Thin Plate Splines, Image Registration and Scientific Visualization

by

James Michael Berkley
B.Math, University of Waterloo, 1990

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

We accept this thesis as conforming to the required standard:

[redacted]

Dr. J. Barrodale, Supervisor (Department of Computer Science)

[redacted]

Dr. D.M. Miller, Departmental Member (Department of Computer Science)

[redacted]

Dr. A.G. Buckley, Departmental Member (Department of Computer Science)

[redacted]

Dr. C. Garrett, Outside Member (Department of Physics)

[redacted]

Dr. G.F. McLean, External Examiner (Department of Mechanical Engineering)

Supervisor: Dr. Ian Barrodale

Abstract

Thin plate splines are excellent tools for image registration, where exact interpolation is desired, but where there are local distortions in the data. Thin plate spline matrices are extremely ill-conditioned, but we have found that we can improve the condition number significantly through scaling onto the unit square and through ensuring a minimum separation between control points.

As with all interpolating functions, thin plate splines are subject to overshoot difficulties. In this thesis, we have investigated some techniques using scientific visualization for detecting severe interpolation overshoot.

Two practical applications of thin plate spline registration are examined, *Change Detection in Side-Scan Sonar Images* and *Survey Monument Registration.* Thin plate spline warping is very effective at registering two side-scan sonar images for comparison, in spite of severe local distortions due to sensor movement. Because of limitations in the data available, thin plate splines are not as effective at transforming between the North American Datums of 1927 and 1983 for survey monument registration.

Examiners:

[redacted]

Dr. J. Barrodale, Supervisor (Department of Computer Science)

[redacted]

Dr. D.M. Miller, Departmental Member (Department of Computer Science)

[redacted]

Dr. A.G. Buckley, Departmental Member (Department of Computer Science)

[redacted]

Dr. C. Garrett, Outside Member (Department of Physics)

[redacted]

Dr. G. F. McLean, External Examiner (Department of Mechanical Engineering)

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank Dr. Barrodale for his direction and patience, and to thank Sheila for all her encouragement. I could not have finished this thesis without their continual support and assistance.

My thanks also go to Dr. McLean for his help in producing the images for this thesis and for his encouragement when I thought I would never finish the thing. Thanks, Ged!

Most of all, I would like to thank Lorri for putting up with me all these years. Dear, you finally got your PHT.

Chapter 1

Introduction

Imagine an infinite thin metal plate, smoothly deformed by specific deflections at a number of points. When a thin metal plate assumes a "least bent"[1] configuration [Bookstein 89], the equation which describes the surface of the plate is known as a thin plate spline. It has been shown that thin plate splines are useful in the interpolation of scattered data [Franke 81] and also in image registration [Goshtasby 88, Skea & Barrodale 90]. However, matrices associated with thin plate splines are extremely ill-conditioned. The goal of this thesis is to investigate ways of improving the condition number of TPS[2] matrices, to explore scientific visual-

[1] A "least bent" surface is one which has assumed a shape of minimum physical bending energy subject to fixed point constraints.

[2] TPS - Thin Plate Spline

ization as a method for dealing with interpolation difficulties, and to examine the application of thin plate splines to two practical registration problems.

The process of overlaying two images of the same scene for the purpose of comparison is a common task in image analysis. Often, there are geometric differences between the images, so that one image must be deformed to lay over the other image. The technique of warping and overlaying the images is called *image registration* or *rubber sheeting.*

A typical registration method is to define *control points*, usually image features, where the two images are supposed to match, and then use the *control point pairs* (a pair of points, one from the each image) to determine some mapping or warping function that can deform one image to overlay the other. One image is defined (often arbitrarily) as the *reference* image. The other image, which will be warped into the shape of the reference image, is designated as the *sensed* image.

Figure 1.1 is a stylized representation of image registration. The reference image represents one view of a scene, while the sensed image represents another view of the same scene, but with some distortion relative to the first image. The X's represent control points where the two images should match. Note that there is a new feature in the sensed image, but this difference may not be immediately apparent when comparing the original images. The new feature is more obvious when we compare the original reference image with the warped sensed image because the two images

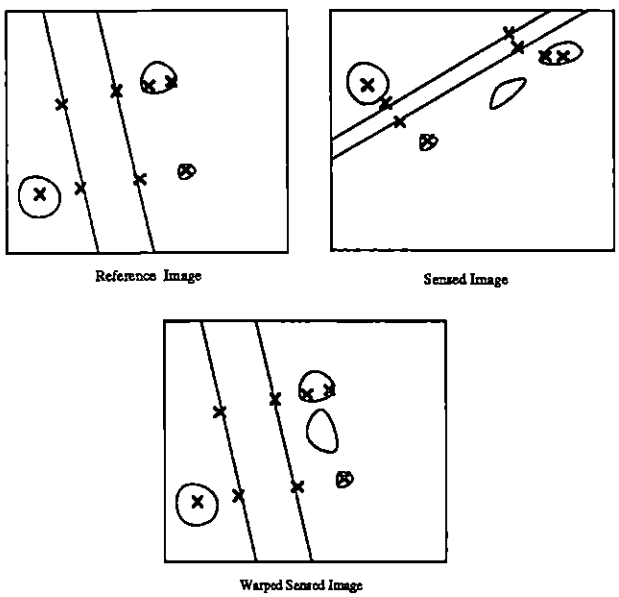

Figure 1.1: Image Warping

look more like the same scene; we can visually compare the images more easily. Also note that corresponding control points in the reference and warped images are now in the same locations.

A warping or registration function maps points (x_s, y_s) from the sensed image to points (x_r, y_r) in the reference image. Usually, two functions are used; one maps (x_s, y_s) values in the sensed image to x_r values in the reference image and the other maps to the y_r values:

$$\begin{aligned} x_r &= f_x(x_s, y_s) \\ y_r &= f_y(x_s, y_s) \end{aligned}$$

In other words, when we register the sensed image to fit the reference image, we map (x_s, y_s) points to (x_r, y_r) points. We determine the new x_r coordinates using $f_x(x_s, y_s)$ and the new y_r coordinates using $f_y(x_s, y_s)$.

Defining two functions f_x and f_y from control point sets is equivalent to fitting two surfaces f_x and f_y to the sets $\{(x_{si}, y_{si}, x_{ri})\}$ and $\{(x_{si}, y_{si}, y_{ri})\}$, respectively. In this thesis, we use thin plate spline surfaces to develop f_x and f_y interpolating functions. For the most part, we will discuss the development of an f_x interpolation surface with the implicit assumption that the f_y surface is developed in the same way.

Often, image registration is expressed in terms of the difference between coordinate values rather than in terms of the new values themselves. The interpolation surfaces f_x and f_y are then defined as fitting the sets $\{(x_{si}, y_{si}, x_{ri} - x_{si})\}$ and $\{(x_{si}, y_{si}, y_{ri} - y_{si})\}$ respectively. In this thesis, we will be using shift x values $(x_{ri} - x_{si})$ and shift y values $(y_{ri} - y_{si})$ for registration.

Surface fitting methods can be classified as *local* or *global* and as *approximating* or *interpolating* [Schumaker 76]. With global methods, the entire set of data values is used to determine the surface. With local methods, the data area is divided into small regions and a surface is fitted to each region using the subset of data values contained in the region. An approximation method fits a given kind of surface (often polynomial) in such a way that some measure of the total error between the data

values and the surface is minimized. An interpolation method produces a surface which is expected to pass through every data value exactly.

Local surface fitting methods, either approximating or interpolating, could be used in image registration, but there would be problems. For example, local methods may have difficulty if there are large featureless areas in the images (hence no control points). Ensuring continuity between regions can also be difficult. No doubt, if these problems can be overcome, local methods have promise as registration techniques (see [Dewhurst 90]), but we decided to concentrate on global methods for this thesis.

Traditionally, image registration has employed global approximation using polynomials. In our applications, data inaccuracies and image acquisition difficulties lead to significant local distortions, but the least-squares technique (which is used to obtain polynomial coefficients) averages local irregularities over the entire image, so that images warped with global approximation using polynomials may be very contorted and may not even approximate the original image. In addition, our side-scan sonar applications require an interpolating registration function because the features we are looking for may be extremely small. An approximation method may not be accurate enough to register small features correctly.

Figure 1.2 gives one example of the ability of TPS warping to handle image distortions versus the wild image that results from warping with a polynomial of degree 2. In the polynomial warped image, note that the wreck has been duplicated

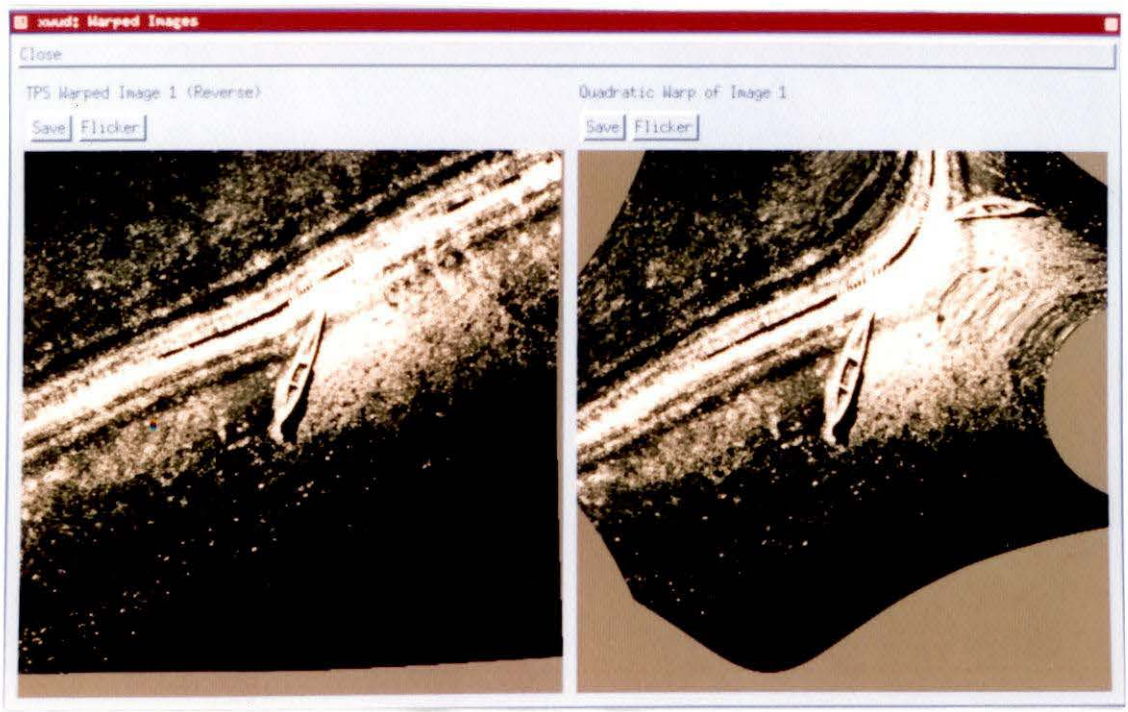

Figure 1.2: Polynomial warping functions are not always good registration functions.

in the upper right corner of the image. This example demonstrates the need for an interpolating function which can handle local distortions without the oscillatory behaviour which is common to polynomial functions.

Thin plate splines are global interpolation functions that can be used in expressing image registration as a surface fitting problem. Two TPS surfaces are required to represent two functions which project x and y values from one image onto the other, mapping control points exactly, and characterising local distortions [Goshtasby 88]. The TPS equation for the x surface is:

$$F_x(x,y) = a_0 + a_1 x + a_2 y + \sum_{i=1}^{n} \frac{1}{2} b_i r_i^2 \log r_i^2$$

where r_i is the normal Euclidean distance, $\sqrt{(x-x_{si})^2 + (y-y_{si})^2}$, between the point (x,y) and the i^{th} control point in the sensed image, (x_{si}, y_{si}).

A key term in the TPS equation is $U(r) = r^2 \log r^2$. Figure 1.3 gives the shape[3] of $-U(r)$ on the square $[-1,1] \times [-1,1]$. The maximum value of $-U(r)$ is e^{-1} and occurs where $r = e^{-\frac{1}{2}}$. $U(r)$ is zero where $r = 1$. At $r = 0$, $U(r)$ is undefined; however $\lim_{r \to 0} U(r) = 0$, so we define $U(0) = 0$.

A thin plate spline contains a linear combination of $U(r)$ terms. For every control point, we have $U(r_i) = r_i^2 \log r_i^2$. Each of these terms represents a $U(r)$ surface which

[3] As in [Bookstein 89], we show positive values of $-r^2 \log r^2$ rather than $r^2 \log r^2$ to simplify interpretation of the surface.

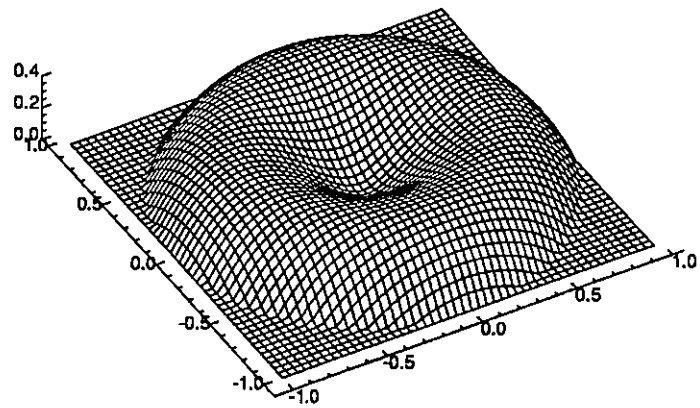

Figure 1.3: $-U(r) = -r^2 \log r^2$

is centred at (x_{si}, y_{si}). Figure 1.4 shows a linear combination of four $U(r)$ terms centred at $(0,1)$, $(-1,0)$, $(0,-1)$, and $(1,0)$.

The linear combination of a plane with $U(r)$ terms results in a thin plate spline surface. Figure 1.5 shows a thin plate spline with six control points. In the example, the points $(0,0)$, $(0,1)$, $(1,1)$ and $(1,0)$ are all shifted by zero, and the points $(0.4,0.5)$ and $(0.6,0.5)$ are shifted by -0.2 and $+0.2$ respectively.

In our applications, we have found thin plate splines to be useful registration functions for problems with local distortions. Unfortunately, TPS matrices are extremely ill-conditioned, which may lead to instability. In Chapter 2, we will examine the theory of thin plate splines, including definitions for TPS matrices and

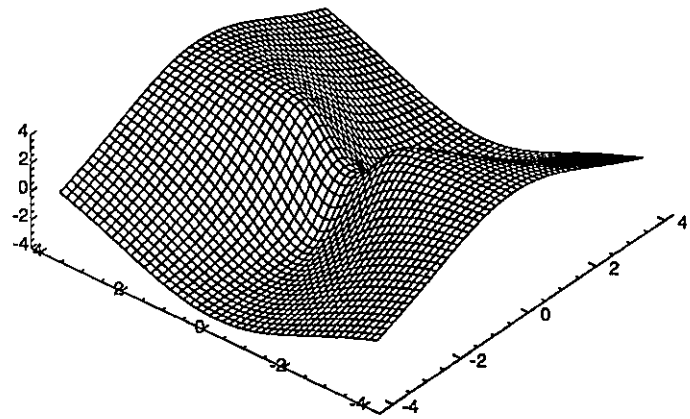

Figure 1.4: $U(r_{(0,1)}) - U(r_{(-1,0)}) + U(r_{(0,-1)}) - U(r_{(1,0)})$

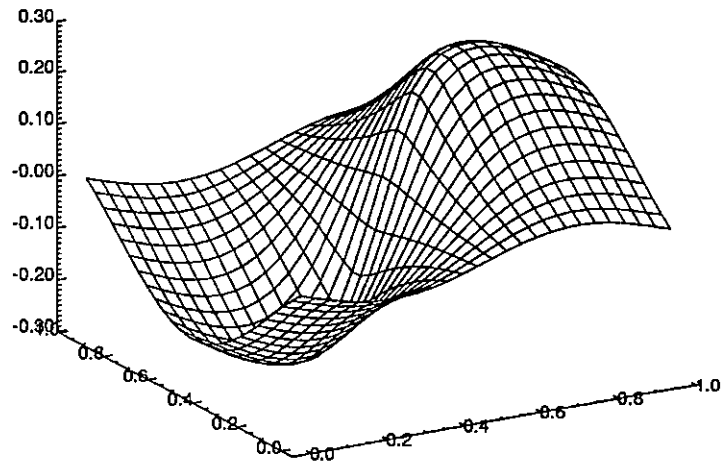

Figure 1.5: $U(r_{(0,1)}) - U(r_{(-1,0)}) + U(r_{(0,-1)}) - U(r_{(1,0)})$

coefficients, an examination of the ill-conditioning problem and some methods for improving the condition number by translation and scaling.

