FREE-FORM SURFACE FITTING FOR PRECISION COORDINATE METROLOGY

XIANGCHAO ZHANG

A thesis submitted to the University of Huddersfield
in partial fulfilment of the requirements for
the degree of Doctor of Philosophy

The University of Huddersfield

April 2009

ABSTRACT

Free-form surfaces are increasingly used in optical and mechanical devices due to their superior optical and aerodynamic properties. The form quality plays an essential role in the characteristics of a free-form component. In order to assess the form error, it is necessary to fit the measurement data with a nominal template or analytical function. This thesis focuses on investigating and developing appropriate fitting (matching) algorithms for different kinds of free-form surfaces.

A new algorithm called the Structured Region Signature (SRS) is proposed to provide a rough matching between the data and template. SRS is a global generalised feature which represents the surface shape by a one dimensional function. The candidate location which occupies the most similar signature with the measurement data is considered to be a correct matching position.

The fitted result is then refined to improve its accuracy and robustness. The widely used Iterative Closest Point technique suffers from a slow convergence rate and the local minimum problem. In this thesis the nominal template is reconstructed into a continuous representation using NURBS or radial basis functions if provided as a CAD model or a discrete-point set. The Levenberg-Marquardt algorithm is then applied to calculate the final result. The solution of the traditional algebraic fitting may be biased. The orthogonal distance fitting techniques can effectively overcome this problem. If the template function is explicit, the projection points can be updated simultaneously with the motion and shape parameters; whereas a nested approach is adopted to update the projection points and motion parameters alternately when the template is in a parametric form.

A proper error metric should be employed according to the distribution of the measurement noise, so that the solution can be guaranteed robust and unbiased. Simulation and experimental results are presented to validate the developed algorithms and techniques.

ACKNOWLEDGEMENTS

I would like to thank all the people who explicitly or implicitly supported me during my doctoral research.

First and foremost, I must record my gratitude to Professor Xiangqian Jane Jiang for her help, encouragement, patience and understanding whilst guiding me through this research. Her knowledge on surface metrology, combined with the practical view on measurement and manufacture, is a tremendous help for this work.

Special thanks go to Professor Paul J Scott, who offered me lots of comments and suggestions on mathematical theories and numerical computations, pointed out the mistakes in my work and corrected the grammar faults in my papers and presentations. Without him, I can never accomplish this research.

I appreciate Professor Liam A Blunt for his advices on surface metrology and measuring techniques to my research, Dr Shaojun Xiao and Feng Xie for guidance on software and metrology, Dr Andrew Crampton and Phillip Cooper for helpful discussions on mathematics, and also Dr Paul J Bills for teaching me to use CMM and other measuring instruments.

I also express my gratitude to Mr Haydn Martin, Allan Kennedy, Dr Leigh T Brown and all the other colleagues in the Centre for Precision Technologies for their kind help on my research and life.

I am grateful for the financial support from the Taylor-Hobson Ltd and technical directors that allowed me to devote my time to this research project.

Finally, I give sincere thanks and appreciation to my parents, and my beloved girlfriend Xiqun Lu, with whom I can share my joy of success and from whom I can gain motivation, confidence, and comfort when I am in need.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

1.1 Definition

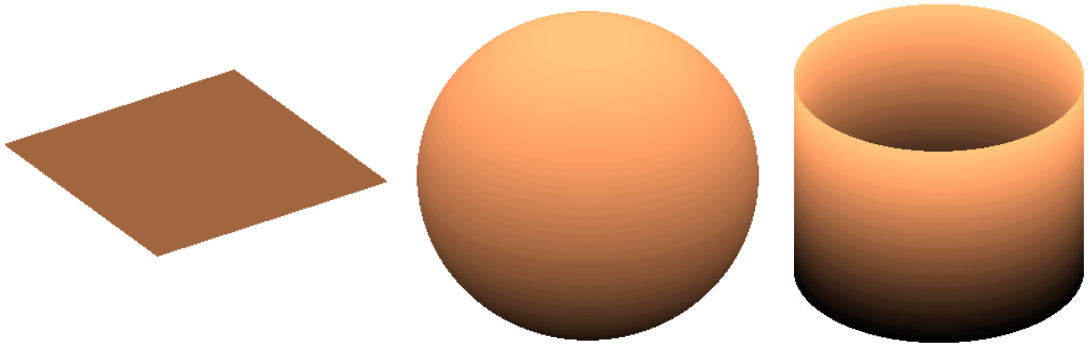
In the metrology field, *free-form surfaces* are defined as the surfaces which have no invariance degree [ISO 17450-1]. This means if translating a free-form surface along any direction or rotating it about an arbitrary axis, the surface cannot remain unchanged. Therefore, a free-form surface has no symmetry in translation or rotation.

The simplest shape in the 3D Euclidean space is a plane. It has three Degrees of Freedom (DoF): two in translation and one in rotation. Another simple surface is a sphere, which has three DoF in rotation. If restricting a plane by one translational DoF, a cylinder comes into being. It is rotationally symmetric about its axis and can remain identical when displaced along the axis. These three shapes are traditionally regarded as 'simple geometries', and appear very commonly in natural objects and artificial products.

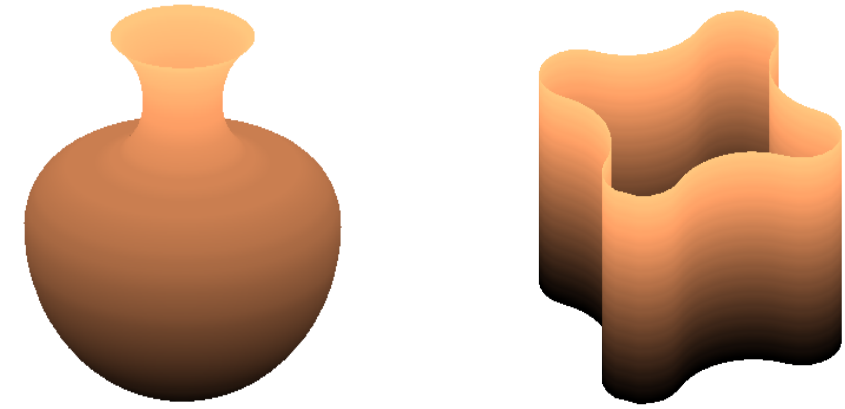
If we eliminate the translational DoF of a cylinder, and make it only rotationally symmetric about the axis, a revolved surface is obtained. It can be created by rotating a curve about one axis. On the contrary, restricting the rotational DoF of a cylinder yields an extruded surface, which is generated by extruding a curve alone a straight line. Instead of eliminating one DoF of rotation or translation, assigning a constraint between these two DoF will lead to a helically symmetric surface, which is termed as a generalized helicoid [Weisstein 2002]. It can be constructed by rotating a twisted curve about a fixed axis and, at the same time, displacing it with a velocity proportional to the angular velocity of rotation.

Finally, by restricting all the DoF of rotation and translation, we can obtain a free-form surface.

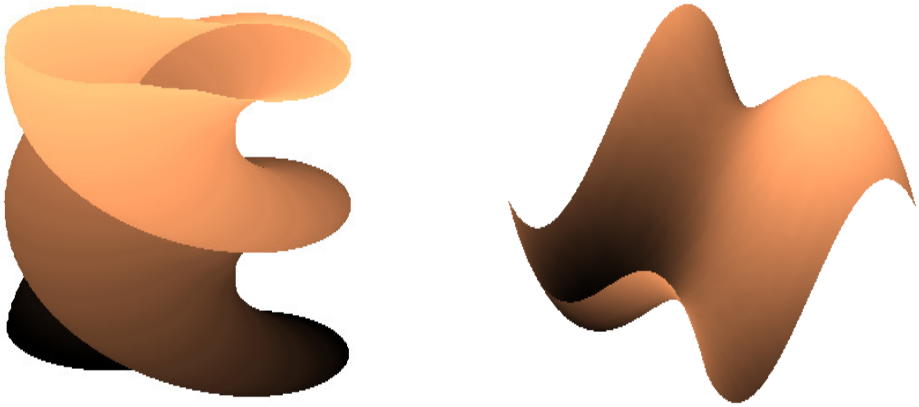
It is proved that all the surfaces have only these seven types of invariance under rigid-body transformations in the 3D Euclidean space. These surfaces are illustrated in Figure 1.1 with their rigid-body invariance (R denotes DoF in rotation and T translation).



(a) Plane (1R+2T) (b) Sphere (3R+0T) (c) Cylinder (1R+1T)



(d) Revolved surface (1R+0T) (e) Extruded surface (0R+1T)



(f) Generalized helicoid (1R+0T)* (g) Free-form surface (0R+0T)

Figure 2.1 Surfaces with different invariance

In other research fields, different definitions have been given for free-form surfaces. Campbell and Flynn [Campbell 2001] defined free-form surfaces as complex surfaces that are not of an easily recognized class such as planes and/or natural quadrics. Another

* The DoF of translation is constrained with rotation.

interpretation was given by Besl [Besl 1990]: “A free-form surface has a well defined surface normal that is continuous almost everywhere except at vertices, edges and cusps”.

1.2 Motivation

With the development of technologically advanced industries, free-form surfaces are more and more widely used in optical and mechanical devices. They have remarkable superiorities over traditional simple-shaped elements.

Firstly, they can simplify the system. Most traditional optical lenses are composed of spherical elements. In order to eliminate aberrations, many pieces of glass are required. On the contrary, if adopting free-form surfaces, only one or two pieces are sufficient to meet all the optical requirements whilst not causing aberrations. Thus free-form surfaces can make the optical system lighter and cheaper. Such examples include microscopes, telescopes and camera lens.

Secondly, free-form surfaces can realize some novel optical functionality. For example, the image height of an F-theta lens used in the scanning system of laser printers is proportional to its scan angle. A Fresnel lens used in a lighthouse enables the construction of lenses with large aperture and short focal length whilst requiring much smaller weight and volume compared with conventional lenses.

Thirdly, free-form surfaces can meet some biological or mechanical requirements. The contacting surfaces of bio-implants should have consistent shapes with real human body bones; otherwise the patient will suffer pain due to conflicting and wear of replacements, then the life length of the implants will be significantly shortened [Blunt 2009]. In aerodynamics and automotive industries, some surfaces have interactions with air or fluid, e.g. 3D cams, seals, turbine blades, impellers, fuselage etc. These surfaces are designed based on their dynamic and mechanical functionality, and imperfect shapes may cause energy waste or even damage of the elements [Savio 2007].

In precision engineering, a fundamental problem is to determine whether a manufactured workpiece meets the requirements of its original design specifications. It is widely recognized that the surface form plays an essential role in the characteristics of a free-form component; hence the component must have extremely high fidelity with the original design. It is critical to evaluate the form error of a free-form surface with respect

to the nominal shape at high precision, and ensure this manufactured item fulfil the design in terms of macro-topography and micro-topography.

The inspection of simple geometries like spheres and cylinders traditionally involves gauges for different shapes and applications [Hume 1970]. Concerning complex-shaped surfaces, e.g. marine propellers [Jastram 1996], it needs highly skilled technicians to check the surface with numerous mechanical gauges. In optical engineering, the form qualities of optics are generally tested with the Newton or Fizeau interferometer [Malacara 2007]. A quality test plate or a reference surface is required. The inspection in this way depends heavily on the technician's proficiency and the manufacturing accuracy of the test plates or gauges. It is evident that the task is very inefficient and expensive, more importantly, the accuracy cannot be guaranteed.

Various automatic techniques have been developed. A component is measured and then a mathematical assessment process follows to quantitatively calculate the form error of the data with respect to the nominal shape. In this way human operation is no longer necessary, thereby greatly saving time and cost, at the same time, improving the evaluation accuracy.

Normally a design template is provided as a reference to represent the nominal shape of a free-form component. The deviation between the measurement data and the template is regarded as the form error of the free-form surface.

When measuring a free-form component, some reference datums like planes or holes on the support are used to establish the measurement coordinate system. Normally the working surface (free-form surface) is machined with higher accuracy than other surfaces, and the alignment of the measured component may not be precise enough, i.e. the measurement data are not exactly located in the same coordinate system with the template, and the form error cannot be calculated by directly subtracting the reference template from the data. Slight misalignment between the two coordinate systems can cause apparent error in the evaluation of form quality. This is fatal for some key free-form elements which have rather high form accuracy and perform critical functionality. Misalignment shall be eliminated to bring the template and data into a common coordinate system, this procedure is called *localization* or *alignment* [Li 2005].

On the other hand if its corresponding standard geometric function is already known, the actual shape of a workpiece can be assessed by recognizing the geometric parameters

(intrinsic characteristics) in the sense of least squares, minimum zone etc [ISO 4291:1985, Forbes 1990], and this kind of manipulation is called *association* [ISO/TS 17450-1: 2005]. Such examples of free-form surfaces include biconic surface, conical surfaces etc [ISO 10110-12:2007]. From the mathematical point of view this procedure can be regarded as the reverse process of manufacturing.

In the present thesis, this association process and the preceding localization problem are both termed as *fitting*.

At present, there is a lack of practical and general-purposed methods to match 3-D free-form surfaces with their templates. This dissertation endeavours to bridge the gap between the free-form measurement data and design functionality. Appropriate fitting techniques will be explored and developed for characterization of free-form surfaces.

1.3 Objectives

Considering their practicability and utility, the fitting algorithms are required to be widely applicable and no prior assumptions or restrictions are assigned onto the surface shape. However, a standardized and universal technique is not desirable for all circumstances; instead, the methods will be application-oriented and surface-shape-related. That is to say, different fitting algorithms will be developed according to the shapes, representations and applications of the free-form surfaces. It is also expected to quantitatively evaluate the form accuracy as an error map, instead of making a simple 'pass/fail' decision.

This research project will address the following major objectives:

1. To review conventional techniques of form error evaluation in the precision metrology field, and survey various matching/fitting methods developed in other research fields, e.g. Computer-Aided Design (CAD), pattern recognition, image processing etc.

2. To generate appropriate mathematical representations for the nominal templates. The design templates sometimes are provided as CAD models or discrete point sets, which are not compatible with the optimization programs of the fitting process. Thus they will be transformed /reconstructed into other proper mathematical representations which are required to have extreme fidelity with the original designed shape.

3. To develop practical and efficient localization techniques to find the best matching between the measurement data and nominal template. Free-form surfaces will be

classified into several categories based on their shapes, and different matching algorithms will be adopted accordingly. Reliable correcting processes are also required to reject false matching results.

4. To improve the accuracy and robustness of the fitting results. Proper error metrics and optimization algorithms will be employed to make sure the fitted results are consistent with the measurement error distributions and robust against outliers and missing data. Compensation may also be implemented to deal with manufacture defects or other physical effects. Extensive attention will be paid on the numerical stability and efficiency. These fitting programs will be coded with MATLAB.

5. To verify the performance of these fitting algorithms with actual experiments. Some case studies will be given to compare the fitted results with some mature commercial software and mathematical tools.

1.4 Approaches

We classify free-form surfaces into three kinds according to their shapes and applications [Jiang 2007],

1. *Smooth surfaces*: surfaces with no steps, edges, or cliffs, in another word, surfaces with a continuous normal vector.

2. *Non-smooth surfaces*: surfaces with very complex topographies, i.e., having many sharp shape-variations like cliffs, small concave and convex parts.

3. *Structured surfaces*: surfaces with a deterministic pattern of usually high aspect ratio geometric features designed to give a specific function [Evans 1999].

The fitting strategy of different free-form surfaces is summarised below,

Case A. If the surface is structured and each part is of a simple geometry, we will fit each section with a quadric function individually, and then determine the form error and position error separately.

Case B. If the surface is non-smooth, it will be very difficult to represent the surface with global mathematical functions. Some nominal points will be sampled on the reference template and the Iterative Closest Point method will be adopted to find the best matching between the two sets of points [Besl 1992].

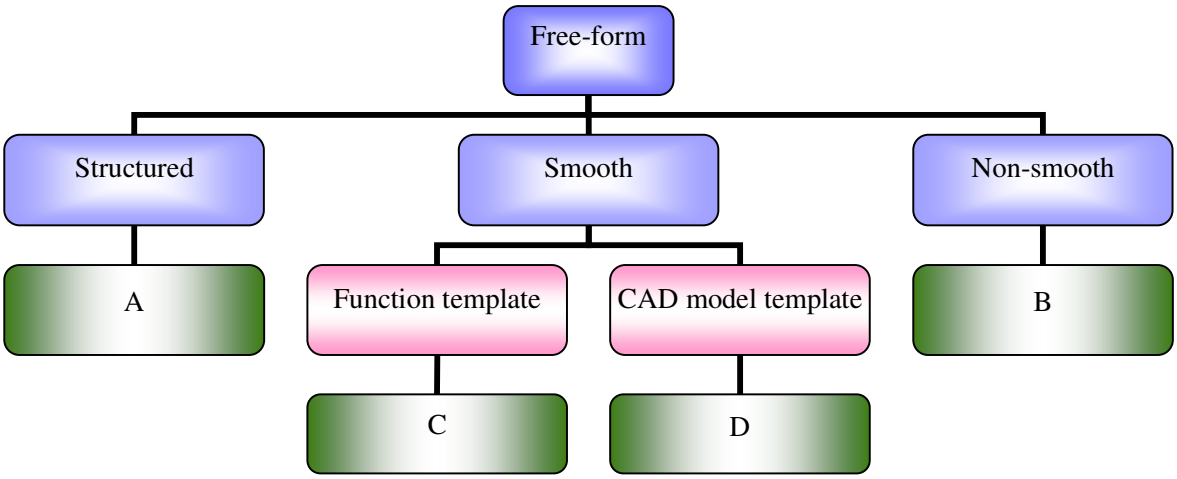


Figure 2.2 Hierarchy of free-form surface evaluation

Case C. Smooth surfaces are of our particular interest in this thesis. This type of surfaces are also termed as *sculptured surfaces* or *curved surfaces*, and they are the most commonly used free-form surfaces. Some special surfaces have their analytical functions, e.g. F-theta surface, biconic surface etc. These surfaces are put into Case C and a design template is not essential for them. The association of these surfaces is very similar with conventional simple geometries, such as sphere and cylinder. If moving the analytical function into a non-standard position, the representation will become rather complicated. It is proved that moving the measurement data is equivalent to moving the template, and their fitting results are the same [Atieg 2003]. As a consequence transformations are always performed onto the measurement data in this dissertation.

Case D. The design template is supplied as a CAD model, and this case is the main task of this thesis. Sometimes it is not straightforward to directly read the design function from the file, but a set of nominal points can be obtained from the template by some software like HOLOS (Carl Zeiss CMM) or Rhinoceros. These discrete points can be reconstructed into a continuous representation with NURBS or Radial Basis Functions. Then the form quality of the workpiece can be evaluated by fitting the measurement data with the reconstructed template. Thereupon,

$$Error_{\text{evaluate}} = Error_{\text{act}} + Error_{\text{measure}} + Error_{\text{fit}} - Error_{\text{reconstrct}} \tag{1.1}$$

Here $Error_{\text{evaluate}}$ and $Error_{\text{act}}$ indicate the evaluated and actual form errors of the component with respect to the design template, whilst $Error_{\text{measure}}$, $Error_{\text{fit}}$ and $Error_{\text{reconstrct}}$ refer to the errors introduced in measurement, fitting programme and

reconstruction programme respectively. In order to make the evaluation result more reliable, i.e. the evaluated form error is closer to the actual form error, the other three terms are hopefully to be as small as possible. The measurement data is provided beforehand, so that the measurement error is fixed and we will not pay much attention to the measuring techniques. Effort will be made to reduce the bias and uncertainty of the fitting algorithms and to improve the accuracy of surface reconstruction. Here the reconstruction and fitting errors are required to be at least one order smaller than the actual form error.

1.5 Structure of the Thesis

In Chapter 2, we review some existing reconstruction techniques for regular-lattice and scattered distributed data. To avoid incorrect results, the whole fitting procedure is divided into two stages, initial matching and final fitting. We briefly introduce some initial matching and final fitting methods in the fields of metrology, computational geometry, CAD, image processing, pattern recognition etc. Moreover, issues about the numerical stability and robustness are also discussed.

NURBS is adopted for reconstructing a surface from discrete points of regular distribution. Chapter 3 gives the five stages of NURBS surface reconstruction: parameterization, selecting knots, determining degree, calculating basis functions and finally, computing the control points. Point inversion is necessary when implementing interpolation; and point projection when finding the closest point on the surface. Some novel techniques are developed to improve the computational efficiency of point inversion and projection.

Chapter 4 focuses on surface reconstruction of scattered points using the radial basis function (RBF). To improve the numerical stability, a centre selection algorithm called orthogonal least squares basis hunting is utilized to build a sparser RBF system. We also suggest adding a circle of new centres outside the domain of interest to improve the boundary behaviour.

Chapter 5 introduces segmentation algorithms to divide a structured surface into patches, and then individually fit each part using a quadric function. A new algorithm, called the structured region signature, is proposed to match smooth free-form surfaces.

When some parts of the template are nearly symmetric, a residual-checking-strategy can be utilized to avoid false matching.

Chapter 6 pays attention to refining the fitting result after initial matching. The traditional Iterative Closest Point (ICP) method suffers from high computational complexity and a local minimum problem. The template is thereby reconstructed into a continuous representation if supplied as a discrete point set, and the Levenberg-Marquardt algorithm is adopted to find the optimal fitting. The fitted parameters of the conventional algebraic fitting may be biased. Hence orthogonal distance fitting programs are developed for explicit and parametric template functions respectively. If the measurement data contain outliers or defects, the ordinary least squares solution will be distorted. In this case, the l_1 norm error metric will be adopted to improve the system robustness.

The thesis concludes in Chapter 7 by summarizing the implemented work in this project and pointing to possibilities of further work.

1.6 References

Atieg, A. and Watson, G. A. 2003 A class of methods for fitting a curve or surface to data by minimizing the sum of squares of orthogonal distances. *J of Computational and Applied Mathematics*. 158(2): 277-296

Besl, P. J. 1990 The free-form surface matching problem. In Freeman, H. Editor. *Machine Vision for Three-Dimensional Scenes*. Academic Press. 25–71

Besl, P. J. and McKay, N. D. 1992 A method for registration of 3-D shapes. *Transactions of Pattern Analysis and Machine Intelligence*.14(2):239-256

Blunt, L., Bills, P., Jiang, X. et al. 2009 The role of tribology and metrology in the latest development of bio-materials. *Wear*. 266(3-4): 424-431

Campbell, R. J. and Flynn, P. J. 2001 A survey of free-form object representation and recognition techniques. *Comp. Vision and Image Understanding*. 81(2):166-210

Evans, C. J. and Bryan, J. B. 1999 'Structured', 'textured' or 'engineered surfaces. *Annals of CIRP*. 48(2): 541-556

Forbes, A. B. 1990 Least squares best fit geometric elements. In Mason, J. C. and Cox, M. G. Editors. *Algorithms for Approximation II*, 311-319

Hume, K. J. 1970 *Engineering Metrology*. 3rd Ed. MacDonald Technical & Scientific: London

ISO 4291: 1985 *Methods for the Assessment of Departure from Roundness-Measurement of Variations in Radius*

ISO 10110-12: 2007 *Optics and Photonics-Preparation of Drawings for Optical Elements and Systems*-Part 12: Aspheric Surfaces

ISO/TS 17450-1:2005. *Geometrical Product Specifications (GPS)-General Concepts*-Part 1: Model for Geometrical Specification and Verification.

Jastram, M. O. 1996 *Inspection and Feature Extraction of Marine Propellers*. MSc Thesis. MIT, USA

Jiang, X., Scott, P. J., Whitehouse, D. J. and Blunt, L. 2007 Paradigm shifts in surface metrology.Part II.The current shift. *Proc Royal Society A*. 463(2085):2071-2099

Li, Y and Gu, P. 2005 Inspection of free-form shaped parts. *Robotics and Computer-Integrated Manufacturing*. 21(4-5): 421-430

Malacara, D. 2007 *Optical Shop Testing*. 3rd Ed. John Wiley & Sons.

Savio, E., de Chiffre, L. and Schmitt, R. 2007 Metrology of freeform shaped parts. *CIRP Annuals-Manufacturing Technology*. 53(2): 810-835

Weisstein, E. W. 2002 *CRC Concise Encyclopedia of Mathematics*. 2nd Ed. Chapman & Hall/CRC. 1174

CHAPTER 2 LITERATURE SURVEY

2.1 Surface Reconstruction

2.1.1 Introduction

Nowadays, precision free-form components are fabricated with Computer-Aided Manufacturing (CAM) techniques, such as single point diamond turning, ultra-precision polishing, electrolytic in-process dressing, plasma chemical vaporization machining etc [Lee 2005]. The design model of a workpiece is generally supplied as a 3D CAD file in formats of IGES, VDA-FS, DXF, SET, and the ISO standard representation STEP [Goldstein 1998]. Since each CAD system has its own method of describing geometries, both mathematically and structurally, exchanging between different CAD systems and formats will more or less lose some information. Moreover, due to the shape complexity of free-form components, the mathematical description of a free-form surface is often composed of a number of separate patches, each individually has its own function, and continuity constraints are assigned at the boundaries between these patches. Consequently it is a tough task to directly read or transform such CAD models.

However, when characterizing the form quality of a free-form surface, we need to know exactly the original design shape as a nominal reference; hence a straightforward continuous representation of the model is required for further mathematical processing. Apparently, it is easy to sample discrete points from the design model through CAD systems; therefore it is feasible to mathematically generate a new continuous representation for the design template from these sampled points for the purpose of surface fitting.

Surface reconstruction (also termed surface *modelling* or *fitting*) is to obtain a continuous surface Q that best explains the given data point set P.

Two closely related concepts are surface *interpolation* and *approximation*. Interpolation generates a surface which passes exactly through all the given data points, while approximation generates a surface which passes near the data points [Dinh 2000]. Usually a ‘good’ reconstruction surface not only fits the given data points well, but also shall satisfy some requirements on their properties, e.g. smoothness and continuity.

Surface representations can be classified into three categories, *explicit*, *implicit* and *parametric forms*.

Explicit In this form, the dependent value is provided explicitly by an equation in terms of the explanatory variables,

$$\mathbf{S}=\{(x,y,z)|z=f(x,y)\} \tag{2.1}$$

Explicit functions include power series, Chebyshev polynomials, radial basis functions, orthogonal bivariate polynomials etc [Huhtanen 2002]. They are easy to understand and implement. However, most closed shaped surfaces cannot be represented in this form. In addition, the geometric meaning of the surface is usually not clearly revealed in the equation.

Implicit The surface is defined by passing through all the given data points where the implicit function evaluates to some specified value (usually zero), i.e.

$$\mathbf{S}=\{(x,y,z)|f(x,y,z)=0\} \tag{2.2}$$

Simple geometries are generally represented in implicit forms, e.g. sphere, paraboloid, hyperboloid etc. Geometric parameters can be revealed in the equations. For general shaped surfaces, Pratt and Taubin proposed to minimize the sum of squared Hausdorff distances from the data points to the zero set of polynomials [Pratt 1987, Taubin 1991]. Muraki adopted a function as a linear combination of three-dimensional Gaussian kernels with different means and spreads [Muraki 1991]. Moore and Warren fitted piecewise polynomials recursively and then enforced continuity between these polynomials using a freeform blending technique [Moore 1990].

Parametric The surface is described by a parametric equation with two parameters,

$$\mathbf{S}(u,v)=\begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix} \tag{2.3}$$

where u and v are called foot-point parameters. Parametric forms are the most general way to specify a surface. They have the following advantages [Campbell 2001],

- They are mathematically complete, i.e. they can completely and faithfully preserve the geometrical information of an original model.
- They are easy to be sampled.

• They facilitate design: models can be designed in terms of patches whose continuity can be controlled at the boundaries.

• Their representation power is strong: they can represent very complex objects and geometries.

• They can be used to generate realistic views.

• Reconstruction technologies for parametric representations have been well developed.

Therefore, parametric surfaces are widely used for surface reconstruction and object modelling.

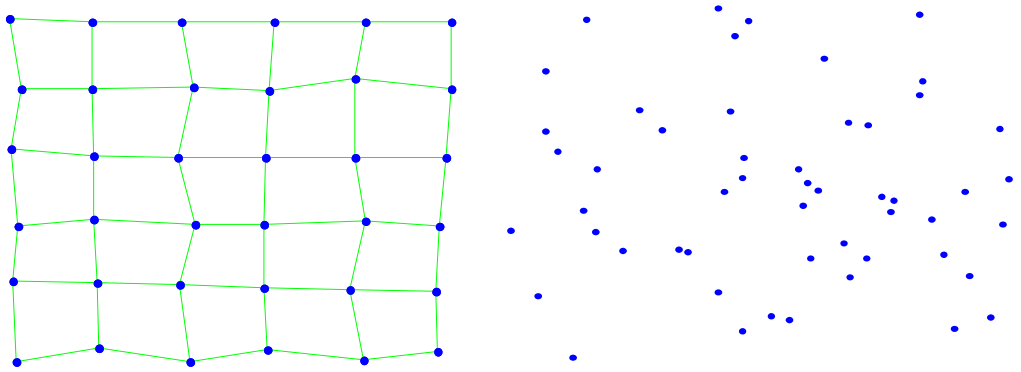
The most common parametric surfaces may be quadric surfaces [Forbes 1990]. In their equations, the radius and azimuth angles are adopted as foot-point parameters. Different with implicit or explicit representations, each parametric coordinate may have distinctive geometric meaning. Hence parametric forms are preferred in some special applications, such as in navigation and astronomy.

If generalizing quadric surfaces further, superquadrics and generalized cylinders come out [Campbell 2001]. They are capable of representing a large class of complex shapes, and are of special interest in geometric modelling.

In order to improve the computational efficiency, some standardized modelling techniques have been developed. In these methods, the surface representations can be derived using some premised techniques and they are invariant under rigid body transformations. Smoothness and continuity conditions will be automatically satisfied. Such examples include B-spline, Bézier surfaces etc, which will be introduced in Sections 2.1.2 and 3.1.

2.1.2 Reconstruction Methods for Regular Lattice Data

An open surface patch can be regarded as a function respect to two independent variables, e.g. x and y. In this thesis all the surfaces are considered to outspread in the 2D domain of X-Y plane, unless stated otherwise. If a 3D point set is unorganized and no information is provided regarding the connectivity relationship between the points, these points are thought to be *scattered*. Conversely if the X-Y coordinates of these points, or their corresponding location parameters after a simple space transformation, are located in a regular grid, they are respected as *regular lattice* points, as shown in Figure 2.1.



(a) Regular lattice points (b) Scattered points

Figure 2.1 Regular lattice and scattered distributed points

The two variables of points located on a regular lattice are separable when implementing surface reconstruction, i.e. we can construct bases in x and y directions independently,

$$\begin{cases} \phi(x;\mathbf{a}) = \sum_{k=1}^{S} a_k \phi_k(x) \\ \psi(y;\mathbf{b}) = \sum_{l=1}^{T} b_l \psi_l(y) \end{cases} \tag{2.4}$$

Then the surface can be represented as a tensor product,

$$z = \phi(x;\mathbf{a}) \times \psi(y;\mathbf{b}) = \sum_{k=1}^{S}\sum_{l=1}^{T} a_k b_l \phi_k(x)\psi_l(y) = \sum_{k=1}^{S}\sum_{l=1}^{T} c_{kl}\varphi_{kl}(x,y) \tag{2.5}$$

In the procedure of surface reconstruction, the form of the bases $\{\phi_k\}$ and $\{\psi_l\}$ is pre-set by the user and the coefficients $\mathbf{a}$ and $\mathbf{b}$ should be calculated from the data.

The reconstruction of tensor product surfaces is very efficient and numerically stable for regularly distributed points. On the other hand, they are less efficient for band surfaces with local areas of great shape variations or for surfaces which have different behaviour in different regions.

Some extensively adopted tensor products are that of two curves represented by Chebyshev polynomials, polynomial splines, B-splines etc. A brief review is given below.

The simplest form of curves is the power series,

$$f_n(x) = \sum_{k=0}^{n} a_k \phi_k(x) = \sum_{k=0}^{n} a_k x^k \tag{2.6}$$

When the curves become very sophisticated, the corresponding degree n is required to be increased simultaneously. Hence the values of the bases can be unacceptably large and an ill-conditioned matrix is generated. To overcome this problem, *Chebyshev polynomials* are proposed [Abramowitz 1965]. They normalize the x coordinates by,

$$z = \frac{x - (x_{\max} + x_{\min})/2}{(x_{\max} - x_{\min})/2} \tag{2.7}$$

and defines the basis $T_k(z)$ recursively,

$$T_0(z) = 1$$

$$T_1(z) = z$$

$$T_k(z) = 2zT_{k-1}(z) - T_{k-2}(z),\ k \geq 2 \tag{2.8}$$

These basis functions are orthogonal to each other with respect to the weighting $w(z) = 1/(1-z^2)^{1/2}$ within the interval [-1,1],

$$\int_{-1}^{1} T_m(z)T_n(z)w(z)dz = \begin{cases} 0 & m \neq n \\ \pi & m = n = 0 \\ \pi/2 & m = n > 0 \end{cases}$$

Therefore, the interpolation matrix is diagonally dominant and the system will be much more stable.

Monomial and Chebyshev polynomials are very flexible and suited for smooth curves which behave similarly at different parts. As regards some curves with specific behaviour, e.g. asymptotic curves, some special functions will be adopted, like asymptotic polynomials [Barker 2004], rational functions [Petrushev 1987] etc.

All the above methods represent the whole curve using a single function. They are relatively easy to calculate. But when surface shapes become rather complex or show distinctive behaviour at each part, these methods need to construct a high-degree function and the Runge's phenomenon will arise. Thus a whole surface/curve can be divided into sections and represented piecewisely by a series of low order polynomials, which is called *spline*.

A common form of polynomial spline curves is,

$$s(x;\mathbf{a},\mathbf{c}) = \sum_{k=1}^{S} c_k (x-\lambda_k)_+^{n-1} + p(x;\mathbf{a}) \tag{2.9}$$

where $p(x;\mathbf{a})$ is a polynomial of degree n-1. $x_{\min} < \lambda_1 < \lambda_2 < \cdots < \lambda_S < x_{\max}$ are called knots or breakpoints [Piegl 1997].

$$\phi_k(x) = (x-\lambda_k)_+^{n-1} = \begin{cases} 0 & x < \lambda_k \\ (x-\lambda_k)^{n-1} & \text{otherwise} \end{cases}$$

are the truncated power functions.

In practice, this kind of splines may suffer a severe ill-conditioning problem. Additionally, the coefficients **a** and **c** convey very little insight about the geometric shape of the curve. In 1960s, Pierre Bézier developed a very interesting representation for curves, now called *Bézier curves* [Bézier 1972],

$$C(u) = \sum_{k=0}^{n} B_{k,n}(u)\mathbf{P}_k \ , 0 \le u \le 1 \tag{2.10}$$

In the equation, $\{B_{k,n}(u)\}$ are the classic n-th degree Bernstein polynomials,

$$B_{k,n}(u) = \frac{n!}{k!(n-k)!} u^k (1-u)^{n-k} \tag{2.11}$$

The coefficients $\{\mathbf{P}_k\}$ are called control points. They form a control polygon and the curve is contained in the convex hull of the control points, as shown in **Figure 2.2**.

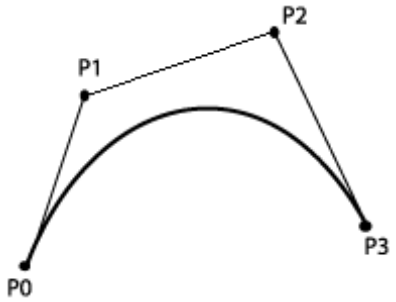


Figure 2.2 A cubic Bézier curve

Bézier curves have some very attractive properties,

- The two ending points lie at the two end control points.
- The tangent directions at the ends are parallel to $\mathbf{P}_1 - \mathbf{P}_0$ and $\mathbf{P}_n - \mathbf{P}_{n-1}$ respectively.
- Moving any control point, the curve moves in the same direction with it.

