Approximation Techniques for High Performance Texture Mapping

Mehmet Demirer
University of Sussex
England

Richard L. Grimsdale
University of Sussex
England *

Abstract

Accurate perspective mapping in real-time requires costly division operations per pixel and therefore approximation techniques are often employed. These permit the mapping to be performed by interpolation, but generally with a significant set-up cost for the computation of the parameters. An efficient approximation technique which achieves good results with modest set-up requirements is presented. The technique uses Chebyshev control points to minimise errors.

1 Introduction

Texture mapping involves both geometric transformations and, to avoid aliasing, filtering operations. Only the former is considered here, with the main emphasis on texture coordinate calculation. Since the correct mapping function involves the evaluation of a quotient per pixel[6], it is computationally expensive. Therefore, approximate methods are generally employed, although these can introduce errors[3]. To improve the accuracy, polygons may be subdivided, however, as the number of polygons increases, the calculation becomes more expensive. Several approximation techniques have been reported [1] [8]. Both scan line and polygon based approximation techniques have been investigated. These approximation techniques in both cases require a considerable number of operations for the calculation of interpolation coefficients. We present here a new quadratic and cubic approximation techniques for both cases with significantly reduced set-up cost and improvement in the accuracy of the results.

2 Perspective Mapping

In order to map a 2D image onto an object in a perspective mapping, equation (1) must be performed at each pixel.

$$u = \frac{Ax + By + C}{Gx + Hy + I}$$
$$v = \frac{Dx + Ey + F}{Gx + Hy + I} \tag{1}$$

where x and y are screen coordinates, u and v are texture coordinates and A to I are coefficients which characterise the mapping function. Although this is an efficient solution to texture mapping in software, the division operation makes real time performance hard to achieve with hardware, since the two texture coordinates (u, v) must be produced every 50ns to satisfy real-time constraints. The distinctive feature of perspective mapping is that it preserves lines in all orientations, however, lines converge to a vanishing point unless they are parallel to the projection plane.

3 Conventional Approximation Techniques

3.1 Linear Approximation

The simplest and most widely used approximation technique is linear approximation. However, it results in severe errors in perspective projection, since the linear approximation is a poor representation of a rational function. Therefore, linear approximation cannot generate the foreshortening effects of true perspective [5] [6].

3.2 Quadratic Approximation along a Scanline

Perspective mapping defined by equation (1) can be approximated by a quadratic function as below:

$$u = Ax^2 + Bx + C$$
$$v = Dx^2 + Ex + F \tag{2}$$

*e-mail: R.L.Grimsdale@sussex.ac.uk

where A to F are constants. In order to calculate these coefficients, the mapping of three points, i.e. the two ends of a scanline and its midpoint of each scan line, are required. Thus, a general solution for the polynomial coefficients can be found [1]. The quadratic coefficients are:

$$
\begin{aligned}
A &= \frac{2(u_0 - 2u_1 + u_2)}{(x_0 - x_2)^2} \\
B &= -\frac{x_0u_0 + 3x_2u_0 - 4x_0u_1 - 4x_2u_1 + 3x_0u_2 + x_2u_2}{(x_0 - x_2)^2} \\
C &= \frac{x_0x_2u_0 + x_2^2u_0 - 4x_0x_2u_1 + x_0^2u_2 + x_0x_2u_2}{(x_0 - x_2)^2} \qquad (3)
\end{aligned}
$$

3.3 Cubic Approximation along a Scanline

In a similar way to quadratic approximation, a cubic expression can also be employed along a scanline. The general expression of the cubic interpolation polynomial is:

$$
\begin{aligned}
u &= a_3x^3 + a_2x^2 + a_1x + a_0 \\
v &= b_3x^3 + b_2x^2 + b_1x + b_0 \qquad (4)
\end{aligned}
$$

where a_0 to b_3 are constants. There are four unknown coefficients for the cubic approximation thus four constraints must be imposed to calculate them. For example, the polynomial passes through the two endpoints of a span and satisfies the imposed conditions on the first derivative. The calculation of derivatives and the coefficients are given in [1].