The fitting of smooth curves or surfaces to points which are close together, but have significantly different function values is a fundamental problem in curve fitting by interpolation. From the *Mean-Value Theorem* [Adams 86], we know that if we constrain a smooth curve to pass through two points, then the slope of the curve at some position between the points will be equal to the slope of the secant line through the points. If the two points are close enough together and the function values are sufficiently different, then the slope of the curve may be very large. In such cases, continuity requirements may force the interpolating curve to overshoot the data points, resulting in interpolation values which are outside the range of data point function values. If the interpolation curve is sampled near the two close points, then the resulting function estimate may fall far outside the range of data point function values.

This overshoot phenomena is not unique to thin plate splines; it could occur wherever interpolation is used. Figure 1.6 gives an illustration of overshoot in a one dimensional spline interpolation. The curve must pass smoothly through the points, but because two of the points are close together and have quite different function values, the curve has to have a steep slope between those two points. In order to remain smooth, the curve overshoots the upper point.

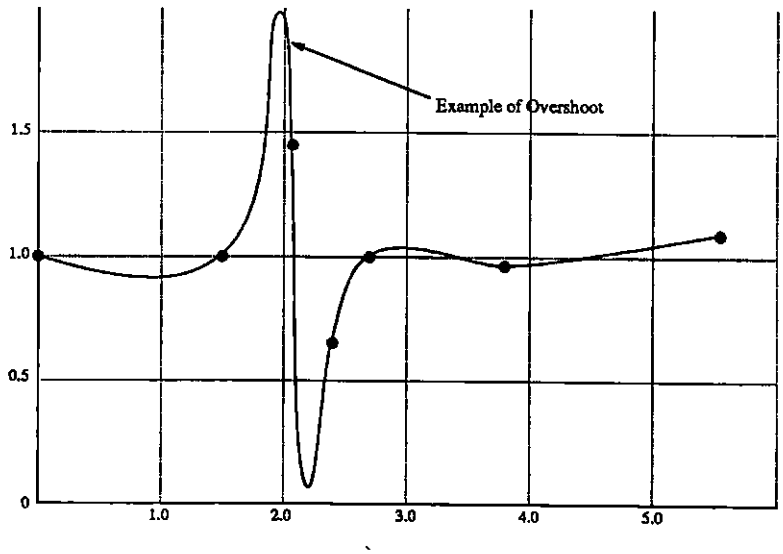

Figure 1.6: Interpolation curves may overshoot control point function values.

In many interpolation applications, moderate overshoot is not a problem. Even a curve with severe overshoot may be a good representation of the underlying data. With image registration, this is not the case because we are using interpolation surfaces to represent the amounts by which a point should be shifted in the x and y directions to properly register two images. Consider a surface using some interpolating function for registration shifts in the x direction and a control point on this surface, x_c, which is near a position of severe interpolation overshoot. Let the x value of x_c be 10cm and the shift at the control point be 1cm in the positive x direction. Consider another point, x', which has an x value of 9.9cm, and which lies

near the peak of the interpolation overshoot so that the interpolated shift value of x' is 5cm, say. This means that x' has overshot x_c by 4cm. With many applications of interpolation, this overshoot is quite acceptable, but when we interpret these values as image registration shifts, we find that according to the interpolation surface the point x' should be shifted 3.9cm **past** x_c because of the overshoot. This would introduce a fold in the image and is clearly not acceptable.

In our research, we have found that control points which are too close together may lead to unreasonable results due to interpolation overshoot, but practical problems indicated that a strict absolute threshold for deciding when a pair of control points are too close is inappropriate. To deal with this problem, we turned to scientific visualization, hoping to acquire insight and a visual method for detecting undesirable behaviour in TPS surfaces. Chapter 3 describes some of the background of scientific visualization, why we hoped it would help us with errors due to interpolation and how we applied scientific visualization to TPS problems.

Chapter 4 describes the application of TPS registration to two real world problems, *Change Detection in Side-Scan Sonar Images* and *Survey Monument Registration.*

Side-scan sonar images contain many local irregularities due to water turbulence, thermal variations, etc. These local distortions make it difficult to register side-scan sonar images using simple warping functions. [Skea & Barrodale 90] indicate that

thin plate splines make good warping functions for side-scan sonar image registration and are able to handle the local distortions. In Section 4.2, we will describe the application of thin plate splines to side-scan sonar.

Two of the main geodetic datums used in the Province of British Columbia are the North American Datum of 1983, NAD83 and the North American Datum of 1927, NAD27. Due to advances in technology, NAD83 is much more accurate, but almost all existing topographic mapping and survey control networks are based on the NAD27 coordinate system. In order to take advantage of the greater accuracy of NAD83 without redoing work based on NAD27, the Province would like to be able to warp data based on NAD27 to the NAD83 coordinate system. This requirement is very similar to image registration and in fact the Province is currently using local approximation methods to warp NAD27 data to NAD83 [Dewhurst 90]. One problem with the Province's current registration method is that the positions of the NAD survey monuments are only warped approximately when they should be warped exactly. The need for exact registration implies an interpolation technique and since NAD27 data contains many local distortions and inaccuracies we suggested survey point registration using thin plate splines. Section 4.1 describes the work we performed and the problems we encountered in developing this application of thin plate splines.

Chapter 2

Theory of Thin Plate Splines

Spline functions are widely used in fitting smooth curves or surfaces to data. The choice of spline function depends upon the degree of smoothness desired and the kind of fit required (for examples see [Cheney & Kincaid 85, Skea & Barrodale 90]). Continuity or smoothness requirements stipulate the behaviour of the spline at control points and are usually defined as follows:[1]

C0 The spline is continuous at control points.

C1 The spline is differentiable at control points.

C2 The spline and its first and second derivatives are continuous at control points.

[1] From lecture notes of a course in Numerical Analysis given by R.H. Bartels at the University of Waterloo, 1988.

Spline functions can also be approximating or interpolating. With an approximating spline, the objective is to fit a particular kind of spline such that some measure of the total error at the control points is minimized. With an interpolating spline, the objective is to fit a curve that passes through each control point exactly and satisfies certain continuity requirements.

In our warping applications, we need to fit surfaces to x and y shift data. We have assumed that our control points are relatively accurate (within the given image), and so we have chosen an interpolating spline. We will also require C1 continuity to ensure a continuous, smooth surface.

Thin plate splines are good candidates for our warping applications because they are interpolating splines (TPS surfaces pass through each control point) with C1 continuity (thin plate splines are continuous and have continuous first partial derivatives). Physically, a thin plate spline corresponds to a thin metal plate slightly deformed by a number of point loads. The metal plate bends smoothly in a way that minimizes the physical bending energy [Bookstein 89]. The thin plate spline equation which represents this metal plate surface can be written as follows [Harder & Desmarais 72]:

$$\begin{aligned} f(x,y) &= a_0 + a_1 x + a_2 y + \sum_{i=1}^{n} \frac{1}{2} b_i r_i^2 \log r_i^2 \\ &\text{where } r_i^2 = (x - x_{si})^2 + (y - y_{si})^2 \end{aligned} \tag{2.1}$$

An (x_{si}, y_{si}) coordinate corresponds to the position of a point load where the metal plate is fixed at some value, $f(x_{si}, y_{si})$. For our registration application, (x_{si}, y_{si}) coordinates correspond to the positions of the sensed image control points. We need two TPS surfaces for image registration, one for the x shift values and another for the y values. At each (x_{si}, y_{si}) position, the x TPS surface is fixed at the x shift value $x_{ri} - x_{si}$ and the y TPS surface is fixed at $y_{ri} - y_{si}$.

The control points give us n equations defining each TPS surface. To obtain an $(n+3) \times (n+3)$ system of equations and solve for the a_j and b_i coefficients ($n+3$ unknowns), we add the following three additional constraints:

$$\sum_{i=1}^{n} b_i = 0 \qquad \sum_{i=1}^{n} x_{si} b_i = 0 \qquad \sum_{i=1}^{n} y_{si} b_i = 0$$

These additional constraints serve to suppress terms which grow faster than linear far away from the interpolation points [Franke 81]. As motivation, the following example shows that because of the above constraints, $\sum_{i=1}^{n} b_i r_i^2$ is a constant (this fact is used in the derivation of equation 2.6).

$$\begin{aligned}
\sum_{i=1}^{n} b_i r_i^2 &= \sum_{i=1}^{n} b_i (x - x_{si})^2 + \sum_{i=1}^{n} b_i (y - y_{si})^2 \\
&= \sum_{i=1}^{n} b_i x^2 + \sum_{i=1}^{n} b_i x_{si}^2 - 2 \sum_{i=1}^{n} b_i x x_{si} + \\
&\quad \sum_{i=1}^{n} b_i y^2 + \sum_{i=1}^{n} b_i y_{si}^2 - 2 \sum_{i=1}^{n} b_i y y_{si}
\end{aligned}$$

$$
\begin{aligned}
&= x^2 \sum_{i=1}^{n} b_i + \sum_{i=1}^{n} b_i x_{si}^2 - 2x \sum_{i=1}^{n} b_i x_{si} + \\
&\quad y^2 \sum_{i=1}^{n} b_i + \sum_{i=1}^{n} b_i y_{si}^2 - 2y \sum_{i=1}^{n} b_i y_{si} \\
&= 0 + \sum_{i=1}^{n} b_i x_{si}^2 - 0 + 0 + \sum_{i=1}^{n} b_i y_{si}^2 - 0 \\
&= \sum_{i=1}^{n} b_i (x_{si}^2 + y_{si}^2)
\end{aligned}
$$

The entire TPS system can be represented by a symmetric, non-positive definite matrix equation

$$
\begin{bmatrix}
0 & 0 & 0 & 1 & 1 & \dots & 1 \\
0 & 0 & 0 & x_{s1} & x_{s2} & \dots & x_{sn} \\
0 & 0 & 0 & y_{s1} & y_{s2} & \dots & y_{sn} \\
1 & x_{s1} & y_{s1} & 0 & r_{21}^2 \log r_{21}^2 & \dots & r_{n1}^2 \log r_{n1}^2 \\
1 & x_{s2} & y_{s2} & r_{12}^2 \log r_{12}^2 & 0 & \dots & r_{n2}^2 \log r_{n2}^2 \\
. & . & . & . & . & \dots & . \\
. & . & . & . & . & \dots & . \\
. & . & . & . & . & \dots & . \\
1 & x_{sn} & y_{sn} & r_{1n}^2 \log r_{1n}^2 & r_{2n}^2 \log r_{2n}^2 & \dots & 0
\end{bmatrix}
\begin{bmatrix}
a_0 \\ a_1 \\ a_2 \\ \frac{b_1}{2} \\ \frac{b_2}{2} \\ . \\ . \\ . \\ \frac{b_n}{2}
\end{bmatrix}
=
\begin{bmatrix}
0 \\ 0 \\ 0 \\ x_{r1} - x_{s1} \\ x_{r2} - x_{s2} \\ . \\ . \\ . \\ x_{rn} - x_{sn}
\end{bmatrix}
\tag{2.2}
$$

where (x_{si}, y_{si}) are the coordinates of the i^{th} control point in the *sensed* image, x_{ri} is the x coordinate of the i^{th} control point in the *reference* image and r_{ij} is the distance between the i^{th} and j^{th} control points in the sensed image, $r_{ij} = \sqrt{(x_{si} - x_{sj})^2 + (y_{si} - y_{sj})^2}$.

The first point to make about TPS matrices (equation 2.2) is that they are extremely ill-conditioned [Skea & Barrodale 90, Sibson & Stone 91], which may cause problems with instability in some situations. Although it is known that TPS matri-

ces are ill-conditioned, there has not been much work on the reasons. Most papers assume that the main cause of instability is control points that are too close together, but there is no definition for "too close." Section 2.1 is based on a study of TPS condition numbers that examines ill-conditioning in TPS matrices.

It has been noted [Franke 81] that the condition number of a TPS matrix may be substantially improved by scaling the problem onto the $[0,1] \times [0,1]$ unit square. Unfortunately, thin plate splines are *not* invariant with respect to scaling, which complicates the process of translating and scaling to the unit square and then using coefficients from the scaled TPS equation in the original, unscaled problem. In Section 2.2, we will show that thin plate splines are invariant with respect to translation, and we will outline a translation and scaling method which seems to lower the condition number of TPS matrices.

2.1 Examining the Condition Number Problem

The condition number is a measure of the sensitivity of a linear system to small perturbations in the input parameters. If the condition number is large, small changes to the input may result in very large differences in the solution. This property, called ill-conditioning, may make it difficult or even impossible to obtain accurate solutions (see, for example, [Golub & Van Loan 89]).

The condition number, $\kappa(M)$, is defined as:

$$\kappa(M) = \| M \| \| M^{-1} \|$$

For all the results in this thesis, we calculate the condition number using the infinity matrix norm , which is defined as:

$$\| M \|_\infty = \max_{x \neq 0} \frac{\| Mx \|_\infty}{\| x \|_\infty}$$

$$\text{where} \quad \| v \|_\infty = \max_{\forall i} \| v_i \|$$

Consider a problem in the form $Mx = b$, such as Equation 2.2 above. The condition number is a good measure of the sensitivity for such a system because it can be shown that the relative perturbation in x, $\frac{\|\delta x\|}{\|x\|}$, is bounded by the relative perturbation in b, $\frac{\|\delta b\|}{\|b\|}$, and the condition number, $\kappa(M)$:

$$\frac{\| \delta x \|}{\| x \|} \leq \kappa(M) \frac{\| \delta b \|}{\| b \|} \tag{2.3}$$

This means that if $\kappa(M)$ is large, then $\frac{\|\delta x\|}{\|x\|}$ may be much larger than $\frac{\|\delta b\|}{\|b\|}$.

It can also be shown that the error in x relative to $x + \delta x$ is bounded by the relative perturbation in M, $\frac{\|\delta M\|}{\|M\|}$, and the condition number of M, $\kappa(M)$:

$$\frac{\| \delta x \|}{\| x + \delta x \|} \leq \kappa(M) \frac{\| \delta M \|}{\| M \|} \tag{2.4}$$

If $\kappa(M)$ is small, say 10^2, then the problem is well-conditioned and can be safely solved in single precision arithmetic. If $\kappa(M)$ is relatively large, even double or quad precision arithmetic may not yield sufficient accuracy for certain values of b.

Thin plate spline matrices (Equation 2.2) often have huge condition numbers, even for a small number of control points [Skea & Barrodale 90, Sibson & Stone 91]. For TPS matrices, the implication of Equations 2.3 and 2.4 is that even if we know our control points (which form the b vector) to a high degree of accuracy, we may not be able to obtain accurate TPS coefficients (from the x vector) because of their high condition number.

We performed some experiments to examine the magnitude of TPS matrix condition numbers and to determine ways of reducing condition numbers.[2] The following sections describe the experiments and our results.

2.1.1 Thin Plate Spline Matrix Condition Number Experiments

The first step in our project was to examine TPS condition numbers through random testing. We designed a number of programs which would create random TPS matrices and calculate the condition number. With these tests we looked for the magnitude of the condition number, tried to find geometric control point patterns which increased the condition number and examined the effect of scaling.

[2]These experiments were performed as the main part of a CSC540 project in Fall 1990.

Once we had some basic results, we went on to look at the actual matrix factorization process. Our original program used a textbook LU factorization algorithm with partial pivoting [Watkins 90]. In the last part of our project, we compared the results of LU factorization on thin plate spline matrices with other factorization algorithms.

Random Condition Number Test Methods

In the random condition number tests, we sought to get an overall picture of TPS condition numbers. We were trying to observe best, worst and median values for various numbers of randomly selected control points. We also wanted to examine the effect of scaling on the condition numbers by adjusting the range of control points. See Table 2.1 for a summary of the numerical results.

The main program, `thtrials`, was used to generate random TPS matrices and check their condition numbers. The first step in `thtrials` is to generate sets of random x and y coordinates for control points. The number, range and granularity of the control points are variable. By controlling the range, we can tell if having all control points in the range $[0,1] \times [0,1]$ makes a difference on the condition number. Granularity means the number of intervals that the range is divided into. By controlling the granularity we can mimic an image registration problem where pixel values occur at discrete intervals.

An important point to make about the `thtrials` program is that the random number series used is a repeatable series. This means that each set of random control points for matrices of a given size in the range $[1000, 10000]$ corresponds to a scaled and translated random control point set generated for the range $[0, 1]$. Thus the condition numbers for the TPS matrices in the two ranges can be compared matrix by matrix.

After generating the random control points, `thtrials` creates a TPS matrix from the control point set, simultaneously checking to see if there is a pair of control points which are too close together. There are two threshold values for measuring "closeness". If a pair of control points are closer than the lower bound, then `thtrials` rejects the matrix and does not include it in condition number statistics. If a pair is closer than the upper bound, but not the lower bound, then `thtrials` generates a warning message. Sample values might be 10^{-7} for an upper bound and 10^{-9} for a lower bound. These threshold values are adjustable and their purpose is to allow us to find a lower bound on how close two control points could be without unduly affecting the condition number. By altering the thresholds, we were able to establish a lower bound on "closeness" which seemed to preclude very large condition numbers.

After creating the matrix, `thtrials` performs an LU factorization with partial pivoting. In later stages of the project, we replaced the LU factorization with alter-

nate methods, seeking to verify that we were factoring the TPS matrices correctly.

