Graduate School Form 9
(Revised 8/89)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis Acceptance

This is to certify that the thesis prepared

By Robert L. Stevenson

Entitled

Invariant Reconstruction of Curves and Surfaces with Discontinuities with Applications in Computer Vision

Complies with University regulations and meets the standards of the Graduate School for originality and quality

For the degree of Doctor of Philosophy

Signed by the final examining committee:

Edward J. Delp, chair

Anthony A. Maciejewski

Saul Gass

J. R. Bell

Approved by:

Richard J. Schwartz
Department Head

6/5/90
Date

☐ is
This thesis ☒ is not to be regarded as confidential

Edward J. Delp
Major Professor

INVARIANT RECONSTRUCTION OF CURVES AND SURFACES WITH DISCONTINUITIES WITH APPLICATIONS IN COMPUTER VISION

A Thesis

Submitted to the Faculty

of

Purdue University

by

Robert Louis Stevenson

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 1990

This is dedicated to my wife Donna.

ACKNOWLEDGMENTS

I wish to thank my major professor Edward J. Delp whose patience, guidance, and understanding lead me through the perils and pitfalls of a doctorate program. I am also indebted to Professors Saul Gelfand, Anthony Maciejewski, Steven Bell and John Sadowsky for their advice.

I would also like to extend my appreciation to the many friends that I have made while studying at Purdue, especially Jisheng Song, Aly Farag, Cameron Wright, Roger McDannell, Qian Lin and Okan Azmak. Their support and friendly help made my studies much more enjoyable. And I especially thank my wife, Donna, for her continual understanding, patience, and support.

Finally, I am grateful to the National Science Foundation and the DuPont Corporation for supporting my work.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

Figures Page

Figures Page

Figures Page

Appendix
Figures

LIST OF TABLES

NOTATION

x	A scalar
x_o	A constant
$<x_1,x_2,...,x_n>$	An ordered n-tuple
$\mathbf{x}$	A vector
$\mathbf{x}_o$	A constant vector
$\mathbb{R}^n$	N-dimensional real arithmetic space
X	A domain, $X \subset \mathbb{R}^n$, $n \geq 1$
V^n	A general n-dimensional vector space
$f(x)$	A scalar function of a scalar, $x \in X$
$\mathbf{f}(x)$	A vector-valued function of a scalar, $x \in X$
$\mathbf{f}(\mathbf{x})$	A vector-valued function of a vector, $\mathbf{x} \in V^n$
$F(X)$	A general function space on X
$F_1(X)$	A dense subset of $F(X)$
$F^*(X)$	The normed conjugate space of F on X
$(\cdot,\cdot)_F$	An inner product on a space F, $(\cdot,\cdot)_F : F \times F \rightarrow \mathbb{R}$
$\rho_F(\cdot,\cdot)$	A metric on a space F, $\rho_F : F \times F \rightarrow \mathbb{R}$
$\|\cdot\|_F$	A norm on a space F, $\|\cdot\|_F : F \rightarrow \mathbb{R}$
$\|\|\|\cdot\|\|\|_F$	A seminorm on a space F, $\|\|\|\cdot\|\|\|_F : F \rightarrow \mathbb{R}$
$(\cdot,\cdot)$	Euclidean inner product (dot product)
$\|\cdot\|$	Euclidean norm on $\mathbb{R}^n$
$C^n(X)$	Space of n^{th}–order continuous functions on X
$\Pi_m(X)$	The space of m^{th} order polynomials on X
$L_2(X)$	Space of functions which are square integrable on X
$L_p(X)$	Space of functions which are p^{th} order integrable on X

$\chi_m(X)$	The m^{th} order Sobolev space on X
$\mathcal{BL}^m(X)$	The m^{th} order Beppo Levi space on X
$D^{\mathbf{p}}(\cdot)$	$D^{\mathbf{p}}(\cdot) = \left[\frac{\partial^{p_1}}{\partial x_{1_{p_1}}}\right]\left[\frac{\partial^{p_2}}{\partial x_2^{p_2}}\right]\cdots\left[\frac{\partial^{p_n}}{\partial x_n^{p_n}}\right](\cdot)$
$R(\cdot,\cdot)$	Regularizing operator
λ	Regularizing parameter
$\Omega[\cdot]$	Stablizing functional, $\Omega : F_1 \rightarrow I\!R$
$M^{\lambda}[\cdot,\cdot]$	Tikhonov stablizing functional, $M^{\lambda} : F_1 \times V^n \rightarrow I\!R$
$\mathcal{T}$	A triangulation of a set of data points
$\mathbf{u}$	parameter vector
u,v	parameters
$\mathbf{r}(\mathbf{u})$	parametric representation of an object
$z(\mathbf{u})$	explicit representation of an object
$\nu(u)$	velocity function
$\mathbf{t}(u)$	tangent vector
$\mathbf{n}(\mathbf{u})$	normal vector
$\kappa(u)$	curvature
$\kappa_n(\mathbf{u})$	normal curvature
$\mathbf{d}_1(\mathbf{u})$, $\mathbf{d}_2(\mathbf{u})$	principle directions
$\kappa_1(\mathbf{u})$, $\kappa_2(\mathbf{u})$	principle curvatures

ABSTRACT

Stevenson, Robert Louis. Ph.D., Purdue University, August 1990. Invariant Reconstruction of Curves and Surfaces with Discontinuities with Applications in Computer Vision. Major Professor: Edward John Delp.

The reconstruction of curves and surfaces from sparse data is an important task in many applications. In computer vision problems the reconstructed curves and surfaces generally represent some physical property of a real object in a scene. For instance, the sparse data that is collected may represent locations along the boundary between an object and a background. It may be desirable to reconstruct the complete boundary from this sparse data. Since the curves and surfaces represent physical properties, the characteristics of the reconstruction process differs from straight forward fitting of smooth curves and surfaces to a set of data in two important areas. First, since the collected data is represented in an arbitrarily chosen coordinate system, the reconstruction process should be invariant to the choice of the coordinate system (except for the transformation between the two coordinate systems). Secondly, in many reconstruction applications the curve or surface that is being represented may be discontinuous. For example in the object recognition problem if the object is a box there is a discontinuity in the boundary curve at the corner of the box.

The reconstruction problem will be cast as an ill-posed inverse problem which must be stablized using a priori information relative to the constraint formation. Tikhonov regularization is used to form a well-posed mathematical problem statement and conditions for an invariant reconstruction are given. In the case where coordinate system invariance is incorporated into the problem, the resulting functional minimization problems are shown to be nonconvex. To form a valid convex approximation to the invariant functional minimization problem a two step algorithm is proposed. The first step forms an approximation to the curve (surface) which is

piecewise linear (planar). This approximation is used to estimate curve (surface) characteristics which are then used to form an approximation of the nonconvex functional with a convex functional. Several example applications in computer vision for which the invariant property is important are presented to demonstrate the effectiveness of the algorithms.

To incorporate the fact that the curves and surfaces may have discontinuities the minimizing functional is modified. An important property of the resulting functional minimization problems is that convexity is maintained. Therefore, the computational complexity of the resulting algorithms are not significantly increased. Examples are provided to demonstrate the characteristics of the algorithm.

CHAPTER 1 - INTRODUCTION

1.1 Problem Addressed

In early computer vision the task is to extract symbolic descriptive information of objects in a scene from multi-dimensional sensory data. The form of the symbolic description (or model) will depend on the overall goal of the processing. For example in tracking problems the goal is to identify moving objects and to estimate their location and trajectory. In this simple case, the model may consist of just a sequence of locations which describe the object motion. In navigation problems the goal may be to identify the current location by matching a three-dimensional model of the terrain with a database of terrain information. The generation of the symbolic description from the sensor data is generally not an easy task. Often model parameters can not be measured directly from the sensor data. For example, in the problem of matching images of three-dimensional objects with a database of three-dimensional models, the sensor data is a two-dimensional intensity projection of the three-dimensional surfaces. The measurement of three-dimensional properties from this two-dimensional data is not straightforward. To overcome this problem it is often desirable to form an intermediate representation of the scene. The purpose of this representation is to bridge the gap between the sensor data and the symbolic description. The representation should be such that the symbolic description can be formed from the representation and the representation should be able to be extracted from the sensor data.

Since in computer vision applications we are dealing with geometric objects in three-dimensional space a common intermediate representation is either a curve in two- or three-dimensional space or a surface in three-dimensional space. For example, a curve in two-dimensional space may represent the boundary of an object or a surface in three-dimensional space may represent the surfaces of an object. A fundamental

problem in deriving these intermediate representations is extracting information from the sensor data. Over the years many algorithms have be developed to extract geometric information relative to the objects in a scene from various two-dimensional image sensors. A limitation of this type of processing however, is that only a noisy partial representation can be obtained from the data. That is, only an incomplete and noisy representation can be formed directly from the sensor data. Therefore another processing step is required to fill in the gaps in the representation and to remove noise artifacts in the representation.

In the simplest sense this can be thought of as a curve or surface interpolation or approximation problem. The first level of processing generates sparse geometric information describing a curve or surface from which it is desired to form a complete representation. Since the curves and surfaces represent physical properties, the characteristics of the reconstruction process differs from straight forward curve and surface reconstruction in two important areas. First, since the collected data is represented in an arbitrarily chosen coordinate system, the reconstruction process should be invariant to the choice of the coordinate system (except for the transformation between the two coordinate systems). This is important because a change in this intermediate representation may cause a change in the symbolic description that is derived, which in effect changes the knowledge that has been gathered about the objects in a scene. This obviously should not happen if all that has occurred is a change in an arbitrarily chosen coordinate system. Secondly, in many reconstruction applications the curve or surface that is being represented may be discontinuous. For example, in the boundary representation problem previously mentioned, if the object is a box there is a discontinuity in the boundary curve at the corner of the box.

Another concern when reconstructing curves and surfaces is the computational complexity of the reconstruction algorithms. Generally in computer vision problems the task that is being performed is in an dynamic environment. The sensors are often continually providing new data for processing so that decisions about the environment can be constantly updated. Therefore the processing time of an intermediate step needs

to be as small as possible. This puts a limitation on the type of processing and algorithms that can realisticly examined for this application.

The problem that is addressed in this research is much more generally applicable than just in the area of computer vision. The invariant approximation of objects with discontinuities from sparse data has many applications.

(i) In seismology, the density of the sedimentary layers of the earth's crust which have discontinuities between layers of different rock types.

(ii) In tomography, the density of an object may have discontinuities due to a change of material type.

(iii) In meteorology, air temperature has discontinuous derivatives at the boundary between the troposphere and the stratosphere.

In all of these examples, constraints are only collected at sparse locations and it is desired to obtain an estimate of some specific geometric parameter(s) everywhere which is invariant to the chosen coordinate system.

1.2 Previous Work

The problem of interpolating or approximating curves and surfaces has received much attention in the mathematical and statistical literature. Until recently, most research has concentrated on algorithms to fit continuous and smoothly varying curves or surfaces to a set of given data points [24, 52, 75]. Additional constraints on the reconstruction algorithm characteristics, such as invariance of the choice of the coordinate system and curves/surfaces with discontinuities, has become more important as the approach has been utilized in more complex applications.

In the visual surface reconstruction problem, Grimson [33] first recognized the need to incorporate discontinuity information into the surface reconstruction process. In the computer vision field Terzopoulos was the first to suggest a possible modification to Grimson's algorithm for including discontinuity information [79, 84]. Shiau [76] and Wahba [101] also proposed several methods for reconstructing curves and surfaces with

discontinuities. These methods were based on either a prior knowledge of the location of discontinuities or a preprocessing step which essentially performed a discontinuity (edge) detection to localize the discontinuities. This approach to the problem is limited because generally there is no a prior knowledge of the discontinuity locations and the fact that the detection of discontinuities from sparse noisy data will produce noisy and unreliable estimates of the discontinuity locations.

More recently several algorithms have been proposed which include the incorporation of discontinuity information as an integral part of the reconstruction process. Weiss [102-104], Lee [54, 55], and Aloimonos *et. al.* [3] have all proposed methods for reconstructing curves with discontinuities. These algorithms work well, however, it is not clear how they can be extended to the problem of reconstructing surfaces with discontinuities. For the problem of reconstructing surfaces, several algorithms based on statistical concepts have been proposed [12, 19, 61, 62]. It is discussed in Chapter 2 that these methods are generally computationally very expensive. An algorithm proposed by Sinha and Schunck [77] does not suffer from the limitations mentioned for the previously discussed algorithms, however, since their algorithm is based on characteristics which vary with the choice of coordinate system the reconstruction will not be invariant. In some applications this may be appropriate, but this will be a limitation for most applications in computer vision.

The property of invariance to the choice of coordinate system has been recognized as an important issue in computer vision. The computation of invariant curve reconstructions was reviewed by Malcolm [58]. However, as discussed by the author, all of these algorithms are instable, and depending on the input data set the algorithms may diverge. For the invariant surface reconstruction problem an algorithm has been proposed by Blake and Zisserman [11, 12]. Unfortunately experimental results have never been published, thus it is not possible to evaluate the effectiveness of the proposed algorithm. In Chapter 4, a adaptation of their algorithm is implemented to compare with the algorithm that is proposed in this research.

The open research problems that remain in the application of curve and surface reconstruction to problems in computer vision are:

1. the development of an algorithm which is essentially invariant to choice of coordinate system,

2. the development of a computationally efficient algorithm which will reconstruct curves and surfaces with discontinuities.

In this thesis both of these problems are addressed.

1.3 Scope and Organization

In this thesis, the emphasis will be on techniques and algorithms which formulate the problem as an ill-posed inverse problem which must be stablized. Ad hoc algorithms are ignored in favor of methods which rely on proven mathematical concepts. Whenever possible, an attempt is made to keep concepts general so that the algorithms can be easily adapted to similar problems in other fields. Chapter 2 begins by discussing the notation and conventions that will be used throughout the presentation. The nature of the abstract problem is then explored and regularizing techniques that will be used to make the problem well-posed and stable are presented. Chapter 3 investigates the invariant reconstruction of curves; a new approximately invariant algorithm is proposed and is shown to always converge to a stable representation. This algorithm is compared to two other proposed algorithms and is shown to to be much more robust relative to the choice of the coordinate system. Two example applications in computer vision are also presented. Chapter 4 examines the extension of these ideas to the invariant reconstruction of surfaces. A new algorithm is proposed, analyzed and compared to similar techniques. Chapter 5 examines how a prior information concerning discontinuities can be incorporated into the problem and how the algorithms can be modified so that discontinuities are automatically detected and incorporated. Results are summarized and possible future research is proposed in Chapter 6.

CHAPTER 2 - ABSTRACT PROBLEM SOLUTION

In this chapter the problem of curve and surface reconstruction is stated abstractly. The abstract problem is shown to be a mathematically ill-posed problem in the sense of Hadamard. Two common techniques for incorporating a prior information into ill-posed problems are examined and the reasons for using the methods of Tikhonov are discussed. Methods for approximating the resulting well-posed continuous problem with a discrete form are also described.

2.1 Definitions and Notations

In the Cartesian coordinate system the simplest way to describe a plane curve or a surface in three-dimensional space is to use the explicit form. With this form a plane curve is expressed by

$$z(x) \qquad x \in X \subset \mathbb{R} \tag{2.1}$$

and a surface in three-dimensions as

$$z(x,y) \qquad (x,y) \in X \times Y \subset \mathbb{R}^2 \tag{2.2}$$

This form is only satisfactory when the curve or surface takes on only single values and has no vertical tangents. This form is unable to represent curves or surfaces in higher dimensional spaces. For example, a curve is three-dimensional space cannot be described using the explicit form.

An alternative way of representing curves and surfaces which does not have these limitations is a parametric form. A concise notation for the parametric description is obtained by using vector-valued functions defined on vector spaces. If an object has n degrees-of-freedom (e.g. for a curve n=1 and for a surface n=2) the coordinates of the object in a m-dimensional space can be expressed as a m-dimensional vector-valued

function, $\mathbf{r}$, of an auxiliary parameter $\mathbf{u} \in U \subset \mathbb{R}^n$. A convenient notation for a vector-valued function is as an ordered m-tuple of scalar-valued functions written

$$\mathbf{u} = \langle u_1, u_2, \ldots, u_n \rangle \tag{2.3}$$

$$\mathbf{r}(\mathbf{u}) = \langle x_1(\mathbf{u}), x_2(\mathbf{u}), \ldots, x_m(\mathbf{u}) \rangle \tag{2.4}$$

For instance a curve in three-dimensional space can be written as

$$\mathbf{r}(u) = \langle x(u), y(u), z(u) \rangle \qquad u \in U \subset \mathbb{R}. \tag{2.5}$$

Similarly, a surface in three-dimensional space can be written

$$\mathbf{u} = \langle u, v \rangle \tag{2.6}$$

$$\mathbf{r}(\mathbf{u}) = \langle x(\mathbf{u}), y(\mathbf{u}), z(\mathbf{u}) \rangle. \tag{2.7}$$

Notice that the explicit form can be thought of as a special case of the parametric form if the following parameterization is used

$$\begin{aligned} x_1(\mathbf{u}) &= u_1 \\ x_2(\mathbf{u}) &= u_2 \\ &\vdots \\ x_{m-1}(\mathbf{u}) &= u_{m-1} \\ x_m(\mathbf{u}) &= z(\mathbf{u}) \end{aligned} \tag{2.8}$$

The function z is the explicit form of the object with $m-1$ degrees of freedom in an m-dimensional space.

These representations can be classified into spaces of functions based on the mathematical characteristics of the functions. The class of functions which are continuous at every point of the domain X is denoted by $C(X)$. The class of functions which are continuous and have a continuous derivative at every point of X is denoted by $C^1(X)$. The space of bivariate polynomials with degree $\leq m$ will be denoted by $\prod_m$.

If $F(X)$ is some space of functions over the real domain X, an inner product on $F(X)$ is a real-valued function $(\cdot,\cdot)_F : F \times F \rightarrow \mathbb{R}$ which has the properties

1. $(f,g)_F = (g,f)_F$, for all $f,g \in F(X)$.

2. $(\alpha f_1+\alpha f_2,g)_F = \alpha(f_1,g)_F + \alpha(f_2,g)_F$, for all $f_1,f_2,g \in F(X)$, and $\alpha \in \mathbb{R}$,

3. $(f,f)_F \geq 0$
   and $(f,f)_F = 0 \iff f = 0$, for all $f \in F(X)$,

a norm on $F(X)$ can be defined by

$$||f||_F = (f,f)_F^{1/2}, \tag{2.9}$$

and the norm $||\cdot||_F$ on $F(X)$ defines a metric on F by

$$||f-g||_F, \qquad \text{for all } f,g \in F(X) \tag{2.10}$$

A function space with a suitably defined metric is referred to as a metric space [57].

The class of spaces most often applied to the problem of object reconstruction is a subfamily of the Sobolev spaces [2]. Sobolev spaces of partially differentiable functions are Banach spaces which generalize the well known L_p spaces. Of interest are the Hilbert spaces formed by generalizing L_2. The space L_2 consist of all functions $f: X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ whose L_2 norm over X,

$$||f||_{L_2} = \left[\int_X |f(\mathbf{x})|^2 d\mathbf{x}\right]^{1/2}, \tag{2.11}$$

is finite. The partial derivatives of f will be denoted by

$$D^{\mathbf{p}} f(\mathbf{x}) = \left[\frac{\partial^{p_1}}{\partial x_1^{p_1}}\right]\left[\frac{\partial^{p_2}}{\partial x_1^{p_2}}\right]\cdots\left[\frac{\partial^{p_n}}{\partial x_1^{p_n}}\right] f(\mathbf{x}) \tag{2.12}$$

where $\mathbf{p} = <p_1,p_2,\cdots,p_n>$ is an index vector which indicates the order of the partial derivative.

The Sobolev spaces, $\chi_m(X)$, are the completion of the vector space of functions which have all partial derivatives of order m in $L^2(X)$. That is, $\chi_m(X)$ is the completion

of the vector space

$$\left\{ f \in C^m(X) : \|D^{\mathbf{p}} f\|_{L_2} < \infty, \text{ for all } |\mathbf{p}| \le m \right\}, \tag{2.13}$$

with respect to the Sobolev norm defined below. Where $|\mathbf{p}| = p_1 + p_2 + \cdots + p_n$ and $C^m(X)$ is the space of continuous functions with m^{th} order partial derivatives. The associated Sobolev norm is

$$\|f\|_{\chi_m} = \left[\sum_{|\mathbf{p}| \le m} \|D^{\mathbf{p}} f\|_{L_2}^2 \right]^{1/2}. \tag{2.14}$$

The Sobolev spaces have the properties that $\chi_0(X) = L_2(X)$ and

$$\chi_m(X) \subset \chi_{m-1}(X), \qquad \text{for all } m>0. \tag{2.15}$$

A scalar product can be defined for any f_1 and f_2 in $\chi_m(X)$ by

$$(f_1, f_2)_{\chi_m} = \sum_{|\mathbf{p}| \le m} (D^{\mathbf{p}} f_1, D^{\mathbf{p}} f_2)_{L_2}, \tag{2.16}$$

where $(\cdot,\cdot)_{L_2}$ is the scalar product of $L_2(X)$. Since the scalar product is the sum of the scalar products in L_2, the Sobolev spaces are Hilbert spaces. A seminorm on $\chi_m(X)$ can be defined by

$$|||f|||_{\chi_m} = \left[\sum_{|\mathbf{p}| = m} \|D^{\mathbf{p}} f\|^2 \right]^{1/2}, \qquad f \in \chi_m(X). \tag{2.17}$$

It is a seminorm because it is zero if $f \in \prod_{m-1}(X)$. The space of functions formed by the completion of the vector space

$$\left\{ f \in C^m(X): \|D^{\mathbf{p}} f\|_{L_2} < \infty, \text{ for all } |\mathbf{p}| = m \right\} \tag{2.18}$$

with respect to the seminorm (2.17) is a semi-Hilbert space, which is commonly referred to as the Beppo Levi space of order m, $\mathcal{BL}^m(X)$ [65]. Examples of other plausible classes that may be appropriate for surface reconstruction can be found in

[14, 15].

A functional can be defined on a function space. For the work in this thesis mostly scalar functionals will be used, that is $W:F(X) \rightarrow \mathbb{R}$, where $F(X)$ is some function space and W is some functional defined on $F(X)$. An important issue is this research is related to the convexity of the functionals that will be examined. A functional W is convex on the function space $F(X)$ if

$$W[\alpha f + (1-\alpha)g] \leq \alpha W[f] + (1-\alpha)W[g], \qquad f,g \in F(X), \alpha \in [0,1]. \quad (2.19)$$

If this is not true than a function is referred to as nonconvex. This property will be important because it is related to the computational efficiency of the algorithms. The task of minimizing nonconvex functionals usually requires much more computational complexity than that of minimizing a convex functional.

2.2 Abstract Problem Statement

Let an object with n–degrees-of-freedom in an m-dimensional space have the parametric representation $\mathbf{r}(\mathbf{u})$, where $\mathbf{r}(\mathbf{u})$ is in the vector space V^m and $\mathbf{u} \in U \subset \mathbb{R}^n$. Low-level computational vision processes, which are capable of measuring characteristics of the object, generate a collection of noise corrupted shape estimates, $S = \{\mathbf{c}_i(\mathbf{u}) \in C_i, i=1,...,M\}$. These shape estimates usually consist of constraints on location and/or normals at specific points on the surface or curve. However, constraints may also be along curves in m-dimensional space or over regions of the object. These types of constraints may represent a prior knowledge about a boundary condition or knowledge of some characteristic over a limited domain.

The process of constraint formation is modeled by

$$\mathbf{c}_i(\mathbf{u}) = \mathbf{l}_i[\mathbf{r}(\mathbf{u})] + \mathbf{e}_i(\mathbf{u}), \qquad i=1,...,M \quad (2.20)$$

where $\mathbf{l}_i : V^m \rightarrow C_i$ denotes measurement functionals of $\mathbf{r}(\mathbf{u})$, and $\mathbf{e}_i(\mathbf{u})$ models the associated measurement errors. The functional $\mathbf{l}_i$ may be either scalar- or vector-valued depending of the particular type of measurement.

The object reconstruction problem, simply stated, is to form an estimate, $\hat{\mathbf{r}}(\mathbf{u})$, of the object $\mathbf{r}(\mathbf{u})$ for all $\mathbf{u} \in U$, given the finite collection of noisy and sparse shape constraints, $\mathcal{S}$.

If the object estimate is to be useful for modeling purposes it should possess the following properties. First, the reconstruction algorithm should be consistent with the collection of constraints. Secondly, if the coordinate system is rotated and/or translated then the estimate of this new collection of constraints should be identical to the original surface estimate, $\hat{\mathbf{r}}(\mathbf{u})$, rotated and translated. Finally, depth and orientation discontinuities should also be made explicit in the representation.

The problem as stated is much more general than that needed for either curve or surface reconstruction. For the reconstruction of a surface in three-dimensional space, n=2, and $\mathbf{u}$ is represented by $\mathbf{u} = <u,v> \in U \subset \mathbb{R}^2$. The parametric representation is given by

$$\mathbf{r}(\mathbf{u}) = \mathbf{r}(u,v) = <x(u,v), y(u,v), z(u,v)>. \tag{2.21}$$

In computer vision the constraints are generally in the form of information relative to the surface's or curve's location in space or information about the normal to the surface or curve at a point. Thus, $\mathbf{l}_i$ will generally have one of the following forms

$$\mathbf{l}_i[\mathbf{r}(u,v)] = \mathbf{r}(u_i,v_i), \tag{2.22}$$

$$\mathbf{l}_i[\mathbf{r}(u,v)] = \frac{\mathbf{r}_u(u_i,v_i)\times\mathbf{r}_v(u_i,v_i)}{\|\mathbf{r}_u(u_i,v_i)\times\mathbf{r}_v(u_i,v_i)\|} \tag{2.23}$$

where $<u_i,v_i>$ is a constant vector which indicates where on the surface the measurement is recorded, and $\mathbf{r}_u(\mathbf{u})$ represents the derivative of the vector with respect to the parameter u.

Similarity, for a curve, n=1, and $\mathbf{u}$ is represented by a scalar u. $\mathbf{l}_i$ will generally have one of the following forms

$$\mathbf{l}_i[\mathbf{r}(u)] = \mathbf{r}(u_i), \tag{2.24}$$

$$\mathbf{l}_i[\mathbf{r}(u)] = \frac{\langle y_u(u_i), -x_u(u_i)\rangle}{(x_u^2(u_i)+y_u^2(u_i))^{1/2}} \tag{2.25}$$

Extensions to other objects with more degrees-of-freedom in higher dimensional spaces can be derived as needed by the application.

2.2.1 Ill-Posed Problems

For the problem to be mathematically well-posed, in the sense of Hadamard [36], on the pair of metric spaces (F,C), it must meet the following criteria:

(i) for every element $c \in C$ there exists a solution f in the space F;

(ii) the solution, f, is unique in F;

(iii) the solution, f, depends continuously on c.

If the problem does not meet these conditions, it is known as ill-posed. Condition (iii) is related to the stability, or robustness, of the solution in the presence of noise. Continuity is a necessary condition for stability, but not sufficient. If a solution is also stable the problem is said to be well-conditioned, otherwise the problem is ill-conditioned.

The reconstruction problem as it is currently stated is ill-posed. The sparse set of estimates only constrain the shape of the object locally; thus an infinite number of feasible objects will fit as estimates. In some cases, overlapping and conflicting estimates locally overdetermine the shape, leading to no solution. Lastly, derivatives may not depend continuously on the data since small perturbations in the estimated data may cause the solution to be dramatically affected.

To obtain a unique and stable reconstruction of the object in the scene, supplementally information must be added so that the problem becomes well-posed. The basic principle common to all methods is to use a priori knowledge about the constraint formation process and the shape measurement techniques to resolve conflicts in estimates, and to restrict the space F so that the constraints, $\mathcal{S}$, uniquely determine a

stable estimate, $\hat{\mathbf{r}}(\mathbf{u})$, of the surface. Several techniques have been developed to integrate the a prior information with the experimental information to form a well-posed mathematical problem statement.

2.3 Solution of Ill-Posed Problems

This section examines two well known techniques for regularizing ill-posed problems.

2.3.1 Tikhonov Regularization

In the early 1960's Tikhonov introduced the concept of the regularizing operator [88] which laid the foundation for a new approach to the solution of ill-posed problems [87-92, 94]. The monograph of Tikhonov and Arsenin [93] consolidates the work done before that time and established the theory as a standard technique for solving ill-posed problems. To make the problem well-posed a continuous operator, known as a regularizing operator, is defined which approximates the inverse operator.

Let $\mathbf{c}_{T,i}(\mathbf{u})$ be the exact left hand member of (2.20) (i.e. $\mathbf{c}_{T,i}(\mathbf{u}) = \mathbf{l}_i[\mathbf{r}(\mathbf{u})]$), and let $\mathbf{c}_{\delta,i}(\mathbf{u})$ be an element of C_i such that

$$\sum_{i=1}^{M} \rho_{C_i}(\mathbf{c}_{T,i}(\mathbf{u}), \mathbf{c}_{\delta,i}(\mathbf{u})) < \delta, \tag{2.26}$$

where $\rho_{C_i}(\cdot,\cdot)$ is a metric on the space C_i. Let the collection of $\mathbf{c}_{T,i}(\mathbf{u})$ be denoted by $\mathcal{S}_T$ and the collection of $\mathbf{c}_{\delta,i}(\mathbf{u})$ be denoted by $\mathcal{S}_\delta$. Denote the inverse operator (i.e. the reconstruction algorithm) by $\mathbf{l}^{-1}(\cdot) : C_1 \times \cdots \times C_M \rightarrow V^m$. For the solution to be stable it is expected that the solution $\mathbf{r}_\delta(\mathbf{u})$ obtained from $\mathcal{S}_\delta$ through the inverse operator should in some sense be close to $\mathbf{r}(\mathbf{u})$ and for

$$\mathbf{r}_\delta(\mathbf{u}) = \mathbf{l}^{-1}(\mathcal{S}_\delta) \quad \rightarrow \quad \mathbf{r}(\mathbf{u}) = \mathbf{l}^{-1}(\mathcal{S}_T) \quad \text{as } \delta \rightarrow 0 \tag{2.27}$$

(i.e. $\rho_{V^m}(\mathbf{r}_\delta(\mathbf{u}), \mathbf{r}(\mathbf{u})) \rightarrow 0$ as $\delta \rightarrow 0$). To formalize this concept, a regularization operator was proposed by Tikhonov and Arsenin [93].

Definition 2.1. An operator $R(\mathcal{S}, \lambda)$ is known as a regularizing operator of the equation

$\mathbf{c}_i(\mathbf{u}) = \mathbf{l}_i[\mathbf{r}(\mathbf{u})] + \mathbf{e}_i(\mathbf{u})$ in a neighborhood of $\mathbf{c}_i(\mathbf{u}) = \mathbf{c}_{T,i}(\mathbf{u})$ if

1. there exists a positive number δ_1 such that the operator $R(\mathcal{S},\lambda)$ is defined for every λ and every $\mathcal{S}$ for which

$$\sum_{i=1}^{M} \rho_{C_i}(\mathbf{c}_i(\mathbf{u}), \mathbf{c}_{T,i}(\mathbf{u})) \le \delta \le \delta_1, \tag{2.28}$$

and

2. there exists a function $\lambda = \lambda(\delta)$ such that, for every $\varepsilon>0$, there exists a number $\delta(\varepsilon) \le \delta_1$ such that the inclusion $\mathcal{S}_\delta$ and the inequality

$$\sum_{i=1}^{M} \rho_{C_i}(\mathbf{c}_{T,i}(\mathbf{u}), \mathbf{c}_{\delta,i}(\mathbf{u})) \le \delta(\varepsilon) \tag{2.29}$$

imply

$$\rho_{V^m}(\mathbf{r}(\mathbf{u}), \mathbf{r}_\lambda(\mathbf{u})) \le \varepsilon, \tag{2.30}$$

where $\mathbf{r}_\lambda(\mathbf{u}) = R(\mathcal{S}_\delta, \lambda(\delta))$.

□

The solution $\mathbf{r}_\lambda(\mathbf{u})$ is an approximate solution of equation (2.20) and is known as the regularized solution. The solution is a function of λ and $\mathcal{S}_\delta$ as well as the form of the regularizing operator $R(\cdot,\cdot)$. Thus in the noise free case (i.e. $\mathcal{S}_T$ is known), there exists a sequence of parameters, $\{\lambda_i\}_{i=0}^{\infty}$, such that the regularized solution, $\{\mathbf{r}_{\lambda_i}(\mathbf{u})\}_{i=0}^{\infty}$, approaches the true solution, (i.e. $\mathbf{r}_{\lambda_i}(\mathbf{u}) \to \mathbf{r}(\mathbf{u})$ with respect to ρ_{V^m}). When noise corrupts the data, the numerical parameter λ can be adjusted to provide a tradeoff between approximation accuracy and the affects of noise. The parameter λ is known as the regularization parameter and its value must be determined from a priori information.

Thus to obtain a regularized inverse, a regularizing operator $R(\cdot,\cdot)$ and the value of the regularization parameter λ must be determined. Note that the regularizing operator for a particular problem is in general not unique; there may exist many operators which

stabilize the ill-posed problem. The choice of the particular operator and the value of the regularization parameter λ is based on supplementary information pertaining to the problem.

To construct a regularizing operator, Tikhonov and Arsenin [93] introduce a stablizing functional, $\Omega[\cdot] : V^m \rightarrow I\!R$, with the following properties: (i) the functional stablizing $\Omega[\cdot]$ is a continuous nonnegative functional defined on a subset V_1^m of V^m that is everywhere dense in V^m, (ii) the solution $\mathbf{r}(\mathbf{u})$ belongs to the domain of definition of $\Omega[\cdot]$, and (iii) for every positive number d, the set of elements $\mathbf{r}^*(\mathbf{u})$ of V_1^m for which $\Omega[\mathbf{r}^*(\mathbf{u})] \leq d$ is a compact subset of V_1^m. This stablizing functional is used to define the functional

$$M^{\lambda}[\mathbf{r}^*(\mathbf{u}),\mathcal{S}] = \Omega[\mathbf{r}^*(\mathbf{u})] + \lambda \sum_{\mathbf{c}_i \in \mathcal{S}} \rho^2_{C_i}(\mathbf{l}_i[\mathbf{r}^*(\mathbf{u})],\mathbf{c}_i) \tag{2.31}$$

Then for certain values of λ the minimization of this functional over V_1^m is a regularizing operator,

$$\mathbf{r}_{\lambda}(\mathbf{u}) = R(\mathcal{S},\lambda), \tag{2.32}$$

where $\mathbf{r}_{\lambda}(\mathbf{u})$ is such that

$$M^{\lambda}[\mathbf{r}_{\lambda}(\mathbf{u}),\mathcal{S}] = \inf_{\mathbf{r}^*(\mathbf{u}) \in V_1^m} M^{\lambda}[\mathbf{r}^*(\mathbf{u}),\mathcal{S}]. \tag{2.33}$$

This method is generally known as Tikhonov regularization.