• The basis functions $\{B_{k,n}(u)\}$ have pre-defined form and do not rely on the data points.

Bézier curves use a single polynomial for the whole curve. When the curve is very complicated, the degree n increases. Thus the Bézier curves suffer the analogous numerical problems as the monomial series. Again, a curve can be divided into several sections. The spline-form expansion of a Bézier curve is a *B-spline curve* [Schoenberg 1967],

$$C(u) = \sum_{k=1}^{S} N_{k,n}(u)\mathbf{P}_k \tag{2.12}$$

Unlike Bézier curves, the degree n of the basis functions $\{N_{k,n}(u)\}$ is not necessarily related to the number of the control points S. The basis functions can be calculated recursively by the de Boor-Cox algorithm [de Boor 1972, Cox 1972],

$$N_{k,0}(u) = \begin{cases} 1 & u_k \le x < u_{k+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{k,p}(u) = \frac{(u-u_k)N_{k,p-1}(u)}{u_{k+p}-u_k} + \frac{(u_{k+p+1}-u)N_{k+1,p-1}(u)}{u_{k+p+1}-u_{k+1}},\ p>0 \tag{2.13}$$

In the equation $0 = u_1 \le u_2 \le \cdots \le u_S \le u_{S+1} = 1$ are knots.

Some attractive properties of B-splines are listed below [Piegl 1997],

• $N_{k,p}(u) = 0$ if u is outside the interval $[u_i, u_{i+p+1})$, hence B-spline curves have local supporting property.

• $N_{k,p}(u) \ge 0$ holds true for all k, p, and u.

• All derivatives of $N_{k,p}(u)$ exist in the interior of a knot span.

• $N_{k,p}(u)$ is p-m times continuously differentiable at a knot, where m is the multiplicity of the knot.

For the sake of computational simplicity, the knots are usually sampled uniformly. If sampling the knots non-uniformly and writing Equation (2.12) in a rational form, we obtain

$$C(u) = \frac{\sum_{k=1}^{S} N_{k,n}(u) w_k \mathbf{P}_k}{\sum_{k=1}^{S} N_{k,n}(u) w_k} \tag{2.14}$$

In the equation, $\{w_k\} \geq 0$ are weighting parameters. This is the well-known *Non-Uniform Rational B-spline* (NURBS). Its properties and the detailed reconstruction procedure will be presented in Chapter 3.

2.1.3 Reconstruction Methods for Scattered Data

The tensor product methods do not apply for the surface reconstruction from scattered points, thereby various techniques have been proposed. These techniques can be roughly classified into global methods and local methods.

Global methods have no restriction on the structure of the data points and the connectivity information is not required. The interpolation value is generally written as a weighted sum of all the data points,

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{N} w_i(\mathbf{x}) f_i}{\sum_{i=1}^{N} w_i(\mathbf{x})} \tag{2.15}$$

Here $\{f_i\}$ are some function values associated with the data points $\{\mathbf{x}_i\}$. The weighting parameters $\{w_i\}$ are assigned based on the distances from the data points to the evaluation location $\mathbf{x}$. The simplest way to assign the weighting is to make it inversely proportional to the distance, $w_i \propto 1/\|\mathbf{x}-\mathbf{x}_i\|$ [Shepard 1968]. The main drawback of this method is that the interpolant is in general not particularly smooth.

If extending the function $f(\mathbf{x}_i)$ further into other functions with respect to the distances from the input data to some preset 'centres', it will become the well know *radial basis functions*. The form of the functions is irrelevant with the interpolation values and the weighting parameters are calculated from the interpolation data. This will be discussed in Chapter 4.

Local methods divide the whole surface into small simplicial complexes, e.g. vertices and triangles. The connectivity and neighbourhood relationship between them is

established. Reconstruction is implemented by interpolating the neighbour data points of the evaluation location.

The finite element method [Burnett 1987] takes the input data as nodes. The node coordinates are interpolated over an element using C^1 interpolation functions. Curvilinear elements can be defined by specifying nodal derivatives.

Franke and Nielson [Franke 1980] modified Equation (2.15) into,

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{N} w_i(\mathbf{x}) Q(\mathbf{x}_i)}{\sum_{i=1}^{N} w_i(\mathbf{x})} \tag{2.16}$$

where $w_i(\mathbf{x}) = \frac{\left(R_w - \|\mathbf{x} - \mathbf{x}_i\|\right)_+}{R_w \|\mathbf{x} - \mathbf{x}_i\|}$ and $Q(\mathbf{x}_i)$ are quadric polynomials.

Each data point only influences the interpolated values within its neighbourhood of radius R_w. It has local supporting property and the observation matrix becomes banded, thus the calculation of the system will be more efficient. The resulting interpolation function is C^1 continuous.

Franke adopted a rectangle based method [Franke 1977]. It represents the interpolation function similarly as Equation (2.16). The weighting is taken as,

$$w_i = \begin{cases} 1 - \left(\frac{d_i}{R_i}\right)^2 \left(3 - 2\frac{d_i}{R_i}\right) & d_i < R_i \\ 0 & \text{otherwise} \end{cases} \tag{2.17}$$

where R_i is the distance between $\mathbf{x}_i$ and its fifth closest neighbour and $d_i = \|\mathbf{x} - \mathbf{x}_i\|$. One of the chief benefits of this approach is, compared with taking w_i with disks centred at the (x_i, y_i) as support region, it is easier to use a smaller number of overlapping rectangles in such a fashion that at most four terms in the sum are nonzero.

Triangulation based methods are very extensively used in computational geometry. They establish the connectivity relationship between the data points with the *Delaunay triangulation algorithm* [Delaunay 1934], which neglects all the non-neighbouring points in the Voronoi diagram of the given point set and avoids poorly shaped triangles. The Delaunay based reconstruction methods can be classified into four categories: tangent plane methods, restricted Delaunay based methods, inside/outside labelling, and empty

balls methods. Alpha shapes and the Crusts algorithm are the two most widely used algorithms [Cazals 2004]. Triangular representation can reconstruct shapes of arbitrary topology and scalability to arbitrary accuracy, as long as the triangular mesh is dense enough and the weighting function is properly selected. However, the quality of the reconstructed surface relies heavily on the accuracy of the data points. The vertices of triangulation surfaces are a subset of the original data points. Any error of the data points will directly translate to the reconstructed surface [Dinh 2000]. A very comprehensive survey for Delaunay triangulation methods is given in [Alexa 2005].

Recently researchers have also introduced the B-spline and NURBS techniques into scattered data interpolation. After organizing the scattered data into triangles or tetrahedrals, a B-spline or NURBS surface can be defined for each element. Continuity conditions are then assigned at the boundaries [Han 1996, Bajaj 2003]. Gregorski et al [Gregorski 2000] decomposed the data with a strip tree. A set of quadric surfaces are fitted through the data points and then blended together to form a set of B-spline surfaces.

Some commercial graphic and modelling software has emerged in the market, e.g. 3Ds Max (Autodesk), AC3D (Inivis), Lightwave 3D (newTek), Maya (Autodesk), and so on. The software implements interpolation based on meshes or NURBS surface patches. It concentrates on salient features, basic shapes, and visualization, therefore works well for virtual reality modelling and animation. But the interpolation accuracy is very poor. As a result it is not suited for the purpose of high precision reconstruction in the metrology field.

2.2 Initial Matching Methods

2.2.1 The Two-Phase Matching Strategy

In order to evaluate the form quality of a free-form surface, it is required to compare the deviation between the measurement data and the nominal surface. But usually the measurement data and the template do not exactly lie in the same coordinate system. Thus it is necessary to transform the measurement data to an appropriate position and to align it with the design template.

Matching (in different research fields, it is also termed *alignment*, *best-fitting*, *registration*, or *localization*) is generally formulated as an optimization problem involving the search for pose parameters that minimize an objective function which

quantifies the matching quality, such as the average squares distance between the measurement data and the template surface,

$$E = \frac{1}{N}\sum_{i=1}^{N} \left\| \mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i \right\|^2 \tag{2.18}$$

where $\mathbf{q}_i$ is the corresponding template point of an arbitrary measurement point $\mathbf{p}_i$. $\mathbf{t}$ is the translation vector and $\mathbf{R}$ is the optimal rotation matrix,

$$\mathbf{R} = \mathbf{R}_z(\theta_z)\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x) = \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 \\ -\sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix} \tag{2.19}$$

Here θ_x, θ_y and θ_z are the rotation angles about the *x*, *y* and *z* axes respectively. A free-form surface has no invariance against rigid-body transformations; hence the six degrees of freedom in transformation will all be taken into account.

In practice the number of measured points is far more than the parameters to be fitted, thus surface fitting is an over-determined problem. In order to eliminate redundancy, a unique solution is required to specify the six motion parameters and sometimes, the best-fitted shape parameters (intrinsic characteristics) as well. These parameters are generally obtained via an iterative optimization procedure under a particular criterion (error metric). Due to the non-convexity of the optimization problem, the solution may be trapped at a local minimum or even become divergent if the initial guess is not properly supplied. Therefore, the whole fitting procedure is divided into two phases: initial matching (coarse matching or rough matching) and final fitting (refinement).

Initial matching intends to find a rough position for the measurement surface with respect to the design template.

Traditional approaches in mechanical engineering are to manually align workpieces with the measuring instruments involving special tools, fixtures or other part presentation/orientation devices [Gunnarson 1987, Sahoo 1991] or to perform human-computer interaction [Pulli 1999, Fan 2001]. These methods are very onerous and slow.

Take the Carl Zeiss Coordinate Measuring Machine (CMM) as an example. It has excellent measurement capability and applies to various shapes. In order to measure complex shaped workpieces, the collateral software HOLOS is embedded for CAD models. Given a regular workpiece, it aligns the measurement coordinate system with the

model by *Base Alignment*, e.g. using a plane to define the direction of an axis and using the centre of a hole to define the origin. If a surface is very smooth and no salient feature exists, the software selects six points on the workpiece and six correspondence points on the model, and then matches the two surfaces by overlapping these six point pairs. The result is very rough and not reliable.

Various automatic matching techniques have been developed in different research fields, such as pattern recognition, computer graphics and vision, computer aided geometric design, image processing, reverse engineering etc.

2.2.2 Review of Initial Matching Techniques

Due to the complexity of surface shape and the huge number of the data points, it is not appropriate to directly utilize the whole surface or all the data values for initial matching. Instead, some *features* (or termed *descriptors* [Bustos 2005] or *signatures* [Yamany 1999]) will be defined and adopted as measures of initial matching.

From the machine learning theory it is known that the more sophisticated an algorithm is, the more likely that it will overfit the experimental data, thus making it less robust [Liu 2004]. With regards to this, the feature should not be too complicated or memory-consuming. It is hopefully to satisfy the following properties [Mortara 2001, Campbell 2001]:

- *Ambiguity* measures the descriptor's ability to completely define the object in the model space. It is also referred to as completeness.
- *Conciseness* represents how efficiently (compactly) the descriptor defines the surface.
- *Uniqueness* measures whether there is more than one way to represent the same object by the given construction methods of the descriptor.
- *Invariance* means not changing under translation, rotation or sometimes scaling.
- *Rich local support* refers to being locally insensitive to modification of the shape occurring far from the current focus.
- *Stability* measures the perturbation of the feature caused by the perturbation of the shape.
- *Saliency* is the qualities that allow surfaces to be discriminated from one another.

Here rough matching algorithms are classified into six categories: global feature based methods, manufacturing feature recognition based methods, local feature based methods, surface geometry based methods, image based methods, and graph based methods.

(a) Global Feature Based Methods

Global feature based methods use global properties of the models such as statistical moments, invariants, Fourier descriptors, and geometry ratios. These methods describe the whole surface using one single or several parameters, thus they fail to capture the specific details of a shape, and fail to discriminate among locally dissimilar shapes.

Paquet et al [Paquet 2000] defined the bounding volume of a 3D object to be the minimal rectangular box that encloses a 3D object. They adopted the occupancy fraction of the object within its bounding volume and the orientation of the box as volume descriptors.

Corney et al [Corney 2002] proposed to calculate the convex hull of a 3D object. Some values are obtained from the convex hull, e.g. hull crumpliness, hull packing and hull compactness. These values can be taken as measures of the similarity between two objects.

Wang et al [Wang 1997] adopted some simple global features: feature points, feature lines, and feature planes. The gravity centre is defined as the feature point and the best fitted plane is taken as the feature plane. The feature line is the vector from the gravity centre pointing to the farthest point on the surface. Then the two surfaces can be aligned by overlapping these features.

Cheung et al developed a simple method called the five-point method [Cheung 2006]. For each surface they defined five characteristic points: gravity centre and four corner points. Then the gravity centres of the two surfaces are overlapped and the measurement surface is rotated to minimise the sum of the distances between the five characteristic points on the two surfaces.

A p+q+r order moment of a 3D model $\mathbf{Q}(x, y, z)$ is defined as [Zhang 2001],

$$M_{pqr} = \iiint x^p y^q z^r \rho(x, y, z) dxdydz \tag{2.20}$$

where $\rho(x, y, z)$ is an indicator function,

$$\rho(x, y, z) = \begin{cases} 1 & \text{if } (x, y, z) \text{ is on the surface} \\ 0 & \text{otherwise} \end{cases} \tag{2.21}$$

The coordinate values in Equation (2.20) have been normalised with respect to the gravity centre in order to make the moments invariant to translation.

A 3×3 matrix can be constructed for each model,

$$\mathbf{S} = \begin{bmatrix} M_{200} & M_{110} & M_{101} \\ M_{110} & M_{020} & M_{011} \\ M_{101} & M_{011} & M_{002} \end{bmatrix} \tag{2.22}$$

In the equation, the three subscripts of each element represent the corresponding orders p, q and r of the x, y and z coordinates respectively. The principal axes can be obtained by principal component analysis upon **S**, and the rotation angles are gained by aligning the principal axes of the two surfaces.

Other moments have also been proposed for some particular applications, e.g. partial moments [Duda 1973], Zernike moments [Kohtanzad 1990], rotation-invariant moments [Ghorbel 2006] etc.

Another kind of global feature is spherical harmonics [Groemer 1996]. The spherical harmonics $\{Y_l^m(\theta,\varphi)\}$ are the angular portion of the solution to Laplace's equation in spherical coordinates where azimuthal symmetry is not present.

Any spherical function $f(\theta,\varphi)$ can be decomposed as the sum of its harmonics,

$$f(\theta,\varphi) = \sum_{l\geq 0}\sum_{|m|\leq l} a_{lm} Y_l^m(\theta,\varphi),\ 0\leq\theta<\pi, 0\leq\varphi<2\pi \tag{2.23}$$

where $\{a_{lm}\}$ are the Fourier coefficients. The similarity between two surfaces can be assessed based on their spherical harmonics.

(b) Manufacturing Feature Recognition Based Methods

Manufacturing feature indicates certain non-unique shape characteristics which the required part possesses, realized as a consequence of applying some manufacturing processes to the stock, e.g. holes, slots, pockets etc [Wang 1989].

Feature recognition techniques generally represent the shape of a 3-D object by a set of features extracted from CAD models or drawings. It is required to provide an intelligent interface to understand the meaning of the product design information. Some

approaches, such as rule-based algorithms [Kyprianou 1980], graph-based algorithms [Joshi 1988], logical inference etc have been developed.

If the design data is represented in a Boundary Representation (B-Rep) form, the graph-based method could be used, because a B-Rep data structure can be easily transformed into a graphical representation [Subrahmanyam 1999].

The group technology describes parts according to the design and manufactory attributes based on drawings or CAD/CAM models, e.g. the main shape features, production quality, material etc [Venugopal 1999, Yager 1994]. All the attributes are represented with binary numbers or numeric values and result in a string of features. Similarities between different parts are determined by comparing their strings.

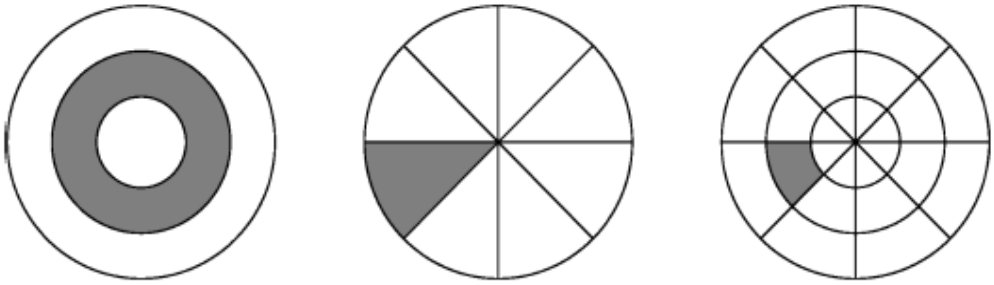
Chen et al [Chen 2001] developed a feature extracting method which combines morphological feature extraction and geometric hashing. They used skeletons to extract features and to compare 3-D objects.

(c) Local Feature Based Methods

Local features can be defined to represent the geometrical information at the neighbourhood around a point. If organizing the local features of a 3D model into a histogram or distribution to represent their frequency of occurrence, similarity between surfaces or models can be determined by comparing their histograms [Iyer 2005]. The effectiveness of these algorithms depends on the number of samples, which is inversely related with the matching efficiency.

Osada et al [Osada 2002] proposed to describe the shape of a 3D object as a 1-D probability distribution sampled from a shape function. The shape function is usually very simple and easy to calculate, e.g. the distance between two points, area of a triangle, angle between two lines etc. The shape distribution is invariant under rigid body transformation and robust against small distortions.

Ankerst et al [Ankerst 1999] partitioned the enclosing space of an object using a shell model, sector model, or combined model, as illustrated in **Figure 2.3**. Then they established a histogram by calculating the point fraction that fall into each partition.



(a) 3 shell bins (b) 8 sector bins (c) 24 combined bins

Figure 2.3 Shell model partitioning

Suzuki et al [Suzuki 2000] proposed to perform principal component analysis onto a 3D model. A unit cube embodying the model is placed with its origin at the centroid of the object and perpendicular to the principal axes. The cubic is partitioned into 7×7×7 cells and the point number contained in each cell is calculated. All the cells are associated to 21 equivalence classes and the point number of each class is aggregated. As a consequence the final descriptor of dimensionality 21 is obtained.

Some researchers adopted another approach. Instead of organizing the local features first and then compare the constructed histograms, they directly established a list of correspondence pairs between some points or local features. For each pair, all the transformations that map them together were computed. The subspace of transformations was discretized and one vote was given for each such transformation. The cell of transformation with the maximal number of votes is regarded as the correct one [Barequet 1999, Olson 1997]. Histograms only work well to match whole objects, but voting algorithms can also be employed for partial matching.

Ko et al adopted a curvature based method called the KH method [Ko 2005]. Given an arbitrary point **p** on a smooth surface, its mean curvature H and Gaussian curvature K are calculated. On the measurement surface, one 3-tuple (a group composed of three points) $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$ is selected. On the template surface, all the points satisfying the curvature constraints,

$$\begin{aligned} &| H_i - H_j' | < \delta_H \\ &| K_i - K_j' | < \delta_K \end{aligned} \tag{2.24}$$

are selected as candidate correspondence points. Some 3-tuples $\{\mathbf{q}_1,\mathbf{q}_2,\mathbf{q}_3\}$ are chosen from them which satisfy the following Euclidean-distance constraints simultaneously,

$$\begin{cases} \left| \|\mathbf{p}_1 - \mathbf{p}_2\| - \|\mathbf{q}_1 - \mathbf{q}_2\| \right| < \delta \\ \left| \|\mathbf{p}_1 - \mathbf{p}_3\| - \|\mathbf{q}_1 - \mathbf{q}_3\| \right| < \delta \\ \left| \|\mathbf{p}_2 - \mathbf{p}_3\| - \|\mathbf{q}_2 - \mathbf{q}_3\| \right| < \delta \end{cases} \tag{2.25}$$

where δ is a user-defined tolerance.

Thus correspondences will be found between $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_j\}$ on the two surfaces and transformation is obtained with the voting method.

Some curvature variations, e.g. curvedness $K = \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}}$ and shape index $\eta = \frac{1}{2} - \frac{1}{\pi} \tan^{-1} \frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}$ can also be employed as local shape descriptors [Sukumar 2004].

The above curvatures and their variations are computed differentially and is not robust against noise. For this reason, an integral volume descriptor is proposed [Gelfand 2005],

$$V_r(\mathbf{p}) = \iiint_{B_r(\mathbf{p}) \cap \bar{S}} dxdydz \tag{2.26}$$

As shown in **Figure 2.4**, the integration kernel $B_r(\mathbf{p})$ is a sphere of radius r centred at the point $\mathbf{p}$, and $\bar{S}$ is the interior of the surface, such that $V_r(\mathbf{p})$ is the volume of the intersection between the sphere $B_r(\mathbf{p})$ with the interior of the model. It is demonstrated that $V_r(\mathbf{p})$ is related with the mean curvature H,

$$V_r(\mathbf{p}) = \frac{2\pi}{3} r^3 - \frac{\pi H}{4} r^4 + O(r^5) \tag{2.27}$$

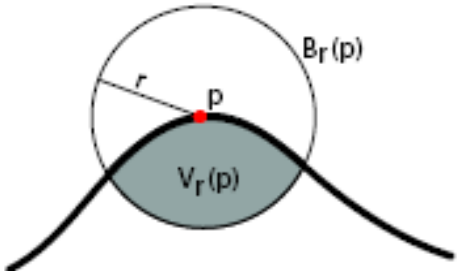


Figure 2.4 Definition of the integral volume descriptor

(d) Surface Geometry Based Methods

For most of the smooth free-form surfaces, there is no salient geometric feature. To take advantage of the simplicity of feature-based methods, researchers have developed

some generalised features mathematically and geometrically for smooth surfaces. This kind of method employs an intermediate representation to aid a matching stage. Usually the 3-D information is broken down into a stack of 2-D descriptors on which robust 2-D shape matching techniques can be applied.

Spin image is the most widely used generalised feature [Johnson 1997]. A surface is presented as a mesh and the normal vectors of the points are given to form oriented points. Some points of interest are sampled on both surfaces. Given a point of interest **p**, a plane **P** is calculated through the point **p** and oriented perpendicularly to the normal vector **n**, as described in **Figure 2.5**. All the points, whose normals possess an angle smaller than a given threshold with respect to **n**, compose a region. The projection distances $\{\beta\}$ from these points to the plane **P** and the distances $\{\alpha\}$ to the normal vector **n** form a 2-D histogram. The histogram is called a spin image associated with the point **p**.

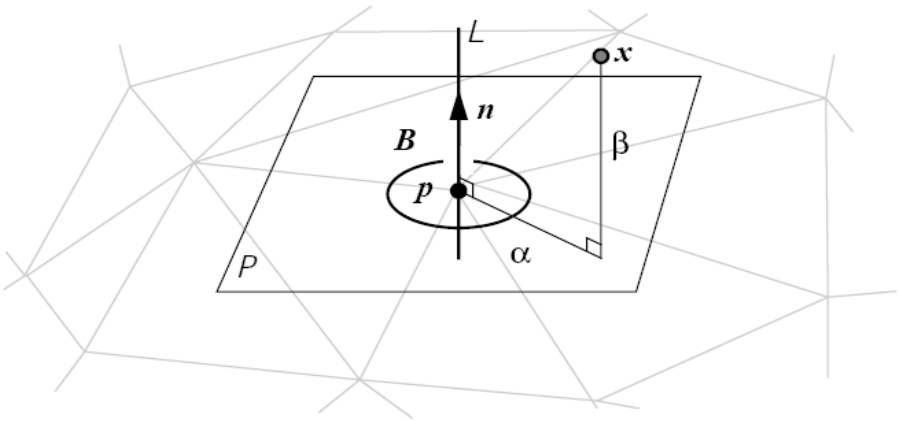


Figure 2.5 Creation of spin image

Then the correspondence points on the two surfaces are decided based on the similarity of their spin images. Transformation is calculated to overlap these point pairs.

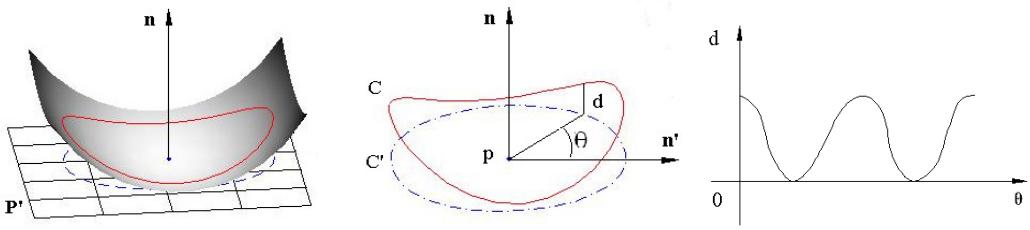
Harmonic shape image is another 2-D feature based method [Zhang 1999]. Provided a 3-D surface **S**, let **v** denote an arbitrary vertex on **S** and $D(\mathbf{v}, R)$ the surface patch centred at **v** with radius R. R is the greatest distance along the surface for all the points within the patch. The unit disc **P** on a 2-D plane is selected to be the target domain and $D(\mathbf{v}, R)$ is mapped onto **P** by minimizing an energy function,

$$E(\phi) = \frac{1}{2}\sum k_{ij} \| \phi(v_i) - \phi(v_j) \|^2 \tag{2.28}$$

where ϕ is the interior mapping and $\mathbf{v}_i$, $\mathbf{v}_j$ are the interior vertices of D and P respectively. k_{ij} is a spring constant.

As long as one correspondence pair has been found, the translation and rotation between them can be determined simultaneously.

Point signature method computes 1-D functions to represent surface shapes [Chua 1997]. For a point **p**, a sphere with a small radius r is placed centred at **p**. The intersection curve C between the sphere and the surface is calculated, as illustrated in **Figure 2.6**(a). A plane **P** is fitted though C and a new plane **P'** which is parallel with **P** is created to go through the point **p**. The curve C is projected onto **P'** and a planar curve C' is formed. The perpendicular distances $\{d\}$ from the points on C to C' form a 1-D function with respect to the azimuth angles $\{\theta\}$ on the plane **P'**. The vector from **p** to the point which has the greatest positive projection distance is taken as the reference direction for the angles. Here the resultant 1-D function in **Figure 2.6**(c) is called point signature of the point **p**.

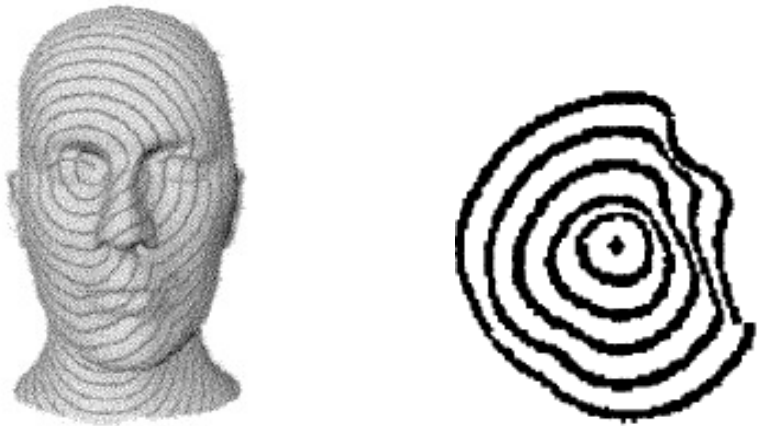


(a) (b) (c)

Figure 2.6 Creation of point signature

After all the signatures on both surfaces have been calculated, the correspondence point pairs are sought by comparing their signatures. For each pair, the transformation is decided by overlapping the interest points and aligning the orthonormal frames. Transformation parameters between the two surfaces are decided using a voting method.

Sun et al [Sun 2003] proposed another 2-D descriptor called point fingerprint. Firstly the points that result in irregular contour shapes are selected as points of interest. For each interest point **p**, a local coordinate system is defined according to the normal vector at **p**. The contours at **p** are projected on to the tangent plane **P** and form a 2-D figure, which is called a point fingerprint, as shown below.



(a) (b)

Figure 2.7 Creation of point fingerprint

In the matching stage, the contour radius variation and the normal variation in the fingerprints are compared to find the correspondence pairs.

There are also other types of geometric descriptors developed for pattern recognition and computer vision. Comprehensive surveys were given by Planitz et al [Planitz 2005] and Bustos et al [Bustos 2005].

(e) Image Based Methods

Image based methods project 3D models onto 2D images. Therefore 2D image retrieval techniques can be employed. Query interfaces were straightforward to design so that a user-supplied 2-D sketch can be input into the search algorithms [Funkhouser 2003].

The Lightfield descriptor is defined as certain image features extracted from a set of silhouettes obtained by projecting a 3D model [Chen 2003]. Cameras are located at the vertices of a dodecahedron centred at the object's centroid, completely surrounding the object. The deviation between two objects is measured by the minimum sum of the distances between all the corresponding image pairs when rotating one camera system, covering all 60 possible alignments. The image metric adopted to compare image pairs is the l_1 norm over 35 coefficients of Zernike moments and 10 Fourier coefficients obtained from the silhouettes. A very comprehensive survey on image registration can be found in [Zitová 2003].

An important application field of image registration is face recognition for decision of the identity of individuals [Heseltine 2005]. The main problem is how to distinguish the

specific characteristics of a person's photo whilst eliminating the influence of the variations of pose, illumination, and facial expression. Traditionally, researchers recognized faces based on the facial features, e.g. eyebrows, nose vertical position and width, mouth position and width, and so on. However, they are not sufficiently descriptive. Only a small group of persons can be distinguished by these features. As a consequence various new methods have been developed applying elastic bunch graphs [Wiskott 1997], curvatures [Tanaka 1998], principal components analysis [Hesher 2002], morphable models [Romdhani 2002], contours [Lee 2003], Kimmel's Eigenforms [Elad 2003] etc.

Extending the human face imaging further, medical images are used very widely to investigate disease processes and to understand normal development and ageing of organs [Hill 2001]. Registration of medical images is very challenging because of the deformation of organs and scanner-induced geometrical distortions. In order to deal with non-rigid registration, patient-related image information is usually required. Maintz and Viergever referred this kind of registration techniques as intrinsic methods and classified them into three types: landmark based methods, segmentation based methods, and voxel property based methods [Maintz 1998].

(f) Graph Based Methods

These methods evaluate the similarity between surfaces by comparing their surface topologies using a relational data structure such as a graph or a tree.

Chung et al [Chung 1997] proposed a refined version of the graph spectra based on the Laplacian of a graph,

$$L_G = \begin{cases} 1 & u = v \text{ and } d_v \neq 0 \\ -1/\sqrt{d_u d_v} & u \text{ and } v \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

where u and v are nodes of a graph G, and d_i represents the degree of a node i. The graph spectra of different graphs are compared with proper measures.

Hilaga et al [Hilaga 2001] presented an approach to describe the topology of 3D objects by a graph structure called the Reeb graph. The Reeb graph can be interpreted as information about the skeletal structure of an object. The similarity between two objects

are compared according to the topology of the Reeb graphs as well as mesh properties of the model parts associated with the corresponding graph nodes.

Sundar et al [Sundar 2003] applied a thinning algorithm on the voxelization of a solid object to obtain a thin skeleton. The matching of two skeletal graphs is performed by establishing a set of node-to-node correspondences between the graphs based on a greedy, recursive bipartite graph matching algorithm [Shokoufandeh 2001].

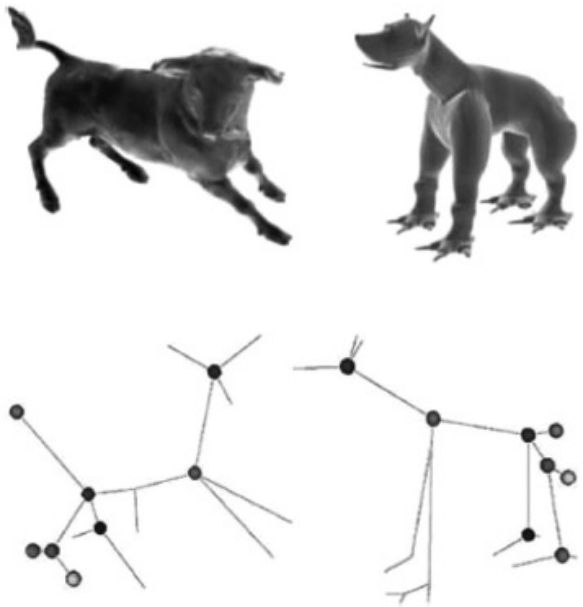


Figure 2.8 Skeletons of two models [Sundar 2003]

2.3 Final Fitting Methods

After a proper rough position is obtained, the fitting result will be refined subsequently. Different with the initial matching, requirements on the final fitting are: accuracy, stability, robustness, and efficiency.

2.3.1 Parameter-Based Algorithms

(a) Quadric Surface Fitting

Quadric surfaces are used very extensively in engineering. It has been reported that approximately 85% of manufactured objects can be well-modelled with quadric surfaces, such as sphere, cylinder, cone, paraboloid etc [Chivate 1993]. The general form of a quadric surface is represented as,

$$Q(\mathbf{x}) = Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + J = 0 \qquad (2.29)$$

The most intuitive fitting approach is to minimize the algebraic distance function,

$$\min \sum_{i=1}^{N} Q^2(\mathbf{x}_i) \qquad (2.30)$$

It is a linear least squares problem and the parameters can be solved directly from the normal equation. Evidently, there exists a trivial solution $A = B = C = D = E = F = G = H = I = J = 0$. To avoid it, various normalization methods are proposed, like J=1 [Cao 1991] or $A + B + C + D + E + F + G + H + I + J = 1$, so that Equation (2.30) becomes a minimization problem with linear constraints. However they all have singularities for some specific kinds of surfaces. The best constraint is $A^2 + B^2 + C^2 + D^2 + E^2 + F^2 + G^2 + H^2 + I^2 + J^2 = 1$. However, it will make the function very difficult to solve if using the ordinary derivative-based algorithms. A generalized eigenvector method is proposed [Taubin 1991, Petitjean 2002]. We rewrite Equation (2.29) in a matrix form,

$$\mathbf{Xp} = 0 \qquad (2.31)$$

with $\mathbf{X} = \begin{bmatrix} x_1^2 & y_1^2 & z_1^2 & x_1y_1 & x_1z_1 & y_1z_1 & x_1 & y_1 & z_1 & 1 \\ x_2^2 & y_2^2 & z_2^2 & x_2y_2 & x_2z_2 & y_2z_2 & x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^2 & y_N^2 & z_N^2 & x_Ny_N & x_Nz_N & y_Nz_N & x_N & y_N & z_N & 1 \end{bmatrix}$

and $\mathbf{p} = \begin{bmatrix} A & B & C & D & E & F & G & H & I & J \end{bmatrix}^T$.

Its normal equation is,

$$\mathbf{X}^T\mathbf{Xp} = 0$$

Evidently if one of the eigenvalues of the matrix $\mathbf{A} = \mathbf{X}^T\mathbf{X}$ vanishes, the solution $\mathbf{p}$ is the corresponding eigenvector. Otherwise $\mathbf{p}$ is the eigenvector associated with the eigenvalue which has the minimum absolute value. In fact, since $\mathbf{A}$ is a positive semi-definite matrix and all its eigenvalues are non-negative, thus $\mathbf{p}$ is the eigenvector associated with the minimum eigenvalue.

Even if the parameters in Equation (2.29) have been obtained, the geometric information is not clear yet. Thus transformation is needed to convert the equation into a standard form.

