CSTD-75-004

A BOOLEAN SUM INTERPOLATION SCHEME TO RANDOM DATA FOR COMPUTER AIDED GEOMETRIC DESIGN

by

Charles Christian Poeppelmeier

A thesis submitted to the faculty of the
University of Utah in partial fulfillment of the requirements
for the degree of

Master of Science

Department of Computer Science

University of Utah

December 1975

UNIVERSITY OF UTAH GRADUATE SCHOOL

SUPERVISORY COMMITTEE APPROVAL

of a thesis submitted by

Charles C. Poeppelmeier

I have read this thesis and have found it t

degree.

19 Sept 75 — Richard F. Riesenfeld

Date

Richard F. Riesenfeld
Chairman, Supervisory Committee

I have read this thesis and have found it to be o

degree.

25 Sept 75 — Elaine Cohen

Date

Elaine Cohen
Member, Supervisory Committee

I have read this thesis and have found it to be of satisfactory quality for a master's degree.

14 August 1975 — Robert E. Barnhill

Date

Robert E. Barnhill
Member, Supervisory Committee

UNIVERSITY OF UTAH GRADUATE SCHOOL

FINAL READING APPROVAL

To the Graduate Council of the University of Utah:

I have read the dissertation of Charles C. Poeppelmeier in its final form and have found that (1) its format, citations, and bibliographic style are consistent and acceptable; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the Supervisory Committee and is ready for submission to the Graduate School.

8 Sept 1975
Date

Robert E. Barnhill
Robert E. Barnhill
Member, Supervisory Committee

Approved

Robert E. Stephenson
Robert E. Stephenson
Chairman/Dean

Appro

Sterling M. McMurrin
Sterling M. McMurrin
Dean of the Graduate School

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to the following persons: Richard Riesenfeld for his frequent invaluable advice, Robert Barnhill for suggesting the topic of this thesis and Elaine Cohen for consenting to be a member of my committee at such a late date. Also, I wish to thank Steven A. Coons and Stan Weinstein for the help they gave me with my research and Mike Milochik for his excellent photographic work.

I owe a special debt of gratitude to my beautiful wife, Linda, who not only typed this thesis but also helped to ease the burdens of graduate school.

This research was supported by the National Science Foundation (DCR74-13017) and the Advanced Research Projects Agency of the Department of Defense (DAHC15-73-C-0363), for which I would like to express my thanks.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This thesis presents a new interpolation function for randomly distributed data. The new interpolation function is capable of exactly reproducing quadratic surfaces. The new function is developed, through boolean sum theory, from Shepard's two dimensional interpolation functions and the Barnhill-Gregory nine parameter interpolation function. Computer generated examples of the new interpolation function are given.

Chapter I

INTRODUCTION

This thesis is a study of the problem of deriving, from randomly distributed data, a mathematical model of a smooth surface for computer display. We seek a scheme that would work directly upon irregularly spaced data. The scheme should not depend upon any topology being imposed upon the data. The data for the model shall involve only the function value and the first two derivatives at the data points. The model should interpolate all the data, and the model should be able to reproduce simple polynomials. That is if our data belongs to a simple polynomial surface then our model should reproduce that surface, exactly.

In 1968, D. Shepard [11] introduced a bivariate interpolation function for irregularly spaced data. It is based on a weighted average of the values at the data points where the weighting for a point on the surface is a power function of the inverse distance to the data points. This scheme will interpolate functional and derivative data. Shepard's function reproduces plane surfaces, but it does not reproduce higher polynomial surfaces.

A new interpolation function for irregularly spaced data will be derived by taking the boolean sum of Shepard's function and the Barnhill-Gregory nine parameter interpolant. We recall that the boolean sum of two projectors P and Q is

$$P \oplus Q = P + Q - PQ .$$

A projector is a linear, idempotent operator [4]. An operator P is linear if

$$P[af + bg] = aP[f] + bP[g]$$

and P is idempotent if

$$P[P[f]] = P[f] .$$

In 1974, R. E. Barnhill [1] showed that the boolean sum $P \oplus Q$ of two projectors has the duality property that $P \oplus Q$ possesses at least the interpolation properties of P and the function precision of Q. The function precision of Q is the set of polynomials that Q will reproduce. The new interpolation function for irregularly spaced data will then have the interpolation properties of Shepard's function and the function precision of the Barnhill-Gregory nine parameter interpolant.

Chapters II and III contain a review of Shepard's function and the Barnhill-Gregory interpolation function, respectively. In Chapter IV, the new interpolation function for irregularly spaced data is derived. Chapter V contains some conclusions.

Chapter II

SHEPARD'S FUNCTION

In 1968, D. Shepard [11] introduced a two dimensional interpolation function for irregularly spaced data. He assumed that a finite number of triplets (x_i, y_i, z_i) were given where the x_i, and y_i are the positional coordinates and z_i is the corresponding data value. The bivariate interpolating function is

$$P(x,y) = \begin{cases} \dfrac{\sum\limits_{i=0}^{m} d_i^{-u} f(x_i,y_i)}{\sum\limits_{i=0}^{n} d_i^{-u}} & \text{if } d_i \neq 0 \text{ for all } i \\ f(x_i,y_i) & \text{if } d_i = 0 \text{ for some } i \end{cases} \qquad [2.1]$$

where $d_i^{-u} = 1/[(x-x_i)^u + (y-y_i)^u]$ and $u > \emptyset$. This function will assign the value $P(x,y)$ to any location in the plane. It thus has the ability to extrapolate outside of the convex polygon hull containing all the data points.

This function interpolates irregularly spaced data. It does not depend on any ordering being imposed upon or presented with the data. It is based on a weighted average of the values at the data points, where the weighting is an inverse power function of the distance to the data points. The most weight is given to the nearest data point and the least weight to the farthest data point.

This form of Shepard's function in [2.1] is numerically unstable because as the $(x,y) \to (x_i,y_i)$, the $d_i^{-u} \to \infty$. If we convert the d_i^{-u} to a common denominator, Shepard's function can be rewritten as

$$S_m[f;x,y] = \sum_{i=0}^{m} A_i(x,y)f(x_i,y_i) \qquad [2.2]$$

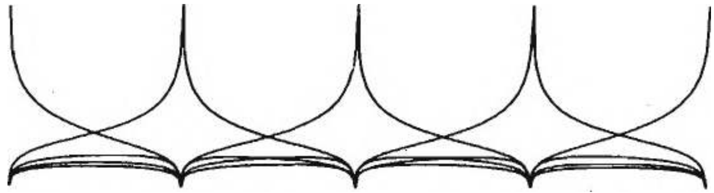
where the $A_i(x,y)$ are

$$A_i(x,y) = \frac{\prod_{\substack{j=0\\ j\neq i}}^{m} (|x-x_j|^u + |y-y_j|^u)}{\sum_{k=0}^{m} \prod_{\substack{\ell=0\\ \ell\neq k}}^{m} (|x-x_\ell|^u + |y-y_\ell|^u)} \qquad [2.3]$$

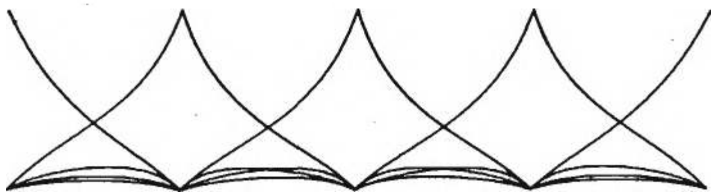
This form of Shepard's function is not only numerically more stable, it is also easier to interpret. For $u > 0$, these basis functions possess the cardinal basis property,

$$A_i(x_j,y_j) = \delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \qquad [2.4]$$