3.4 Biquadratic Approximation

Equation (1) can be approximated with a quadratic function in two variables [3], [8].

$$u \approx (Ax + By + C)^2 \qquad (5)$$

Re-arranging the equation above, this equation can be expressed in the form:

$$u \approx A_1x^2 + A_2y^2 + A_3xy + A_4x + A_5y + A_6 \qquad (6)$$

similarly,

$$v \approx B_1x^2 + B_2y^2 + B_3xy + B_4x + B_5y + B_6 \qquad (7)$$

The biquadratic equation given in equation (6) is for a quadratic surface and quadratic interpolation is thus set up by fitting this surface to six points: the vertices and side midpoints of a triangle. Thus, using Gaussian elimination method, six simultaneous equations are solved to obtain the interpolation coefficients. These six coefficients must be calculated before texture coordinates are generated. Then, using the forward differencing method, interpolation is performed.

3.5 Bicubic Approximation

A bicubic interpolation is an improvement over the biquadratic technique. The mapping equation is:

$$
\begin{aligned}
u &\approx (Ax + By + C)^3 \\
&\approx A_1x^3 + A_2y^3 + A_3x^2y + A_4xy^2 + A_5x^2 + A_6y^2 \\
&\quad + A_7xy + A_8x + A_9y + A_{10} \qquad (8)
\end{aligned}
$$

The 10 coefficients must also be calculated before texture coordinate generation. The control points can be chosen in several ways, provided certain constraints are satisfied over the polygon. However, in order to solve the 10 simultaneous equations, Gaussian elimination is again required, which requires approximately 1000 operations. This represents an excessive computational cost.

4 Chebyshev Approximation

The solution that we adopt to approximate the perspective mapping equation is a combination of the well-known Chebyshev and the Lagrange interpolation methods. The approximation with Chebyshev roots exploits the rather special properties of the use of unequally distributed data points and evenly distributed errors. Chebyshev points are located in the span [-1, 1] and can be applied to any other range by mapping it into the range of interest. Thus, the approximation polynomial using Chebyshev points can be derived. The arithmetic operations for these derivations are given in [2] and [7].

4.1 Second Order Polynomial

The second order approximation polynomial along a scanline uses 3 Chebyshev points in the span [a,b]. The 3 points are calculated by: 2

$$z_n = \frac{1}{2}\left[(b-a)\cos\left(\frac{K + \frac{1}{2} - n}{K}\right)\pi + a + b\right] \qquad (9)$$

Then, the general formula for the interpolation polynomial is:

$$g(z) = T_0 + T_1z + T_2z^2 \qquad (10)$$

where

$$
\begin{aligned}
T_0 &= Az_1z_2 + Bz_0z_2 + Cz_0z_1 \\
T_1 &= -[A(z_1 + z_2) + B(z_0 + z_2) + C(z_0 + z_1)] \\
T_2 &= A + B + C \qquad (11)
\end{aligned}
$$

and

$$A = \frac{f(z_0)}{(z_0 - z_1)(z_0 - z_2)}$$

$$
\begin{aligned}
B &= \frac{f(z_1)}{(z_1 - z_0)(z_1 - z_2)} \\
C &= \frac{f(z_2)}{(z_2 - z_0)(z_2 - z_1)}
\end{aligned} \tag{12}
$$

4.2 Third Order Polynomial

In a similar way to that described in the section, the third order interpolation polynomial can be obtained. The general formula is:

$$g(z) = T_0 + T_1 z + T_2 z^2 + T_3 z^3 \tag{13}$$

where

$$
\begin{aligned}
T_0 &= -[z_2 z_3(Az_1 + Bz_0) + z_0 z_1(Cz_3 + Dz_2)] \\
T_1 &= z_1 z_2(A + D) + z_1 z_3(A + C) + z_2 z_3(A + B) \\
&\quad + z_0 z_2(B + D) + z_0 z_3(B + C) + z_0 z_1(C + D) \\
T_2 &= -[z_1(A + C + D) + z_2(A + B + D) \\
&\quad + z_3(A + B + C) + z_0(B + C + D)] \\
T_3 &= A + B + C + D
\end{aligned} \tag{14}
$$

and

$$
\begin{aligned}
A &= \frac{f(z_0)}{(z_0 - z_1)(z_0 - z_2)(z_0 - z_3)} \\
B &= \frac{f(z_1)}{(z_1 - z_0)(z_1 - z_2)(z_1 - z_3)} \\
C &= \frac{f(z_2)}{(z_2 - z_0)(z_2 - z_1)(z_2 - z_3}} \\
D &= \frac{f(z_3)}{(z_3 - z_0)(z_3 - z_1)(z_3 - z_2)}
\end{aligned} \tag{15}
$$

Once the coefficients are calculated, u and v values follow through the use of forward differencing [1]. Therefore, three forward difference constants are used to approximate perspective mapping represented by a cubic polynomial. The same procedure is valid for the quadratic polynomial.

