Press, 1994
[12] Anders Hast, Tony Barrera and Ewert Bengtsson, “Improved Shading Performance by Avoiding Vector Normalization”, Doctoral theses, Uppsala University, Sweden, 2001
[13] C.W.A.M. van Overveld and Brian Wyvill, “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, pp.25-30, 1996
[14] Youngkwan Cho, Ulrich Neumann and Jongwook Woo, “Improved Specular Highlights with Adaptive Shading” Computer Graphics International , June pp.24 - 28, 1996 ,Pohang, KOREA
[15] Yua-Yi Lai and Wen-Kai Tai, “Highlight Track Shading” International Computer Symposium Workshop on Computer Graphics & Virtual Reality, pp 45-51,2000.
[16] Wen Kai Tai and Shih-Chieh Hsu, “3-Tier Hierarchical Shading Based on Highlight Test” Fifth International workshop on Advanced Image Technology, Hualien, Taiwan, pp 179-182, 2002.
[17] Yeh-Chun Lin, Yueh-Yi Lai, Yao-Xun Chang and Chen-Duo Liu, “Fast Phong Shading Based on Local Maximal Specular Estimation”, Fifth International workshop on Advanced Image Technology, Hualien, Taiwan, pp 183-188, 2002.
[18] Larry Seiler, “Quadratic Interpolation for Near-Phong Quality Shading”, Technical Sketch, *SIGGRRAPH* 1998.
[19] Russ Brown, “Modeling Specular Highlights Using Bezier Triangles”, Technical Report, Sun MicroSystems Laboratories, TR-99-75,1999.