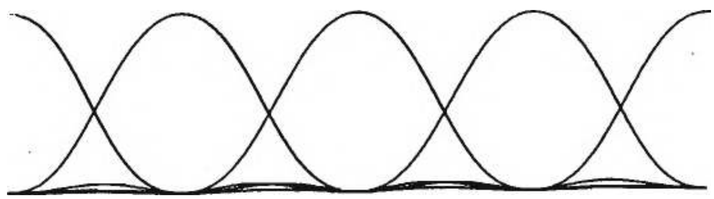
It remains to investigate the effects of the value of the exponent u in the inverse distance basis functions. If $u \leq 0$ the basis functions are not cardinal and Shepard's function will not interpolate the data. Specifically, if $u = 0$, then $A_i(x,y) = 1/n$ and $S_m[f;x,y]$ gives us the mean of the data values. If $0 < u \leq 1$, then there is a discontinuity at the data points in the basis functions [Fig. 2.1a and 2.1b]. This discontinuity in the basis functions is inherited by the surface. Therefore, if we want a smooth function that will interpolate



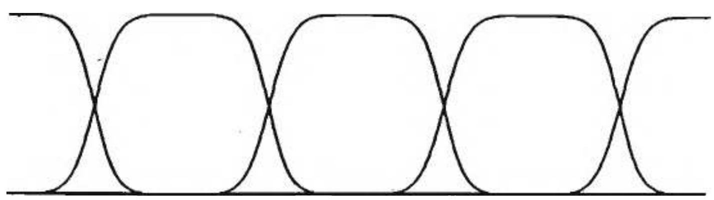
(a) u = .5



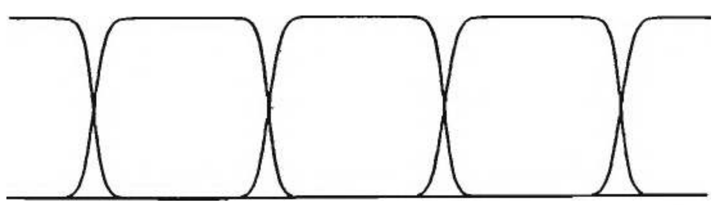
(b) u = 1.0



(c) u = 2.0



(d) u = 4.0



(e) u = 8.0

Figure 2.1 Shepard's inverse distance basis functions.

the data, then $u > 1$. However, not all values of $u > 1$ give satisfactory results. For large values of u, the basis functions are relatively flat near the data points and then between the data points the basis functions have a large change in slope [Fig. 2.1d and 2.1e]. A surface with these basis functions will share the relative flatness near the data points and have a large change in slope between the data points. A value of $u = 2$ for the exponent gives seemingly satisfactory basis functions [Fig. 2.1c]. They have no large changes in slope that can be inherited by the surface. We shall use only $u = 2$ for the surfaces in this thesis.

Since this basic interpolation function has zero derivatives at all the data points [Fig. 2.2], Shepard extended this function to also interpolate the derivatives at all the data points by using the Taylor series expansion about the data points [Fig. 2.3]. The new function which also interpolates the derivatives at the data points is

$$S_m[f;z,y] = \sum_{i=0}^{m} A_i(x,y)[f(x_i,y_i)+(x-x_i)f_x(x_i,y_i)+(y-y_i)f_y(x_i,y_i)] \qquad [2.5]$$

where the $A_i(x,y)$ are the same as [2.3].

This function [2.5] is precise for planes. If all the data is from a plane surface then the higher order derivatives in the Taylor series expansion are zero and the Taylor series represents the plane, $f(x,y)$. Now we can replace the Taylor series with the function $f(x,y)$ and since the basis functions sum to one for any value of (x,y), Shepard's interpolation function will reproduce the plane.

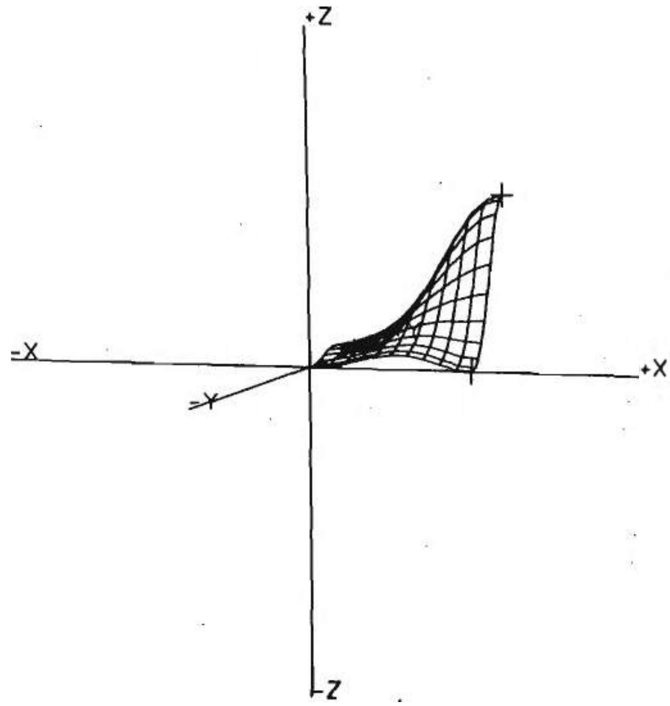


Figure 2.2 Shepard's bivariate interpolation function with zero derivatives at the data points.

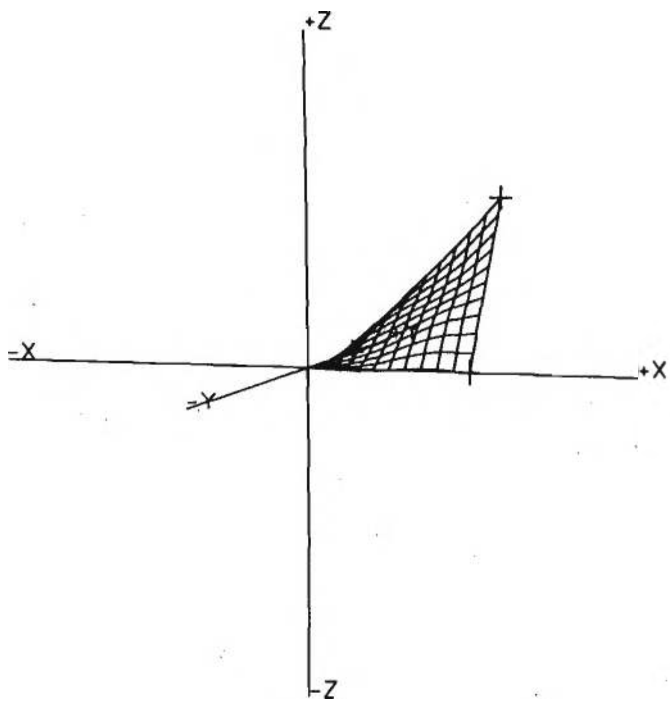


Figure 2.3 Shepard's bivariate interpolation function which also interpolates the derivatives at the data points.

$$S_m[f;x,y] = \sum_{i=0}^{m} A_i(x,y)[f(x_i,y_i) + (x-x_i)f_x(x_i,y_i) + (y-y_i)f_y(x_i,y_i)]$$

$$= \sum_{i=0}^{m} A_i(x,y)f(x,y) \qquad [2.6]$$

$$= f(x,y)$$

Equation [2.5] is the bivariate form of Shepard's function. Shepard's function can be extended to n-dimensions. However, we will only consider the univariate and bivariate forms for curves and surfaces, respectively.