4.3 Biquadratic and Bicubic Interpolation on Chebyshev Points

In our approach, 9 control points for the biquadratic and 16 control points for the bicubic are used, which are the roots of the second and the third order Chebyshev polynomial along both the x and y axes. Figure 1 shows the positions of the Chebyshev points over a polygon. Although the equal intervals along both axes have been chosen, Figure 1(a) and (b), unequal intervals can also be evaluated as shown in Figure 1(c) and (d). In the first case, four Chebyshev points for the span [0,1] are: $z_0 = 0.038060234, z_1 = 0.308658295, z_2 = 0.691341736$ and $z_3 = 0.961939758$. Since interpolation will be performed across a polygon and span lengths are equal along x and y axes, the true values of $F(x_m, y_n)$ are calculated for $x_0, y_0, \ldots, x_3, y_3$ where $z_0 = x_0 = y_0, z_1 = x_1 = y_1, z_2 = x_2 = y_2$ and $z_3 = x_3 = y_3$.

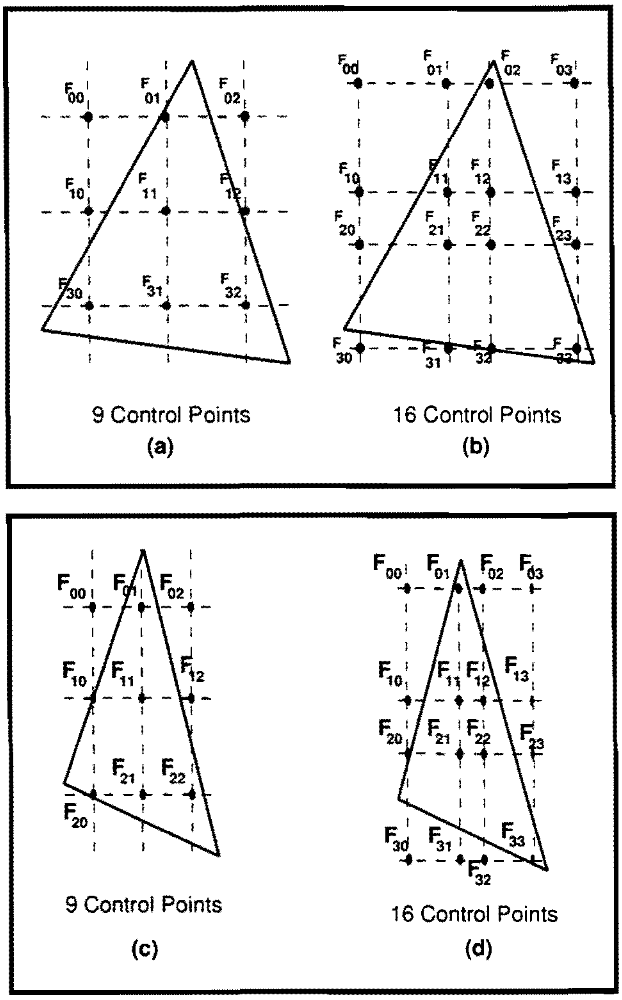

Figure 1: Chebyshev Control Points

Thus, after a four-corner correspondence has been established between the unit square in texture space and an arbitrary quadrilateral in screen space, scan-conversion is performed by the interpolation function obtained by using Chebyshev points. The bicubic polynomial can be derived in several ways. We have used the transfinite interpolation method. The general transfinite interpolation function is:

$$
\begin{aligned}
F(x, y) &= \sum_{m=0}^{M} \Phi_m(x) F(x_m, y) + \sum_{n=0}^{N} \Psi_n(y) F(x, y_n) \\
&\quad - \sum_{m=0}^{M} \sum_{n=0}^{N} \Phi_m(x) \Psi_n(y) F(x_m, y_n)
\end{aligned} \tag{16}
$$

where

$$\Phi_m(x) = \prod_{k=0,k\neq m}^{M} \frac{x-x_k}{x_m-x_k}$$

$$\Psi_n(y) = \prod_{k=0,k\neq n}^{N} \frac{y-y_k}{y_n-y_k} \quad (17)$$

and $M = N = 3$.