In summary, to find a regularized solution to an ill-posed problem, a metric, ρ_{C_i}, on each C_i and a stablizing functional, $\Omega[\cdot]$, must be specified based on a priori information. Then an appropriate λ must be found such that the minimization of $M^{\lambda}[\mathbf{r}^*(\mathbf{u}),\mathcal{S}]$ is a regularizing operator. The choice of λ will be based on the choice of the stablizer, the metrics on the C_i's, and supplementary information pertaining to noise in the data. Once this information is obtained, the regularized solution, $\mathbf{r}_{\lambda}(\mathbf{u})$, to the ill-posed problem is determined by minimizing the functional $M^{\lambda}[\mathbf{r}^*(\mathbf{u}),\mathcal{S}]$ in the space V_1^m. The hardest task is to find a stablizer which not only yields a unique and stable solution to the inverse problem, but also accurately measures the consistency of the

estimate with respect to the true solution.

If it is assumed that the $\mathbf{l}_i$'s are linear, the ρ_{C_i} are quadratic, and Ω is quadratic then it can be shown that the solution space is convex and a unique solution exists [93]. Most applications of this technique will define V_1^m to be a Hilbert or semi-Hilbert space and the stablizer as some norm on that space. When this is not the case then the functional may be nonconvex. This makes finding the optimal solution more difficult since there may exist many suboptimal local minimum.

2.3.2 Stochastic Regularization

The stochastic solution to ill-posed problems is a straight forward application of Bayesian estimation. The Tikhonov method makes the problem well-posed by restricting the space, V^m, of possible solutions to a space, V_1^m (a dense subset of V^m), so that a stable and unique solution can be found. The manner in which this restriction is made is based on a priori information. In contrast, a stochastic approach uses a priori information relative to the likelihood of a function, $\mathbf{r}^*(\mathbf{u})$, being a solution to define a probability distribution, $P_{\mathbf{r}^*}$, on the space V^m. A priori information about the observation noise process is used to determine a conditional probability distribution, $P_{\mathcal{S}|\mathbf{r}^*}$. Using these distributions, the posterior probability distribution can be obtained by

$$P_{\mathbf{r}^*|\mathcal{S}} = \frac{P_{\mathcal{S}|\mathbf{r}^*} P_{\mathbf{r}^*}}{P_{\mathcal{S}}} \tag{2.34}$$

which represents the likelihood of a solution, $\mathbf{r}^*$, given that the data, $\mathcal{S}$, was observed. If noise from different measurement techniques are independent, the joint posterior distribution can easily be computed using

$$P_{\mathcal{S}} = \prod_{i=1}^{M} \frac{P_{\mathbf{c}_i|z} P_z}{P_{\mathbf{c}_i}}. \tag{2.35}$$

Thus there is a natural way to integrate information from various sources. An estimate, $\hat{\mathbf{r}}(\mathbf{u})$, can then be found with either a MAP estimator or by defining a loss functional on

V^m and computing a Bayesian estimate. The MAP estimate is found by simply maximizing the probability distribution (2.35) over the space, V^m, to find the function, $\hat{\mathbf{r}}(\mathbf{u})$, which is the most likely solution given the data, $\mathcal{S}$. To find the Bayes' estimate, a loss functional, $L : V^m \rightarrow \mathbb{R}$, must be defined. This loss functional may be used to incorporate more a priori knowledge into the algorithm. The estimate is then the function, $\hat{\mathbf{r}}(\mathbf{u})$, which minimizes the expected loss with respect to (2.35).

In summary, to make the problem well-posed, a probabilistic model on the space of possible solutions must be specified based on a priori information. The quality of the solution will depend largely on the quality of the model chosen; thus it is critical that the model accurately reflect the true space of surfaces. The estimated solution is then found by minimizing (or maximizing) a functional. This, however, is often not easy since even very simple probabilistic models usually result in nonconvex functionals. For computer vision problems this method was used extensively by Marroquin [61, 62].

2.4 A Well-Posed Reconstruction Problem

To devise a mathematically well-posed problem statement from the currently ill-posed problem statement given in Section 2.2, either of the techniques described in Section 2.3 can be used. However, as discussed in Section 2.3.2 when using the stochastic regularizing techniques the resulting mathematical problem is often the minimization of a nonconvex functional. Since nonconvex functionals may have many local minimum this task is computational very expensive. In computer vision problems the reconstruction of curves and surfaces is generally an intermediate process in a more complex algorithm. In most computer vision applications it is desirable that the algorithms run in close to real-time, thus computational complex steps need to avoided. For this reason this study will concentrate on the Tikhonov regularization approach for making problems well-posed and convexity issues will be emphasized. Recall that the functional

$$M^{\lambda}[\mathbf{r}^*(\mathbf{u}), \mathcal{S}] = \Omega[\mathbf{r}^*(\mathbf{u})] + \lambda \sum_{\mathbf{c}_i \in \mathcal{S}} \rho^2_{C_i}(\mathbf{l}_i[\mathbf{r}^*(\mathbf{u})], \mathbf{c}_i) \tag{2.36}$$

must be defined based on a prior information.

Defining a functional of this form does not guarantee that this new problem statement is well-posed. This fact must be proved. The functionals examined in this thesis all will have the form

$$M^{\lambda}[\mathbf{r}(\mathbf{u}),\mathcal{S}] = \frac{1}{2}a[\mathbf{r}(\mathbf{u}),\mathbf{r}(\mathbf{u})] - f[\mathbf{r}(\mathbf{u}),\mathcal{S}] \tag{2.37}$$

where $a(\cdot,\cdot):F\times F\rightarrow\mathbb{R}$ is a continuous bilinear form and $f(\cdot,\mathcal{S}):F\rightarrow\mathbb{R}$ is a continuous linear form. Conditions for a well-posed problem are given by the following theorem.

Theorem: [21] There exists a unique solution, $\hat{\mathbf{r}}(\mathbf{u})$ to the minimization of (2.37) if

1. the admissible space F is a Hilbert space (with norm $\|\cdot\|_F$),
2. the bilinear form $a(\cdot,\cdot)$ is symmetric, $a[\mathbf{r}(\mathbf{u}),\mathbf{s}(\mathbf{u})] = a[\mathbf{s}(\mathbf{u}),\mathbf{r}(\mathbf{u})]$,
3. $a(\cdot,\cdot)$ is F–elliptic, i.e. there exists a constant $\alpha > 0$ such that

$$a[\mathbf{r}(\mathbf{u}),\mathbf{r}(\mathbf{u})] \geq \alpha\|\mathbf{r}(\mathbf{u})\|_F^2 \tag{2.38}$$

□

The proof of this theorem can be found in [21]. In this work, the functionals will be minimized over second order Sobolev spaces, $\chi_2(X)$, which are Hilbert spaces, so condition 1 of the theorem is met. Conditions 2 and 3 are met if $a(\cdot,\cdot)$ is an inner product on the space. In this work $a(\cdot,\cdot)$ will have the following form

$$a[\mathbf{r}(\mathbf{u}),\mathbf{s}(\mathbf{u})] = \sum_{p=2}\sum_{q=2}(w_{pq}(\mathbf{u})D^{p}\mathbf{r}, w_{pq}(\mathbf{u})D^{q}\mathbf{s})_{L_2} + \lambda\sum_{\mathbf{c}_i\in\mathcal{S}}(\mathbf{r}(\mathbf{u}_i),\mathbf{s}(\mathbf{u}_i)), \tag{2.39}$$

where $w_{pq}(\mathbf{u})$ are constant functions which are determine by the particular problem.

Proposition: For the reconstruction of curves: if the set of constraints, $\mathcal{S}$, contains

at least two nonconsequential points, then (2.39) is an inner product on the Sobolev spaces.

Proof: To prove the propositions it needs to be shown that (2.39) satisfies the three properties of an inner product.

1. $a(\cdot,\cdot)$ is symmetric by inspection.

2. $a(\cdot,\cdot)$ is bilinear by inspection.

3. To show property three, it is sufficient to show that $a[\mathbf{r}(\mathbf{u}),\mathbf{r}(\mathbf{u})] = 0$ only if $\mathbf{r}(\mathbf{u}) = 0$. The first term in (2.39) is the square of a semi-norm on a second order Beppo Levi space. Therefore, its null space is $\prod_1$, i.e. all lines. Choose $\lambda > 0$ by definition, then the second term is always positive and is only zero if $\mathbf{r}(\mathbf{u}_i) = 0$ for all data points. If $\mathcal{S}$ contains two nonconsequential data points then $a[\mathbf{r}(\mathbf{u}),\mathbf{r}(\mathbf{u})] = 0$ only if $\mathbf{r}(\mathbf{u}) = 0$.

Thus $a(\cdot,\cdot)$ is an inner product on $\chi_2(X)$. □

Therefore, the curve reconstruction problem is well posed if $a(\cdot,\cdot)$ has the form of (2.39). Similarly, for the surface reconstruction problem a proposition can be proved showing that the minimization problem is well-posed if $\mathcal{S}$ contains three nonlinear data points.

It is also important for M^λ to be convex, so that computationally efficient minimization algorithms can be developed.

Proposition: If M^λ is of the form

$$M^\lambda[\mathbf{r}(\mathbf{u}),\mathcal{S}] = \frac{1}{2}a[\mathbf{r}(\mathbf{u}),\mathbf{r}(\mathbf{u})] - f[\mathbf{r}(\mathbf{u}),\mathcal{S}] \tag{2.40}$$

where $a(\cdot,\cdot)$ is a continuous bilinear form and $f(\cdot,\mathcal{S})$ is a continuous linear form. Then

M^{λ} is convex.

Proof: The parameter **u** is removed for convenience.

$$M^{\lambda}[\alpha\mathbf{r}+(1-\alpha)\mathbf{s},\mathcal{S}] = \frac{1}{2}a(\alpha\mathbf{r}+(1-\alpha)\mathbf{s},\alpha\mathbf{r}+(1-\alpha)\mathbf{s}) - f(\alpha\mathbf{r}+(1-\alpha)\mathbf{s},\mathcal{S}) \quad (2.41)$$

$$\leq \frac{1}{2}a(\alpha\mathbf{r},\alpha\mathbf{r}) + \frac{1}{2}a((1-\alpha)\mathbf{s},(1-\alpha)\mathbf{s}) - f(\alpha\mathbf{r},\mathcal{S}) - f((1-\alpha)\mathbf{s},\mathcal{S}) \quad (2.42)$$

since $a(\cdot,\cdot)$ is an inner product. Then

$$= \frac{1}{2}\alpha^2 a(\mathbf{r},\mathbf{r}) + \frac{1}{2}(1-\alpha)^2 a(\mathbf{s},\mathbf{s}) - \alpha f(\mathbf{r},\mathcal{S}) - (1-\alpha)f(\mathbf{s},\mathcal{S}) \quad (2.43)$$

$$\leq \frac{1}{2}\alpha a(\mathbf{r},\mathbf{r}) + \frac{1}{2}(1-\alpha)a(\mathbf{s},\mathbf{s}) - \alpha f(\mathbf{r},\mathcal{S}) - (1-\alpha)f(\mathbf{s},\mathcal{S}) \quad (2.44)$$

since $\alpha \in [0,1]$. Therefore

$$M^{\lambda}[\alpha\mathbf{r}+(1-\alpha)\mathbf{s},\mathcal{S}] \leq \alpha M^{\lambda}[\mathbf{r},\mathcal{S}](1-\alpha)M^{\lambda}[\mathbf{s},\mathcal{S}] \quad (2.45)$$

hence M^{λ} is convex. □

If the reconstruction algorithm is to be invariant then the dense space V_1^m must be chosen so that if $\mathbf{r} \in V_1^m$ then all rotations and translation of **r** must also be in V_1^m (all of the function spaces from Section 2.1 have this property). Note that if V_1^m has this property and the functional M^{λ} is convex and invariant to rotations and translations of the constraints then the reconstructed object will also be invariant. This is easily shown by considering a collection of constraints $\mathcal{S}$ and any object $\mathbf{r}^*(\mathbf{u})$. Let the rotated and translated constraints be denoted by $\tilde{\mathcal{S}}$ and the rotated and translated object by $\tilde{\mathbf{r}}^*(\mathbf{u})$. Invariance of the functional M^{λ} implies that

$$M^{\lambda}[\tilde{\mathbf{r}}^{*}(\mathbf{u}),\tilde{\mathcal{S}}] = M^{\lambda}[\mathbf{r}^{*}(\mathbf{u}),\mathcal{S}], \tag{2.46}$$

thus

$$M^{\lambda}[\mathbf{r}^{1}_{\lambda}(\mathbf{u}),\tilde{\mathcal{S}}] = \inf_{\tilde{\mathbf{r}}^{*}(\mathbf{u})\in V_1^m} M^{\lambda}[\tilde{\mathbf{r}}^{*}(\mathbf{u}),\tilde{\mathcal{S}}]$$

$$= \inf_{\mathbf{r}^{*}(\mathbf{u})\in V_1^m} M^{\lambda}[\mathbf{r}^{*}(\mathbf{u}),\mathcal{S}], = M^{\lambda}[\mathbf{r}^{2}_{\lambda}(\mathbf{u}),\mathcal{S}]. \tag{2.47}$$

Since the minimization of M^{λ} is well-posed, $\mathbf{r}^{1}(\mathbf{u})$ and $\mathbf{r}^{2}(\mathbf{u})$ are unique and by (2.46)

$$\mathbf{r}^{1}_{\lambda}(\mathbf{u}) = \tilde{\mathbf{r}}^{2}_{\lambda}(\mathbf{u}) \tag{2.48}$$

Thus the reconstruction algorithm will be invariant to rotations and translations. The invariance of M^{λ} can be achieved by finding invariant metrics on the constraint spaces and an invariant stablizer. The study of the appropriate forms of the invariant regularizing functional will be discussed in Chapter 3 for curves and Chapter 4 for surfaces.

To form piecewise smooth reconstructions the stablizing functional, Ω, must be defined such that it not only measures the smoothness of a particular curve or surface, but also it allows some discontinuities in the curve or surface so that the curve or surface better fits the data. The study of the modifications to the stablizing functional to have this characteristic is detailed in Chapter 5.

2.5 Solution using the Finite Difference Method

Once the form of the functional has been decided, the next step is to determine a suitable computational method for solving the minimization problem. The discrete implementation of functional minimization problems has been extensively studied [21, 33, 39, 63, 65, 78, 80-83] and the application of these techniques to this work is straight forward. Since the emphasis of this research will be on the form of the functional minimization problems and not the computational methods used to achieve the solution, not much time will be spent on the implementation aspects. It will be

noted the type of techniques that are applicable to a particular problem. It will however help the reader to understand one of these techniques, therefore in this section the discrete solution using finite differences to discretize the problem and a steepest descent type algorithm to minimize the the functional will be described.

A higher level of mathematical elegance can be achieved by using the finite element method [21, 63, 78, 82] to discretize the continuous problem. When using this method the convergence of the discrete problem to the continuous problem can be discussed. The finite element method also allows the problem to be discretized on nonrectangular grids. The greater power of the finite element method is not needed in this work, thus the finite difference method was chosen to discretize the problem. Note that a nonconformal element on a rectangular grid can be defined so that the discrete problems formed by the finite element method and the finite differences method are identical [82].

The steepest descent minimization technique is the simplest to understand, however it is not the most computationally efficient. When applicable the conjugate gradient method (described in Appendix A) [39] will achieve greater efficiency. In some cases, it may also be desirable to use multiresolution techniques [35, 82] to achieve even greater efficiency. However, the gain in efficiency will not be as great as others have reported [82] since, as in [48, 49], the algorithms developed in this work first form a crude estimate of the solution before minimizing the functional. The largest gains with multiresolution techniques are achieved when the initial estimate is far from the solution. It this work the algorithms in Chapter 3 and 4 were implemented using the conjugate gradient method while the algorithms in Chapter 5 were implemented using the steepest descent method.

To form the discrete problem statement the representation is sampled on a uniform grid in the parameter space. For simplicity, a sampling interval of 1 is assumed. For example, a curve will be represented by the collection of sample

$$\{\mathbf{r}(i) : i \text{ integer}, i \in U\}. \tag{2.49}$$

Similarly for surface, which has 2 degrees-of-freedom, the collection of samples is

$$\{\mathbf{r}(i,j) : i,j \text{ integer}, (i,j) \in U\}. \tag{2.50}$$

The collection of samples can be represented by concatenating the vector elements and samples into a single vector, $\mathbf{r}$, e.g. for curves in two-dimensional space the $2\times N$ vector is formed by

$$\mathbf{r} = <\cdots, x(i-1), y(i-1), x(i), y(i), x(i+1), y(i+1), \cdots>, \tag{2.51}$$

where N is the number of samples in the domain. To discretize the problem it is necessary to form discrete approximations to first- and second-order derivatives of the continuous representation using the collection of samples. For curves the following discrete approximations to the differential operators are used [1]

$$\mathbf{r}'(i) \approx \mathbf{r}(i+1) - \mathbf{r}(i), \tag{2.52}$$

$$\mathbf{r}''(i) \approx \mathbf{r}(i+1) - 2\mathbf{r}(i) + \mathbf{r}(i-1). \tag{2.53}$$

Similarly for surfaces the following approximations are used [1]

$$\mathbf{r}_u(i,j) \approx \mathbf{r}(i+1,j) - \mathbf{r}(i,j), \tag{2.54}$$

$$\mathbf{r}_v(i,j) \approx \mathbf{r}(i,j+1) - \mathbf{r}(i,j), \tag{2.55}$$

$$\mathbf{r}_{uu}(i,j) \approx \mathbf{r}(i+1,j) - 2\mathbf{r}(i,j) + \mathbf{r}(i-1,j), \tag{2.56}$$

$$\mathbf{r}_{uv}(i,j) \approx \mathbf{r}(i+1,j+1) - \mathbf{r}(i+1,j), - \mathbf{r}(i,j+1) + \mathbf{r}(i,j), \tag{2.57}$$

$$\mathbf{r}_{vv}(i,j) \approx \mathbf{r}(i,j+1) - 2\mathbf{r}(i,j) + \mathbf{r}(i,j-1). \tag{2.58}$$

Using these approximations the continuous functional minimization problem can be approximated by the discrete functional minimization problem

$$M_d^{\lambda}[\mathbf{r}(\mathbf{u}),\mathcal{S}] = A(\mathbf{r},\mathcal{S}), \tag{2.59}$$

where M_d^{λ} denotes the discrete form of the continuous functional M^{λ}.

The steepest descent method is an iterative method for finding the minimum function. Let $\mathbf{r}^{(k)}$ denote the estimate after the k^{th} iteration. An initial estimate of the surface, $\mathbf{r}^{(0)}$, is iteratively updated by forming an update vector, $\mathbf{g}$, in the direction of steepest descent. The steepest descent direction is computed by evaluating the gradient of the functional at the current function estimate

$$\mathbf{g}^{(k)} = \nabla A(\mathbf{r}^{(k)},\mathcal{S}). \tag{2.60}$$

The update to the function is

$$\mathbf{r}^{(k+1)} \quad \mathbf{r}^{(k)} + \beta^{(k)}\mathbf{g}^{(k)}, \tag{2.61}$$

where $\beta^{(k)}$ is some constant. The optimal $\beta^{(k)}$ is found by minimizing $A(\cdot,\cdot)$ with respect to $\beta^{(k)}$, i.e. find $\beta^{(k)}$ such that

$$\frac{\delta}{\delta\beta^{(k)}} A(\mathbf{r}^{(k)}+\beta^{(k)}\mathbf{g}^{(k)},\mathcal{S}) = 0. \tag{2.62}$$

For the functional minimization problems in Chapters 3 and 4 the functional $A(\cdot,\cdot)$ has a quadratic form and can be written using vector matrix notation as

$$A(\mathbf{r},\mathcal{S}) = \mathbf{r}^T\mathbf{A}\mathbf{r} - \mathbf{c}^T\mathbf{r}, \tag{2.63}$$

where $\mathbf{A}$ is some matrix, $\mathbf{c}$ is some vector and $\mathbf{r}^T$ denotes the vector transpose. In these cases, the gradient vector is given by

$$\mathbf{g}^{(k)} = \mathbf{A}\mathbf{r}^{(k)} - \mathbf{c} \tag{2.64}$$

and the optimal β is given by

$$\beta^{(k)} = \frac{\mathbf{g}^{(k)T}\mathbf{g}^{(k)}}{\mathbf{g}^{(k)T}\mathbf{A}\mathbf{g}^{(k)}}. \tag{2.65}$$

The actual form of the matrix $\mathbf{A}$ and the functional $A(\cdot,\cdot)$ are specific to the particular functional and the details will be discussed for each of the proposed algorithms. In general for all of the algorithms discussed in this thesis, the form of $\mathbf{A}$ will be sparse and banded, that is, most of the elements of $\mathbf{A}$ are zero except for relatively few nonzero diagonals. This is very important from a computational perspective because it will mean that all computations are local in nature leading to very fast implementations on parallel mesh architectures.

CHAPTER 3 - INVARIANT CURVE RECONSTRUCTION

In this chapter the reconstruction of smooth curves in two and higher dimensions is examined. To construct an invariant estimate of a curve it is necessary to find a stablizer, Ω, which not only produces plausible reconstructions but which is also invariant to rotations and translations of the constraint set. Thus the study of the problem will begin with an examination of several invariant characteristics of curves and a discussion of the nature of the reconstructions which are constructed using these characteristics. From this analysis a mathematical problem statement will be defined for the reconstruction of curves in m-dimensions. The functional that is defined, however, will be shown to be nonconvex. Thus the actual computation of the solution will be difficult.

In Section 3.5 two new approximations to the nonconvex function are proposed. The case of a curve in two-dimensional space is first examined and a new convex approximation to the nonconvex problem is analyzed. In Section 3.7 popular techniques for the reconstruction of curves are shown to be very noninvariant to rotation of the constraints. That is, the reconstructed curve varies dramatically if the coordinate system is rotated. The reasons behind the new approximation are shown to be valid and it will be demonstrated that the curves reconstructed with this algorithm are much more resistant to deformation with rotations of the constraints than previously proposed methods. In Section 3.5.3 the new reconstruction algorithm is extend to curves in higher dimensional spaces. Section 3.9 describes two applications of the new algorithms to some problems in computer vision.

3.1 Invariant Characteristics of Curves

The results in this section can be found in most textbooks on differential geometry [34]. Also see [6, 9] for a discussion of invariant characteristics of curves and surfaces

in two- and three-dimensions. Let $\mathbf{r}(u)$, $u \in U \subset \mathbb{R}$, be a parametric representation of the curve in a m-dimensional space. Assume the components of $\mathbf{r}(u)$ have continuous n^{th} order derivatives respectively. Let $\mathbf{r}'(u)$, $\mathbf{r}''(u)$ and $\mathbf{r}^{(j)}(u)$ denote the first, second and j^{th} derivative of $\mathbf{r}(u)$. The following quantities can be defined. The velocity of a curve is defined as

$$v(u) = ||\mathbf{r}'(u)|| \tag{3.1}$$

The arc length of the curve is given by

$$\int_U v(u)\, du. \tag{3.2}$$

The unit tangent vector function $\mathbf{t}(u)$ is defined as

$$\mathbf{t}(u) = \frac{\mathbf{r}'(u)}{||\mathbf{r}'(u)||}, \tag{3.3}$$

and the normal unit vector function $\mathbf{n}(u)$ is defined as

$$\mathbf{n}(u) = \frac{\mathbf{t}'(u)}{||\mathbf{t}'(u)||}, \tag{3.4}$$

For planar curves the vector functions $\mathbf{t}(u)$ and $\mathbf{n}(u)$ define a right-handed coordinate system on the curve as shown in Figure 3.1. The curvature of the curve is given by

$$\kappa(u) = \frac{||\mathbf{t}'(u)||}{v(u)} \tag{3.5}$$

The problem of determining invariant quantities is addressed by computing the Frent-Serret equations using vector calculus. For the two-dimensional case these equations can be written as

$$\begin{bmatrix} \mathbf{t}'(u) \\ \mathbf{n}'(u) \end{bmatrix} = \begin{bmatrix} 0 & \kappa(u)v(u) \\ -\kappa(u)v(u) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{t}(u) \\ \mathbf{n}(u) \end{bmatrix}. \tag{3.6}$$

From this it is known that $\kappa(u)$ and $v(u)$ are invariant. $v(u)$ is known as a first-order invariant since it depends only on the first derivative of $\mathbf{r}(u)$, while $\kappa(u)$ is a second-

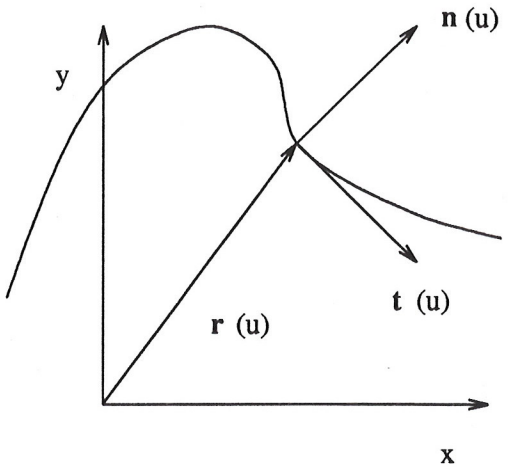

Figure 3.1. Tangent-Normal coordinate system at a point on the curve

order invariant since it also depends on $\mathbf{r}''(u)$. Thus in the two dimensional case the stablizing functional should be based on the quantities $\nu(u)$ and/or $\kappa(u)$.

In higher dimensions the Frent-Serret equations yield the matrix

$$\begin{bmatrix} 0 & \kappa_1(u)\nu(u) & 0 & & 0 & 0 & 0 \\ -\kappa_1(u)\nu(u) & 0 & \kappa_2(u)\nu(u) & \cdots & 0 & 0 & 0 \\ 0 & -\kappa_2(u)\nu(u) & 0 & & 0 & 0 & 0 \\ & \vdots & & & & \vdots & \\ 0 & 0 & 0 & \cdots & -\kappa_{m-2}(u)\nu(u) & 0 & \kappa_{m-1}(u)\nu(u) \\ 0 & 0 & 0 & & 0 & -\kappa_{m-1}(u)\nu(u) & 0 \end{bmatrix} \tag{3.7}$$

where $\kappa_j(u)$ is the j^{th} curvature of $\mathbf{r}(u)$. The curvatures are invariants and characterize $\mathbf{r}(u)$ up to an Euclidean motion. $\kappa_j(u)$ is an j+1-order invariant meaning that $\kappa_j(u)$ depends on $\mathbf{r}^{(j+1)}(u)$ but $\kappa_i(u)$ $i = 1,2,...j-1$ do not. $\kappa_1(u)$ is the curvature as defined by (3.5). So in higher dimensional spaces higher order curvatures can be included in the stablizing definition.

3.2 Invariant Metrics on C_i

For each type of constraint forming process an invariant metric must be determine. In this section invariant metrics for location and normal constraints will be defined.

Recall for location constraints

$$\mathbf{l}_i[\mathbf{r}(u)] = \mathbf{r}(u_i). \tag{3.8}$$

Since this constraint is on the location of the curve in space a natural measure on the constraint space in the Euclidean distance of the object at u_i to the constraint location. Thus for this type of constraint a possible metric can be defined as

$$\rho_{C_i}(\mathbf{r}(u_i),\mathbf{c}_i) = \left[\sum_{j=1}^{m}(x_j(u_i) - c_{j,i})^2\right]^{1/2} \tag{3.9}$$

where $c_{j,i}$ denotes the j^{th} component of the vector $\mathbf{c}_i$. This metric is invariant to rotations and translations since it is the measure of the distance between two points which is an invariant quantity. If an explicit representation is used for the curve this metric becomes

$$\rho_{C_i}(<u_i,f(u_i)>,\mathbf{c}_i) = \left[(f(u_i)-c_{2,i})^2 + (u_i-c_{1,i})^2\right]^{1/2}. \tag{3.10}$$

This metric is also invariant, however, its practical application often requires that

$$u_i = c_{1,i} \tag{3.11}$$

and the metric reduces to

$$\rho_{C_i}(<u_i,f(u_i)>,\mathbf{c}_i) = |f(u_i)-c_{2,i}|. \tag{3.12}$$

This metric, however, is not invariant to rotations.

For a curve in two-dimensional space the normal is given by (2.25). The angle between the normal and the constraint is invariant to rotations and shifts; thus it can be used as a basis for the metric defined on C_i. Let θ_i be the angle between $\mathbf{l}_i[\mathbf{r}(u)]$ and $\mathbf{c}_i$, recall from vector algebra that

$$\cos(\theta_i) = \frac{(\mathbf{l}_i[\mathbf{r}(u)],\mathbf{c}_i)}{\|\mathbf{l}_i[\mathbf{r}(u)]\|_{C_i}\,\|\mathbf{c}_i\|_{C_i}} \tag{3.13}$$

since $\mathbf{l}_i[\mathbf{r}(u)]$ and $\mathbf{c}_i$ represent unit normals, $\|\mathbf{l}_i[\mathbf{r}(u)]\| = \|\mathbf{c}_i\| = 1$ so (3.13) reduces to

$$\cos(\theta_i) = \frac{(<y_u(u_i), -x_u(u_i)>, <c_{x,i}, c_{y,i}>)}{(x_u^2(u_i) + y_u^2(u_i))^{1/2}}$$

$$= \frac{(y_u(u_i)c_{x,i} - x_u(u_i)c_{y,i})}{(x_u^2(u_i) + y_u^2(u_i))^{1/2}}. \tag{3.14}$$

Thus

$$\sin^2(\theta_i) = 1 - \cos^2(\theta_i) = 1 - \frac{(y_u(u_i)c_{x,i} - x_u(u_i)c_{y,i})^2}{(x_u^2(u_i) + y_u^2(u_i))}$$

$$= \frac{(1 - c_{x,i}^2)y_u^2(u_i) + 2c_{x,i}c_{y,i}x_u(u_i)y_u(u_i) + (1 - c_{y,i}^2)x_u^2(u_i)}{(x_u^2(u_i) + y_u^2(u_i))}. \tag{3.15}$$

Since $\mathbf{c}_i$ is a unit vector $(1 - c_{x,i}^2) = c_{y,i}^2$ and $(1 - c_{y,i}^2) = c_{x,i}^2$ so

$$\sin^2(\theta_i) = \frac{c_{y,i}^2 y_u^2(u_i) + 2c_{x,i}c_{y,i}x_u(u_i)y_u(u_i) + c_{x,i}^2 x_u^2(u_i)}{(x_u^2(u_i) + y_u^2(u_i))}$$

$$= \frac{(y_u(u_i)c_{y,i} + x_u(u_i)c_{x,i})^2}{(x_u^2(u_i) + y_u^2(u_i))}. \tag{3.16}$$

The explicit form of this equation can be found by setting $x_u = 1$ and defining the slope of the constraint vector as m, $(m_i = -c_{x,i}/c_{y,i})$, yielding

$$\sin^2(\theta_i) = \frac{(y_u(u_i) - m_i)^2}{(1 + y_u^2)\, c_{y,i}} \tag{3.17}$$

An invariant metric can be defined on C_i by

$$\rho_{C_i}(\mathbf{l}_i[\mathbf{r}(u)], \mathbf{c}_i) = |\theta_i|. \tag{3.18}$$

Invariant metrics defined on normals in higher dimensional spaces can be found using this method, however, the formulas become very complex. In general, invariant metrics on other types of constraints can be found if there is some invariant characteristic that can be computed to measure the closeness of the reconstruction to the

constraint.

3.3 Invariant Stablizers

A prior knowledge of the type of reconstruction that is desired is used to determine an appropriate stablizer based on invariant characteristics. It is desirable to reconstruct curves which are smooth. Recall that the stablizer, Ω, is used in the definition of M^{λ} which is then minimized.

3.3.1 A First-Order Stablizer

As a first attempt the stablizer can be based on the first-order invariant $\nu(u)$. One idea is to use a measure of the arc length as a stablizer. For this the stablizer can be defined as

$$\Omega^1[\mathbf{r}(u)] = \int_U \nu(u)\, du$$

$$= \int_U \|\mathbf{r}'(u)\|\, du \tag{3.19}$$

By using the calculus of variations and the Euclidean norm the Euler-Lagrange differential equation is given by

$$\frac{\partial \nu(u)}{\partial x_i(u)} - \frac{d}{du}\left[\frac{\partial \nu(u)}{\partial x_i'(u)}\right] = 0, \qquad i = 1,2,\ldots,m \tag{3.20}$$

$$\Rightarrow \quad \frac{d}{du}\left[\frac{x_i'(u)}{\left[\sum_{j=1}^{m} x_j'(u)^2\right]^{1/2}}\right] = 0 \tag{3.21}$$

$$\Rightarrow \quad x_i''(u)\left[\sum_{j=1}^{m} x_j'(u)^2\right] - x_i'(u)\left[\sum_{m=1}^{m} x_j'(u)x_j''(u)\right] = 0. \tag{3.22}$$

Let $(\cdot,\cdot)$ denote the normal Euclidean inner product, Then (3.22) can be written as the vector equation.

$$(\mathbf{r}'(u),\mathbf{r}'(u))\mathbf{r}''(u) - (\mathbf{r}'(u),\mathbf{r}''(u))\mathbf{r}'(u) = \mathbf{0}. \tag{3.23}$$

Dot this equation with $\mathbf{r}''(u)$ to attain

$$(\mathbf{r}'(u),\mathbf{r}'(u))(\mathbf{r}''(u),\mathbf{r}''(u)) - (\mathbf{r}'(u),\mathbf{r}''(u))^2 = \|\mathbf{r}'(u)\times\mathbf{r}''(u)\| = 0. \tag{3.24}$$

Therefore, either $\mathbf{r}''(u) = \mathbf{0}$ or $\mathbf{r}''(u) = c\mathbf{r}'$, c is a constant. Assuming the nontrivial second case, this implies that

$$\mathbf{r}(u) = \mathbf{a}e^{cu} + \mathbf{b} \tag{3.25}$$

where **a** and **b** are constant vectors. By making the change of parameterization $t = e^{cu}$ the form of **r** is found to be

$$\mathbf{r}(t) = \mathbf{a}t + \mathbf{b}, \tag{3.26}$$

which is the equation of a straight line. This equation must be satisfied between constraint points in order that M^{λ} be minimized. This implies that between constraints the reconstructed curve will be a line. This is an expected result since the shortest distance between two points in a straight line. While this will yield a continuous solution to the problem, the curve reconstructed with this stablizer will not be smooth. To obtain a smooth reconstruction higher-order invariants must be used in the stablizer definition.