We can consider S_m to be an operator defined on the domain $R \times R$. S_m is a linear operator:

$$S_m[af + bg] = aS_m[f] + bS_m[g]. \qquad [2.7]$$

Also S_m is an idempotent operator:

$$S_m[S_m[f]] = S_m[f]. \qquad [2.8]$$

This follows by definition of the uniqueness of interpolation. Therefore, S_m is a projector since it is a linear, indempotent operator.

The univariate form of Shepard's projector for curves is

$$S_m[f;x] = \sum_{i=0}^{m} B_i(x)f(x_i) \qquad [2.9]$$

where

$$B_i(x) = \frac{\prod_{\substack{j=0 \\ j\neq i}}^{m} |x - x_j|^u}{\sum_{k=0}^{m} \prod_{\substack{\ell=0 \\ \ell\neq k}}^{m} |x - x_\ell|^u}, \quad u > 0 . \qquad [2.10]$$

This form of Shepard's projector has zero derivatives at all the data points.

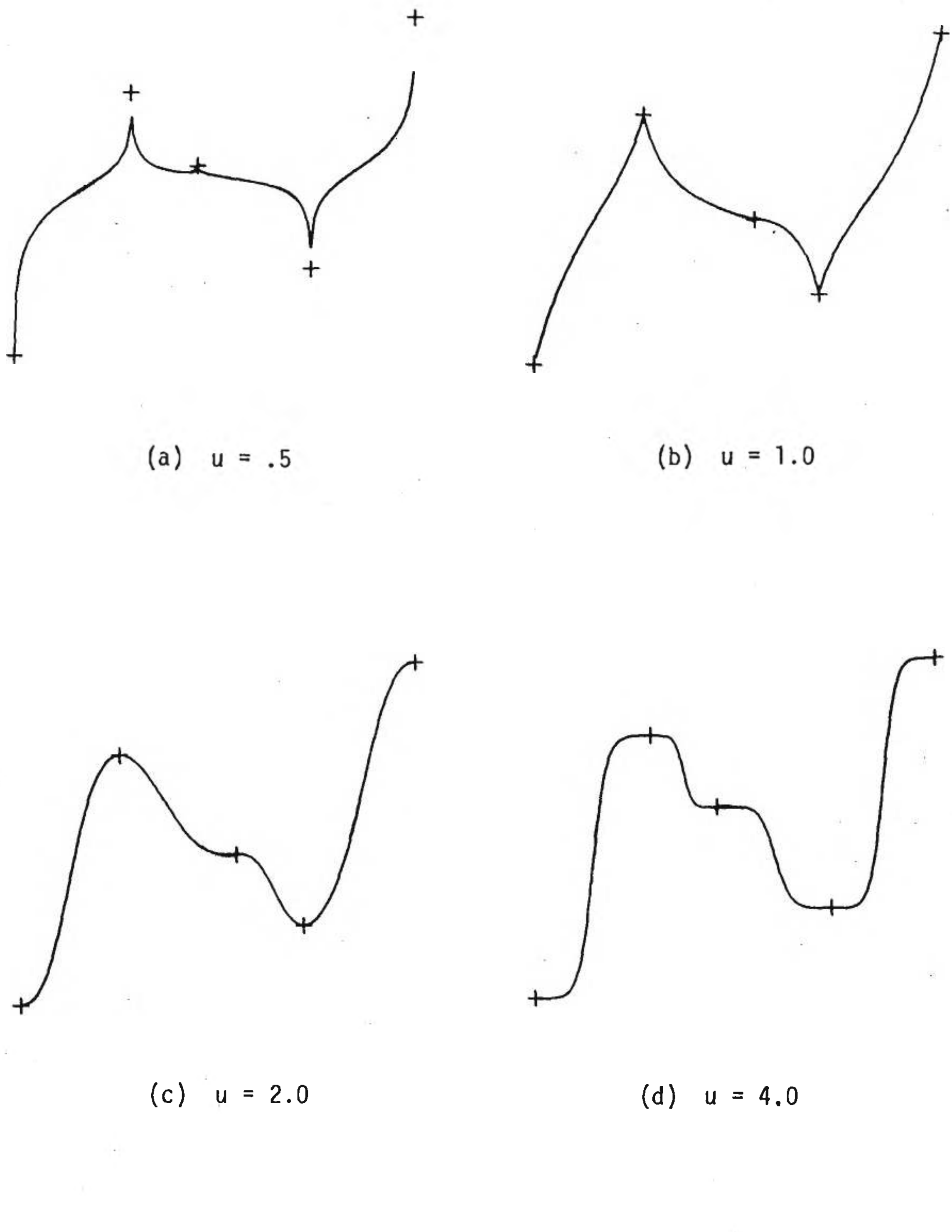
$$\left.\frac{dB_i(x)}{dx}\right|_{x=x_i} =$$

$$= \frac{\left[\sum_{\substack{r=0\\ r\neq 1}}^{m} u|x_i-x_r|^{u-1} \prod_{\substack{j=0\\ j\neq r\\ j\neq i}}^{m} |x_i-x_j|^u\right] \sum_{k=0}^{m} \prod_{\substack{\ell=0\\ \ell\neq k}}^{m} |x_i-x_\ell|^u - \prod_{\substack{j=0\\ j\neq i}}^{m} |x_i-x_j|^u \left(\sum_{k=0}^{m} u|x_i-x_k|^{u-1} \prod_{\substack{\ell=0\\ \ell\neq k}}^{m} |x_i \cdots\right.}{\left(\sum_{k=0}^{m} \prod_{\substack{\ell=0\\ \ell\neq k}}^{m} |x_i-x_\ell|^u\right)^2}$$

$$= 0$$

We can see the effect that changing the value of the exponent u in the basis functions has on the curves. In Figure 2.4a and 2.4b, the exponent $u = .5$ and $u = 1$, respectively. The curve interpolates the data points but is discontinuous in slope at all the data points. In Figure 2.4c the exponent $u = 2$ and the curve is now smooth. In Figure 2.4d the exponent u is equal to four. The curve is flat at the data points and has large changes in slope midway between the data points. A value of two for the exponent in the cardinal basis functions gives the most satisfactory results.

In Chapter IV, we will use the bivariate form of Shepard's function [2.5] that interpolates the function and gradients at the data points. This form of Shepard's function has the interpolating properties that we need for our new two dimensional interpolation function.



(a) u = .5

(b) u = 1.0

(c) u = 2.0

(d) u = 4.0

Figure 2.4 Shepard's univariate interpolation function.

Chapter III

THE BARNHILL-GREGORY NINE PARAMETER INTERPOLANT

In this section we review the Barnhill-Gregory nine parameter interpolant over the standard triangle T and some of the properties of the interpolant. The standard triangle T has vertices V1 = (1,Ø), V2 = (Ø,1) and V3 = (Ø,Ø).

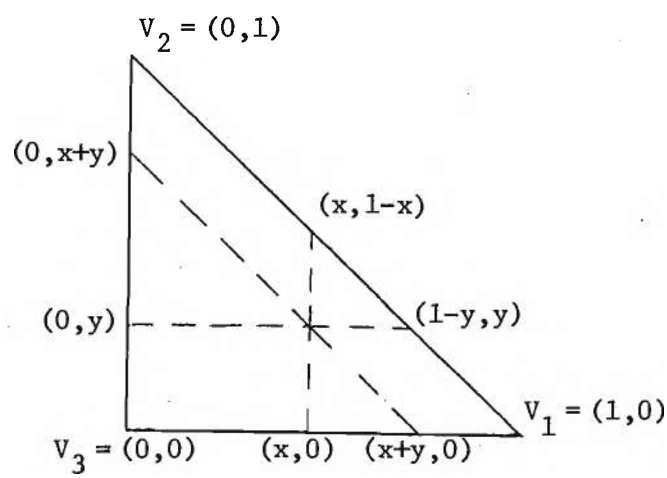


Figure 3.1 Standard Triangle T

The nine parameters are the three functionals F, $F_{1,0}$, and $F_{0,1}$ (function value, partial with respect to x, and the partial with respect to y) at the three vertices of the triangle.