Also, $F(x, y_n)$ and $F(x_m, y)$ can be expressed as:

$$F(x, y_n) = \sum_{m=0}^{3} \Phi_m(x) F_{mn}$$

$$F(x_m, y) = \sum_{n=0}^{3} \Psi_n(y) F_{mn} \quad (18)$$

where F_{mn} values are the true values on Chebyshev points. When the auxiliary arithmetic operations are completed, the interpolation function takes the form:

$$\begin{aligned} F(x,y) = {} & H_1 y^3 x^3 + H_2 y^3 x^2 + H_3 y^3 x + H_4 y^3 \\ & + H_5 y^2 x^3 + H_6 y^2 x^2 + H_7 y^2 x + H_8 y^2 \\ & + H_9 y x^3 + H_{10} y x^2 + H_{11} y x + H_{12} y \\ & + H_{13} x^3 + H_{14} x^2 + H_{15} x + H_{16} \end{aligned} \quad (19)$$

The coefficients H_1 to H_{16} are calculated via another coefficient matrix Θ.

$$\Theta = \begin{bmatrix} C_X C_Y F_{00} & -C_Y F_{10} & C_Y F_{20} & -C_X C_Y F_{30} \\ -C_X F_{01} & F_{11} & -F_{21} & C_X F_{31} \\ C_X F_{02} & -F_{12} & F_{22} & -C_X F_{32} \\ -C_X C_Y F_{03} & C_Y F_{13} & -C_Y F_{23} & C_X C_Y F_{33} \end{bmatrix} \quad (20)$$

$$\Theta = \begin{bmatrix} \theta_{00} & \theta_{10} & \theta_{20} & \theta_{30} \\ \theta_{01} & \theta_{11} & \theta_{21} & \theta_{31} \\ \theta_{02} & \theta_{12} & \theta_{22} & \theta_{32} \\ \theta_{03} & \theta_{13} & \theta_{23} & \theta_{33} \end{bmatrix} \quad (21)$$

where $C_X = (x_2 - x_1)/(x_3 - x_0)$ and $C_Y = (y_2 - y_1) / (y_3 - y_0)$. Further algebraic manipulation results in simplified expressions for the coefficients which are now given by:

$$H_1 = \sum_{i=0}^{3}\sum_{j=0}^{3} \theta_{ij}$$

$$H_2 = -\sum_{i=0}^{3} D_i \sum_{j=0}^{3} \theta_{ij}$$

$$H_3 = \sum_{i=0}^{3} E_i \sum_{j=0}^{3} \theta_{ij}$$

$$H_4 = -\sum_{i=0}^{3} F_i \sum_{j=0}^{3} \theta_{ij}$$

$$H_5 = -\sum_{j=0}^{3} A_j \sum_{i=0}^{3} \theta_{ij}$$

$$H_6 = \sum_{j=0}^{3} A_j \sum_{i=0}^{3} D_i \theta_{ij}$$

$$H_7 = -\sum_{j=0}^{3} A_j \sum_{i=0}^{3} E_i \theta_{ij}$$

$$H_8 = \sum_{j=0}^{3} A_j \sum_{i=0}^{3} F_i \theta_{ij}$$

$$H_9 = \sum_{j=0}^{3} B_j \sum_{i=0}^{3} \theta_{ij}$$

$$H_{10} = -\sum_{j=0}^{3} B_j \sum_{i=0}^{3} D_j \theta_{ij}$$

$$H_{11} = \sum_{j=0}^{3} B_j \sum_{i=0}^{3} E_i \theta_{ij}$$

$$H_{12} = -\sum_{j=0}^{3} B_j \sum_{i=0}^{3} F_i \theta_{ij}$$

$$H_{13} = -\sum_{j=0}^{3} C_j \sum_{i=0}^{3} \theta_{ij}$$

$$H_{14} = \sum_{j=0}^{3} C_j \sum_{i=0}^{3} D_i \theta_{ij}$$

$$H_{15} = -\sum_{j=0}^{3} C_j \sum_{i=0}^{3} E_i \theta_{ij}$$

$$H_{16} = \sum_{j=0}^{3} C_j \sum_{i=0}^{3} F_i \theta_{ij} \quad (22)$$

5 Accuracy

The Chebyshev polynomial has n zeros in the interval $[-1, 1]$ as given in equation (9) and $n + 1$ maxima and minima, located at:

$$x_m = \cos\left(\frac{k\pi}{n}\right) \quad (23)$$

$k = 0, 1, \ldots, n$.