Two invariants can be defined as [Korn 1968],

$$\Delta = \begin{vmatrix} A & D/2 & F/2 & G/2 \\ D/2 & B & E/2 & H/2 \\ F/2 & E/2 & C & I/2 \\ G/2 & H/2 & I/2 & J \end{vmatrix} \tag{2.32}$$

and

$$K = \begin{vmatrix} A & D/2 & F/2 \\ D/2 & B & E/2 \\ F/2 & E/2 & C \end{vmatrix} \tag{2.33}$$

The type of the quadric surface can be determined by these two invariants:

	K≠0 Central quadric surfaces	K=0 Non-central quadric surfaces
Δ>0	Single-sheet hyperboloid	Hyperbolic paraboloid
Δ=0	Ellipsoid or dual-sheer hyperboloid	Elliptical paraboloid
Δ<0	Cone	Cylinder or plane

Table 2.1 Quadric surface types

After all the parameters have been extracted, the eigen-decomposition technique can be utilized to get the standardized form of a general quadric surface. Further details will be presented in Subsection 5.1.3.

If the specific type of a quadric surface has been known, some type-specific fitting algorithms can be employed. Type-specific fitting has more benefits in terms of occlusion and noise-insensitivity than general methods. In addition, the increased stability of the algorithm widens its scope of application to cases where the data is not strictly, say, elliptical, but needs to be minimally represented by an elliptical 'blob' [Fitzgibbon 1999, Banegas 1999].

In the previous methods, algebraic distance is involved in the error metric to assess the quality of fit. This is a linear least squares problem and commonly applied in the metrology field because of its ease of implementation. However, its definition of error distances does not coincide with measurement guidelines. The estimated fitting parameters are biased, especially in the case there exist errors in the explanatory variables [DIN 1986, Ahn 2001, Sun 2007]. Consequently researchers have developed the *orthogonal distance fitting* (or termed *geometric fitting*) method. This technique attempts to minimize the sum of the squared orthogonal distances from the measurement points to the model. It successfully overcomes the bias problem of the algebraic fitting.

Jung et al [Jung 2000] compared five algorithms of sphere fitting: linear least squares, non-linear least squares, minimum zone, four-point, and sphere fit by error curve analysis. The nonlinear least squares method solves the parameters by,

$$\min \sum_{i=1}^{N} \left\{ \left[(x_i - a)^2 + (y_i - b)^2 + (z_i - c)^2 \right]^{1/2} - r \right\}^2 \tag{2.34}$$

This equation takes the squared orthogonal distances in the error metric. It is proved that the nonlinear least squares method is the best option for spherical surfaces with random irregularities. The minimum zone algorithm is the best when the surface irregularity is skewed or rotationally symmetric [Jung 2000].

Forbes [Forbes 1990] performed parameterization on sphere, cylinder, and cone accordingly. The orthogonal distance can be represented in a closed form using location parameters. Lukács et al [Lukács 1998] approximated the orthogonal distance with a faithful function. For example, Equation (2.34) can be approximated as,

$$\min \sum_{i=1}^{N} \left[\frac{(x_i - a)^2 + (y_i - b)^2 + (z_i - c)^2 - r^2}{2r} \right]^2 \tag{2.35}$$

For general quadric surfaces, Cao et al [Cao 1994] proposed an approximate orthogonal distance fitting approach. They calculated the distances from a point to the surface along several fixed directions. The minimum distance is regarded as the real orthogonal distance. The parameters are optimized iteratively and converge to a very good result.

Instead of minimizing the residual error, Dai et al optimized the shape parameters directly [Dai 1998]. For hyperboloids and ellipsoids, the matching error is presented as,

$$E = \frac{1}{3} \left(\frac{|\hat{a} - a|}{a} + \frac{|\hat{b} - b|}{b} + \frac{|\hat{c} - c|}{c} \right) \tag{2.36}$$

where $\hat{a}, \hat{b}, \hat{c}$ are the fitted parameters and a, b and c are the real lengths of the three principal axes. In this method, sampled points located within three special regions are used to estimate a, b and c, each region corresponding to one parameter respectively.

(b) Aspheric Surface Fitting

Researchers have also paid attention to aspheric surfaces, especially in the field of optics manufacture. Aspheric lenses show notable superiority over conventional spherical lenses in that a multiple element lens can be replaced by a single aspheric lens. Aspheric surfaces can be represented with this function [ISO 10110-12:2007],

$$z = f(r) = \frac{\frac{r^2}{R}}{1+\left(1-(1+k)\frac{r^2}{R^2}\right)^{1/2}} + A_4 r^4 + A_6 r^6 + A_8 r^8 + A_{10} r^{10} \qquad (2.37)$$

with $r = (x^2 + y^2)^{1/2}$.

Here R is the radius of curvature of the underlying sphere. k is the conic constant determining the nature of the basic (second order) deviation from sphericity: when $k>0$ it is an oblate spheroid; $k=0$, a sphere; $-1 < k < 0$, a prolate spheroid; $k = -1$, a paraboloid and $k < -1$, hyperboloid. $\{A_i\}$ are the magnitudes of any higher order deviations from sphericity.

Because of the fractionality and high order terms in the equation, a non-standard form of Equation (2.37) will be very complicated. It is better to pre-process the measurement data and align it to a standard position, and derive the shape parameters thereafter.

Scott [Scott 2002] firstly corrected the measurement data for the geometry of the stylus tip using an areal morphological erosion filter, and then carried out a pitch-yaw rotation and 3D translation to move the corrected data to the standard position. The intrinsic characteristics (R, k etc) are fitted by minimizing the squared algebraic distances with the Gauss-Newton method.

Hill et al [Hill 2002] presented a two-stage pre-processing technique using the contour-line fit and local axis search to evaluate the orientation and position parameters respectively. After pre-processing and alignment, a least-squares technique is adopted to find the best-fitted parameters in Equation (2.37).

2.3.2 Iterative Closest Point Method

ICP (*Iterative Closest Point*, though *Iterative Corresponding Point* is a better expansion [Rusinkiewicz 2001]) is a most widely used final matching algorithm. It was

initially adopted by Horn [Horn 1987] and popularized by Besl and McKay [Besl 1992] and Chen and Medioni [Chen 1992]. It is able to register several types of geometric data such as point sets, triangle sets, implicit surfaces or parametric surfaces.

Given an initial relative position between two point sets, ICP iteratively refines the transform by repeatedly generating pairs of correspondences on the point sets and minimizing the error metric e.g. the sum of squared distances between the correspondence point pairs.

Plenty of variants and improvements of ICP have been developed. They contribute to different stages of the matching procedure. These techniques are classified into the following five groups.

(a) Searching Closest-Point Pairs

Usually the closest template point of each measurement datum is taken as the correspondence. If directly searching the closest points, the computational complexity of establishing the correspondences is $O(MN)$, where M and N are the point numbers of the template and measurement data respectively. It is demonstrated that more than 90% of the computation is spent on closest-point searching [Jost 2002].

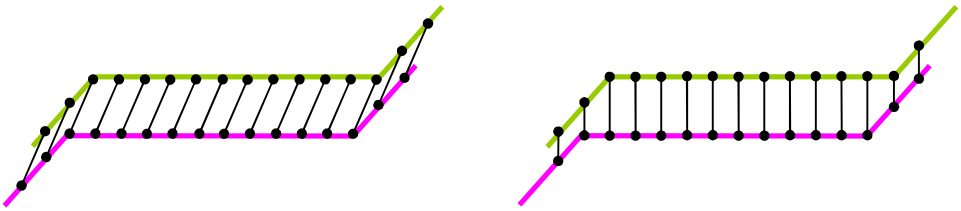
In order to accelerate the matching procedure, the first option is to sample fewer points from the given point sets. The points can be sampled evenly on the whole surface [Turk 1994], or selected randomly [Masuda 1996]. Sometimes it is better to choose the points with high intensity gradient [Weik 1997] or the points in smooth areas [Chen 1992].

Another procedure is to utilize some efficient searching techniques. Several data structures have been developed to speed up the closest point searching, such as the multidimensional binary search tree (the k-D tree) [Bentley 1990], geometric cashing [Simon 1996], Elias method [Greenspan 2000], triangle inequality [Greenspan 2001] etc. The k-D tree will be introduced in detail in Subsection 6.1.1. Employing fast searching techniques, the computational complexity can be reduced down to $O(N \log M)$.

(b) Other Correspondence Relations

Initially, most of the authors took the nearest points as correspondences; however, it may lead to false matching. Here an example is given. The correct correspondences

between two surfaces are presented in **Figure 2.9**(a), but if taking the closest point, false correspondences will be caused, see **Figure 2.9**(b).



(a) (b)

Figure 2.9 False correspondence problem of ICP

To avoid false correspondences, another two forms of correspondences, point-to-plane and point-to-projection are proposed [Park 2003]. In the three correspondence approaches, the closest point (point-to-point) method is most sensitive to noise and outliers, and generates the most false correspondences. By contrast, the-point-to-plane technique is the most accurate one. The point-to-projection method is the fastest, because it is performed in constant time and no searching work is required. However, it is not as accurate as the other two techniques.

Park and Subbarao [Park 2003] proposed a new method called the contractive projective point (CPP) technique which combines the advantages of the point-to-plane and point-to-projection methods.

Suppose the normal vector at an arbitrary measurement point $\mathbf{p}_0$ is $\hat{\mathbf{p}}$ and the back projection of $\mathbf{p}_0$ onto a 2D image plane $\mathbf{I}_Q$ is $\mathbf{p}_q$. Forward project $\mathbf{p}_0$ on the template surface $\mathbf{Q}$, and calculate the perpendicular foot $\mathbf{p}_1$ of the projection point $\mathbf{q}_{p0}$ onto $\hat{\mathbf{p}}$. Repeat this procedure k times until the orthogonal projection point $\mathbf{p}_k$ sufficiently achieves the surface $\mathbf{Q}$.

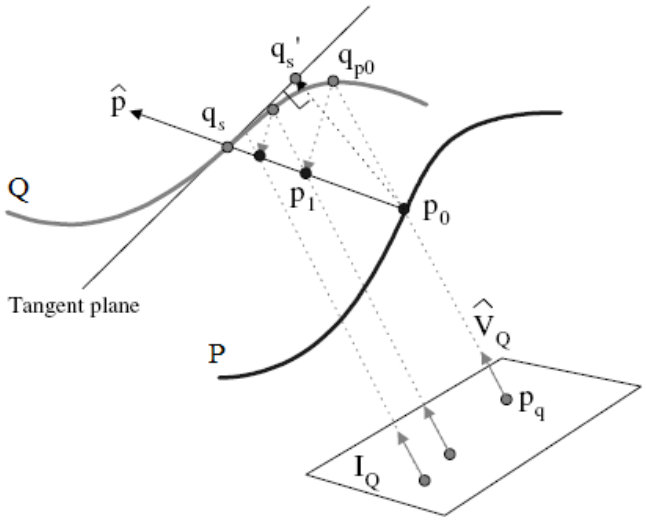


Figure 2.10 CPP method to find correspondences

(c) Improving Robustness and Stability

The sum of the squared distances is commonly adopted as an error metric. However, it is not robust against outliers [Meer 1991]. In order to improve the matching accuracy, some 'bad' matching pairs can be rejected, for instance

- correspondence points more than a given distance apart [Rusinkiewicz 2001],
- the worst *n*% of pairs based on some metric, e.g. distance (this method is also called the trimmed ICP [Chetverikov 2005]),
- point pairs that are not consistent with neighbouring pairs [Dorai 1998],
- boundary point pairs [Turk 1994],

and so on.

For the remaining point pairs, weighting can be assigned either based on the distance [Godin 1994] or the relative angle between the normal vectors [Rusinkiewicz 2001]. Additionally, some robust estimators, such as the least median squares, can also be adopted [Meer 1991].

In order to make the matching result more reliable, other features of the models or images can also be involved in the error metric e.g. reflectance, colour, temperature [Akca 2005], invariant features [Sharp 2002], measurement error properties [Okatani 2002] etc.

(d) Calculating Motion Parameters

The error metrics are nonlinear with respect to the motion parameters; hence some recursive algorithms such as the Newton or Gauss-Newton algorithms shall be employed. Researchers have also developed some closed-form techniques for this specific purpose. Eggert et al [Eggert 1997] compared four closed form methods quantitatively: singular value decomposition (SVD), orthogonal matrices (OM), unit quaternion (UQ) and dual unit quaternion (DQ). The qualitative rating result is shown in Table 2.2 (1 is the best and 4 the worst) [Eggert 1997].

Method	3D accuracy		2D stability			1D stability			0D stability			Efficiency	
	ideal	noise	ideal	i-noi	a-noi	ideal	i-noi	a-noi	ideal	i-noi	a-noi	small N	large *N*
SVD	1	1	1	1	1	2	2	2	3	1	1	2	2
OM	3	1	4	4	4	1	1	1	1	1	1	1	4
UQ	2	1	2	1	1	3	3	3	1	1	1	2	3
DQ	4	1	3	1	1	4	4	4	4	4	4	4	1

ideal denotes ideal correspondence points without noise. *i-noi* and *a-noi* refer to isotropic and anisotropic noise respectively.

Table 2.2 Qualitative comparison of the four closed form algorithms

In this thesis we want to match 3D surfaces, thus the 3D matching accuracy and efficiency are of our interest. Therefore, the SVD algorithm is the best choice for our purpose.

The ICP method exhibits linear convergence [Pottmann 2006]. In order to accelerate the convergence rate, Besl and McKay [Besl 1992] performed extrapolation onto the transformation parameters based on the residual, so that the iteration number can be decreased. The main problem of extrapolation is overshoot, which will lead to a local minimum [Jost 2002]. Therefore the update will be ignored or reduced if the mean squared error is worse than that of the last iteration.

(e) Overcoming the Local Minimum Problem

The ICP method is prone to being trapped at a local minimum because of the non-convexity of the cost function with respect to the motion parameters.

In order to handle this problem, Simon [Simon 1996] started the optimization with several perturbations in the initial conditions, and then selected the best result. Blais and Levine [Blais 1995] adopted stochastic search using simulated annealing.

Boughorbel et al [Boughorbel 2004] proposed a method called the Gaussian field. Instead of using the sum of squared distances, they calculated an optimal transformation to maximize,

$$\max_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{N} \exp\left\{-\left\|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\right\|\right\} \tag{2.38}$$

It is demonstrated that the cost function is always differentiable and convex in a large neighbourhood, so that the convergence region is greatly enlarged.

2.4 Numerical Issues: Stability and Robustness

2.4.1 Numerical Stability of the Solution

In previous sections, the following linear system appears very frequently,

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{2.39}$$

In the equation, $\mathbf{A} \in \Re^{N\times M}$ is a non-singular design matrix and $\mathbf{x} \in \Re^{M\times 1}$ is the least squares solution. In practice, usually the data number N is very large and the system is an over-determined problem, i.e. $N > M$.

Now we investigate the stability of the solution $\mathbf{x}$ against perturbations in $\mathbf{A}$ and $\mathbf{b}$ [Golub 1996].

Suppose the perturbations in $\mathbf{A}$ and $\mathbf{b}$ are $\delta\mathbf{A}$ and $\delta\mathbf{b}$ respectively, and the new solution of the perturbed system is $\hat{\mathbf{x}}$, i.e.

$$\hat{\mathbf{x}} = \arg\min \left\|(\mathbf{A} + \delta\mathbf{A})\hat{\mathbf{x}} - (\mathbf{b} + \delta\mathbf{b})\right\| \tag{2.40}$$

Set $\varepsilon = \max\left\{\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}, \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}\right\}$,

and $\sin(\theta) = \frac{\left\|\mathbf{b} - \mathbf{A}\mathbf{x}\right\|}{\left\|\mathbf{b}\right\|}$

then, $\frac{\left\|\hat{\mathbf{x}} - \mathbf{x}\right\|}{\left\|\mathbf{x}\right\|} \le \varepsilon\left\{\frac{2\kappa_2(\mathbf{A})}{\cos(\theta)} + \tan(\theta)\kappa_2(\mathbf{A})^2\right\} + O(\varepsilon^2)$, (2.41)

where $\kappa_2(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{\dagger}\|$ is the l_2 norm condition number of the matrix $\mathbf{A}$ and $\mathbf{A}^{\dagger}$ is the pseudoinverse of $\mathbf{A}$.

If introducing a perturbation $\mathbf{E} \in \Re^{M \times M}$ into the normal equation, i.e.

$$(\mathbf{A}^T\mathbf{A} + \mathbf{E})\hat{\mathbf{x}} = \mathbf{A}^T\mathbf{b} \tag{2.42}$$

Then $$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|} \cong \frac{\|\mathbf{E}\|}{\|\mathbf{A}^T\mathbf{A}\|}\kappa_2(\mathbf{A}^T\mathbf{A}) = \frac{\|\mathbf{E}\|}{\|\mathbf{A}^T\mathbf{A}\|}\kappa_2(\mathbf{A})^2 \tag{2.43}$$

That is to say, the system stability is determined by the condition number of the matrix $\mathbf{A}$. If the matrix is rank-deficient or ill-posed, the error in the solution maybe very large even there is only a small perturbation in $\mathbf{A}$ or $\mathbf{b}$. We can also see the normal equation is less stable than the original one. When the size *M* and *N* are in the same order or the design matrix $\mathbf{A}$ is ill-posed, direct decomposition of $\mathbf{A}$ is recommended, although inversion of the normal equation is more efficient (its complexity is $\frac{NM^2}{2} + \frac{M^3}{6}$).

In order to overcome the ill-conditioning problem, some stabilized inversion techniques have been developed.

(a) Rank-Revealing QR Decomposition

A popular decomposition method is the QR decomposition,

$$\mathbf{A} = \mathbf{QR} \tag{2.44}$$

where $\mathbf{Q} \in \Re^{N \times N}$ is a unitary matrix and $\mathbf{R} \in \Re^{N \times M}$ is an upper triangular matrix.

The complexity of the decomposition is $NM^2 - \frac{M^3}{3}$ if using the Householder algorithm [Householder 1958].

We introduce a permutation matrix $\mathbf{\Pi}$ satisfying,

$$\mathbf{A\Pi} = \mathbf{Q}\begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ 0 & \mathbf{R}_{22} \end{bmatrix} \begin{matrix} r \\ N-r \end{matrix} \tag{2.45}$$

If $\mathbf{R}_{22} = 0$, we can get $r = rank(\mathbf{A})$. In the case of rank-deficiency, the orthogonalization process will be stopped when $\mathbf{R}_{22}$ is sufficiently small [Hong 1992].

Denoting $\mathbf{\Pi}^T\mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} \begin{matrix} r \\ M-r \end{matrix}$ and $\mathbf{Q}^T\mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} \begin{matrix} r \\ N-r \end{matrix}$, then the new solution will be,

$$\hat{\mathbf{x}} = \mathbf{\Pi}^T \begin{bmatrix} \mathbf{R}_{11}^{-1}(\mathbf{c} - \mathbf{R}_{12}\mathbf{z}) \\ \mathbf{z} \end{bmatrix} \tag{2.46}$$

If we set $\mathbf{z} = 0$, a basic solution is obtained,

$$\mathbf{x}_B = \mathbf{\Pi}^T \begin{bmatrix} \mathbf{R}_{11}^{-1}\mathbf{c} \\ 0 \end{bmatrix} \tag{2.47}$$

This method is called the *Rank-revealing QR decomposition* [Björck 1996] and its complexity is $2NMr - r^2(M+N) + 2r^3/3$ [Golub 1996].

(b) Truncated SVD

The SVD (*Singular Value Decomposition*) of a matrix $\mathbf{A} \in \Re^{N \times M}$ is defined as [Björck 1996],

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{2.48}$$

where $\mathbf{U} \in \Re^{N \times N}$ and $\mathbf{V} \in \Re^{M \times M}$ are two unitary matrices and $\mathbf{S} \in \Re^{N \times M}$ is a diagonal matrix,

$$\mathbf{S} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_M \\ & 0 & & \end{bmatrix} \begin{matrix} \\ \\ \\ \\ N-M \end{matrix} \tag{2.49}$$

$\sigma_1 \geq \sigma_1 \geq \ldots \geq \sigma_M \geq 0$ are called singular values of the matrix $\mathbf{A}$.

The pseudo-inverse matrix of $\mathbf{A}$ is,

$$\mathbf{A}^{-1} = \mathbf{V}\mathbf{S}'\mathbf{U}^T \tag{2.50}$$

where $\mathbf{S} = \begin{bmatrix} \sigma_1' & & & \\ & \sigma_2' & & \\ & & \ddots & 0 \\ & & & \sigma_M' \end{bmatrix}$ with $\sigma_i' = 1/\sigma_i, i = 1, \cdots, M$.

The l_2 norm condition number of $\mathbf{A}$ is $\kappa_2(A) = \sigma_1/\sigma_M$. SVD is particularly useful because it permits us to quantify the notion of near rank-deficiency. In fact, it is the most numerically reliable and the only completely reliable method of calculating the inverse of

a rank-deficient matrix. However, it is very computationally expensive and its number of flops is $2NM^2 - \frac{2}{3}M^3$ [Golub 1996].

If **A** is rank-deficient, the minimum singular value σ_M' will be rather small. The Truncated SVD calculates the new singular values by,

$$\sigma_i' = \begin{cases} 1/\sigma_i & \sigma_i > \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{2.51}$$

In the equation, ε is a user-set criterion.

2.4.2 Robustness of the Solution

Before introducing robust regression techniques, we firstly define two critically important terminologies.

Robustness — A statistical procedure is described as robust if it is not very sensitive to departure from the assumptions on which it depends [Rey 1983].

Breakdown point — The breakdown point of an estimator is the smallest fraction or percentage of discrepant data (i.e. outliers or data grouped at the extreme end of the tail of the distribution) that the estimator can tolerate without producing an arbitrary result [Anderson 2007]. It is a common measure of the robustness of an estimator.

In previous sections, most solutions are based on the least squares method,

$$E = \sum_{i=1}^{N} \rho(r_i) = \sum_{i=1}^{N} r_i^2 \tag{2.52}$$

In the equation, ρ is called *cost function* (also *loss function*) and r_i is the residual error associated with the datum $\mathbf{x}_i = [x_i, y_i]^T$.

The least squares method is applied very extensively because of its ease of implementation. More importantly, it is unbiased when the measurement error obeys Gaussian distribution (Normal distribution), as asserted by the Gauss-Markoff theorem [Björck 1996].

However, the assumption of normality does not always hold true. In practice, deviations such as gross errors (outliers), rounding and grouping errors, and departure from an assumed sample distribution will take place because of defects, improper

operation, external influence [Nasraoui 2002] and some functional properties, e.g. lubrication and friction.

Gross errors are data severely deviating from the pattern set by the majority of the data. They are the most dangerous type of errors and one single outlier could make the least squares fitting result break down. It is worth noting that outliers do not necessarily mean bad data or wrong data. Outliers should never be blindly discarded. They are usually analyzed separately with the clean data [Olive 2007].

Rounding and grouping errors result from the inherent inaccuracy in the collection and recording of data which is usually rounded, grouped, or even coarsely classified.

The departure from an assumed model means that real data are probable to deviate from the often assumed normal distribution. The actual distribution may be skewed or with a long tail. Estimators are required to be consistent with the error distributions.

(a) M-Estimators

M-estimators are generalizations of the usual maximum likelihood estimates. They are initially proposed by Peter J. Huber [Huber 1964]. M-estimators will be very robust when formulated properly and more efficient than other robust regression methods for large samples. The cost function must be strictly convex to make sure the uniqueness of the solution.

Huber [Huber 1964] proposed a robust estimator, now called the *Huber estimator*,

$$\rho(r;\theta)=\begin{cases} r^2/2 & |r|\le c \\ c|r|-c^2/2 & |r|>c \end{cases} \tag{2.53}$$

The performance of this estimator relies on the value of the threshold c. When $c\to\infty$, it reduces to the least squares estimator; as $c\to 0$, a l_1 norm estimator is obtained. In practice, $c=2MAD$ is recommended, where MAD is the median absolute deviation.

Mosteller and Tukey [Mosteller 1977] proposed the *biweight estimator* (also called Tukey's bisquare estimator),

$$\rho(r;\theta)=\begin{cases} \dfrac{c^2}{6}\left\{1-\left[1-\left(\dfrac{r}{c}\right)^2\right]^3\right\} & |r|\le c \\ c^2/6 & |r|>c \end{cases} \tag{2.54}$$

$c=7MAD$ is recommended. The biweight and the Huber estimators behave similarly for most of the distribution, except in the very centre and at the extreme tails of the distribution. For larger errors, the bisquares estimator tapers off.

The *fair estimator* is defined as [Rey 1983],

$$\rho(r;\theta) = c^2\left[\frac{|r|}{c} - \log\left(1+\frac{|r|}{c}\right)\right] \tag{2.55}$$

with $c = 2MAD$. It is three-ordered differentiable everywhere. Its performance is between the least squares and the least absolute value regression.

Compared with the least squares method, these estimators pay less attention to the gross error. In fact, they can be regarded as the iterative reweighted least squares,

$$\rho(r;\theta) = wr^2 \tag{2.56}$$

The weighting parameters are assigned inversely proportional to the residual errors, i.e. smaller weighting parameters are assigned onto larger residuals and vice versa. Various reweighted least squares techniques have been proposed and some relevant reviews can be found in [Heiberger 1992, Zhang 1997].

Additionally, many new estimators have been developed based on the M-estimators, e.g. GM-estimators, and MM-estimators [Anderson 2007].

(b) L-Estimators

L-estimators are linear combinations of order statistics and firstly proposed by Lloyds [Rey 1983, Lloyd 1952]. The *k*-th order statistic of a statistical sample is equal to its *k*-th smallest value.

The first L-estimator is the least absolute values (LAV), also known as l_1 norm, which intends to minimize the sum of the absolute deviations. This will be introduced in the l_p norm estimators later.

Another famous L-estimator is the least median of squares (LMS) proposed by Rousseeuw [Rousseeuw 1984],

$$\theta = \arg\min Median(r_i^2) \tag{2.57}$$

It is resistant to outliers and the resulting *breakdown point* may be as high as 0.5, which is the highest possible value of all regression techniques. However, it is very complex to solve and two times slower than the ordinary least squares [Anderson 2007].

Rousseeuw developed a Least Trimmed Squares (LTS) regression method [Rousseeuw 1984], which minimizes the sum of the trimmed squared residuals,

$$E = \sum_{i=1}^{q} r_i^2 \tag{2.58}$$

where $q = N(1-\alpha)+1$ is the number of data points included in the error metric and α is the proportion of trimming. The above case is sometimes called the α-least trimmed squares. Its breakdown point is α [Maronna 2006] and it is more than 10 times slower than ordinary least squares [Anderson 2007]. It is so slow that LTS is not commonly applied in practice.

(c) R-Estimators

In *R-estimators*, the residuals are weighted based on their ranks [Jaeckel 1972],

$$\theta = \arg\min \sum_{i=1}^{N} a_N(R_i) r_i \tag{2.59}$$

where R_i is the rank of the *i*-th residual in $\{r_1, r_2, \cdots, r_N\}$ and a_N is a nondecreasing score function satisfying $\sum_{i=1}^{N} a_N(i) = 0$. Many forms have been proposed for the score function, such as [Anderson 2007],

- Wilcoxon score $a_N(i) = i - \frac{N+1}{2}$
- Median score $a_N(i) = \sin\left(i - \frac{N+1}{2}\right)$
- Van der Waerden score $a_N(i) = \Phi^{-1}\left(\frac{i}{n+1}\right)$, where Φ is the normal probability density function.

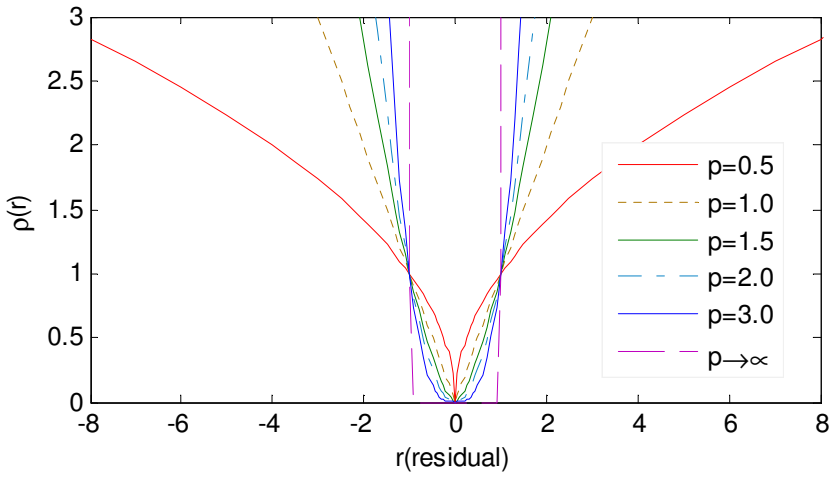
An advantage of R-estimators over others is that they are scale invariant. But the choice of the optimal score function is not clear. Additionally, they are not easy to solve and their breakdown points never achieve more than 0.20.

(d) l_p Norm Estimators

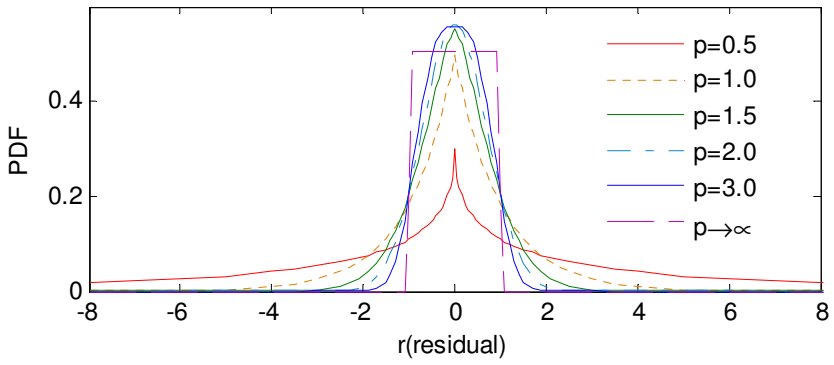
The ordinary least squares regression can be extended into the l_p norm [Gonin 1989],

$$E_p = \frac{1}{N}\left(\sum_{i=1}^{N} | r_i |^p\right)^{1/p}, \quad p > 0 \tag{2.60}$$

The cost function $\rho_p = |r|^p$ and the corresponding probability density function (PDF) $PDF_p = \frac{\exp\{-| r |^p\}}{A_p}$ for different p values are depicted in **Figure 2.11**.



(a) Cost functions



(b) Probability density functions

Figure 2.11 Comparison of different p values

It can be seen the l_p norm with a smaller p value assigns a smaller cost function for the wild points, therefore being more robust against outliers. It is good at dealing with long

tailed errors. If the errors are uniformly distributed or with a sharply truncated PDF, $p = \infty$ will be a good choice.

When p<1, the objective function is concave, thus not of our interest. If the errors are skewly distributed, it may be useful [Ekblom 1974].

If $p = 1$, it is the well known least absolute values (also called minsummod [Cox 1993]) mentioned above in L-estimators. The object function is not strictly convex, thus the solution cannot be guaranteed to be unique. Furthermore, it is not differentiable at r=0, so that the l_1 norm cannot be directly solved based on the derivatives. A common approach is to transform it into a minimization problem with inequality constraints and solve it with linear programming techniques [Barrodale 1973]. Based on this algorithm, lots of variants have been developed [Lei 2002].

However, linear programming methods are not straightforward to use. In order to improve the calculation efficiency, some approximates of the l_1 norm have been proposed, such as,

$\rho(r) = \sqrt{r^2 + \varepsilon^2}$ [El-Attar 1979]

and $\rho(r) = \begin{cases} r^2 / 2 & | r | < \varepsilon \\ \varepsilon(|r| - \varepsilon / 2) & \text{otherwise} \end{cases}$ [Madsen 1990].

When the threshold $\varepsilon \to 0$, the approximations approach the l_1 norm.

The l_p norm with $1 < p < 2$ goes to the category of M-estimators. They are smooth and differentiable with respect to the residual r. Most of the authors handled this problem using Newton or quasi-Newton algorithms [Gonin 1989]. Cooper and Mason transformed it into a reweighed least squares problem [Cooper 2004].

At the k-th iteration, the weighting is assigned as,

$$w_i^{(k)} = \left(\frac{w_i^{(k-1)}}{| r_i^{(k-1)} |} \right)^{\frac{2-p}{3-p}} \tag{2.61}$$

In the equation, $w_i^{(k-1)}$ and $r_i^{(k-1)}$ are the weighting and residual error at the previous iteration. When k becomes larger, $\frac{| r_i^{(k)} |}{| r_i^{(k-1)} |} \to 1$ and $\frac{w_i^{(k)}}{w_i^{(k-1)}} \to 1$, so that $w_i^{(k)} \to \left| r_i^{(k)} \right|^{p-2}$ i.e. $w_i^{(k)} (r_i^{(k)})^2 \to \left| r_i^{(k)} \right|^p$. To avoid infinite or very large weights caused from Equation (2.61),

an upper bound is set for them. After all the weightings are worked out, they are normalized subsequently. At the first iteration, the weightings are initialized as $w_i^{(0)} = 1/N$, here N is the number of data points.

p=2 is the ordinary least squares problem. More details will not be presented here.

If $p > 2$, the l_p norm is even less robust that the least squares. When $p \to \infty$, it is the Chebyshev norm,

$$\min_{\theta} \max_{i} |r_i| \tag{2.62}$$

It works well for the uniformly distributed error. In mechanical engineering, form errors are defined based on the l_∞ norm, like out-of-sphericity, out-of-cylindricity, and out-of-flatness. Researchers have developed various methods to assess form errors of workpieces, such as minimum zone [ISO 4291:1985], support vector machine [Balakrishna 2008], genetic algorithm [Lai 2000], or computational geometric techniques [Samuel 1999]. Concerning general mathematical l_∞ optimization problems, most existing algorithms are based on the linear programming due to the discontinuity of such problems [Gonin 1989, Lei 2002]. Lawson proposed to transform the l_∞ problem into iterative reweighted least squares [Lawson 1961]. Rice and Usow [Rice 1968] generalized it into $p > 2$. At the k-th iteration, the weighting is calculated as,

$$w_i^{(k)} = \left(w_i^{(k-1)} \left| r_i^{(k-1)} \right| \right)^{\frac{p-1}{p-2}} \tag{2.63}$$

Analogues to the circumstance of 1<p<2, $w_i^{(k)} (r_i^{(k)})^2 \to \left| r_i^{(k)} \right|^p$ when k becomes larger.

There are also many other kinds of estimators, such as W-estimators, S-estimators etc [Maronna 2006, Nasraoui 2002].

As summary, the relative performance of different robust estimators is listed [Anderson 2007].

Estimator	Breakdown Point	Bounded Influence	Asymptotic Efficiency
Ordinary Least Squares	0	No	100
Least Absolute Value	0	Yes	64
Least Median Squares	0.5	Yes	37
Least Trimmed Squares	α	Yes	8
Least Trimmed Median	0.5	Yes	66
Bounded R-estimator	<0.2	Yes	90
M-estimator(Huber, Biweight)	0	No	95
GM-estimator (Mallows, Schweppe)	$1/(p+1)$	Yes	95
GM-estimator (Schweppe one-step estimator)	0.5	Yes	95
S-estimator	0.5	Yes	33
GS-estimator	0.5	Yes	67
MM-estimator	0.5	Yes	75
Generalized estimator	0.5	Yes	95

Table 2.3 Comparison of various estimators

2.5 Summary

Sometimes a reference template needs to be reconstructed into a continuous function if it is provided as a discrete point set. Existing reconstruction techniques are reviewed for regular-lattice and scattered points respectively.

If the given points are distributed regularly, but not exactly located in a grid format, the surface cannot be directly interpolated based on the coordinates using tensor product techniques. Hence the coordinates will be transformed into a parametric space and reconstructed using splines, such as B-splines.

For scattered data, most of the existing reconstruction methods attempt to interpolate a point according to its neighbourhood. The shape of the interpolated surface may not be consistent with the target surface and the accuracy is not satisfactory.

In order to make the fitted result more reliable, the whole fitting procedure is divided into two stages. Rough matching is performed beforehand to supply an approximate relative position between the data and the design template. Various methods have been developed in different research fields. Among them, generalized signatures are the most promising techniques. They represent the shape of a surface with figures or curves, and sufficient information can be involved. But most of them are burdensome to be

constructed. Hence a new descriptive and easy-to-calculate generalized signature needs to be developed.