3.3.2 A Second-Order Stablizer

For smoothly varying curves a more appropriate technique is to model the curve as an ideal thin flexible beam of elastic material. The stablizer would then be a measure of

the strain energy. The reconstructed curve would then have the physical representation of the minimum energy configuration of a beam with deformations. Daniel Bernoulli first suggested that the strain energy is proportional to the integral of the square of the curvature taken along the curve [5]. Thus define the stablizer as

$$\Omega[\mathbf{r}(u)] = \int_U \kappa^2(u)\, \nu(u)\, du. \tag{3.27}$$

In this case the Euler-Lagrange differential equation is nonlinear [58]. The reconstructed curve, $\mathbf{r}(u)$, obtained by the minimization of M^λ with this stablizer is referred to as the spline approximate. This stablizer is based on a second-order invariant whereas the arc length stablizer, Ω^1, is based on a first-order invariant. Thus it is expected that a curve reconstructed with, Ω, would be smoother than the straight-line approximate obtained with Ω^1. As will be shown in the examples for smooth invariant curves reconstruction (3.27) is an adequate stablizer and will be the stablizer used in this chapter. In higher-dimensional spaces more smoothness can be obtained by adding higher-order curvature functions to the stablizer.

3.4 An Invariant Problem Statement

The problem of reconstructing a curve in m-dimensional space given a set of constraints $\mathcal{S}$ is now posed as a functional minimization problem. The estimate of the curve is formed by minimizing the functional

$$M^\lambda[\mathbf{r}^*(u),\mathcal{S}] = \int_U \kappa^2(u)\, \nu(u)\, du + \lambda \sum_{\mathbf{c}_i \in \mathcal{S}} ||\mathbf{r}^*(u_i)-\mathbf{c}_i||^2. \tag{3.28}$$

The constant λ determines a tradeoff between the smoothness of the curve and closeness of fit of the surface to the data. To show that this problem is well-posed it must be shown that the minimization of this functional exists and is unique. In fact, this functional minimization problem is still not well-posed. Consider the simple example of the two location constraints

$$\mathbf{l}_1[\mathbf{r}(u)] = \mathbf{r}(0) = <0,0> \qquad \mathbf{l}_2[\mathbf{r}(u)] = \mathbf{r}(1) = <1,0> \tag{3.29}$$

then the curve

$$\mathbf{r}(u) = <1,0>\, u \tag{3.30}$$

minimizes the functional (3.28) (since $M^{\lambda} = 0$). However so does

$$\mathbf{r}(u) = <1,0> + <-1,0>\, u. \tag{3.31}$$

This curve has the same shape but a different parametrization. Similar examples can be constructed when there are more constraints. The functional minimization problem is not well-posed since the solution is not unique. However, since all curves which are global minimums of the functional (3.28) will have the same shape (they will have different parametrizations) any $\mathbf{r}(u)$ which minimizes (3.28) will be a sufficient estimate for the curve.

However, the stablizer Ω can also be shown to be nonconvex. To show that Ω is nonconvex, it must be shown that there exist some $\mathbf{f}(u)$, $\mathbf{g}(u)$ and $\alpha \in [0,1]$ such that

$$\Omega[\alpha\mathbf{f}(u) + (1-\alpha)\mathbf{g}(u)] > \alpha\Omega[\mathbf{f}(u)] + (1-\alpha)\Omega[\mathbf{g}(u)] \tag{3.32}$$

To show this, it will first be shown that

$$\Omega[\beta\mathbf{r}(u)] = \frac{1}{|\beta|}\Omega[\mathbf{r}(u)]. \tag{3.33}$$

Let $\mathbf{r}_2(u) = |\beta|\, \mathbf{r}_1(u)$, then

$$\nu_2(u) = |\beta|\, \nu_1(u) \qquad \mathbf{t}_2(u) = \mathbf{t}_1(u). \tag{3.34}$$

So

$$\kappa_2(u) = \frac{\|\mathbf{t}_2'(u)\|}{\nu_2(u)} = \frac{1}{|\beta|}\,\frac{\|\mathbf{t}_1'(u)\|}{\nu_1(u)} = \frac{1}{|\beta|}\,\kappa_1(u) \tag{3.35}$$

thus

$$\Omega[\beta\,\mathbf{r}_1] = \frac{1}{|\beta|}\,\Omega[\mathbf{r}_1]. \tag{3.36}$$

Now to show nonconvexity, let $\alpha = 0.5$ and let $\mathbf{g}(u) = (2\beta-1)\mathbf{f}(u)$. Then

$$\Omega[\alpha\mathbf{f}(u) + (1-\alpha)\mathbf{g}(u)] = \Omega[\beta\mathbf{f}] = \frac{1}{|\beta|}\Omega[\mathbf{f}] \tag{3.37}$$

and

$$\alpha\Omega[\mathbf{f}(u)] + (1-\alpha)\Omega[\mathbf{g}(u)] = \frac{1}{2}\left[1 + \frac{1}{|2\beta-1|}\right]\Omega[\mathbf{f}]. \tag{3.38}$$

Therefore the inequality that needs to proved for some value of β reduces to

$$\frac{1}{|\beta|} > \frac{1}{2}\left[1 + \frac{1}{|2\beta-1|}\right] \tag{3.39}$$

Letting $\beta = 0.1$ gives

$$10 > 1.125 \tag{3.40}$$

which is of course true. Therefore Ω is nonconvex. Note, however, that the the metric for the location constraints is convex.

3.5 Convex Approximation to the Invariant Problem

In this section, a two-stage algorithm is described which constructs a curve estimate which is more robust to variations of the coordinate system. In the first stage, a piecewise linear invariant approximation of the curve is found by minimizing an invariant functional. This piecewise linear curve is used to form an approximate constant velocity parameterization of the curve. With this parameterization of the curve it is possible to obtain a valid approximation to the invariant stablizer, (3.27), which is

well-posed and convex. The special case of planar curves will be examined first, followed by the more general case of curves in m-dimensional space.

3.5.1 Stage 1: Continuous Piecewise Linear Approximation of the Curve

A continuous linear curve can be completely specified by the locations of the endpoints of each linear segment. If we have M location constraints we will form a piecewise linear curve with $M-1$ segments. This curve can be specified with the collection of M locations, $\{\ell_i\}$. To form the reconstruction the following quantity is minimized with respect to the collection $\{\ell_i\}$

$$M^{\lambda_\ell}[\ell_i,\mathcal{S}] = \sum_{i=1}^{M-1} \|\ell_i - \ell_{i+1}\|^2 + \lambda_l \sum_{\mathbf{c}_i \in \mathcal{S}} \|\ell_i - \mathbf{c}_i\|^2. \tag{3.41}$$

The function (3.41) is convex since each term is quadratic. The function is also invariant since each term is based on an invariant quadratic (the distance between two points). Let the collection of ℓ_i that minimize (3.41) be denoted by ℓ_i^*.

This function is based on the physical model of finding the minimum energy configuration of a set of springs. The first term of (3.41) represents the energy of a collection of $M-1$ springs, one along each segment of the continuous piecewise linear curve. The second term represents the energy of a collection of M springs, each one connecting a constraint location with its corresponding location of the curve. The term λ_ℓ controls the ratio of the spring constants for the two sets of springs.

A constant velocity parameterization along the continuous piecewise linear curve is given by

$$\mathbf{r}_\ell(u) = \ell_i^* + (u-u_i)\frac{(\ell_{i+1}^* - \ell_i^*)}{\|\ell_{i+1}^* - \ell_i^*\|}, \qquad u \in [u_i, u_{i+1}), \tag{3.42}$$

where

$$u_0 = 0,$$

$$u_i = u_{i-1} + ||\ell_i^* - \ell_{i-1}^*||. \tag{3.43}$$

Note that for this curve the velocity is given by

$$v_\ell(u) = ||\mathbf{r}'_\ell(u)|| = 1. \tag{3.44}$$

The curve $\mathbf{r}_\ell(u)$ approximates the smooth reconstructed curve $\hat{\mathbf{r}}(u)$ so if the same parameterization is used for the smooth curve, the approximation that

$$v(u) = ||\hat{\mathbf{r}}'(u)|| \approx 1. \tag{3.45}$$

will be valid. By the same parameterization we mean that we use the u_i determined for the piecewise linear curve to determine where on the curve the constraint should be located.

3.5.2 Stage 2: Special Case - Planar Curves

Assuming that the curve is planar (use the notation $\mathbf{r}(u) = <x(u), y(u)>$) and also that the approximation (3.45) is valid. Then the invariant stablizer, (3.27), can be approximated by

$$\Omega_1[\mathbf{r}(u)] = \int_U (x''(u)y'(u) - y''(u)x'(u))^2 \, du \tag{3.46}$$

This functional has several interesting properties:

Property: For a given $x(u)$ the functional

$$p[y(u)] = \Omega_1[<x(u), y(u)>]^{0.5} \tag{3.47}$$

is a convex mapping of $y(u)$ (similarly, for a given $y(u)$, (3.46) is a convex mapping of

$x(u)$).

Proof: The proof of this property requires proving the convexity condition (2.19). This is accomplished by first showing that

$$p[y_1(u)+y_2(u)] \leq p[y_1(u)]+p[y_2(u)], \qquad y_1(u),y_2(u) \in Y(U) \tag{3.48}$$

and then proving that

$$p[\beta y(u)] = \beta p[y(u)], \qquad y(u) \in Y(U), \ \beta \in [0,1]. \tag{3.49}$$

The u parameter will be dropped to ease the notation. Let $y_1, y_2 \in Y(U)$ then

$$p(y_1+y_2)^2 = \int_U (x''(y'_1+y'_2)-(y''_1+y''_2)x')^2 \, du \tag{3.50}$$

$$= \int_U ((x''y'_1 - y''_1x')+(y''_2x'-y'_2x'))^2 \, du \tag{3.51}$$

$$= \int_U \{|(x''y'_1 - y''_1x')+(y''_2x'-y'_2x')| \times |(x''y'_1 - y''_1x')+(y''_2x'-y'_2x')|\} \, du \tag{3.52}$$

$$\leq \int_U |(x''y'_1 - y''_1x')+(y''_2x'-y'_2x')||x''y'_1 - y''_1x'| \, du$$

$$+\int_U |(x''y'_1 - y''_1x')+(y''_2x'-y'_2x')||x''y'_2 - y''_1x'| du \tag{3.53}$$

by the Riez-Hölder inequality

$$p(y_1+y_2)^2 \le \left[\int_U ((x''y'_1 - y''_1x') + (y''_2x' - y'_2x'))^2\, du\right]^{1/2}$$

$$\times\left[\left[\int_U (x''y'_1 - y''_1x')^2\, du\right]^{1/2} + \left[\int_U (x''y'_2 - y''_2x')^2\, du\right]^{1/2}\right] \quad (3.54)$$

$$\Rightarrow \qquad p(y_1+y_2) \le p(y_1)+p(y_2) \quad (3.55)$$

For the second property

$$p(\alpha y) = \left[\int_U (x''\,\alpha y' - \alpha y''x')^2\, du\right]^{1/2} \quad (3.56)$$

$$= \alpha\left[\int_U (x''y' - y''x')^2\, du\right]^{1/2} = \alpha p(y). \quad (3.57)$$

These two properties show that

$$p(\alpha y_1 + (1-\alpha)y_2) \le \alpha p(y_1) + (1-\alpha)p(y_2) \quad (3.58)$$

which proves convexity. □

Note, however, that M^{λ} is not convex with respect to the mapping $\mathbf{r}(u)$. To form the reconstructed curve the functional is minimized twice. First, $x'(u)$ and $x''(u)$ are approximated and the functional is minimized with respect to the $y(u)$ component. Then, the functional is minimized with respect to $x(u)$ while $y(u)$ is held constant. This minimization process does not necessarily find a global minimum to the functional, however, the curves that are reconstructed will be smooth and approximately invariant. To form approximations to $x'(u)$, $x''(u)$, $y'(u)$, and $y''(u)$ the continuous piecewise

linear approximation to the curve is used. Let the components of $\mathbf{r}_\ell(u)$ and ℓ_i^* be given by $<x_\ell(u),y_\ell(u)>$ and $<x_i^*,y_i^*>$, respectively. For example, when minimizing with respect to $y(u)$, $x'(u)$ and $x''(u)$ are approximated by

$$x_l'(u) = \frac{x_{i+1}^* - x_i^*}{\|\ell_{i+1}^* - \ell_i^*\|}, \qquad u \in [u_i,u_{i+1}), \quad i = 1,...,M-1, \tag{3.59}$$

and

$$x_l''(u) = \left\{ \frac{x_{i+1}^* - x_i^*}{\|\ell_{i+1}^* - \ell_i^*\|} - \frac{x_i^* - x_{i-1}^*}{\|\ell_i^* - \ell_{i-1}^*\|} \right\} \delta(u - u_i), \qquad i = 2,...,M-1, \tag{3.60}$$

where $\delta(u)$ is the Dirac delta function. Define the constants A_i and B_i such that

$$x_l'(u) = A_i, \qquad x_l''(u) = B_i\delta(u-u_i), \qquad u \in [u_i,u_{i+1}). \tag{3.61}$$

The then stablizer, Ω_1, reduces to

$$\Omega_1[<x_\ell(u),y(u)>] = \sum_{i=1}^{M-1} B_i^2 \int_{u_i}^{u_{i+1}} y''(u)^2 du$$

$$+ \sum_{i=2}^{M-1} \left[A_i^2 y'(u_i)^2 - 2A_iB_iy'(u_i)y''(u_i) \right]. \tag{3.62}$$

The functional is minimized with respect to $y(u)$ using Ω_1 as the approximate stablizer to form the smooth approximation of the $y(u)$ component of $\hat{\mathbf{r}}(u)$, i.e. $\hat{y}(u)$. In a similar fashion, $\Omega_3[<x(u),y_\ell(u)>]$ is defined and minimized to form the smooth approximation $\hat{x}(u)$.

To summarize, the proposed reconstruction algorithm for curves in two-dimensional space is outlined below.

Algorithm: planar case

1. Compute the invariant continuous piecewise linear approximation $\mathbf{r}_\ell(u) = \langle x_\ell(u), y_\ell(u) \rangle$ by minimizing (3.41) with respect to the collection $\{t_i\}_{i=1}^{M}$.

2. Compute the smooth reconstruction, $\hat{\mathbf{r}}(u)$, by minimizing the regularizing functional with the stablizers $\Omega_1[\langle x_\ell(u), y(u) \rangle]$ and $\Omega_1[\langle x(u), y_\ell(u) \rangle]$.

Examples of curves reconstruction with this algorithm are given in Section 3.7.

3.5.3 Stage 2: General Case - Curves in M-Dimensional Space

Unfortunately the algorithm presented in the previous subsection cannot be easily extended to higher-dimensional spaces. However, if the constant velocity approximation is introduced at an earlier stage in the derivation of the stablizer the computational complexity of the algorithm can be reduced with satisfactory results and the algorithm can be extended to curves in higher-dimensional spaces. The approximation is the same, however, it is introduced earlier in the functional definition to produce a slightly different functional to minimize. Recall that the approximation used is

$$\mathrm{v}(u) = \|\mathbf{r}'(u)\| \approx 1. \tag{3.63}$$

Thus

$$\mathbf{t}(u) = \frac{\mathbf{r}'(u)}{\|\mathbf{r}'(u)\|} \approx \mathbf{r}'(u), \tag{3.64}$$

and

$$\kappa(u) = \frac{\|\mathbf{t}'(u)\|}{\mathrm{v}(u)} \approx \|\mathbf{r}''(u)\|. \tag{3.65}$$

So the approximate stablizer for (3.27) becomes

$$\Omega_2[\mathbf{r}(u)] = \int_U ||\mathbf{r}''(u)||^2 \, du. \tag{3.66}$$

This stablizer is convex since the integrand is just a squared norm. The complete algorithm is outlined below.

Algorithm: general case

1. Compute the invariant continuous piecewise linear approximation $\mathbf{r}_\ell(u) = \langle x_\ell(u), y_\ell(u) \rangle$ by minimizing (3.41) with respect to the collection $\{\ell_i\}_{i=1}^{M}$.

2. Compute the smooth reconstruction, $\hat{\mathbf{r}}(u)$ by minimizing the regularizing functional with the stablizer $\Omega_2[\mathbf{r}(u)]$.

3.6 Implementation

To discretized the continuous functional M^λ the approach outline in Section 2.5 was used. For both of the algorithms described in the previous section the following vector-matrix equation is obtained

$$M_d^\lambda[\mathbf{r}, \mathbf{c}] = \frac{1}{2}\mathbf{r}^T(\mathbf{Q}+\lambda\mathbf{I}_S)\mathbf{r} - \mathbf{c}^T\mathbf{r}, \tag{3.67}$$

where the matrix $\mathbf{I}_S$ is a diagonal matrix with a 1 in row i if $\mathbf{c}_i \in S$. The i^{th} component of $\mathbf{c}$ is $\mathbf{c}_i$ if it exists, else it is zero. The form of Q is dependent on the particular algorithm implemented.

For the first algorithm described in Section 3.5.2 only one component of $\mathbf{r}$ is minimized at a time. Therefore the problem can be split into two. First, the minimization is performed with respect to the y component. Let

$$\alpha_{(i)} = x'(i) + x''(i), \tag{3.68}$$

$$\beta_{(i)} = x'(i) - x''(i). \tag{3.69}$$

Then the **Q** matrix can be given by

$$\begin{bmatrix} 2\alpha_{(1)}^2 & -4x'(1)\alpha_{(2)} & 2\alpha_{(1)} & 0 & 0 \\ -4x'(2)\alpha_{(2)} & 2\alpha_{(2)}^2+8x'(2)^2 & -8x'(2)^2 & 2\alpha_{(2)}\beta_{(2)} & 0 \\ 2\alpha_{(3)}\beta_{(3)} & -8x'(3)^2 & 12x'(3)^2+4x''(3)^2 & -8x'(3)^2 & 2\alpha_{(3)}\beta_{(3)} \cdots \\ 0 & 2\alpha_{(4)}\beta_{(4)} & -8x'(4)^2 & 12x'(4)^2+4x''(4)^2 & -8x'(4) \\ & & \vdots & & \end{bmatrix}$$

Once the minimization is done with respect to y a new **Q** matrix is constructed with the x's replaced with y's and the functional is minimized with respect to x.

For the algorithms described in Section 3.5.3 the minimization can once again be split into two separate minimizations done over each of the components. This time the **Q** is given by

$$\begin{bmatrix} 2 & -4 & 2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ -4 & 10 & -8 & 2 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 2 & -8 & 12 & -8 & 2 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -8 & 12 & -8 & \dots & 0 & 0 & 0 & 0 & 0 \\ & & \vdots & & & & & & \vdots & & \\ 0 & 0 & 0 & 0 & 0 & \dots & 2 & -8 & 12 & -8 & 2 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 2 & -8 & 10 & -4 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 2 & -4 & 2 \end{bmatrix} \tag{3.70}$$

3.7 Comparison

This section describes two commonly used approximations to the nonconvex problem which are well-posed. Both of these algorithms use the explicit representation of a curve and are therefore limited to planar curves. The explicit representation of a curve can be considered a special parameterization where one coordinate is a function of the other. We will use the following notation for the explicit form

$$\mathbf{r}(x) = \langle x, y(x) \rangle. \tag{3.71}$$

When the explicit form of the curve is used, the metric for the location constraints becomes

$$\rho(\mathbf{r}(x_i), \mathbf{c}_i) = (x_i - c_{1,i})^2 + (y(x_i) - c_{2,i})^2. \tag{3.72}$$

In order for this equation to be quadratic, we must set x_i equal to a constant (usually $c_{1,i}$). If this is done, however, the metric is no longer invariant.

3.7.1 Cubic Splines

The most common approximation made is to assume that the curve is approximately flat (i.e. $(1+y'(x)^2)^{1/2} = b$ a constant) [18]. The explicit form of the stablizer in (3.27) then becomes

$$\Omega_3[\mathbf{r}(u)] = \frac{1}{b^5} \int_U y''(x)^2 \, dx. \tag{3.73}$$

This functional is convex and quadratic. The stablizer is a semi-norm on the class of functions having a square integrable second derivative on U. The null space of the stablizer is spanned by $\{1, x\}$. The Euler-Lagrange differential equation of the stablizer is

$$y''''(x) = 0. \tag{3.74}$$

Thus the reconstruction is a natural cubic spline. This stablizer has been widely used for smoothing data [70].

3.7.2 Weighted Cubic Splines

The stablizer in the previous section can include a positive weighting factor, $w(x)$, to form a closer approximation to the invariant stablizer. The stablizer is defined as [52]

$$\Omega_4[\mathbf{r}(u)] = \int_U w(x)\, y''(x)^2 \, dx. \tag{3.75}$$

The weighting factor can be used to incorporate discontinuity information ($w(x) = 0$ at discontinuities). To closer approximate the invariant stablizer in (3.27), Salkauskas [72] chooses a piecewise constant weight function defined by

$$w(x) = \left\{ 1 + \left[\frac{y(x_i) - y(x_{i-1})}{x_i - x_{i-1}} \right]^2 \right\}^{-3}, \qquad x \in [x_{i-1}, x_i), \quad i = 1,\ldots,M. \tag{3.76}$$

Examples using Ω_3 and Ω_4 are presented in the next section.

3.8 Examples

To demonstrate the effectiveness of the new algorithms in reconstructing invariant curves, a symmetric example was constructed so that it is easy to observe the effect of rotating and translating the data. Figures 3.2 to 3.5 show the curves that were reconstructed using stablizers Ω_1 to Ω_4 respectively. In all examples the regularizing parameters were set to $\lambda_t = 1.0$ and $\lambda = 0.01$. This choice was based on empirical evidence. The $\times$ mark indicate location constraints placed on the reconstruction.

The solid lines in Figures 3.6 to 3.9 were constructed by first rotating the coordinate system by 45° and then forming the reconstructed curve estimate with stablizers Ω_1 to Ω_4, respectively. The dashed lines in Figures 3.6 to 3.9 were formed by rotating Figures 3.2 to 3.5 by 45° to show how the reconstructions differ with a change in the coordinate system. Notice that the reconstructions formed by approximating (3.27) with Ω_3 or Ω_4 vary greatly with the orientation of the data. The curves reconstructed with Ω_1 and Ω_2, however, are much more robust to data orientation.

The four stablizers discussed in this chapter $\Omega_1 - \Omega_4$ are all convex approximations to the invariant stablizer (3.27). To determine which algorithm which best approximates the nonconvex invariant functional the value of the functional (3.28) is computed for each algorithm. The results are listed in Table 3.1 for both the original

constraints and the rotated constraints.

Table 3.1. Value of M^{λ} ($\lambda = 0.01$)

Data Set	Functional Minimized			
	M_1^{λ}	M_2^{λ}	M_3^{λ}	M_4^{λ}
Original	0.091070	0.116043	0.603425	0.369461
Rotated	0.098802	0.113685	0.366588	0.383489

As is to be expected minimizing M_1^{λ} has the best overall reconstruction for this example, however, the functional M_2^{λ} achieves good results with lower computational cost. Since the convex stablizers that were examined are approximations to the invariant stablizer (3.27) the reconstructions do varying some with a change in the coordinate system, this can be seen in the plots. To give a quantitative measure of this variance the distance between the two reconstructions was computed using the metric induced by the L^2 and L^{∞} norms. The results are listed in Table 3.2 for both metrics.

Table 3.2. Distance between reconstructions

Metric	Functional Minimized			
	M_1^{λ}	M_2^{λ}	M_3^{λ}	M_4^{λ}
L^2	7.829494	4.045463	91.872093	66.404761
L^{∞}	1.336573	0.671044	12.311476	10.089072

Figures 3.10 and 3.11 show the algorithms used to reconstruct curve estimates from noisy data. Figure 3.10 is a collection of noisy constraint data, marked with ×'s, of the dashed curve, and Figure 3.110 is the curve reconstructed with Ω_2. Figures 3.12 and 3.13 demonstrate that the new algorithms can be used to form invariant curve reconstructions in higher-dimensional spaces. Figure 3.12 shows location constraints and their ordering in three-dimensional space, and Figure 3.13 shows the curve reconstructed with Ω_2.

3.9 Applications in Computer Vision

This section describes two curve reconstruction problems which arise in computer vision problems.

3.9.1 Approximating Endocardial and Epicardial Boundaries

Digital two-dimensional echocardiography is an ultrasonic imaging technique that is used as an important noninvasive technique in the comprehensive characterization of the left ventricular structure and function [20, 106]. Quantitative analysis of the cardiac function often uses shape attributes such as thickness of the heart wall, the enclosed area, and the measurement of the variation of these shape attributes throughout the cardiac cycle [28]. These analyses require the complete determination of inner (endocardial) and outer (epicardial) boundaries of the heart wall. The echocardiograms, however, are of poor quality and the wall boundaries can be reliably detected at only sparse locations along the heart walls. This sparse information must therefore be used to recover an estimate of the heart wall boundaries.

The image in Figure 3.14 is a typical echocardiographic image of the left vertricle. The detected edge points are marked in white The edges were detected using a 41×41 $\nabla^2 G$ operator followed by a radial search to find zero-crossing points [20]. The detected edge points for the endocardial and epicardial boundaries are also plotted in Figure 3.15. The smooth and continuous reconstruction in Figure 3.16 was formed using Ω_2 as the stablizer. To form a closed curve reconstruction the assumption that $\mathbf{r}(u)$ is periodic with a period equal to the arc length of the piecewise linear reconstruction is included. Figure 3.17 shows the approximated boundaries overlaid on the original image.

3.9.2 Approximating Motion Trajectories

An important problem in machine vision is the detection of a target and estimation of its motion in a field of view. Several techniques have been developed which form

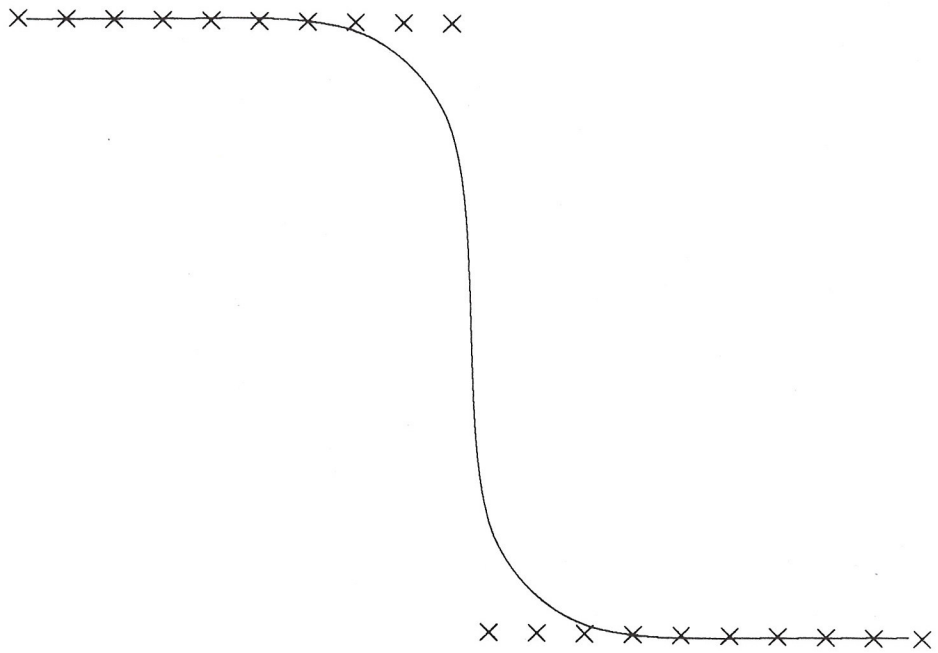

Figure 3.2. Curve reconstructed with Ω_1, (3.46)

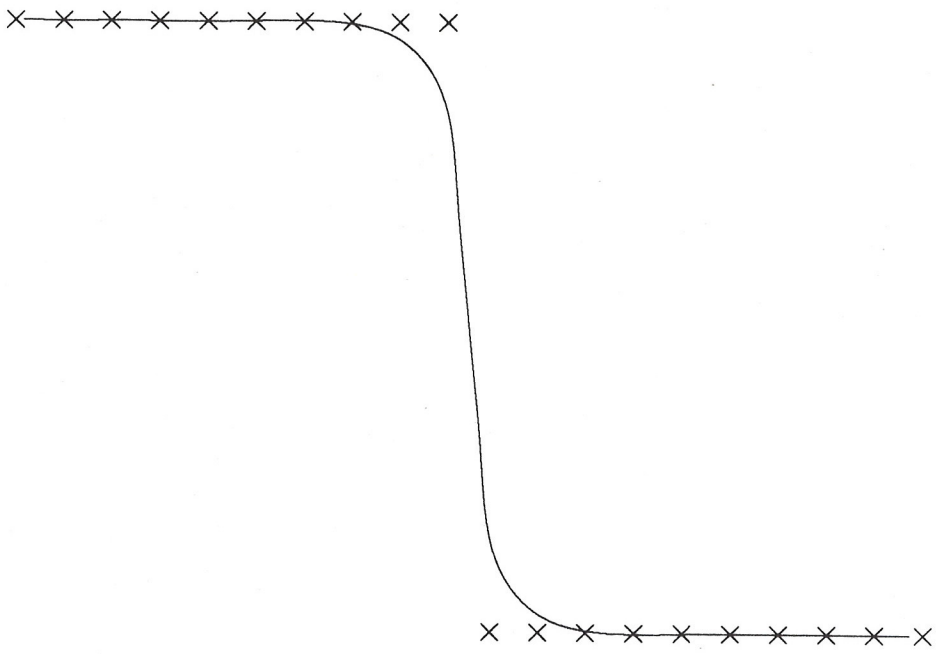

Figure 3.3. Curve reconstructed with Ω_2, (3.66)

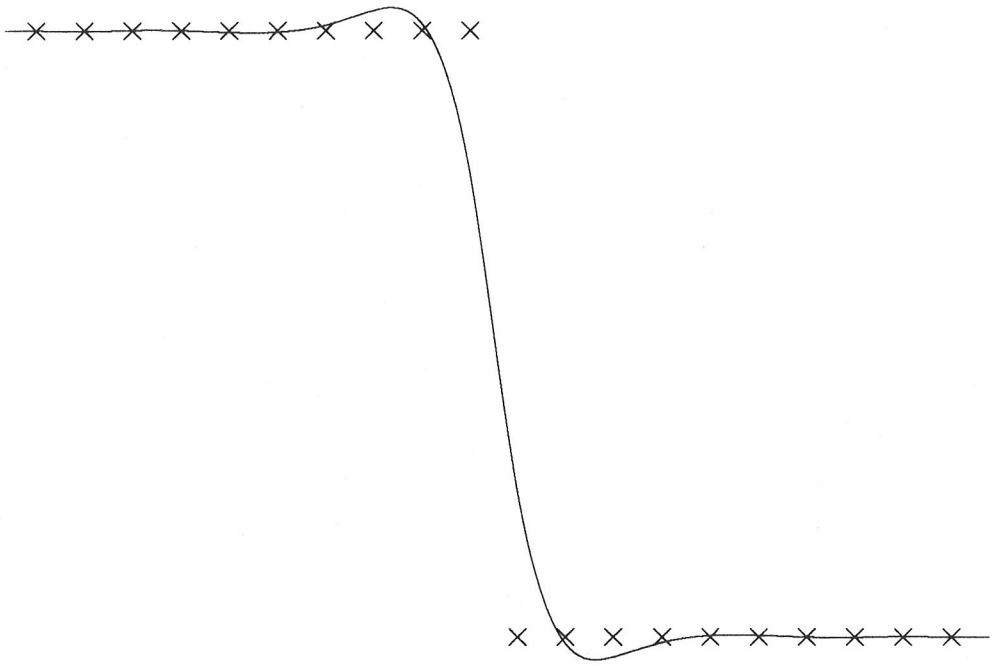

Figure 3.4. Curve reconstructed with Ω_3, (3.73)

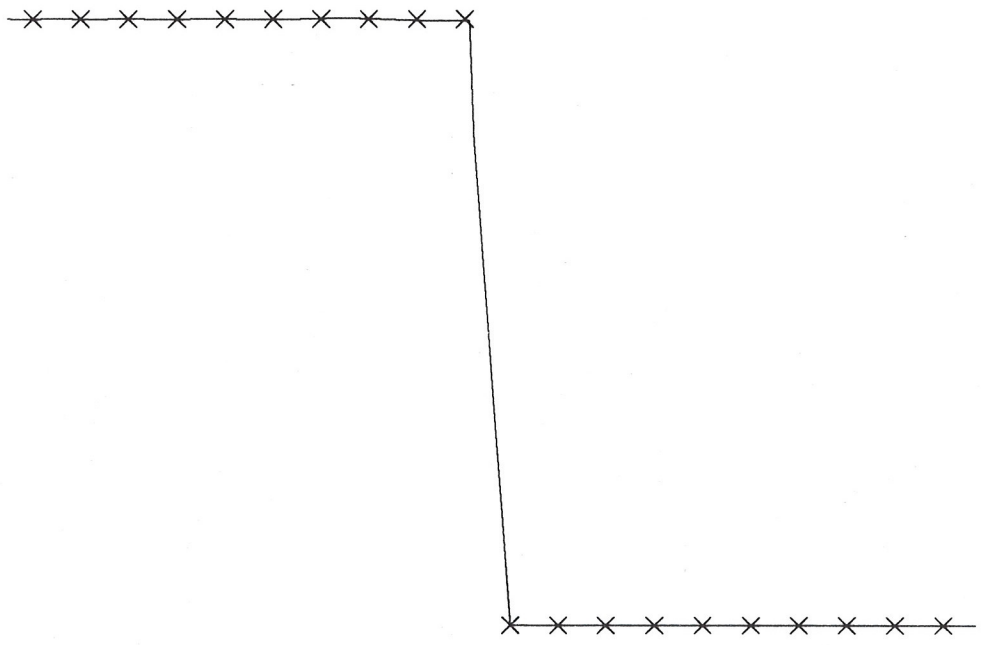

Figure 3.5. Curve reconstructed with Ω_4, (3.75)