Finally, `thtrials` solves for the inverse matrix (using standard forward and backward substitution on the elementary matrices) and calculates the infinity norm for the matrix and matrix inverse.

For each set of control points, `thtrials` prints out the condition numbers of the matrix. By examining the output, we could select TPS matrices with condition numbers that were abnormally high or abnormally low and analyze them using the `gnuplot`[3] and `vectorin`[4] programs, looking for patterns which could lead to better or worse condition numbers.

Conclusions from Random Condition Number Tests

Through this random testing, we were able to come to a number of interesting conclusions. First of all, we verified the result on scaling in [Franke 81], that is, we found that on average, given the same granularity, the condition number of TPS matrices for control points in the range $[0,1] \times [0,1]$ was considerably lower than the condition number for matrices with control points in a larger range. Typical reductions were from 10^{19} to 10^5 and from 10^{20} to 10^7 (see Table 2.1). From this

[3] `gnuplot` is a very flexible public domain plotting package we used to examine control point patterns.

[4] `vectorin` was designed to give an extensive analysis of TPS matrices.

Range	Number of Points	Number of Trials	Condition Number Mean	Minimum	Maximum	Median
[0,1000]	10	200	4.5×10^{18}	6.6×10^{17}	9.1×10^{18}	4.6×10^{18}
	25	200	1.5×10^{19}	3.8×10^{18}	2.4×10^{19}	1.5×10^{19}
	75	200	5.3×10^{19}	3.9×10^{19}	7.0×10^{19}	5.3×10^{19}
	100	200	7.2×10^{19}	5.5×10^{19}	8.8×10^{19}	7.2×10^{19}
	150	200	1.1×10^{20}	9.4×10^{19}	1.3×10^{20}	1.1×10^{20}
	300	50	2.2×10^{20}	2.0×10^{20}	2.5×10^{20}	2.2×10^{20}
	400	50	3.0×10^{20}	2.8×10^{20}	3.8×10^{20}	3.0×10^{20}
[0,1]	10	200	3.1×10^{3}	4.3×10^{2}	6.5×10^{4}	1.4×10^{3}
	25	200	3.9×10^{4}	5.7×10^{3}	3.4×10^{5}	2.2×10^{4}
	75	200	7.8×10^{5}	1.1×10^{5}	1.3×10^{7}	4.6×10^{5}
	100	200	2.0×10^{6}	2.3×10^{4}	2.3×10^{7}	9.0×10^{5}
	300	50	2.8×10^{7}	4.4×10^{6}	8.6×10^{7}	2.1×10^{7}
	150	200	4.7×10^{6}	8.5×10^{5}	3.9×10^{7}	2.7×10^{6}
	400	50	5.8×10^{7}	1.5×10^{7}	1.4×10^{8}	5.0×10^{7}

Table 2.1: Condition Number Trials

we can conclude that we should scale our problems onto the unit square before calculating and solving the TPS matrix.

Next, we found that if a pair of control points was too close together then the condition number for the TPS matrix would be extremely large. We were seeing the condition number increase by a factor of 10^{20} if two of the control points were too close. By adjusting the granularity and upper/lower thresholds of `thtrials` we were able to put an upper bound on the magnitude of the condition number for a given number of control points in a given range, inferring that we can improve stability in our registration problems by putting a limit on how close a pair of control points

can be. The results in Table 2.1 reflect condition numbers for TPS matrices that have no control point pairs closer together than a lower bound of 10^{-9}.

We were unable to find any geometric control point patterns which consistently gave large condition numbers. Having control points on a line or an arc did not affect the condition number. We did find that a small cluster of three control points may affect the condition number by a slight amount (less than an order of magnitude), even if the points are separated by more than the upper threshold we have determined.

When we changed our factorization routines to use other methods, we found that the alternate methods were much more efficient than our textbook LU method, but were no more accurate when calculating the TPS matrix inverse or verifying survey project control point shifts. This reassured us that our results were not affected by a poor factorization algorithm. We tried two alternative routines; the first is IBM's Engineering Scientific Subroutine Library routine called `DGEF`, and the other was the LINPACK routine, `DSPCO`. `DGEF` was optimized for the IBM RS/6000 machine that we were using and was approximately five times faster than our original LU routine. `DSPCO` is specifically designed for symmetric, non-positive definite matrices, and uses a packed matrix format to cut memory requirements almost by half. `DSPCO` was not as fast as `DGEF`, but it was still four times faster than our original routine. Since it uses less memory and is more portable to other platforms, we decided to

use DSPCO as our standard TPS matrix factorization routine.

Both DGEF and DSPCO also offer inexpensive condition number estimates. We found that these estimates were generally of the same magnitude as the actual condition number, so the estimates may be useful as simple confidence checks when factoring TPS matrices.

2.2 Translation and Scaling

How are thin plate splines affected by translation and scaling? This is an important question because if we can deal with translation and scaling without changing the problem, we can improve the condition number of a TPS matrix by mapping control points to the unit square [Franke 81].

We will begin by considering translation of the TPS equation:

$$\begin{aligned} f(x,y) &= a_0 + a_1 x + a_2 y + \sum_{i=1}^{n} \tfrac{1}{2} b_i r_i^2 \log r_i^2 \\ &= x_r - x_s \end{aligned}$$

Obviously $f(x,y) = x_r - x_s$ is independent of translation, but what happens to $a_0 + a_1 x + a_2 y + \sum_{i=1}^{n} \frac{1}{2} b_i r_i^2 \log r_i^2$? Consider the effect on r_i^2 of a translation by t_x in the x direction and t_y in the y direction:

$$
\begin{aligned}
x' &= x + t_x \\
x'_i &= x_i + t_x \\
y' &= y + t_y \\
y'_i &= y_i + t_y \\
\\
r'^2_i &= (x' - x'_i)^2 + (y' - y'_i)^2 \\
&= ((x + t_x) - (x_{si} + t_x))^2 + ((y + t_y) - (y_{si} + t_y))^2 \\
&= (x - x_{si})^2 + (y - y_{si})^2 \\
&= r_i^2
\end{aligned}
$$

This shows that r_i^2 is independent of translation, so $\sum_{i=1}^{n} \frac{1}{2} b_i r_i^2 \log r_i^2$ is also unaffected by translation. What effect does translation have on a_0, a_1 and a_2? We will set $a'_1 = a_1$, $a'_2 = a_2$ and $b'_i = b_i$, then consider again translation by t_x in the x direction and t_y in the y direction:

$$
\begin{aligned}
x'_r - x'_s &= a'_0 + a'_1 x' + a'_2 y' + \sum_{i=1}^{n} \frac{1}{2} b'_i r'^2_i \log r'^2_i \\
x_r - x_s &= a'_0 + a_1(x + t_x) + a_2(y + t_y) + \sum_{i=1}^{n} \frac{1}{2} b_i r_i^2 \log r_i^2 \\
x_r - x_s &= (a'_0 + a_1 t_x + a_2 t_y - a_0) + (a_0 + a_1 x + a_2 y + \sum_{i=1}^{n} \frac{1}{2} b_i r_i^2 \log r_i^2) \\
0 &= a'_0 + a_1 t_x + a_2 t_y - a_0 \\
a'_0 &= a_0 - a_1 t_x - a_2 t_y
\end{aligned}
$$

Thus, translation only causes a change in the constant term. Our next task will be to find out how scaling affects the TPS equation. Other authors (for example, [Franke 81]) have stated that thin plate splines are invariant with respect to scaling. This is, in fact, not the case, but we will show an empirically determined method for scaling TPS coefficients.

Consider the following example from [Goshtasby 88], where the function $f(x, y)$ is determined by the given points and function values:

i	1	2	3	4	5	6	7
x_{si}	1	1	7	11	16	20	20
y_{si}	1	20	15	8	12	1	20
$f(x_{si}, y_{si})$	10	10	25	15	20	10	10

We can calculate the TPS coefficients for this problem by solving the TPS matrix equation 2.2 with the $x_{ri} - x_{si}$ elements on the right hand side replaced by the $f(x_{si}, y_{si})$ values as given above.

We know from [Franke 81] and from our condition number experiments that the TPS matrix will be better conditioned if we scale the control points onto the unit square. Our goal is to empirically determine the relationship between the TPS coefficients for the problem above and the TPS coefficients for the same problem scaled to the unit square.

Table 2.2[5] gives the TPS coefficients for the above problem and for a version of the problem where the (x, y) values in the TPS matrix are scaled onto the unit square by a scaling factor of $\max(x, y) = 20$, that is $x'_j = \frac{x_j}{20}$ and $y'_j = \frac{y_j}{20}$. Notice the improvement in the condition number. The right hand sides, $f(x_{si}, y_{si})$, in the scaled and unscaled versions of the problem are the same.

From the ratios, we see that a_1 and a_2 for the unscaled problem are equal to a'_1 and a'_2 for the scaled problem divided by the scale factor of 20. The unscaled

[5]The table was calculated using Maple to an accuracy of twenty decimal digits, but only ten digits are shown.

	Unscaled	Scaled	Ratio
a_0	61.621894007	21.387757215	0.3471
a_1	0.078924094	1.578481882	20.0000
a_2	-0.323199305	-6.463986100	20.0000
b_1	0.000066925	0.026770004	400.0000
b_2	-0.031282794	-12.513117778	400.0000
b_3	0.069504640	27.801855857	400.0000
b_4	-0.049783215	-19.913286023	400.0000
b_5	0.034397535	13.759013939	400.0000
b_6	-0.001398762	-0.559504664	400.0000
b_7	-0.021504328	-8.601731336	400.0000
$\kappa(M)$	8.4×10^7	57	

Table 2.2: Ratio of coefficients between original and scaled problem

b_i coefficients can be obtained by dividing the scaled b_i' coefficients by the scale factor squared, $20^2 = 400$. Therefore, most of the coefficients for the unscaled TPS equation can be calculated from the corresponding coefficients in the scaled TPS problem.

The formula for calculating the unscaled a_0 coefficient from the scaled coefficients is not a simple ratio of a_0' with the scaling factor, but since we can calculate the unscaled coefficients, a_1, a_2 and b_i from the scaled coefficients, we can calculate a_0 by subtracting the last $n+2$ terms of the TPS equation from $f(x_{si}, y_{si})$ for any one of the control points.

$$a_0 = f(x_j, y_j) - (a_1 x_j + a_2 y_j + \sum_{i=1}^{n} \frac{1}{2} b_i r_i^2 \log r_i^2)$$

With the above example, we have been scaling by the maximum value of x and y,

which was 20, but what changes must we make if $\max(x_j) \neq \max(y_j)$? Numerically, the most attractive solution would be to scale x by $\max(x_j)$ and y by $\max(y_j)$. Unfortunately, the ratios for scaling coefficients depend upon x and y being scaled by the same factor, so we have to scale by the larger of $\max(x)$ and $\max(y)$.

The next question concerns having $f(x, y)$ defined on a rectangle far away from the origin. Assuming positive coordinates, scaling by $\max(\max(x_j), \max(y_j))$ would compress the rectangle into a small region at the upper right corner of the unit square, which is numerically undesirable. We have demonstrated that the coefficients a_1, a_2 and b_i are unaffected by translation. We can produce a rectangle which touches at least an entire edge and portions of two adjacent edges of the unit square if we first translate the problem so that its lower left corner becomes the origin and then scale by $\max(\max(x_j), \max(y_j))$ of the translated square,

We will illustrate this combined translation and scaling strategy by changing y_2 and y_7 from the previous example to 27 and adding offset values of $t_x = 1009$ and $t_y = 659$. $f(x, y)$ is now defined on $[1010, 1029] \times [660, 686]$:

i	1	2	3	4	5	6	7
x_{si}	1010	1010	1016	1020	1025	1029	1029
y_{si}	660	686	674	667	671	660	686
$f(x_{si}, y_{si})$	10	10	25	15	20	10	10

Table 2.3 gives the scaled and unscaled TPS coefficients for the above problem, where $\min(x) = 1010$, $\max(x) = 1029$, $\min(y) = 660$, $\max(y) = 686$, and the scaling

factor is now $\max((\max(x) - \min(x)), (\max(y) - \min(y))) = 26$. The unscaled coefficients are recovered using the same technique as for the other example, $a_1 = a_1'/26$, $a_2 = a_2'/26$, $b_i = b_i'/26^2$ and $a_0 = f(x_j, y_j) - (a_1 x_j + a_2 y_j + \sum_{i=1}^{n} \frac{1}{2} b_i r_i^2 \log r_i^2)$.

	Unscaled	Scaled	Ratio
a_0	127.505952046	19.490798353	0.1529
a_1	0.037120887	0.965143063	26.0000
a_2	-0.159865728	-4.156508929	26.0000
b_1	-0.002628279	-1.776716770	676.0000
b_2	-0.011948817	-8.077400556	676.0000
b_3	0.039539243	26.728528405	676.0000
b_4	-0.034404467	-23.257419788	676.0000
b_5	0.018148720	12.268534504	676.0000
b_6	-0.000949291	-0.641721016	676.0000
b_7	-0.007757108	-5.243804779	676.0000
$\kappa(M)$	4.5×10^{11}	79	

Table 2.3: Ratio of coefficients between original and translated/scaled problem

2.2.1 Scaling and Translation Summary

The translation and scaling method which we have outlined appears to improve the condition number of TPS matrices significantly, allowing us to solve TPS linear systems with $n \leq 2000$ control points in double precision arithmetic. This scaling is not necessarily an optimal scaling, but it is an effective one.

The complete strategy for obtaining problem space TPS coefficients a_j and b_i from unit square space coefficients a_j' and b_i' is summarized below. First the scaling

factor is calculated as the larger of the x or y range:

$$c = \max((\max_{\forall i} x_i - \min_{\forall i} x_i), (\max_{\forall i} y_i - \min_{\forall i} y_i))$$

Next, the x'_i and y'_i values for unit square space are calculated by translation to $(0,0)$ and scaling onto $[0,1] \times [0,1]$:

$$x'_i = (x_i - \min_{\forall i} x_i)/c$$

$$y'_i = (y_i - \min_{\forall i} y_i)/c$$

The TPS linear system (equation 2.2) for the unit square space is then, $M'a' = f$, where $a' = [a'_0, a'_1, a'_2, \frac{b'_1}{2}, \frac{b'_2}{2}, ..., \frac{b'_n}{2}]$ and $f = [0, 0, 0, f_1, f_2, ..., f_n]$. Note that the right hand side of the unit square system is not scaled by c; the right hand side is the same as for the problem space system. Solving the unit square TPS system gives us the coefficients a'_j and b'_i for the unit square problem. To obtain problem space TPS coefficients, we use the following empirically determined equations:

$$a_1 = \frac{a'_1}{c} \qquad a_2 = \frac{a'_2}{c} \qquad b_i = \frac{b'_i}{c^2}$$

We can determine the problem space a_0 coefficient by simple substitution into the TPS equation:

$$a_0 = f(x_j, y_j) - (a_1 x_j + a_2 y_j + \sum_{i=1}^{n} \frac{1}{2} b_i r_{ij}^2 \log r_{ij}^2) \tag{2.5}$$

An alternative formula for a_0 which does not require logarithms of r_i^2 is given in [Barrodale, Berkley & Skea 92]:

$$a_0 = a_0' - a_1'(\min_{\forall i} x_i/c) - a_2'(\min_{\forall i} y_i/c) - (\log c) \sum_{i-1}^{n} b_i'(x_i'^2 + y_i'^2) \tag{2.6}$$

2.3 Conclusions on Theory of Thin Plate Splines

In Chapter 2, we have concentrated on the ill-conditioning problems of TPS matrices. In our condition number project, we verified that TPS matrices are extremely ill-conditioned; we verified the findings in [Franke 81], that the condition number can be improved by scaling the control points onto the unit square and we developed an empirical strategy for scaling TPS coefficients.

Through our experiments we found that we could significantly reduce the condition number of TPS equations by rejecting control points which were too close together and by translation and scaling. These techniques give us more stable TPS linear systems that can be solved using double precision arithmetic.

There still remains the completely separate problem of errors caused by the use of interpolation. As explained in Chapter 1, if two points are close enough together,

but have function values which are different enough, then an interpolating curve may overshoot the point function values. This overshoot can result in undesirable oscillations in the TPS surface and poor interpolation values.

In Chapter 3, we will look at the application of scientific visualization to TPS interpolation problems, looking for graphical measures that would indicate when a TPS surface contained interpolation anomalies and that would help us to find the cause of these anomalies.

Chapter 3

Analyzing Thin Plate Splines Using Scientific Visualization

It is often very difficult to analyze large amounts of numerical data or complicated mathematical models because the information is so abstract. The aim of scientific visualization is to enable us to build a mental model of the problem using graphical images to take advantage of our natural ability to process visual information.