The Barnhill-Gregory nine parameter scheme is constructed from the smooth C1 interpolant of Barnhill and Gregory. A smooth interpolant is a surface that interpolates to function and derivative boundary curves. The smooth C1 interpolant over the standard triangle T was introduced in 1974 by R. E. Barnhill and J. A. Gregory [2]. The smooth C1 interpolant is

$$U(x,y) = \sum_{i=0}^{1} \phi_i\left(\frac{y}{1-x}\right)(1-x)^i F_{0,i}(x,\emptyset) + \sum_{i=0}^{1} \psi_i\left(\frac{y}{1-x}\right)(1-x)^i F_{0,i}(x,1-x)$$

$$+ \sum_{i=0}^{1} \phi_i\left(\frac{x}{1-y}\right)(1-y)^i \left[F_{i,0}(\emptyset,y) - \left(\frac{\partial^i P_2F}{\partial x^i}\right)(\emptyset,y)\right] \qquad [3.1]$$

$$- \frac{x^2 y(x+y-1)}{x+y}\left[\left(\frac{\partial}{\partial y}\left(\frac{\partial F}{\partial x}\right)\right)(\emptyset,\emptyset) - \left(\frac{\partial}{\partial x}\left(\frac{\partial F}{\partial y}\right)\right)(\emptyset,\emptyset)\right]$$

where

$$(P_2F)(\emptyset,y) = \phi_0(y)F(\emptyset,\emptyset) + \phi_1(y)F_{0,1}(\emptyset,\emptyset) + \psi_0(y)F(\emptyset,1) + \psi_1(y)F_{0,1}(\emptyset,1)$$

$$\left(\frac{\partial P_2F}{\partial x}\right)(\emptyset,y) = y[\phi_0'(y)F(\emptyset,\emptyset) + \phi_1'(y)F_{0,1}(\emptyset,\emptyset) + \psi_0'(y)F(\emptyset,1)$$

$$+ \psi_1(y)F_{0,1}(\emptyset,1)] + \phi_0(y)F_{1,0}(\emptyset,\emptyset) + \phi_1(y)\left[-F_{0,1}(\emptyset,\emptyset)\right.$$

$$\left. + \left(\frac{\partial F_{0,1}(x,\emptyset)}{2x}\right)_{x=0}\right] + \psi_0(y)[F_{1,0}(\emptyset,1) - F_{0,1}(\emptyset,1)]$$

$$+ \psi_1(y)\left[-F_{0,1}(\emptyset,1) + \left(\frac{\partial F_{0,1}(x,1-x)}{\partial x}\right)_{x=0}\right]$$

The $\phi_i(t)$ and $\psi_i(t)$ are the cardinal basis functions for Hermite two point Taylor interpolation on $(\emptyset,1)$. They are

$$\phi_0(t) = (t-1)^2(2t+1) \qquad \phi_1(t) = (t-1)^2 t$$
$$\psi_0(t) = t^2(-2t+3) \qquad \psi_1(t) = t^2(t-1) \qquad [3.2]$$

The smooth C1 interpolant has quintic precision. The derivation and other details of the C1 smooth interpolant can be found in [2].

Now by defining the function and normal derivatives along a side of the triangle as polynomials which interpolate the data on that side, an interpolation scheme can be constructed which involves only point functional data on the triangle. An interpolant that depends only on a finite number of scalar parameters is called a finite dimensional interpolant. To construct the nine parameter finite dimensional interpolant we define the boundary functions as the cubic polynomials which interpolate to the function F and its first directional derivatives along the side at each vertex, and the normal derivatives as linear polynomials which interpolate the normal derivative at each vertex. This process of constructing the boundary curves from discrete data is called discretization.

The discretized boundary curves needed to construct the Barnhill-Gregory nine parameter interpolant from the smooth C1 interpolant are

$$\tilde{F}(x,0) = \phi_0(x)F(0,0) + \phi_1(x)F_{1,0}(0,0) + \psi_0(x)F(1,0) + \psi_1(x)F_{1,0}(1,0)$$

$$\tilde{F}_{0,1}(x,0) = (1-x)F_{0,1}(0,0) + xF_{0,1}(1,0)$$

$$\tilde{F}(x,1-x) = \phi_0(x)F(0,1) + \phi_1(x)[F_{1,0}(0,1) - F_{0,1}(0,1)] + \psi_0(x)F(1,0)$$
$$+ \psi_1(x)[F_{1,0}(1,0) - F_{0,1}(1,0)] \qquad [3.3]$$

$$\tilde{F}(\emptyset,y) = \phi_0(y)F(\emptyset,\emptyset) + \phi_1(y)F_{0,1}(\emptyset,\emptyset) + \psi_0(y)F(\emptyset,1) + \psi_1(y)F_{0,1}(\emptyset,1)$$

$$\tilde{F}_{1,0}(\emptyset,y) = (1-y)F_{1,0}(\emptyset,\emptyset) + yF_{1,0}(\emptyset,1)$$

$$\tilde{F}_{1,0}(x,1-x) + \tilde{F}_{0,1}(x,1-x) = (1-x)[F_{1,0}(\emptyset,1) + F_{0,1}(\emptyset,1)]$$

$$+ x[F_{1,0}(1,\emptyset) + F_{0,1}(1,\emptyset)]$$

Substituting the discretized boundary curves [3.3] into [3.1] gives the nine parameter finite dimensional interpolant

$$\begin{aligned}
U(x,y) = {}& \phi_0(\tfrac{y}{1-x})[\phi_0(x)F(0,0) + \phi_1(x)F_{1,0}(0,0) + \psi_0(x)F(1,0) + \psi_1(x)F_{1,0}(1,0)] \\
& + \phi_1(\tfrac{y}{1-x})(1-x)[(1-x)F_{0,1}(0,0) + xF_{0,1}(1,0)] \\
& + \psi_0(\tfrac{y}{1-x})[\phi_0(x)F(0,1) + \phi_1(x)[F_{1,0}(0,1) - F_{0,1}(0,1)] + \psi_0(x)F(1,0) \\
& + \psi_1(x)[F_{1,0}(1,0) - F_{0,1}(1,0)]] \\
& + \psi_1(\tfrac{y}{1-x})(1-x)[\tfrac{1}{2}\{(1-x)[F_{1,0}(0,1) + F_{0,1}(0,1)] + x[F_{1,0}(1,0) + F_{0,1}(1,0)] \\
& - \phi_0'(x)F(0,1) - \phi_1'(x)[F_{1,0}(0,1) - F_{0,1}(0,1)] - \psi_0'(x)F(1,0) \\
& - \psi_1'(x)[F_{1,0}(1,0) - F_{0,1}(1,0)]\}] \qquad [3.4] \\
& + \phi_0(\tfrac{x}{1-y})(1-y)\{(1-y)F_{1,0}(0,0) + yF_{1,0}(0,1) - y[\phi_0'(y)F(0,0) + \phi_1'(y)F_{0,1}(0,0) \\
& + \psi_0'(y)F(0,1) + \psi_1'(y)F_{0,1}(0,1)] - \phi_0(y)F_{1,0}(0,0) \\
& - \phi_1(y)[-F_{0,1}(0,0) - F_{0,1}(0,0) + F_{0,1}(1,0)] - \psi_0(y)[F_{1,0}(0,1) - F_{0,1}(0,1)] \\
& - \psi_1(y)[-F_{0,1}(0,1) + \tfrac{1}{2}[-F_{0,1}(0,1) - F_{1,0}(0,1) + F_{0,1}(1,0) + F_{1,0}(1,0) + 6F(0,1) \\
& + 4(F_{1,0}(0,1) - F_{0,1}(0,1)) - 6F(1,0) + 2(F_{1,0}(1,0) - F_{0,1}(1,0))]]\} \\
& - \frac{x^2y(x+y-1)}{x+y}[-F_{1,0}(0,0) + F_{1,0}(0,1) + F_{0,1}(0,0) - F_{0,1}(1,0)].
\end{aligned}$$