Given a good initial solution, refinement follows to improve the fitting accuracy. The Iterative Closest Point method is most widely adopted to match two surfaces that are given as discrete point sets. It suffers from problems of high computational cost and slow convergence rate.

On the other hand, derivative-based algorithms can be adopted when a continuous representation is supplied for the design template. The shape parameters can be derived from the measurement data if they exist.

2.6 References

Abramowitz, M., Stegun, I. A. Editors. 1965 *Handbook of Mathematical Functions: with Formulas Graphs, and Mathematical Tables*. Dover, New York.

Ahn, S. J., Rauh, W. and Warnecke, H. J. 2001 Least-squares orthogonal distances fitting of circle sphere, ellipse, hyperbola and parabola. *Patt Recog*. 34(12): 2283-2303

Akca, D. 2005 Registration of point clouds using range and intensity information. *Int Workshop on Recording, Modeling& Visualization of Cultural Heritage*.115-126

Alexa, M. 2005 *Survey Acquisition and Reconstruction*. AIM@SHAPE State-of-the-Art-Report

Anderson, R. 2007 *Modern Methods for Robust Regression*. SAGE Inc

Ankerst, M., Kastenmüller, G., Keiegel, H. P. and Seidl, T. 1999 3D shape histograms for similarity search and classification in spatial databases. *Proc 6th Int Symp on Advances in Spatial Database*, Springer-Verlag, London, UK. 207-226

Bajaj, C. L., Xu, G., Holt, R. J. and Netravali, A. N. 2003 NURBS Approximation of A-splines and A-patches. *Int J Comput Geom and Appl*. 13(5): 359-390

Balakrishna, P., Raman, S., Trafalis, T. B. and Santosa, B. 2008 Support vector regression for determining the minimum zone sphericity. *The Int J Adv Manuf Technol*. 35(9-10): 916-923

Banegas, F., Michelucci, D., Roelens, M. and Jaeger, M. 1999 Automatic extraction of significant features from 3D point clouds by ellipsoidal skeleton. *Proc of Int Conf on Visual Computing* 58-67

Barequet, G. and Sharir, M. 1999 Partial surface matching by using directed footprints. *Comput Geom*. 12(1-2): 45-62

Barker, R. M, Cox, M. G., Forbes, A. B. and Harris, P. M. 2004 *Discrete Modelling and Experimental Data Analysis*. Ver 2. NPL Report

Barrodale, I. and Roberts, F. D. K. 1973 An improved algorithm for discrete l_1 linear approximation. *SIAM J of Numer Anal*.10(5): 839-848

Bentley, J. L. 1990 K-D trees for semidynamic point sets. *Proc of the 6th Annual Symp on Comp Geom*.187-197

Besl, P. J. and McKay, N. D. 1992 A method for registration of 3-D shapes. *Trans Patt Anal and Mach Intell* 14(2):239-256

Bézier, P. E. 1972 *Numerical Control: Mathematics and Applications*. New York. John Wiley

Björck, Å. 1996 *Numerical Methods for Least Squares Problems*. SIAM

Blais, G. and Levine, M. 1995 Registering multiview range data to create 3D computer objects. *Trans Patt Anal and Mach Intell*.17(8): 820-824

Boughorbel, F., Koschan, A., Abidi, B., and Abodi, M. 2004 Gaussian fields: a new criterion for 3D rigid registration. *Patt Recog*. 37(7): 1567-1571

Burnett, D. S. 1987 *Finite Element Analysis-From Concepts to Applications*. Addison-Wesley-Publication Company Inc

Bustos, B., Keim, D. A., Saupe, D., Scherck, T., and Vranić, D. V. 2005 Feature-based similarity search in 3D object databases. *ACM Comput Surveys*.37(4): 345-387

Campbell, R. J. and Flynn, P. J. 2001 A survey of free-form object representation and recognition techniques. *Comput Vis and Image Underst*. 81(2):166-210

Cao X. and Shrikhande, N. 1991 Quadric surface fitting for sparse range data. *Proc IEEE/SMC Int Conf on Sys, Man and Cybernetics*. 123-128

Cao, X., Shrikhande, N. and Hu, G. 1994 Approximate orthogonal distance regression method for fitting quadric surfaces to range data. *Patt Recog Lett*. 15(8): 781-796

Cazals F. and Giesen, J. 2004 *Delaunay Triangulation Based Surface Reconstruction: Ideas and Algorithms*. INRIA Technical Report 5393

Chen, C. S., Hung, Y. P and Wu, J. L. 2001 Combining morphological feature extraction and geometric hashing for three-dimensional object recognition using range images. *J Infor Sci and Eng.* 17:247-369

Chen, D. Y., Tian, X. P., Shen, Y. T., and Ouhyoung, M. 2003 On visual similarity based 3D model retrieval. *Computer Graphics Forum*. 22(3): 223-232

Chen Y. and Medioni, G. 1992 Object modelling by registration of multiple range images. *Image and Vision Computing*. 10(3): 145-155

Chetverikov, D., Stepanov, D. and Krsek, P. 2005 Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*. 23(3): 299-309

Cheung, C. F., Li, H. F. Kong, L. B. et al. 2006 Measuring ultra-precision freeform surfaces using a robust form characterization method. *Meas Sci and Technol*. 17(3): 488-494

Chivate, P. N. and Jablokow, A. G. 1993 Solid-model generation form measured point data. *Computer Aided Design*. 25(9): 587-600

Chua, C. S. and Jarvis, R. 1997 Point signatures: a new representation for 3D object recognition, *Int J. Comput. Vision*. 25 (1): 63–85

Chung, F. R. 1997 *Spectral Graph Theory*. American Mathematical Society

Cooper, P. and Mason, J. C. 2004 Rational and l_p approximations-renovating existing algorithms. *Proc Conf on Math Methods for Curves and Surf*. Tromso, Norway

Corney, J., Rea, H., Clark, D. et al. 2002 Coarse filters for shape matching. *IEEE Comp Graphics and Appl*. 22(3): 65-74

Cox, M. G. 1972 The numerical evaluation of B-splines. *J Inst Maths Appl* 10(2):134-149

Cox, M. G. 1993 Survey of numerical methods and metrology applications: discrete processes. In Ciarlini, P., Cox, M. G., Pavese, F. and Richter, D. Editors. *Adv Math Tools in Metrol.* World Scientific. Singapore. 1-21

Dai, M., and Newman, T. S. 1998 *Huperbolic and Parabolic Quadric Surface Fitting Algorithms-Comparison between the Least Squares Approach and the Parameter Optimization Approach.* Technical Report TR-UAH-CS-1998-02, University of Alabama in Huntsville

de Boor, C. 1972 On calculating with B-splines. *J of Approximation Theory*. 6(1): 50-62

Delaunay, B. 1934 Sur la sphère vide. *Izvestia Akademii Nauk SSSR. Otdelenie Matematicheskikh i Estestvennykh Nauk*. 7(6):793-800

DIN 32880-1:1986 Coordinate Metrology; Geometrical Fundamental Principles, Terms and Definitions. *German Standard*. Beuth Verlag, Berlin

Dinh, H. Q. 2000 *A Sampling of Surface Reconstruction Techniques*. GVU Technical Report; GIT-GVU-00-28

Dorai, C., Weng, J. and Jain, A. 1998 Registration and integration of multiple object views or 3D model construction. *Trans Patt Anal and Machine Intell*. 20(1): 83-89

Duda, R. M. and Hart, P. E. 1973 *Pattern Classification and Scene Analysis*. New York. Wiley

Eggert, D. W., Lorusso, A. and Fisher, R. B. 1997 Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Mach Vis and Appl*. 9(5-6): 272-290

Ekblom, H. 1974 l_p-methods for robust regression. *BIT*.14(1): 22-32

El-Attar, R. A., Vidyasagar, M. and Dutta, S. R. K. 1979 An algorithm for l_1-norm minimization with applications to nonlinear l_1-approximation. *SIAM J of Numer Anal*.16(1): 70-86

Elad, A. and Kimmel, R. 2003 On bending invariant signatures for surfaces. *IEEE Trans on Patt Anal and Mach Intell*. 25(10): 1285-1295

Fan, F. C. and Tsai, T. H. 2001 Optimal shape error analysis of the matching image for a free-form surface. *Robotics and Computer Integrated Manufacturing*. 17(3): 215-222

Fitzgibbon, A., Pilu, M. and Fisher, R. 1999 Direct least squares fitting of ellipses. *IEEE Trans on Patt Anal and Mach Intell*. 21(5): 476-480

Forbes, A. B. 1990 Least squares best fit geometric elements. In Mason, J. C. and Cox, M. G. Editors. *Algor for Approx II*, 311-319

Franke, R. 1977 Locally determined smooth interpolation at irregularly spaced points in several variables. *J Inst for Math and its Appl*. 19(4): 471-482

Franke, R. and Nielson, G. 1980 Smooth interpolation of large sets of scattered data. *Int J of Numer Methods in Eng*. 15(11): 1697-1704

Funkhouser, T., Min, P., Kazhdan, M. et al. 2003 A search engine for 3D models. *ACM Trans on Graphics*. 22(1): 83-105

Gelfand, N., Mitra, N. J., Guibas, L. J. and Pottmann, H. 2005 Robust global registration. In Desbrun, M. and Pottmann, H. Editors. *Eurographics Symposium on Geometry Processing*. 197-205

Ghorbel, F., Derrorde, S., Mezhoud, R. et al. 2006 Image reconstruction from a complete set of similarity invariants extracted from complex moments. *Patt Recog Lett*.27(12): 1361-1369

Godin, G., Rioux, M. and Baribeau, R. 1994 Three-dimensional registration using range and intensity information. *Proc SPIE: Vidiometrics III*. 2350: 279-290

Goldstein, B. L. M., Kemmerer, S. J. and Parks, C. H. 1998 *A Brief History of Early Product Data Exchange Standards*. NISTIR6221, NIST, USA

Golub, G. H, and van Loan, C. F. 1996 *Matrix Computations*. 3rd Ed. John Hopkins University Press

Gonin, R. and Money, A. H. 1989 *Nonlinear l_p-Norm Estimation*. Marcel Dekker Ltd

Greenspan, M. A., Godin, G. and Talbot, J. 2000 Acceleration of binning nearest neighbour methods. *Proc of Vision Interface*. Montreal, Canada. 337-344

Greenspan, M. and Godin, G. 2001 A nearest neighbour method for efficient ICP. *Proc of the 3rd Int Conf on 3-D Digital Imaging and Modeling* 161-168

Gregorski, B. F., Hamann, B. and Joy, K. I. 2000 Reconstruction of B-spline surfaces from scattered data points. *Proc of Computer Graphics International*, 163-170

Groemer, H. 1996 *Geometric Applications of Fourier Series and Spherical Harmonics*. New York: Cambridge University Press

Gunnarson, K. T. and Prinz, F. B. 1987 CAD model based localization of parts in manufacturing. *Computer.*20(8): 66-74

Han, S. and Medioni, G. 1996 Triangular NURBS surface modeling of scattered data. *IEEE Visualization. Proc of the 7th Conf on Visualization* 295-302

Heiberger, R. M. and Becker, R. A. 1992 Design of an S function for robust regression using iteratively reweighted least squares. *J of Comp and Graphical Stat*. 1(3):181-196

Heseltine, T. 2005 *Face Recognition: Two-Dimensional and Three Dimensional Techniques*. Ph.D Thesis. University of York

Hesher, C., Srivastava, A. and Erlebacher, G. 2002 Principal component analysis of range image for facial recognition. *Proc of IEEE CISST*. Las Vagas

Hilaga, M., Shinagawa, Y., Kohmura, T. et al. 2001 Topology matching for fully automatic similarity estimation of 3D shapes. *Proc of the 28th Annual Conf on Computer Graphics and Interactive Techniques*. 203-212

Hill, D. L. G., Batchelor, P. G., Holden, M. and Hawkes, D. J. 2001 Medical image registration. *Physics in Medicine and Biology*. 46(3): R1-R45

Hill, M., Jung, M. and McBride, J. W. 2002 Separation of form from orientation in 3D measurements of aspheric surfaces with no datum. *Int J of Machine Tools & Manufacture*. 42(4): 457-466

Hong, Y. P. and Pan, C. T. 1992 Rank-revealing QR factorizatios and the singular value decomposition. *Mathematics of Computation*. 58(197): 213-232

Horn, B. K. P. 1987 Closed-form solution of absolute orientation using unit quaternions. *J Opt Soc Am*. 4(4): 629-642

Householder, A. S. 1958 Unitary triangulation of a nonsymmetric matrix. *J of ACM*. 5(4): 339-342

Huber, P. J. 1964 Robust Estimation of a Location Parameter. *Annals of Math Stat*. 35(1): 73-101

Huhtanen, M. and Larsen, R. M. 2002 On generating discrete orthogonal bivariate polynomials. *BIT Numerical Mathematics*. 42(2): 393-407

ISO 4291: 1985 *Methods for the Assessment of Departure from Roundness-Measurement of Variations in Radius*

ISO 10110-12: 2007 *Optics and Photonics-Preparation of Drawings for Optical Elements and Systems*-Part 12: Aspheric Surfaces

Iyer, N., Jayanti, S., Lou, K. et al. 2005 Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*. 37(5): 509-530

Jaeckel, L. A. 1972 Estimating regression coefficients by minimizing the dispersion of the residuals. *Annals of Mathematical Statistics*. 43(5): 1449-1458

Johnson, A. E. 1997 *Spin-Images: A Representation for 3-D Surface Matching*. PhD Thesis. Carnegie Mellon University, USA

Joshi, S. and Chang, T. C. 1988 Graph-based heuristics for recognition of matching features from a 3D solid model. *Computer Aided Design*. 20(2): 58-66

Jost, T. 2002 *Fast Geometric Matching for Shape Registration*. Ph.D Thesis. Université de Neuchâtel. Switzerland

Jung, M., Cross, K. J., McBride, J. W. and Hill, M. 2000 A method for the selection of algorithms for form characterization of nominally spherical surfaces. *Precision Engineering*. 24(2): 127-138

Ko, K. H., Maekawa, T. and Patrikalakis, N. M. 2005 Algorithms for optimal partial matching of free-form objects with scaling effects. *Graphical Models*. 67(2): 120–148

Kohtanzad, A. and Yong, Y. H. 1990 Invariant image recognition by Zernike moments. *IEEE Trans Pattern Anal Mach Intell*. 12(5): 489-497

Korn, G. and Korn, H. 1968 *Mathematical Handbook for Scientists and Engineers*. 2nd Ed. McGraw-Hill, New York

Kyprianou, L. 1980 *Shape Classification in Computer-Aided Design*. Ph.D Thesis. Cambridge University, UK

Lai, H. Y., Jywe, W. Y., Chen, C. K. and Liu, C. H. 2000 Precision modelling of form errors for cylindricity using genetic algorithms. *Precision Engineering*. 24(4): 310-319

Lawson, C. L. 1961 *Contributions to the Theory of Linear Least Maximum Approximation*. Ph.D Thesis. University of California, Los Angeles

Lee, W. B., To, S., and Cheung, C. F. 2005 *Design and Advanced Manufacturing Technology for Freeform Optics*. The Hong Kong Polytechnic University (in Chinese)

Lee, Y., and Yi, T. 2003 3D face recognition using multiple features for local depth information. *4th EURASIP Conf on Video/Image Processing and Multimedia Comm*. 1: 429-434

Lei, D. 2002 *Robust and Efficient Algorithms for l_1 and l_∞ Approximations*. Ph.D thesis. University of Huddersfield

Liu, Y. 2004 Improving ICP with easy implementation for free-form surface matching. *Pattern Recognition*, 37(2):211 – 226

Lloyd, E. H. 1952 Least-squares estimation of location and scale parameters using order statistics. *Biometrika*. 39(1-2): 88-95

Lukács, G., Martin, R. and Marshall, D. 1998 Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. *Proc of the 5th Euro Conf on Computer Vision*, 1: 671-686

Madsen, K. and Nielson, H. B. 1990 Finite algorithms for robust linear regression. *BIT*. 30(4): 682-699

Maintz, J. B. A. and Viergever, M. A. 1998 An overview of medical image registration methods. *Medical Image Analysis*. 2(2): 1-36

Maronna, R., Martin, D. and Yohai, V. 2006 *Robust Statistics-Theory and Methods*. Wiley

Masuda, T., Sakaue, K. and Yokoya, N. 1996 Registration and integration of multiple range images for 3-D model construction. *13th Int Conf on Patt Recog*, 1: 879-883

Meer, P., Mintz, D. and Rosenfeld, A. 1991 Robust regression methods for computer vision: a review. *Int J of Computer Vision*. 6(1): 59-70

Moore, D. and Warren, J. 1990 *Approximation of Dense Scattered Data using Algebraic Surfaces*. TR 90-135, Rice University

Mortara, M. 2001 Similarity measures for blending polygonal shapes. *Computers & Graphics*. 25(1): 13-27

Mosteller, F. and Tukey, J. W. 1977 *Date Analysis and Regression*. Reading, MA. Addison-Wesley

Muraki, S. 1991 Volumetric shape description of range data using 'blobby model'. *Computer Graphics*. 25(4): 227-235

Nasraoui, O. 2002 A brief overview of robust statistics. http://louisville.edu/~o0nasr01/Websites/tutorials/RobustStatistics/RobustStatistics.html

Okatani, I. S. 2002 A method for fine registration of multiple view range images considering the measurement error properties. *Computer Vision and Image Understanding*. 87(1-3): 66-77

Olive, D. J. 2007 Applied Robust Statistics. http://www.math.siu.edu/olive/run.pdf

Olson, C. F. 1997 Efficient pose clustering using a randomized algorithm. *Int J of Comp Vision*. 23(2): 131-147

Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D. 2002 Shape distributions. *ACM Trans on Graphics*. 21(4): 807-832

Paquet, E., Murching, A., Naveen, T. et al.2000 Description of shape information for 2-D and 3-D objects. *Signal Processing: Image Communication*. 16(1-2): 103-122

Park, S. Y. and Subbarao, M. 2003 An accurate and fast point-to-plane registration technique. *Patt Recog Lett*. 24(16):2967-2976

Paulli, K. 1999 Multiview registration of large data sets. *Proc of 2nd Int Conf on 3-D Digital Imaging and Modeling*. 160-168

Petitjean, S. 2002 A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys*. 34(2): 211-262

Petrushev, P. P. and Popov, V. A. 1987 *Rational Approximation of Real Functions*. Cambridge University Press

Piegl, L. and Tiller, W. 1997 *The NURBS Book*. 2nd Ed. Springer-Verlag, New York

Planitz, B. M., Maeder, A. J. and Williams, J. A.2005 The Correspondence framework for 3D surface matching algorithms. *Computer Vision and Image Understanding* 97(3):347–383

Pottmann, H., Huang, Q. X., Yang, Y. L. and Hu, S. M. 2006 Geometry and convergence analysis of algorithms for registration of 3D shapes. *Int J of Computer Vision*. 67(3): 277-296

Pratt, V. 1987 Direct least-squares fitting of algebraic surfaces. *Computer Graphics*. 21(4): 145-1652

Rey, W. J. J. 1983 *Introduction to Robust and Quasi-Robust Statistical Methods*. Springer-Verlag

Rice, J. R. and Usow, K. H. 1968 The Lawson algorithm and extensions. *Mathematics of Computation*. 22(101): 118-127

Romdhani, S., Blanz, V. and Vetter, T. 2002 Face identification by fitting a 3D morphable model using linear shape and texture error functions. *Proc of the Euro Conf on Computer Vision*. 3-19

Rousseeuw, P. J. 1984 Least median of squares regression. *J of the American Statistical Association*. 79(388): 871-880

Rusinkiewicz, S. and Levoy, M. 2001 Efficient variants of the ICP algorithm. *Proc of the 3rd Conf on 3-D Digital Imaging and Modeling*, 145-152

Sahoo, K. C. and Menq, C. H. 1991 Localization of 3-D objects having complex sculptured surfaces using tactile sensing and surface description. *ASME J of Engineering for Industry*. 113(1): 85-92

Samuel, G. L. and Shunmugam, M. C. 1999 Evaluation of straightness and flatness error using computational geometric techniques. *Computer Aided Design*. 31(13): 829-843

Schoenberg, I. J. 1967 On spline functions. In Sischa, O. Editor. *Inequalities*. Academic Press, New York. 255-291

Scott, P. J. 2002 Recent developments in the measurement of aspheric surfaces by contact stylus instrumentation. *Conf on Optical Design and Testing. Proc. SPIE* Vol. 4927,199-207

Sharp, G. C., Lee, S. W. and Wehe, D. K. 2002 ICP registration using invariant features. *IEEE Trans on Patt Anal and Mach Intell* 24(1): 90-102

Shepard, D. 1968 A two dimensional interpolations function for irregularly spaced data. *Proc of the 23rd National Conference of the ACM*. 517-523

Shokoufandeh, A. and Dickinson, S. J. 2001 A unified framework for indexing and matching hierarchical shape structures. *Proc of the 4th Int Workshop on Visual Form*. 67-84

Simon, D. A. 1996 *Fast and Accurate Shape-Based Registration*. Ph.D Thesis. Carnegie Mellon University, USA

Subrahmanyam, S. and Wozny, M. 1995 An overview of automatic feature recognition techniques for computer-aided process planning. *Comp in Industry*. 26(1):1-21

Sukumar, S. R. 2004 *Curvature Variation as Measure of Shape Information*. Master Thesis. University of Tennessee, USA

Sun, W. 2007 *Precision Measurement and Characterisation of Spherical and Aspheric Surfaces*. Ph.D Thesis, University of Southampton, UK.

Sun, Y., Paik, J., Koschan, A. et al. 2003 Point fingerprint: a new 3-D object representation scheme. *IEEE Trans on Sys, Man and Cybernetics-B* 33(4):712-717

Sundar, H., Silver, D., Gagvani, N. and Dickinson, S. J. 2003 Skeleton based shape matching and retrieval. *Proc of the Shape Modeling International*. IEEE Computer Society. 130-142

Suzuki, M. T., Kato, T. and Otsu, N. 2000 A similarity retrieval of 3D polygonal models using rotation invariant shape descriptors. *Porc of the IEEE Int Conf on Sys, and Cybernetics*. Vol 4: 2946-2952

Tanaka, H., Ikeda, M. and Chiaki, H. 1998 Curvature-based surface recognition using spherical correlation principal directions for curved object recognition. *Proc of the 3rd Int Conf on Automated Face and Gesture Recognition*. 372-377

Taubin, G. 1991 Estimation of planar curves, surfaces and nonplanar spaces curves defined by implicit equation with applications to edge and range image segmentation. *IEEE Trans on Patt Anal and Mach Intell*. 13(11): 1115-1138

Turk, G. and Levoy, M. 1994 Zippered polygon meshes from range images. *Proc of the 21st Annual Conf on Computer Graphics and Interactive Techniques*. 311-318

Venugopal, V. 1999 Soft-computing-based approaches to the group technology problem: a state-of-the-art and review. *Int J of Production Research*,37(14): 3335-3357

Wang, P. J., Chen, J. H., Li, Z. Q. and Zhou, J. 1997 A new algorithm for the profile error of a parameter surface. *J of Huazhong Univ of Sci and Technol*. 25(3):1-4

Wang, R. F. and Turner, J. 1989 *Recent Research in Feature-Based Design*. Technical Report. No. 89020, Rensselaer Design Research Centre, Rensselaer Polytechnic Institute, Troy, NY

Weik, S. 1997 Registration of 3-D partial surface models using luminance and depth information. *Proc of the International Conf on Recent Advances in 3-D Imaging and Modelling*. 93-100

Wiskott, L., Fellous, J. M., Kruger, N. and von der Malsburg, C. 1997 Face recognition by elastic bunch graph matching. *Proc of 7th Int Conf on Computer Analysis of Images and Patterns*. 456-463

Yager, R. R. and Zadeh, L. 1994 *Fuzzy Sets, Neural Networks and Soft Computing*. Van Nostrand Reinhold. New York

Yamany, S. M. and Farag, A. A. 1999 Free-form surface registration using range signatures. *Proc 7th IEEE Int Conf on Comp Vis*. 106-113

Zhang, C. and Chen, T. 2001 Efficient feature extraction for 2D/3D objects in mesh representation. *IEEE Int Conf on Image Processing*. Thessakoniki, Greece

Zhang, D. 1999 *Harmonic Shape Images: A 3D Free-Form Surface Representation and Its Applications in Surface Matching*. Ph.D Thesis,Carnegie Mellon University,USA

Zhang, Z. 1997 Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing*. 15(1): 59-79

Zitová, B. and Flusser, J. 2003 Image registration methods: a survey. *Image and Vision Computing*. 21(11): 977-1000

CHAPTER 3 SURFACE RECONSTRUCTION WITH NURBS

In CAD-CAM, the design model of a free-form component is generally provided with a 3D CAD model in the format of IGES, STEP etc. To evaluate the form quality of the measurement data, we need the nominal template as a reference. A straightforward continuous representation of the template is required for further processing. As stated before, reading or transforming the CAD models is rather tough, and some geometric information may be lost therein. Thus in this thesis a complex CAD file will be firstly sampled into discrete points accordingly, and then reconstructed into a continuous surface via proper mathematical tools under some restrictions on accuracy and surface smoothness property. Additionally, the measurement data may sometimes need to be resampled or interpolated for subsequent filtering or other post-processing, thus surface reconstruction is also required.

Most optical instruments like Talysurf CCI and other interferometers record measured results through CCD and generate data in a form of regular grid. In the metrology area, surface data are also organized in regular grid formats (2.5 D data), because they are convenient for subsequent mathematical calculations like window filtration. Here we adopt NURBS for surface reconstruction of regular points.

3.1 Introduction to NURBS

In late 1980s, Les Piegl and Wayne Tiller proposed the *Non-Uniform Rational B-Spline* (NURBS), which represents a surface as [Piegl 1997],

$$\mathbf{S}(u,v)=\frac{\sum_{k=1}^{S}\sum_{l=1}^{T}N_{k,m}(u)N_{l,n}(v)w_{k,l}\mathbf{p}_{k,l}}{\sum_{k=1}^{S}\sum_{l=1}^{T}N_{k,m}(u)N_{l,n}(v)w_{k,l}} \tag{3.1}$$

In the equation, m and n are degrees of the spline in the u and v directions respectively, and $\{N_{k,m}(u)\}$ and $\{N_{l,n}(v)\}$ are basis functions. $\{\mathbf{p}_{k,l} \mid \mathbf{p}_{k,l}\in\Re^{3\times1},\ k=1,\cdots,S,\ l=1,\cdots,T\}$ are control points. $\{w_{k,l} \mid w_{k,l}\geq 0, k=1,\cdots,S, l=1,\cdots,T\}$ are weighting parameters, which are

used to measure the relative influence of each control point onto the NURBS surface. Foot point parameters u and v are usually normalized into the span [0, 1].

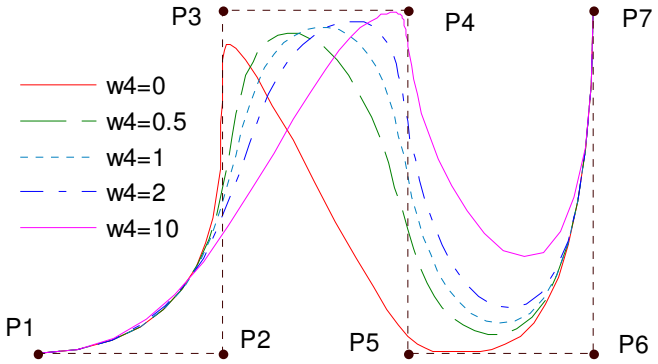


Figure 3.1 Effect of weighing on NURBS curve

To clarify the effect of the weights, we give a NURBS curve in Figure 3.1. Firstly set all the weights to be $\{ w_k = 1 \mid k = 1, \cdots, 7\}$ for all the seven control points. If w_4 (the weight associated with the control point $\mathbf{p}_4$) increases (respectively, decreases), the curve section near $\mathbf{p}_4$ is pulled toward (respectively, pushed away from) $\mathbf{p}_4$. Obviously, if $w_4 \to \infty$, the curve passes through the control point $\mathbf{p}_4$. On the other hand, if $w_4 \to 0$, the point $\mathbf{p}_4$ will have no influence onto the curve.

The advantages and disadvantages of the NURBS reconstruction are listed below [Barker 2004],

Advantages

- For regularly distributed data, NURBS reconstruction is very efficient and numerically stable.
- For data representing similar qualitative behaviours, it is usually possible to determine good approximations.
- For regularly distributed data it is easy to check whether the knots are well chosen.
- Because of the local supporting property, errors only affect the local neighbourhood. If one data point is invalid, other areas will still be correct.
- To modify one part of the surface, it is only necessary to change the control points and/or basis functions at this area. It does not need to recalculate the whole surface.

• NURBS is able to represent highly complex curves and surfaces and can represent analytical features exactly.

• NURBS has unified representations of 2-D and 3-D curves.

• NURBS is invariant under perspective transformation, while B-spline is invariant under affine transformation.

Disadvantages

• If the data or surface exhibits different behaviours in different regions, the choice of knots will affect the reconstruction quality. In this case the tensor product approach will not be efficient.

• For scattered data, there is no easily tested criterion to determine a *priori* whether or not the approximation with splines is well posed.

• The interpolation matrix is often rank-deficient or poorly conditioned, especially when the number of data or control points is very large.

For the sake of its superior characteristics, NURBS is nearly ubiquitous in computer aided design, manufacturing and reverse engineering, and is widely used in some standard formats, e.g. STEP, ACIS, and PHIGS.

There are two approaches to control the shape of NURBS curves/surfaces: weight modification and control point movement. Certain standard techniques have been developed to assign weights for some basic geometric elements [Piegl 1997]; whereas concerning general shaped surfaces, the calculation of weights is not so straightforward [Wang 2001]. In fact, it is a practical approach to utilize the same weights for all the control points when fitting general-shaped surfaces. As a consequence the denominator in Equation (3.1) can be neglected and NURBS surfaces become the *Non-Uniform B-spline* surfaces,

$$\mathbf{S}(u,v) = \sum_{k=1}^{S}\sum_{l=1}^{T} N_{k,m}(u) N_{l,n}(v) \mathbf{p}_{k,l} \tag{3.2}$$

3.2 Reconstruction Procedure of NURBS Surfaces

Suppose the input data $\{\mathbf{x}_{ij} \mid \mathbf{x}_{ij} \in \Re^{3\times 1}, i = 1,\cdots,N, j = 1,\cdots,M\}$ are regularly distributed in N rows and M columns. Without loss of generality, we assume the x and y coordinates are in an ascending order in each row and column respectively, i.e.

$$\begin{cases} x_{ij} > x_{ik} & \text{if } j>k \\ y_{ij} > y_{kj} & \text{if } i>k \end{cases}$$

A continuous surface can be constructed through the following steps: parameterization, selecting knots, determining degrees, calculating basis functions and calculating control points.

(a) Parameterization

Normally the foot-point parameters of a NURBS surface lie within the interval [0, 1], but in fact the abscissa of the data points rarely satisfy this. Hence the location coordinates of the input data need to be scaled first so that their corresponding location parameters can be obtained.

If the data are exactly located in a grid format, i.e. the data points have the same x coordinates in each column and the same y coordinates in each row, the corresponding foot-point parameters can be calculated by a simple linear transformation,

$$\begin{cases} \tilde{u}_j = \dfrac{x_j - x_1}{x_M - x_1} \\ \tilde{v}_i = \dfrac{y_i - y_1}{y_N - y_1} \end{cases} \tag{3.3}$$

Thereupon the resulting location parameters satisfy $0=\tilde{u}_1<\tilde{u}_2<\cdots<\tilde{u}_M=1$, $0=\tilde{v}_1<\tilde{v}_2<\cdots<\tilde{v}_N=1$.

If the data are not exactly located in a grid format, the coordinates can be transformed into a parameter space to make their foot-point parameters located in a grid. As a result the NURBS surface can be constructed in the manner of tensor product. The most simple location parameters are a uniform system. Take the calculation of $\{\tilde{u}_j\}$ as an example.

$$\tilde{u}_j = \frac{j-1}{M-1} \tag{3.4}$$

When the data points are uniformly distributed, i.e. the distances between all the adjacent points within one row are nearly the same, equally spaced parameters work well. But if the data are unevenly spaced, it will produce erratic shapes. In this case non-uniform parameters are needed. Obviously it is intuitive to assign parameters according to the distances between adjacent points. Some common parameterization techniques are listed here [Piegl 1997, Yin 2004].

• Cumulative chord length

$$\tilde{u}_{i,j} = \frac{\sum_{k=1}^{j-1} \| \mathbf{x}_{i,k+1} - \mathbf{x}_{i,k} \|}{\sum_{k=1}^{M-1} \| \mathbf{x}_{i,k+1} - \mathbf{x}_{i,k} \|} \tag{3.5}$$

It is capable of giving a good parameterisation and thus is very widely used.

• Centripetal model

$$\tilde{u}_{i,j} = \frac{\sum_{k=1}^{j-1} \| \mathbf{x}_{i,k+1} - \mathbf{x}_{i,k} \|^{1/2}}{\sum_{k=1}^{M-1} \| \mathbf{x}_{i,k+1} - \mathbf{x}_{i,k} \|^{1/2}} \tag{3.6}$$

It can give better results when the data occupy highly curved parts.

• Generalization of the centripetal model: exponential model

$$\tilde{u}_{i,j} = \frac{\sum_{k=1}^{j-1} \| \mathbf{x}_{i,k+1} - \mathbf{x}_{i,k} \|^{e}}{\sum_{k=1}^{M-1} \| \mathbf{x}_{i,k+1} - \mathbf{x}_{i,k} \|^{e}} \tag{3.7}$$

It is a generalisation of the centripetal method. For different types of data, the parameter *e* can be adjusted to make the fitted surface more accurate.

After all of $\{\tilde{u}_{i,j}\}$ are calculated, they are averaged by

$$\tilde{u}_{j} = \frac{1}{N} \sum_{i=1}^{N} \tilde{u}_{i,j} \tag{3.8}$$

(b) Selection of Knots

The number of knots can be determined by the user according to the data size and surface shape. It is clear that selecting more knots can improve the reconstruction accuracy, whilst reducing the efficiency. Hence an appropriate compromise should be made between the accuracy and efficiency.

From the view point of computational complexity, a uniform B-spline system is preferred. But the generated surface may not be consistent with the surface shape and distribution of data points. Here a criterion is given to decide whether to use uniform or non-uniform knots [Zhu 1981],

Set $\begin{cases} \alpha = \max\limits_{i,j} \dfrac{\| \mathbf{x}_{i,j+1} - \mathbf{x}_{i,j} \|}{\| \mathbf{x}_{i,j} - \mathbf{x}_{i,j-1} \|} \\ \beta = \min\limits_{i,j} \dfrac{\| \mathbf{x}_{i,j+1} - \mathbf{x}_{i,j} \|}{\| \mathbf{x}_{i,j} - \mathbf{x}_{i,j-1} \|} \end{cases}$. If $\begin{cases} \alpha \le 3 \\ \beta \ge 1/3 \end{cases}$ and the surface is sufficiently smooth, adopt uniform knots; otherwise non-uniform knots $\{u_k\}$ will be employed. At the regions where surface shape varies sharply or the data spacing is smaller, denser knots will be placed, and vice versa.

In practice, the knots are often clamped, i.e. $u_1 = u_2 = \cdots = u_{m+1} = 0$, and $u_N = u_{N+1} = \cdots = u_{N+m} = 1$, so that the boundary data points coincide with the starting and end control points respectively.

The parameterization and selection of knots in the *v* direction can be implemented in the same way.