We can see an application of this natural pattern matching skill in Figure 3.1. The drawing only contains a few dots, but for most people, it will be readily apparent that the dots are in a line. How do we recognize the line? In a pattern matching program, we would have to perform some mathematical calculations to see if all

the shapes were along a straight line. What if the dots were only approximately on a line, or if they formed a curve? With a pattern matching program, it would be difficult to determine the relationship between the dots, but the human visual system preprocesses the scene and gives us the fact that the dots are approximately on a line or gives us the shape of the curve. In scientific visualization, we try to take advantage of visual recognition skills by pushing as much information processing as possible off of our reasoning and onto our visual system.

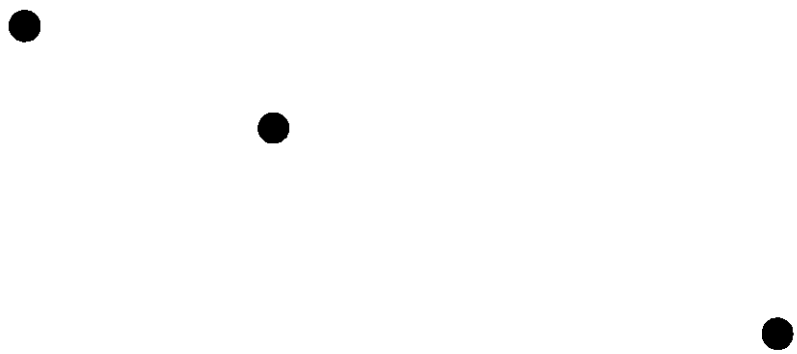

Figure 3.1: Can you see a line here?

The process of selecting a visualization method involves analysis of the problem to determine what kind of display will facilitate the desired level of interpretation. In examining surveying or mapping data, it would be easy to determine the altitude at any point given the (x, y, z) three-tuple, but very difficult to visualize the terrain or to interpolate between points. A contour map makes it possible for the user to

interpolate altitude values between points and to determine local gradients, maxima and minima, but it may still be relatively difficult to see the overall structure of the surface. With a shaded surface display it is much easier to visualize the relief or shape of the region, but it is difficult to determine point values from a perspective view.

The best visual representation of a problem depends upon the kind of information that we are trying to understand. [Robertson 91] introduces a representation methodology which distinguishes between three different interpretation aims:

1. values at a point
2. local distribution of values, such as gradients and features
3. global distribution of values, such as trends and structure

Deciding upon the correct representation for thin plate spline analysis was difficult. Since we did not fully understand the cause of our interpolation problems when we began our project, we were not sure of what data to examine; did we need to look at TPS surfaces, their partial derivatives, control point vector maps, combined surfaces, local features, or global trends? In Section 3.2, we will outline the visualization techniques we used in developing our TPS analysis method, focusing on why some techniques were more suited to thin plate spline analysis than others.

In Section 3.1, we will describe the visual analysis of thin plate splines, focusing

our attention on what we learned about TPS surfaces through visualization. Some of the material will overlap Section 3.2, but we feel it is as important to examine our conclusions about the usefulness of visualization to TPS interpolation problems as it is to describe the visualization techniques we used.

3.1 Thin Plate Spline Analysis Using Visualization

In the following case study, we applied our TPS visual analysis method to survey data which had been culled of control points closer than the threshold established in Section 2.1.1. The visual analysis helped us to find problems in the TPS surface which would lead to unacceptable interpolation values.

The first steps in our analysis were to scale the control points onto $[0,1] \times [0,1]$, create and factor a TPS matrix using the scaled control points, solve for TPS equation coefficients in the x direction, and unscale the coefficients using the method outlined in Section 2.2.1. Next, we used the coefficients to calculate delta x, and delta x partial derivative grids which could be displayed as surfaces. By delta x, we mean a surface which models the x TPS shift function.

In a perspective view of the delta x surface, we saw a fairly smooth TPS curve, but with an enlarged delta x range (see Figure 3.2). In our control point set, the range of shifts in the x direction was $[-107, -83]$, but the range on a plot of the

delta x surface was $[-140, -80]$. The greater range indicated that there was some kind of overshoot in the thin plate spline surface.

With other control points sets, we had found large spikes in the TPS delta x surface marking the source of interpolation problems, but in Figure 3.2, there was no particularly prominent peak in the delta x surface, so there was no indication of what was causing the enlarged range. With the other control point sets, we had conjectured that the first partial derivative surfaces might be better markers of interpolation problems because the peaks in the partial derivative surfaces were sharper. As can be seen in Figures 3.3 and 3.4, the delta x partial derivative surfaces for our control point set do indeed show prominent peaks at two suspected problem areas.

The thin plate spline surface which we were examining had an enlarged range, which would result in some unacceptable interpolation values, and since we suspected that the two peaks in the partial derivative surfaces were near the cause of the interpolation problem, we needed to determine the positions of the peaks and examine nearby control points. In a perspective view, it is difficult to determine position values, but by viewing the first partial derivative surfaces as flat shaded images we could determine the coordinates of the large irregularities using colour to identify the peaks and valleys (see Figure 3.5).

After reading the position of one of the peaks from the flat image, we used a data

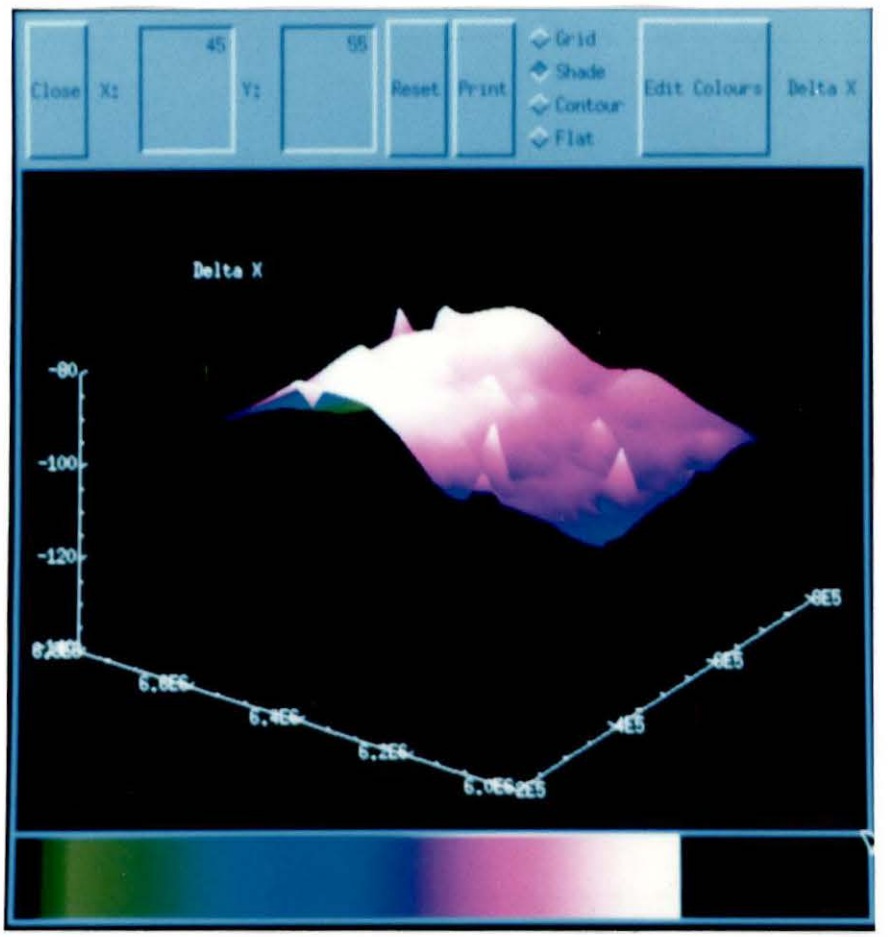

Figure 3.2: Original Shift X TPS Surface

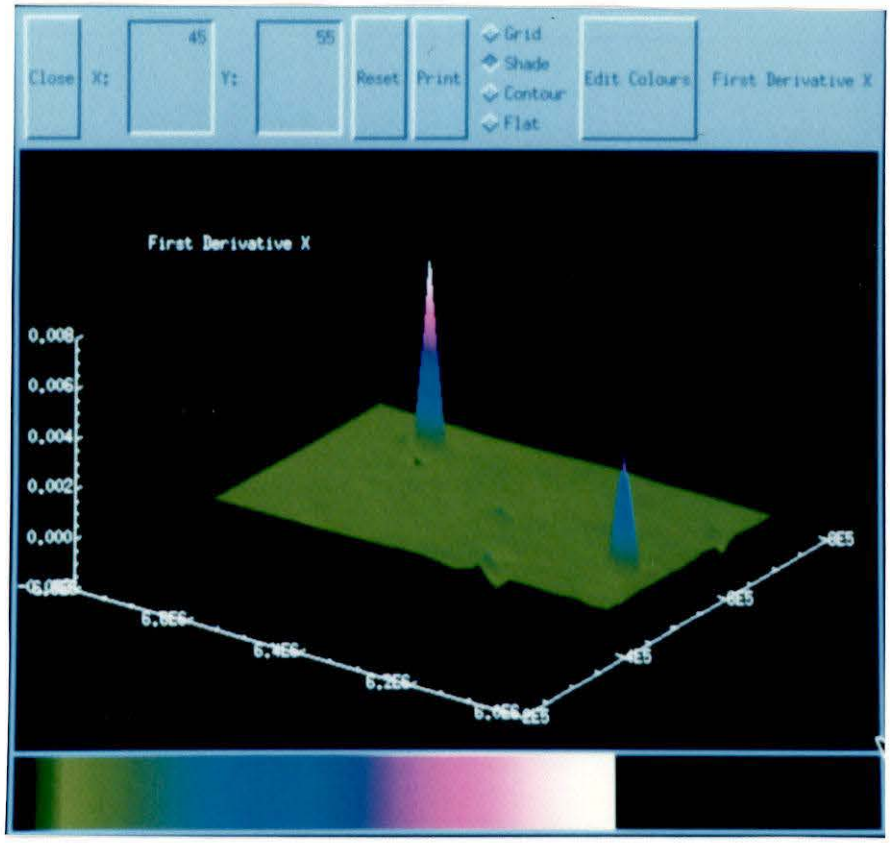

Figure 3.3: Original $\frac{\partial f_x(x,y)}{\partial x}$ Surface

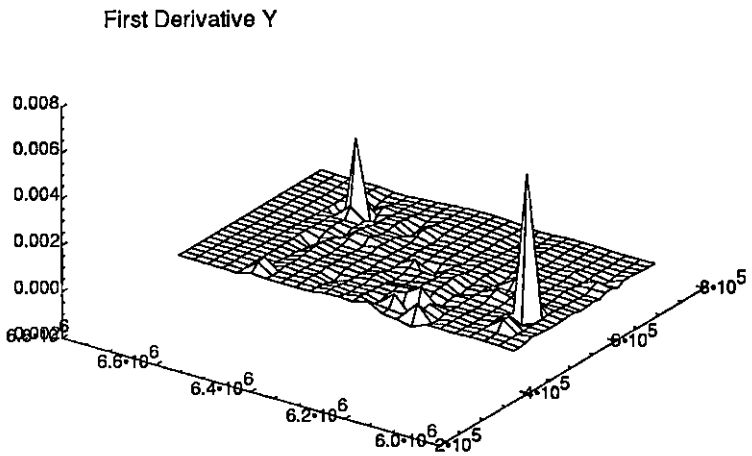

Figure 3.4: Original $\frac{\partial f_x(x,y)}{\partial y}$ Surface

analysis tool, IDL, to select all control points which were within some radius of the peak. We tried different radii until we found a region containing forty control points around the peak. By examining the points, we found that there were two pairs of points which were separated by less than $25m$. In Figure 3.6, we have plotted the shift vectors near our peak and placed boxes around the control points which were closest together. The shift vectors for each pair of the close control points are almost indistinguishable on the vector plot, but numerically, there was a slight difference, which caused an almost vertical slope on the delta x surface and led to overshoot in the thin plate spline. Once we had removed one of the two points from each pair, we eliminated one of the peaks from the partial derivative surfaces and brought

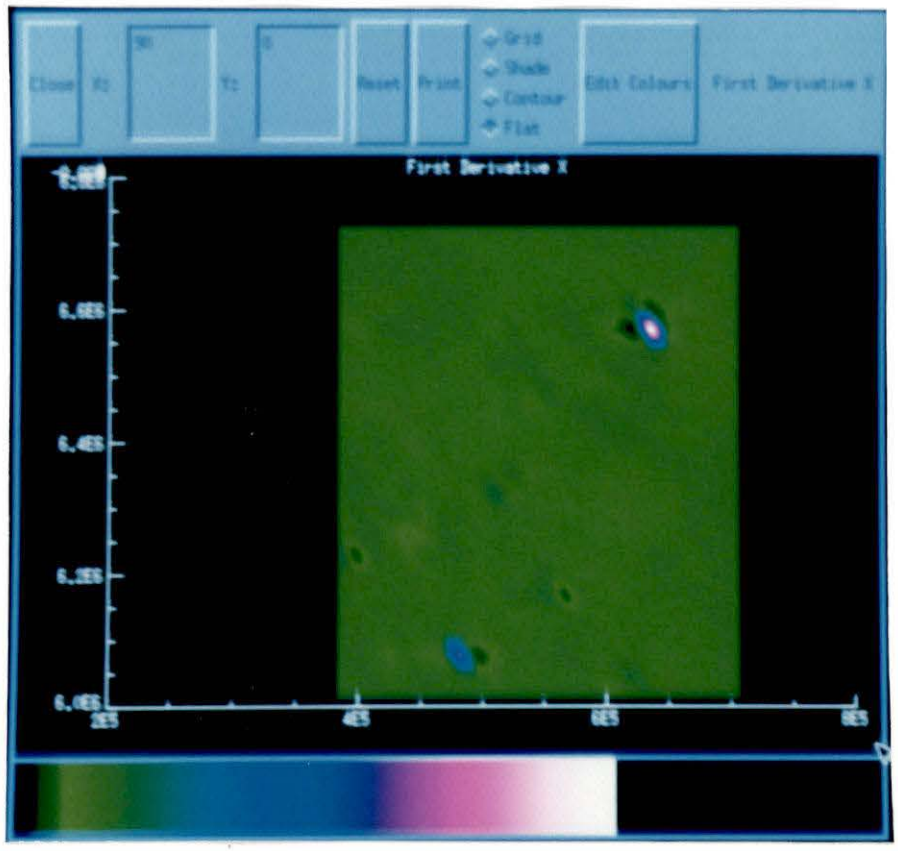

Figure 3.5: Flat Shaded View of $\frac{\partial f_x(x,y)}{\partial x}$ Surface

the delta x range from $[-140, -80]$ to $[-105, -80]$, which corresponds to the actual control point range (see Figures 3.7, 3.8 and 3.9).