The nine parameter interpolant has quadratic precision. The precision of the nine parameter interpolant has been lowered to quadratic because of the lower precision of the boundary data interpolants. The boundary data interpolants [3.3] can only reproduce quadratics. The precision of a finite dimensional interpolant is limited by the precision achieved by the boundary data interpolants.

The Barnhill-Gregory nine parameter interpolation function was selected for use in deriving the new two dimensional interpolation function because data was available for its parameters and because of its precision. In the next chapter we will use the Barnhill-Gregory nine parameter interpolation function to improve the precision of the Shepard's two dimensional interpolation function.

Chapter IV

A NEW TWO DIMENSIONAL INTERPOLATION FUNCTION

This chapter introduces a new two dimensional interpolation function for irregularly spaced data. The new interpolation function is derived by taking the boolean sum of Shepard's two dimensional interpolation projector P_1 and the Barnhill-Gregory nine parameter interpolation projector P_2. The new function will have the interpolation properties of Shepard's projector P_1 and the precision of the Barnhill-Gregory nine parameter projector P_2.

A severe shortcoming of Shepard's interpolation projector is that it is precise only for planes. If all the data is from a higher polynomial surface, then Shepard's projector will not, in general, give back the polynomial surface.

In 1974, R. E. Barnhill [1] showed that the boolean sum $P \oplus Q$ of two projectors has the duality property that $P \oplus Q$ has at least the interpolation properties of P and the function precision of Q. Shepard's bivariate interpolation projector has the desired interpolation properties. Further, we can improve the precision of Shepard's projector by taking the boolean sum of it and another projector Q, where Q has the desired precision.

For purposes of illustration, we shall first consider the univariate case of Shepard's projector. For example, we can take the boolean sum of Shepard's univariate projector [2.9] and a linear projector to get linear precision for our new interpolation function. If we let P_1 be Shepard's univariate projector and let P_2 be the linear projector

$$P_2[f] = \frac{x - x_m}{x_1 - x_m} f(x_1) + \frac{x - x_1}{x_m - x_1} f(x_m) \qquad [4.1]$$

then the boolean sum

$$P_1 \oplus P_2 = P_1 + P_2 - P_1P_2 \qquad [4.2]$$

will have the interpolation properties of P_1 and the precision of P_2.

The boolean sum function is

$$P_1 \oplus P_2[f] = \sum_{i=0}^{m} B_i(x) f(x_i) + \frac{x - x_m}{x_1 - x_m} f(x_1) + \frac{x - x_1}{x_m - x_1} f(x_m) - \sum_{i=0}^{m} B_i(x) \left[\frac{x_i - x_1}{x_m - x_1} f(x_m) + \frac{x_i - x_m}{x_1 - x_m} f(x_1)\right] \qquad [4.3]$$

$$= \sum_{i=0}^{m} B_i(x) \left[f(x_i) - \frac{x_i - x_1}{x_m - x_1} f(x_m) - \frac{x_i - x_m}{x_1 - x_m} f(x_1)\right] + \frac{x - x_1}{x_m - x_1} f(x_m) + \frac{x - x_m}{x_1 - x_m} f(x_1) \qquad [4.4]$$

$$= \sum_{i=0}^{m} B_i(x) G_i + \frac{x - x_1}{x_m - x_1} f(x_m) + \frac{x - x_m}{x_1 - x_m} f(x_1) \qquad [4.5]$$

where

$$G_i = f(x_i) - \left[\frac{x_i - x_1}{x_m - x_1} f(x_m) + \frac{x_i - x_m}{x_1 - x_m} f(x_1)\right] \qquad [4.6]$$

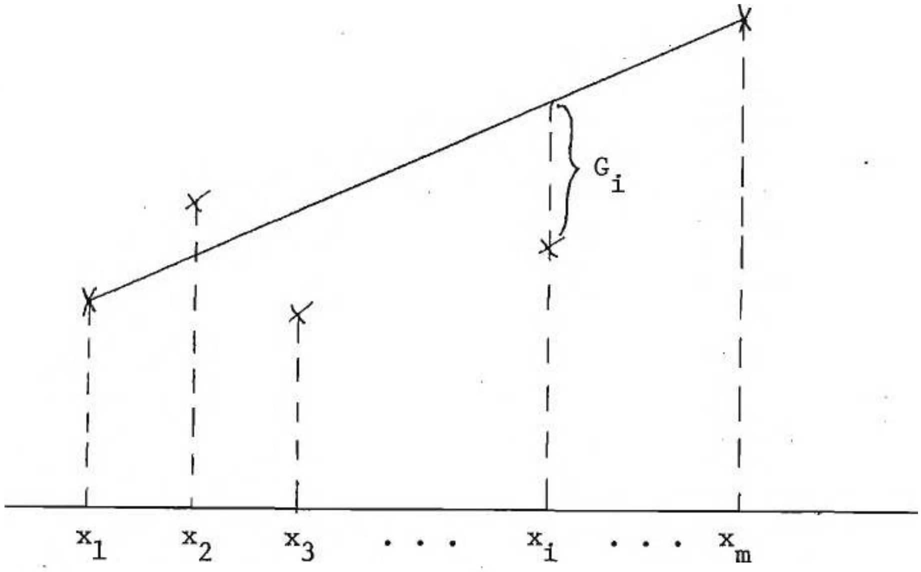


Figure 4.1 Geometric interpretation of the boolean sum of Shepard's univariate projector and the linear projector.

We can give the boolean sum $P_1 \oplus P_2$ a geometric interpretation [Fig. 4.1]. The G_i is the difference between the data point at x_i and the straight line. Now if all the data is linear then G_i will be equal to zero and we will be left with only the last part of the boolean sum function, and the line between the first and n-th data points. P_1, Shepard's univariate projector was based on the inverse distance weighting of the data points. The boolean sum interpolation function [4.5] is based on the inverse distance weighting of the difference between the data points and the line defined by P_2.

In Figure 4.2, we can see how Shepard's univariate projector interpolates linear data. This can be compared with the boolean sum function in Figure 4.3 which interpolates the same data. The boolean sum interpolation function is precise for straight lines.

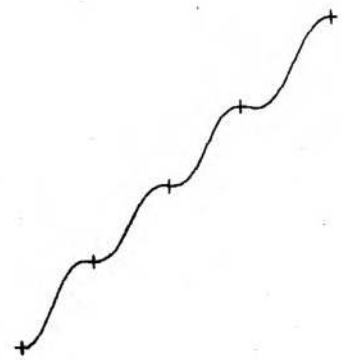


Figure 4.2 Shepard's univariate projector with linear data.