(c) Determination of Degrees

The degrees *m* and *n* directly determine the shape properties of the surface such as smoothness and continuity. Surfaces with higher degrees are more flexible, and at the same time more computationally complex. In practice, *m*=*n*=3, i.e. a bi-cubic surface is commonly applied. These surfaces are C^2 continuous at the knots and infinitely differentiable at the interior of the knot spans. A cubic spline curve is the one which minimizes the functional,

$$J(f) = \int_a^b \left| f''(x) \right|^2 dx \tag{3.9}$$

over the function $f(x)$ in the Sobolev space $H^2([a,b])$.

This means the cubic spline is also the approximation of the curve with minimal curvature, i.e. it is an elastic strip with the minimal strain energy constrained to pass the given data points [de Boor 1978].

(d) Calculation of Basis Functions

Now all the basis functions associated with each data point can be calculated. If the knots are non-uniform, the basis functions should be calculated recursively using the de Boor-Cox algorithm [de Boor 1972, Cox 1972], whereas for uniform knots, we have worked out the explicit formulations. Thus the design matrix of NURBS reconstruction is obtained, and only the control points need to be calculated.

(e) Calculation of Control Points

In Equation (3.2), all but the control points are already known. The subsequent steps are the same with the reconstruction procedure of common tensor products. The bases in the equation are separable,

$$\mathbf{x}_{ij} = \sum_{k=1}^{S}\sum_{l=1}^{T} N_{km}(\tilde{u}_j) N_{\ln}(\tilde{v}_i)\mathbf{p}_{kl} = \sum_{k=1}^{S} N_{km}(\tilde{u}_j)\sum_{l=1}^{T} N_{\ln}(\tilde{v}_i)\mathbf{p}_{kl} \tag{3.10}$$

It can be rewritten in a matrix form,

$$\mathbf{x}_{ij} = \mathbf{\Phi}(\tilde{u}_j)\mathbf{P}\mathbf{\Psi}(\tilde{v}_i) \tag{3.11}$$

with $\mathbf{\Phi} = [N_{1m}, N_{2m}, \cdots, N_{Sm}]$, $\mathbf{\Psi} = [N_{1n}, N_{2n,} \cdots, N_{Tn}]^T$ and $\mathbf{P} = \{\mathbf{p}_{kl}\} \in \Re^{S\times T}$.

Firstly the data are processed row by row,

$$\begin{bmatrix} N_{1m}(\tilde{u}_1) & N_{2m}(\tilde{u}_1) & \cdots & N_{Sm}(\tilde{u}_1) \\ N_{1m}(\tilde{u}_2) & N_{2m}(\tilde{u}_2) & \cdots & N_{Sm}(\tilde{u}_2) \\ \vdots & \vdots & \ddots & \vdots \\ N_{1m}(\tilde{u}_M) & N_{2m}(\tilde{u}_M) & \cdots & N_{Sm}(\tilde{u}_M) \end{bmatrix} \mathbf{P}\mathbf{\Psi}(\tilde{v}_i) = \begin{bmatrix} \mathbf{x}_{i1} \\ \mathbf{x}_{i2} \\ \vdots \\ \mathbf{x}_{iM} \end{bmatrix}, i = 1, \cdots, N \tag{3.12}$$

It is simplified into,

$$\mathbf{A}\mathbf{K}_i = \mathbf{X}_i \tag{3.13}$$

The number of the variables S is less than that of the constraints M, hence this is an over-determined system. Usually it is solved in the sense of least squares,

$$\mathbf{K}_i = \mathbf{A}^{\dagger}\mathbf{X}_i \tag{3.14}$$

Here $\mathbf{A}^{\dagger}$ is the pseudoinverse of $\mathbf{A}$. Then we can solve the control points from $\{\mathbf{K}_i = \mathbf{P}\boldsymbol{\Psi}(\tilde{v}_i), | \, i = 1, \cdots, N\}$. Each row is processed as follows,

$$\mathbf{P}(k,:)\begin{bmatrix} N_{1n}(\tilde{v}_1) & N_{1n}(\tilde{v}_2) & \cdots & N_{1n}(\tilde{v}_N) \\ N_{2n}(\tilde{v}_1) & N_{2n}(\tilde{v}_2) & \cdots & N_{2n}(\tilde{v}_N) \\ \vdots & \vdots & \ddots & \vdots \\ N_{Tn}(\tilde{v}_1) & N_{Tn}(\tilde{v}_2) & \cdots & N_{Tn}(\tilde{v}_N) \end{bmatrix} = \begin{bmatrix} K_{k1} & K_{k2} & \cdots & K_{kN} \end{bmatrix} \tag{3.15}$$

In the equation, $(k,:)$ denotes the k-th row of the matrix.

Again it is simplified into,

$$\mathbf{P}(k,:)\mathbf{B} = \mathbf{K}(k,:), k = 1, \cdots, S \tag{3.16}$$

Similarly, we obtain,

$$\mathbf{P} = \mathbf{K}\mathbf{B}^{\dagger} \tag{3.17}$$

Here $\mathbf{A}^{\dagger}$ and $\mathbf{B}^{\dagger}$ need only to be calculated once. In fact $\mathbf{P}$ is a matrix of size $S \times T \times 3$.The x, y and z components in $\{\mathbf{x}_{ij}\}$ and $\{\mathbf{p}_{kl}\}$ should be handled separately in Equations (3.14) and (3.17). That is to say, utilizing the same observation matrix, gain the x coordinates of the control points from the x values of the input data, and then y and z respectively.

The matrices $\mathbf{A}$ and $\mathbf{B}$ are banded and only four elements in each row are nonzero. Some computational techniques specially developed for sparse matrices can be employed to save computation cost and memory space [Björck 1996]. If the data size M and N are very large, the design matrix $\mathbf{A}$ may become ill-conditioned. Thus stable inversing techniques such as the Truncated SVD and Rank-Revealing QR Decomposition mentioned in Subsection 2.4.1 can be utilized.

3.3 Point Inversion and Projection

3.3.1 Point Inversion

When implementing surface interpolation, usually the x and y coordinates at the interpolated locations are given and the corresponding z coordinates need to be computed. However, NURBS surfaces are in parametric forms, so that the foot-point parameters u and v should be worked out first. This is called *point inversion* [Piegl 1997].

The procedure of point inversion can be divided into two steps:

(1) Find the parameter spans associated with the given point based on the strong convex hull property of NURBS, so that proper basis functions can be used.

(2) Iteratively refine the solution (u_0, v_0) with the Newton-Raphson algorithm.

Suppose the NURBS surface is $\overline{\mathbf{S}}(u,v)$ and the given point is $\overline{\mathbf{x}} = [x, y]^T$.

The target is to solve

$$E = \overline{\mathbf{r}}^T\overline{\mathbf{r}} = (\overline{\mathbf{S}} - \overline{\mathbf{x}})^T(\overline{\mathbf{S}} - \overline{\mathbf{x}}) \tag{3.18}$$

only x and y coordinates are involved in the equation.

It is evident the solution (u_0, v_0) satisfies,

$$\begin{cases} \dfrac{\partial E}{\partial u}\Big|_{u_0} = 2\overline{\mathbf{r}}^T\overline{\mathbf{S}}_u\Big|_{u_0} = 0 \\ \dfrac{\partial E}{\partial v}\Big|_{v_0} = 2\overline{\mathbf{r}}^T\overline{\mathbf{S}}_v\Big|_{v_0} = 0 \end{cases} \tag{3.19}$$

The solution can be updated iteratively as,

$$\begin{bmatrix} \overline{\mathbf{S}}_u^T\overline{\mathbf{S}}_u + \overline{\mathbf{r}}^T\overline{\mathbf{S}}_{uu} & \overline{\mathbf{S}}_u^T\overline{\mathbf{S}}_v + \overline{\mathbf{r}}^T\overline{\mathbf{S}}_{uv} \\ \overline{\mathbf{S}}_u^T\overline{\mathbf{S}}_v + \overline{\mathbf{r}}^T\overline{\mathbf{S}}_{uv} & \overline{\mathbf{S}}_v^T\overline{\mathbf{S}}_v + \overline{\mathbf{r}}^T\overline{\mathbf{S}}_{vv} \end{bmatrix} \begin{bmatrix} \delta u \\ \delta v \end{bmatrix} = -\begin{bmatrix} \overline{\mathbf{r}}^T\overline{\mathbf{S}}_u \\ \overline{\mathbf{r}}^T\overline{\mathbf{S}}_v \end{bmatrix} \tag{3.20}$$

and

$$\begin{cases} u \leftarrow u + \delta u \\ v \leftarrow v + \delta v \end{cases} \tag{3.21}$$

In Equation (3.20),

$$\overline{\mathbf{S}}_u = \frac{\partial \overline{\mathbf{S}}(u,v)}{\partial u} = \sum_{k=1}^{S}\sum_{l=1}^{T} \frac{\partial N_{k,m}(u)}{\partial u} N_{l,n}(v)\overline{\mathbf{p}}_{k,l}$$

$$\overline{\mathbf{S}}_{uv} = \frac{\partial^2 \overline{\mathbf{S}}(u,v)}{\partial u \partial v} = \sum_{k=1}^{S} \sum_{l=1}^{T} \frac{\partial N_{k,m}(u)}{\partial u} \frac{N_{l,n}(v)}{\partial v} \overline{\mathbf{p}}_{k,l}$$

$\overline{\mathbf{S}}_v$, $\overline{\mathbf{S}}_{uu}$ and $\overline{\mathbf{S}}_{vv}$ are analogous. Here the derivatives of the basis functions with respect to the foot-point parameters are required. Due to the following relationship [Piegl 1997],

$$N_{k,m}^{(d)}(u) = m\left(\frac{N_{k,m-1}^{(d-1)}(u)}{u_{k+m} - u_k} - \frac{N_{k+1,m-1}^{(d-1)}(u)}{u_{k+m+1} - u_{k+1}}\right), \quad d < m \tag{3.22}$$

the derivatives can be derived from the lower order basis functions.

Alternatively it can also be worked out via another approach,

$$N_{k,m}^{(d)}(u) = \frac{m}{m-d}\left(\frac{u - u_k}{u_{k+m} - u_k} N_{k,m-1}^{(d)}(u) + \frac{u_{k+m+1} - u}{u_{k+m+1} - u_{k+1}} N_{k+1,m-1}^{(d)}(u)\right), \quad d < m \tag{3.23}$$

For non-uniform knots, the derivatives can be recursively calculated from Equation (3.22) or (3.23), whereas for uniform knots, we worked out the explicit formulae for the derivatives of cubic B-spline basis functions.

However, sometimes the initial guess of the parameter intervals is not very reliable, especially when the solution lies near the boundaries of parameter spans. The solution may go beyond the current span during the procedure of iterative minimization. Consequently a 'jumping' mechanism is established. A pointer is defined to determine the incremental direction of the solution. When the new solution in Equation (3.21) goes outside the current span, the pointer is changed.

Suppose the current span is $[u_k, u_{k+1})$ and $[v_l, v_{l+1})$.

```
pointer=0;
if  u<u_k
    %jump to the left span
    pointer=pointer-1;
elseif  u > u_{k+1}
    %jump to the right span
    pointer=pointer+1;
end
if  v < v_l
    %jump to the lower span
    pointer=pointer-3;
```

```
elseif  v > v_{l+1}
    %jump to the upper span
    pointer=pointer+3;
end
```

This mechanism yields a jumping map, as depicted in Figure 3.2.

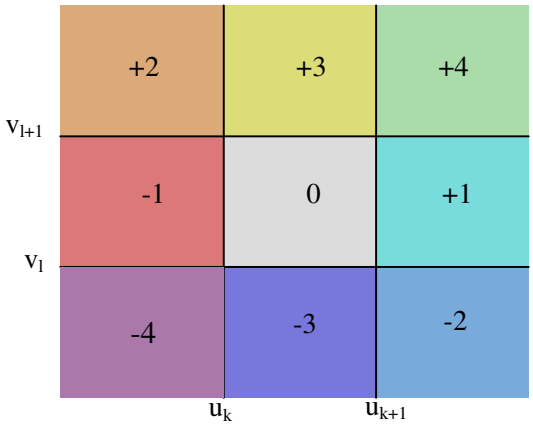


Figure 3.2 Jumping map of point inversion

According to the value of the pointer, we can gain the correct parameter spans at the next iteration.

3.3.2 Point Projection

When matching measurement data with a NURBS surface, it is often demanded to find the closest template point for each measurement datum. Given a point $\mathbf{x} = [x, y, z]^T$, *point projection* is the operation to find a closest point $\mathbf{y} = [x(u,v), y(u.v), z(u,v)]^T$ on the NURBS surface.

The procedure of point projection can also be divided into two stages,

(1) Supply a rough guess for the foot-point parameters.

(2) Refine the solution using the Newton-Raphson algorithm.

The refinement of point projection is the same with that of point inversion. The only difference is that Equations (3.18)-(3.20) apply all x, y and z coordinates instead of only x and y. However, it is very difficult to supply a reliable initial guess, since the convex hull property does not apply in such a situation. Piegl and Tiller proposed to decompose the whole surface into quadrilaterals, and a rough solution can be found by projecting the point onto the closest quadrilateral [Piegl 2001]. But this method is very expensive. Here

we follow the suggestion of Ma and Hewitt [Ma 2003] to decompose the NURBS surface into Bézier patches by knot insertion.

Knot Insertion

For simplicity and clarity, here we take a NURBS curve as an example,

$$\mathbf{C}(u)=\sum_{k=1}^{S} N_{k,m}(u)\mathbf{p}_k \tag{3.24}$$

Its knot vector is $U=[u_1,u_2,\cdots,u_{T1}]$. If inserting a new knot $\bar{u}$ and obtaining a new knot vector $\overline{U}=[u_1,u_2,\cdots,u_a,\bar{u},u_{a+1},\cdots,u_{T1+1}]$, the resultant curve is,

$$\mathbf{C}(u)=\sum_{k=1}^{S+1} \overline{N}_{k,m}(u)\mathbf{q}_k \tag{3.25}$$

The curve is required to remain unchanged either geometrically or parametrically. Thus the key part of knot insertion is to calculate the new control points $\{\mathbf{q}_k\}$. The relationship between the new and old control points is proved to be [Piegl 1997],

$$\mathbf{q}_k=\alpha_k\mathbf{p}_k+(1-\alpha_k)\mathbf{p}_{k-1} \tag{3.26}$$

$$\alpha_k=\begin{cases}1 & k\le a-m\\ \dfrac{\bar{u}-u_k}{u_{k+m}-u_k} & a-m+1\le k\le a\\ 0 & k\ge a+1\end{cases} \tag{3.27}$$

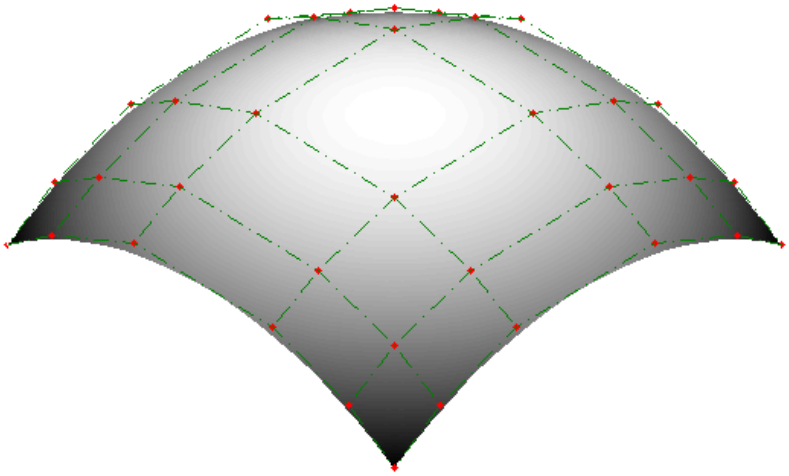
Equation (3.26) can be rewritten as,

$$\mathbf{q}_k=\sum_{i=1}^{S}\alpha_{ik}\mathbf{p}_i \tag{3.28}$$

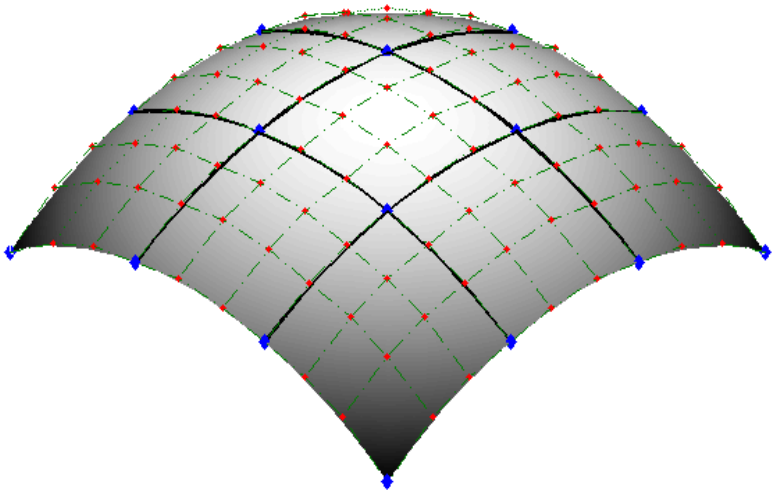
So that the essential task becomes to work out the coefficients $\{\alpha_{ik}\}$. If we want to divide a NURBS curve into Bézier curve sections, the new knot vector should meet these requirements: the multiplicities of two ending knots are *m*+1 and the multiplicities of interior knots are *m*, here *m* indicates the degree of basis functions. That is to say, we shall insert plenty of new knots simultaneously. An efficient insertion algorithm proposed by Pan et al [Pan 2003] is adopted, which will not be presented in detail here.

After knot insertion in the *u* direction, the same manipulation is performed in the *v* direction. As a consequence the NURBS surface is now decomposed into Bézier patches.

Figure 3.3 (a) depicts a simple bi-cubic NURBS surface; by 'simple' here we mean there is no crossing edge. Its knot vectors in u and v directions are both [0, 0, 0, 0, 1/3, 2/3, 1, 1, 1, 1]. 6×6 control points are denoted with red dots. If decomposing this surface into 3×3 Bézier patches, new knot vectors turn out to be [0, 0, 0, 0, 1/3, 1/3, 1/3, 2/3, 2/3, 2/3, 1, 1, 1, 1]. The resulting control polygon is plotted in Figure 3.3 (b). Apparently, all the corner control points of each Bézier patch (denoted with blue diamonds) are located on the surface. Then the parameter spans associated with the projection point of each out-of-surface point can be determined according to the new control polygon.



(a) NURBS surface



(b) Bézier patches

Figure 3.3 Dividing a NURBS surface into Bézier patches

Find the Corresponding Surface Patch

If a control polygon is convex and simple, the corresponding patch is regarded to be valid. Given a valid Bézier patch, the following criterion can be adopted to determine whether the projection point is located at this span [Ma 2003].

For clarity, we firstly investigate the case of a 2D curve. Given a 2D point **x**, we determine whether its projection is within this Bézier section as follows,

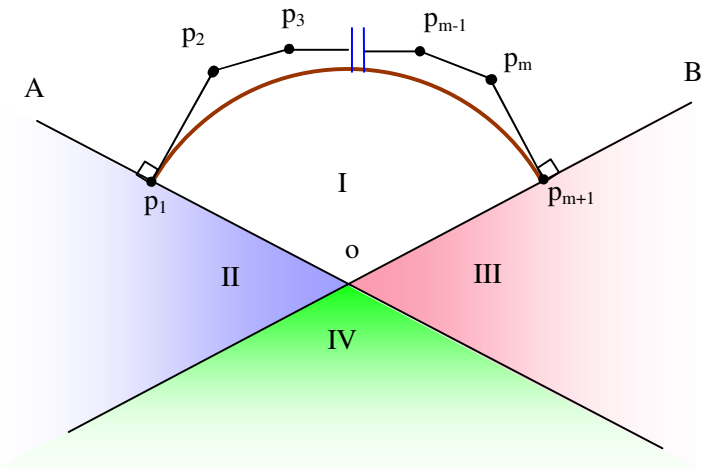


Figure 3.4 Determine the span by Bézier section

Define two dot products $R_1 = \mathbf{p}_1\mathbf{p}_2 \cdot \mathbf{p}_1\mathbf{x}$ and $R_2 = \mathbf{p}_{m+1}\mathbf{p}_m \cdot \mathbf{p}_{m+1}\mathbf{x}$;
if $R_1 \geq 0$ and $R_2 \geq 0$
    x locates at region I and the parameter is at this span;
elseif $R_1 < 0$ and $R_2 \geq 0$
    x locates at region II and the parameter is at the left span;
elseif $R_1 \geq 0$ and $R_2 < 0$
    x locates at region III and the parameter is at the right span;
else
    x locates at region IV;
    if $\| \mathbf{p}_1\mathbf{x} \| \leq \| \mathbf{p}_{m+1}\mathbf{x} \|$
        the parameter is at the left span;
    else
        the parameter is at the right span;
    end
end

Check the control polygon in u and v directions respectively and then a jumping mechanism is built in the same way as point inversion.

If this control polygon is not valid, the patch will be decomposed further until it is valid or flat enough. So that the point can be projected onto the fitted plane of this small planar patch and a rough guess of the foot-point parameters is obtained.

3.4 Numerical Example

The NURBS programmes are coded in MATLAB R2007A and run on a NEC PC with Intel Pentium 4 CPU 3.00GHz, 2.00GB of RAM, and Microsoft Windows XP.

The Carl Zeiss PRISMO Coordinate Measuring Machine (CMM) embeds a software HOLOS to define scanning routes on 3D CAD models and to evaluate the form errors of the measured workpices with respect to the ideal shapes. Figure 3.5 shows a design model of the meniscal bearing component in a knee joint replacement. Through HOLOS we sample 58×45 template points uniformly with spacing 0.4 mm from the bearing surface at the right side of the model, as plotted in Figure 3.6.

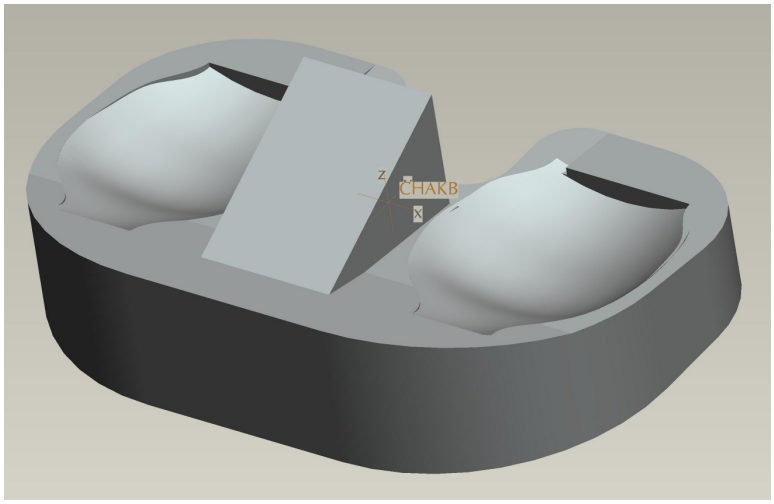


Figure 3.5 Meniscal bearing component

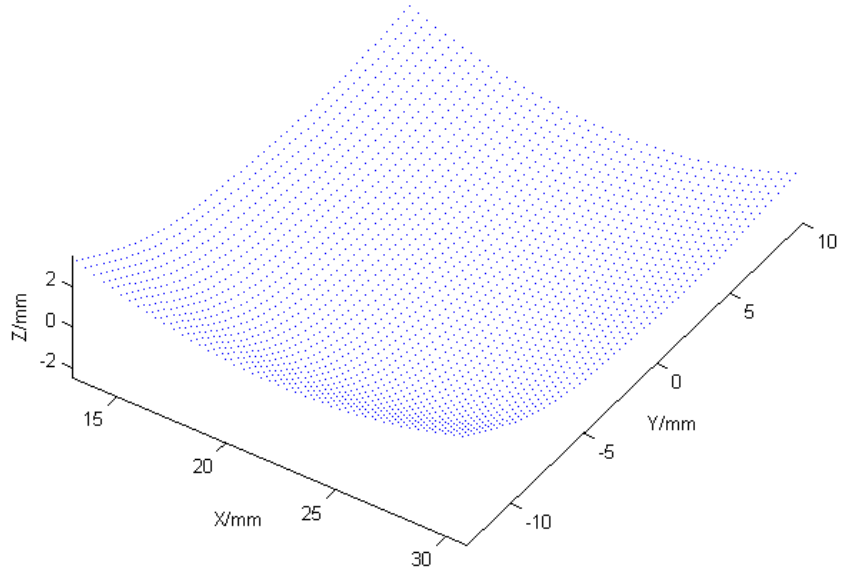


Figure 3.6 58×45 model points

We create a uniform bi-cubic B-spline system to represent this surface. 15 and 18 knots are employed in *u* and *v* knots respectively, yielding a control polygon of size 14×11, as depicted in Figure 3.7. Obviously, this surface is concave. Since the control polygon is its convex hull, i.e. the surface is completely contained within the control polygon, thus all the control points are on or beneath the surface.

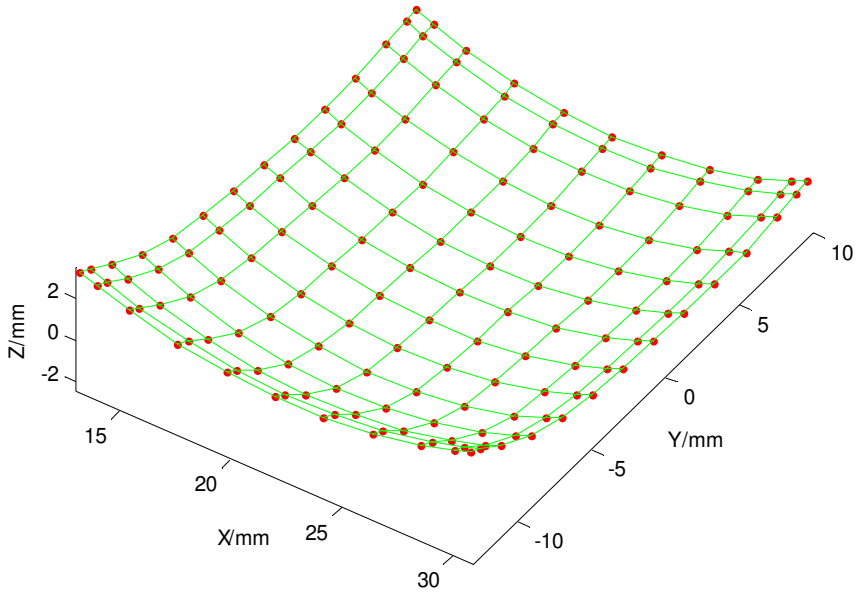


Figure 3.7 A 14×11 control polygon

To assess the accuracy of this NURBS system, 151×114 new points are taken from the original CAD model with spacing 0.15 mm, and point inversion is implemented on

the NURBS surface at the same locations. We use a rather small termination threshold (10^{-6}) in the Newton-Raphson point inversion programme, thus the obtained points on the NURBS surface can be very close to the target positions and the round-off errors introduced at this stage will be very small. It suggests that the reconstruction errors dominate in the relative deviations between the sampled model points and the inversed NURBS points. The relative residuals of their z coordinates are adopted to evaluate the reconstruction error, as plotted in Figure 3.8.

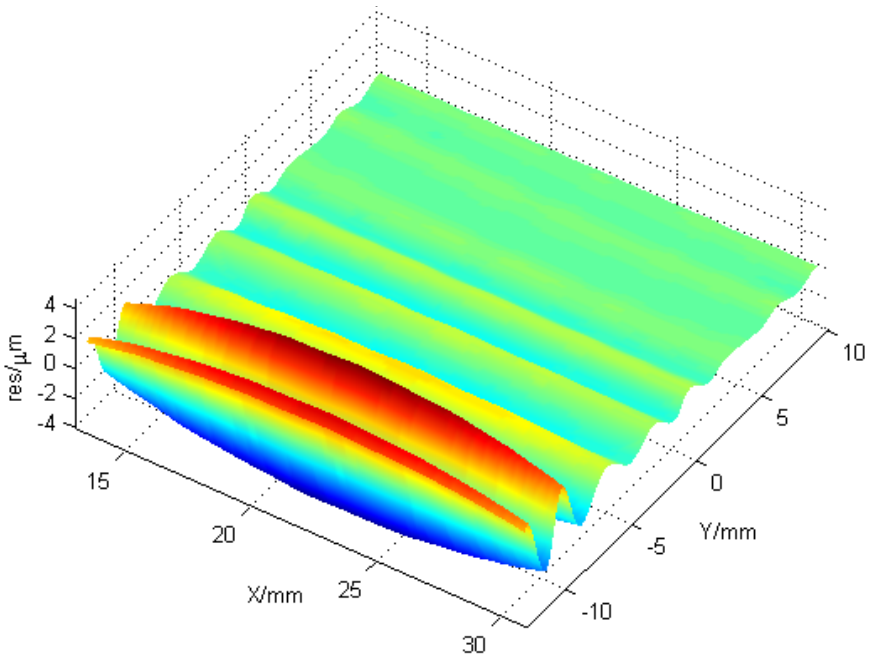


Figure 3.8 Reconstruction residuals with a 14×11 control polygon

Then we change the numbers of knots in the u and v directions into 36 and 46 respectively, and construct a new bi-cubic NURBS surface. The control polygon will be of the size 42×32. In Figure 3.9 it can be seen that the reconstruction residuals are now greatly reduced.

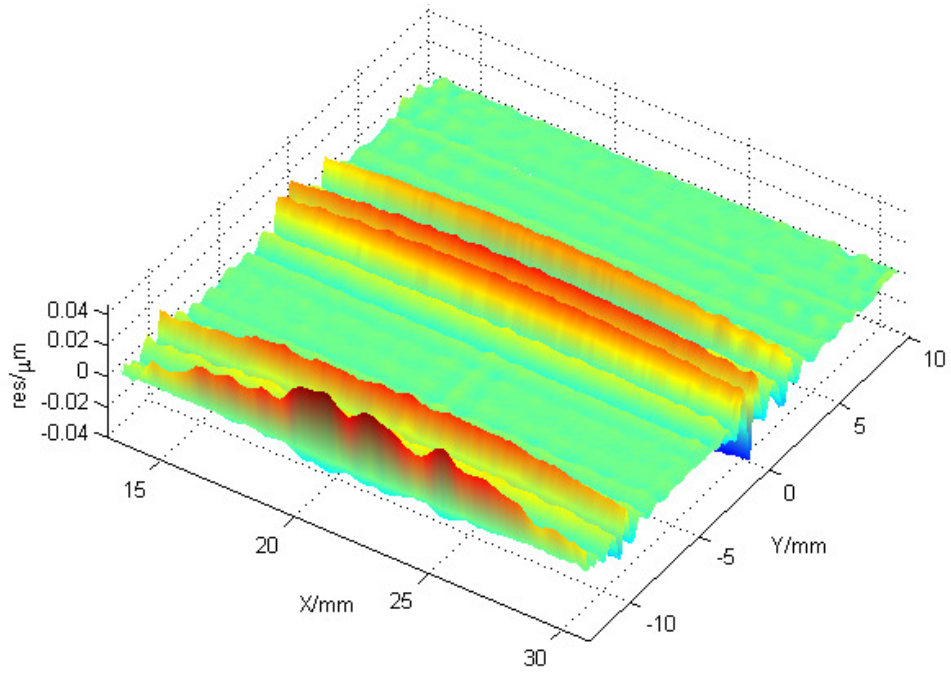


Figure 3.9 Reconstruction residuals with a 42×32 control polygon

To compare their accuracy quantitatively, the S_a (arithmetical mean error), S_q (root mean square error) and S_z (maximum height error) [ISO/DIS 25178-2: 2007] of the residuals are calculated in these two cases. The running time spent on surface reconstruction and point inversion is also recorded, as listed in Table 3.1.

Knot numbers	*u* knot	15	36
	v knot	18	46
Size of control polygon		14×11	42×32
Evaluation Errors/μm	S_a	0.724	0.005
	S_q	1.198	0.009
	S_z	8.726	0.086
Reconstruction time/second		0.110	0.113
Point inversion time/second		10.094	10.386

Table 3.1 Comparison of reconstruction accuracy and efficiency

Applying more control points, the reconstruction accuracy could be improved further, whilst reducing the efficiency. Actually, NURBS reconstruction is very efficient, and very little time is required. In contrast, point inversion is processed successively one point by one point, and a Newton-Raphson iteration is carried out for each point, thereby making the programme very slow. The computational complexity of point inversion is

proportional to the number of evaluation points, whilst not significantly affected by the size of the control polygon. Adopting more control points can bring higher fidelity to the NURBS surface with the original model, and greatly improve the reconstruction accuracy. Hence more control points and knots are preferable when creating NURBS surfaces. But it needs to obey some restrictions. Assume the data is of size $N_1 \times M_1$ and the degrees of basic functions in two directions are n and m. If always using clamped knots, the size of the control polygon $N_2 \times M_2$ should meet,

$$\begin{cases} n+1 \le N_2 \le N_1 \\ m+1 \le M_2 \le M_1 \end{cases} \tag{3.30}$$

When $M_1 = N_1$ and $M_2 = N_2$, it is the exact interpolation; whereas the surface will be a Bézier patch if $M_1 = m+1$ and $N_1 = n+1$.

3.5 Summary

NURBS has gained more and more attraction because of its versatility and powerful capability of modelling and reconstruction.

Actually it is a generalization of a tensor product by two parametric spline curves; hence the input data points are required to be distributed on a regular grid. If the data points are regularly distributed but not exactly in a grid format, parameterization can be implemented to transform the data into a parameter space.

Like other tensor product techniques, the u and v basis functions are separable in the function of a B-spline surface, thus the control points at each row or column can be gained individually, instead of involving all the data points as a whole. The size of the design matrix can thereby be greatly reduced. The design matrix needs only to be constructed and inversed once in the x and y directions respectively. Therefore the reconstruction of NURBS surface is very efficient compared with other methods.

The reconstruction accuracy is determined by the number and positions of knots, which lead to a trade-off between the accuracy and efficiency. A bi-cubic B-spline surface is recommended in practical use for reconstruction of a smooth surface.

Due to the parametric form of a NURBS surface, the corresponding foot-point parameters are required at the given locations when implementing interpolation. It is a

non-linear problem and generally solved by the Newton-Raphson algorithm. Interpolation of NURBS surfaces is not very efficient.

Finding the closest point on a NURBS surface for an out-of-surface point is called point projection. It is a very complicated problem because the representations vary at each parameter span. The entire surface can be divided into Bézier patches and the resultant control polygon is employed to help find a rough guess for the foot-point parameters.

3.6 References

Barker, R. M., Cox, M. G., Forbes, A. B. and Harris, P. M. 2004 *Discrete Modelling and Experimental Data Analysis*. Ver 2. NPL Report

Björck, Å. 1996 *Numerical Methods for Least Squares Problems*. SIAM

Cox, M. G. 1972 The numerical evaluation of B-splines.*J Inst Maths Appl.*10(2):134-149

de Boor, C. 1972 On calculating with B-splines. *J of Approx Theory*. 6(1): 50-62

de Boor, C. 1978 *A Practical Guide to Splines*. Springer

ISO/DIS 25178-2: 2007 *Geometrical Product Specifications-Surface Texture: Areal*-Part 2: Terms, definitions and surface texture parameters

Ma, Y. L. and Hewitt, W. T. 2003 Point inversion and projection for NURBS curve and surfaces: control polygon approach. *Comp Aided Geom Design*.20(2): 79-99

Pan, R. J., Pan, R. H. and Yao, Z. Q. 2003 A fast algorithm for inserting a series of knots into a B-spline curve simultaneously. *Mini-Micro Sys*. 24(12): 2295-2298

Piegl, L. and Tiller, W. 1997 *The NURBS Book*. 2nd Ed. Springer-Verlag, New York

Piegl, L. A. and Tiller, W. 2001 Parametrization for surface fitting in reverse engineering. *Computer-Aided Design* 33(8): 593-603

Wang, X. B. and Li, S. Y. 2001 Automatic calculation of initial weights for NURBS. *Acta Aeronautica et Astronautica Sinica*. 22(2): 184-186

Yin, Z. 2004 Reverse engineering of a NURBS surface from digitized points subject to boundary condition. *Compters & Graphics*. 28(2): 207-212

Zhu, X. 1981 *The Principles and Applications of B-Spline Curves and Surfaces*. Lecture Notes, University of Minnesota

CHAPTER 4 SURFACE RECONSTRUCTION WITH RBF

4.1 Introduction to Radial Basis Functions

Traditional tensor product methods using polynomials or splines are not suitable for interpolating scattered data. Around 1970, Roland L. Hardy proposed the *Radial Basis Function* (RBF) method to interpolate multivariate scattered nodes [Hardy 1971]. Light [Light 2001] asserted that *Radial Basic Function* is a stricter terminology. However, due to its popularity, we still adopt the name *Radial Basis Function* in this thesis.