The maxima and minima of the polynomial can only be equal to 1 and -1 respectively. This property makes Chebyshev polynomials useful in approximating functions [7]. Although the particular approximation in a certain interval using N Chebyshev roots (provided that N is big enough) may not be better than any other having some other set of N data points, the approximation polynomial can be truncated to a polynomial of lower degree m in a very graceful way that yields the most accurate approximation of degree m. Since the polynomial is bounded between -1 and 1, the difference between N^{th} and m^{th} ($m \ll N$) order polynomials cannot be larger than the sum of neglected coefficients, which typically decrease rapidly. Therefore, the error is dominated by an oscillatory function with $m+1$ extrema distributed over the interval. This truncated approximation is nearly the same polynomial as the *minimax polynomial* [7], which has the smallest maximum deviation from the true function among all polynomials of the same degree and is very difficult to find.

6 Cost

We have considered several approximation techniques in previous sections. In order to calculate approximation coefficients for the conventional biquadratic and bicubic interpolation polynomials, the Gaussian elimination method is used. This method requires approximately N^3 operations where N denotes the number of equations. Therefore, much computational power is needed. As for the Chebyshev approximation, many fewer operations are required and the results are more accurate as a greater number of control points are used. The requirements for the traditional and Chebyshev approximation techniques are given in Table 1.

Set-up Operations		
Operations	Scanline	Polygon
Traditional Quad.	36	> 216
Traditional Cubic	> 75	> 1000
Chebyshev Quad.	24	138
Chebyshev Cubic	62	310

Table 1: Number of Operations

7 Results

Figures 4 to 8 show the results of scanline based approximation techniques. In order to compare the methods, a rather difficult perspective view has been chosen. Generally, all images generated by approximate methods exhibit errors related to preserving straight and diagonal lines. Careful examination of squares near the viewing point shows deformation of the nearest black square. Line errors and distortions of squares become less visible in Figure 8, which uses the Cubic Chebyshev approximation along a scanline. Figures 9 and 10 also show two different graphs of approximate coordinate versus true coordinates. The straight line represents the accurate result and it can be seen that the error with the use of the Chebyshev approximation method are less than with the others. Figures 2 and 3 also compare the results of the true algorithm and the Bicubic Chebyshev approximation technique. The perspective projection chosen for these views is also considered to be a difficult case and can give rise to severe deformations in the checkerboard pattern at large perspective distances if the mapping technique is not sufficiently accurate. However, the errors in the Bicubic Chebyshev approximation technique are acceptably small and distortion of straight lines is only apparent in a few diagonals, after careful examination.

8 Discussion

The method described in section 4.3 has been applied to a single rectangular surface. It uses 16 control points for the bicubic polynomial, which are chosen as the roots of the third order Chebyshev polynomial. It may not always be possible to restrict the surface to a rectangular shape and it may take the form of a general quadrilateral or a triangle. Furthermore, in a model with a number of facets, there can be a multiplicity of polygons with the same texture pattern. Since the Chebyshev points would be chosen independently for two adjacent surfaces, there would be a discontinuity between the texture patterns along the common edge. Therefore the use of Chebyshev points is impractical in the case. However, as can be observed from equations (16) and (17), the control points can be arbitrarily chosen over a polygon at a cost of a few more multiplication operations. Although the errors will now be greater, the technique is still more accurate than other bicubic approximation methods due to extra control points.

9 Conclusion

A linear approximation is a poor fit for a rational function and in an extreme case, a quadratic one also fails. Therefore, a cubic approximation can be considered a reasonable technique. Traditional techniques are reasonably accurate in the middle range of the interval, but the error increases towards the edges. On the other hand, using Chebyshev approximation methods, errors become more evenly distributed throughout the interpolation and their magnitudes become less than the other techniques. The reason for this is that the traditional methods use equi-spaced data points whereas Chebyshev approximation employs roots of Chebyshev polynomial as data points, which are defined by a cosine function and the spacing between these points is greatest at the centre of the interpolation domain, decreasing towards the edges. The results show that the Cubic and Bicubic Chebyshev approximation techniques provide rather acceptable images at much less cost in comparison with the conventional techniques. Hardware design can be based on either scan line or polygon rendering algorithms. Therefore, real time or near real time performance can be achieved.