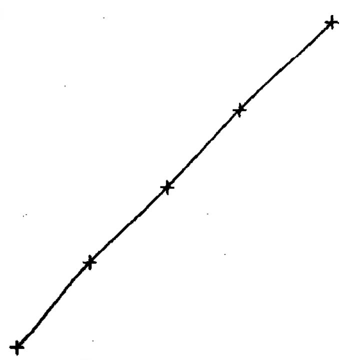


Figure 4.3 Boolean sum function with linear data.

We have seen [2.11] that the partial derivatives of the first projector, P_1, evaluated at the data points is zero. The partial derivative of the boolean sum function [4.5] evaluated at the data point x_i is

$$\left.\frac{\partial P_1 \oplus P_2[f]}{\partial x}\right|_{x=x_i} = \frac{f(x_m) - f(x_1)}{x_m - x_1} \qquad [4.7]$$

This is simply the derivative of the second projector P_2. If the derivative of first projector P_1 is zero at all the data points, then the boolean sum function $P_1 \oplus P_2$ inherits the derivative of the second projector P_2 at each data point. If we compare Shepard's univariate interpolation projector [2.9] in Figure 2.4c with the boolean sum interpolation function [4.3] in Figure 4.4, we can see that the boolean sum interpolation function inherits the slope of the second projector at each data point.

Similarly, we can now define the new two dimensional interpolation function by taking the boolean sum of Shepard's two dimensional interpolation projector and the Barnhill-Gregory nine parameter interpolation projector. The new interpolation function then has quadratic precision. Let P_1 be Shepard's projector [2.5] and P_2 be the Barnhill-Gregory nine parameter interpolation projector [3.4], then the boolean sum function is

$$\begin{aligned} P_1 \oplus P_2 = \sum_{i=0}^{m} A_i(x,y)\Bigg[&\left[f(x_i,y_i) - U(x_i,y_i)\right] \\ &+ (x-x_i)\left[f_{1,0}(x_i,y_i) - \frac{\partial U(x_i,y_i)}{\partial x}\right] \\ &+ (y-y_i)\left[f_{0,1}(x_i,y_i) - \frac{\partial U(x_i,y_i)}{\partial y}\right]\Bigg] + U(x,y) \end{aligned} \qquad [4.8]$$

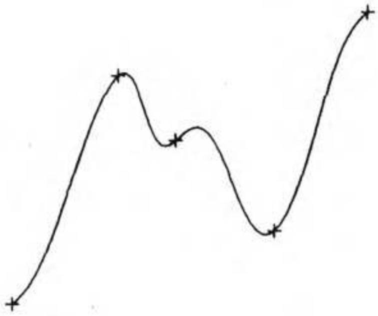


Figure 4.4 The boolean sum function.

where the cardinal blending functions are

$$A_i(x,y) = \frac{\prod_{\substack{j=0 \\ j\neq i}}^{m} \left(|x - x_j|^2 + |y - y_j|^2 \right)}{\sum_{k=0}^{m} \prod_{\substack{\ell=0 \\ \ell\neq k}}^{m} \left(|x - x_j|^2 + |y - y_j|^2 \right)} \qquad [4.9]$$

The value of a point of Shepard's bivariate interpolation projector [2.5] is based on the inverse distance weighting of the data point values. The new two dimensional interpolation function [4.9] is based on the inverse distance weighting of the difference between the values at the data points and the value of the quadratic surface defined by the Barnhill-Gregory nine parameter interpolation projector.

In Figure 4.5 and Figure 4.6, we can compare Shepard's two dimensional interpolation projector with the new two dimensional interpolation function. In Figure 4.5 and Figure 4.6 the data are from the functions $f(x,y) = xy$ and $f(x,y) = -x^2 - y^2 + 1$, respectively. These quadratic functions are in the precision set of the new two dimensional interpolation function. In Figures 4.5b and 4.6b, the new two dimensional interpolation function reproduces these functions exactly. Although Shepard's function does interpolate the data, it does not model the function very well [Figs. 4.5a and 4.6a].

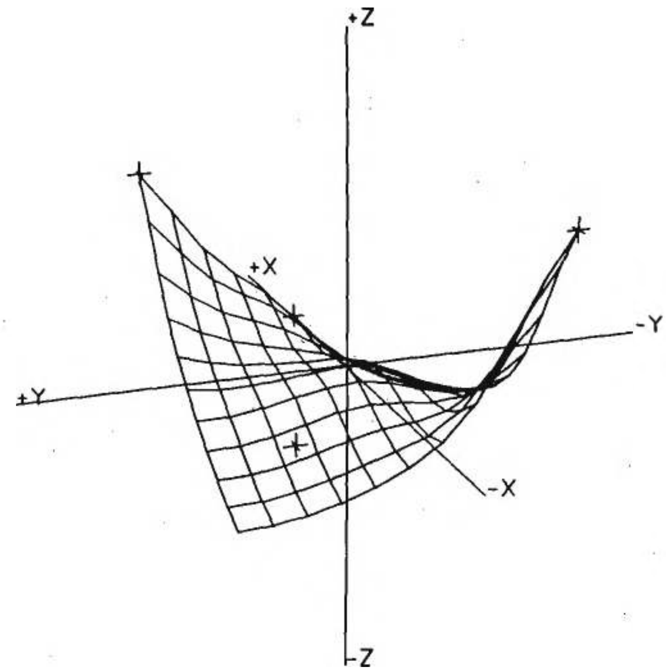


Figure 4.5(a) Shepard's bivariate interpolation function.

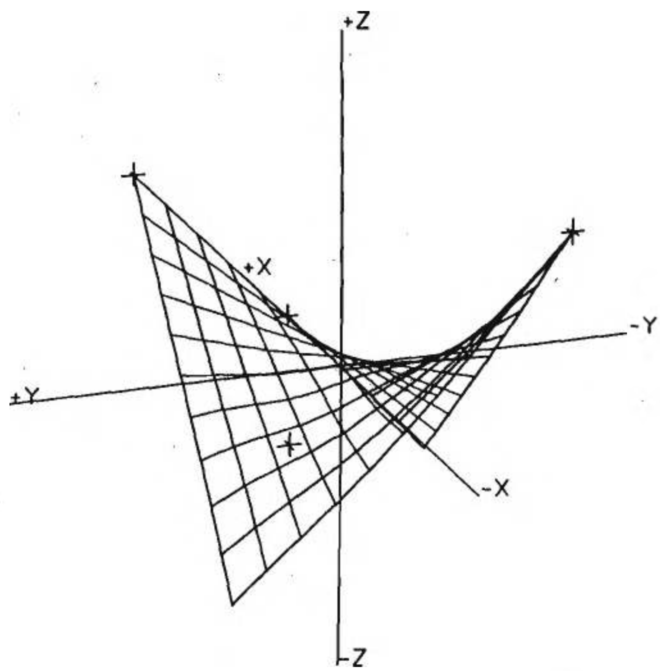


Figure 4.5(b) The new bivariate interpolation function with data from the function $f(x,y) = xy$.