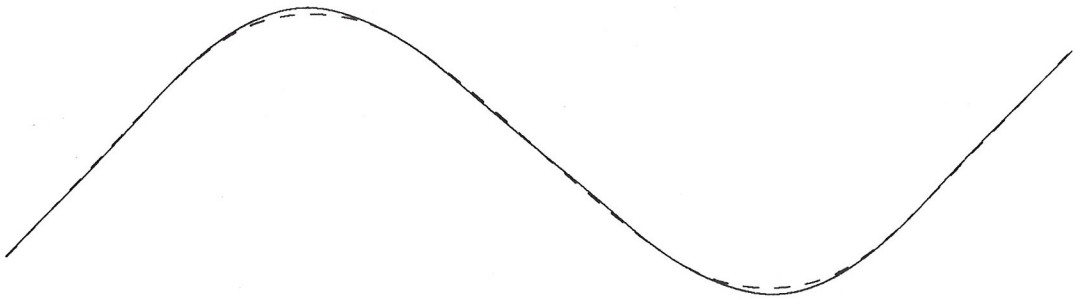

Figure 3.6. Rotated curves, reconstruction done with Ω_1, (3.46)

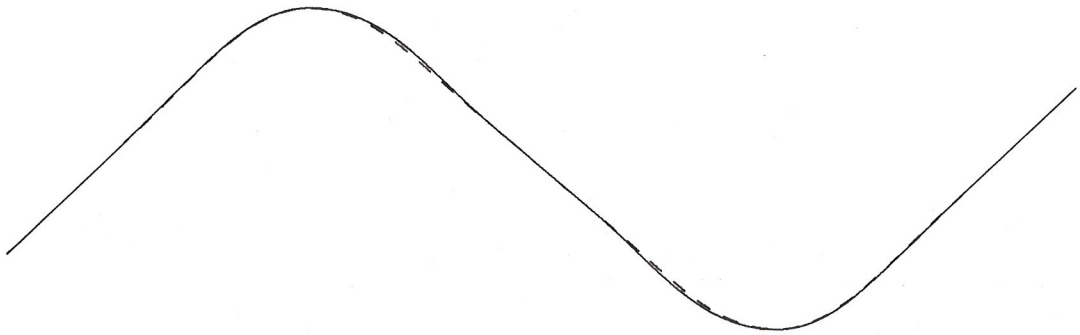

Figure 3.7. Rotated curves, reconstruction done with Ω_2, (3.66)

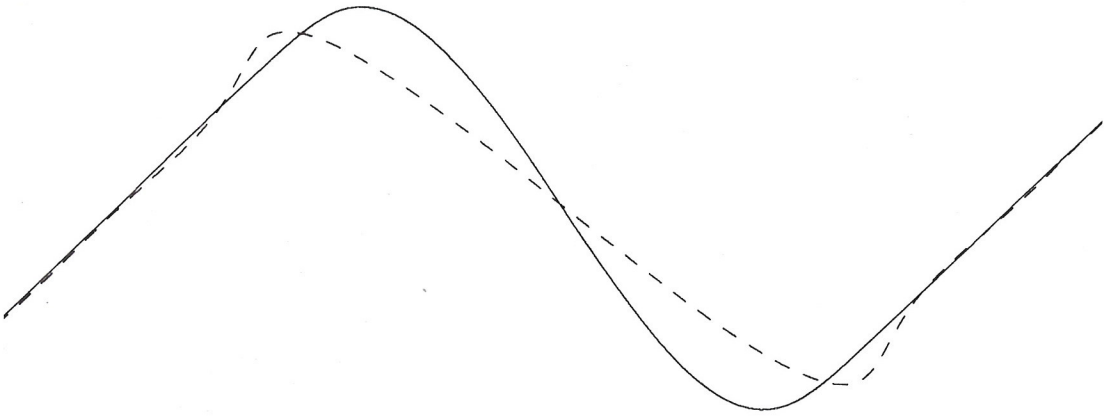

Figure 3.8. Rotated curves, reconstruction done with Ω_3, (3.73)

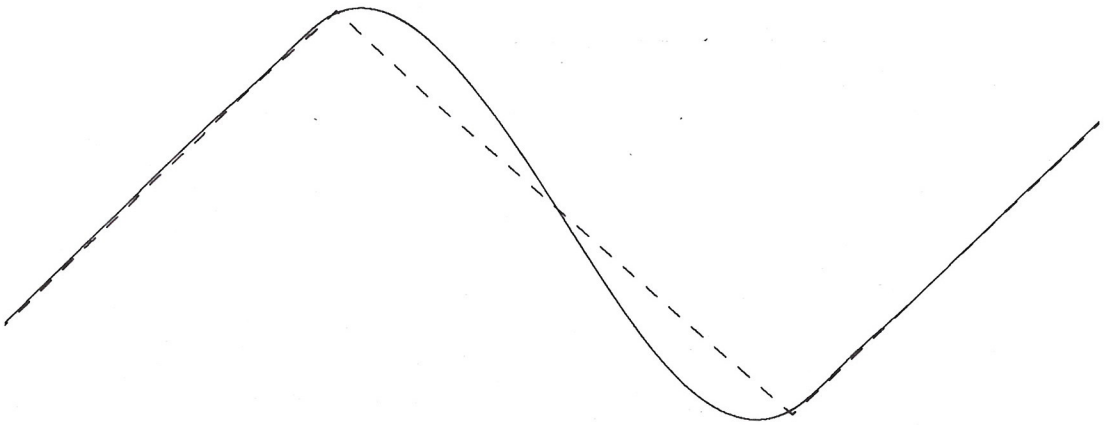

Figure 3.9. Rotated curves, reconstruction done with Ω_4, (3.75)

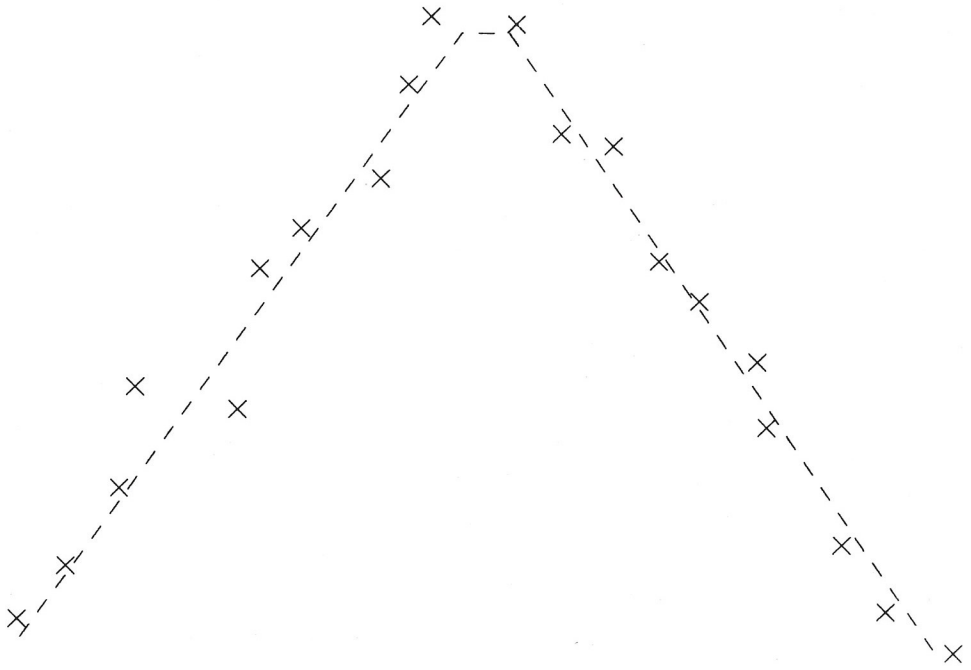

Figure 3.10. Noisy and sparse data

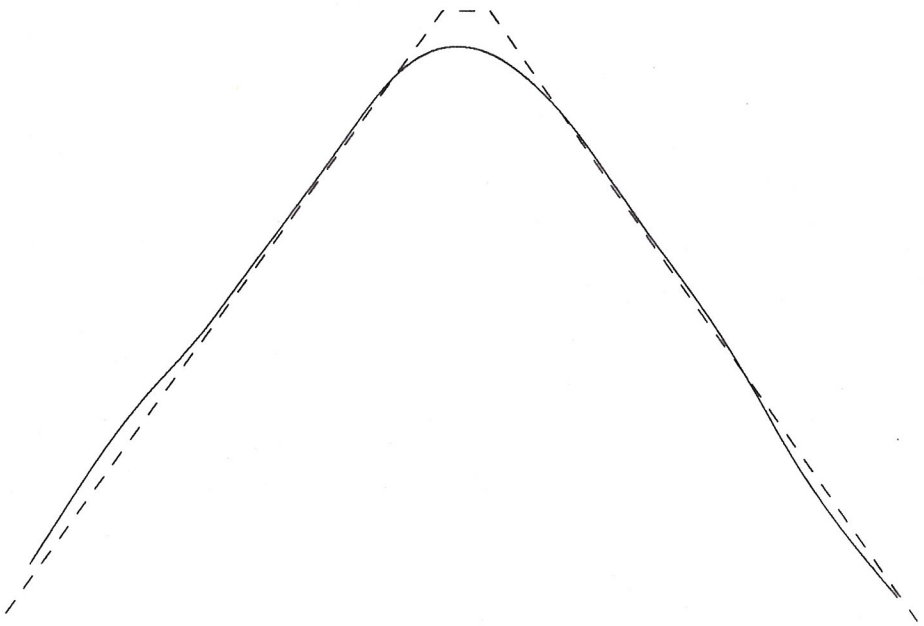

Figure 3.11. Curve reconstructed from noisy data with Ω_2, (3.66)

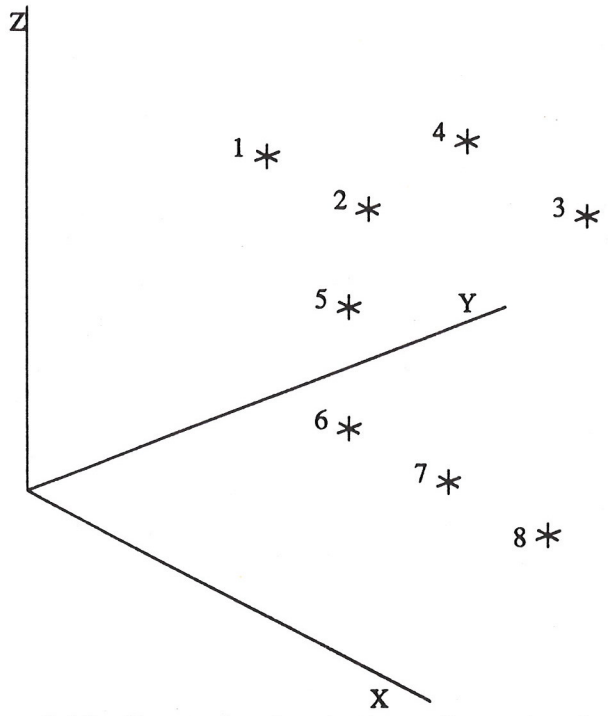

Figure 3.12. Constraint data in three-dimensional space

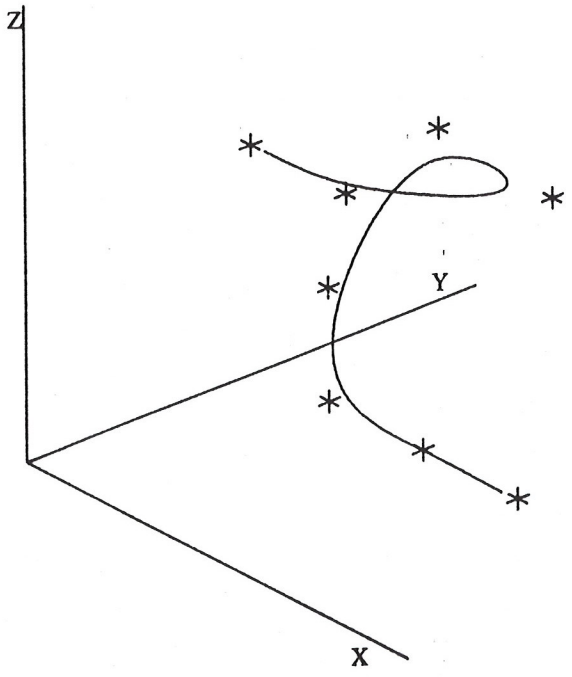

Figure 3.13. Curve reconstructed with Ω_2, (3.66), in three-dimensional space

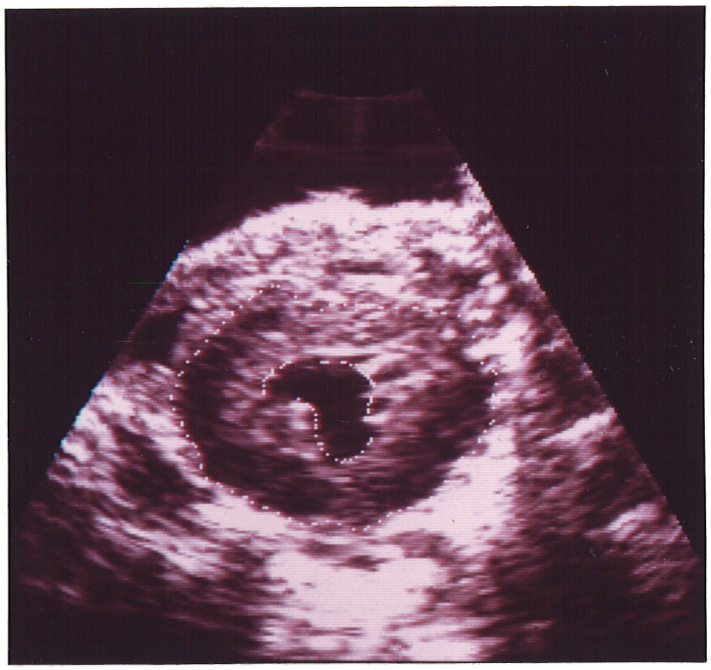

Figure 3.14. Echocardiographic image of left ventricle

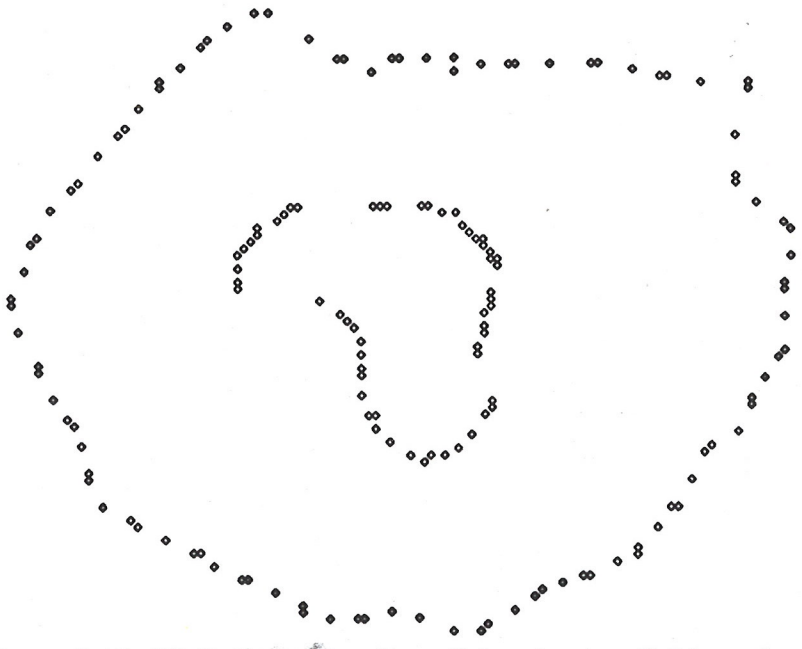

Figure 3.15. Well detected endocardial and epicardial boundary points

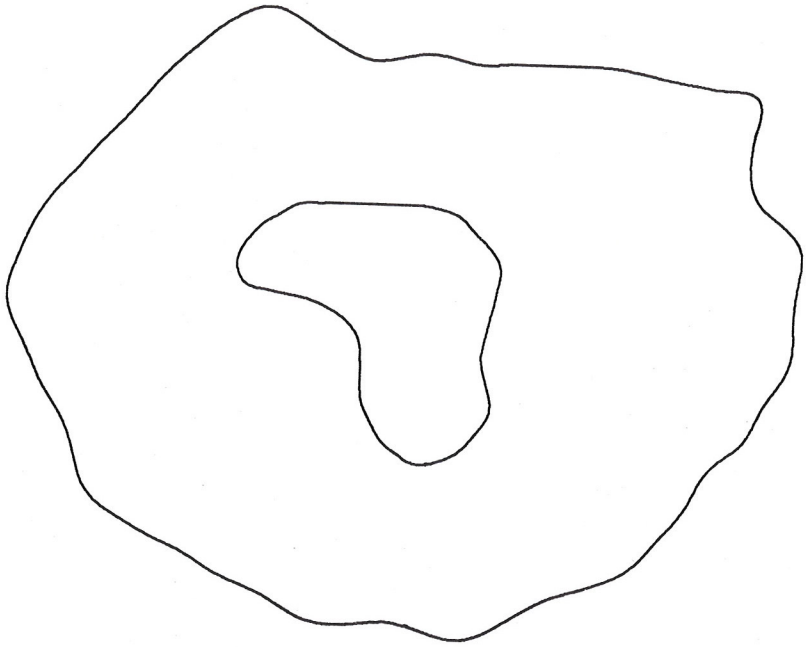

Figure 3.16. Approximated endocardial and epicardial boundaries

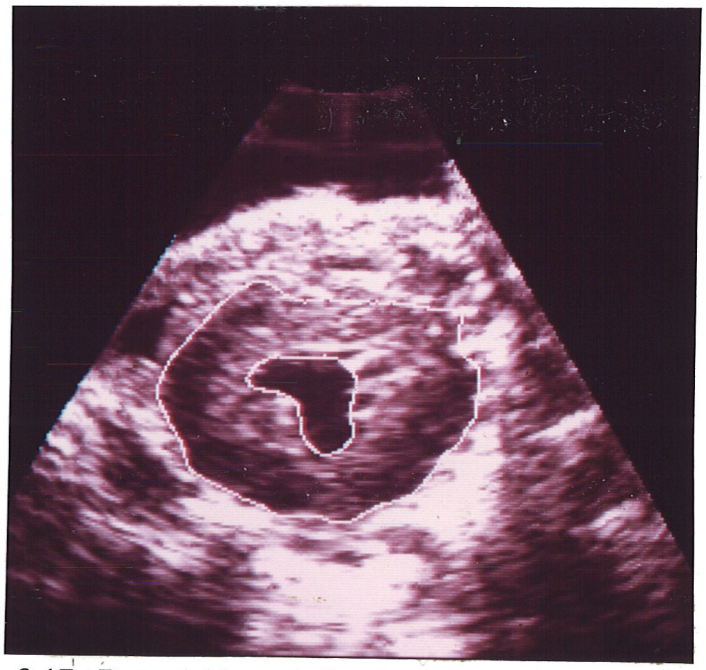

Figure 3.17. Detected boundaries overlaid on the original image

noisy measurements of the object trajectory from either a sequence of images [44, 56, 86]. or a single image using time-sequential sampling [51]. To estimate the trajectory from these data points a smooth continuous approximation to the data points is found.

The image in Figure 3.18 is one time-sequentially sampled image of a moving target on a background. The image is not sampled in the common lexicographic (or raster) pattern, rather the sampling is done on a bit-reversed dot-interlaced pattern [51]. Since the object is moving while the single frame is being sampled, the moving object appears dispersed. In Figure 3.19, the object location data is plotted. The third axis is the time at which the sample was taken The approximated trajectory is plotted in Figure 3.20 and is overlaid on the original image in Figure 3.21.

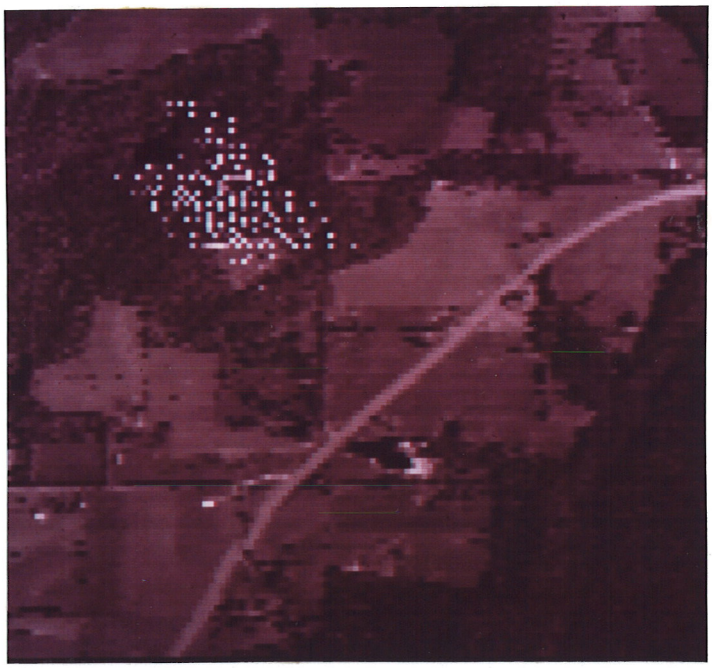

Figure 3.18. One frame of a time-sequentially sampled scene with a moving object

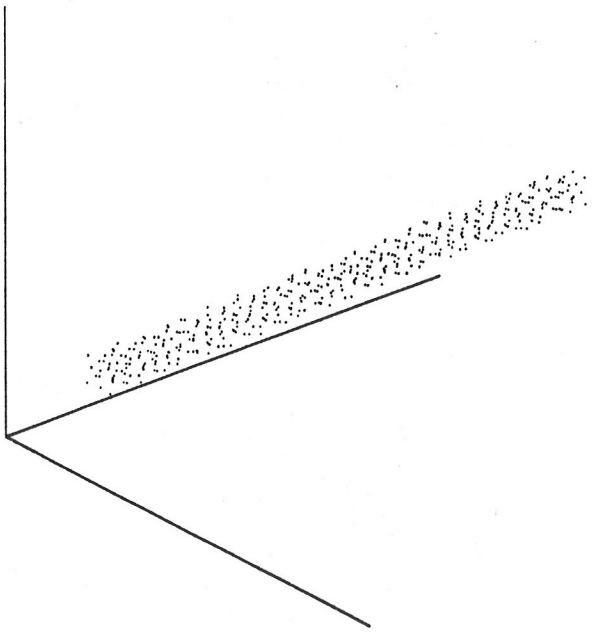

Figure 3.19. Estimated object locations from a time-sequentially sampled image

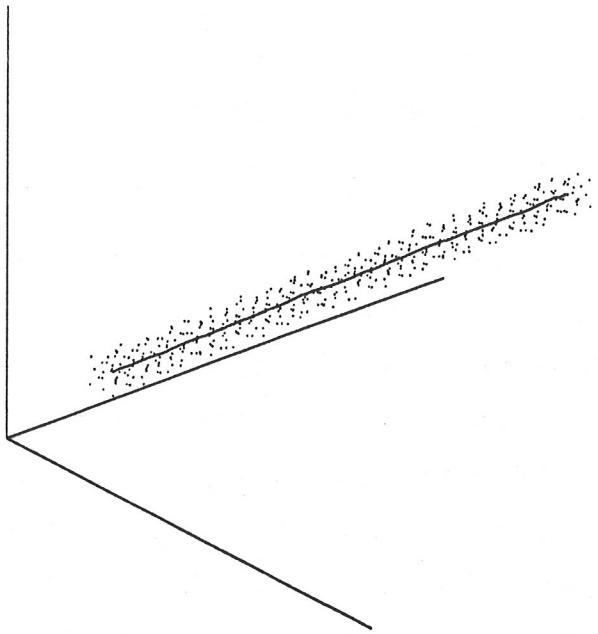

Figure 3.20. Approximated object trajectory

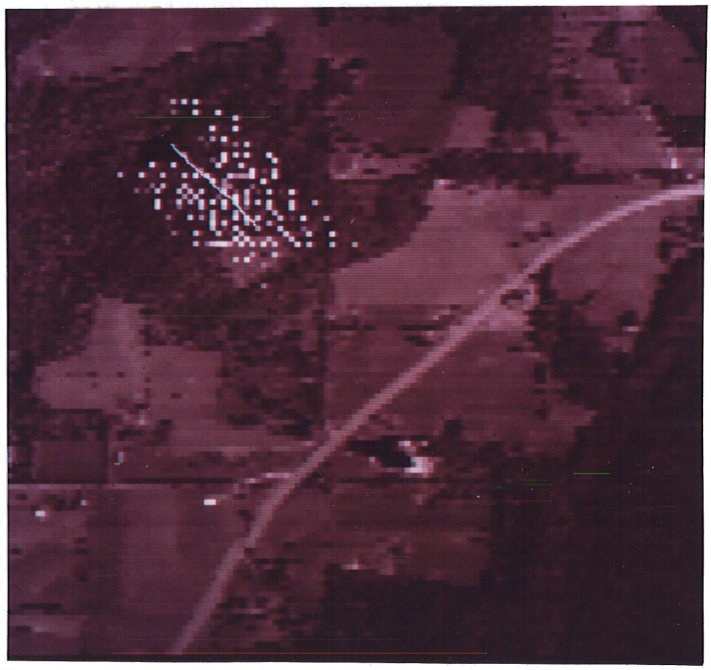

Figure 3.21. Approximated object trajectory overlaid on the original image

$$L = (-\mathbf{r}_u(\mathbf{u}),\mathbf{n}_u(\mathbf{u})) \qquad M = -\frac{1}{2}[(\mathbf{r}_u(\mathbf{u}),\mathbf{n}_v(\mathbf{u})) + (\mathbf{r}_v(\mathbf{u}),\mathbf{n}_u(\mathbf{u}))] \tag{4.8}$$

$$N = (-\mathbf{r}_v(\mathbf{u}),\mathbf{n}_v(\mathbf{u}))$$

is known as the second fundamental form. The coefficients L, M, and N can also be given by [57]:

$$L = (\mathbf{r}_{uu}(\mathbf{u}),\mathbf{n}(\mathbf{u})) \qquad M = (\mathbf{r}_{uv}(\mathbf{u}),\mathbf{n}(\mathbf{u})) \qquad N = (\mathbf{r}_{vv}(\mathbf{u}),\mathbf{n}(\mathbf{u})). \tag{4.9}$$

These six coefficients in (4.3) and (4.9) can be shown to completely characterize a surface in three-dimensional space [9]. These fundamental coefficients can be combined into invariant functions with easily interpretable surface shape characteristics. The Gaussian curvature function of a surface can be defined from the first and second fundamental form matrices as

$$K(\mathbf{u}) = \det\left(\begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1}\begin{bmatrix} L & M \\ M & N \end{bmatrix}\right) = \frac{LN - M^2}{EG - F^2}, \tag{4.10}$$

where det() indicates the determinate. The mean curvature function of a surface can be defined similarly as

$$H(\mathbf{u}) = \frac{1}{2}\mathrm{tr}\left(\begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1}\begin{bmatrix} L & M \\ M & N \end{bmatrix}\right) = \frac{EN + GL - 2FM}{2(EG - F^2)}, \tag{4.11}$$

where tr() is the trace operator. The matrix

$$\Gamma = \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1}\begin{bmatrix} L & M \\ M & N \end{bmatrix} \tag{4.12}$$

is known as the Weingarten mapping, which maps a tangent vector to a tangent vector in the tangent associated plane with each point [9]. The Gaussian and mean curvatures

$$H(\mathbf{u}) = \frac{(\kappa_1(\mathbf{u}) + \kappa_2(\mathbf{u}))}{2}. \tag{4.20}$$

If the normal curvature at a point is constant for all directions, the Weingarten map will not have two distinct eigenvectors. In this case, if $\kappa_n(\mathbf{u}) = 0$ at this point, then the point represents a point on a planar surface patch. For this point all directions are considered principle directions. If $|\kappa_n(\mathbf{u})| > 0$, this point is known as an umbilical point of the surface. As an example, all the points on the surface of a sphere are umbilical points.

The local characteristics of a surface at a point P on the surface is completely specified by the surface normal, the principle directions and principle curvatures. Following the convention of [73, 74] the collection of information $(P, \mathbf{n}, \mathbf{d}_1, \mathbf{d}_2, \kappa_1, \kappa_2)$ is referred to as the augmented Darboux frame at P, $\mathcal{D}(P)$.

4.2 Invariant Metrics on the Constraint Spaces

The metrics on the constraint space for the surface reconstruction problem are very similar to the metrics used in reconstructing curves. For location constraints the Euclidean distance of the surface at $\mathbf{u}_i$ to the constraint location is used as the metric,

$$\rho_{C^i}(\mathbf{r}(\mathbf{u}_i), \mathbf{c}_i) = \left[\sum_{j=1}^{m} [x_j(\mathbf{u}_i) - \mathbf{c}_{j,i}]^2 \right]^{1/2}$$

$$= ||\mathbf{r}(\mathbf{u}_i) - \mathbf{c}_i||. \tag{4.21}$$

For normal constraints situation is more complex than the curve reconstruction case but essentially the same. $\mathbf{l}_i$ is given by (1.12) which is equal to

$$\frac{\mathbf{r}_u \times \mathbf{r}_v}{||\mathbf{r}_u \times \mathbf{r}_v||} = \frac{<(y_u z_v - y_v z_u), (z_u x_v - z_v x_u), (x_u y_v - x_v y_u)>}{((y_u z_v - y_v z_u)^2 + (z_u x_v - z_v x_u)^2 + (x_u y_v - x_v y_u)^2)^{1/2}} \tag{4.22}$$

and

based on the total length of the curve in Section 3.3.1. As was the case for the first-order curve stablizer, the surface reconstructed with this stablizer will not be sufficiently smooth. In fact, they will be piecewise planar [67].

4.3.2 Second-Order Stablizer

For smoothly varying surfaces, the surface can be modeled as an ideal thin flexible plate of elastic material [80-82, 84, 85]. The stablizer is a measure of the strain energy of the deformed plate. The potential energy density of a thin plate is given by [23]

$$A\left[\frac{\kappa_1^2(\mathbf{u})+\kappa_2^2(\mathbf{u})}{2}\right] + B\kappa_1(\mathbf{u})\kappa_2(\mathbf{u}), \tag{4.26}$$

where A and B are constants of the material. This can be written as

$$2A\left[\frac{\kappa_1(\mathbf{u})+\kappa_2(\mathbf{u})}{2}\right]^2 - (A-B)\kappa_1(\mathbf{u})\kappa_2(\mathbf{u}), \tag{4.27}$$

or in terms of Gaussian and mean curvatures as

$$2AH(\mathbf{u})^2 - (A-B)K(\mathbf{u}). \tag{4.28}$$

To simplify this equation, let $A = 1$ and $B = 0$. With this assumption the equation still models a valid thin plate [80]. An invariant stablizer can be defined by integrating this energy density over the surface area [80]

$$\Omega[\mathbf{r}(\mathbf{u})] = \int_U (2H^2(\mathbf{u}) - K(\mathbf{u}))\, dA, \tag{4.29}$$

or in terms of the principle curvatures as

$$\Omega[\mathbf{r}(\mathbf{u})] = \int_U (\kappa_1^2(\mathbf{u}) + \kappa_2^2(\mathbf{u}))\, dA. \tag{4.30}$$

The reconstructed $\mathbf{r}$ obtained by the minimization of M^λ with this stablizer is referred to as the spline approximate. For smooth invariant surface reconstruction (4.30) is an adequate stablizer and will be the stablizer used in this paper.

4.4 An Invariant Problem Statement

The problem of reconstructing a surface given a set of constraints $\mathcal{S}$ is now posed as a functional minimization problem. The estimate of the surface is formed by minimizing the functional

$$M^\lambda[\mathbf{r}(\mathbf{u}),\mathcal{S}] = \iint_U \left[\kappa_1(\mathbf{u})^2 + \kappa_2(\mathbf{u})^2 \right] \|\mathbf{r}_u(\mathbf{u})\times\mathbf{r}_v(\mathbf{u})\| \, du \, dv$$

$$+ \lambda \sum_{i=1}^{M} \rho_{C_i}(\mathbf{r}(\mathbf{u}_i),\mathbf{c}_i)^2, \tag{4.31}$$

where $\kappa_1(\mathbf{u})$ and $\kappa_2(\mathbf{u})$ is given by (4.17) and (4.18), respectively and $\rho_{C_i}(\cdot,\cdot)$ is described in Section 4.2. The explicit form of the functional can be written as

$$M^\lambda[z,\mathcal{S}] = \iint_U \left\{ \frac{[(1+z_y^2)z_{xx} - 2z_x z_y z_{xy} + (1+z_x^2)z_{yy}]^2}{2(1+z_x^2+z_y^2)^{5/2}} \right.$$

$$\left. - \frac{z_{xx}z_{yy} - z_{xy}^2}{(1+z_x^2+z_y^2)^{3/2}} \right\} dxdy + \lambda \sum_{i=1}^{M} \rho_{C_i}(z(x_i,y_i),\mathbf{c}_i)^2. \tag{4.32}$$

The constant λ determines a tradeoff between the smoothness of the curve and closeness of fit of the surface to the data. To show that this problem is well-posed it must be shown that the minimization of this functional exists and is unique. As was the case with the curve reconstruction problem a simple example can be constructed to show that this problem is not well-posed.