Given an arbitrary point $\mathbf{x} \in \Re^d$, RBF defines certain fixed centres $\{\mathbf{c}_j | \mathbf{c}_j \in \Re^d, j=1,\ldots, M\}$. A radial basis function is defined as,

$$\Phi_j(\mathbf{x}) = \phi(\| \mathbf{x} - \mathbf{c}_j \|) = \phi(r_j) \tag{4.1}$$

where $r_j = \left\| \mathbf{x} - \mathbf{c}_j \right\|$ denotes the Euclidian distance. For reconstruction of 3D surfaces, set d=2, because $\mathbf{x}$ and $\{\mathbf{c}_j\}$ only represent the abscissa of the points.

Then the function value f associated with the point $\mathbf{x}$ can be written as,

$$f(\mathbf{x}) = \sum_{j=1}^{M} w_j \phi(r_j) \tag{4.2}$$

where $\{w_j\}$ are weighting parameters.

RBF has several interesting properties [Barker 2004],

- RBF is uniquely solvable under rather mild conditions on the centres.
- RBF applies to scattered data.
- RBF applies to multivariate data in any dimension. The computational complexity of RBF reconstruction is $O[MN(M+d)]$, where N is the number of data points, M the number of centres and d the dimension.
- Centres can be appropriately chosen so that the approximation problem is well-posed.
- RBF is easy to implement.

Due to these superiorities, the RBF technique has become a standard tool of geometric data analysis in pattern recognition, statistical learning and neural networks.

Basis Functions

The performance of RBF interpolants relies heavily on the choice of the radial basis function $\phi(r)$. Table 4.1 lists some commonly used basis functions.

Name	Function
Linear	$\phi(r) = r$
Cubic	$\phi(r) = r^3$
Gaussian	$\phi(r) = \exp(-\alpha^2 r^2)$
Multiquadric (MQ)	$\phi(r) = (r^2 + \alpha^2)^{1/2}$
Inverse multiquadric (IMQ)	$\phi(r) = (r^2 + \alpha^2)^{-1/2}$
Thin plate spline (TPS)	$\phi(r) = r^2 \log r$

Table 4.1 Several commonly used radial basis functions

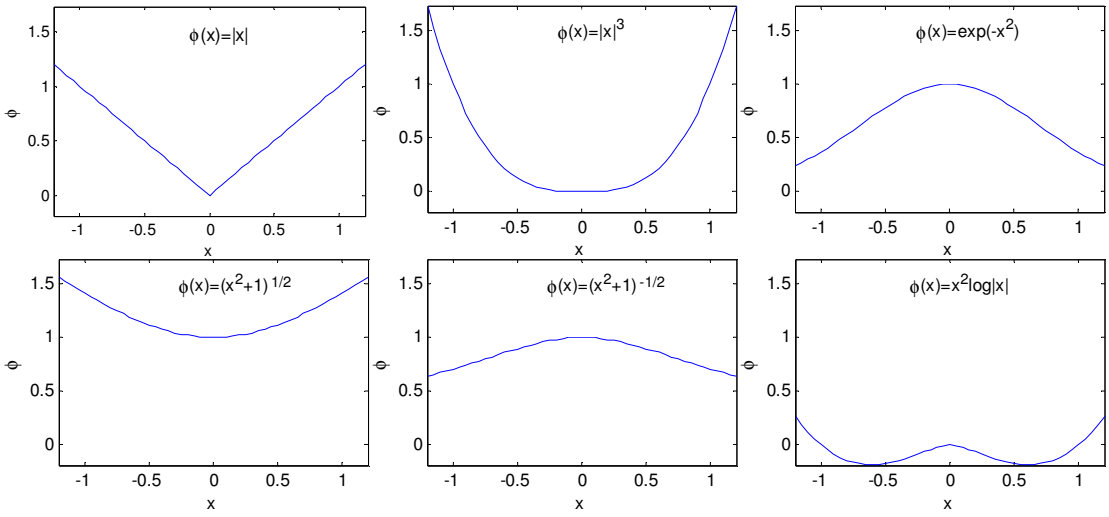


Figure 4.1 Some common radial basis functions

Franke [Franke 1982] compared 34 reconstruction algorithms for scattered data and found the RBF with multiquadric provides the highest accuracy. This method works best when the scale parameter α^2 is close to the average distance between the centres. But Powell found that a larger scale parameter is preferred when the centres form a regular grid [Baxter 1992]. By far, it is not clear yet how to select an optimal scale parameter for a general function.

The Gaussian basis function exhibits excellent smoothness properties. It has a local supporting advantage and a rapid decay. In addition, its design matrix is guaranteed to be positive definite if the centres are distinct, thus it has been extensively used in neural networks. But it is very sensitive to the scale parameter. When the scale parameter is not properly selected, its behaviour may be rather poor. In fact, for Gaussian, multiquadric, and inverse multiquadric basis functions, the scale parameter is always a key factor that determines the quality of the interpolated surface.

The thin plate spline (TPS) is proposed by Duchon [Duchon 1977]. This function is a fundamental solution of the bi-harmonic equation,

$$\Delta\phi(r) = 0 \tag{4.3}$$

where Δ is the Laplacian operator, e.g. a 2-D case,

$$\Delta\phi(r) = \frac{\partial^2\phi(r)}{\partial x^2} + \frac{\partial^2\phi(r)}{\partial y^2} \tag{4.4}$$

More general forms of TPS are given by

$$\phi_k^{(d)}(r) = \begin{cases} r^{2k-d}\log r & \text{if } k \geq d \text{ and } d \text{ is even} \\ r^{2k-d} & \text{if } k \geq d \text{ and } d \text{ is odd} \end{cases} \tag{4.5}$$

where d is the dimension of input nodes. The function is forced to have a value zero at the origin.

TPS function is the one which passes through the given data points with the minimum bending energy in the 2D case,

$$I(f) = \iint_{\Re^2} \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 dxdy \tag{4.6}$$

It suggests that TPS is the smoothest interpolant. Moreover, it is scale-independent. So TPS is well-suited for surface reconstruction of scattered data.

Wendland proposed a group of compactly-supported basis functions [Wendland 1971], which are defined as a product of a truncated power function and a polynomial of minimal degree k-1 in r,

$$\phi(r) = (1-r)_+^d P_k(r) \tag{4.7}$$

For instance, $P_k(r) = 1 + 4r$ when k=2. These functions are local supporting and can generate sparse design matrices, hence they are more efficient and numerically stable than the global basis functions such as TPS and multiquadric. However, these basis functions have discontinuous higher derivatives; thus they are not very suitable to approximate smooth functions.

Non-Singularity of the Design Matrix

Micchelli's theorem states that for distinct data points $\{\mathbf{x}_i\}$ and selection of particular radial basis functions, the design matrix **A** is non-singular [Micchelli 1986]. Thus one of the most attractive features of the RBF method is that a unique interpolant can often be guaranteed under rather mild conditions on the centres. In several important cases, the only restrictions are there exist at least two centres and they are all distinct. But TPS is an exception. Its design matrix may be singular even for non-trivial sets of distinct centres.

A low order polynomial is proposed to be augmented into the RBF system [Schaback 1995],

$$f(\mathbf{x}) = \sum_{j=1}^{M} w_j \phi(r_j) + p(\mathbf{x}) \tag{4.8}$$

If the degree of the polynomial is one, i.e. $p(\mathbf{x}) = a + bx + cy$, three additional constraints are required to eliminate the extra three degrees of freedom introduced by this polynomial,

$$\begin{cases} \sum_{j=1}^{M} w_j = 0 \\ \sum_{j=1}^{M} w_j x_j = 0 \\ \sum_{j=1}^{M} w_j y_j = 0 \end{cases} \tag{4.9}$$

where $[x_j, y_j] = \mathbf{c}_j$ is an arbitrary centre.

In this way, the design matrix can be ensured non-singular, even for TPS.

Compactly supported basis functions, e.g. Wendland functions have a sparse design matrix, thus are well-posed. But for globally defined radial basis functions, e.g. TPS and MQ, the resultant design matrix is dense and will be ill-conditioned when the number of

data points is greater than several thousand or there are near-coincident centres which are very near to each other. In order to overcome this ill-conditioning problem, Truncated SVD or Rank-Revealing QR Decomposition can be employed to calculate the weighting parameters from Equation (4.2) or (4.8).

4.2 Centre Selection

Given a group of function values $\{z_i\}$, i=1,…, N associated with some points $\{\mathbf{x}_i | \mathbf{x}_i \in \Re^d\}$, certain fixed centres $\{\mathbf{c}_j\}$, $\mathbf{c}_j \in R^d$, j=1,…, M are selected appropriately and a model is built as Equation (4.2). It is rewritten as,

$$\mathbf{z} = \mathbf{A}\mathbf{w} \tag{4.10}$$

Elements in the design matrix **A** are the corresponding basis functions,

$$A_{ij} = \phi(\| \mathbf{x}_i - \mathbf{c}_j \|) \tag{4.11}$$

Its least squares solution is,

$$\mathbf{w} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{z} \tag{4.12}$$

The most intuitive way to locate centres is directly taking all the data points as centres, which is called exact interpolation. When the data points are very dense, it may cause oscillations onto the curve or surface due to the noise and unconstraint at locations between the data points, although the reconstruction values at the input nodes still remain very accurate. This phenomenon is called an over-fitting problem [Bishop 1995]. See Figure 4.2. In this figure, the dots denote the input data, and the dashed and solid lines represent the original and fitted RBF curves respectively.

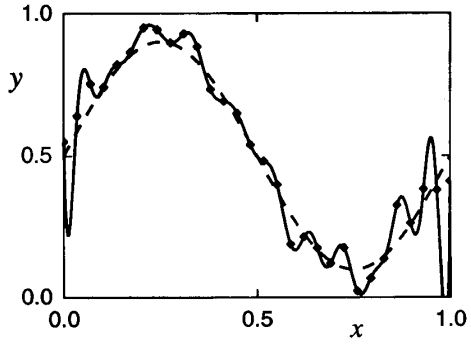


Figure 4.2 Over-fitting problem

There are two ways to avoid over-fitting. The first is regularisation, which augments the objective function of sum-of-squared-residual with a term which penalises large weights [Orr 1996],

$$E = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 + \lambda \sum_{j=1}^{M} {w_j}^2 \tag{4.13}$$

i.e. $$E = \mathbf{e}^T \mathbf{e} + \lambda \mathbf{w}^T \mathbf{w} \tag{4.14}$$

This approach is also known as zero-order regularisation or ridge regression. The solution of Equation (4.14) is,

$$\mathbf{w} = (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^T \mathbf{z} \tag{4.15}$$

where $\mathbf{I}$ is an $M \times M$ identity matrix. The regularization parameter $\lambda > 0$ controls the balance between the accuracy of fit and the smoothness of the surface. It is chosen in advance or estimated from the data. Orr [Orr 1996] calculated the regularization parameter λ by cross-validation. More details on regularization can be found in the PhD thesis [Björkström 2007].

The second approach to avoid over-fitting is to allow only a subset of the candidate centres, i.e. to employ centre selection techniques. When the centres are not the same with the input nodes, the linear systems are rarely singular [Fornberg 2002]. Therefore, the incremental polynomial will not be considered in this situation.

If the data is on a uniform regular grid, the centres are also arranged on a regular grid employing an appropriate space. Given an arbitrary scattered point cloud, Broomhead [Broomhead 1988] chose centres randomly from the input data points, but the reconstruction accuracy cannot be guaranteed in this way. Orr [Orr 1996] adopted a forward selection method, in which centres are chosen one by one from the candidate point locations until some criterion is satisfied, and the ridge regression technique is involved as well. Other centre selection methods, such as geometric selection [Valdés 1999], immunological approach [de Castro 2001], hierarchical clustering [Crampton 2002], the predicted residual sum of squares [Chen 2004], Voronoi method [Samozino 2006] etc have also been developed.

Recently Sheng Chen et al [Chen 2008] proposed an orthogonal least squares basis hunting (OLS-BH) method to select centres for RBF surfaces. It is introduced in detail as follows.

Given a set of candidate centres $\{\mathbf{c}_j\}, i = 1, \cdots, M$, and input nodes $\{\mathbf{x}_i\}, i = 1, \cdots, N$, the design matrix constructed by them is $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_M] \in \Re^{N \times M}$ and the resulting RBF system is $\mathbf{Aw} = \mathbf{z}$. Here any column $\mathbf{a}_j$ in $\mathbf{A}$ corresponds to one centre $\mathbf{c}_j$. The matrix $\mathbf{A}$ can be decomposed into the multiplication of an orthogonal matrix $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_M] \in \Re^{N \times M}$ and an upper diagonal matrix $\mathbf{R} \in \Re^{M \times M}$,

$$\mathbf{A} = \mathbf{QR} \tag{4.16}$$

with $\mathbf{R} = \begin{bmatrix} 1 & R_{12} & \cdots & R_{1M} \\ & 1 & \cdots & R_{2M} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}$. So that,

$$\mathbf{Aw} = \mathbf{QRw} = \mathbf{Q\Lambda} = \mathbf{z} \tag{4.17}$$

In the equation, $\mathbf{\Lambda} = \mathbf{Rw} = [\lambda_1, \lambda_1, \cdots \lambda_M]^T$ is the new weighting vector in the orthogonal space **Q**.

The least squares error of reconstruction is,

$$E = (\mathbf{z} - \mathbf{Q\Lambda})^T (\mathbf{z} - \mathbf{Q\Lambda}) = \mathbf{z}^T \mathbf{z} - \sum_{j=1}^{M} \mathbf{q}_j^T \mathbf{q}_j \lambda_j^2 \tag{4.18}$$

If selecting centres recursively one by one from the candidate set, at the *k*-th stage the total error will be reduced as,

$$E_k = E_{k-1} - \mathbf{q}^{(k)T} \mathbf{q}^{(k)} (\lambda^{(k)})^2 \tag{4.19}$$

Here $\mathbf{q}^{(k)}$ is the deign vector and $\lambda^{(k)}$ is weight associated with the newly selected centre $\mathbf{c}^{(k)}$. Therefore, minimizing the total error *E* is equivalent to maximizing the error $e_k = \mathbf{q}^{(k)T} \mathbf{q}^{(k)} (\lambda^{(k)})^2$ each time. Before selection the centre set is null and the initial total error of the system is $E_0 = \mathbf{z}^T \mathbf{z}$.

Assuming *k*-1 centres have been determined already, we present the pseudo-code of the OLS-BH algorithm of selecting the *k*-th centre.

Set N_p: the population size of randomly sampled candidate centres;
N_g: the number of generations in repeated search;
N_s: the search iteration for each candidate centre

ξ_B: the criterion to terminate the weighted boosting search;

for m=1:1: N_g

Initialize the first candidate centre $\mathbf{c}_1^{(m)} = \mathbf{c}_{best}^{(m-1)}$;

%$\mathbf{c}_{best}^{(m-1)}$ is the optimal candidate centre found at the previous generation

%if m=1, $\mathbf{c}_1^{(m)}$ is randomly chosen.

Randomly generate the rest of the candidate centres $\mathbf{c}_i^{(m)}, 2 \le i \le N_p$;

Initialize the distribution weights $\delta_i^{(0)} = 1/N_p, 1 \le i \le N_p$;

for i=1:1: N_p

Calculate the vector of the design matrix $\mathbf{a}_k^{(i)}$ for $\mathbf{c}_i^{(m)}$;

$$\alpha_{j,k}^{(i)} = \frac{\mathbf{q}_j^T \mathbf{a}_k^{(i)}}{\mathbf{q}_j^T \mathbf{q}_j}, 1 \le j \le k-1 \tag{4.20}$$

$$\mathbf{q}_k^{(i)} = \mathbf{a}_k^{(i)} - \sum_{j=1}^{k-1} \alpha_{j,k}^{(i)} \mathbf{q}_j \tag{4.21}$$

%Gram–Schmidt orthogonalization

end

for i=1:1: N_p

$$\lambda_k^{(i)} = \frac{\left(\mathbf{q}_k^{(i)}\right)^T \mathbf{z}}{\left(\mathbf{q}_k^{(i)}\right)^T \mathbf{q}_k^{(i)}} \tag{4.22}$$

$$E_k^{(i)} = E_{k-1} - \left(\mathbf{q}_k^{(i)}\right)^T \mathbf{q}_k^{(i)} \left(\lambda_k^{(i)}\right)^2 \tag{4.23}$$

%calculate cost function associated with each candidate centre

for t=1:1: N_s

$i_{best} = \arg\min_{1 \le i \le N_p} E_k^{(i)}$ and $i_{worst} = \arg\max_{1 \le i \le N_p} E_k^{(i)}$;

Denote $\mathbf{c}_{best}^{(m)} = \mathbf{c}_{i_{best}}^{(m)}$ and $\mathbf{c}_{worst}^{(m)} = \mathbf{c}_{i_{worst}}^{(m)}$;

%find the best and the worst candidate centres

$$\overline{E}_k^{(i)} = \frac{E_k^{(i)}}{\sum_{j=1}^{N_p} E_k^{(j)}}, 1 \le i \le N_p \tag{4.24}$$

%normalize the errors

$$\eta_t = \sum_{i=1}^{N_p} \delta_i^{(t-1)} \overline{E}_k^{(i)}, \beta_t = \frac{\eta_t}{1-\eta_t} \tag{4.25}$$

%Compute the weighting factors

$$\delta_i^{(t)} = \begin{cases} \delta_i^{(t-1)} \beta_t^{\overline{E}_k^{(i)}}, \beta_t \le 1 \\ \delta_i^{(t-1)} \beta_t^{1-\overline{E}_k^{(i)}}, \beta_t > 1 \end{cases}, 1 \le i \le N_p \tag{4.26}$$

% Update the distribution weights

$$\delta_i^{(t)} = \frac{\delta_i^{(t)}}{\sum_{j=1}^{N_p} \delta_j^{(t)}}, 1 \le i \le N_p \tag{4.27}$$

%normalize the weights

$$\mathbf{c}_{N_p+1} = \sum_{i=1}^{N_p} \delta_i^{(t)} \mathbf{c}_i^{(m)} \tag{4.28}$$

% construct the (N_p+1)-th candidate centre

$$\mathbf{c}_{N_p+2} = 2\mathbf{c}_{best}^{(m)} - c_{N_p+1} \tag{4.29}$$

% construct the (N_p+2)-th candidate centre

Calculate design matrices $\mathbf{a}_k^{(N_p+1)}$ and $\mathbf{a}_k^{(N_p+1)}$;
Orthgonalize them as Equations (4.20) and (4.21);

$$i_* = \arg \min_{i=N_p+1,\ N_p+2} E_k^{(i)}; \tag{4.30}$$

Replace $\left(\mathbf{c}_{worst}^{(m)}, E_k^{(i_{worsrt})}\right)$ with $\left(\mathbf{c}_{i_*}, E_k^{(i_*)}\right)$;
If $\left\|\mathbf{c}_{N_p+2} - \mathbf{c}_{N_p+1}\right\| < \xi_B$
%termination criterion is satisfied
break
end
end
end

So that the optimal candidate centre $\mathbf{c}_{best}^{(N_g)}$ is selected as the *k*-th centre.

This procedure is repeated until the total error E_k is less than a user-set threshold. One manifest benefit of this searching strategy is that each time two extra candidate centres can be generated from Equations (4.28) and (4.29), so that the resulting centres

are not restricted to be the initial candidate centres. In order to ensure the numerical stability of this linear system, we generate a set of uniform candidate points within the domain of interest using an appropriate spacing. Then in each generation N_p candidates are randomly sampled from this point set. If the newly generated candidate centre $\mathbf{c}_{N_p+1}$ or $\mathbf{c}_{N_p+2}$ is too close to the already selected centres $\{\mathbf{c}_j\}, j=1,2,\cdots k-1$, this candidate centre will be neglected.

Through this approach much fewer centres are required to construct this RBF system, so that the size of the design matrix will be greatly reduced.

Figure 4.3 plots the flowchart of the OLS-BH point selection algorithm.

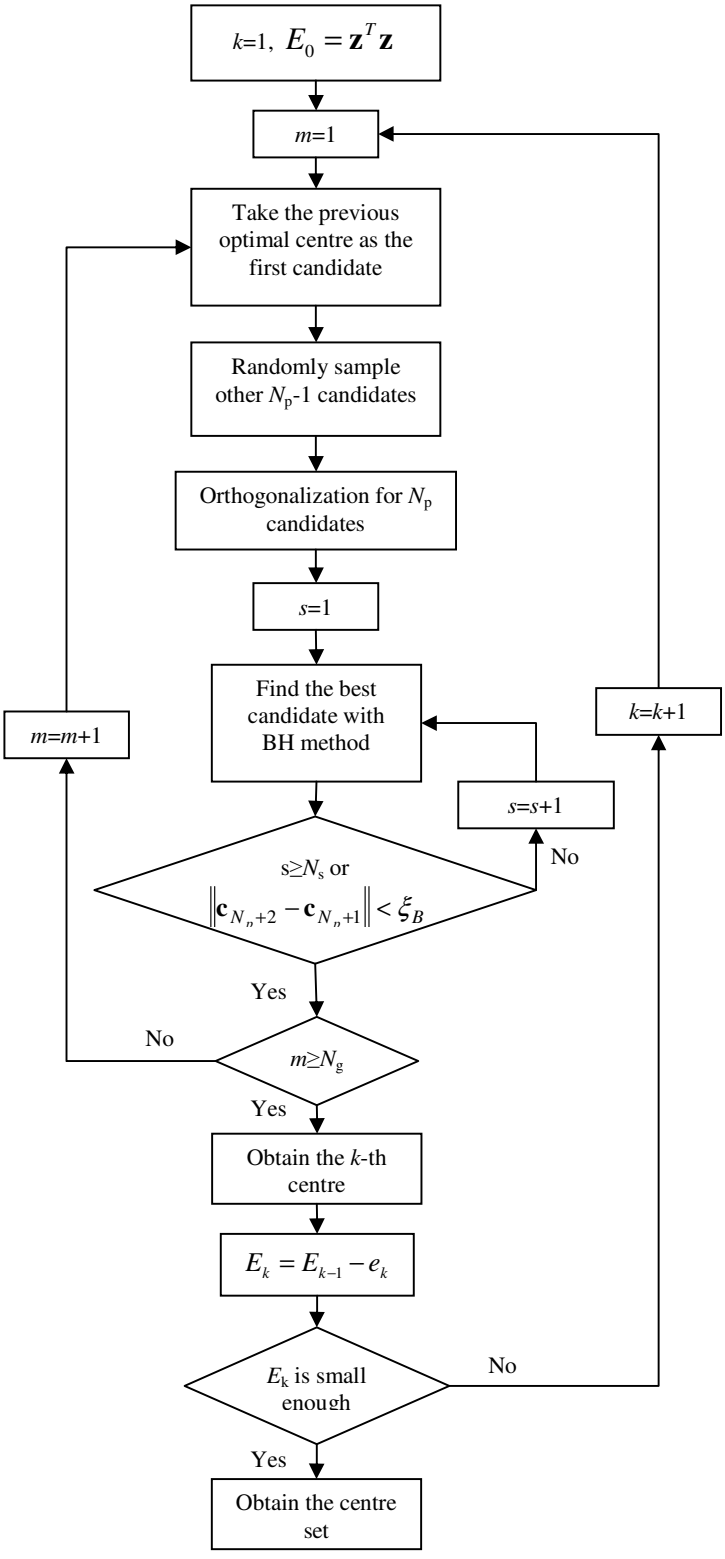


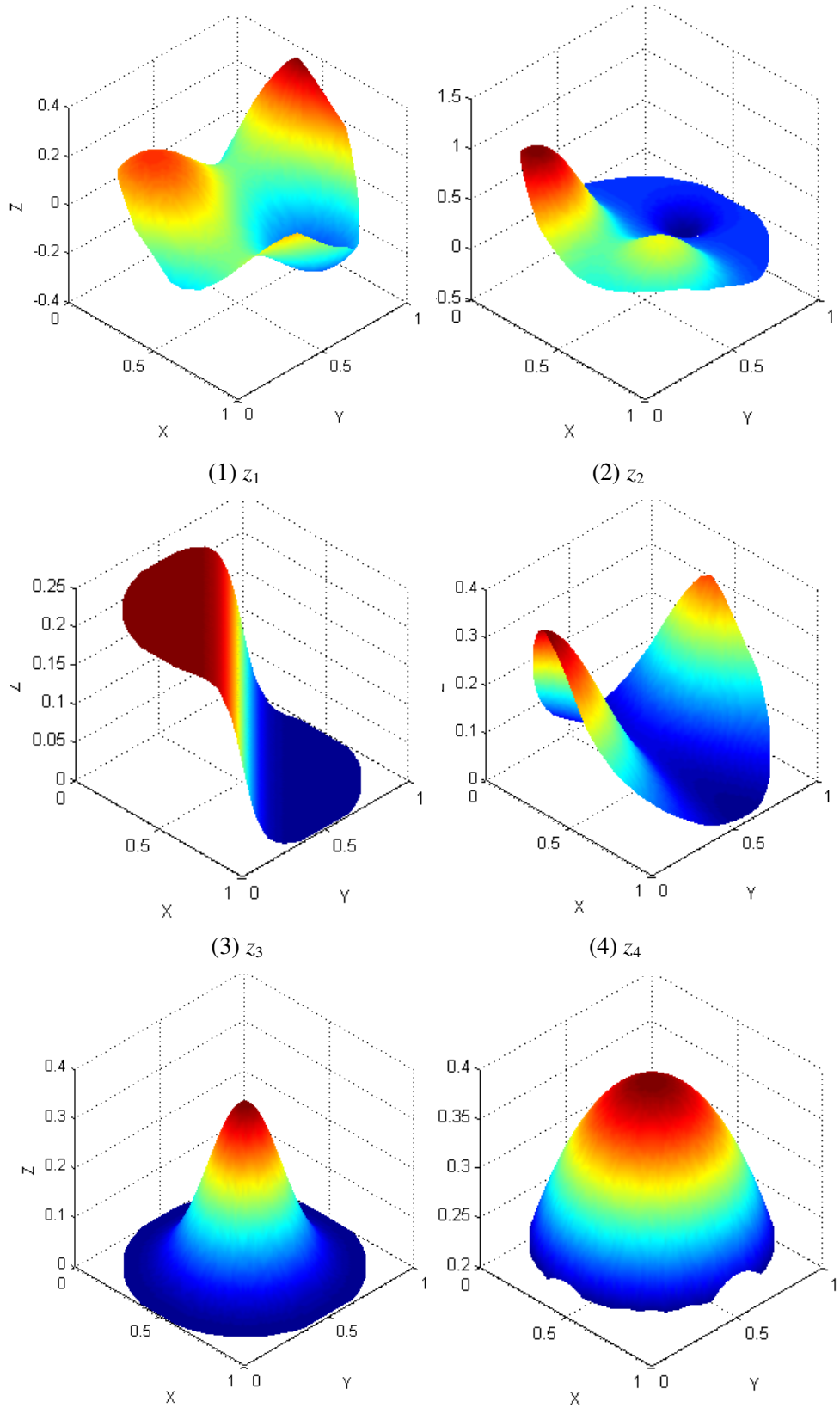
Figure 4.3 Flowchart of the OLS-BH centre selection algorithm

4.3 Boundary Effect

The surface used in precision metrology is generally an open surface patch. The reconstruction accuracy near the surface boundary will be degraded compared with the interior area. Hangelbroek et al proved that the reconstruction error in the interior region of smooth surfaces may attain $O(h^4)$ when adopting the thin plate splines as basis functions, whereas at the boundary it is not better than $O(h^{5/2})$ in the sense of l_2 norm. Here the fill distance h is defined to measure the density of the data points [Hangelbroek 2007]. This effect seriously limits the application of the RBF method, especially when the boundary information is of our particular interest. Almost all the proposed approaches deal with this problem by changing the arrangement of the boundary centres, e.g. using a larger density for outer centres [Hangelbroek 2007], deploying some extra centres outside the domain of data [Fedoseyev 2002, Morandi 2002] or moving the boundary centres outward [Fornberg 2002]. However, the relationship between the accuracy and the centre density and/or moving distance has not been clearly indicated.

(a) Comparison of Some Common Boundary Treatments

The condition numbers of the observation matrices formed by infinitely smooth basis functions like Gaussian, multiquadric etc are terribly large compared to non-smooth basis functions like TPS [Schaback 1995]. Additionally, their scaling is a crucial issue for accuracy and stability. To concentrate on the influence of the centre distribution, we adopt the TPS as a basis function. The following six typical smooth functions are selected as test surfaces to investigate the behaviours of RBF in different situations [Franke 1982, Lee 1997], as presented in Figure 4.4.



(1) z_1 (2) z_2

(3) z_3 (4) z_4

(5) z_5 (6) z_6

Figure 4.4 Test surfaces

The functions of the six test surfaces are,

$$\begin{cases} z_1 = & 4x^3+4y^3-6.4x^2-5.4y^2+2.72x+1.72y-0.27 \\ z_2 = & 0.75\exp\left[-\dfrac{(9x-2)^2+(9y-2)^2}{4}\right]+0.75\exp\left[-\dfrac{(9x+1)^2}{49}-\dfrac{(9y+1)^2}{10}\right] \\ & +0.5\exp\left[-\dfrac{(9x-7)^2+(9y-3)^2}{4}\right]-0.2\exp\left[-(9x-4)^2-(9y-7)^2\right] \\ z_3 = & [\tanh(9-9x-9y)+1]/9 \\ z_4 = & \dfrac{1.25+\cos(5.4y)}{6+6(3x-1)^2} \\ z_5 = & \exp\left\{-20.25[(x-0.5)^2+(y-0.5)^2]\right\}/3 \\ z_6 = & \sqrt{\dfrac{64}{81}-(x-0.5)^2-(y-0.5)^2}-0.5 \end{cases} \tag{4.31}$$

Data points are sampled uniformly in the domain of a unit disc $(x-0.5)^2+(y-0.5)^2 \le 0.25$ with spacing h=0.035. Centres are also uniformly selected within this domain with a greater spacing H=0.05. The residuals at the input nodes cannot completely reflect the reconstruction quality due to the over-fitting phenomenon, hence we sample evaluation points in the domain of interest with a smaller spacing h_1=0.015. The reconstruction error with respect to the ideal test surface is depicted in Figure 4.5. For the purpose of quantitative comparison, the boundary region is defined as a narrow annular region with a width w=0.15. The fitting errors at the interior and outer areas are assessed separately.

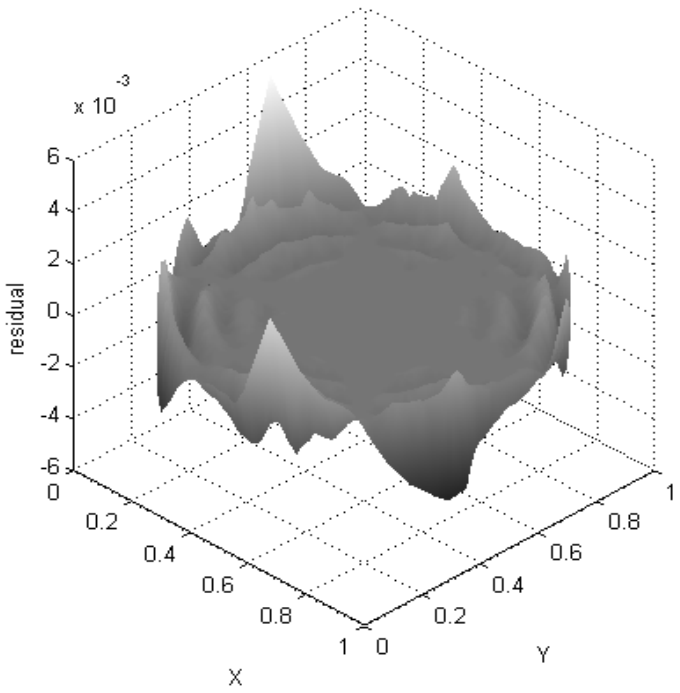


Figure 4.5 Initial reconstruction residual

Six boundary treatments are presented, as listed in Figure 4.6.

(I) Adding one circle of centres outside the domain of interest;

(II) Adding two circles of centres outside the domain of interest;

(III) Moving the outmost one circle of centres outward with a distance $\delta = H$;

(IV) Moving the outmost two circles of centres together with a distance $\delta = H$;

(V) Moving the outmost two circles of centres together with a distance $\delta = 2H$ and

(VI) Moving the outmost two circles with distances $\delta_1 = 2H$ and $\delta_2 = H$ respectively.

The l_2 norm condition number (Cond) of the design matrix is adopted to measure the numerical stability of each case, as listed in Table 4.2. For comparison, the initial case without boundary treatment is called Case 0. It can be seen that all the six condition numbers are worse than Case 0, especially Case 2. This means the numerical stability is degraded. To make the solutions more trustworthy, Truncated SVD is applied for Cases II and V, and QR Decomposition for the rest cases.