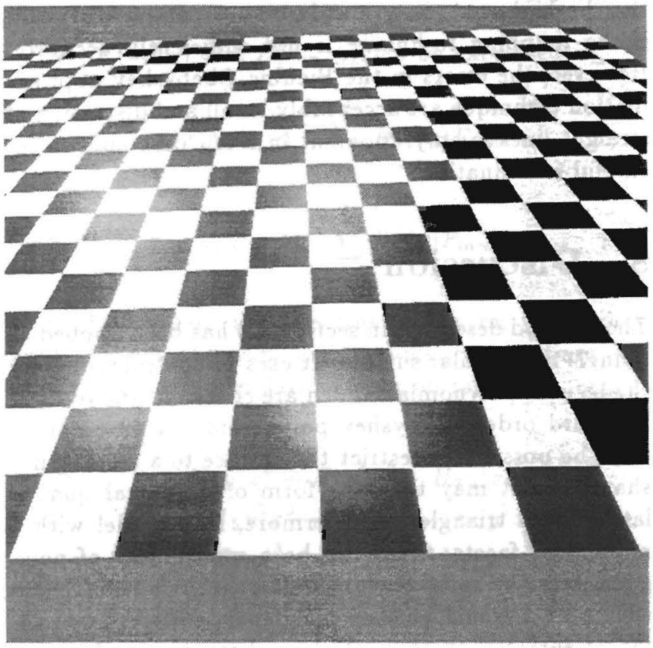

Figure 2: True Texture Coordinates

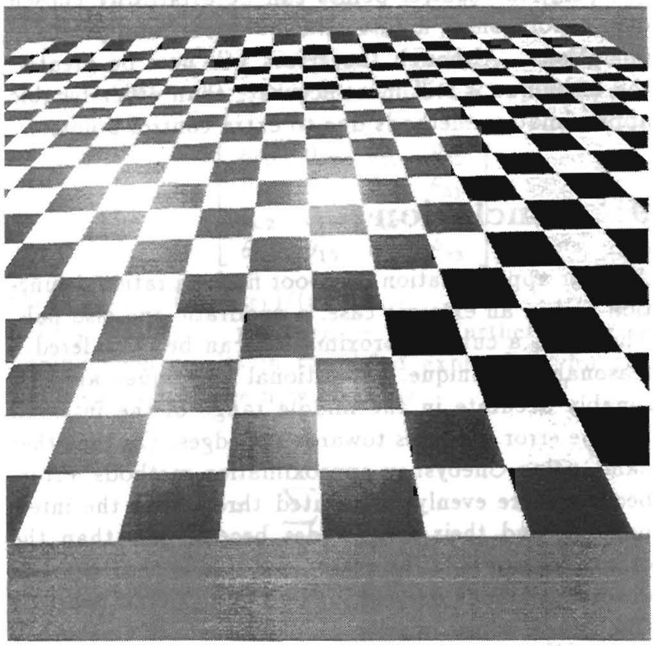

Figure 3: The Bicubic Chebyshev Approximation

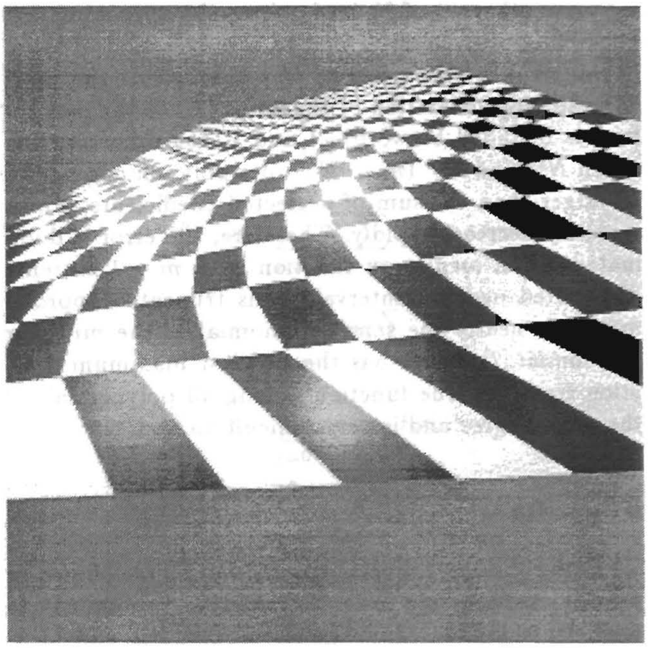

Figure 4: Linear Approximation

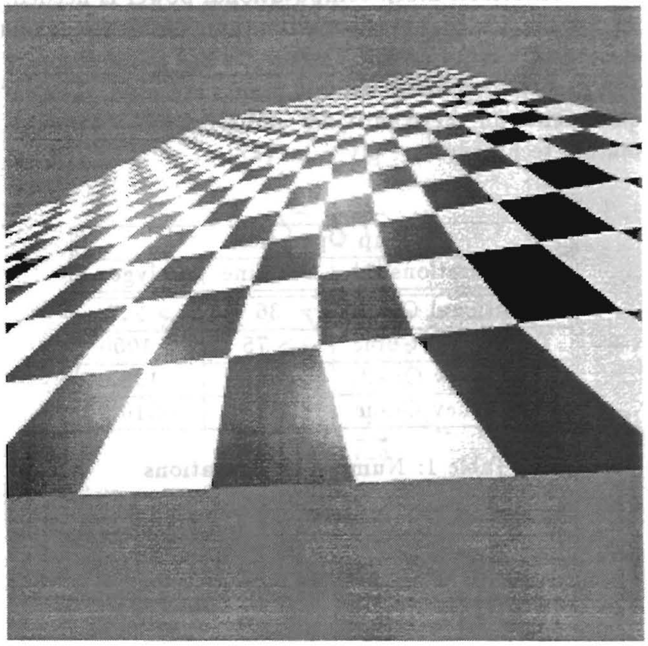

Figure 5: Direct Quadratic Approximation

Bibliography

1. Wolberg, G., Digital Image Warping, IEEE Computer Society Press, 1990.

2. Nakamura, S., Applied Numerical Methods in C, Prentice-Hall, 1993.

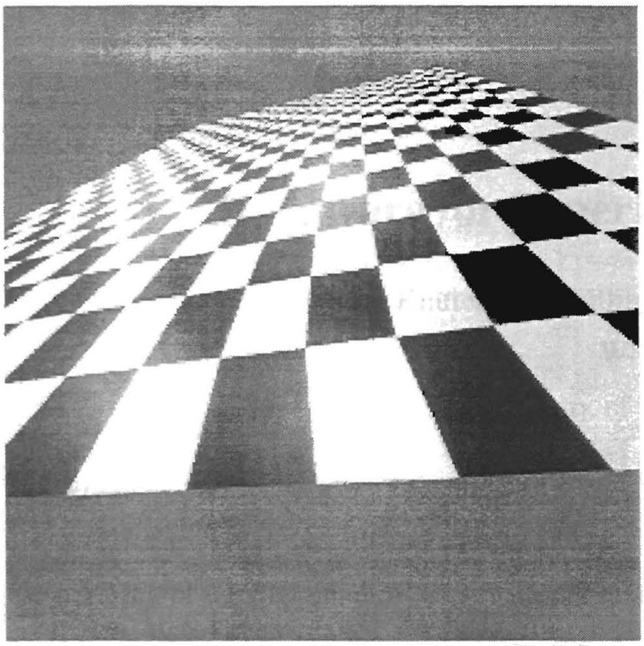

Figure 6: Quadratic Chebyshev Approximation

3. Kirk,D., Vorrhies, D., The Rendering Architecture of the DN10000VS Computer Graphics, Vol 24, no 4, August 1990.

4. Heckbert, P., Texture Mapping Polygons in Perspective, NYIT Tech.Memo No 13, April 1983.

5. Heckbert, P., A Survey of Texture Mapping, IEEE Computer Graphics and Applications, Nov. 1986

6. Heckbert, P., Moreton, HP., Interpolation for Polygon Texture Mapping and Shading, State of the Art in Computer Graphics Visualization and Modelling, Springer-Verlag 1991.

7. Press, W.H., Numerical Recipes in C, Cambridge University Press, 1992.

8. Dunnett, G.J., Texture Mapping Architectures for High Performance Image Generation, DPhil Thesis, Sussex University, 1994.