The stablizer Ω, (4.30), can be shown to be nonconvex by first showing that

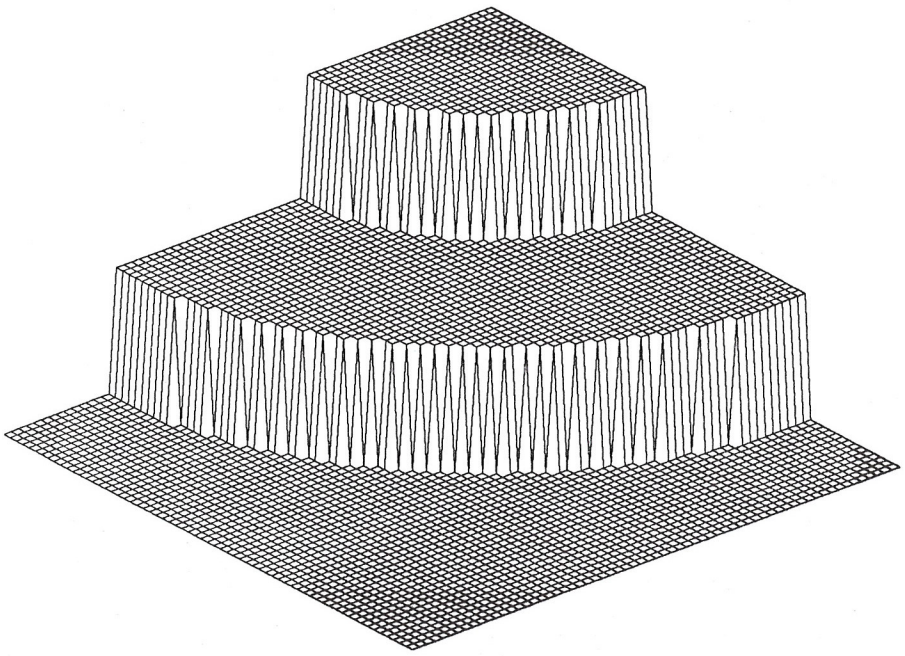

Figure 4.1. Synthetic Surface

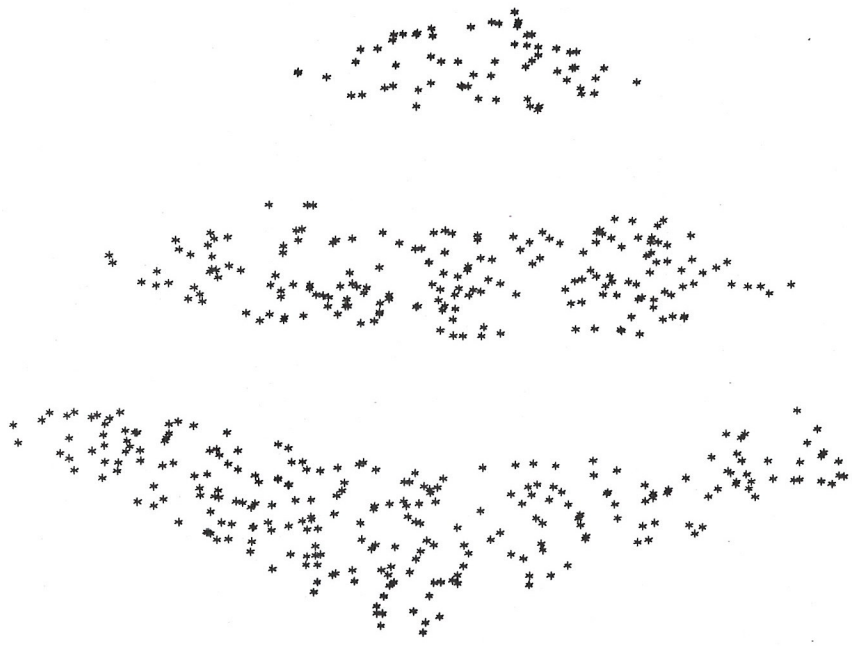

Figure 4.2. A collection of 500 noisy surface constraints

approximation. A uniform mesh is placed on this surface, and surface characteristics are estimated at the nodes of the mesh.

A Thiessen triangulation, $\mathcal{T}$, of the surface is found using an algorithm which was adapted from an algorithm proposed by A. K. Cline *et. al.* [22, 71]. This algorithm is guaranteed to terminate with a valid Thiessen triangulation of the surface after a finite number of steps [71]. Definition of a Thiessen triangulation and the algorithm are outlined in Appendix B. The resulting triangulation of the constraints is shown in Figure 4.3 (projected into the XY plane). Other triangulation methods such as those proposed by Cavendish [18] and Lawson [53] can also be applied to form the triangulation.

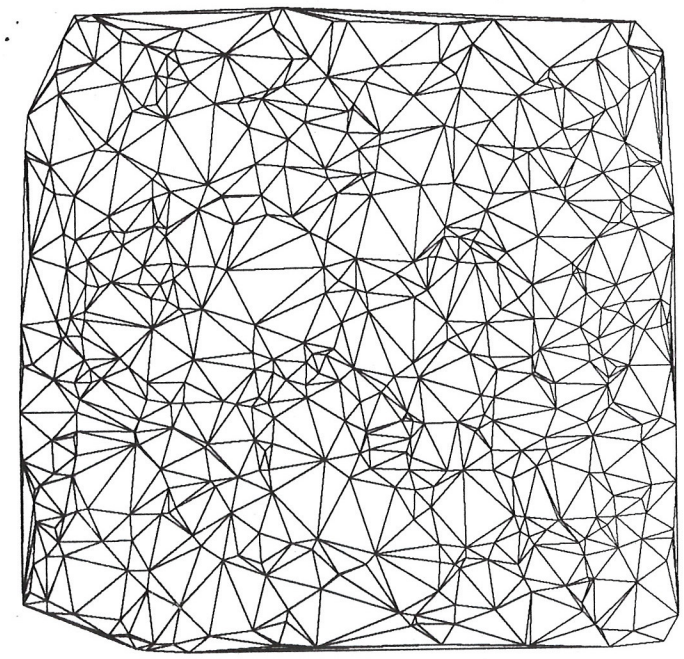

Figure 4.3. Thiessen triangulation of the constraint data set

The set of constraint $\mathcal{S}$ and the triangulation, $\mathcal{T}$, define a continuous piecewise planar surface estimate (i.e. the surface within a triangle is given by the plane which passes through the three constraint nodes). This surface estimate is first smoothed before the surface characteristics are estimated. The smoothing is performed by using a weighted

least squares smoother. An arc in the triangulation can be described by the indices of the two end nodes. Let (e_1, e_2) denote the indices of an arc in the triangulation. The collection of arcs in the triangulation $\mathcal{T}$ is denoted by $\mathcal{E}$. Let $\mathcal{S}^*$ be a collection of nodes $\{\mathbf{c}_i^* \in C_i, i=1, \cdots, M\}$ which with $\mathcal{T}$ describe an arbitrary planar surface. A smoothed piecewise planar surface is computed by minimizing the functional

$$M^{\lambda_p}[\mathcal{S}^*, \mathcal{S}] = \sum_{(e_1,e_2)\in\mathcal{E}} w_{e_1,e_2} ||\mathbf{c}_{e_1}^* - \mathbf{c}_{e_2}^*||^2 + \lambda_p \sum_{i=1}^{M} ||\mathbf{c}_i^* - \mathbf{c}_i||^2 \tag{4.34}$$

with respect to the collection $\mathcal{S}^*$. The constant weight term is given by

$$w_{e_1,e_2} = \frac{1}{||\mathbf{c}_{e_1} - \mathbf{c}_{e_2}||} \tag{4.35}$$

and is used so that nodes which are spatially far apart (e.g. across jump discontinuities) have little smoothing effect on each other even if they are adjacent in the triangulation. The function (4.34) is convex since each term is quadratic. The function (4.34) is also invariant since each term is based on an invariant quantity (the distance between two nodes). Let the collection of $\mathcal{S}^*$ that minimizes (4.34) be denoted by $\hat{\mathcal{S}} = \{\hat{\mathbf{c}}_i \in C_i, i=1, \cdots, M\}$.

The function (4.34) is based on the physical model of finding the minimum energy configuration of a set of springs. The first term represents the energy of a collection of springs, one along each arc in the triangulation. The weight term, w_{e_1,e_2}, is related to the spring coefficient, a larger weight corresponds to a stronger spring. The second term in (4.34) represents the energy of a collection of M springs, each one connecting a constraint location with its corresponding location on the surface. The term λ_p controls the ratio of the spring constants between the two sets of springs. The continuous piecewise planar surface estimate resulting from this minimization with, $\lambda_p = 0.1$, for our example is given in Figure 4.4.

On this continuous planar surface estimate a uniform grid (uniform in the XY plane) is placed. At the grid nodes surface characteristics for the smooth continuous surface estimate are estimated. For points which are outside the convex hull of the constraint

points (i.e. not interior to any triangle) the plane of the nearest triangle is extended to form a surface estimate. The grid on our example surface is shown in Figure 4.5.

In order to make the desired approximations to (4.31), we need estimates of the normal, $\mathbf{n}$, and the principle directions, $\mathbf{d}_1$ and $\mathbf{d}_2$ at each node on the mesh. While local least square estimation techniques are usually sufficient to ensure stable estimates of first-order characteristics (e.g. $\mathbf{n}$), they are rarely adequate for the estimation of second-order properties (e.g. $\mathbf{d}_1, \mathbf{d}_2$). Our approach, which is similar to [73, 74], is to first form local estimates of the Darboux frame $\mathcal{D}$ and then to iteratively refine the estimates to obtain a stable reconstruction of $\mathcal{D}$. The algorithm for estimating and refining the Darboux frame is described in Appendix C. When the difference between the principle curvatures is small the estimate of the principle directions becomes unreliable. Therefore, whenever $|\kappa_1(\mathbf{u}) - \kappa_2(\mathbf{u})| < 0.2h$, where h is the distance between samples in the XY plane (h=1 in example), it is assumed that the estimate is not accurate enough to be used. In Figure 4.6 the vector $\mathbf{d}_1$ is plotted at each grid location. At points where the estimate of $\mathbf{d}_1$ is unreliable a dot is plotted in place of the vector.

4.5.2 Stage 2: A Smooth invariant surface estimate

Using the estimates of $\mathcal{D}(P)$ formed in Stage 1, estimates of $\mathbf{r}_u(\mathbf{u})$, $\mathbf{r}_v(\mathbf{u})$, $\theta_1(\mathbf{u})$, $\theta_2(\mathbf{u})$ and $\mathbf{n}(\mathbf{u})$ can be used to compute approximations to the principle curvatures κ_1 and κ_2 by

$$\kappa_1(\mathbf{u}) = \omega_{11}(\mathbf{u})(\mathbf{r}_{uu}(\mathbf{u}),\mathbf{n}(\mathbf{u})) + \omega_{12}(\mathbf{u})(\mathbf{r}_{uv}(\mathbf{u}),\mathbf{n}(\mathbf{u})) + \omega_{13}(\mathbf{u})(\mathbf{r}_{vv}(\mathbf{u}),\mathbf{n}(\mathbf{u})), \quad (4.36)$$

and

$$\kappa_2(\mathbf{u}) = \omega_{21}(\mathbf{u})(\mathbf{r}_{uu}(\mathbf{u}),\mathbf{n}(\mathbf{u})) + \omega_{22}(\mathbf{u})(\mathbf{r}_{uv}(\mathbf{u}),\mathbf{n}(\mathbf{u})) + \omega_{23}(\mathbf{u})(\mathbf{r}_{vv}(\mathbf{u}),\mathbf{n}(\mathbf{u})), \quad (4.37)$$

where the ω's are constant functions given by

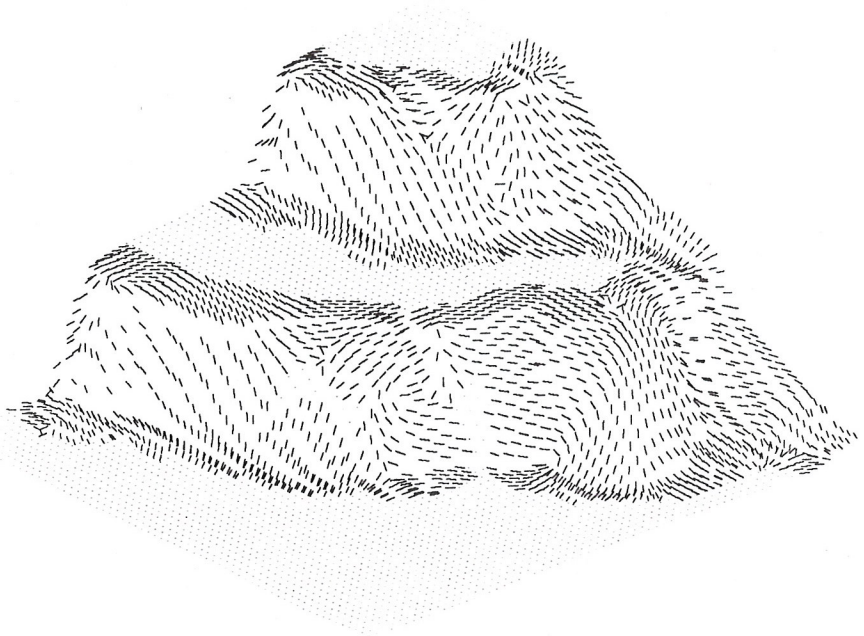

Figure 4.6. Estimated principle direction field

$$\omega_{k1} = \frac{\cos^2\hat{\theta}_k}{(\hat{\mathbf{r}}_u,\hat{\mathbf{r}}_u)\cos^2\hat{\theta}_k + 2(\hat{\mathbf{r}}_u,\hat{\mathbf{r}}_v)\cos\hat{\theta}_k \sin\hat{\theta}_k + (\hat{\mathbf{r}}_v,\hat{\mathbf{r}}_v)\sin^2\hat{\theta}_k} \tag{4.38}$$

$$\omega_{k2} = \frac{2\cos\hat{\theta}_k \sin\hat{\theta}_k}{(\hat{\mathbf{r}}_u,\hat{\mathbf{r}}_u)\cos^2\hat{\theta}_k + 2(\hat{\mathbf{r}}_u,\hat{\mathbf{r}}_v)\cos\hat{\theta}_k \sin\hat{\theta}_k + (\hat{\mathbf{r}}_v,\hat{\mathbf{r}}_v)\sin^2\hat{\theta}_k} \tag{4.39}$$

$$\omega_{k3} = \frac{\sin^2\hat{\theta}_k}{(\hat{\mathbf{r}}_u,\hat{\mathbf{r}}_u)\cos^2\hat{\theta}_k + 2(\hat{\mathbf{r}}_u,\hat{\mathbf{r}}_v)\cos\hat{\theta}_k \sin\hat{\theta}_k + (\hat{\mathbf{r}}_v,\hat{\mathbf{r}}_v)\sin^2\hat{\theta}_k} \tag{4.40}$$

and a stablizer can be defined by

$$\iint_U [\omega_{11}(\mathbf{u})(\mathbf{r}_{uu}(\mathbf{u}),\mathbf{n}(\mathbf{u})) + \omega_{12}(\mathbf{u})(\mathbf{r}_{uv}(\mathbf{u}),\mathbf{n}(\mathbf{u})) + \omega_{13}(\mathbf{u})(\mathbf{r}_{vv}(\mathbf{u}),\mathbf{n}(\mathbf{u}))]^2$$

$$[\omega_{24}(\mathbf{u})(\mathbf{r}_{uu}(\mathbf{u}),\mathbf{n}(\mathbf{u})) + \omega_{25}(\mathbf{u})(\mathbf{r}_{uv}(\mathbf{u}),\mathbf{n}(\mathbf{u})) + \omega_{26}(\mathbf{u})(\mathbf{r}_{vv}(\mathbf{u}),\mathbf{n}(\mathbf{u}))]^2 \, dudv \quad (4.41)$$

This functional is convex and leads to efficient computational solutions. However, when computing the solution using standard techniques a numerical instability can occur. When using iterative methods for computing the solution two points on the surface can come close together. Resulting in errors in the following iterations possibly causing the surface parameterization to have nonsingularities (i.e. overlap) which leads to further instability.

To overcome this numerical instability after each iteration the surface is resampled onto a uniform grid, which keeps the surface points from coming close to each other. The iteration and resampling can be combined into one step, the idea is diagramed in Figure 4.7. At iteration k an update vector $\delta\mathbf{r}^{(k)}(i,j)$ (when using the steepest descent algorithm described in Section 2.5, $\delta\mathbf{r}^{(k)} = \beta^{(k)}\mathbf{g}^{(k)}$) is computed for each point $\mathbf{r}^{(k)}(i,j)$ in the representation. Instead of setting $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} + \delta\mathbf{r}^{(k)}$ use $\delta\mathbf{r}^{(k)}$ to define a planar approximation to the new surface. The planar surface patch contains the point $\mathbf{r}^{(k)} + \delta\mathbf{r}^{(k)}$ and is orthogonal to the update vector $\delta\mathbf{r}^{(k)}$. Using this approximation the change just along the Z-axis is computed by

$$\delta z'(k)(i,j) = \frac{\|\delta\mathbf{r}^{(k)}(i,j)\|^2}{\delta z^{(k)}(i,j)} \quad (4.42)$$

where $\delta z^{(k)}$ is the Z-axis component of $\delta\mathbf{r}^{(k)}$ (see Figure 4.7). Thus instead of updating $\mathbf{r}^{(k)}$ with $\delta\mathbf{r}^{(k)}$ the update is computed by

$$\mathbf{r}^{(k+1)}(i,j) = \mathbf{r}^{(k)} + <0,0,\delta z'^{(k)}(i,j)>. \quad (4.43)$$

Since the second-order derivatives of x and y are zero on a uniform grid and since the update vector $\delta\mathbf{r}^{(k)}$ is always in the normal direction the stablizer can be reduced to the much simplified form

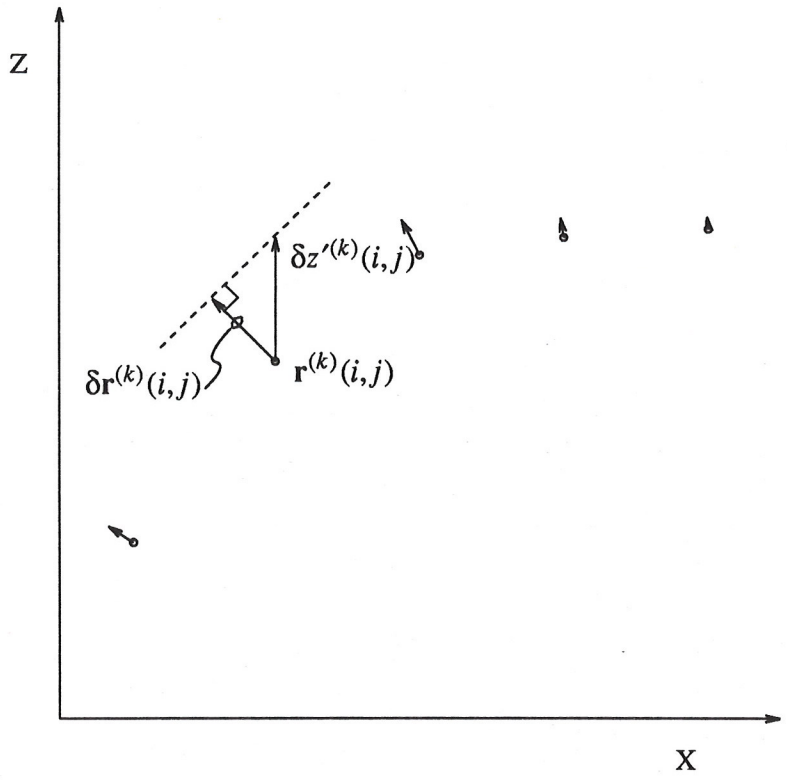

Figure 4.7. Updating along the Z axis

$$\Omega_1[\mathbf{r}(\mathbf{u})] = \iint_U [\omega_{11} z_{xx}+\omega_{12} z_{xy}+\omega_{13} z_{yy}]^2 + [\omega_{21} z_{xx}+\omega_{22} z_{xy}+\omega_{23} z_{yy}]^2 \, dxdy. \quad (4.44)$$

Since both terms are quadric the functional is convex. The surface resulting from the functional minimization for the example data set is shown in Figure 4.8.

4.6 Implementation

The are many methods by which the minimization of this functional can be performed. All of the methods used by [15, 26, 33, 65, 80-82] can be adapted to solve this minimization problem. For simplicity, the functional was discretized using finite difference methods and the minimization was performed using the conjugate gradient method. See Appendix A for details of this method.

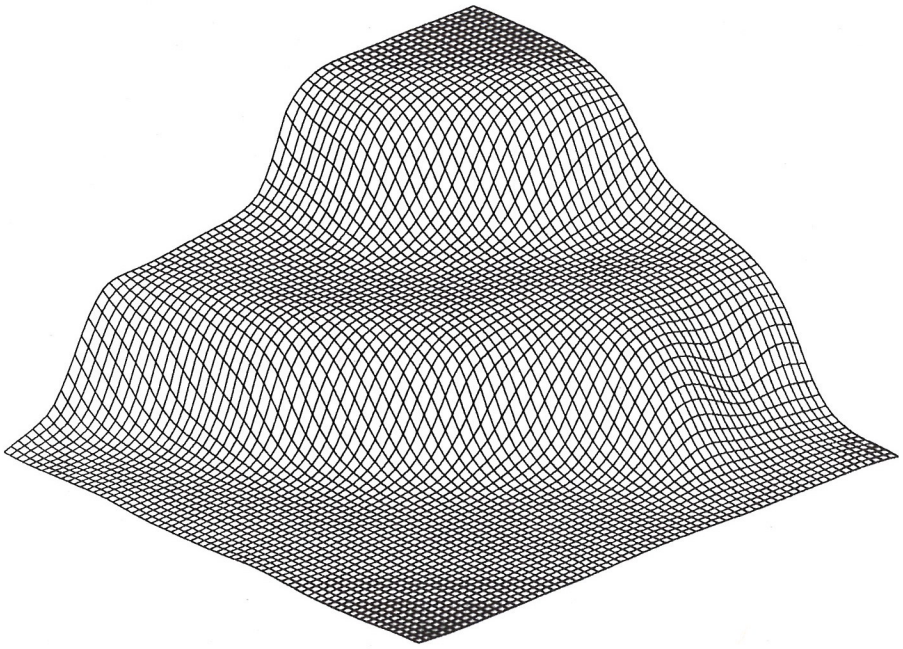

Figure 4.8. Reconstructed invariant surface estimate

The discrete version of the functional can be written as

$$M_d^{\lambda}[\mathbf{r},\mathbf{c}] = \frac{1}{2}\mathbf{r}^T(\mathbf{Q}_1 + \mathbf{Q}_2 + \lambda \mathbf{I}_S)\mathbf{r} - \mathbf{c}^T\mathbf{r}. \tag{4.45}$$

This matrix vector equation has the same form as (3.67) for the curve reconstruction problem, however the $\mathbf{Q}$ matrices are very different. The form of the $\mathbf{Q}$ matrices is given in Appendix D.

4.7 Comparison

This section describes two commonly used approximations to the nonconvex problem which are well-posed. Both of these algorithms use the explicit representation of a surface and are therefore limited to surfaces in three-dimensional space.

4.7.1 Quadratic Variation

The most common approximation made is to use the assumption

$$z_x \approx 0 \qquad z_y \approx 0. \tag{4.46}$$

Then the Gaussian and mean curvatures can be approximated by

$$H(x,y) \approx \frac{z_{xx} + z_{yy}}{2} \tag{4.47}$$

$$K(x,y) \approx z_{xx}z_{yy} - z_{xy}^2, \tag{4.48}$$

and the stablizer in (4.29) becomes

$$\Omega_2[z] = \frac{1}{2}\iint_U z_{xx}^2 + 2z_{xy}^2 + z_{yy}^2 \, dxdy. \tag{4.49}$$

This stablizer is not invariant to rigid three-dimensional transforms, and the assumption that z_x and z_y are small is often invalid. This model is usually referred to as the planar-plate model.

The planar-plate approximation is used in the work of Grimson [33] and Terzopoulos [80-82, 84, 85]. The null space of this stablizer is spanned by the set of all possible planes over the domain. The Euler-Lagrange differential equation of the stablizer is given by

$$z_{xxxx} + z_{yyyy} = 0. \tag{4.50}$$

This equation is referred to as the biharmonic equation [82].

There have been several methods developed to find the surface which minimizes the functional (4.29) under the planar-plate model assumptions. Grimson [33] uses the finite difference method to obtain a discrete problem which is then minimized by using

the conjugate gradient method. Terzopoulos [80] applies the finite-element method to transform the continuous variational principle into a discrete problem which is then solved using an efficient multigrid algorithm. The finite element method also has the advantage over the finite difference method of being able to be defined on nonrectangular grids. A third method proposed by Boult and Kender [15] uses the fact that the functional space is a reproducing kernel space. Duchon [26] and Meinguet [65] provide methods of obtaining the continuous function which minimizes the functional using this fact. Notice that this method obtains the solution to the continuous problem. This method is used extensively by Wahba [97, 100, 101] who posed several surface reconstruction problems in meteorology as functional minimizations of the quadratic variation (4.49). This method, however, becomes very computationally expensive when the number of constraints increases.

4.7.2 An Approximation to the Explicit Invariant Stablizer

If the first-order derivatives of z are approximated (i.e. z_x and z_y) then a closer approximation to the invariant stablizer (4.32) can be made. Define the new stablizer as

$$\Omega_3[z] = \iint_U \left\{ 2[A(x,y)z_{xx} - B(x,y)z_{xy} + C(x,y)z_{yy}]^2 - D(x,y)(z_{xx}z_{yy} - z_{xy}^2) \right\} dxdy \quad (4.51)$$

If the approximations to z_x and z_y are denoted by $\hat{z}_x$ and $\hat{z}_y$, then the constant functions A, B, C and D are given by

$$A(x,y) = \left[\frac{1+\hat{z}_y^2}{(1+\hat{z}_x^2+\hat{z}_y^2)^{5/4}} \right]^2 \qquad B(x,y) = \left[\frac{2\hat{z}_x\hat{z}_y}{(1+\hat{z}_x^2+\hat{z}_y^2)^{5/4}} \right]^2$$

$$C(x,y) = \left[\frac{1+\hat{z}_x^2}{(1+\hat{z}_x^2+\hat{z}_y^2)^{5/4}} \right]^2 \qquad D(x,y) = \frac{1}{(1+\hat{z}_x^2+\hat{z}_y^2)^{3/2}} \quad (4.52)$$

Using this stablizer (4.51) it can be shown that the problem is well-posed and that the

functional is convex. Any of the methods used to minimize the functional with the quadratic variation stablizer (4.49) can be modified to form the a discrete functional minimization problem with the added weight terms.

To form an estimate of the first-order derivatives, Blake and Zisserman [11, 12] suggest first fitting an invariant weak membrane to the constraints. Since the constraints are obtained only at discrete points this forms a piecewise planar approximation to the surface. However, the functional minimization problem that arises from fitting the weak membrane to the constraints is nonconvex. For the examples given in Section 4.8 the piecewise planar surface approximation from the first stage of the proposed algorithm is used to form estimates of z_x and z_y.

As will be demonstrated in the examples, the surface estimated using Ω_2 as the stablizer is much more robust to variations in the viewpoint than the surface estimated using the quadratic variation, Ω_1. However, the inaccuracy in estimating z_x and z_y when either (or both) are large causes variation in the surface estimates wherever z_x or z_y is large. This will also be demonstrated in the examples.

4.8 Examples and Analysis

In this section several examples are given which demonstrate the effectiveness of the proposed algorithm in reconstructing viewpoint invariant surfaces. For the noise corrupted constraints in Figure 4.2, the surface estimates reconstructed using the quadratic variation stablizer (Section 4.7.2) and the stablizer which approximates the explicit stablizer (Section 4.7.3) are shown in Figures 4.9 and 4.10, respectively.

Since it is hard to demonstrate the invariance property with this constraint set, the synthetic surface in Figure 4.1 was resampled on a uniform grid. The constraint set for the following examples is displayed in Figure 4.11. For each of the discussed stablizers a surface estimate was first reconstructed with the constraint set in Figure 4.11. The constraint set in Figure 4.11 was then rotated around the $x-$axes by 30 degrees (the orientation and rotation are shown in Figure 4.12) to form a new collection of constraints. A new surface estimate was then formed for the rotated constraints. So that the surfaces could be compared visually this surface was then rotated back 30

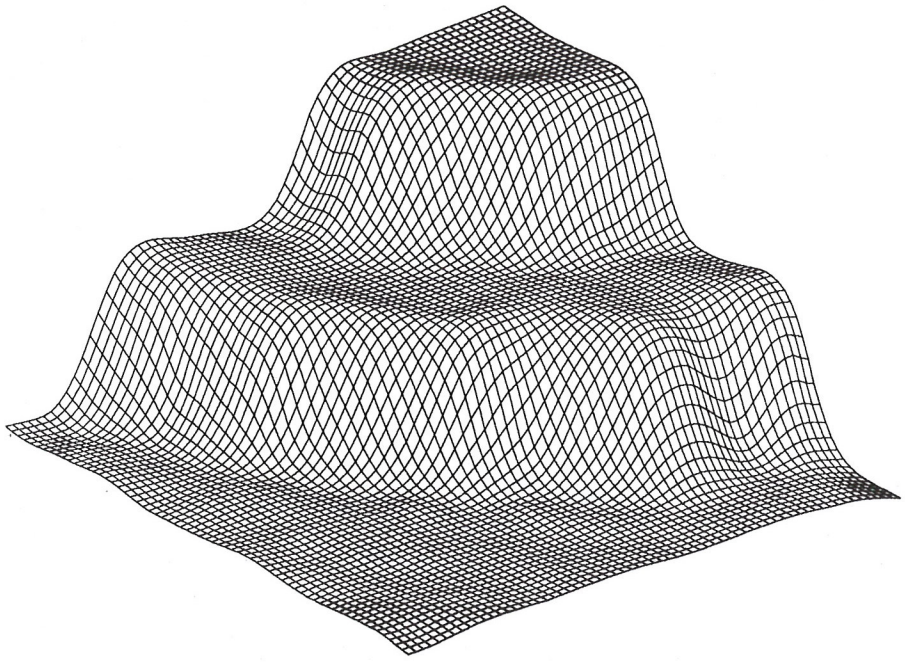

Figure 4.9. Surface reconstructed for noisy constraints with a quadratic variation stablizer, Ω_2, (4.49)

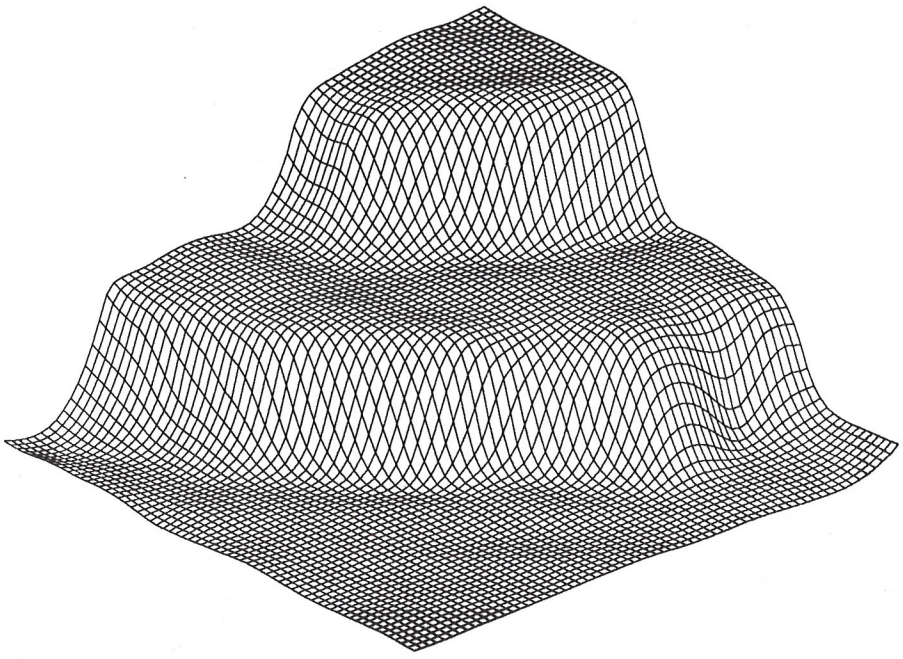

Figure 4.10. Surface reconstructed for noisy constraints using the approximation to the explicit invariant stablizer, Ω_3, (4.51)