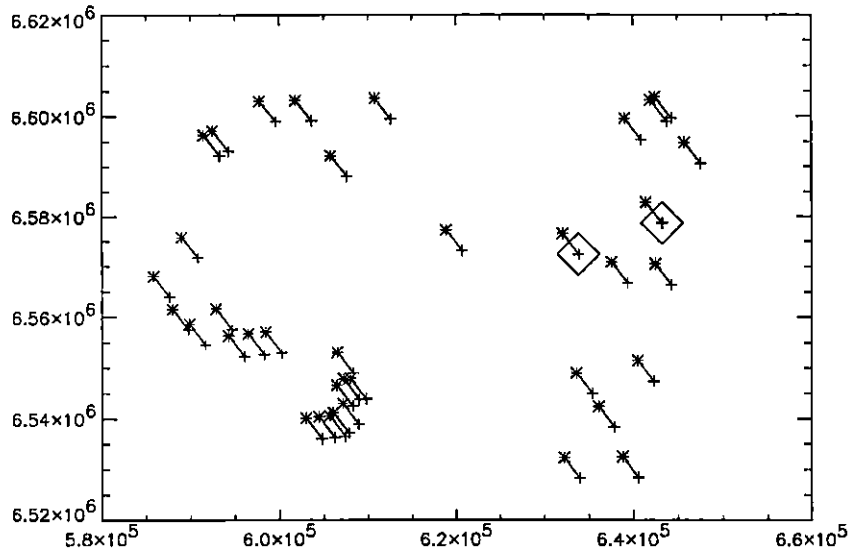

Figure 3.6: Vector Plot of Control Point Shift Values

Even though the range was now acceptable, the remaining peak in the partial derivative surfaces indicated that there was still a problem with the TPS. By following the same technique as outlined above, we found two more pairs of control points which were close together, but these were separated by distances of $178m$ and $268m$ (see Figure 3.10). This corresponds approximately to a separation of 5×10^{-4} when scaled to $[0, 1] \times [0, 1]$ which is very large compared to the lower threshold we had been using. We found 26 control point pairs in the complete data set which

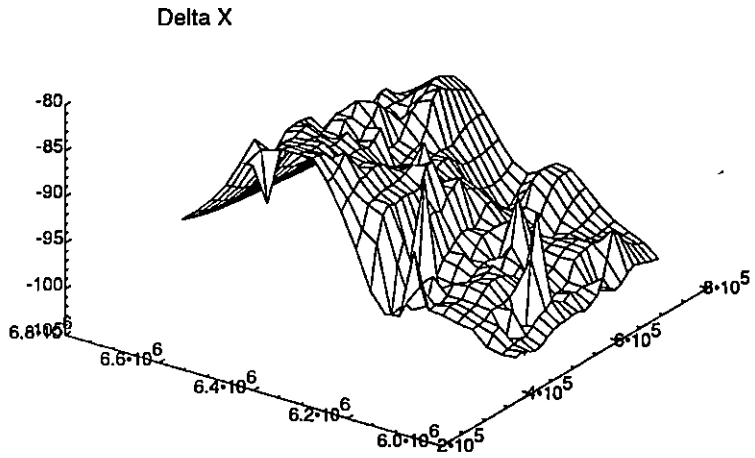

Figure 3.7: New Shift X TPS Surface

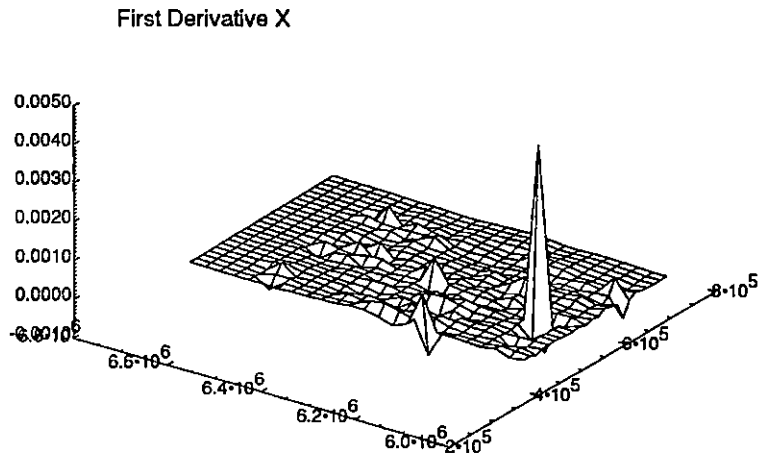

Figure 3.8: New $\frac{\partial f_x(x,y)}{\partial x}$ Surface

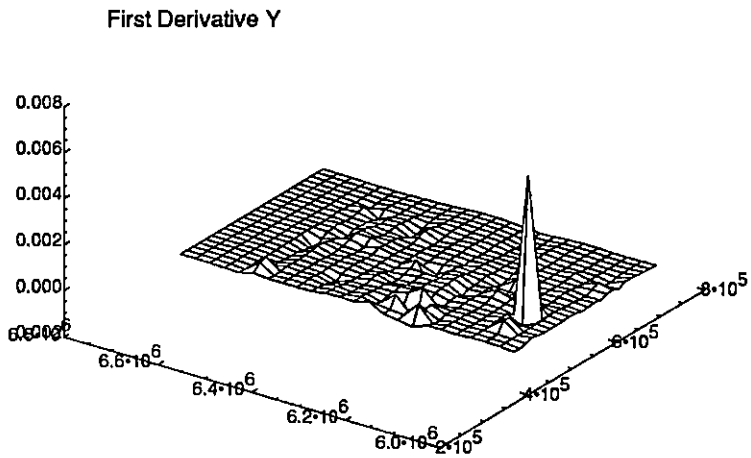

Figure 3.9: New $\frac{\partial f_x(x,y)}{\partial y}$ Surface

were separated by less than 5×10^{-4}, but only these two pairs were causing large interpolation overshoot, probably because the two pairs were both in a small region. By removing one point from the pair whose points were closest together ($178m$), we were able to improve the thin plate spline (see Figures 3.11, 3.12 and 3.13). It is important to emphasize that we were unable to find the source of this irregularity using a simple thresholding method, but using visual techniques, the location of the problem was obvious.

The new $\frac{\partial f_x(x,y)}{\partial x}$ surface, Figure 3.12, appeared to have more peaks and more irregularities than the old surfaces, Figures 3.3 and 3.8. This illusion was due to the difference in ranges between the two sets of surfaces. With the new surface, the range

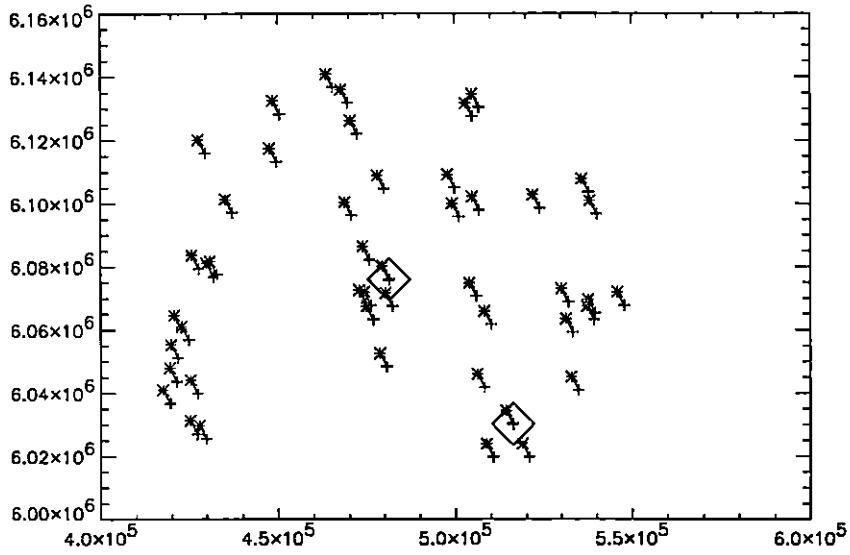

Figure 3.10: Vector Plot of Control Point Shift Values

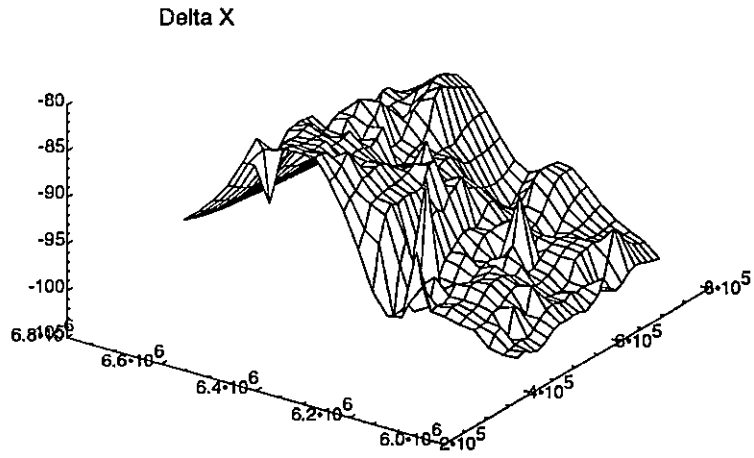

Figure 3.11: Final Shift X TPS Surface

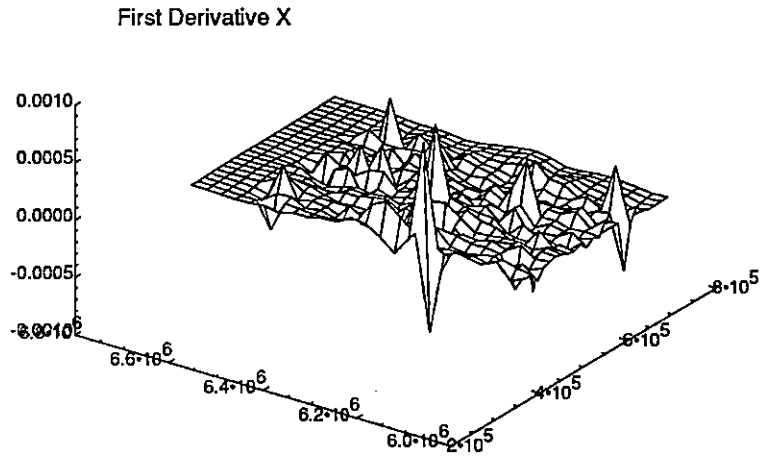

Figure 3.12: Final $\frac{\partial f_x(x,y)}{\partial x}$ Surface

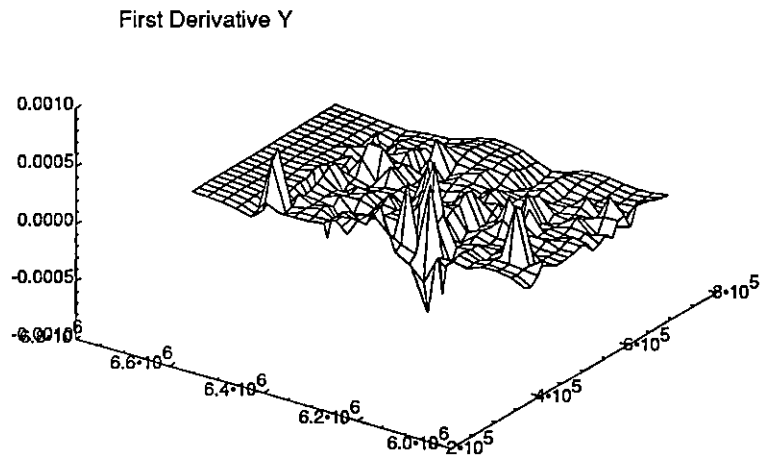

Figure 3.13: Final $\frac{\partial f_x(x,y)}{\partial y}$ Surface

was only $[-1\times10^{-3}, 1\times10^{-3}]$ while the old surfaces had ranges of $[-2\times10^{-3}, 8\times10^{-3}]$ and $[-1 \times 10^{-3}, 5 \times 10^{-3}]$. The greater ranges in the old surfaces hid the smaller peaks and valleys that became evident in the new partial derivative surface. What should we do with these new peaks and valleys?

If we were to examine and "fix" each peak and each valley, we would eventually end up with a flat plane, which defeats the purpose of fitting a surface to the data points. Are there limits on the height of the partial derivative peaks which we should correct? We conjecture that if we worked with thin plate splines which were scaled to $[0,1] \times [0,1]$, then it might be possible to establish maximum absolute values for the partial derivatives, but without further research, we cannot make this conclusion. Instead, we have found that the range of the TPS surface is the best indicator we have of the existence of interpolation problems that will affect our solution. If the range of the new TPS surface falls within the range of our shift values, then we accept the surface as is. Even though the range in Figure 3.7 was acceptable, we corrected for the peak in Figure 3.8 because the two peaks seen in Figures 3.3 and 3.4 appear to be equally large.

In the above example, we detected and corrected interpolation problems which were due to control points being too close together. One other condition which will cause interpolation difficulties is the presence of conflicting shift values. Figure 3.14 shows an example of consistent and inconsistent shift vectors. On the left, the

consistent shift vectors will scale data towards the centre and down. On the right, the inconsistent shift vectors cross, and there is no interpolation function which could properly model that kind of behaviour. If the input data contains conflicting shift vectors, then either there are errors in the data that should be corrected, or the problem should be modelled using an approximating function, not an interpolating thin plate spline.

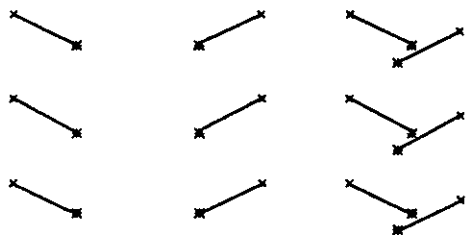

Figure 3.14: Consistent vs Inconsistent Shift Vectors

Thin Plate Spline Visual Analysis Summary

In summary, our technique for visually analyzing thin plate splines involves looking for overshoot in the range of TPS surfaces, then identifying particular sources of interpolation difficulties in the partial derivative surfaces. The entire process is as follows:

1. Create and factor the TPS matrix using scaled control points.
2. Solve for the TPS equation coefficients in the x direction.
3. Unscale the coefficients using the method outlined in Section 2.2.1.
4. Calculate delta x, and delta x first partial derivative surface grids.
5. Display the TPS surfaces and if the range for the delta x shift surface is much greater than the actual range of control point shift values, then there are definite interpolation problems in the thin plate spline.
6. If the range for the delta x surface is too large, then sharp peaks and valleys in the partial derivative surfaces pinpoint problems due to points being too close or due to inconsistencies in the data. To obtain the position of an irregularity, rotate the surface so that the observation point is directly above the surface, and read the position of the irregularity from the axes.
7. Select a region around each irregularity containing between twenty and fifty control points.
8. Display vector plots of the shifts in the selected control points and examine the plots for points which are too close or have conflicting shifts.
9. If two points are found to be too close, then discard one of the points. If the shift vectors for the two points are of the same length and direction, then the choice of which point to discard is arbitrary, but if one of the points has a shift vector which is inconsistent with nearby shift vectors, then the control point with the inconsistent shift vector should be discarded.
10. If there are inconsistent shift values, determine if there is an error in the data, but if there is not, then it may not be possible to model the data using interpolation techniques.

3.2 Visualization Techniques Applied to Thin Plate Splines

In the previous section, we outlined a method for visually analyzing thin plate splines using tools which we had developed over a period of time. In this section, we will be examining the various visualization techniques we experimented with and we will discuss some positive and negative features of each technique.

When we began our development, we did not know what type of visual representation would be suitable for thin plate splines. What kind of information would we get from a TPS surface or a vector plot of $r^2 \log r^2$ terms? What kind of information were we trying to conceptualize?

Our main goal was to understand what caused a TPS to give unsatisfactory results and to find some way of avoiding the situation. From numerical results for the Survey Data Project, we were expecting to find noticeable irregularities in the TPS surface, so we decided to begin with wire-grid and shaded displays.

3.2.1 3D Surface Grids

The human visual system is well suited to analyzing natural scenes and making judgments about the heights and shapes of terrain features. We can exploit this natural talent by displaying functions which are defined over 2D regions as wire

grid simulations of a natural scene. At a glance we can tell how smooth a function is, where there are irregularities, etc. Some information is hidden behind features of the scene simulation, but by providing a means to interact with the scene (to rotate the surface, for example), the user can examine any area of interest.

A wire surface grid is a natural way of representing a thin plate spline (see Figure 3.7 for an example). The spline is a function of two variables, x and y, defined over a normal 2D spatial region. The function is smooth, so we can expect no gaps in the surface. When we view a TPS grid surface, we can assess the relative smoothness of the surface, see any spikes, and determine the range of function values.

We started looking at surface grids because we were searching for an explanation to some unacceptable shifts in numerical results from the survey data project. When we plotted the offending TPS surface, we found a few very sharp spikes, and in one area there was a very sharp positive and a very sharp negative spike, indicating interpolation overshoot in the TPS surface. These local features were very easy to see from the wire grid plot.

In spite of their advantages as natural scene simulations, TPS grid surfaces can be misleading if the observer is not aware of all the information a surface grid is showing. For instance, in the survey data example noted above, we were initially quite pleased with the results and attributed the spikes to inconsistency in the data that could be ignored. We assumed that the smoothness of most of the surface was

an indication of a good thin plate spline fit. It was only when we examined the plot further, that we realized the vertical range for the surface was too large and that the large range (due to the spikes) would tend to smooth out the surface, making the TPS seem better than it actually was. After the worst problems were handled, we found that the surface was still very bumpy and contained more irregularities due to interpolation overshoot. Even a good TPS surface contains many sharp features when viewed in its proper range.

We also had difficulty in determining the location of the spikes, because of the perspective view. We could see the spikes on the plot, but it was difficult to determine their x and y positions using the wire frame surface. In Section 3.2.3, we will describe the methods we developed to determine feature positions.

Finally, the IDL drawing software we were using appears to fit a $C1$[1] polynomial surface to the data, so that we are only able to view an approximation of a TPS surface. For example, the second derivative of a thin plate spline has discontinuities at every control point, but the discontinuities do not appear in wire grid surfaces because IDL's surface plotting routines try to fit a smooth surface to the input data. IDL does provide polygon drawing routines that would allow us to draw more correct TPS surfaces, but we would have had to design a complete TPS surface

[1] C1: a continuous function with continuous first derivatives

fitting routine to use the low-level drawing routines. The limitations of the IDL surface fitting routines are not severe, so we decided to use the built-in routines, while remaining cognizant of interpretation difficulties..

3.2.2 Shaded Surfaces

Shaded surfaces are often better representations than surface grids because the shading makes the surface look more like the natural scenes we are accustomed to (see Figure 3.2). The addition of shading usually enhances our ability to determine the overall shape of a surface.

With TPS surfaces, we found that in most instances the shading would help in visualizing the spline, but at acute angles we found that it was difficult to separate foreground and background features because the colours would blend together making edge detection difficult. In these case, gridding supplied a better visualization of the thin plate spline, although non-Gouraud shading[2] might have helped us to visually separate parts of some surfaces by adding a degree of false texturing.

The most significant advantage of shading over gridding was the extra informa-

[2]Shaded surfaces are drawn using polygons. Gouraud shading is a method of interpolating the shades across a polygon so that there are no abrupt shade changes at the edge between two polygons. The human eye is adept at using sudden colour changes to find edges, and Gouraud shading is designed to interfere with this edge detection ability.

tion provided by the shade colours. With alternative shading models and interactive display (to allow rotation), it was possible to obtain point, local and global information from the same shaded surface representation.