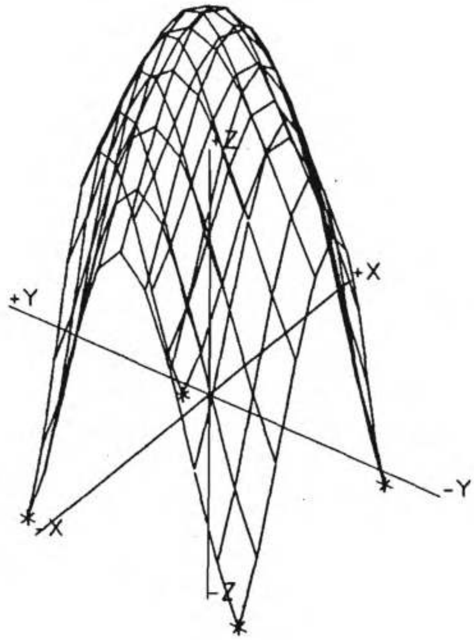


Figure 4.6(a) Shepard's bivariate interpolation function.

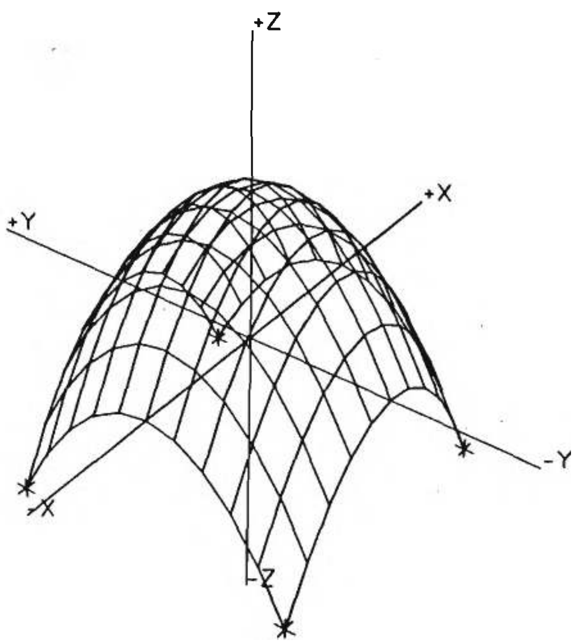


Figure 4.6(b) The new two dimensional interpolation function with data from the function $f(x,y) = - x^2 - y^2 + 1$.

Chapter V

CONCLUSION

The new two dimensional interpolation function satisfies all the requirements stated at the beginning. It produces smooth surfaces for computer display. It interpolates the function and derivative values at all the data points. The data points can be arbitrarily distributed in the Cartesian plane. Also, the new two dimensional interpolation function will reproduce quadratic surfaces.

A lack of generality in the two dimensional random data interpolation function is that it does not have a parametric representation. A search for a satisfactory domain for the parameters that would keep the surface from becoming folded was negative. An immediate idea was to use the data point numbers as the domain for the parametrization, but since the data is random the surface may become folded.

The inverse distance blending functions used in this thesis are an area for future research. Their main use has been as cardinal blending functions for random data interpolation. We have also seen how they can be used to produce nonparametric curves. Shepard's function for curves can also be expressed parametrically as

$$\overline{S}(t) = \sum_{i=0}^{m} B_i(t)[\overline{S}(t_i)] \qquad [5.1]$$

where

$$B_i(t) = \frac{\prod\limits_{\substack{j=0 \\ j \neq 1}}^{m} (t - t_j)^u}{\sum\limits_{k=0}^{m} \prod\limits_{\substack{\ell=0 \\ \ell \neq k}}^{m} (t - t_i)^u} \qquad [5.2]$$

and

$$1 \leqslant t \leqslant m$$

$$t_i = i$$

$$\bar{S}(t_i) = [x_i \; y_i]$$

The exponent u in the blending functions may be used as a parameter to change the shape of the surface. This may prove useful for surface representation in computer aided geometric design. In extension from one to two or more variables there are at least four possibilities to be considered. These are the operators S_m and S_n themselves in parametric form, the tensor product operator S_mS_n and the Boolean sum operator

$$S_m \oplus S_n = S_m + S_n - S_mS_n \, . \qquad [5.3]$$

APPENDIX

An interactive computer graphic system (IDSYS, for Interpolant Display SYStem) was designed and implemented to aid the study of the interpolation functions for irregularly spaced data. The system was implemented on an Evans and Sutherland Picture System. The Picture System is a stand alone, general purpose interactive computer graphics system which can display smoothly moving pictures of two or three dimensional objects.

The main components of the Picture System are a Digital Equipment Corporation PDP-11/45, a hardware matrix arithmetic processor, a refresh buffer and a 21 inch refresh display and a Summagraphics tablet. The PDP-11/45 is the picture controller. It contains the data base which describes the picture to be viewed. Also, it controls all the input and output required for the interactive graphics. The hardware matrix arithmetic processor performs digital operations on the data received from the picture controller and deposits it in the refresh buffer. The refresh buffer contains in digital form the picture to be displayed on the refresh display terminal. The tablet serves as an interactive graphic input device. It can perform the same interactive graphics functions as light pens, joysticks, and function switches.

IDSYS was written in FORTRAN for the Picture System. It uses the graphics software subroutine package furnished with the Picture System. IDSYS allows the user to dynamically control the display of the interpolation functions. IDSYS requires that the user enter the data from the keyboard of the teletype. The user must enter the position, function value, and the first two derivatives. After the user has entered all the data, he can then select from the menu [Figure A.1] of light buttons the commands he wishes to be executed. The interpolation functions are listed vertically along the left hand side of the screen and the commands to dynamically manipulate the picture are listed along the bottom of the screen. The remainder of the screen is a viewport for the three dimensional display of the surface produced by the interpolation function.

The user can interactively select one of the options from the menu through the use of the tablet. A cursor is drawn on the screen at a position which corresponds to the position of the pen on the tablet. This provides a visual feedback of the current position of the pen on the tablet. When the cursor is positioned over the chosen command, the user depresses the pen against the tablet. The program will then execute that command. One of the advantages of using light buttons over entering the commands through the keyboard is that the light buttons offer a much faster means of control.

The user can select for display one of three interpolation functions from the light buttons along the left hand side of the screen. The first light button is Shepard's function which interpolates the function and derivative values. The next light button selects the new random data interpolation function which is precise for quadratics. The last light button selects for display the version of Shepard's function which interpolates only the function value.

From the bottom of the screen the user can select from the many light buttons to dynamically manipulate the picture. The two columns of