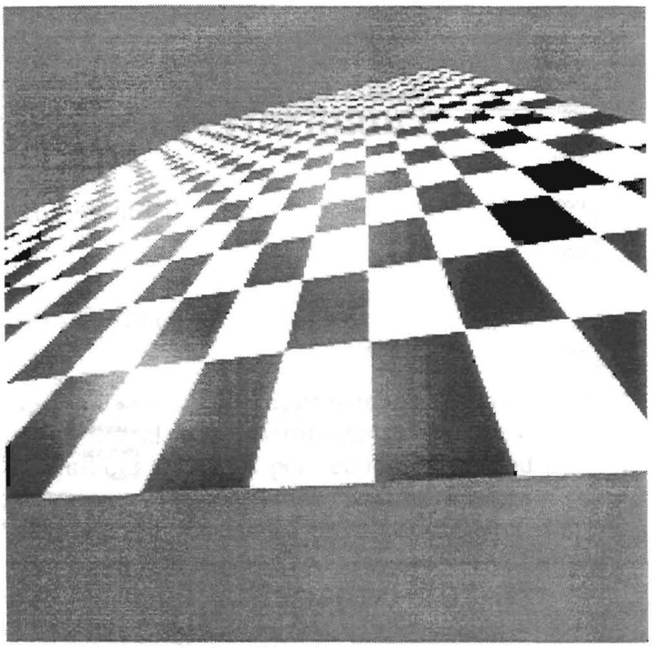

Figure 7: True Texture Coordinates

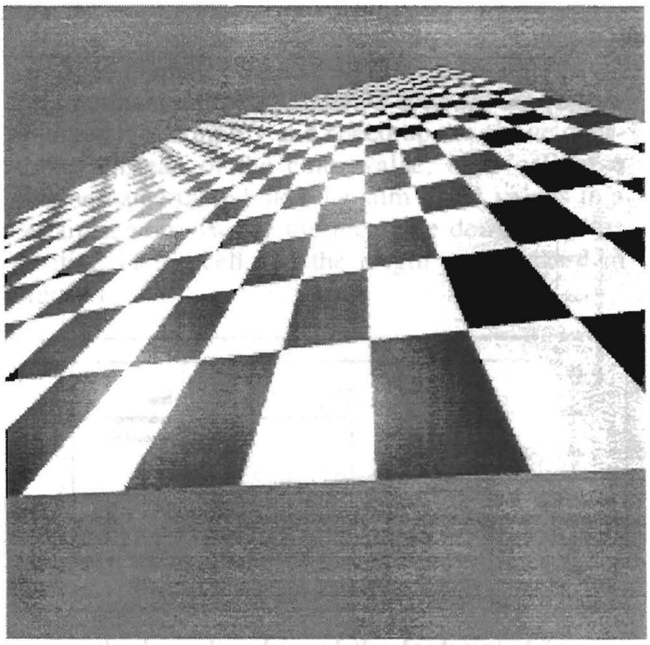

Figure 8: Cubic Chebyshev Approximation

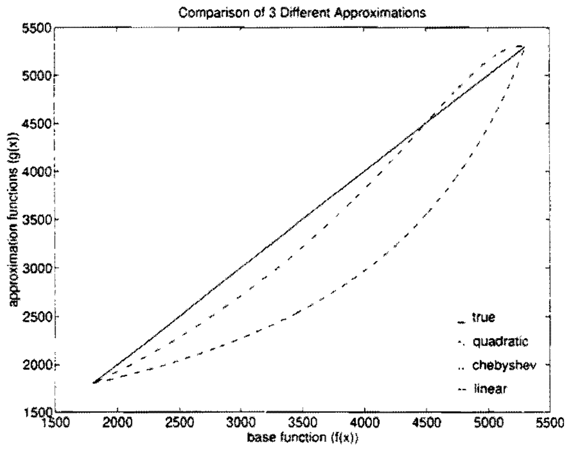

Figure 9: Comparisons of Case 1

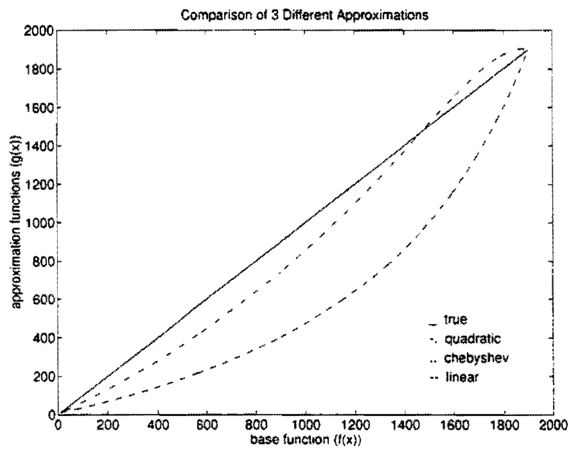

Figure 10: Comparisons of Case 2