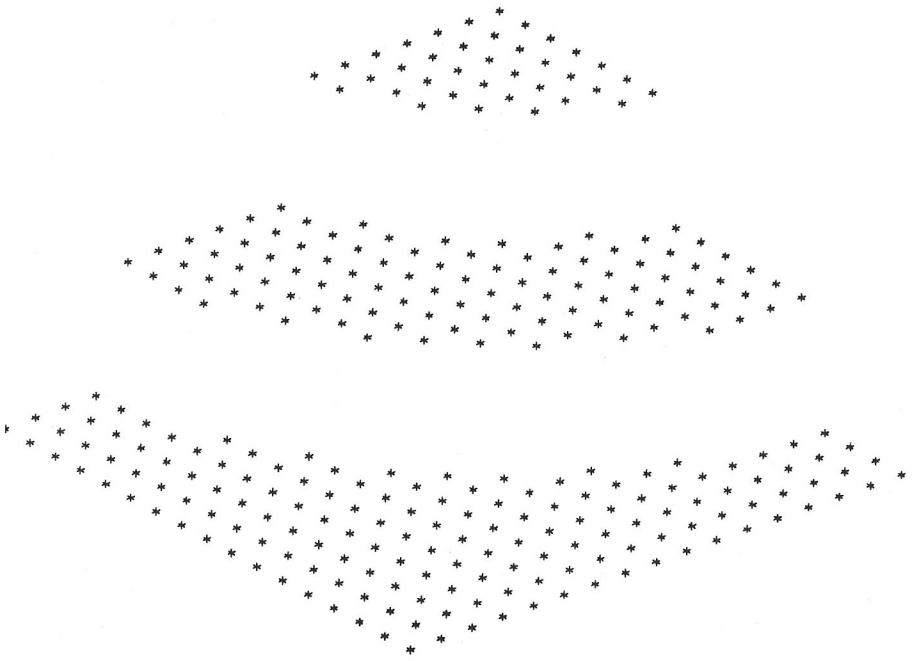

Figure 4.11. Uniformly sampled constraint set

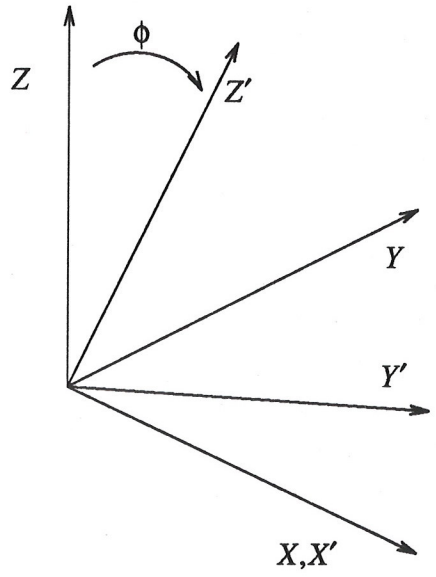

Figure 4.12. Original (X, Y, Z) and rotated (X', Y', Z') axes

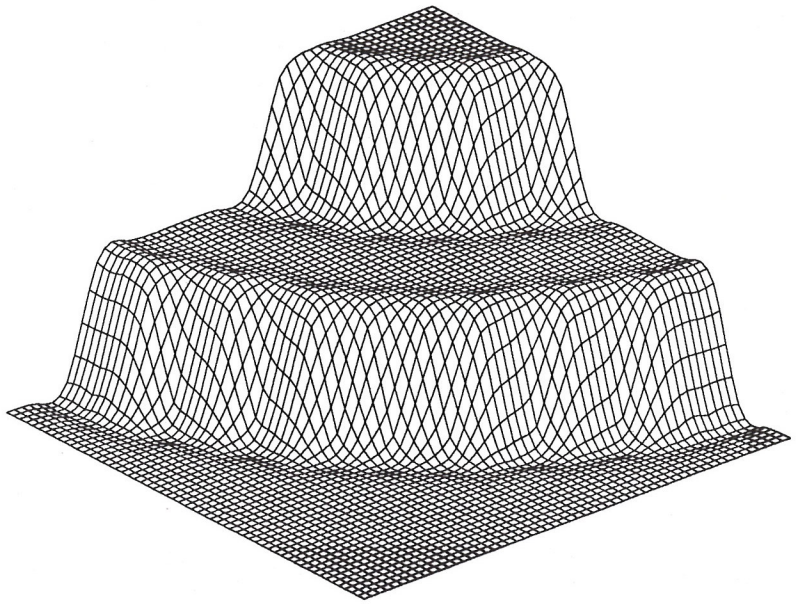

Figure 4.13. Surface reconstructed for original constraints with a quadratic variation stablizer, Ω_2, (4.49)

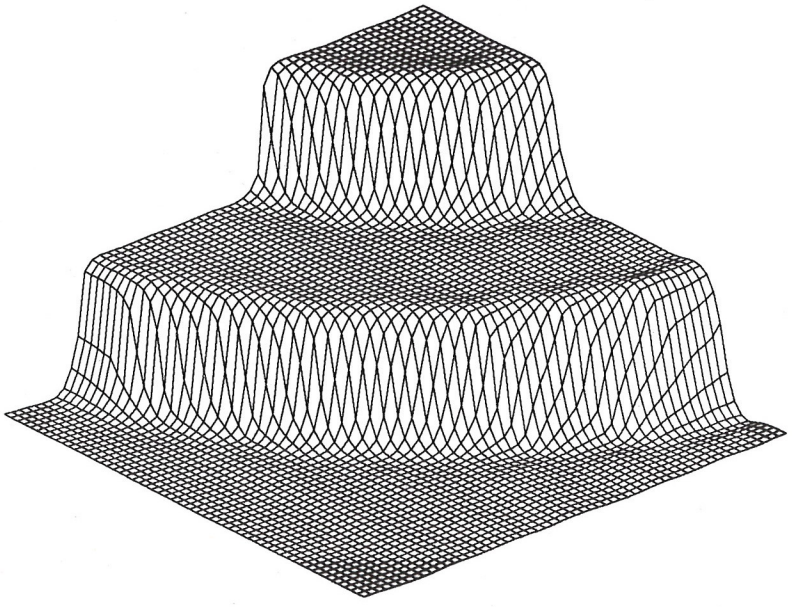

Figure 4.14. Surface reconstructed for rotated constraints with a quadratic variation stablizer, Ω_2, (4.49)

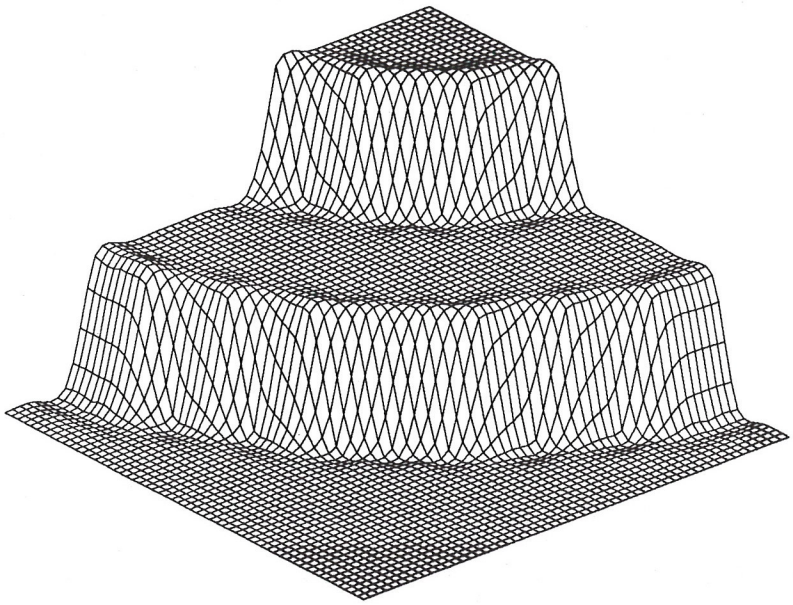

Figure 4.15. Surface reconstructed for original constraints using the approximation to the explicit invariant stablizer, Ω_3, (4.51)

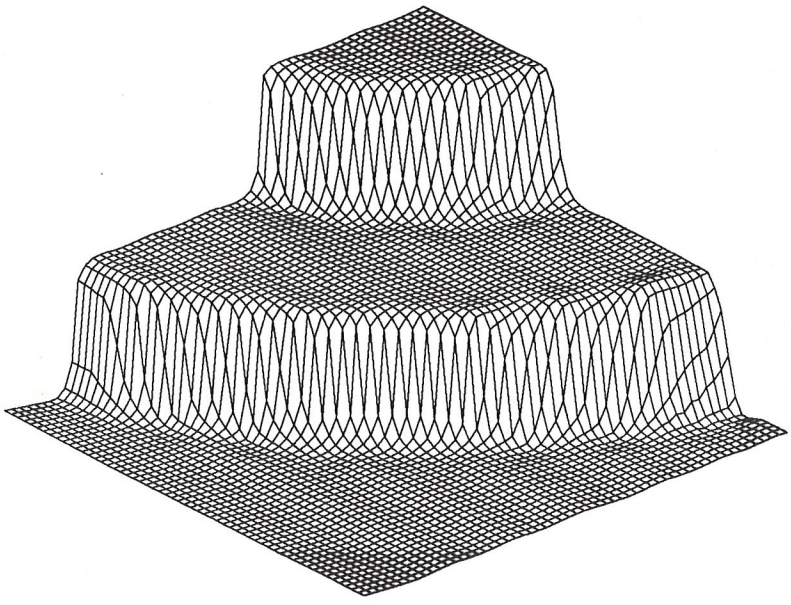

Figure 4.16. Surface reconstructed for rotated constraints using the approximation to the explicit invariant stablizer, Ω_3, (4.51)

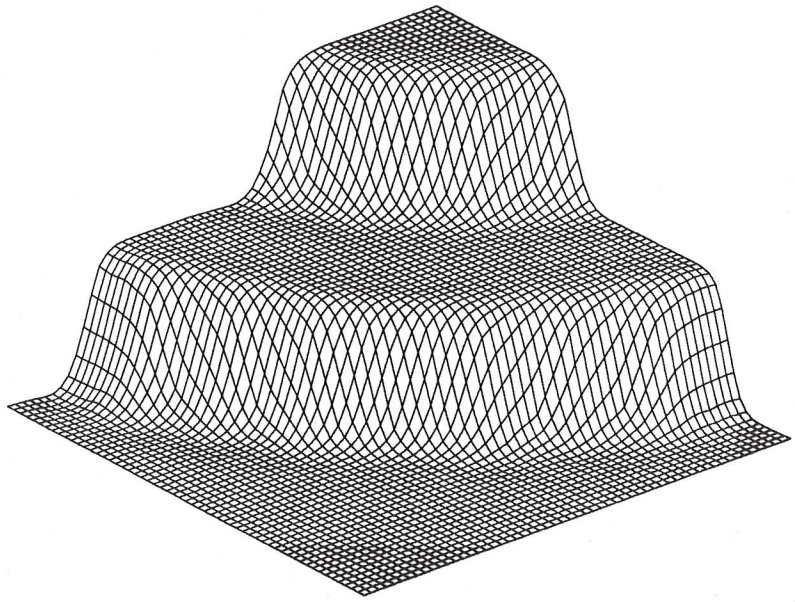

Figure 4.17. Surface reconstructed for original constraints with the proposed invariant algorithm, Ω_1, (4.44)

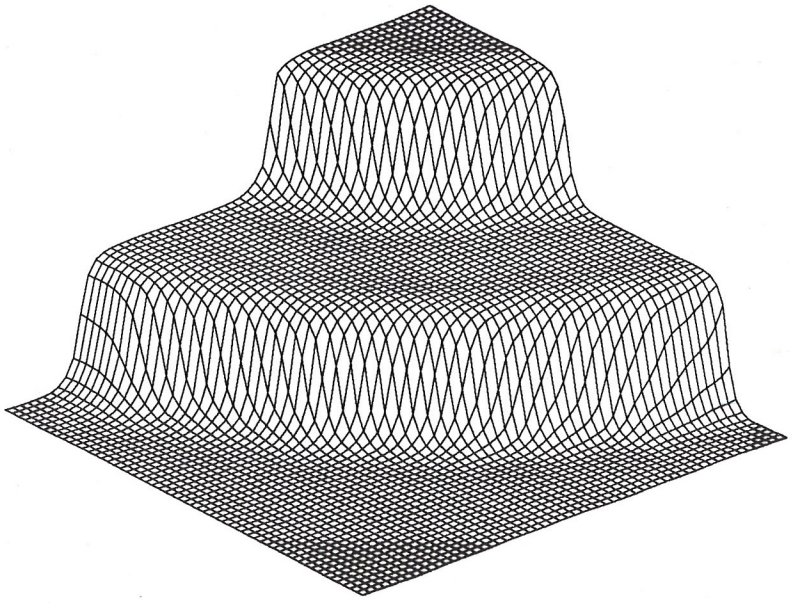

Figure 4.18. Surface reconstructed for rotated constraints with the proposed invariant algorithm, Ω_1, (4.44)

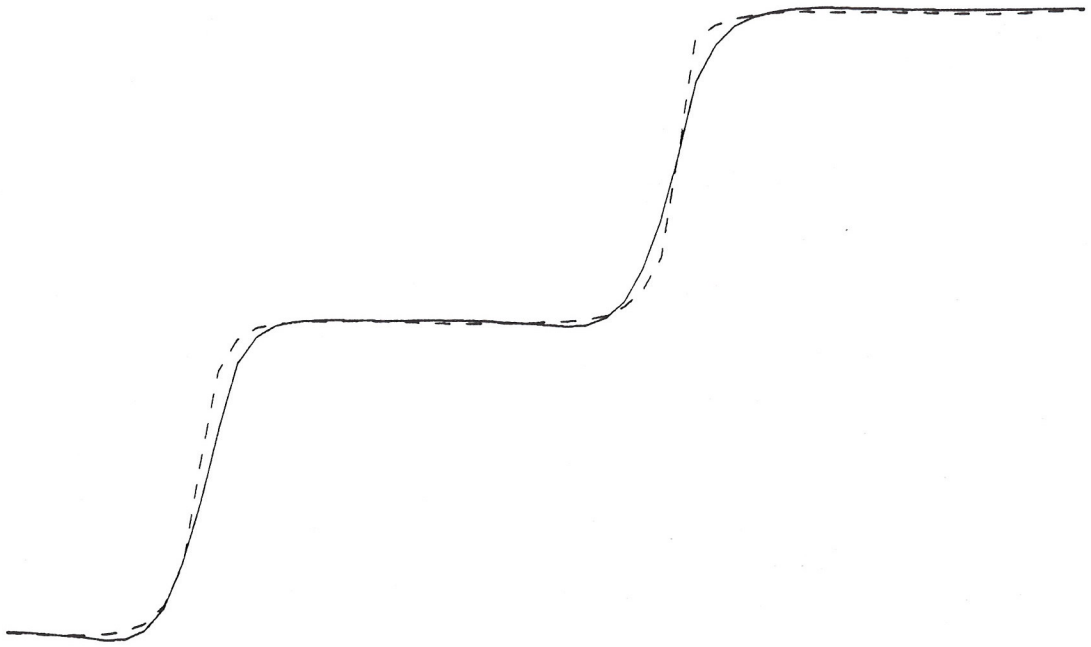

Figure 4.19. Slices of reconstructed surface for Ω_1, (4.44)

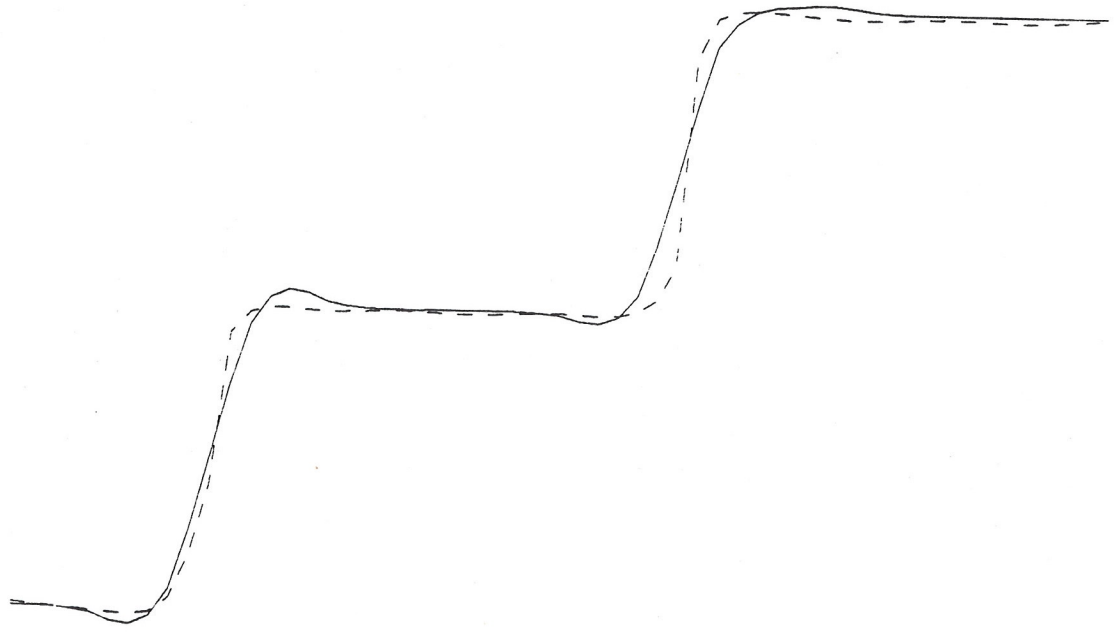

Figure 4.20. Slices of reconstructed surface for Ω_2, (4.49)

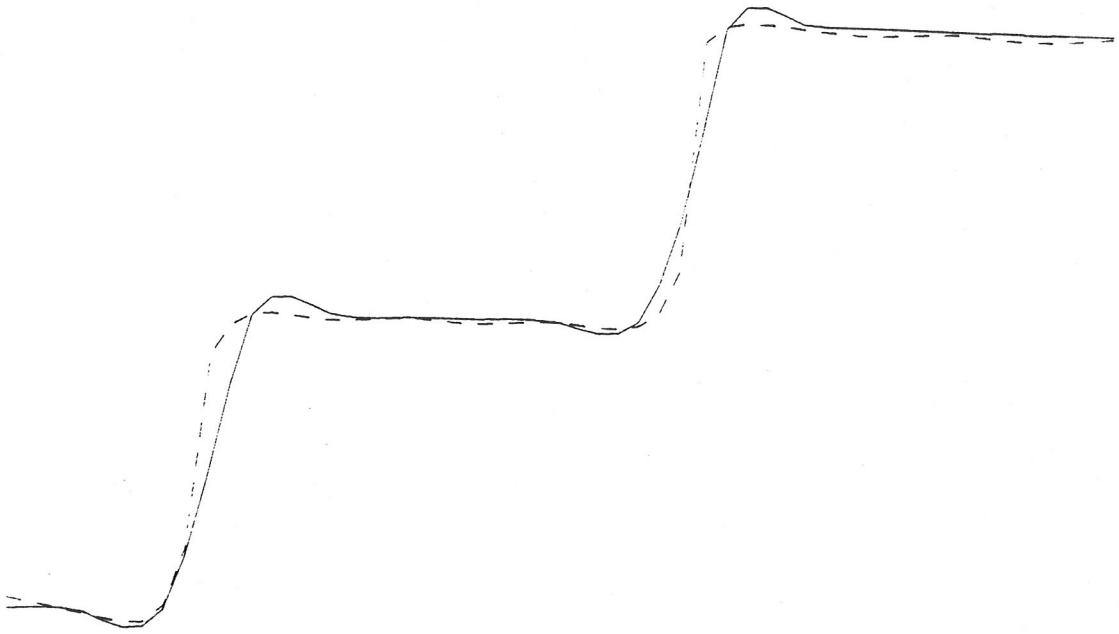

Figure 4.21. Slices of reconstructed surface for Ω_3, (4.51)

degrees around the x–axes, then resampled onto a uniform grid and plotted.

The surface reconstructed using the non-invariant stablizer, (4.49), is shown in Figure 4.13 for the original constraint set and in Figure 4.14 for the rotated constraint set. Notice that also the shape of the surfaces are very different. The reconstructed surface in Figure 4.13 has more of an over- and under-shoot where the surface height changes than the surface in Figure 4.14. The surface reconstructed using Ω_3, (4.51), is shown in Figure 4.15 for the original constraint set and in Figure 4.16 for the rotated constraint set. Once again there is noticeable difference between the two reconstructions. For the original constraints the surface orientation appears to be discontinuous at the location the surface height changes. This surface characteristic does not appear in the reconstruction for the rotated constraints.

Figures 4.17 and 4.18 are the surfaces reconstructed with the proposed reconstruction algorithm for the original and the rotated constraints. Notice that the reconstructed surfaces have similar shapes even near the point that the surface height changes.

Since it is difficult to compare the three-dimensional plots to determine invariant properties, a "slice" of the reconstructed surfaces for the original and rotated data sets were extracted and plotted together for each stablizer. In Figures 4.19, 4.20, and 4.21 the extracted slices are plotted for stablizer Ω_1 (4.44), Ω_2 (4.49), and Ω_3 (4.51), respectively. The solid line is a slice of the surface reconstructed with the original constraints and the dashed line is a slice of the surface reconstructed with the rotated constraints.

To compare the quality of the reconstructions, similar quality measures to the ones computed for the curve reconstruction problem are tabulated in Tables 4.1 and 4.2.

Table 4.1. Value of M^{λ} ($\lambda = 0.01$)

Data Set	Functional Minimized		
	M_1^{λ}	M_2^{λ}	M_3^{λ}
Original	1.123	2.385	2.340
Rotated	1.048	2.205	2.301

Table 4.2. Distance between reconstructions

Metric	Functional Minimized		
	M_1^{λ}	M_2^{λ}	M_3^{λ}
L^2	14.408	24.377	20.276
L^{∞}	1.167	2.597	2.997

Notice that the proposed algorithm performs substantially better by both measures of quality.

4.9 Application in Computer Vision: Visual Surface Reconstruction

For humans, retinal images provide sufficient information for the complete understanding of three-dimensional shapes in a scene. Humans generate complex representations of what they see quite easily, however, a satisfactory computational theory of human vision is still an area of active research. The fundamental goal of current early vision research has been the synthesis of a completely invariant shape description of a scene from two-dimensional intensity images. By extracting information about visible surfaces and their contours in scenes object shape information is obtained. This information includes location and orientation as well as surface characteristics such as texture and color or intensity. An approach proposed by Marr [60] is to form two intermediate representations from which important information is more easily inferred about the scene. These intermediate representations are known as the primal sketch and the full 2 1/2-dimensional sketch, see Figure 4.22.

The initial data set consists of the raw image data obtained directly from the scene by photometric sensors. Several modules act in parallel on this data to extract various features of visual surfaces. The type of information made explicit in this stage depends on the application and on the image characteristics that are most pertinent and easily obtained. To obtain this information, many techniques have been developed to extract depth, orientation, and boundary information from intensity images. Various shape-from-"X" techniques are capable of estimating visible surfaces depth and/or orientation at sparse locations in the scene in a viewer oriented coordinate frame. These techniques include shape from stereopsis [59, 64], shape from shading [41, 43, 47], shape from texture [25, 50, 105], etc. Marr [60] denotes the data obtained by these various techniques as the primal sketch, or the raw 2 1/2-dimensional sketch, of the observed scene.

The next step is to integrate and interpolate the information obtained at sparse locations from various sources into one unified representation. This representation makes explicit the visible surface depth and orientation at all points in the scene. This representation corresponds to Marr's [60] total or full 2 1/2-dimensional sketch, and to Barrow and Tenenbaum's intrinsic images [4]. These range maps, as they are

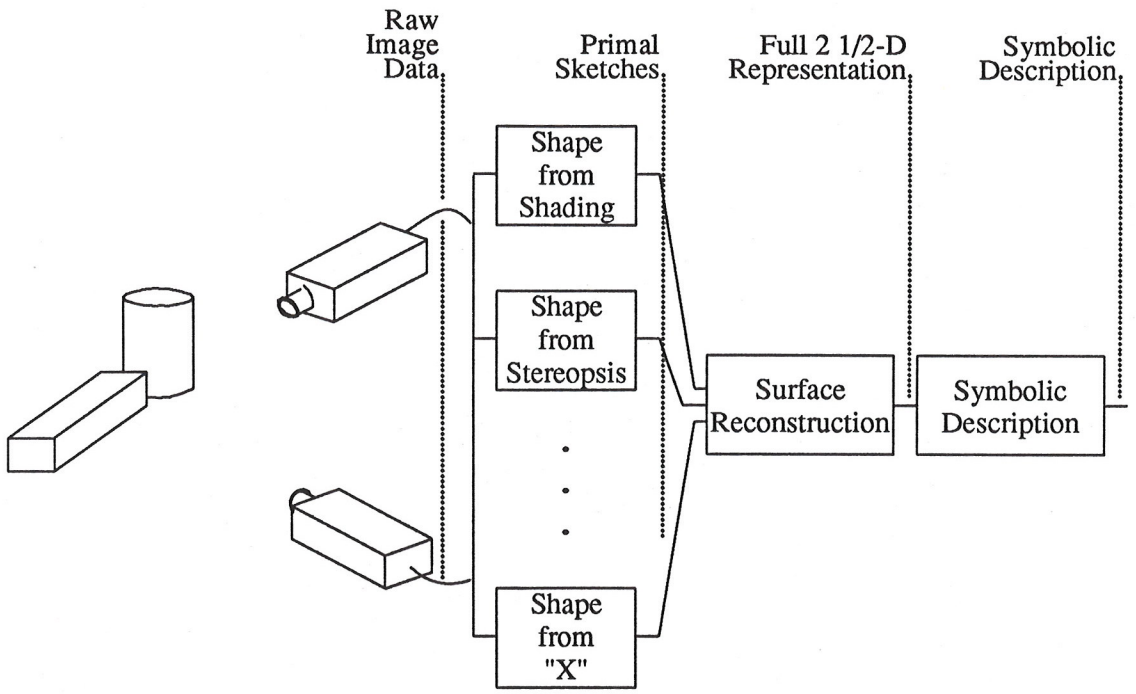

Figure 4.22. Multi-stage approach to three-dimensional scene representation

sometimes know as, is next transformed into a symbolic 3-dimensional surface, or volume representation in an object oriented coordinate frame. This representation, which is the goal in the early vision problem, is then used by higher level recognition or understanding processes. It is desired that this representation be very general and applicable to arbitrarily complex scenes; practical considerations, however, have led to many application specific representations. The techniques used and the representations formed generally depend upon what the researcher views as the universe of possible objects in a scene, e.g. see [7, 10, 16, 17, 27, 29, 32, 37, 38, 40, 42, 66, 95, 96]. Some techniques bypass the second stage where the full 2 1/2-dimensional sketch is generated and form symbolic representations based on the raw 2 1/2-dimensional sketches [69].

As an example of the type of surfaces that are reconstructed in this type of work, a typical collection of sparse three-dimensional data is shown in Figure 4.23. This constraint data was produced by a Technical Arts 100X scanner (White scanner) at the Michigan State University's Pattern Recognition and Image Processing Lab. The

reconstruction of the surface with the proposed algorithm is shown in Figure 4.24 as an intensity image (a range image) and in Figure 4.25 as a three-dimensional plot. In the image the intensity is linearly related to the estimated dept at a particular point.

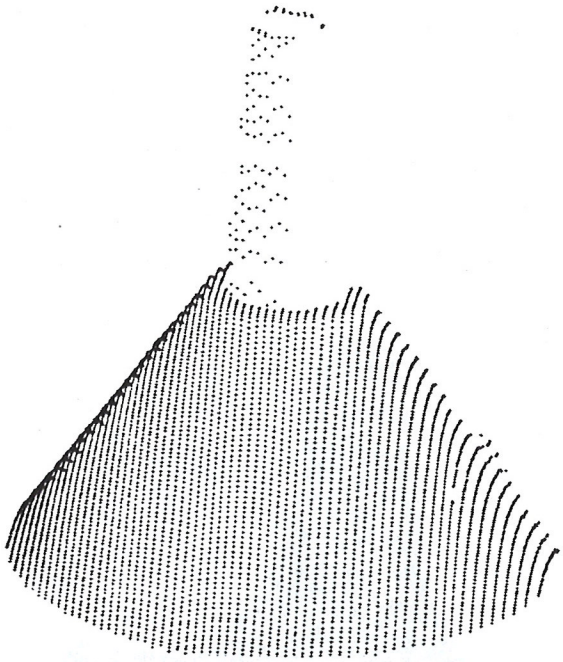

Figure 4.23. Original constraints

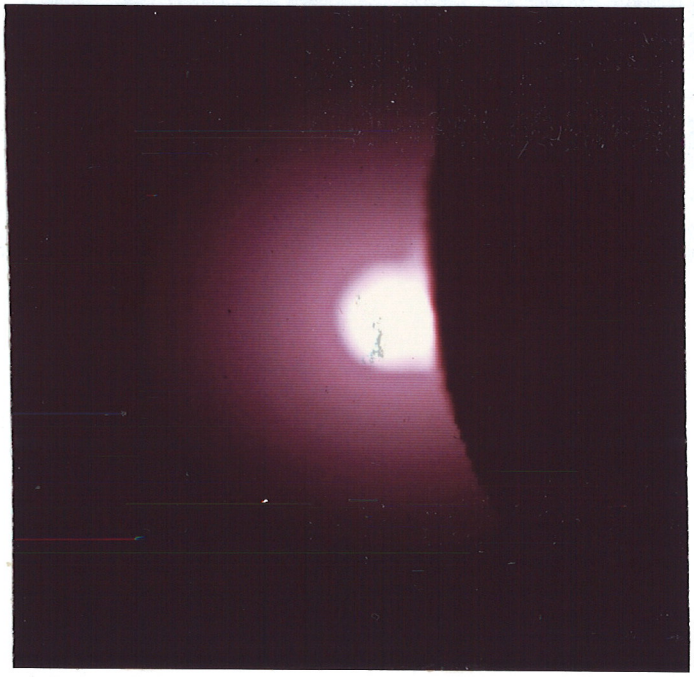

Figure 4.24. Reconstructed intensity range image

$$M^{\lambda}[z(x),S] = \int_U z''^2(x)\,dx + \sum_{c_i \in S} (z(x_i) - c_i)^2, \tag{5.1}$$

which leads to the discrete problem

$$M_d^{\lambda}[\mathbf{z},S] = \mathbf{z}^T(\mathbf{Q} + \lambda \mathbf{I}_S)\mathbf{z} - \mathbf{c}^T\mathbf{z}, \tag{5.2}$$

where $\mathbf{Q}$ is given by (3.70). For the iterative minimization algorithm it is important to understand how to compute $\mathbf{Qz}$. When $\mathbf{Q}$ has a simple form as in (3.70) this is relatively straight forward. In more complex cases it is easier to understand how to compute $\mathbf{Qz}$ by providing the computational molecules [85]. Molecules consist of atoms, indicated by circles, arranged on the spatial lattice and contain coefficients of the associated nodal variable. Figure 5.1 illustrates the three computational molecules for the basic cubic spline approximation problem. A double circle indicates the i^{th} node central to the nodal equation.