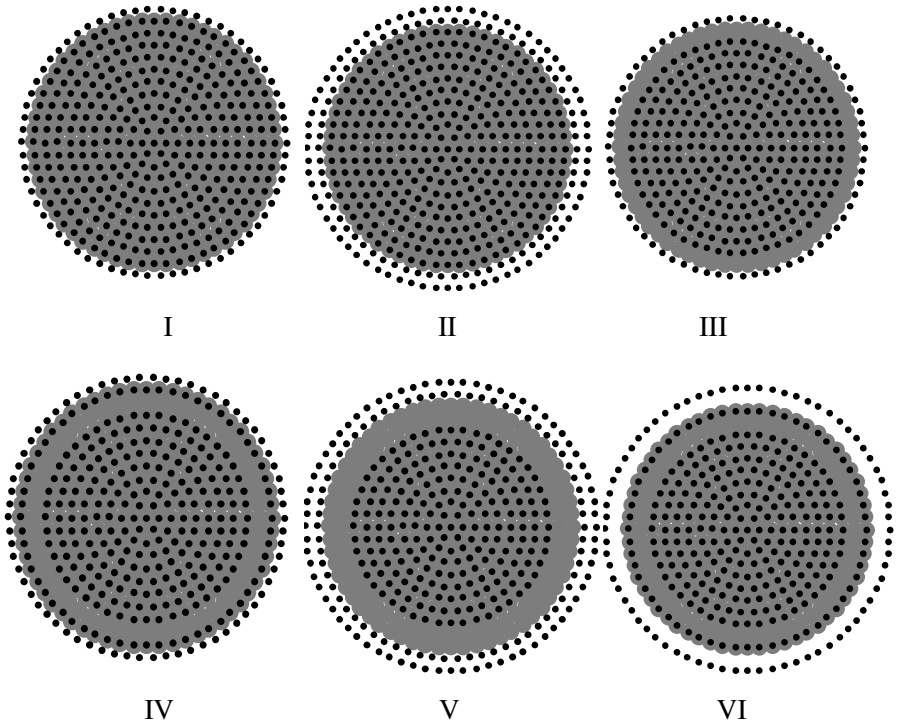
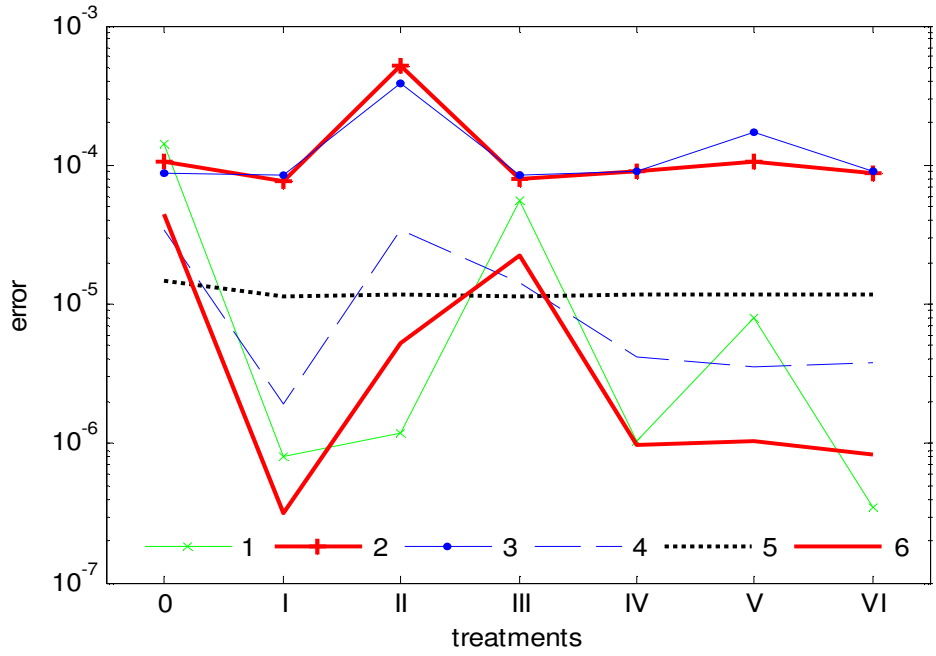


Figure 4.6 Centre arrangements

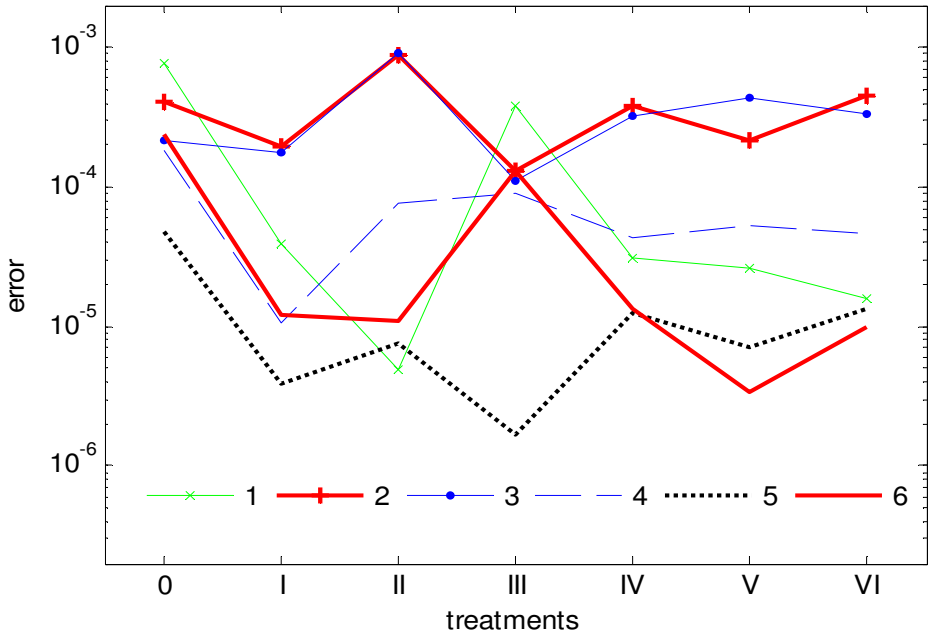
Case	0	I	II	III	IV	V	VI
Cond	1.39×10^{5}	1.79×10^{7}	1.78×10^{12}	1.00×10^{6}	5.87×10^{6}	4.10×10^{8}	6.25×10^{7}

Table 4.2 Condition numbers of different treatments

We found that S_a (arithmetical mean height), S_q (root mean square error) and S_z (maximum height error) show very similar behaviours, thus only S_q values are presented here, see Figure 4.7. Cases IV and V are termed 'Not-a-Knot' and 'Super Not-a-Knot' respectively by Fornberg et al [Fornberg 2002]. Moving boundary centres outward as Cases IV and V does not necessarily improve the reconstruction quality, such as in Surfaces 2 and 3. Case I can greatly improve the fitting accuracy both at the inner and outer areas of all the six test surfaces. It is proved to be the most reliable method. Therefore we adopt this technique for boundary improvement. It is also apparent that the influence of boundary enhancement techniques onto the inner area is in positive correlation with the slope at the boundary region. That means when the boundary is relatively planar and varies slightly, the technique works well, and vice versa.



(a) Interior errors



(b) Boundary errors

Figure 4.7 Boundary errors of different treatments

(b) Factors Influencing the Boundary Behaviour

In the previous section, adding one circle of new centres is found to be the best boundary enhancement technique. The distance from the added circle to the region boundary and the spacing between the added centres are fixed to be *H*. Now we

investigate the relationship between the fitting quality and the distribution of the centres, i.e. the number N of the new centres and the distance δ from the boundary to the added circle.

The corresponding condition numbers associated with different N and δ are plotted in Figure 4.8. With N and δ increasing, the condition number increases exponentially, thereby degrading the stability. For this reason we adopt Truncated SVD to solve the weighting parameters.

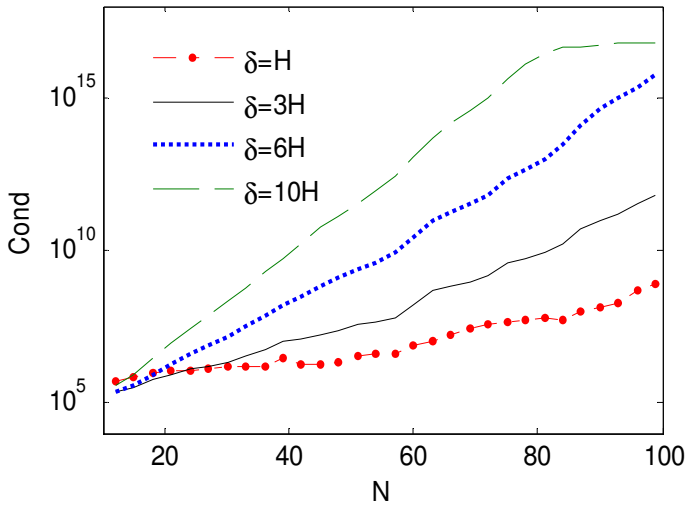


Figure 4.8 Condition numbers for different N and δ

Now we switch the distance δ from H to $15H$ and simultaneously change the point number N from 12 to 120, the optimal result at each δ is recorded in Figure 4.9. Surfaces 2, 4, and 5 achieve the best result at the interval $\delta \in [2H, 4H]$, whilst Surfaces 3 and 6 prefer a smaller δ value, and $\delta = 8H$ is the best choice for Surface 1. When δ is large enough, all the six surfaces behave very steadily and remain nearly unchanged.

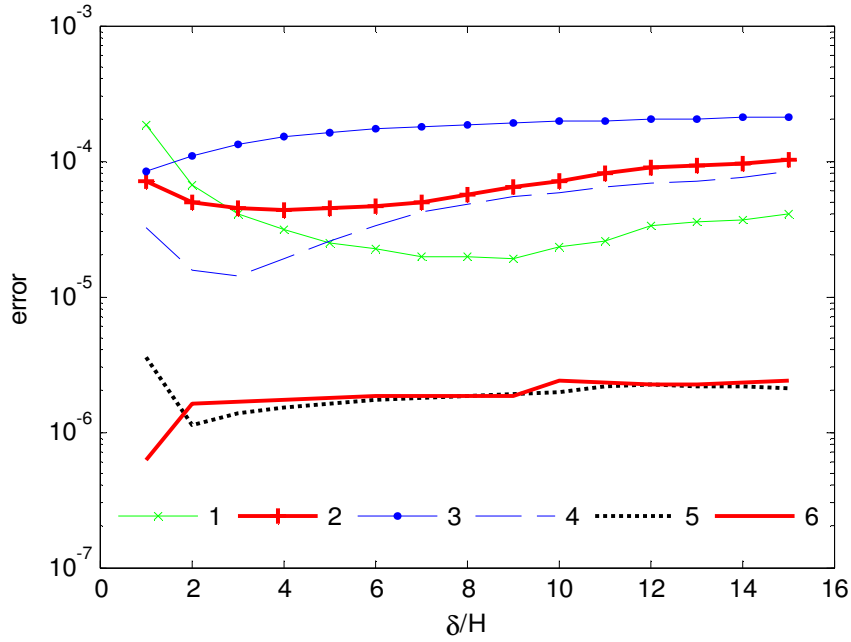
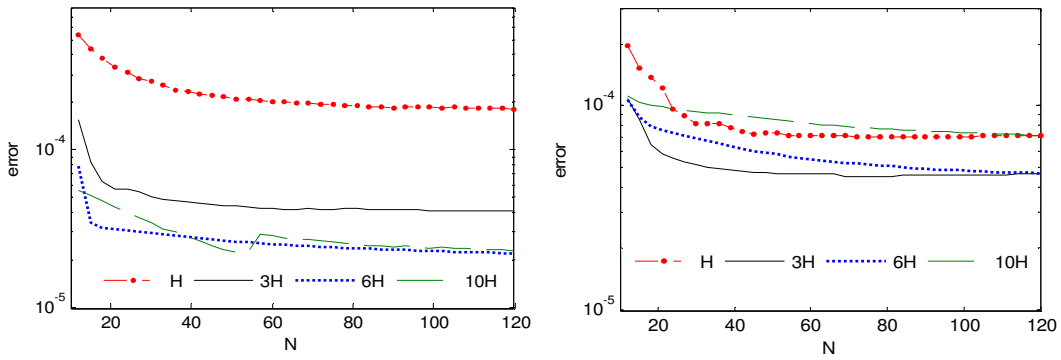
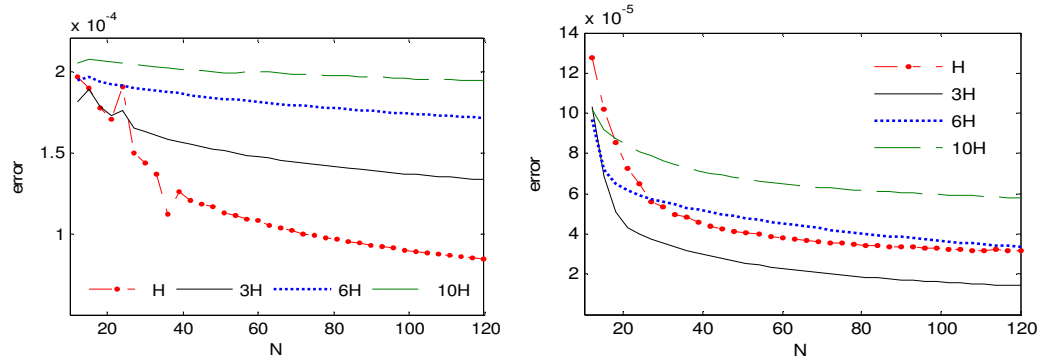


Figure 4.9 Optimal results for different δ

Again for each surface, we select $\delta = H, 3H, 6H$ and $10H$ respectively and change the point number N. The resultant S_q curves are plotted in Figure 4.10. When $\delta = H$, S_q is very sensitive to the added point number N, and a larger N is preferable for all test functions. With δ increasing, the reconstruction quality is less and less sensitive to N and differentiated by surface shapes. Therefore it is impossible to give an optimal δ and N which are always the best choice in all situations. Taking the numerical stability into account, we select δ=3H and N=40. In this case, the corresponding spacing between the added new centres is $\varepsilon = 2H$.

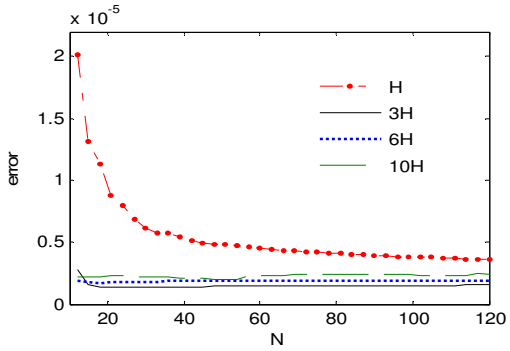


(a) Surface 1 (b) Surface 2

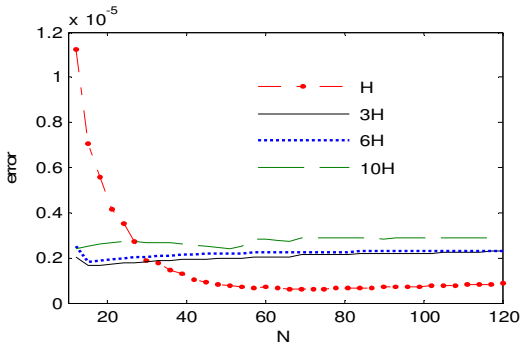


(c) Surface 3

(d) Surface 4



(e) Surface 5



(f) Surface 6

Figure 4.10 S_q values for different N and δ

To clarify the effect of the aforementioned optimal treatment, the abscissa domain is divided into six annular regions. At each region the error S_q is calculated individually. We work out the quotient between the S_q of Case 0 and the optimal boundary treatment in each region for the six test surfaces, as plotted in Figure 4.11. The effect of this technique is concerned with the surface shape. The amount of accuracy improvement at the boundaries of the six surfaces can be sorted in this order: Surfaces 6> 5> 1> 2> 4> 3. It is interesting to note that the height variations of the six boundary curves descend exactly in this order. In another word, the effect of this technique is in negative correlation with the boundary height variations. Thus for surfaces with planar boundary curves, it is an appropriate approach to add extra centres outside to improve the boundary accuracy.

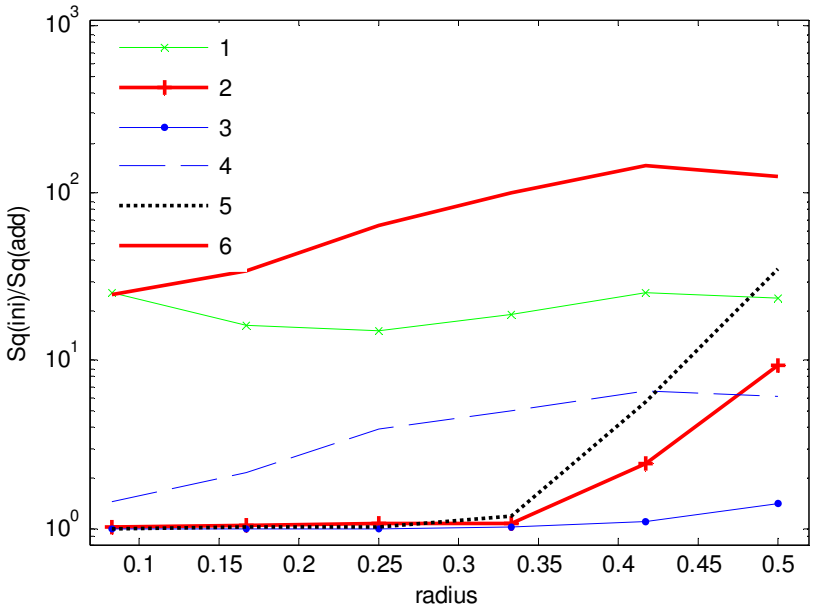


Figure 4.11 Accuracy improvements at different parts

(c) How to Get the Boundary Points?

For RBF reconstruction, the input data points are usually scattered in the domain of interest, as a result it is not easy to obtain the boundary points and add extra centres outside. In the case of 3D surface fitting, the abscissa domain is a 2D area. In 1972, Ronald L. Graham proposed an optimal algorithm, called the *Graham scan* to compute the convex hull of a set of discrete points [Graham 1972]. When a domain is convex, its convex hull can be respected as the boundary. However, the convexity of the boundary cannot always be guaranteed in practice. We improved the ordinary Graham scan algorithm. The new program can find the boundary points as long as the domain of interest is connected without holes, and the boundary is a closed curve without crossings.

Given a point set $\mathbf{P}=\{\mathbf{p}_i\}, i=1,2,...,N$, this algorithm attempts to find the boundary point set $\mathbf{Q}=\{\mathbf{q}_j\}, j=1,2,...,M$. The boundary points will be searched with a counter-clockwise order, thus there should be no sharp right-turn between them. Once the turning angle between $\mathbf{q}_{k-2}\mathbf{q}_{k-1}$ and $\mathbf{q}_{k-1}\mathbf{q}_k$ is smaller than a user-set threshold, say -60˚, as depicted in Figure 4.12, $\mathbf{q}_{k-1}$ is not a real boundary point and will be removed from the boundary point set.

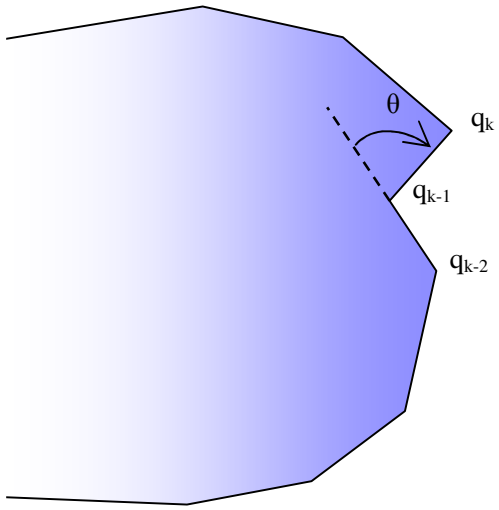


Figure 4.12 Sharp right-turn check

The pseudo-codes are shown below,

```
Find the rightmost lowest point q_1;
Sort other points by polar angles in a counter-clockwise order
around q_1. If more than one point has the same angle, remove all
but the farthest one from q_1;
Add q_1 at the rear of the point list, consequently form a new
point set P~={p~_i},i=1,2,...,N';
Add q_1, p~_1, p~_2 into Q;
Initiate the number k of the points contained in Q to be 3;
for i=2:N'
    while k≥3
        if there is a sharp right-turn between q_{k-2}q_{k-1} and q_{k-1}q_k
            q_{k-1} ← q_k;
            k ← k-1;
            % q_{k-1} is not a boundary point, remove it
        else
        %q_{k-1} passes the test at this step
            k ← k+1;
            q_k ← p~_{i+1};
            % check the next point
            break
            % If q_{k-1} is not removed, it will be kept in Q and thought as
a
            % boundary point
```

```
        end
    end
end
```

The median distance between the adjacent points of the boundary **Q** is taken as the average spacing H of the interior centres. The polygonal lines connecting these points may be very irregular, thus causing the reconstruction domain to be non-regular as well. The boundary polygon can be fitted into a closed smooth curve. Then some new boundary centres are sampled on the curve and moved outward an appropriate distance along the local normal vectors of the curve. These moved points are added as new auxiliary centres of the RBF system.

4.4 Numerical Examples

Example 1 Verification of RBF Surface Reconstruction

Here we carry on the reconstruction of the meniscal bearing component in Section 3.4. To avoid sharp corners, 1733 points are sampled with spacing 0.5 mm in an elliptical area on the CAD model using the software HOLOS, as illustrated in Figure 4.13.

Firstly we check the behaviour of the RBF exact interpolation, i.e. directly employing all the 1733 input nodes at centres. TPS is adopted as the basis function. To ensure the numerical stability, a linear polynomial is augmented in the reconstruction function and the Rank-Revealing QR Decomposition is utilized to solve the weighting vector. At the positions of the input data, the reconstruction error is as small as 10^{-10} μm. Evidently, it is only caused by the numerical round-off error of the MATLAB program. However this does not suggest that this RBF surface is very accurate. If sampling some new evaluation points with spacing 0.2 mm on the CAD model and at the same locations on this RBF surface, the relative deviations between their z coordinates turn out to be very large, especially at the boundary. It clearly reveals the over-fitting phenomenon and the poor boundary performance of the RBF exact interpolation.

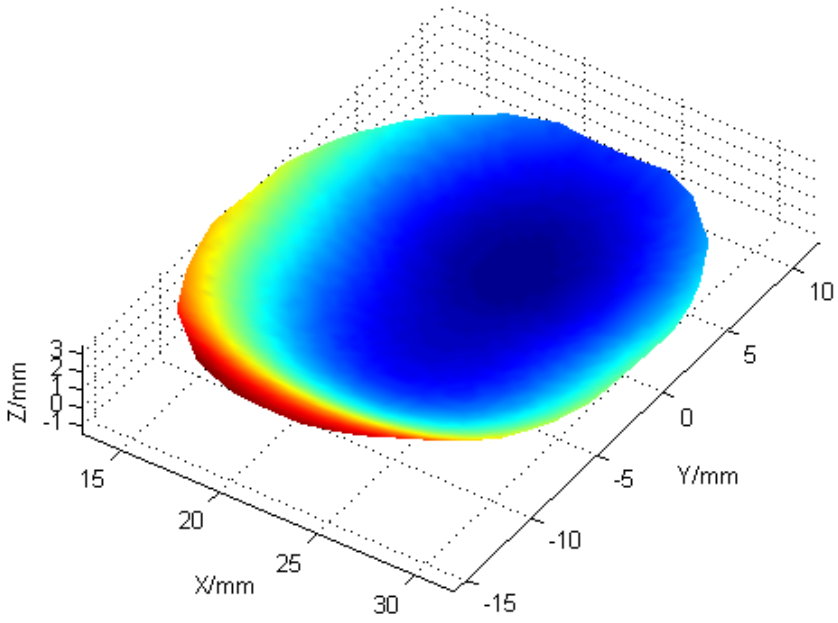
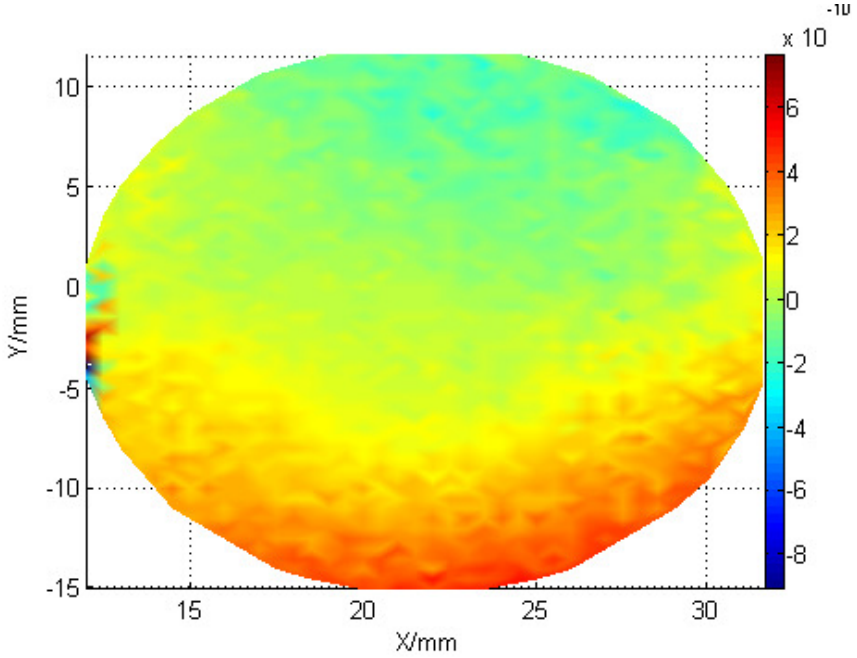
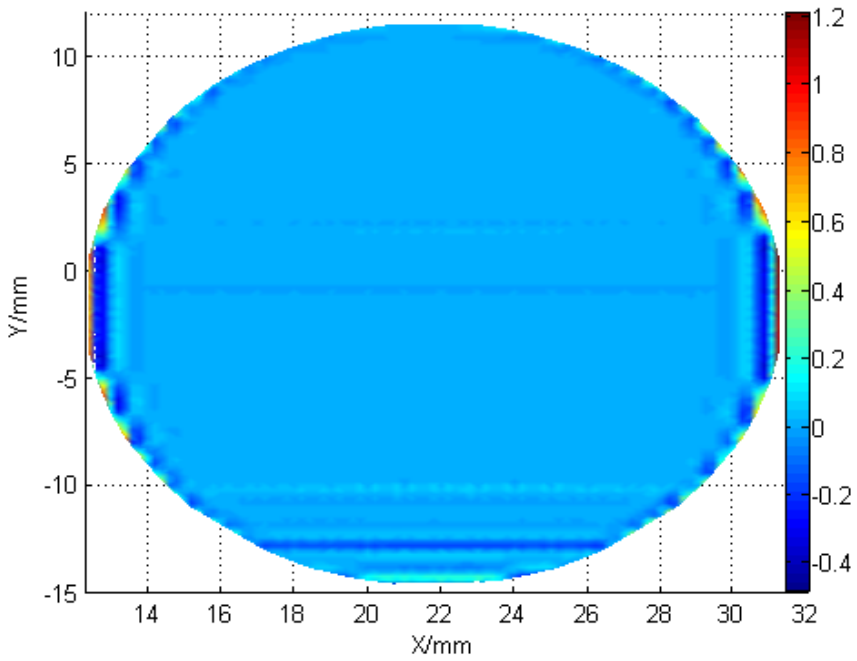


Figure 4.13 Reconstruction area



(a) Fitting residuals at the input nodes (μm)



(b) Evaluation errors (μm)

Figure 4.14 Reconstruction errors of RBF exact interpolation

Then we use a new set of RBF centres to improve the reconstruction accuracy. The input nodes are evenly distributed on a smooth surface, thus the centres are also evenly sampled with spacing 0.6 mm within the domain of interest. In addition, a circle of auxiliary centres are placed outside to overcome the boundary effect, in accordance with the boundary enhancement technique proposed in Section 4.3. Figure 4.15 depicts these 1232 RBF centres. It is worth noting that to make sure the uniqueness of the solution, the number of the weighting, i.e. the centre number, cannot exceed the input data. Hence the spacing between the interior centres should always be greater than the input data.

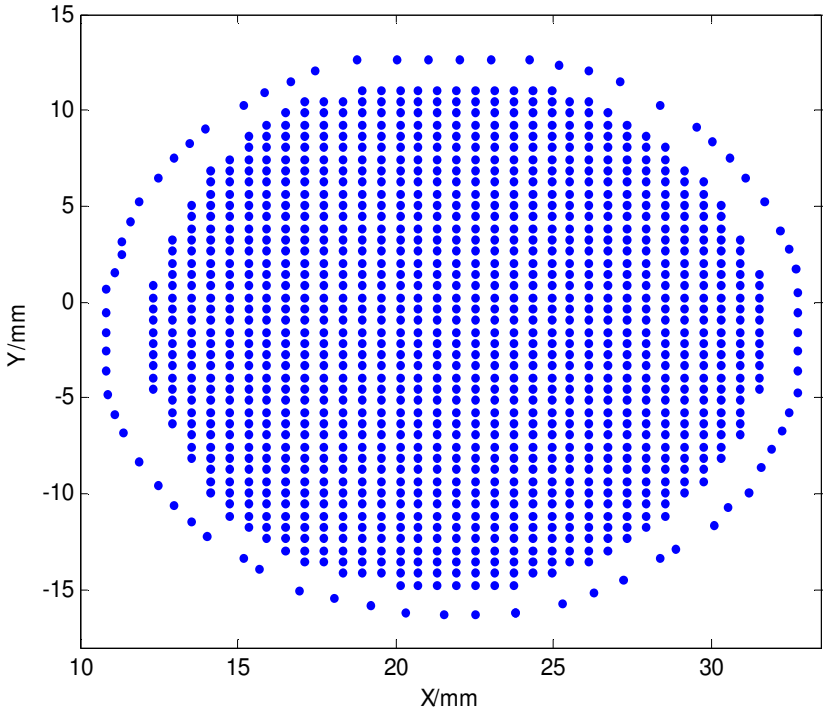
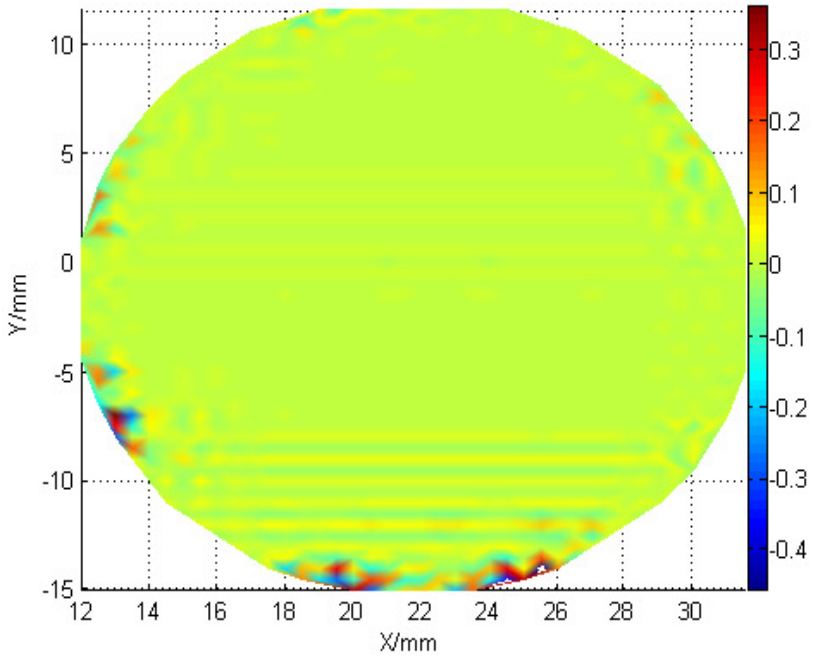
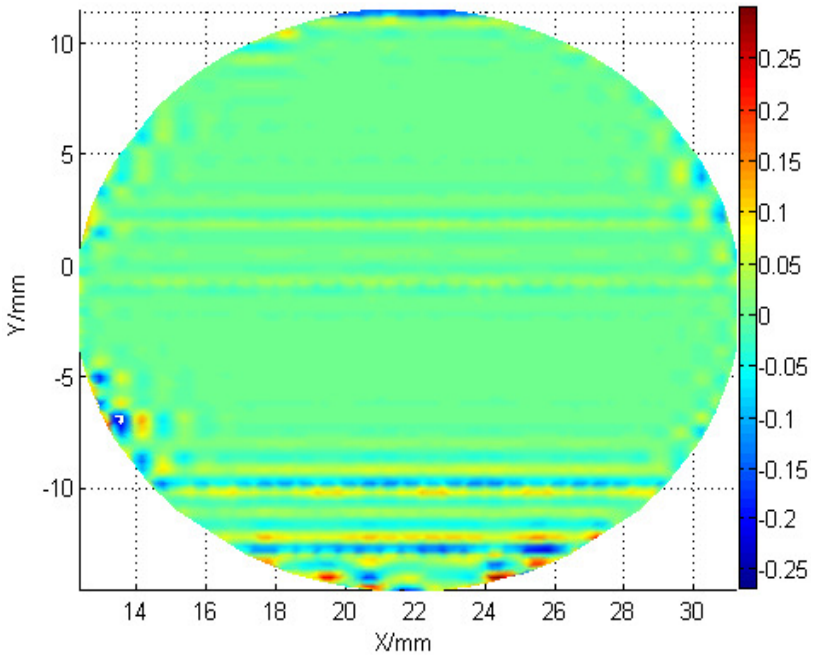


Figure 4.15 Sampled uniform centres

The reconstruction residuals at the locations of input data and evaluation points are shown in Figure 4.16 (a) and (b) respectively.



(a) Deviations at the input nodes (μm)



(b) Evaluation errors (μm)

Figure 4.16 Reconstruction errors of RBF approximation

In order to compare the fitting accuracy quantitatively, the corresponding error parameters are calculated for the exact interpolation and approximation, as listed below.

Reconstruction		Exact interpolation	Approximation
Number of centres		1733	1232
Fitting Errors /μm	S_a	1.281e-10	0.015
	S_q	1.698e-10	0.037
	S_z	1.670e-9	0.816
Evaluation Errors /μm	S_a	0.0213	0.014
	S_q	0.088	0.029
	S_z	1.687	0.567
Reconstruction time /sec		8.611	6.441
Evaluation time /sec		6.445	4.376

Table 4.3 Comparison of reconstruction errors

Due to the reduction of the centre number, the RBF system becomes over-determined, that is to say, the fitting values at the input data cannot be all satisfied. Their deviation now comes to the order of 0.1 μm. But the accuracy of the whole surface is significantly improved, achieving the order of 0.1μm as well. The errors at the boundary approach the

same order with the interior area. This effectively proves the validity of the boundary enhancement technique.

Furthermore, sparser centres are sampled and a smaller design matrix is built, thereby speeding up the surface reconstruction and numerically stabilizing the system. In the table we can see that the reconstruction and evaluation of the new RBF system are both about two seconds quicker than the exact interpolation. In fact, the complexity of evaluation is proportional to the numbers of the evaluation points and centres, whilst the running time of matrix inversion when calculating the weighting vector is in the order of $O(M^2N\text{-}M^3/3)$, as stated in Section 2.4. Here M and N indicate the numbers of the input data and centres respectively. More time spent on RBF approximation is to obtain the boundary circle with the modified Graham scan technique, whose complexity is $O(N\log(N))$. The time of other operations, e.g. resampling uniform points and pushing boundary circle outward is very little, thus can be neglected.

If the surface is very smooth, it is acceptable to resample the centres uniformly. When the shape variation of the surface is rather large or the data distribution is very irregular, the optimal centre densities will be related with the distribution of data points and the shape of the surface. Therefore, a manipulation of centre selection is required.

Example 2 Centre Selection of RBF

The MATLAB built-in function *peaks* is applied very extensively in numerical computations. Its representation is,

$$f(x,y)=3(1-x)^2\exp[-x^2-(y+1)^2]-10(\frac{x}{5}-x^3-y^5)\exp(-x^2-y^2)-\frac{1}{3}\exp[-(x+1)^2-y^2] \qquad (4.33)$$

We adopt the following function as a model surface,

$$z = f(x, y) = peaks(\frac{x}{8}, \frac{y}{8}) \qquad (4.34)$$

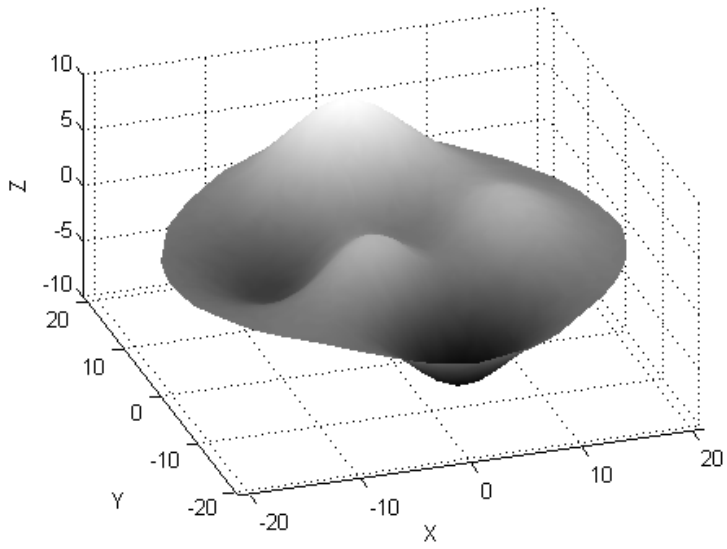


Figure 4.17 Reconstruction surface

1623 points are randomly selected within the domain $x^2 + y^2 \le 400$ as data points. See Figure 4.18.