IDL allows the user to shade using the standard lighting model or to specify the shade to use at each data point. We looked at three different choices: standard lighting model, function value and fourth variable.

The standard lighting model puts a virtual light source at some set angle in the scene and paints each pixel of the surface according to the colour of the pixel and the angle of the surface to the light source at the pixel. Areas which face the light source are bright, those which face away are dark, and those at an angle are in between. This simulates the natural shading model that the human visual system is most adept at interpreting.

Our problem with the standard lighting model is that the shading for an area is derived from both the natural colour of the area and the angle of the area to the light source. The colour does not depend solely on the height of the surface, so we cannot use shade information in the absence of shape information.

For example, suppose we saw, as in the survey data, that there was a sharp peak and a sharp valley in the scene and we wanted to determine the position of this anomaly. It is not easy to derive point data from a perspective view of the scene, but we could rotate the surface to obtain an overhead view and use surface shades to

locate surface features. However, because the shades in the natural lighting model do not depend only on the data, we would have to mentally record the shades of the features which we are examining and look for those shades rather than looking for simple light or dark features.

Instead of using the standard lighting model, IDL also allows the user to specify the shade of each data point. By scaling the TPS function values onto the colourmap range, we were able to colour TPS surfaces with shades that depended solely upon the height of the surface. This is similar to the standard lighting model, but since the colour depends only upon shift values, we could obtain feature information using the colour values alone. After rotating the view point to directly over the top of the surface, we could read feature positions by looking for particular colours. This representation, a surface shaded with shift values, quickly became our preferred representation because it supplied us with all of the information we wanted in a format that was easy to understand and interpret. Whenever we refer to a "shaded surface" in this thesis, we refer to a surface which has been shaded using shift values.

We could also use shade values to represent a fourth variable. With this method, a surface plot depicts four-tuples: (x,y,z,*shade*). For a TPS surface we could, for example, include partial derivative information and obtain a plot of (x,y,shift x,$\frac{\partial f(x,y)}{\partial x}$). Unfortunately, we were unable to find a fourth variable that would give us further insights. The shift values and partial derivatives were too interdependent to clearly

give us more information about the TPS, and in any case, normal TPS surfaces are so bumpy that it is difficult to interpret the fourth variable.

3.2.3 Determining the Position of Surface Anomalies

An important part of our analysis was to determine the position of various surface features. Knowing the position of surface anomalies would allow us to do a more detailed examination of nearby control points, hopefully leading to some conclusion about the cause of the irregularities.

As noted above, an important advantage of shading with shift values is that we can view the surface from above and identify surface features by their shade values alone (see Figure 3.5). Because the surface was flat, we could determine the position of various features on the surface by using the (x, y) axes. This method of positioning was the most useful and natural for us to use. We found that it was very easy to interpret flat shaded images and to determine the positions of surface features from the image.

Another, more traditional, method for determining the position of surface deviations is to use contour plots. We found that because of the irregularity of some TPS surfaces, it was sometimes difficult, numerically, to produce good contour plots using the software available. Gaps in the contour shading or crossing contour lines were common.

We also found that, for our purposes, contour plots contain no more useful information than a flat shaded surface image and were more difficult to interpret. The only extra information that is available in a contour plot is the z value label on each contour, but we were interested in local/global shape and (x, y) position, not exact height values, so we had no use for the extra height information.

According to the criteria in [Robertson 91], the contour plot was not a useful visualization tool for our research. Contour plots do not enhance our understanding of TPS problems; they do not provide any extra information and they do not improve our mental model of thin plate splines.

3.2.4 Marking Important Points

The surface representations allowed us to picture overall trends in the TPS surface and to discover surface anomalies. Once we had identified any irregularities in a TPS surface, we wanted to visually relate these irregularities to particular control points or data points which we suspected of causing interpolation difficulties. IDL surface plots include labeled axes, so we could have located points on the surface by using the axes, but we wanted to be able to identify the points in perspective plots, where axes are not as useful.

Our first attempt was to add vertical vectors to mark point positions. For each point we added a vector to the surface plot from the maximum z-value of the surface

to the shift value of the point. This approach was disastrous because IDL does not perform hidden line removal for vectors that are added after a surface plot. Some vectors would, in a perspective view, appear to be in front of a surface feature when, in fact, they should have been behind. Drawing the vectors first would not have helped, since vectors which should appear in the foreground would then be covered by background surface features.

The next approach was to add symbols at $(x, y, shift)$ or $(x, y, \max z)$ positions to locate the points. This solution suffered from the same problem of hidden line removal: a point may appear to be in front of a surface feature when, in fact, it is behind. As well, a few point markers tended to disappear into background clutter with wire grids or to disappear on shaded surfaces because they happened to be drawn in the colour of the surface at the point in question.

We finally concluded that because of limitations within IDL, there was no straightforward way to mark important points that would add useful information to our visualization. Reading point positions from the axes has been good enough.

3.2.5 Vector Plots of Control Points

Besides visualizing the actual TPS surface, we also needed to look at the relationship between control points which form the thin plate spline. When we found some irregularity in a thin plate spline, we wanted to examine the control points and

control point shifts in some region around the anomaly. The hope was that we would find some obvious pattern in nearby control points that was causing the irregularity.

Using IDL, we selected a set of control points from some small region around the anomaly, varying the radius until we had between twenty and fifty control points. Next, we performed a vector plot of all the selected control points, displaying the sensed control point position and the shift from sensed to reference position (see Figure 3.6). Unfortunately, with our survey data, the (x, y) scale was much larger than the size of shifts, so the vectors were almost of zero length. By magnifying each vector by a scaling factor, we were able to see the direction of the vectors and their relative lengths.

Once we had the final vector plot, it was easy to see any irregularities or inconsistencies in the pattern of vectors and to identify which few control points were causing the irregularities in the thin plate spline. We could also decide whether the problem was in the selection of control points (points too close) or in the data (inconsistent shifts).

Using the Robertson criteria, the key points to this aspect of our visualization work are:

1. We wanted to find patterns, so we were interested in global properties of the

control point set. The vector plot supplied this information for us.

2. Once we found questionable patterns, we wanted to determine which control points were causing the problem. The vector plot allowed us to determine the point values (position) of control points.

3. In our visualization, the position of the reference points (at the head of each vector) was not as important as the direction of the vectors, so scaling up the length of the vectors facilitated the kind of interpretation that we needed.

As well as the standard vector plot described above, we also examined 3D vector plots of the relationship between some small number of control points and their surrounding neighbours. Although there might be some potential for enhancing our investigation if we could depict the vectors in the right way, the representations we attempted rapidly became too complex for meaningful analysis and suffered from the lack of hidden line removal in IDL. In essence, this visualization technique complicated instead of simplified our analysis, so we rejected three dimensional vector plots.

3.3 Visualization Summary

In Chapter 2, we outlined some of the theory behind thin plate splines, including the ill-conditioning of TPS matrices and methods of improving the condition number. In Chapter 3, we have examined a method for visually analyzing thin plate splines, giving us the ability to detect and correct unacceptable interpolation results. The

main features of the method involve detecting interpolation problems by looking for overshoot in the range of TPS surfaces and identifying the sources of the problems by looking for sharp peaks in the partial derivative surfaces.

Not all of the techniques which we developed were useful in thin plate spline analysis, but by following the criteria outlined in [Robertson 91], we were able to focus on the most promising graphical tools. IDL was useful as a graphical programming environment, but the lack of hidden line removal algorithms made it impossible for us to combine various vector plots with surface plots.[3]

Scientific visualization is a powerful tool for analyzing complicated functions. Even if we can develop numerical methods for detecting TPS interpolation overshoot, the development of visual analysis tools for thin plate splines has helped us immeasurably in understanding the behaviour of thin plate splines, their derivatives and the causes of interpolation problems.

[3] Research Systems Inc. has since added software Z-buffer capabilities to IDL which will provide hidden line removal, but we have not yet taken advantage of this new feature.

Chapter 4

Application of Thin Plate Splines

In Chapter 2, we discussed some of the ill-conditioning problems that affect TPS matrices and in Chapter 3 we investigated the visual analysis of thin plate spline interpolation problems. In spite of ill-conditioned TPS matrices and interpolation difficulties, thin plate splines seem to be very useful in registration problems. In this chapter, we will examine two applications of thin plate splines: registration of geodetic data to convert old coordinates to new and registration of side-scan sonar data to compare sonar images.

The Survey Data Project examines the application of thin plate splines to the

transformation between the North American Datums of 1927 and 1983. In order to make use of geodetic data that are based on the 1927 datum (for example, forest cover maps used for logging inventory), we need to be able to convert 1927 coordinates to 1983 coordinates. The conversion process is very similar to image registration, so we have experimented with the application of thin plate spline warping to datum conversion in Section 4.1.

The image acquisition process for side scan sonar is very prone to distortion from wave action, water turbulence, thermal variations, etc. These non-linear distortions make it very difficult to compare two images of the same scene without some kind of image registration process. [Skea & Barrodale 90] indicated that thin plate splines would be good side-scan sonar warping functions because thin plate splines handle local geometric distortions well and because thin plate splines are interpolating rather than approximating registration functions. Section 4.2 will describe the application of thin plate splines to side-scan sonar image registration using scientific visualization techniques in the manual selection of control points and in the visual detection of changes to an area through image comparison.

4.1 Survey Monument Registration

The 1983 and 1927 North American Datum sets are geodetic reference systems for North America, including Canada, Greenland and the United States. The reference systems provide known geodetic points of reference for land surveyors, cartographers, and others.

NAD83[1] is a much more accurate and consistent geodetic reference than NAD27. An inexpensive, but correct, way of transforming NAD27 based data sets to NAD83 is needed so that older data can be incorporated into the NAD83 standard.

This problem is very similar to image registration. Two pictures have been taken (the 1927 and 1983 surveys) of the same scene (North America), but there are different errors and local distortions in each image. We have a set of control points (the NAD survey monuments) and would like to know how much to shift points that are between survey monuments so that the NAD27 image can be warped into the same shape as the NAD83 image.

Most of the transformation methods rely on least squares. In British Columbia, these methods have been troublesome because they have required so much expert knowledge to use and in some cases, the errors at survey monuments have been $18m$ or more. We postulated that thin plate splines are more suited to calculat-

[1]NAD83: North American Datum of 1983. NAD27: North American Datum of 1927.

ing the shifts because they handle local distortions much better than polynomial based models. As well, polynomial methods are approximating, but NAD27 and NAD83 survey points are supposed to be reasonably accurate, which would suggest an interpolation scheme.

We were given NAD data sets by the Government of British Columbia to use in testing our hypothesis. There were four data sets: NAD27 *Fixed*, NAD83 *Fixed*, NAD27 *Check* and NAD83 *Check*. Points from NAD27 *Fixed* and *Check* sets correspond exactly to points in the NAD83 *Fixed* and *Check* sets, respectively. Our goal was to use a thin plate spline based on the *Fixed* data sets to predict shift values for the NAD27 *Check* points and compare our predictions to the actual NAD83 *Check* points.

Although we did have some problems and discovered some TPS limitations, we found that thin plate splines generally performed well and yielded some impressive results with the check points. Unfortunately, the NAD data sets do contain pairs of NAD survey monuments which were close enough to cause overshoot and other pairs which had conflicting shift values. Without culling the NAD sets, some thin plate spline interpolation difficulties were inevitable.

4.1.1 Warping Method

Our NAD27 and NAD83 data sets consisted of 1708 *Fixed* points and 541 *Check* points in UTM format. We took the NAD27 *Fixed* data as the control point set for our sensed image. Before creating the TPS system, we found some control points which were too close together according to the criteria established by our condition number experiments. In Section 2.1 we had shown that generally when points are very close together, either the TPS matrix was singular or of very high condition number. To preclude stability problems, we eliminated one point at random out of each problem point pair.

After culling the control points which were closer together than the lower threshold we had established in Section 2.1, we created and factored a thin plate spline matrix (equation 2.2), and obtained coefficients for x shift and y shift TPS equations (equation 2.1). To verify the TPS coefficients, we calculated shifts for the control points and showed that there was virtually no error between the calculated shifts and the expected shifts used in creating the TPS matrix.

After calculating shifts for the *Check* points we compared the calculated shifts with those expected from the NAD83 *Check* data. Unfortunately, we found that some of our *Check* points were being shifted by unreasonable amounts. For example, one point in particular should have been warped by approximately $-100m$, but

instead was warped by $40m$. How could the thin plate spline produce a positive shift value when all of the control points had negative shifts? In fact we had discovered some interpolation overshoot due to points being too close together, but it was difficult to come to this conclusion without some analysis. There were far too many control points to be able to understand the problem, so we turned to the graphical techniques which we outlined in Section 3.1.

First we used our IDL surface plotting routines on the TPS shift surfaces and on the first partial derivative surfaces (see Figures 4.1, 4.2). We found that there was a large irregularity in all of the surfaces near the points with strange shift values. In the delta x surface, the irregularity consisted of a sharp positive peak and a sharp negative peak side by side, which is representative of interpolation overshoot. At the irregularity in the TPS surface, the positive peak went above zero, and as a result some points near this feature would be shifted by positive instead of negative amounts, which was the case with the point that was shifted by $40m$. The range of the delta x TPS surface was also an indicator of overshoot, since the range was $[-400, 200]$ instead of approximately $[-110, -80]$.

We used IDL to find roughly fifty control points which were within a small radius around the irregularity, and by examining the control point subset with a vector plot, we found that two of the points were very close together and had inconsistent shift vectors. The points were only separated by $20m$, but their shift vectors had lengths

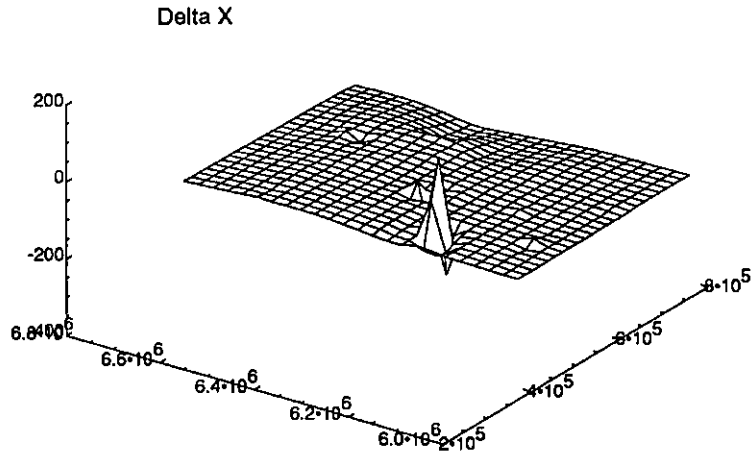

Figure 4.1: Original NAD Survey Monument Delta x Surface

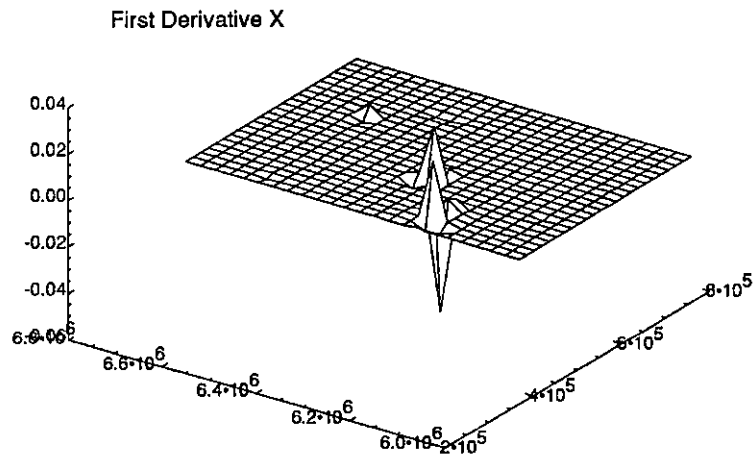

Figure 4.2: Original NAD Survey Monument $\frac{\partial f_x(x,y)}{\partial x}$ Surface

of $215m$ and $229m$. When scaled to $[0,1] \times [0,1]$, the two points were separated by approximately 10^{-6}, which was acceptable according to our condition number experiments in Section 2.1. Since the points were so close, but the shift vectors were inconsistent, the TPS surface suffered from interpolation overshoot.

As in Section 3.1, we removed one of the two points from the control point set, then recalculated and redisplayed the TPS surfaces. We found that the prominent peak was gone from all surfaces and that the delta x and $\frac{\partial f_x(x,y)}{\partial x}$ ranges were decreased.

The partial derivative surface in Figure 4.2 gave evidence that there were more instances of unacceptable interpolation overshoot. The two large peaks in Figure 4.2 indicated two possible sources of interpolation overshoot. The first peak was the same as we found in Figure 4.1, but the second peak did not appear in Figure 4.1. We used the same method as before, finding a small subset of points near the irregularity and looking for control points that were close together, but had inconsistent shift vectors. Here we found two control points that were separated by $19m$, but had shift vectors of length $219m$ and $236m$. After removing one of the points and recalculating, we produced Figures 4.3 and 4.4. These surfaces were much better than the originals, but Figure 4.4 shows that there were still more difficulties to investigate.