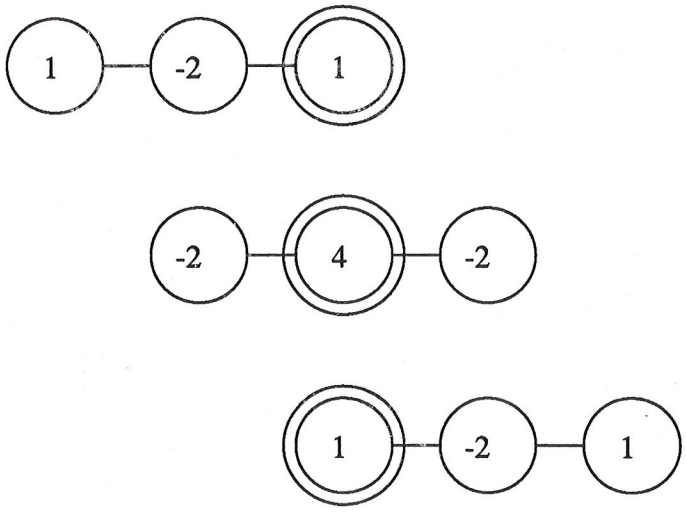

Figure 5.1. Computational molecules for cubic splines

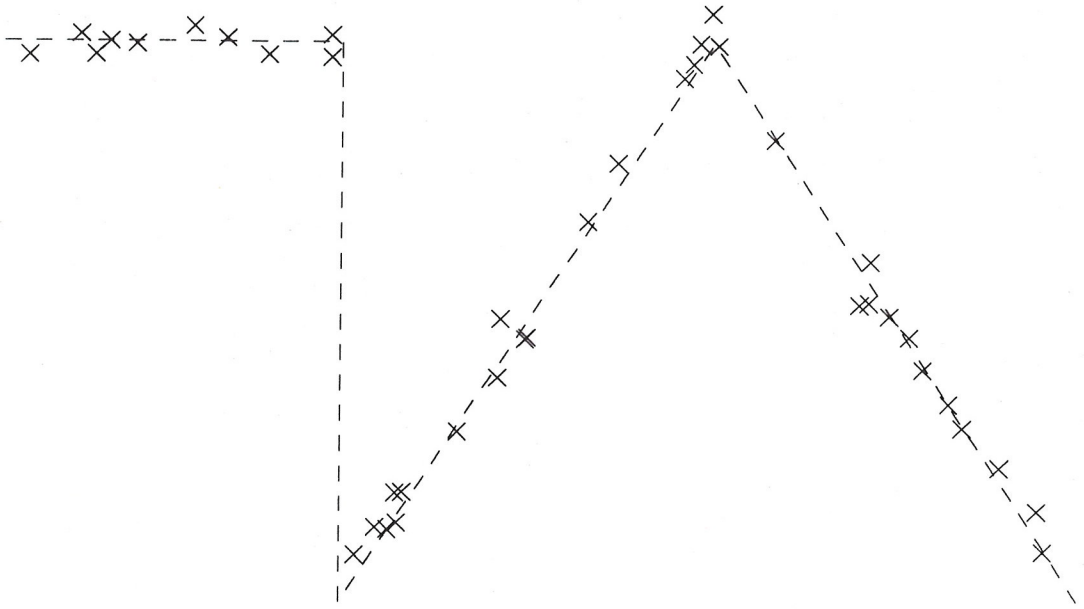

Figure 5.3. Original curve and noisy data

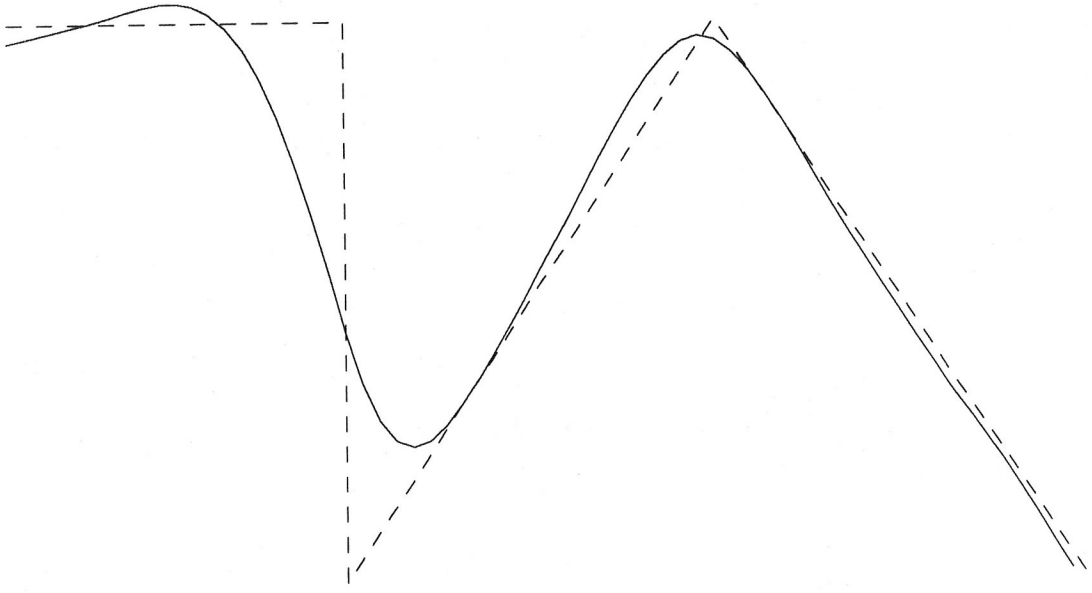

Figure 5.4. Cubic spline reconstruction (λ=1.0)

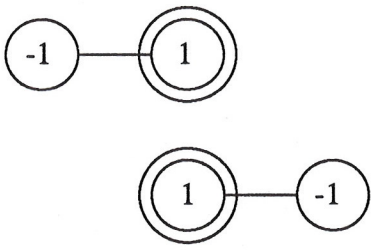

Figure 5.6. Computational molecules for first-order term

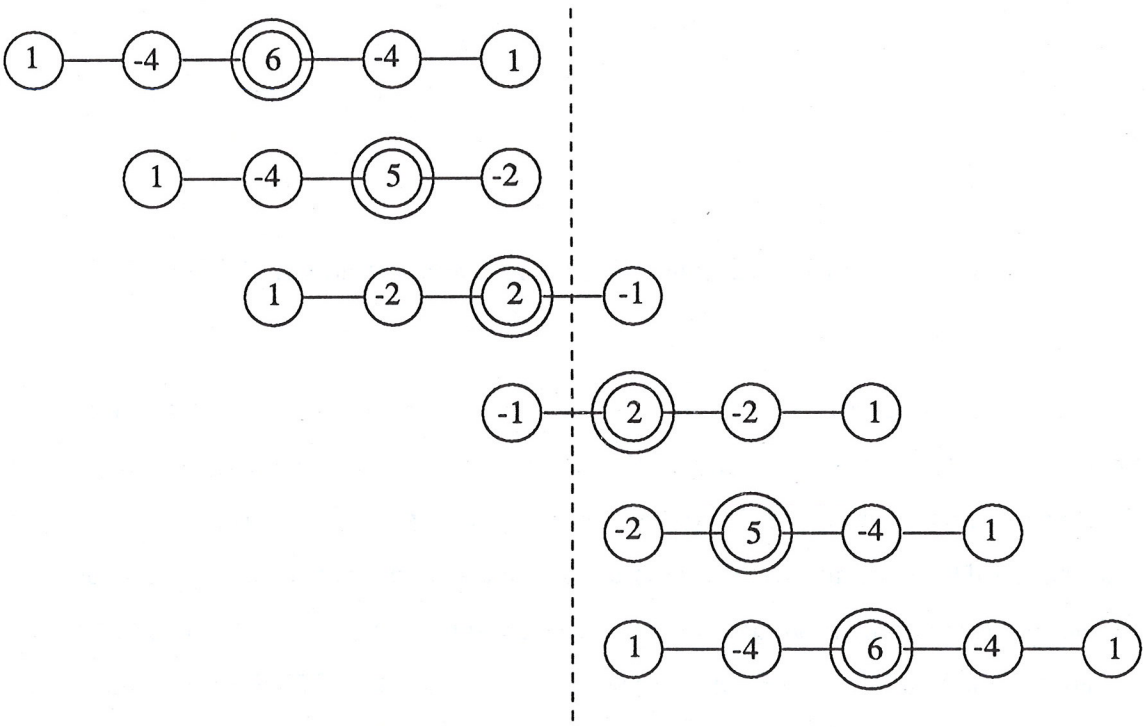

Figure 5.7. Computational molecules near orientation discontinuity

problem with this idea is that the preprocessing amounts to performing an edge detection step on sparse data before reconstructing. Edge detection, however, is greatly effected by noise, causing this step to be very prone to errors, and therefore degrading

consistency measure. That is, too much importance is placed on locations where there is a discontinuity. To overcome this a weight term is added where the weights are chosen such that discontinuities regions are not emphasized as much (i.e. make $w_1(x)$ small in discontinuous regions). The problem, as pointed out before, with adding the weight term is that this term has to be fixed before the minimization is performed. That is, discontinuous regions must be determined before the reconstruction. Another method for decreasing the weight on discontinuous regions is to change the form of the stablizer to

$$\Omega[z(x)] = \int \sigma[z''(x)]\, dx \tag{5.7}$$

where instead of squaring the second derivative of the curve, some yet unspecified function, $\sigma(\cdot)$, is used. This new function should be symmetric since the sign of $z''(x)$ should not change its importance. It should also be convex so that the functional minimization problem remains convex and it should weight large values of $z''(x)$ less than the square function. The desirable properties of $\sigma(\cdot)$ can be summarized by:

1. $\sigma(x) = \sigma(-x)$, $\qquad x \in I\!R$,
2. $\sigma[\alpha x+(1-\alpha)y] \leq \alpha\sigma(x) + (1-\alpha)\sigma(y)$, $\qquad x,y,\alpha \in I\!R$,
3. $\sigma(x) < x^2$, $\qquad x \geq$ some constant T.

One idea for $\sigma(\cdot)$ is to set some threshold T for which the weight on $z''(x)$ remains constant for values above the threshold, i.e. define

$$\sigma_1(x) = \begin{cases} x^2, & x \leq T, \\ T^2, & x > T. \end{cases} \tag{5.8}$$

This idea, although not derived in this fashion, was used by Blake and Zisserman [11, 12]. Unfortunely, since $\sigma_1(\cdot)$ is nonconvex the resulting functional minimization problem is nonconvex. They propose a deterministic algorithm for solving this minimization problem, however like other techniques that minimize nonconvex functionals, the computation is expensive.

The function proposed here is to use a $\sigma(\cdot)$ of the form

$$\sigma_2(x) = \begin{cases} x^2, & x \le T, \\ T^2+2T\,|x-T\,|, & x > T. \end{cases} \tag{5.9}$$

This function varies as a square for values below the threshold and it varies linearity above the threshold. For comparison, the three discussed σ functions are shown in Figure 5.9. Notice that σ_1 meets conditions 1 and 3 of the desired properties, but does not meet condition 2. The proposed σ meets all of the conditions, and therefore is investigated further in the next section.

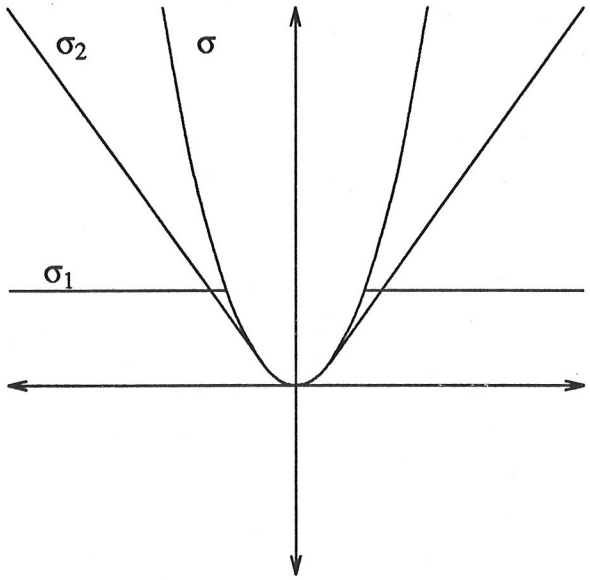

Figure 5.9. Possible σ functions

5.3 Reconstruction with Discontinuities

In this section the reconstruction of curves with discontinuities is examined using

$$\Omega[z(x)] = \int \sigma_2[z''(x)]\,dx \tag{5.10}$$

as the stablizing functional. The functional is discretized and a steepest descent algorithm is used to form the minimization. The discrete form of the functional minimization problem is

$$M_d^{\lambda}[\mathbf{z},\mathcal{S}] = \sum_{i\in U} \sigma(z_{i+1}-2z_i+z_{i-1}) + \lambda \sum_{c_i\in \mathcal{S}} (z_i - c_i)^2. \tag{5.11}$$

Notice that this functional is not quadratic, thus the updating functional must be derived. The steepest descent direction is given by

$$g_i = \frac{\partial M_d^{\lambda}[\mathbf{z},\mathcal{S}]}{\partial z_i} = \psi_2(z_i-2z_{i-1}+z_{i-2}) - 2\psi_2(z_{i+1}-2z_i+z_{i+1})$$

$$+ \psi_2(z_{i+2}-2z_{i+1}+z_i) + \lambda(z_i - c_i)I_{\mathcal{S}}(c_i), \tag{5.12}$$

where $\psi_2(\cdot)$ is the first derivative of σ_2 and $I_{\mathcal{S}}(\cdot)$ is the indicator function. ψ is plotted in Figure 5.10. The update is then computed by

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \beta^{(k)}\mathbf{g}^{(k)}. \tag{5.13}$$

To find the optimal $\beta^{(k)}$ compute the β which satisfies

$$\frac{\partial M_d^{\lambda}[\mathbf{z}+\beta\mathbf{g},\mathcal{S}]}{\partial \beta} = 0. \tag{5.14}$$

This leads to solving the nonlinear equation

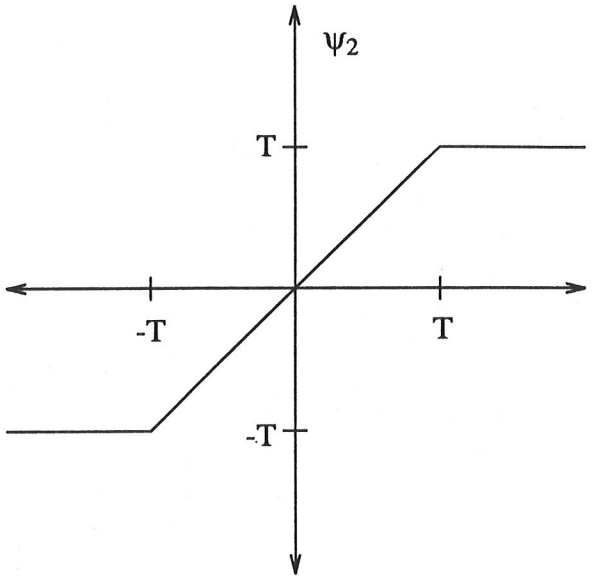

Figure 5.10. Derivative of σ_2, ψ_2

$$\sum_{i\in U} \psi(z_{i+1}-2z_i+z_{i+1}+\beta g_{i+1}-2\beta g_i+\beta g_{i-1})(g_{i+1}-2g_i+g_{i-1})$$

$$+\lambda \sum_{c_i\in S} (z_i+\beta g_i-c_i)g_i = 0. \qquad (5.15)$$

There are techniques to solve this nonlinear equation, however, since the curve is initially approximated with a piecewise linear curve the approximation

$$\psi(z_{i+1}-2z_i+z_{i+1}+\beta g_{i+1}-2\beta g_i+\beta g_{i-1}) \approx \psi(z_{i+1}-2z_i+z_{i+1}) \qquad (5.16)$$

is valid. Therefore, the optimal β is approximated by

$$\beta = \frac{\sum_{i\in U} \psi(z_{i+1}-2z_i+z_{i+1})(g_{i+1}-2g_i+g_{i-1}) + \lambda \sum_{c_i\in S} (z_i-c_i)g_i}{\lambda \sum_{c_i\in S} g_i^2}. \qquad (5.17)$$

The computational molecules for this new stablizer are given in Figure 5.11 and the curve reconstructed using this stablizer for several values of T is shown in Figure 5.12.

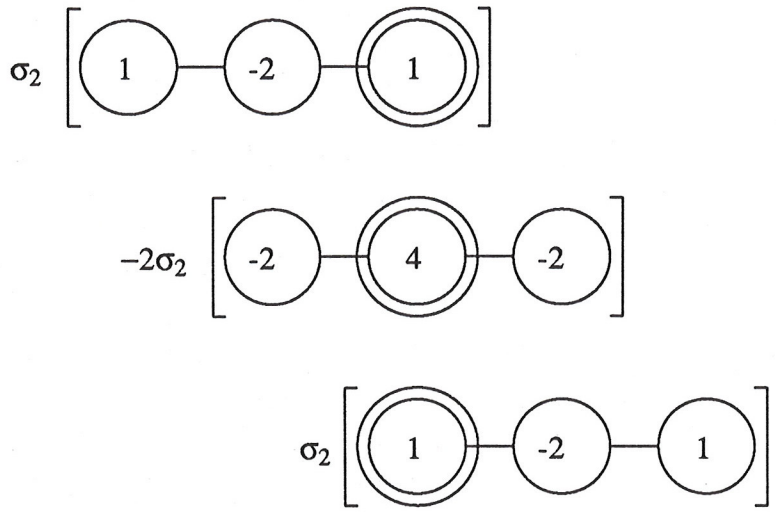

Figure 5.11. Computational molecules for proposed stablizer

5.4 Invariant Reconstruction with Discontinuities

Extending the ideas presented in the previous section to form invariant reconstructions is very straight forward. Section 5.4.1 and 5.4.2 discuss the changes needed to reconstruct invariant curves and surfaces with discontinuities, respectively.

5.4.1 Invariant reconstruction of curves with discontinuities

For invariant curve reconstruction instead of the second derivative being used as the measure of smoothness, the invariant quantity κ is used. Therefore, an invariant stablizer which will allow discontinuities in the reconstruction can be defined by

$$\Omega[\mathbf{r}(u)] = \int_U \sigma(\kappa(u))\, ds. \tag{5.18}$$

To approximate this nonconvex functional with a convex functional either of the approximations discussed in Sections 3.5.2 and 3.5.3 can be used for $\kappa(u)$. If the

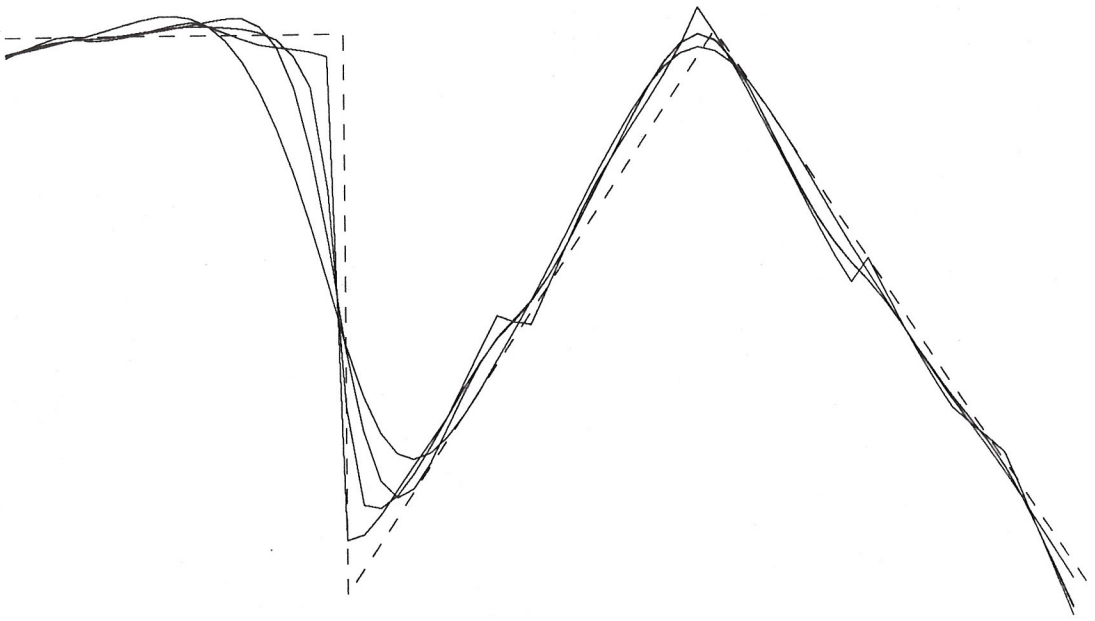

Figure 5.12. Reconstruction using the proposed stablizer (λ=1.0, T=0.1, 1.0, 10.0, ∞)

approximation in Section 3.5.3 is used then the approximate convex stablizer becomes

$$\Omega[\mathbf{r}(u)] = \int_U \sigma(\|\mathbf{r}''(u)\|)du. \tag{5.19}$$

The curve reconstructed using this stablizer in shown in Figure 5.13.

5.4.2 Invariant reconstruction of surfaces with discontinuities

For invariant surface reconstruction the second order invariant quantities of κ_1 and κ_2 were used as the measure of smoothness. An invariant stablizer which incorporates discontinuities can be defined by

$$\Omega[\mathbf{r}(\mathbf{u})] = \iint_U \sigma[\kappa_1(\mathbf{u}] + \sigma[\kappa_2(\mathbf{u})]\, dA. \tag{5.20}$$

Using the same techniques as discussed in Section 4.5 this nonconvex invariant stablizer can be approximated with the convex stablizer

$$\Omega[\mathbf{r}(\mathbf{u})] = \iint_U \sigma[\omega_{11} z_{xx} + \omega_{12} z_{xy} + \omega_{13} z_{yy}]$$

$$+ \sigma[\omega_{21} z_{xx} + \omega_{22} z_{xy} + \omega_{23} z_{yy}] dxdy. \qquad (5.21)$$

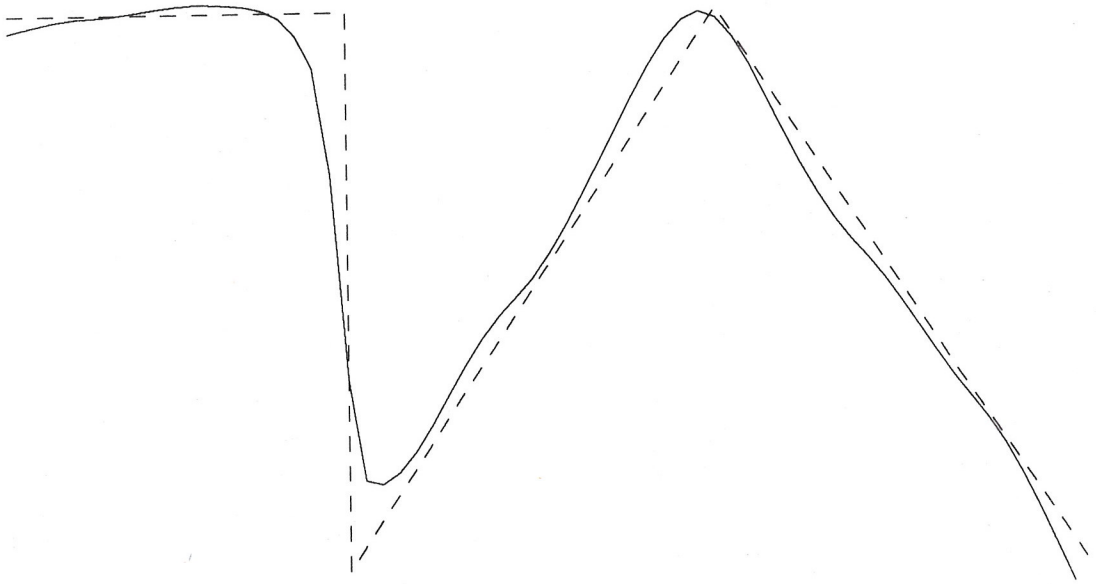

Figure 5.13. Invariant curve reconstruction with discontinuities (λ=1.0, T=1.0)

CHAPTER 6 - CONCLUSIONS

6.1 Summary of Research

The goal of this research was the development of an algorithm(s) for the reconstruction of curves and surface which would be applicable to problems in computer vision. The applications in computer vision requires some very interesting restrictions on the characteristics of the reconstruction algorithms. First, since the coordinate system is arbitrarily chosen the reconstructions should be invariant to the choice of a particular coordinate system. Second, since the objects being represented may have discontinuities, the reconstruction algorithm should produce results which will retain this information. Finally, since it is usually desired to solve computer vision problems in real-time, the proposed reconstruction algorithm should be computationally efficient.

To form a well-posed mathematical problem statement, the method of Tikhonov regularization was utilized. To obtain the invariance property the reconstruction algorithm was based on invariant quantities from differential geometry. In Chapter 3, the invariant reconstruction of curves was examined and two new algorithms were proposed for the invariant reconstruction of curves. The first algorithm was applicable to only curves in the plane, but the second algorithm could be used for the reconstruction of curves in any dimensional space. The invariant reconstruction of surfaces was examined in Chapter 4. Invariant characteristics of surfaces were discussed and based on these an invariant reconstruction algorithm was proposed.

The inclusion of discontinuities into the reconstruction process was discussed in Chapter 5. It was observed that discontinuities could be included by changing the processing near regions of discontinuities. This can be accomplished by decreasing the effect of high curvature regions (i.e. discontinuities) in the consistency measure.

Previously proposed techniques required either that the importance of each region be computed by a preprocessing step or that the computational complexity of the reconstruction algorithm to be increased significantly. In Chapter 5 a new concept was introduced for decreasing the effect of discontinuous regions without fixing the regions before reconstruction and without significantly increasing the computational complexity.

Throughout this research examples have been provided to help validate the hypothesises of the proposed algorithms. These experimental results verify the usefulness of the proposed algorithms.

6.2 Open Research Problems

There are several application dependent problems which may arise in a particular computer vision problems. The following is a brief list of the type of problems that can occur, and for which answers need to be sought.

1. The proposed reconstruction technique is a least squares approach to fitting the data. It is well known that least squares techniques are only optimal when the noise that corrupts that data is i.i.d. Gaussian. This is often not true in computer vision applications (e.g. constraint data from a stereo algorithm [13]). In some applications it may become important to modify the the functional to obtain an optimal solution.

2. Many algorithms in computer vision which produce the constraints that are used as inputs to our algorithm, will form relatively reliable data with the exception of a few outliers. For these problems it will become important to build into the algorithms a robustness to outliers in the constraint set. One possible idea for accomplishing this goal was proposed by Huber [45, 46].

3. In the reconstruction algorithm there are two parameters λ and T which control the tradeoff between the smoothness and the closeness of fit and between the smoothness and the discontinuities. These parameters are user selected. In this work the parameters were selected empirically. It would however be

advantageous if these parameters can be derived from the data set. There have be several ideas proposed [97-99], in some sense to select "optimal" parameters for the set of data. However, the computation required to determine these parameters is prohibitive in computer vision applications. In fact, there is more computation required to select these parameters then there is to reconstruct the object. It has been our experience that the reconstructions are relatively insensitive to the selection of these parameters and an adequate solution is to perform some experiments with a collection of representative data set and hardwire these parameters. If however a quick method can be devised for the selection of the parameters, these algorithms would benefit.

4. The invariant stablizers (3.27) and (4.30) are not unique. There are many possible way in which to combine the invariant characteristics of curves and surfaces to form an invariant consistency measure. For some applications it may be beneficial to examine the the other possibilities.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Abramowitz, M. and I. A. Stegun, *Handbook of Mathematical Functions,* New York: Dover Publications, Inc., 1965.

2. Adams, R. A., *Sobolev Spaces,* New York: Academic Press, 1975.

3. Aloimonos, J. and D. Shulman, "Learning early-vision computations," *Journal of the Optical Society of America A*, vol. 6, no. 6, pp. 908-919, June 1989.

4. Barrow, H. G. and J. M. Tenenbaum, "Recovering intrinsic scene characteristics from images," in *Computer Vision Systems*, A. Hanson and E. Riseman, eds., New York, NY: Academic, pp. 76-86, 1978.

5. Bernoulli, D., "The 26th letter to Euler," in *Correspondence Mathematique et Physique*, Fuss, ed., St. Petersburg, 1843.

6. Besl, P. J. and R. C. Jain, *Surface characterization for three-dimensional object recognition in depth maps,*Tech. Rep. RSD-TR-20-84, Ann Arbor, MI: University of Michigan Center for Research on Integrated Manufacturing, December 1984.

7. Besl, P. J. and R. C. Jain, "Three-dimensional object recognition," *ACM Computing Surveys*, vol. 17, no. 1, pp. 75-145, March 1985.

8. Besl, P. J. and R. C. Jain, "Segmentation through symbolic surface descriptions," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, Miami, FL, June 22-26, 1986, pp. 77-85.

9. Besl, P. J. and R. C. Jain, "Invariant surface characteristics for 3D object recognition in range data," *Computer Vision, Graphics and Image Processing*, vol. 33, no. 1, pp. 33-80, 1986.

10. Besl, P. J. and A. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 2, pp. 167-192, March 1988.

11. Blake, A. and A. Zisserman, "Invariant surface reconstruction using weak continuity constraints," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, Miami, FL, June 22-26, 1986, pp. 62-67.

12. Blake, A. and A. Zisserman, *Visual reconstruction,* Cambridge, Massachusetts: MIT Press, 1987.

13. Blostein, S. D. and T. S. Huang, "Error analysis in stereo determination of 3-D point positions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 6, pp. 752-765, November 1987.

14. Boult, T., *Smoothness Assumptions in Human and Machine Vision: Their Implications for Optimal Surface Interpolation,* New York, NY: Computer Science Department, 1985.

15. Boult, T. E. and J. R. Kender, "Visual surface reconstruction using sparse depth data," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, Miami, FL, June 22-26, 1986, pp. 68-76.

16. Brice, C. R. and C. L. Fennema, "Scene analysis using regions," *Artificial Intelligence*, vol. 1, pp. 205-226, 1970.

17. Brooks, R. A., "Representing possible realities for vision and manipulation," in *Proceedings of the Pattern Recognition and Image Processing Conference*, Las Vegas, NV, June 14-17, 1982, pp. 587-592.

18. Cavendish, J. C., "Automatic triangulation of arbitrary planar domains for the finite element method," *International Journal for Numerical Methods in Engineering*, vol. 8, pp. 697-696, 1974.

19. Chu, C. and A. Bovik, "Visible surface reconstruction via local minimax approximation," *Pattern Recognition*, vol. 21, no. 4, pp. 303-312, 1988.

20. Chu, C. H., E. J. Delp and A. J. Buda, "Detecting left ventricular endocardial and epicardial boundaries by digital two-dimensional echocardiography," *IEEE Transactions on Medical Imaging*, vol. 7, no. 2, pp. 81-90, June 1988.

21. Ciarlet, P. G., *The Finite Element Method for Elliptic Problems,* Amsterdam: North-Holland Publishing Company, 1978.

22. Cline, A. K. and R. L. Renka, "A storage-efficient method for construction of a Thiessen Triangulation," *Rocky Mountain Journal of Mathematics*, vol. 14, no. 1, pp. 119-139, Winter 1984.

23. Courant, R. and D. Hilbert, *Methods of Mathematical Physics,* New York: Interscience Publishers, Inc., vol. I, 1953.

24. Davis, J. C., *Interpolation and Approximation,* New York, N.Y.: Blaisdell, 1963.

25. Davis, L. S., L. Janos and S. M. Dunn, "Efficient recovery of shape from texture," *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 5, pp. 485-492, September 1983.

26. Duchon, J., "Splines minimizing rotation-invariant semi-norms in Sobolev spaces," in *Constructive Theory of Functions of Several Variables*, 1976, pp. 85-100.

27. Duda, R. O., D. Nitzan and P. Barrett, "Use of range and reflectance data to find planar surface regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 3, pp. 259-271, July 1979.

28. Eaton, L. W., W. L. Maughan, A. A. Shoukas and J. L. Weiss, "Accurate volume determination in the isolated ejecting canine left ventricle by two-dimensional echocardiography," *Circulation*, vol. 60, pp. 320-326, August 1979.

29. Fan, T. J., G. Medioni and R. Nevatia, "Description of surfaces from range data using curvature properties," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, Miami, FL, June 22-26, 1986, pp. 86-91.

30. Faux, I. D. and M. J. Pratt, *Computational Geometry for Design and Manufacture,* Chichester, England: Ellis Horwood Limited, 1979.

31. Ferrie, F. P., J. Lagarde and P. Whaite, "Darboux frames, snakes, and super-quadrics: geometry from the bottom-up," in *Proceedings of the Workshop on Interpretation of 3D Scenes*, Austin, Texas, November 27-19, 1989, pp. 170-176.

32. Gil, B., A. Mitiche and J. K. Aggarwal, "Experiments in combining intensity and range edge maps," *Computer Vision, and Image Processing*, vol. 21, pp. 395-411, 1983.

33. Grimson, W. E. L., *From images to surfaces: a computational study of the human early visual system,* Cambridge, MA: MIT Press, 1981.

34. Guggenheimer, H. W., *Differential Geometry,* New York: Dover Publications, Inc., 1977.

35. W. Hackbusch and U. Trottenberg, eds., *Multigrid Methods,* Berlin, West Germany: Springer-Verlag, 1980.

36. Hadamard, J., *Lectures on the Cauchy Problem in Linear Partial Differential Equations,* New Haven, CT: Yale University Press, 1923.

37. Henderson, T. C., "Efficient 3-D object representation for industrial vision systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 6, pp. 609-618, November 1983.

38. Herman, M., "Generating detailed scene descriptions from range images," in *Proceedings of the IEEE International Conference on Robotics and Automation*, St. Louis, MO, March 25-28, 1985, pp. 426-431.

39. Hestenes, M. R. and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, December 1952.

40. Hoffman, R. and A. K. Jain, "Segmentation and classification of range images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 608-620, September 1987.

41. Horn, B. K. P., "Obtaining shape from shading information," in *The Psychology of Computer Vision*, P. H. Winston, ed., New York, NY: McGraw-Hill, 1975.

42. Horn, B. K. P., "Extended Gaussian images," *Proceedings of the IEEE*, vol. 72, no. 12, pp. 1671-1686, December 1984.

43. Horn, B. K. P. and M. J. Brooks, "The variational approach to shape from shading," *Computer Vision, Graphics and Image Processing*, vol. 33, no. 2, pp. 174-208, 1986.

44. Huang, T. S. and R. Y. Tsai, "Image sequence analysis: motion estimation," in *Image Sequence Analysis*, T. S. Huang, ed., New York: Springer-Verlag, pp. 1-48, 1981.

45. Huber, P. J., "Robust smoothing," in *Robustness in Statistics*, R. L. Launer and G. N. Wilkinson, eds., New York: Academic Press, pp. 33-47, 1979.

46. Huber, P. J., *Robust Statistics,* New York: John Wiley & Sons, 1981.

47. Ikeuchi, K. and B. K. P. Horn, "Numerical shape from shading and occluding boundaries," *Artificial Intelligence*, vol. 17, pp. 141-185, 1981.

48. Jou, J. and A. C. Bovik, "Improving visible-surface reconstruction," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Ann Arbor, MI, June 5-9, 1988, pp. 138-143.

49. Jou, J. and A. C. Bovik, "Improved initial approximation and intensity-guided discontinuity detection in visible-surface reconstruction," *Computer Vision, Graphics and Image Processing*, vol. 47, no. 3, pp. 292-325, September 1989.

50. Kender, J. R., *Shape from texture,* Pittsburgh, PA: Department of Computer Science, Carnegie-Mellon University, 1980.

51. Koskol, J. B. and J. P. Allebach, "Estimating the motion of a maneuvering target with time-sequentially sampled imagery," *Journal of the Optical Society of America A*, vol. 3, pp. 1504-1511, September 1986.

52. Lancaster, P. and K. Salkauskas, *Curve and Surface Fitting,* London: Academic Press, 1986.

53. Lawson, C., *Generation of a triangular grid with applications to contour plotting,* Technical Report T.M. 299, Jet Propulsion Labortary, 1972.

54. Lee, D., "Coping with discontinuities in computer vision: their (detection, classification), and measurement," in *Proceedings of the Second International Conference on Computer Vision*, Tampa, Florida, December 5-8, 1988, pp. 546-557.

55. Lee, D. and T. Pavlidis, "One-dimensional regularization with discontinuities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 822-829, November 1988.

56. Limb, J. O. and J. A. Murphy, "Estimating the velocity of moving images in television signals," *Computer Graphics and Image Processing*, vol. 4, pp. 311-327, 1975.

57. Lipschutz, M. M., *Differential Geometry,* New York, NY: McGraw-Hill, 1969.

58. Malcolm, M. A., "On the computation of nonlinear spline functions," *SIAM Journal of Numerical Analysis*, vol. 14, pp. 254-282, 1977.

59. Marr, D. and T. Poggio, "A theory of human stereo vision," *Proceedings of the Royal Society of London B*, vol. 204, pp. 301-328, 1979.

60. Marr, D., *Vision,* San Francisco, CA: W. H. Freeman and Company, 1982.

61. Marroquin, J. L., *Probabilistic Solution of Inverse Problems,* Cambridge, MA: Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, September 1985.

62. Marroquin, J., S. Mitter and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 76-89, March 1987.

63. Marti, J. T., *Introduction to Sobolev Spaces and Finite Element Solution of Elliptic Boundary Value Problems,* London: Academic Press, 1986.

64. Mayhew, J. E. W. and J. P. Frisby, "Psychophysical and computational studies towards a theory of human stereopsis," *Artificial Intelligence (Special Issue on Computer Vision*, vol. 17, 1981.

65. Meinguet, J., "Multivariate interpolation at arbitrary points made simple," *Journal of Applied Mathematics and Physics*, vol. 30, pp. 292-304, 1979.

66. Oshima, M. and Y. Shirai, "Object recognition using three-dimensional information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 4, pp. 353-361, July 1983.

67. Osserman, R., *A Survey of Minimal Surfaces,* New York: Dover Publications, Inc., 1986.

68. Parent, P. and S. W. Zucker, "Curvature consistency and curve detection," *Journal of the Optical Society of America A*, vol. 2, no. 13, pp. 5, 1985.

69. Rao, K. and R. Nevatia, "From sparse 3-D data directly to volumetric shape descriptions," in *Proceedings of the DARPA Image Understanding Workshop*, February 1987, pp. 360-396.

70. Reinsch, C. H., "Smoothing by Spline Functions," *Numerische Mathematik*, vol. 10, pp. 177-183, 1967.

71. Renka, R. J., *A storage-efficient method for construction of a Thiessen triangulation,* Technical Report ORNL/CSD-101, Oak Ridge National Laboratory, 1982.