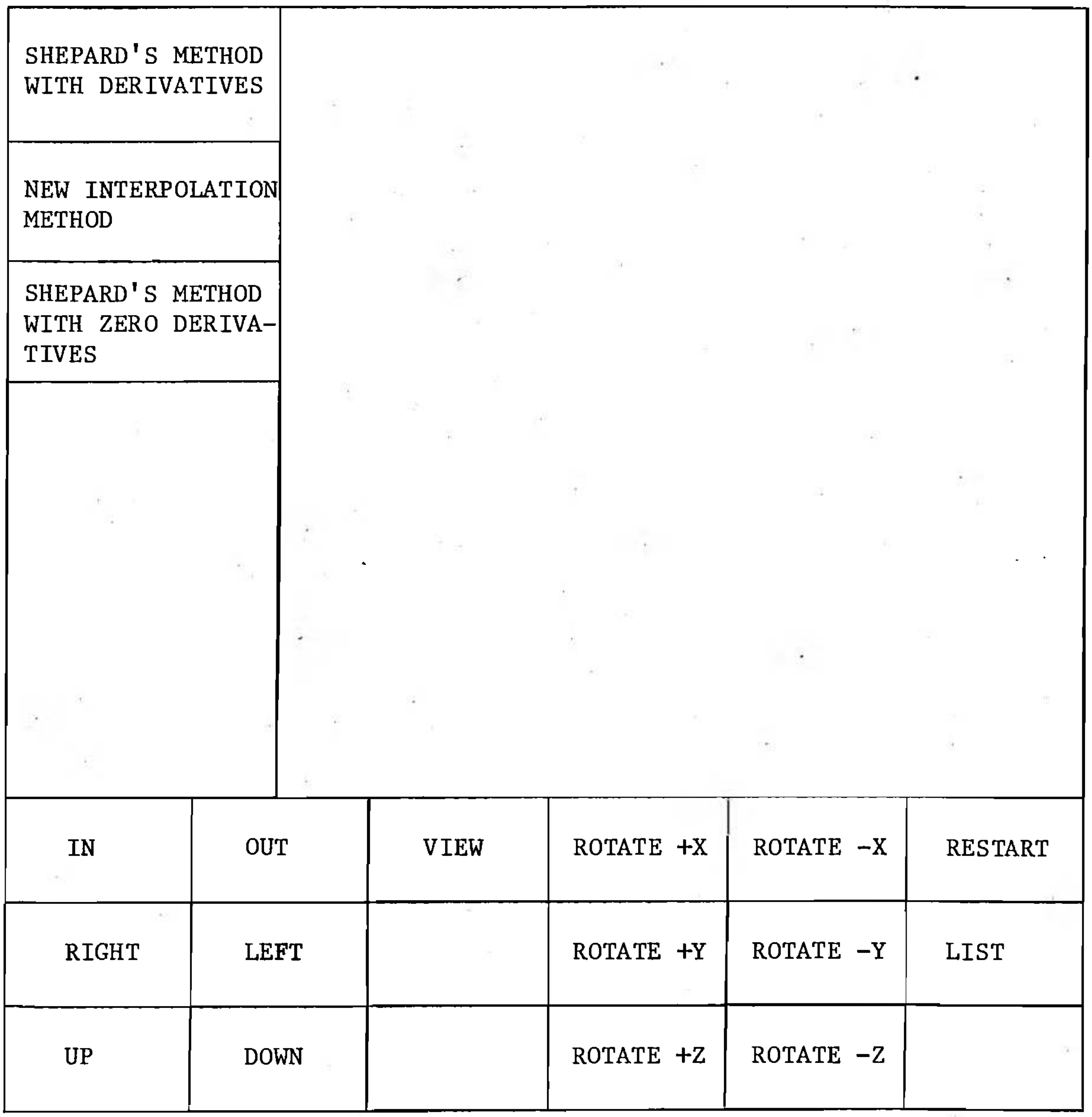


Figure A.1 IDSYS Menu

light buttons on the left allow the user to smoothly zoom in or out of the picture or to move the picture right, left, up, or down. These commands work by changing the window parameters. The window is a three dimensional enclosure of the data space to be displayed. The window and viewport define the transformation from object coordinates to screen coordinates. The parameters for the window are the center coordinates of the three dimensional enclosure and the length of the sides. We can zoom in or out of the picture by changing the length of the sides of the window. We can move the picture right, left, up or down by changing the center coordinates of the window. Using the window and viewport for the transformation is more convenient than scale and translation.

The light button VIEW allows the user to specify a particular view of the picture for display. The user must specify the position of the eye and the point at which the eye is looking. This command has proved very useful for precisely positioning the picture on the screen the same way every time. The same view of different functions and data can then be compared.

By selecting the appropriate light button from the next two columns the user can smoothly rotate the picture. The picture can be rotated clockwise or counterclockwise about the X , Y , or Z axis.

The RESTART light button starts the program over again. The LIST light button will produce a hardcopy list of the three dimensional points used to display the surface on the screen. They can be used to more accurately interpret the surface.

IDSYS keeps the three dimensional transformation matrices necessary to smoothly window or rotate the object separate. When the user has selected the light button to dynamically manipulate the object, it is only necessary to update one matrix. IDSYS also has a special transformation matrix to convert the right handed

$$[x\ y\ z]\begin{bmatrix} \\ \text{ROTATION} \\ \\ \end{bmatrix}\begin{bmatrix} \text{RIGHT-HAND} \\ \text{to} \\ \text{LEFT-HAND} \end{bmatrix}\begin{bmatrix} \\ \text{WINDOW} \\ \\ \end{bmatrix} = [x'\ y'\ z'] \quad . \qquad [A.1]$$

coordinate system of the object space into the left hand coordinate system required by the Picture System. These three transformation matrices are then concatenated together. Since matrix multiplication is not commutative, it is necessary to concatenate the matrices in the correct order. The concatenated matrix is then given to the hardware matrix arithmetic processor.

All the pictures in this thesis were photographed directly from the refresh display screen of the Picture System. A hidden line algorithm would have helped make these static pictures more clear but it would have made the real time motion unacceptably slow. A hidden line algorithm is not necessary with real time dynamic motion.

REFERENCES

[1] R. E. Barnhill, "Smooth Interpolation over Triangles," in *Computer Aided Geometric Design*, R. E. Barnhill and R. F. Riesenfeld, Ed. New York: Academic Press, Inc., 1974, pp. 45-70

[2] R. E. Barnhill and J. A. Gregory, "Compatible Smooth Interpolation in Triangles," to appear in *Journal of Approximation Theory*.

[3] S. Geary, "A Computer Graphics System to Aid the Study of Interpolants to Curved Surfaces," M.S. Thesis, University of Utah, Salt Lake City, 1973.

[4] W. J. Gordon, "Distributive Lattices and the Approximation of Multivariate Functions," in *Proceedings of the Symposium on Approximation with Special Emphasis on Splines*, I. J. Schoenberg, Ed. University of Wisconsin Press, 1969.

[5] W. J. Gordon and R. F. Riesenfeld, "B-Spline Curves and Surfaces," in *Computer Aided Gometric Design*, R. E. Barnhill and R. F. Riesenfeld, Ed. New York: Academic Press, Inc., 1974, pp. 95-126.

[6] J. A. Gregory, "Smooth Interpolation without Twist Constraints," in *Computer Aided Geometric Design*, R. E. Barnhill and R. F. Riesenfeld, Ed. New York: Academic Press, Inc., 1974, pp. 71-87.

[7] R. J. McDermott, "Graphical Representation of Surfaces over Triangles and Rectangles," in *Computer Aided Geometric Design*, R. E. Barnhill and R. F. Riesenfeld, Ed. New York: Academic Press, Inc., 1974, pp. 89-93.

[8] W. Newman and R. Sproull, *Principles of Computer Graphics*. New York: McGraw-Hill, 1973.

[9] R. F. Riesenfeld, "Applications of B-Spline Approximation to Geometric Problems of Computer Aided Design," Ph.D. dissertation, Syracuse University, Syracuse, New York, 1973.

[10] R. F. Riesenfeld, "Aspects of Modelling in Computer Aided Geometric Design," in *Proceedings of the National Computer Conference*, vol. 44. Mountvale, New Jersey: AFIPS Press, 1975.

[11] D. Shepard, "A Two Dimensional Interpolation Function for Irregularly Spaced Data," in *Proceedings 23rd Association for Computing Machinery National Conference*, 1968, pp. 517-524.

VITA

Name	Charles Christian Poeppelmeier
Birthplace	St. Louis, Missouri
Birthdate	April 28, 1948
High School	Orchard Farm High School St. Charles, Missouri
University 1965-1969	University of Missouri Columbia, Missouri
Degree 1969	B.A., University of Missouri Columbia, Missouri
Professional Organizations	Association for Computing Machinery, Special Interest Group for Computer Graphics, Special Interest Group for Design Automation, Special Interest Group on Artificial Intelligence, IEEE Computer Society.