As explained in the introduction to Chapter 2, thin plate splines are interpolat-

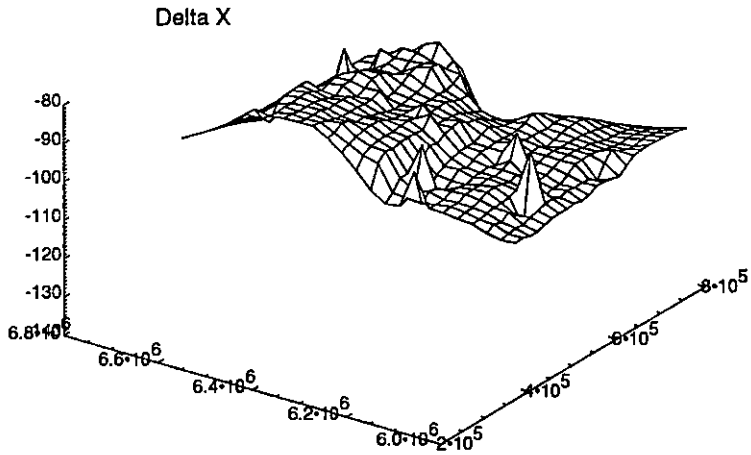

Figure 4.3: Corrected NAD Survey Monument Delta x Surface

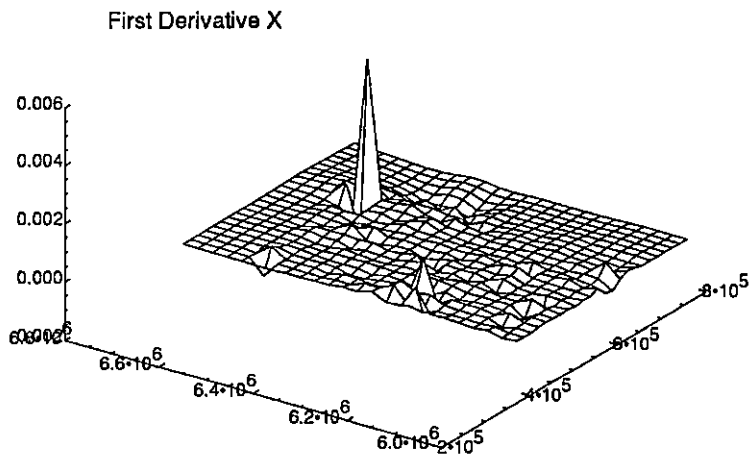

Figure 4.4: Corrected NAD Survey Monument $\frac{\partial f_x(x,y)}{\partial x}$ Surface

ing splines, so TPS shift surfaces pass through each control point exactly. If the control points are inaccurate, and conflicting control points are close enough then the TPS surface may display severe interpolation overshoot. In such situations, thin plate splines are inappropriate warping functions. [Ragozin 91] suggested adding a smoothing function to the thin plate spline to handle control point conflicts. He proposed a warping function that combined a thin plate spline with some kind of least squares function as follows:

$$\mathrm{SMOOTH}_x(x,y) \;=\; \beta\,\mathrm{TPS}_x(x,y) + (1-\beta)\,\mathrm{LST}_x(x,y) \qquad (4.1)$$

The parameter β in equation 4.1, allows for the adjustment of the smoothed spline between least squares fit ($\beta = 0$) and thin plate spline ($\beta = 1$). The LST function would serve to smooth out distortions introduced by any thin plate spline overshoot.

One very practical problem with this solution is that if β is small enough that the smoothing function can handle sharp TPS irregularities, then the combined warping function may not be able to handle local distortions very well. It is the ability of thin plate splines to adjust to small distortions that causes problems with small inaccuracies in the control point data.

The fundamental issue is the conflicting fitting requirement in NAD data regis-

tration. On the one hand, exact registration of control points is desirable because most of the points are known to significant accuracy. The monuments are, in fact, real objects on the ground; their positions are not subject to adjustment or error. On the other hand, there are large errors and inconsistencies in many of the other points, which would suggest an approximation scheme. Thin plate splines are designed to provide exact interpolation of control points, yet to handle local irregularities in the data. To some extent, thin plate splines are quite suited to survey monument registration, however TPS matrices have proven to be vulnerable to some of the severe inconsistencies we have encountered in surveying data.

4.1.2 Survey Data Project: Conclusions

Errors in the NAD27 data and differences between the NAD27 and NAD83 models of the Earth lead to significant non-linear differences between the datums. Some registration process was needed to convert NAD27 coordinates to NAD83 coordinates and that registration process must be able to handle the local distortions introduced by errors in the datums.

It was desirable that any registration process would convert NAD27 datum points to NAD83 points exactly, so we suggested interpolation using thin plate splines. It has been shown, [Franke 81, Goshtasby 88], that TPS surfaces are able to handle local distortions in image registration problems, so we conjectured that thin plate

splines would be able to model the local distortions introduced by errors in the NAD data sets.

We found that although it was possible to produce a registration function using thin plate splines, the presence of control points which were too close together with inconsistent shift vectors led to significant problems with interpolation overshoot.

As explained above, the root of this survey data registration problem lies with the conflicting requirements for NAD data registration. If data inconsistencies which made interpolation ineffective could be repaired, then thin plate splines would make excellent tools for survey monument registration. If it is undesirable to correct these inconsistencies, then an approximating method of survey monument registration is indicated, which may lead to inaccuracy in the registration of control points.

Correcting data inconsistencies is not the only obstacle to warping NAD27 to NAD83. [Skea 92] found that the survey monuments and other available control points do not provide sufficient control to model all the errors. This lack of control will affect *any* warping function, not just thin plate splines. In the regions studied, large areas had no control points and of the control points available, many were subject to differences of interpretation. By examining the methodology used in selecting control points, it may be possible to improve the choice of control points and to make better use of thin plate spline registration in transforming NAD27 to NAD83.

4.2 Change Detection in Side-Scan Sonar Images

All sonar transducers have a practical limit on their field of view due to the critical angle of sound waves in water. Because normal sonar uses a single downward looking transducer, there is a limit to how much of the ocean bottom can be viewed. Side-scan sonar uses sound waves from a pair of transducers on the sides of a towed sensor to produce a grey scale[2] image of the ocean bottom. The side-mounted transducers are directed outwards, so the combined image contains more of the ocean bottom than an image from a single downward looking transducer.

In our application, two side-scan sonar images of the same scene, taken at different times, must be visually compared to detect changes in the area. Therefore we must overlay the two images using some kind of warping function. Unfortunately, there are some difficult registration problems to be overcome. For example, movements of the towed sensor, turbulence and thermal layer fluctuations lead to local, non-linear distortions in the side-scan sonar images.

[Skea & Barrodale 90] indicated that thin plate splines appear to be good warping functions for handling this registration problem. Side-scan sonar images contain local geometric distortions, therefore a registration method which handles local distortions is required. In the applications cited in [Skea & Barrodale 90], the control

[2]Grey scale means that the images are coloured with white, black and shades of grey only.

points are considered accurate relative to the images, so an interpolating registration function is desirable.

In the fall of 1990, a research project was begun at Barrodale Computing Services Ltd. by M. Dunham-Wilkie to create a graphical system using thin plate splines to detect changes in pairs of side-scan sonar images. In January of 1991, the author took over the Change Detection System, converting it from PV~Wave windows to IDL widgets, improving the user interface and enhancing visualization aspects of the system. Section 4.2.1 details some of the scientific visualization design issues that were important to the Change Detection System and Section 4.2.2 describes the user interface.

4.2.1 Designing the Change Detection System

When designing the system, we aimed at a user-interface that was not only intuitive and easy to use, but also assisted the operator's analysis through the use of scientific visualization techniques. The two main issues, graphically, were the selection of control points and the comparison of warped images.

When using an interpolation technique for image registration, control point selection is an important issue because the fidelity of the interpolation depends upon accurate selection of control points. In the Change Detection System, a user can select control points visually by displaying the two images on the computer screen,

moving the mouse cursor over identifiable features and clicking the mouse button to mark a control point.

The manual selection of control points is a difficult procedure and requires a skilled operator. We can encourage accurate selection by providing visual tools for the operator to make fine corrections to control point positions. A doctor trying to perform fine surgery on a small part of the body, will use a microscope to get an enlarged view of the area. He may also use special tools so that large movements of his hand are translated into small movements of the instruments he is using. We give the same kind of control to our users by providing them with larger versions of the original sonar images. The magnified images make it easier for the operator to differentiate small details and because the mouse cursor moves over larger images, the user has finer control over the position of the pointer. Most computer screens are not large enough to display huge images, so the Change Detection System uses two smaller windows that show complete images and two viewports that can be scrolled over the larger versions of the images.

One useful measure in control point selection is the consistency of shift vectors. TPS interpolation is based on the assumption that all points in some small region will be shifted by similar amounts in similar directions. In Section 4.1.1 we saw an example of the instability problems that can arise from inconsistent shift vectors. It is expected that there may be some recognizable pattern to control point shift

vectors (no vector crossings, no abrupt changes in magnitude, etc.) and since the human visual system excels at pattern matching, the Change Detection System provides a graphical display of the control point shift vectors, which the operator can use to get an approximate assessment of control point selections.

In order to position control points correctly, the user will have to compare the position of a control point in the Reference image with the point's position in the Sensed and fine tuning images and with the associated shift vector. Since we expect each image pair in our application to require a hundred or more control points, it may be difficult to associate a control point from the Sensed image to the corresponding points in the other images. Side-scan sonar images are grey scale images, so we can change the colour of associated points in each image to a bright, high contrast colour, and take advantage of the eye's ability to discern between objects of different colours. Once the point is highlighted, the operator is easily able to associate control points between images. By default, the highlight colour is green, but this can be changed to assist those who are colour blind.

The final important scientific visualization design feature in the Change Detection System concerns the comparison of the original Reference image with the warped Sensed image. With a computer windowing system, placing the two images side by side is a very simple, but ineffective way of allowing for comparison. There are many numerical image processing techniques which would allow us to combine

the two images and show only the differences or mark the differences on a copy of the Reference image. An alternative is to use a visual approach, placing both images into a single window and flickering between them. If the images are similar enough to be well registered, differences will stand out as blinking objects. This approach uses the human visual system's ability to detect motion as a means of detecting changes in the two images.

If there are confusing differences in the images (such as differences in the noise band which appears underneath the towed sensor or marked differences in shading), then the mouse cursor serves as an additional visual tool in flicker comparison, acting as a movable visual anchor, and allowing the user to associate scene features between the two images.

4.2.2 The Change Detection System

The system contains two main parts, the Original Images subsystem for displaying the original images and for control point selection and the Image Warping subsystem for selecting warping methods, warping images, displaying the warped images, and providing the flicker window for comparing warped Sensed and original Reference images.

Original Images Subsystem

The Original Images subsystem is principally concerned with the selection of control points in the original Reference and Sensed images. Image 1 corresponds to the Sensed Image and Image 2 corresponds to the Reference Image. The main display area consists of two windows to display the original images, a third window which displays the shift vectors over a copy of Image 1, a button area for a few common functions, and a drop down menu for special input/output functions (see Figure 4.5).

The Image 1 and Image 2 windows are used for manipulating control point pairs.[3] Each pair is denoted by a small coloured square in the two image windows, and in the shift vector window, by a coloured line from the original position in the Sensed image to a small coloured square marking the final position in the Reference image. The following interactions are available in either the Image 1 or the Image 2 windows:

Add Clicking with the left mouse button adds a control point pair.

Move The left mouse button can also be used to select and drag a control point to a new position in window. The control point position and associated shift vector is updated when the button is released.

Highlight Clicking over a control point with the middle mouse button changes the colour of the control point pair in the image windows and of the associated

[3] A control point pair consists of one control point from the Reference and the corresponding control point from the Sensed image.

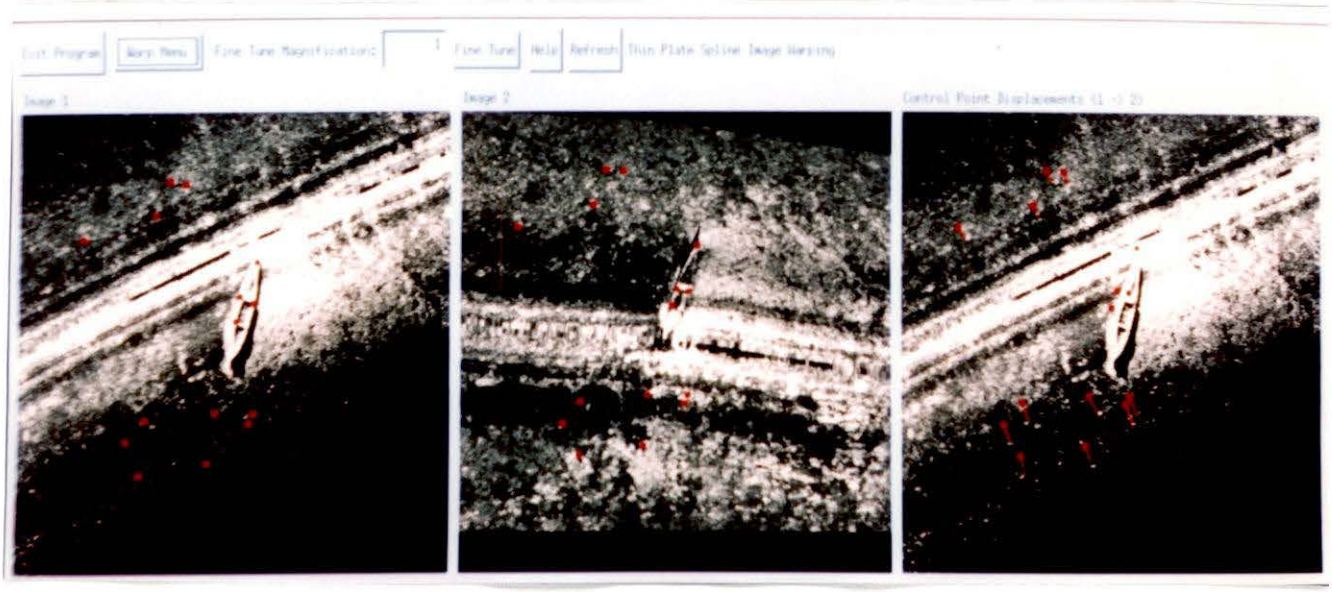

Figure 4.5: Side-Scan Sonar Change Detection System

line in the shift vector window to green. The green highlighting allows a user to easily identify a control point pair in all images.

Delete Clicking over a control point with the right mouse button deletes the corresponding pair.

The button area includes a few commonly used functions for controlling the Change Detection System. There are buttons for exiting the application, for selecting from the input/output menu bar, for launching the Image Warping subsystem and for creating fine tuning windows.

The fine tuning windows display full-size or magnified portions of the original images and have scroll bars which allow the user to view different parts of the large images. All of the mouse interactions in the main display windows also apply to the fine tuning windows. In addition, when the middle mouse button is used to highlight a control point pair, the fine tuning windows are each centered over their respective control points if the control point is not visible in the viewport.

The drop down menu is mainly designed for managing control point sets. There are entries for reading and writing control point sets, for erasing all control points and for temporarily hiding control points. The menu also allows the user to refresh the images or restart the system.

Image Warping Subsystem

Four warping methods are available: polynomial of degree 2, polynomial of degree 3, reverse TPS warping and forward TPS warping. Having four methods available allows the user to make comparisons between polynomial and TPS methods and between reverse and forward thin plate spline evaluation.

The polynomial methods apply a least squares surface fitting function provided with IDL using polynomials of degree 2 or 3 to warp the Sensed image (Image 1).

The forward TPS evaluation method calculates a new location in the warped image for each pixel in the original Sensed image. For each pixel in the warped image, the reverse TPS method calculates a location in the *original* image and takes the pixel value for the warped image from the pixel at the original location in the Sensed image. In [Dunham-Wilkie & Barrodale 91], there is an indication that the reverse TPS evaluation method may work better than the forward method because of the discrete nature of the images, but the choice between forward and reverse evaluation is still a matter for further research.

Once the warping methods have been selected, the user clicks on the "Warp" button to commence the image registration process. The system displays a window for the warped, Sensed image from each method, then calls subroutines to perform image registrations using the selected warping methods. When the subroutines have

completed, the warped images are displayed and the user can compare the warped images to the originals. By using the "Flicker" button, the user can display an overlay of a warped Sensed image (a warped Image 1) with the original Reference image (Image 2). The system continually flicks between the two images, displaying one, then the other.

4.2.3 Change Detection Summary

When comparing side-scan sonar images, exact registration in the presence of local distortions is an essential requirement. Without precise registration, it is difficult to find small differences between the images. As can be seen in Figure 4.6, thin plate splines are able to provide stable image warping in situations where polynomial approximation is unsuitable. [Skea & Barrodale 90, Dunham-Wilkie & Barrodale 91] show that thin plate splines were excellent registration functions for side-scan sonar images because thin plate splines can deal well with small local distortions and because thin plate splines are interpolating rather than approximating.