72. Salkauskas, K., "C1 splines for interpolation of rapidly varying data," *Rocky Mountain Journal of Mathematics*, vol. 14, no. 1, pp. 239-250, Winter 1984.

73. Sander, P. T., *On Reliably Inferring Differential Structure from Three-Dimensional Images,* Quebec, Canada: Ph.D. Thesis, Department of Electrical Engineering, McGill University, 1988.

74. Sander, P. T. and S. W. Zucker, "Inferring differential structure from three-dimensional images: smooth cross sections of fibre bundles," in *(to appear) IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990.

75. Schumaker, L. L., "Fitting surfaces to scattered data," in *Approximation Theory II*, New York, 1976, pp. 203-268.

76. Shiau, J. H., *Smoothing Spline Estimation of Functions with Discontinuities,* PhD Thesis, University of Wisconsin-Madison, 1985.

77. Sinha, S. S. and B. G. Schunck, "Discontinuity preserving surface reconstruction," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, California, June 4-8, 1989, pp. 229-234.

78. Strang, G. and G. J. Fix, *An analysis of the finite element method,* Englewood Cliffs, NJ: Prentice-Hall, 1973.

79. Terzopoulos, D., "The role of constraints and discontinuities in visible-surface reconstruction," in *Proceedings 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, August 8-12, 1983, pp. 1073-1077.

80. Terzopoulos, D., "Multilevel computational processes for visual surface reconstruction," *Computer Vision, Graphics, and Image Processing*, vol. 24, no. 1, pp. 52-96, October 1983.

81. Terzopoulos, D., *Multiresolution computation of visible surface representation,* Cambridge, MA: Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, January 1984.

82. Terzopoulos, D., "Multilevel reconstruction of visual surfaces: variational principles and finite-element representations," in *Multiresolution Image Processing and Analysis*, A. Rosenfeld, ed., New York, NY: Springer-Verlag, pp. 237-310, 1984.

83. Terzopoulos, D., "Image analysis using multigrid relaxation methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 2, pp. 129-139, March 1986.

84. Terzopoulos, D., "Regularization of inverse visual problems involving discontinuities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 4, pp. 413-424, July 1986.

85. Terzopoulos, D., "The computation of visible-surface representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-10, no. 4, pp. 417-438, July 1988.

86. Thompson, W. B. and S. T. Barnard, "Lower-level estimation and interpretation of visual motion," *IEEE Transactions on Computers*, vol. V-14, pp. 20-28, 1981.

87. Tikhonov, A. N., "The solution of ill-posed problems," *Doklady Akademii Nauk SSSR*, vol. 151, no. 3, 1963.

88. Tikhonov, A. N., "Regularization of ill-posed problems," *Doklady Akademii Nauk SSSR*, vol. 153, no. 1, 1963.

89. Tikhonov, A. N., "Solution of nonlinear integral equations," *Doklady Akademii Nauk SSSR*, vol. 156, no. 6, 1964.

90. Tikhonov, A. N., "Stable methods of summing Fourier series," *Doklady Akademii Nauk SSSR*, vol. 156, no. 1, 1964.

91. Tikhonov, A. N. and V. B. Glasko, "Approximate solution of Fredholm integral equations of the first kind," *Zhurnal vychislitel' noy matematiki i matematicheskoy fiziki*, vol. 4, no. 3, 1964.

92. Tikhonov, A. N., "Nonlinear equations of the first kind," *Doklady Akademii Nauk SSSR*, vol. 161, no. 5, 1965.

93. Tikhonov, A. N. and V. Y. Arsenin, *Solutions of Ill-Posed Problems,* Washington, D. C.: V. H. Winston & Sons, 1977.

94. A. N. Tikhonov and A. V. Goncharsky, eds., *Ill-Posed Problems in the Natural Sciences,* Moscow: MIR Publishers, 1987.

95. Vemuri, B. C., A. Mitiche and J. K. Aggarwal, "Curvature-based representation of objects from range data," *Image and Vision Computing*, vol. 4, no. 2, pp. 107-114, May 1986.

96. Vemuri, B. C. and J. K. Aggarwal, "Representation and recognition of objects from dense range maps," *IEEE Transactions on Circuits and Systems*, vol. CAS-34, no. 11, pp. 1351-1363, November 1987.

97. Wahba, G., "Spline bases, regularization, and generalized cross validation for solving approximation problems with large quantities of noisy data," in *Approximation Theory III*, W. Cheney, ed., Academic Press, pp. 905-912, 1980.

98. Wahba, G., *Cross validation and constrained regularization methods for mildly ill posed problems,* Technical Report 629, Madison, WI: Department of Statistics, University of Wisconsin, February 1980.

99. Wahba, G., "Surface fitting with scattered noisy data on Euclidean d-space and on the sphere," *Rocky Mountain Journal of Mathematics*, vol. 14, no. 1, pp. 281-299, Winter 1984.

100. Wahba, G., "Variational methods for multidimensional inverse problems," in *Advances in Remote Sensing Retrieval Methods*, 1985, pp. 385-407.

101. Wahba, G., "Three topics in ill-posed problems," in *Inverse and Ill-Posed Problems*, vol. 4, H. W. Engl and C. W. Groetsch, eds., Orlando, FL: Academic Press, pp. 37-51, 1987.

102. Weiss, I., "Shape reconstruction on a varying mesh," in *Proceedings of the DARPA Image Understanding Workshop*, February 1987, pp. 749-765.

103. Weiss, I., "3D shape representation by contours," in *Proceedings of the IEEE Workshop on Computer Vision*, Miami Beach, Florida, November 30 - December 2, 1987, pp. 358-362.

104. Weiss, I., "Curve fitting using a varying mesh," in *Proceedings of the IEEE Workshop on Computer Vision*, Miami Beach, Florida, November 30 - December 2, 1987, pp. 311-314.

105. Witkin, A. P., "Recovering surface shape from orientation and texture," in *Computer Vision*, M. Brady, ed., North-Holland Publishing, 1980.

106. Wolfe, E. R., E. J. Delp, C. R. Meyer, F. L. Bookstein and A. J. Buda, "Accuracy of automatically determined borders in digital two-dimensional echocardiography using a cardiac phantom," *IEEE Transactions on Medical Imaging*, vol. MI-6, no. 4, pp. 292-296, December 1987.

APPENDICES

Appendix A - Conjugate Gradient Method

This appendix will address the solution of the discrete problem defined by

$$\text{minimize} \quad \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{c}^T\mathbf{x} \tag{A.1}$$

where $\mathbf{A}$ is an $n \times n$ symmetric positive definite matrix. Since the function is convex one of the simplest methods for the solution of this problem is using a steepest descent algorithm which will converge to the solution. If $\mathbf{A}$ is banded, a steepest descent algorithm can be implemented on a very efficient parallel architecture. Unfortunely, the convergence of this algorithm will be very slow. The convergence can be accelerated while maintaining the parallelism of the steepest descent type algorithms. The basic idea is to optimize the direction in which each new step is taken. The results in this appendix were obtained from [39].

By taking a derivative with respect to $\mathbf{x}$ and setting equal to zero it can easily be shown that the unique minimization of (A.1) is also the unique solution to the linear equation

$$\mathbf{A}\mathbf{x} = \mathbf{c}. \tag{A.2}$$

Given an $n \times n$ symmetric matrix $\mathbf{A}$, two vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ are said to be $\mathbf{A}$–orthogonal, conjugate, with respect to $\mathbf{A}$ if

$$\mathbf{d}_1^T\mathbf{A}\mathbf{d}_2 = 0. \tag{A.3}$$

If the $\mathbf{A}$ matrix is in addition positive definite, then a set of nonzero $\mathbf{A}$–orthogonal vectors, $\mathbf{d}_0, \mathbf{d}_1, \cdots, \mathbf{d}_{n-1}$ are linearly independent. Since this vectors are linearly independent the solution to the optimization problem can be expanded in terms of them as

$$\mathbf{x}' = \alpha_0\mathbf{d}_0 + \cdots + \alpha_{n-1}\mathbf{d}_{n-1}. \tag{A.4}$$

The orthogonality of the vectors imply that

$$\alpha_i = \frac{\mathbf{d}_i^T \mathbf{A}\mathbf{x}'}{\mathbf{d}_i^T \mathbf{A}\mathbf{d}_i} = \frac{\mathbf{d}_i^T \mathbf{c}}{\mathbf{d}_i^T \mathbf{A}\mathbf{d}_i}, \tag{A.5}$$

thus the unknown solution vector $\mathbf{x}$ can be written as

$$\mathbf{x}' = \sum_{i=0}^{n-1} \frac{\mathbf{d}_i^T \mathbf{c}}{\mathbf{d}_i^T \mathbf{A}\mathbf{d}_i} \mathbf{d}_i. \tag{A.6}$$

The following result can be proven.

Conjugate Direction Theorem: Let $\{\mathbf{d}_i\}_{i=0}^{n-1}$ be a set of non-zero A–orthogonal vectors. For any $\mathbf{x}_0$ the sequence $\{\mathbf{x}_k\}$ generated according to

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{D}_k, \tag{A.7}$$

with

$$\alpha_k = -\frac{\mathbf{g}_k^T \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{A}\mathbf{d}_k} \tag{A.8}$$

and

$$\mathbf{g}_k = \mathbf{A}\mathbf{x}_k - \mathbf{c}, \tag{A.9}$$

converges to the unique solution, $\mathbf{x}'$, after n steps. □

The proof of this theorem can be found in [39].

To generate the sequence of A–orthogonal direction vectors an initial direction is selected and successive direction vectors are computed as a conjugate version of the successive gradients obtained as the method proceeds. The complete algorithm as was used in this study is outlined below.

Conjugate Gradient Algorithm

Starting at any point $\mathbf{x}_0$ define $\mathbf{d}_0 = -\mathbf{g}_0 = \mathbf{c} - \mathbf{A}\mathbf{x}_0$ and compute the following vectors iteratively:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \tag{A.10}$$

$$\alpha_k = -\frac{\mathbf{g}_k^T \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{A} \mathbf{d}_k}, \tag{A.11}$$

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k, \tag{A.12}$$

$$\beta_k = -\frac{\mathbf{g}_{k+1}^T \mathbf{A} \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{A} \mathbf{d}_k}, \tag{A.13}$$

$$\mathbf{g}_k = \mathbf{A}\mathbf{x}_k - \mathbf{c}. \tag{A.14}$$

Once the residual norm $||\mathbf{x}_{k+1} - \mathbf{x}_k||$ becomes small the iteration process is stopped and $\mathbf{x}_k$ is used as the solution.

Appendix B - Construction of a Thiessen Triangulation

This appendix outlines an approach for constructing a triangulation of a set of surface points. The problem can be stated as follows. Given a set $\mathcal{S}$ of nodes $\{(x_i,y_i),i=1,\cdots,M\}$ arbitrarily distributed in the XY plane, construct a triangulation with the nodes as vertices such that the triangles are as equiangular as possible. A node is identified by its index and an arbitrary index will be denoted by one of $N_0, N_1, N_2, \cdots$. The definitions and algorithms presented in this appendix are taken from [22].

Definition B.1. For three points $P_0 = (X_0,Y_0)$, $P_1 = (X_1,Y_1)$, and $P_0 = (X_0,Y_0)$ with $P_1 \neq P_2$, P_0 LEFT $P_1 \rightarrow P_2$ if and only if

$$(X_2-X_1)(Y_0-Y_1) \geq (X_0-X_1)(Y_2-Y_1). \tag{B.1}$$

We also say P_0 STRICTLY LEFT $P_1 \rightarrow P_2$ if the inequality is strict. □

Definition B.2. A region of the plane is convex if and only if for any two points contained in the region the line segment connecting the two points also lies in the region. The convex hull of a finite set of points in the plane is the smallest convex region which contains the points. □

Let $\mathcal{H}$ be the convex hull of $\mathcal{S}$ and let $\mathcal{B}$ be the boundary of $\mathcal{H}$.

Definition B.3. A triangle is the convex hull of three noncollinear nodes, referred to as vertices. The triangle with vertices N_1, N_2, and N_3 is denoted (N_i,N_j,N_k) with N_k STRICTLY LEFT $N_i \rightarrow N_j$ for $(i,j,k) \in \{(1,2,3),(2,3,1),(3,1,2)\}$. □

Definition B.4. A triangulation of $\mathcal{S}$ is a set of triangles $\mathcal{T}$ with the following properties.

1. Each triangle contains no node other than its vertices,
2. the interiors of the triangles are pairwise disjoint, and
3. $\mathcal{T}$ covers $\mathcal{H}$.

□

Definition B.5. Let $\mathcal{T}$ be a triangulation of $\mathcal{S}$. For each triangle (N_1,N_2,N_3) of $\mathcal{T}$, the

vectors $N_1 \rightarrow N_2$, $N_2 \rightarrow N_3$, and $N_3 \rightarrow N_1$ are referred to as edges of $\mathcal{T}$. The undirected line segment corresponding to an edge $N_1 \rightarrow N_2$ is referred to as an arc and is denoted by N_1–N_2. □

Definition B.6. Two nodes are adjacent to each other if and only if they are the endpoints of a common arc. A neighbor of a node N_0 is a node which is adjacent to N_0. □

Definition B.7. A pair of triangles of $\mathcal{T}$, (N_1,N_2,N_3) and (N_2,N_1,N_4), which share a common arc form a quadrilateral of $\mathcal{T}$denoted by (N_1,N_2,N_3,N_4). □

Definition B.8. Given an interior arc N_1–N_2 with corresponding quadrilateral (N_1,N_2,N_3,N_4), the swap test is a decision on whether to perform a swap based on either of the following criteria (which can be shown to be equivalent).

1. The max-min angle criterion is the choice of the pair of triangles which maximizes the minimum of the six interior angles when (N_1,N_2,N_3,N_4) is strictly convex. The decision will be positive if and only if a larger minimum angle would result from the swap. The decision is negative if (N_1,N_2,N_3,N_4) is not strictly convex.

2. The circle criterion is the choice of the pair of triangles whose circumcircles do not contain the remaining vertices in their interiors.

An arc is locally optimal if and only if it is a boundary arc or, if it is interior, application of the swap test to it would not result in a decision to swap. □

Definition B.9. A Thiessen triangulation is one in which all arcs are locally optimal. □

An algorithm to form a Thiessen triangulation from a set of nodes is presented in [22] and is outlined below.

Algorithm to generate a Thiessen triangulation:

1. Presort the nodes by sorting the X components.

2. Construct an initial Thiessen triangulation, $\mathcal{T}_j$, of the first j nodes where j is the smallest integer such that nodes $1, \cdots, j$ are not all collinear.

3. For $k = j+1, \cdots, M$ construct a Thiessen triangulation, $\mathcal{T}_k$ of nodes $1, \cdots, k$ as follows.

   — Connect k to all boundary nodes which are visible from k.

   — Optimize the mesh by applying a sequence of swap tests and the appropriate swaps to the interior arcs which are opposite k.

This algorithm is guaranteed to terminate after a finite number of steps and is guaranteed to generate a Thiessen triangulation of the nodes. See [22] for more details.

Appendix C - Estimating Surface Characteristics

This appendix describes the method used to first form a local estimate of the surface characteristics and then the method used to refine these estimates. The approach used is based on [73,74], with many of the suggestions made by [31] incorporated. The problem can be stated as follows: for a smooth continuous surface, estimate the augmented Darboux frame at a uniformly sampled collection of nodes on the surface only knowing the surface location at the sampled nodes. This is accomplished by first estimating the frame $\mathcal{D}(P)$ at each node using local estimates, see [8,29] for typical approaches. These estimates are then refined using an iterative technique based on minimizing the variation in $\mathcal{D}(P)$ on the surface.

Let $\mathbf{r}(u,v)$ be a parametric representation of the surface. Estimates of the Darboux frame are computed at locations on a grid on the surface. The surface is sampled uniformly on the parameter space, thus it can be assumed without loss of generality that the samples are located at $\mathbf{r}(i,j)$, $i,j = 1,2,3,....$

C.1 Initial Estimates of $\mathcal{D}(P)$

First and second-order estimates at point $\mathbf{r}(i,j)$ are computed using a local neighborhood of known node values. For this work the 3×3 neighborhood of $\mathcal{N}= \{(k,l);\ i-1\le k\le i+1,\ j-1\le l\le j+1\}$ is used to form the local estimates. To form an initial estimate of the surface normal at (i,j), $\mathbf{n}(i,j)$, the best fit (in the least squares sense) plane is computed. That is, the sum

$$\sum_{(k,l)\in\mathcal{N}} \left(\mathbf{n}(i,j),\mathbf{r}(k,l)-\mathbf{r}(i,j)\right)^2 \qquad \text{(C.1)}$$

is minimized with respect to $\mathbf{n}(i,j)$ subject to the condition that $||\mathbf{n}(i,j)|| = 1$. Recall that $(\cdot,\cdot)$ is an inner (dot) product.

Using the estimate of the normal at $\mathbf{r}(i,j)$ a coordinate system can be setup such that $\mathbf{r}(i,j)$ is at the origin and the surface normal is one of the axes, see Figure B.1. A parametric representation in this coordinate system can be found by rotating and

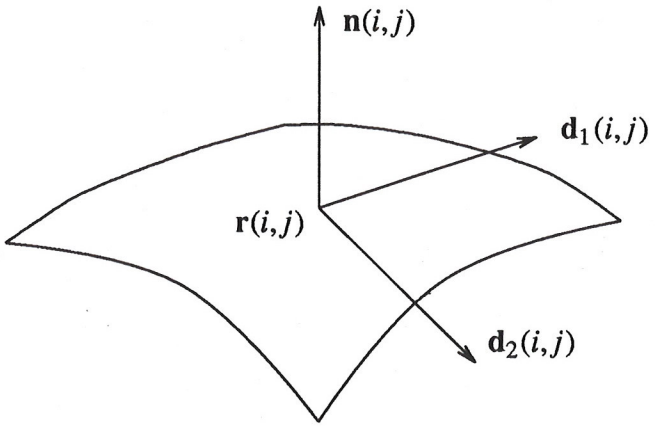

Figure C.1. Local surface representation

translating $\mathbf{r}(u,v)$ appropriately, let this transformation be denoted by $T\mathbf{r}(u,v)$. Using vector-matrix notation for the vectors this affine transformation can be given by [30]

$$T\mathbf{r}(u,v) = \begin{bmatrix} \dfrac{z_{\mathbf{n}}}{x_{\mathbf{n}}^2+z_{\mathbf{n}}^2} & \dfrac{-y_{\mathbf{n}}x_{\mathbf{n}}}{x_{\mathbf{n}}^2+z_{\mathbf{n}}^2} & x_{\mathbf{n}} \\ 0 & x_{\mathbf{n}}^2+z_{\mathbf{n}}^2 & y_{\mathbf{n}} \\ \dfrac{-x_{\mathbf{n}}}{x_{\mathbf{n}}^2+z_{\mathbf{n}}^2} & \dfrac{-y_{\mathbf{n}}z_{\mathbf{n}}}{x_{\mathbf{n}}^2+z_{\mathbf{n}}^2} & z_{\mathbf{n}} \end{bmatrix} \left\{ \mathbf{r}(u,v) - \mathbf{r}(i,j) \right\} \tag{C.2}$$

where $x_{\mathbf{n}}$, $y_{\mathbf{n}}$, and $z_{\mathbf{n}}$ are the components of $\mathbf{n}(i,j)$. In a local neighborhood of the origin in the new coordinate system the surface can be represented by

$$\mathbf{r}(s,t) = <s,t,\frac{1}{2}(as^2+2bst+ct^2)>, \qquad s,t \approx 0. \tag{C.3}$$

Using the information at the transformed neighborhood nodes $\{T\mathbf{r}(k,l),\ (k,l)\in\mathcal{N}\}$ the optimal a, b, and c are computed (in the least squares sense). The Weingarten mapping matrix for this representation is given by

$$\beta = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \tag{C.4}$$

The eigenvalues of β are the principal curvatures

$$\kappa_1 = a + c + \sqrt{(a-c)^2 + 4b^2}, \tag{C.5}$$

$$\kappa_2 = a + c - \sqrt{(a-c)^2 + 4b^2}. \tag{C.6}$$

At non-umbilic points, the corresponding eigenvectors are

$$\mathbf{x}_1 = \begin{cases} \langle a - c + \sqrt{(a-c)^2 + 4b^2}, 2b \rangle, & \text{when } a \geq c, \\ \langle 2b, -a + c + \sqrt{(a-c)^2 + 4b^2} \rangle, & \text{otherwise;} \end{cases} \tag{C.7}$$

$$\mathbf{x}_2 = \begin{cases} \langle -2b, a - c + \sqrt{(a-c)^2 + 4b^2} \rangle, & \text{when } a \geq c, \\ \langle a - c - \sqrt{(a-c)^2 + 4b^2}, 2b \rangle, & \text{otherwise;} \end{cases} \tag{C.8}$$

which after normalization are the principle directions in the tangent plane. By transforming the normalized vectors by T^{-1}, estimates of $\mathbf{d}_1(i,j)$ and $\mathbf{d}_2(i,j)$ are computed.

C.2 Refinement of the $\mathcal{D}(P)$ estimate

The description of the algorithm is taken from [31]. Local consistency of curvature, subject to orthogonality constraints on $\mathcal{D}(P)$, is the basis of the minimization algorithm [68, 74] Since $\mathcal{D}(P)$ is a dual form of a parabolic quadric, one can extrapolate outward from a point Q to its neighbor P to get an idea of what the surface at P looks like according to its neighbor Q, see Figure C.2. By performing this operation for each neighbor of P, one obtains a set of frames, $\xi_{P\alpha}$, each providing an estimate of P from

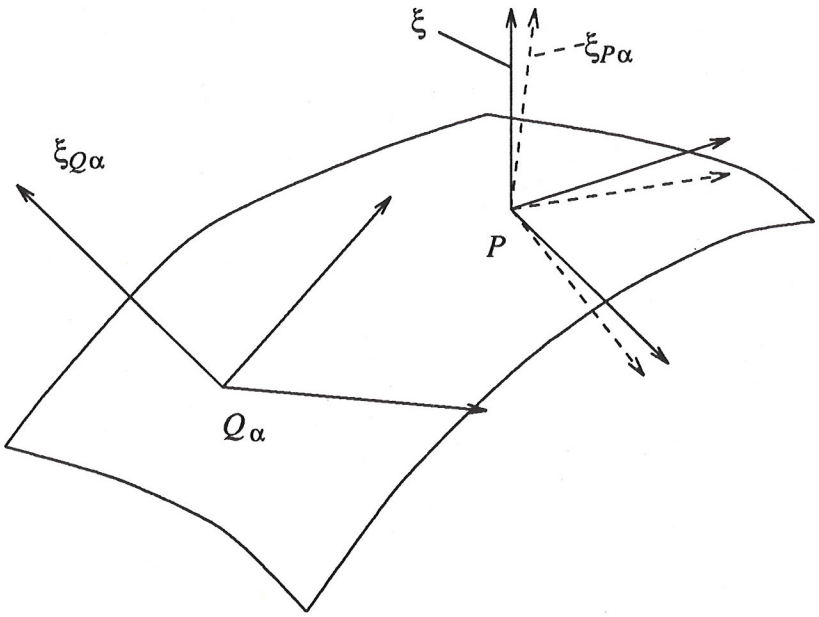

Figure C.2. Local extrapolation

its associated neighbor [73, 74]. This provides a mechanism for setting up a minimization which enforces the local model over the surface, somewhat analogous to the constant curvature assumption in [68]. In practice, the algorithm iteratively updates each $\mathcal{D}(P)$ with a least-squares estimate computed from $\xi_{P\alpha}$. Among the considerations in formulating the algorithm are, (i) the particular form of extrapolation along the surface to obtain $\xi_{P\alpha}$ and (ii) the form of minimization functional applied in updating $\mathcal{D}(P)$. With densely sampled range data, the particular form of parallel is not too critical. Ferrie *et. al.* suggested that the set of frames that comprise the local neighborhood of P is a reasonable approximation of $\xi_{P\alpha}$, we found this to be true and used this approximation in our work. A better approximation can be achieved using either a parabolic quadric [73, 74] or a toroidal patch to estimate $\xi_{P\alpha}$.

The functional itself is set up to minimize the variation in $\mathcal{D}(P)$ subject to constraints on $\mathbf{d}_1(P)$, $\mathbf{d}_2(P)$, and $\mathbf{n}(P)$. These are

$$(\mathbf{n}(P),\mathbf{n}(P)) = 1 \qquad (\mathbf{d}_1(P),\mathbf{d}_1(P)) = 1 \qquad (\mathbf{n}(P),\mathbf{d}_1(P)) = 0. \tag{C.9}$$

As formulated in [73, 74], the minimization consists of two terms corresponding to (i) the surface normal $\mathbf{n}(P)$ and principle curvatures $\kappa_1(P)$ and $\kappa_2(P)$, and (ii) the principle direction $\mathbf{d}_1(P)$. To simplify the analysis, each is minimized independently. The first term

$$E_1 = \sum_{\alpha} ||\mathbf{n}(P) - \mathbf{n}(P_\alpha)||^2 + (\kappa_1(P) - \kappa_1(P_\alpha))^2$$

$$+ (\kappa_2(P) - \kappa_2(P_\alpha))^2 + \lambda_1((\mathbf{n}(P),\mathbf{n}(P)) - 1) \tag{C.10}$$

where

$$\xi_P = (P, \kappa_1(P), \kappa_2(P), \mathbf{d}_1(P), \mathbf{d}_2(P), \mathbf{n}(P)) \tag{C.11}$$

and

$$\xi_{P\alpha} = (P_\alpha, \kappa_1(P_\alpha), \kappa_2(P_\alpha), \mathbf{d}_1(P_\alpha), \mathbf{d}_2(P_\alpha), \mathbf{n}(P_\alpha)). \tag{C.12}$$

The minimization of this quantity is obtained by using the following updating functionals for $\mathbf{n}(P)$, κ_1, and κ_2:

$$\mathbf{n}^{(i+1)}(P) = \frac{<\sum_\alpha x_{\mathbf{n}}^{(i)}(P_\alpha), \sum_\alpha y_{\mathbf{n}}^{(i)}(P_\alpha), \sum_\alpha z_{\mathbf{n}}^{(i)}(P_\alpha)>}{\sqrt{\left[\sum_\alpha x_{\mathbf{n}}^{(i)}(P_\alpha)\right]^2 + \left[\sum_\alpha y_{\mathbf{n}}^{(i)}(P_\alpha)\right]^2 + \left[\sum_\alpha z_{\mathbf{n}}^{(i)}(P_\alpha)\right]^2}} \tag{C.13}$$

$$\kappa_1^{(i+1)}(P) = \sum_\alpha \frac{\kappa_1^{(i)}(P_\alpha)}{n_\alpha} \tag{C.14}$$

$$\kappa_2^{(i+1)}(P) = \sum_\alpha \frac{\kappa_2^{(i)}(P_\alpha)}{n_\alpha} \tag{C.15}$$

where the superscript i refers to the current iteration step and n_α is the number of neighbors used in the approximation.

Since $\mathbf{d}_1$ and $\mathbf{d}_2$ are directions, there is an 180° ambiguity in orientation. The ambiguity is avoided by minimizing the difference of directions in the tangent plane at point P as follows. Express $\mathbf{d}_1$ in tangent plane coordinates as

$$\mathbf{d}_1^\theta = \mathbf{b}_1 \cos\theta + \mathbf{b}_2 \sin\theta \tag{C.16}$$

such that $\mathbf{b}_1$ and $\mathbf{b}_2$ are tangents to the surface at point P and

$$||\mathbf{b}_1|| = ||\mathbf{b}_2|| = 1 \quad \text{and} \quad (\mathbf{b}_1, \mathbf{b}_2) = 0. \tag{C.17}$$

Then

$$E_2 = \min_\theta \sum_{\alpha=1}^{n} [1 - (\mathbf{d}_1^\theta(P), \mathbf{d}_1(P_\alpha))^2]. \tag{C.18}$$

The θ which minimizes this term is found by iterating

$$\theta^{(i+1)} = \tan^{-1}\left[\frac{(A_{22}-A_{11}) + \sqrt{(A_{11}-A_{22})^2 + 4A_{12}^2}}{2A_{12}}\right], \tag{C.19}$$

$$A_{ij} = \sum_{\alpha=1}^{n} (\mathbf{d}_1(P_\alpha), \mathbf{b}_i)(\mathbf{d}_1(P_\alpha), \mathbf{b}_j).$$

The updating function for $\mathbf{d}_2$ is

$$\mathbf{d}_2^{(i+1)} = \mathbf{d}_1^{(i+1)} \times \mathbf{n}^{(i+1)} \tag{C.20}$$

Control over iteration is maintained by tracking the convergence o the derative of a composite measure R_S, which is the sum of local difference measures computed over the surface,

$$R_S^{(i)} = \sum_j R_j(\xi_P^{(i)}, \xi_{P_\alpha}^{(i)}) = \sum_j E_{1j}^{(i)} + E_{2j}^{(i)}, \quad P_j \in S. \tag{C.21}$$

The algorithm is allowed to interate until the difference $|R_S(i) - R_S(i-1)|$ falls below a specified threshold. We have found that the algorithm generaly converges after only 7 iterations, which agrees others experience [31].

Appendix D - Form of Q Matrix

The stablizer, Ω_1 from Section 4.6 can be split into two parts, each part of the form

$$\Omega_{k1}[\mathbf{r}(\mathbf{u})] = \int\int_U \left\{ \omega_{k1}(\mathbf{u}) z_{uu}(\mathbf{u}) + \omega_{k2} z_{uv}(\mathbf{u}) + z_{vv}(\mathbf{u}) \right\} d\mathbf{u}. \tag{D.1}$$

The discrete form of this functional is given by

$$\begin{aligned} \Omega^d_{k1}[\mathbf{r}(\mathbf{u})] = \sum_{(i,j)\in U} \sum \Big\{ & \omega_{k1}[z(i+1,j) - 2z(i,j) + z(i-1,j)] \\ & + \omega_{k2}[z(i+1,j+1) - z(i+1,j) - z(i,j+1) + z(i,j)] \\ & + \omega_{k3}[z(i,j+1) - 2z(i,j) + z(i,j-1)] \Big\}^2 \end{aligned} \tag{D.2}$$

which can be written in vector-matrix form as

$$\Omega^d_{k1}[\mathbf{r}(\mathbf{u})] = \mathbf{z}^T \mathbf{Q}_k \mathbf{z}, \tag{D.3}$$

where $\mathbf{z}$ is the row concatenated vector of z's. The job now becomes one of determining the form of $\mathbf{Q}_k$. This amounts to solving the nodel equation

$$\begin{aligned} \frac{\partial \Omega_{k1}}{\partial z(i,j)} = {} & \omega^2_{k1}(i,j-1)[z(i,j-2) - 2z(i,j-1) + z(i,j)] \\ & -2\omega^2_{k1}(i,j)[z(i,j-1) - 2z(i,j) + z(i,j+1)] \\ & +\omega^2_{k1}(i,j+1)[z(i,j) - 2z(i,j+1) + z(i,j+2)] \end{aligned}$$

$$+\omega_{k1}(i-1,j)\omega_{k3}(i-1,j)[z(i-1,j-1)-2z(i-1,j)+z(i-1,j+1)]$$

$$-2\omega_{k1}(i,j)\omega_{k3}(i,j)[z(i,j-1)-2z(i,j)+z(i,j+1)]$$

$$+\omega_{k1}(i+1,j)\omega_{k3}(i,j)[z(i+1,j-1)-2z(i+1,j)+z(i+1,j+1)]$$

$$+\omega_{k1}(i,j-1)\omega_{k3}(i,j-1)[z(i-1,j-1)-2z(i,j-1)+z(i+1,j-1)]$$

$$-2\omega_{k1}(i,j)\omega_{k3}(i,j)[z(i-1,j)-2z(i,j)+z(i+1,j)]$$

$$+\omega_{k1}(i,j+1)\omega_{k3}(i,j+1)[z(i-1,j+1)-2z(i,j+1)+z(i+1,j+1)]$$

$$+\omega_{k3}^2(i-1,j)[z(i-2,j)-2z(i-1,j)+z(i,j)]$$

$$-2\omega_{k3}^2(i,j)[z(i-1,j-1)-2z(i,j)+z(i+1,j)]$$

$$+\omega_{k3}^2(i+1,j)[z(i,j)-2z(i+1,j)+z(i+2,j)]$$

$$+\omega_{k2}^2(i,j-1)[z(i+1,j-1)-z(i+1,j)-z(i,j-1)+z(i,j)]$$

$$-\omega_{k2}^2(i-1,j-1)[z(i,j-1)-z(i,j)-z(i-1,j-1)+z(i-1,j)]$$

$$-\omega_{k2}^2(i,j)[z(i+1,j)-z(i+1,j+1)-z(i,j)+z(i,j+1)]$$

$$+\omega_{k2}^2(i-1,j)[z(i,j)-z(i,j+1)-z(i-1,j)+z(i-1,j+1)]$$

$$+\omega_{k1}(i-1,j-1)\omega_{k2}(i-1,j-1)[z(i-1,j-2)-2z(i-1,j-1)+z(i-1,j)]$$

$$-\omega_{k1}(i,j-1)\omega_{k2}(i,j-1)[z(i,j-2)-2z(i,j-1)+z(i,j)]$$

$$-\omega_{k1}(i-1,j)\omega_{k2}(i-1,j)[z(i-1,j-1)-2z(i-1,j)+z(i-1,j+1)]$$

$$+\omega_{k1}(i,j)\omega_{k2}(i,j)[z(i,j-1)-2z(i,j)+z(i,j+1)]$$

$$+\omega_{k2}(i,j+1)\omega_{k1}(i,j+1)[z(i+1,j+2)-z(i+1,j+1)-z(i,j+2)+z(i,j+1)]$$

$$-2\omega_{k2}(i,j)\omega_{k1}(i,j)[z(i+1,j+1)-z(i+1,j)-z(i,j+1)+z(i,j)]$$

$$+\omega_{k3}(i-1,j-1)\omega_{k2}(i-1,j-1)[z(i-2,j-1)-2z(i-1,j-1)+z(i,j-1)]$$

VITA