However, a good warping function is only one part of the process of image registration. As noted in [Goshtasby 88]:

> Determination of corresponding control points in the images is a very difficult and important step in the registration process.

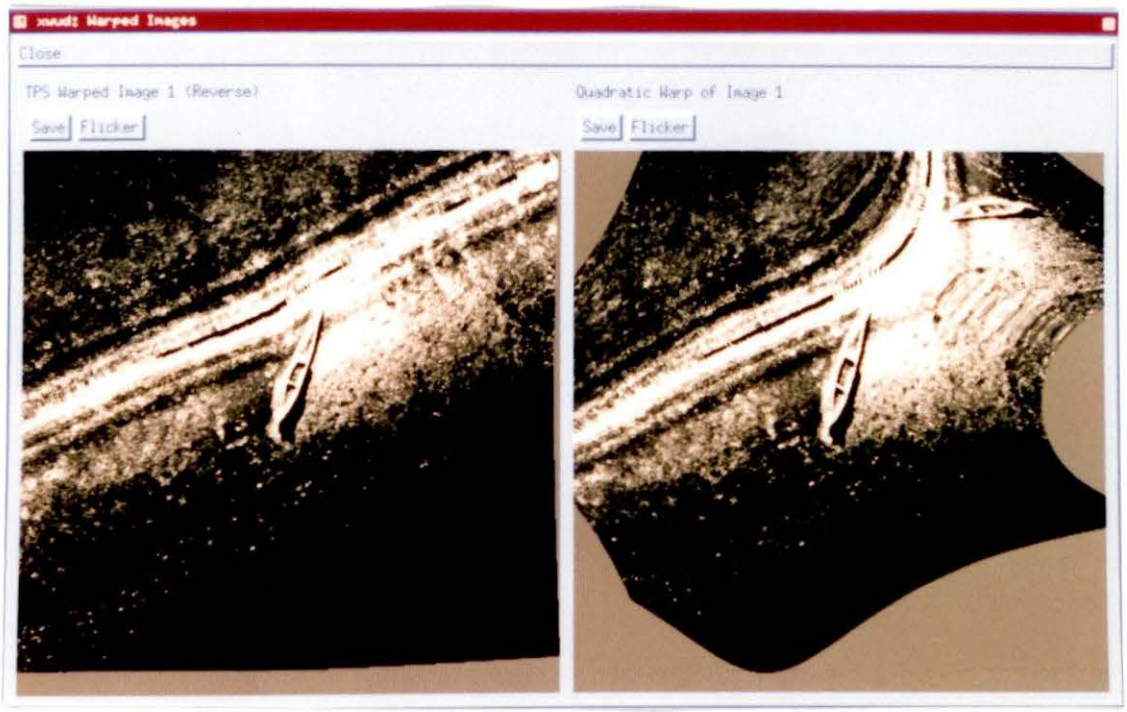

Figure 4.6: Polynomial vs Thin Plate Spline Image Warping

Once we have a good set of control points, thin plate splines have proven to be excellent interpolating registration functions, but getting a good set of control points is a difficult process. The Change Detection System has been designed to use scientific visualization techniques to make manual selection of control points as easy and as accurate as possible.

Finally, the Change Detection System provides interactive graphics to perform comparisons by overlaying the images and flicking back and forth. Because thin plate splines provide exact registration, this traditional flickering comparison technique works well with TPS warping.

Chapter 5

Conclusions

In this thesis, we have investigated ways of improving the condition number of TPS matrices and ways of dealing with interpolation difficulties using scientific visualization. In Chapter 2, we found that we could improve the stability of TPS matrices by scaling the matrix onto the unit square and by putting a minimum separation on the control points; in Chapter 3, we found some techniques that proved useful in detecting severe interpolation overshoot in TPS surfaces; and in Chapter 4, we described two applications of TPS warping, *Change Detection in Side-Scan Sonar Images* and *Survey Monument Registration.*

In Chapter 5, we will draw some conclusions about the usefulness of thin plate splines in the applications which we have studied; we will summarize some of our

findings about scientific visualization as applied to thin plate splines and we will introduce some ideas for future research.

5.1 Applicability of Thin Plate Spline Interpolation

In general, we found that TPS warping is well suited to side-scan sonar image registration because TPS functions can deal with the small local distortions that are present in sonar images and can register the small features we are interested in. As can be seen in Figure 4.6, traditional least-squares techniques are sensitive to distortions in the data.

In other applications, such as fitting scattered data, an approximating technique may be appropriate, but in change detection, exact registration of control points is desirable to simplify image comparison. Since TPS warping is an interpolating technique, the expected error for mapping control points is zero.

One significant problem with TPS interpolation for side-scan sonar image registration is the cost of evaluation. Calculating thin plate spline coefficients is relatively inexpensive, but naive evaluation of a TPS function with h control points for an $n \times n$ image requires the calculation of hn^2 logarithms, which is very costly. Fortunately, TPS evaluation can be closely approximated using a fast evaluation algorithm developed in [Powell 91]. This algorithm enables TPS image registration to

be performed orders of magnitude faster than naive evaluation, but without significant loss of accuracy, and makes image registration using TPS warping a practical solution.

The application of thin plate splines to survey monument registration was not as successful as the side-scan sonar application. As explained in Section 4.1.2, our difficulties are centred on the conflicting problem requirements. Because the NAD survey monuments are real physical monuments whose positions are known to considerable accuracy, exact interpolation of these points is required. However, there are not enough NAD monuments to accurately register the data, so additional, less accurate, control points are required. Inconsistencies in these control points and a lack of control points in some regions resulted in poor registration.

None of the other proposed solutions for the NAD registration problem are entirely satisfactory. For example, traditional least squares based approximation is severely affected by distortions in the data and does not warp NAD monuments exactly. Piecewise interpolation is being investigated to perform NAD data registration [Dewhurst 90], but this solution requires more analysis than TPS warping and will suffer from the lack of control points. In both of these example solutions, the problems are not insurmountable, but they will require the same kind of manual work to correct and augment the data that is required in TPS registration.

In hindsight, we can see that because of limitations in the currently available

data, the use of thin plate splines for NAD registration will require further, careful investigation.

5.2 Conclusions on Scientific Visualization

There are two main areas of research in scientific visualization: tools and interpretation. The majority of visualization research is focused on the development of graphical tools. This research includes advanced rendering techniques for data representation, graphical languages or interfaces for describing scientific data processing and special purpose display hardware (e.g. virtual reality).

The other main area of research is data interpretation. The focus is on finding ways of simplifying the analysis of complicated problems by presenting the data in a form that can be preprocessed by the human visual system. Research in this area draws together graphics, cognitive reasoning and scientific data analysis.

An important goal of this thesis was to learn something about data interpretation using scientific visualization by applying visualization techniques to thin plate spline interpolation. This involved analyzing our interpolation problem, categorizing the problem, deciding upon a display technique, sometimes writing the rendering program, extracting information from the data to be used in the rendering process, rendering and finally interpreting the results. The process is iterative, since new

ideas for graphical data analysis will arise from the final interpretation.

This approach gave us a broad introduction to scientific visualization from both the tools and interpretation research areas. We tried a variety of visualization techniques, some of which were useful in understanding our problem. We "discovered" some of the pitfalls to interpreting graphical information (e.g. discontinuous TPS second derivative surfaces which were rendered as continuous). In the end, we were successful in developing a visual technique for finding areas where TPS interpolation overshoot is particularly severe.

5.3 Areas for Future Research

In this thesis, we have shown that thin plate splines are effective warping functions, but to use them effectively, we have to pay attention to numerical instability and to some undesirable side-effects of interpolation (overshoot). As a consequence of this research, some questions have arisen.

5.3.1 Finding an Optimal Scaling Method

The scaling strategy which we outlined in Chapter 2 has proven to be effective in reducing the condition number of TPS matrices. In practice, we need graphical accuracy for image registration and our strategy allows us to obtain this level of

accuracy using double precision arithmetic. Our scaling strategy, however, is not necessarily optimal. Determining why the scaling strategy is effective would be an interesting theoretical problem, leading to a better understanding of TPS matrix ill-conditioning and possibly to an optimal scaling strategy.

5.3.2 Stiffness Parameters in Thin Plate Spline Equation

According to [Bookstein 89] a thin plate spline represents the equation of minimum bending energy for a thin metal plate. From an examination of the dimensionality of the terms one finds that there is a missing constant of elasticity, D_i, in the TPS equation:

$$f(x,y) = a_0 + a_1 x + a_2 y + \sum_{i=1}^{n} \frac{1}{2} b_i r_i^2 \log(D_i r_i)^2$$

Physically, the constant would be a measure of the stiffness of the thin metal plate. A smaller constant D_i would indicate a stiffer plate since the $r_i^2 \log(D_i r_i)^2$ terms would not fall off with increasing radius as rapidly with a small constant as with a large constant.

Currently, TPS registration functions use a stiffness constant of 1. An interesting question for further research would be whether this constant is useful in controlling the behaviour of registration using thin plate splines. In NAD data registration, for example, a smaller stiffness parameter might be useful in areas where there is a lack

of control points, since the influence of each control point would be spread over a larger region. Using different D_i values for different control points might lead to a useful weighted thin plate spline.

5.3.3 Use of a Z-Buffer in Visualizing Thin Plate Splines

One of the drawbacks to our use of IDL in this research was the lack of hidden-line removal or Z-Buffer capabilities, because many scientific visualization techniques depend upon some kind of Z-Buffer ability.

In the last few months, RSI has added a software Z-Buffer to IDL, which would allow us to explore some of the more advanced visualization techniques in TPS analysis. Possibilities include: point indicators on TPS surfaces, three dimensional plots of control point shift vectors, and three dimensional vector fields combined with flat TPS surface images. This last example would be a powerful way of associating TPS surface information with the distribution and consistency of control points.

5.3.4 Investigation of Overshoot Folds in Image Registration

Interpolation overshoot occurs when an interpolating curve goes outside the range of control point values. Figure 1.6 shows an example of overshoot in a two-dimensional spline.

In most applications of interpolation, overshoot is not a problem, but in image

registration even moderate overshoot is undesirable. As explained in the introduction (Chapter 1), if the amount of overshoot is greater than the spatial separation between the overshoot peak and a nearby control point, then the peak will be registered past the control point. In effect, a fold is introduced into the registration.

In Chapter 3, we investigated ways of visually detecting and dealing with severe interpolation overshoot. This method has been satisfactory and simple to perform, but it is an indirect method because it uses partial derivative surfaces to pinpoint large changes in the slope of the TPS surface rather than an actual measurement of interpolation overshoot. An interesting area of future research would be to find a more direct way of looking for interpolation folds. A visual detection scheme would certainly be useful, but perhaps a purely numerical approach could be found.

5.3.5 Use of Thin Plate Splines in Mammography

One possible application of image registration that was considered during the research for this thesis was the use of image warping to compare mammograms. Most of the work in mammography is concerned with detecting tumours from a single imaging session, not with detecting changes over time that might indicate new tumours. We wondered whether there was some way to use image registration and change detection techniques to look for tumours by making comparisons with previous images, rather than the interpretation of a single set of images.

When a mammogram is taken, the breast is squeezed between two plates, in order to make the resulting X-ray image as two dimensional as possible. This squeezing and distorting makes it very difficult to compare two images taken at different times. A straightforward solution would be to use thin plate splines to register the images. Thin plate splines are ideally suited to this type of application, since they can handle local distortions, twists and scaling changes and they provide accurate registration for small image features.

Unfortunately, this straightforward approach is basically flawed because it assumes that the images are two-dimensional objects. Since the breast is three dimensional, any two-dimensional projection (such as an X-ray) will depend upon the amount of shearing in the vertical direction. For example, in one image, three lumps may happen to line up in the vertical direction so that they appear to be one lump, but in the next image, they may be slightly misaligned and appear as three separate lumps or one elongated lump. Another set of lumps may be lined up horizontally in one direction in the first image and then reversed in the next image. Although TPS warping might be able to handle the first problem, we have seen in Chapter 3 that TPS warping could not handle the inconsistent shift vectors from the second example.

This is a very interesting problem with application to current research in three dimensional data representation. One possible solution would be three dimensional

registration using volume data (e.g. CAT scan) and a three dimensional radial basis. Whatever the eventual solution, three dimensional registration of mammography images would be an interesting area of research.

Bibliography

[Adams 86] Robert A. Adams. *Single-Variable Calculus, Revised Edition.* Don Mills: Addison-Wesley, 1986.

[Barrodale, Berkley & Skea 92] I. Barrodale, M. Berkley, & D. Skea. "Warping Digital Images Using Thin Plate Splines." Presented at the Seventh Texas International Symposium on Approximation Theory, Austin, Texas, January 3-7, 1992.

[Bookstein 89] F.L. Bookstein. "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations." *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. 11, No. 6, June 1989.

[Cheney & Kincaid 85] Ward Cheney & David Kincaid. *Numerical Mathematics and Computing*, 2nd ed. Pacific Grove: Brooks/Cole, 1985.

[Dewhurst 90] W.T. Dewhurst. *The application of Minimum Curvature-Derived Surfaces in the Transformation of Positional Data from the North American Datum of 1927 to the North American Datum of 1983.* NOAA Technical Memorandum NOS NGS-50. Rockville: National Geodetic Information Center, January 1990.

[Dunham-Wilkie & Barrodale 91] Michael Dunham-Wilkie & Ian Barrodale. *Investigation into Warping by Approximate Thin Plate Splines* Victoria: Barrodale Computing Services Ltd., 1991.

[Franke 81] R. Franke. *Smooth Interpolation of Scattered Data by Local Thin Plate Splines.* Monterey: Naval Postgraduate School, March 1981.

[Golub & Van Loan 89] G.H. Golub & C.F. Van Loan. *Matrix Computations*, 2nd Ed. Baltimore: Johns Hopkins, 1989.

[Goshtasby 88] A. Goshtasby. "Registration of Images with Geometric Distortions." *IEEE Transactions on Geoscience and Remote Sensing.* Vol. 26, No 1. January 1988.

[Goshtasby 91] A. Goshtasby. *Image Processing Techniques in Visualization.* San Diego: Visualization '91 Conference, October 1991.

[Harder & Desmarais 72] R.L. Harder & R.N. Desmarais. "Interpolation Using Surface Splines." *Journal of Aircraft.* Vol. 9, No 2. February 1972.

[McCormick et al. 87] McCormick, Bruce H. et al., ed. "Visualization in Scientific Computing." *Computer Graphics.* 21 6 (November 1987) New York: ACM SIGGRAPH.

[Powell 91] Powell, M.J.D. *Tabulation of Thin Plate Splines on a Very Fine Two-Dimensional Grid.* Presentation at Oberwolfach Meeting. 25-29 November 1991.

[Ragozin 91] D. L. Ragozin. Professor of Mathematics at University of Washington. Personal Interview at the Pacific Northwest Numerical Analysis Seminar, Boeing Computer Services, Bellevue, Washington. 21 September 1991.

[RSI 91] Research Systems Inc. *IDL User's Guide - Interactive Data Language Version 2.0* Boulder: RSI Inc., 1991.

[Robertson 91] P. K. Robertson. "A Methodology for Choosing Data Representations." *IEEE Computer Graphics and Applications.* May 1991.

[Schumaker 76] L.L. Schumaker. "Fitting Surfaces to Scattered Data." *Approximation Theory II.* ed. G.G. Lorentz, Ck. Chui & L.L. Schumaker. Boston:Academic Press, 1976.

[Sibson & Stone 91] R. Sibson & G. Stone. "Computation of Thin-Plate Splines." *SIAM J. Sci. Stat. Comput.* Vol. 12, No. 6 November 1991.

[Skea & Barrodale 90] D. Skea & I. Barrodale. *Sonar Image Matching.* Victoria: Barrodale Computing Services Ltd., 1990.

[Skea 92] D. Skea. *TPS Thematic Rectification Test.* Victoria: Minerva Research Ltd., 1991.

[Visualization 91] IEEE Computer Society. *Proceedings Visualization '91.* San Diego: IEEE Computer Society Press 1991.

[Watkins 90] D. Watkins. *Fundamentals of Matrix Computations.* Washington State University, 1990.

Vita

Surname: Berkley Given Names: James Michael

Place of Birth: Toronto, Ontario Date of Birth: January 22, 1959

Educational Institutions Attended:

Royal Roads Military College	1976 to 1979
University of Waterloo	1986 to 1990
University of Victoria	1990 to 1992

Degrees Awarded:

B.Math (Honours with Distinction)	University of Waterloo	1990

Honours and Awards:

Natural Sciences and Engineering Research Council Scholarship	1990 to 1991
University of Victoria President's Award	1990 to 1991

Partial Copyright License

Title of Thesis:

Thin Plate Splines, Image Registration and Scientific Visualization

Author: [redacted]

JAMES MICHAEL BERKLEY

April 5th, 1992