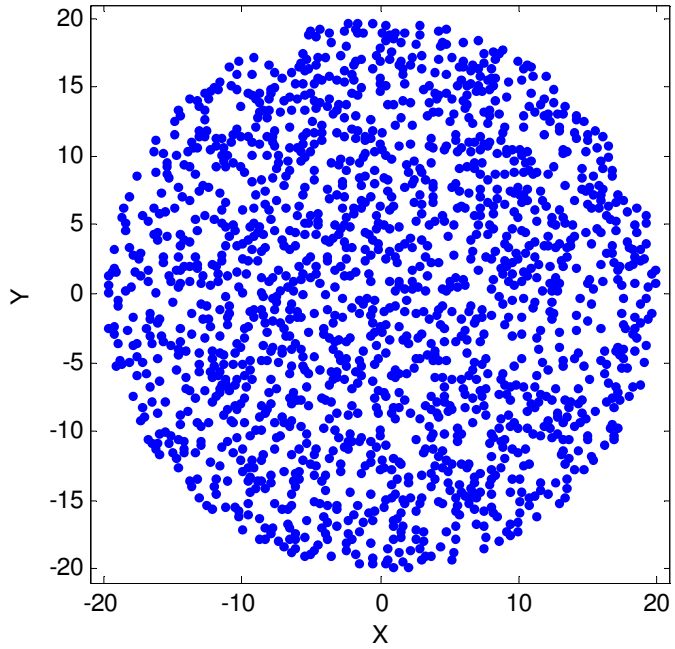
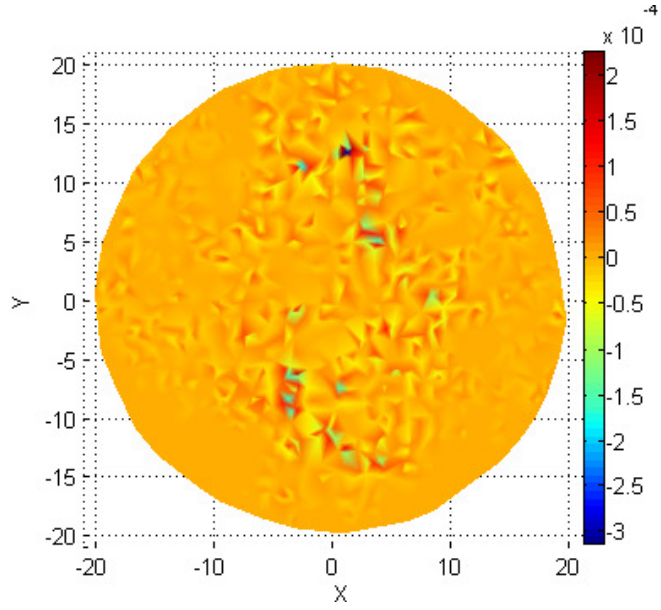


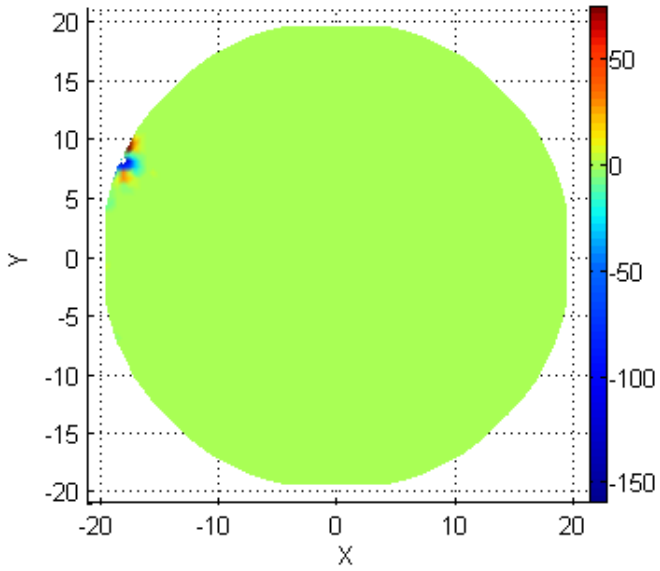
Figure 4.18 Randomly sampled data

Firstly the centres are uniformly sampled in the area $x^2 + y^2 \le 441$ with spacing $h = 1.0$, yielding 1373 centres. The calculation of weighting parameters costs 6.277 seconds and the fitting errors at the data locations are illustrated in Figure 4.19 (a). In

order to check the reconstruction quality more completely, some new locations are uniformly sampled with spacing $h = 0.3$ and the corresponding deviations with respect to the ideal surface are given in Figure 4.19 (b).



(a) Fitting error



(b) Evaluation error

Figure 4.19 Reconstruction errors of uniform centres

These centres fit the given data points very well, and the resultant error is at the order of 10^{-4}. But due to the high irregularity of the data points, the evaluation error is rather large. That is to say, this RBF system is not stable and the resultant over-fitting phenomenon is unacceptably serious.

In order to make the centres distributed better and sparser, the orthogonal least squares basis hunting (OLS-BH) technique is adopted to select centres recursively. 417 centres are finally obtained in the programme, as depicted in Figure 4.20. It is interesting that the centre distribution is in high accordance with the surface shape. The centres are denser at plateau and valley regions (curved areas) and sparser at transitional regions (smoother areas). There are also some centres outside the domain of data, which are involved to overcome the boundary effect. From another respect it also justifies the necessity and validity of our proposed adding-one-circle method to improve the boundary behaviour.

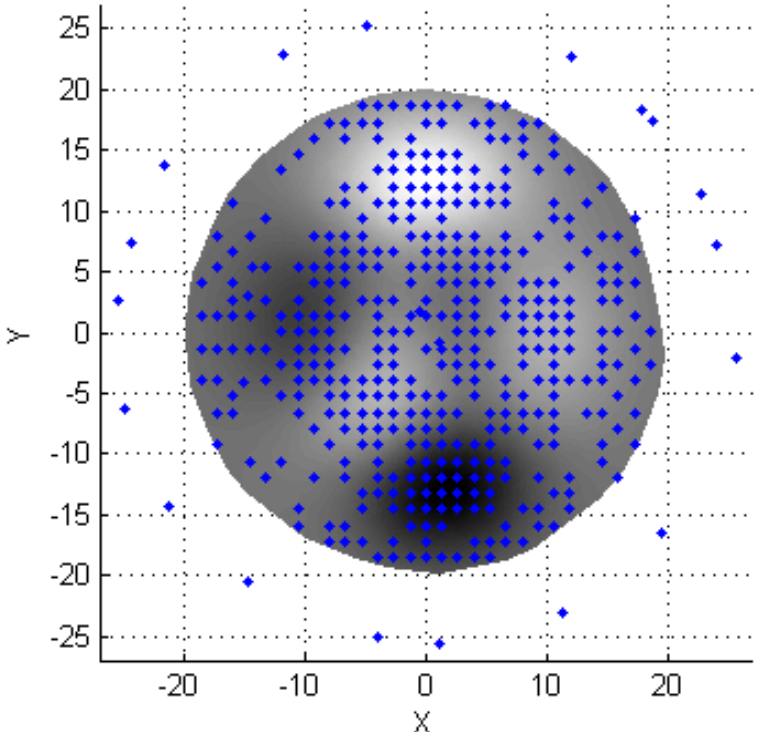
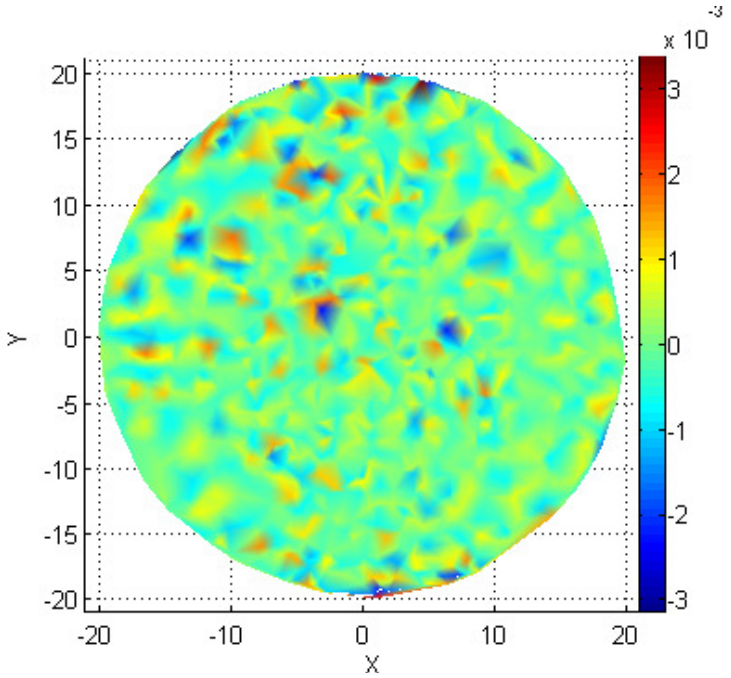
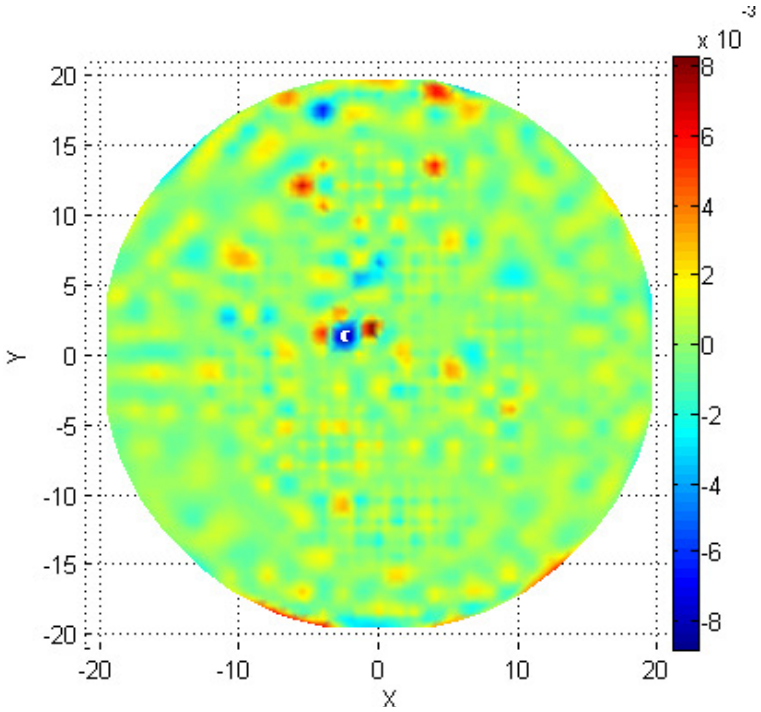


Figure 4.20 Selected centres using the OLS-BH method

To compare its reconstruction quality with the case of uniform centres, the fit errors and evaluation errors are also calculated, as plotted in Figure 4.21. The quantitative comparison by their error parameters is presented in Table 4.4. It can be seen that the interpolation errors in the two cases are in the same order, whereas the evaluation accuracy by the selected centres is four orders higher than uniform centres. That is to say, the reconstructed surface of the selected centres is much smoother and more faithful to the original model surface.



(a) Fitting error



(b) Evaluation error

Figure 4.21 Reconstruction errors of the OLS-BH method

Cases		Uniform centres	Selected centres
Fitting errors	S_a	1.510e-4	4.376e-4
	S_q	2.833e-5	5.861e-4
	S_z	3.501e-4	6.490e-3
Evaluation errors	S_a	0.2014	6.442e-4
	S_q	3.640	9.864e-4
	S_z	233.6	1.708e-2

Table 4.4 Comparison of reconstruction errors

Every time this algorithm selects only one new centre and the randomly generated candidate centres have to be checked successively, so that the program runs very slowly. In this example, the running time of point selection is as long as 173.186 seconds. The yielded centre set is very sparse, thus the numerical stability can be significantly improved. When sampling points from the RBF surface, the programme will run much faster than exact interpolation because of its much smaller centre number.

4.5 Summary

RBF has no specific requirements on the distribution of data points and thus is a very useful tool of surface reconstruction for scattered data.

The arrangement of centres is the key factor influencing the reconstruction quality. Exact interpolation, i.e. adopting all the given data as centres usually causes an unstable system or prones to over-fit the template surface. Hence it is necessary to resample the centres. When the surface is very smooth and the data points are distributed very uniformly, the centres can be uniformly sampled on the domain of interest; otherwise a point selection procedure called the orthogonal least squares basis hunting will be performed to recursively sample a new set of optimal centres.

The reconstruction quality near the boundary region is usually very poor compared with the interior domain. It is suggested that adding some auxiliary points outside the domain of data is able to overcome this problem effectively. In addition, RBF behaves very poor at sharp edges and corners. When the boundary of the surface is very irregular, severe errors may also arise at the sharp-turn areas.

In contrast with NURBS, RBF takes all the data points and centres as a whole, thus leading to a very large-sized interpolation matrix. The solution may be very unstable,

computationally complex, and memory consuming. For this reason RBF will only be employed when the data points are no more than several thousand. If the data size is very large, the whole data set can be divided into several regions and processed separately.

Fortunately RBF requires to select centres and to calculate weights only one time. Once the RBF system has been established, an explicit representation is obtained for the surface. When implementing interpolation, it just needs to substitute the abscissa of the evaluated location into the interpolation matrix, thus is very efficient. This is particularly attractive during the iterative fitting algorithm, which interpolates the free-form surface many times. By contrast, although a NURBS system is very efficient and easy to be built up, it is a parametric model. Point inversion is essential for surface interpolation, thereby greatly hindering the efficiency of the program.

4.6 References

Barker, R. M., Cox, M. G., Forbes, A. B. and Harris, P. M. 2004 *Discrete Modelling and Experimental Data Analysis*. Ver 2. NPL Report

Baxter, B. J. C. 1992 *The Interpolation Theory of Radial Basis Functions*. Ph.D Thesis. Cambridge University

Bishop, C. M. 1995 *Neural Networks for Pattern Recognition*. Oxford University Press

Björkström, A. 2007 *Regression Methods in Multidimensional Prediction and Estimation*. Ph.D Thesis. Stockholm University

Broomhead, D. S. and Lowe, D. 1988 Multivariate functional interpolation and adaptive networks. *Complex Systems*. 2(3): 321-355

Chen, S., Hong, X., Harris, C. J. and Sharkey, P. M. 2004 Sparse modelling using orthogonal forward regression with PRESS statistic and regularization. *IEEE Trans on Sys, Man and Cybernetics B: Cybernetics*. 34(2): 898-911

Chen, S., Wang, X. W. and Harris, C. J. 2008 NARX-based nonlinear system identification using orthogonal least squares basis hunting. *IEEE Trans on Cont Sys Technol*. 16(1): 78-84

Crampton, A. 2002 *Radial Basis and Support Vector Machine Algorithms for Approximating Discrete Data*. Ph.D Thesis. University of Huddersfield, UK

de Castro, L. N. and Von Zuben, F. J. 2001 An immunological approach to initialize centres of radial basis function neural networks. *Proc of V Brazilian Conf on Neural Networks*. 79-84

Duchon, J. 1977 Splines minimising rotation-invariant seminorms in Sobolev spaces. In Schempp, W. and Zeller, K. Editors. *Lecture Notes in Mathematics*. Springer. 571: 85-100

Fedoseyev, A. I., Friedman, M. J. and Kansa, E. J. 2002 Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary. *Comp and Math with Appl*. 43(3-5): 439-455

Fornberg, B., Driscoll, T. A., Wright, G. and Charles, R. 2002 Observations on the behaviour of radial basis function approximations near boundaries. *Comp and Math with Appl*. 43(3-5): 473-490

Franke, R. 1982 Scattered data interpolation: tests of some methods. *Math of Comp*. 38(157): 181-200

Graham, R. L. 1972 An efficient algorithm for determining the convex hull of a finite planar set. *Info Proc Lett*. 1(4): 132-133

Hangelbroek, T. 2007 Error estimates for thin plate spline approximation in the disc. ftp://ftp.cs.wisc.edu/Approx/TPSerror.pdf

Hardy, R. L. 1971 Multiquadric equation of topography and other irregular surfaces. *J of Geophysical Research*. 76(8): 1905-1915

Lee, S., Wolberg, G. and Shin, S. Y. 1997 Scattered data interpolation with multilevel B-splines. *IEEE Trans Visual and Comp Graphics*. 3(3): 1-17

Light, W. 2001 Computing with radial basic functions the Beatson-Light way! In Levesley, J., Anderson, I. J. and Mason, J. C. Editors. *Algor for Approx IV*. Huddersfield, UK, 220-235

Micchelli, C. A. 1986 Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*. 2(1): 11-22

Morandi, R. and Sestini, A. 2001 Geometric knot selection for radial scattered data approximation. In Levesley, J., Mason, J. C. and Anderson, I. J. Editors. *Algor for Approx IV*. Huddersfield, UK, 244-251

Orr, M. J. L. 1996 *Introduction to Radial Basis Function Networks*. Technical Report. Centre for Cognitive Science, University of Edinburgh

Samozino, M., Alexa, M., Alliez, P. and Yvinec, M. 2006 Reconstruction with Voronoi centred radial basis functions. *Eurographics Symposium on Geometry Processing* 51-60

Schaback, R. 1995 Error estimates and condition numbers for radial basis function interpolation. *Adv in Comput Math*. 3(3): 251-264

Valdés, J. L., Biscay, R. and Jimenez, J. C. 1999 Geometric selection of centres for radial basis function approximations involved in intensive computer simulations. *Mathematics and Computers in Simulation*. 48(3): 295-306

Wendland, H. 1995 Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. *Adv in Comput Math*. 4(1): 389-396

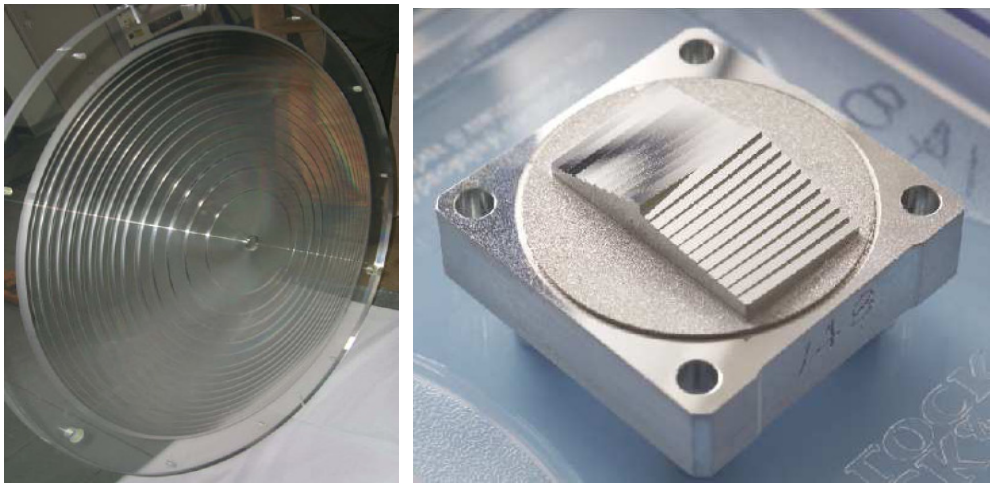
CHAPTER 5 INITIAL MATCHING OF FREE-FORM SURFACES

Given measurement data and the corresponding nominal template, a correct matching should be established so that the relative deviation between them can be obtained. The purpose of initial matching is to supply a reliable rough guess for the relative position between the two surfaces.

Due to the complexity of free-form surfaces, it is not appropriate to use one single method to deal with all kinds of surfaces. Here free-form surfaces are classified into three categories: structured surfaces, smooth surfaces and non-smooth surfaces. Structured surfaces are composed of simple surface patches and can be segmented into regions. Each region can be fitted individually into a quadric and its form error is assessed thereafter by the corresponding shape parameters and residuals. As regards smooth and non-smooth surfaces, a generalized feature called Structured Region Signature is proposed to find a correct matching position between the measurement surface and the design template.

5.1 Segmentation Method

Most working surfaces of engineering products are smooth ones. But there exist some structured surfaces which are constituted of a group of simple surfaces, e.g. Fresnel lens developed for lighthouse, and the image slicer used in the James Webb Space Telescope [Shore 2006], as shown in Figure 5.1. Therefore the entire surface cannot be represented simultaneously using one single function. If measuring the surface as a whole, the data can be divided into smooth surface patches and then processed separately. *Segmentation* is a manipulation which partitions a scale limited surface into distinct regions [ISO/DIS 25178-2: 2007]. Here we introduce a surface segmentation algorithm based on discrete curvatures.



(a) Fresnel lens (b) Image Slicer

Figure 5.1 Examples of structured surfaces

5.1.1 Definition of Discrete Curvatures

In differential geometry, the coefficients of the first and second fundamental forms of the surface at a point **x** on a continuous surface $\mathbf{S}(u,v)$ are defined as [Struik 1950],

$$\begin{cases} E = \mathbf{S}_u \cdot \mathbf{S}_u \\ F = \mathbf{S}_u \cdot \mathbf{S}_v \\ G = \mathbf{S}_v \cdot \mathbf{S}_v \end{cases} \text{ and } \begin{cases} L = \mathbf{n} \cdot \mathbf{S}_{uu} \\ M = \mathbf{n} \cdot \mathbf{S}_{uv} \\ N = \mathbf{n} \cdot \mathbf{S}_{vv} \end{cases} \tag{5.1}$$

where $\mathbf{n} = \dfrac{\mathbf{S}_u \times \mathbf{S}_v}{\|\mathbf{S}_u \times \mathbf{S}_v\|}$ is the normal vector at **x**.

The *Gaussian* and *mean curvatures* are defined as,

$$K = \frac{LN - M^2}{EG - F^2} \text{ and } H = \frac{GL + EN - 2FM}{2(EG - F^2)} \tag{5.2}$$

and the two *principal curvatures* are

$$\kappa_1 = H + \sqrt{H^2 - K} \text{ and } \kappa_2 = H - \sqrt{H^2 - K} \tag{5.3}$$

Here, κ_1 and κ_2 are the two principal curvatures, i.e. the maximum and minimum curvatures at the point **x** along different directions. In fact, the corresponding directions of the two principal curvatures (called *principal directions*) are perpendicular to each other.

In previous equations, the calculation is carried out based on differentiation, which requires a global continuous function of the surface. However, measurement data are generally in discrete forms. Continuous representations are not straightforwardly available for structured surfaces. Therefore, some discrete curvatures are defined. The discrete point set is organized with Delaunay triangulation [Delaunay 1934] and the connecting relationship between data points is thereby established. As presented in Figure 5.2, the neighbour points of an arbitrary vertex $\mathbf{x}$ is supposed to be $N(\mathbf{x}) = \{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_K\}$. The central angle associated with the *j*-th face at $\mathbf{x}$ is θ_j and the two round angles opposite to the edge $\mathbf{x}\mathbf{x}_j$ are α_j and β_j respectively.

According to the Gauss-Bonnet theorem, the integral Gaussian curvature can be obtained by [Desbrun 2000],

$$\iint_{A_M} K dA = 2\pi - \sum_{N(\mathbf{x})} \theta_j \tag{5.4}$$

In this equation, A_M (called *finite volume*) is a family of special regions contained within the 1-ring neighbourhood of $\mathbf{x}$. Normally there are two types of finite volume: *barycentric cell* and *Voronoi cell*. In Figure 5.3 (a), the dot inside each triangle of the 1-ring neighbourhood of $\mathbf{x}$ is the barycentre of the triangle; whereas in Figure 5.3(b), the dot denotes the circumcentre of each triangle. It is proved that the Voronoi cells provide provably tight error bounds under mild assumptions of smoothness. The Voronoi area can be calculated by,

$$A_{Voronoi} = \frac{1}{8} \sum_{\mathbf{x}_j \in N(\mathbf{x})} (\cot \alpha_j + \cot \beta_j) \left\| \mathbf{x}_j - \mathbf{x} \right\|^2 \tag{5.5}$$

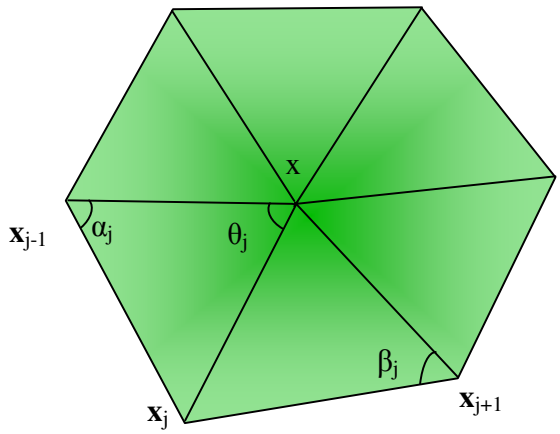
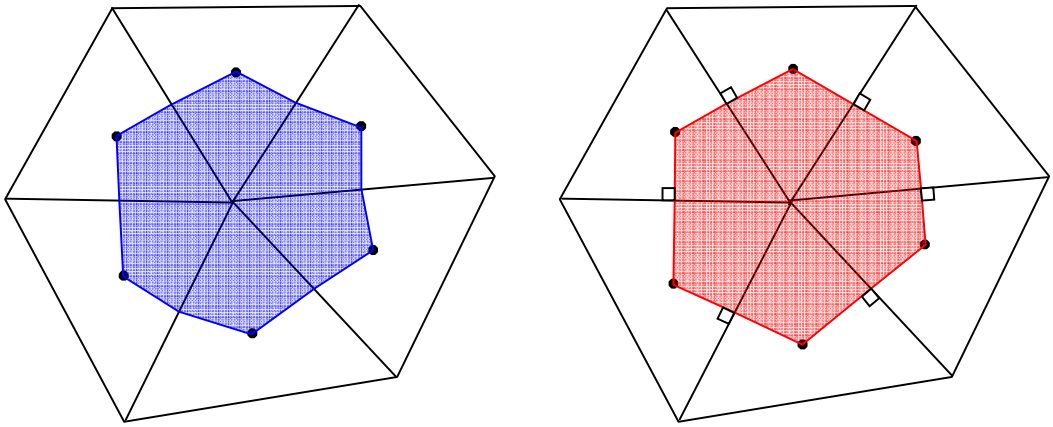


Figure 5.2 Neighbourhood of a vertex $\mathbf{x}$



(a) Barycentric area (b) Voronoi area

Figure 5.3 Two common definitions of finite volume region

It is apparent that the circumcentre of an obtuse triangle locates outside the triangle, and then the corresponding Voronoi area will become meaningless. In order to guarantee a proper accuracy for the discrete curvatures in the presence of obtuse triangles in the 1-ring neighbourhood, a mixed area A_{mixed} is adopted: if a triangle is obtuse, take its barycentric area to define the finite volume region; otherwise, adopt its Voronoi area. As a consequence the discrete Gaussian curvature becomes,

$$K(\mathbf{x}) = \frac{1}{A_{mixed}} \left(2\pi - \sum_{N(\mathbf{x})} \theta_j \right) \tag{5.6}$$

As regards the mean curvature, the Laplace-Beltrami operator is employed,

$$LB(\mathbf{x}) = 2H(\mathbf{x})\mathbf{n}(\mathbf{x}) \tag{5.7}$$

where $H(\mathbf{x})$ is the mean curvature and $\mathbf{n}(\mathbf{x})$ is the normal vector at the vertex $\mathbf{x}$.

It can be worked out as follows,

$$LB(\mathbf{x}) = \frac{1}{2A_{mixed}} \sum_{\mathbf{x}_j \in N(\mathbf{x})} (\cot \alpha_j + \cot \beta_j)(\mathbf{x}_j - \mathbf{x}) \tag{5.8}$$

The normalized vector of the above equation provides a good approximation of the normal vector $\mathbf{n}(\mathbf{x})$. From Equation (5.7) it is known that the mean curvature is half of the magnitude of $LB(\mathbf{x})$,

$$H(\mathbf{x}) = \frac{1}{4A_{mixed}} \left\| \sum_{\mathbf{x}_j \in N(\mathbf{x})} (\cot \alpha_j + \cot \beta_j)(\mathbf{x}_j - \mathbf{x}) \right\| \tag{5.9}$$

5.1.2 Segmentation Procedure

When the discrete curvatures at all the vertices are calculated, this point set is ready to be segmented into patches.

(a) Curvature Classification

For each vertex, the two principal curvatures are calculated with Equation (5.3), and they are regarded as a 2D point. Then these curvature points $\{\mathbf{c}_i \mid i = 1, \cdots, N\}$ are clustered into different groups using the *k-means clustering method* [MacQueen 1967]. This algorithm constructs clusters in an adaptive way:

Step 1: Given a point set $\{\mathbf{c}_i\}$, pre-assign the number of the clusters k and initialize the seed points $\{\mathbf{m}_j \mid j = 1, 2, \cdots, k\}$ for these clusters.
Step 2: For each point $\mathbf{c}_i$, find the corresponding cluster j it belongs to,

$$j = \arg\min \left\| \mathbf{c}_i - \mathbf{m}_j \right\| \tag{5.10}$$

Step 3: Update $\{\mathbf{m}_j \mid j = 1, 2, \cdots, k\}$ by the gravity centre of all the points located in each cluster.
Repeat Steps 2 and 3 until all the seed points converge.

It is demonstrated that for a fixed cloud of points, the clustered result is not affected by the initial selection of the seed points [MacQueen 1967]. Hence in practice the seed points can be randomly sampled from the input points. An example of constructing two clusters from ten 2-D points is depicted in Figure 5.4.

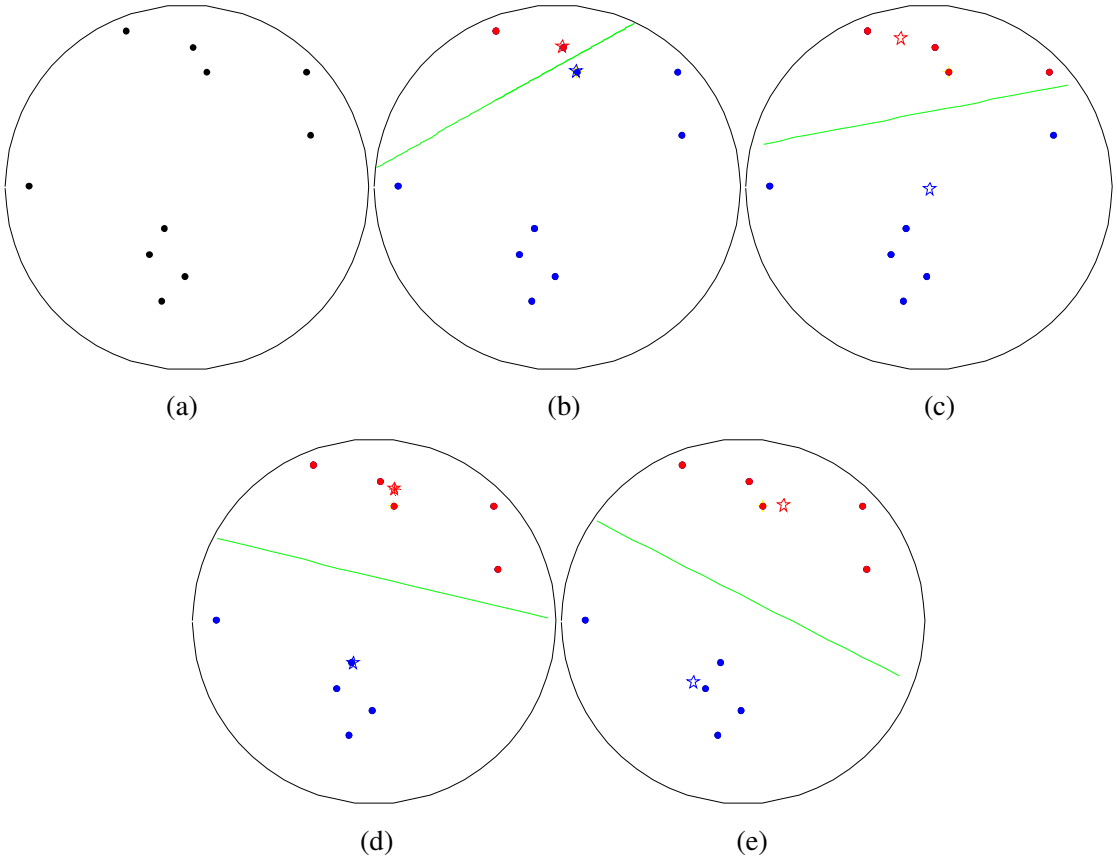


Figure 5.4 Dividing ten points into two clusters

(b) Region Growing

The vertices lying on sharp edges or corners will possess one or two rather large principal curvatures. Such vertices are termed '*sharp*' and the other vertices are termed '*normal*'. Through *k*-clustering, normal and sharp vertices can be classified into different groups. A triangle is regarded as a *seed triangle* if it has three normal vertices in the same curvature cluster or has one normal vertex and two sharp vertices. All seed triangles are then labelled with the curvature cluster label *j* of its normal vertices.

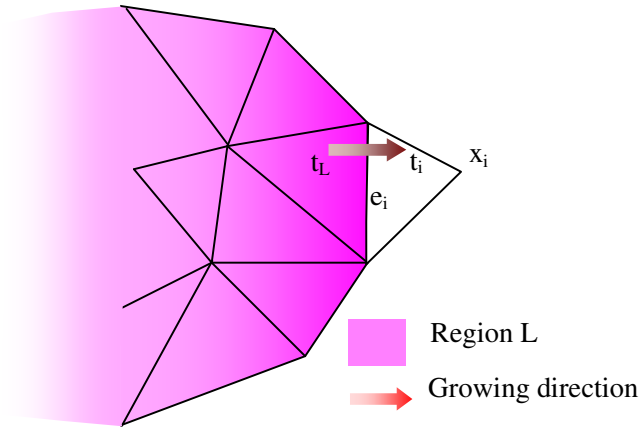


Figure 5.5 Region growing mechanism

When encountering a seed triangle t_L, a new region L is created, as depicted in Figure 5.5. If one of its edge e_i is not a sharp one, check the opposite triangle t_i. The triangle t_i will be integrated into the region L if the vertex $\mathbf{x}_i$ is sharp or also located within the same curvature cluster with region L. This process is repeated for every seed triangle until all the regions cannot grow further. At the end, some triangles may have not been labelled yet. A simple crack filling process will be performed to assign region labels onto these triangles according to their neighbourhoods.

(c) Region Merging

In order to overcome over-segmentation occurred at the region growing step, the regions which are adjacent and have close discrete curvatures with each other can be merged into larger ones.

Firstly a region adjacency graph is established. Two regions sharing one common edge are linked in the graph. A region distance is defined on two adjacent regions to measure the necessity of merging them [Guillaume 2004],

$$D_{rs} = DC_{rs} \cdot N_{rs} \cdot S_{rs} \tag{5.11}$$

In the equation, DC_{rs} measures the difference between their curvatures,

$$DC_{rs} = \left\| \mathbf{c}_r - \mathbf{c}_{rs} \right\| + \left\| \mathbf{c}_s - \mathbf{c}_{rs} \right\| \tag{5.12}$$

with $\mathbf{c}_r$, $\mathbf{c}_s$, and $\mathbf{c}_{rs}$ indicating the average curvatures of regions r, s and their boundary respectively.

N_{rs} measures the nesting between these two regions,

$$N_{rs} = \frac{P_r P_s}{P_{rs}} \tag{5.13}$$

with P_r, P_s, and P_{rs} indicating the perimeters of the regions *r*, *s* and the size of their common boundary respectively.

S_{rs} is used to accelerate fusing the smallest regions,

$$S_{rs} = \begin{cases} \varepsilon & \min(A_r, A_s) < A_{\min} \\ 1 & \text{otherwise} \end{cases} \tag{5.14}$$

Here A_r and A_s are the areas of the two regions, $A_{\min}$ is a user-set minimum area and ε is a very small positive value.

Two adjacent regions with very small distance D_{rs} can be merged into one larger region and their region curvatures are then averaged, so that the whole surface is divided into large surface patches. These regions are separated by sharp edges or have different curvatures. In fact, the final segmentation result is mainly determined by the merging criteria instead of the initial number of clusters *k* [Guillaume 2004]. Therefore, this segmentation procedure is very stable and insensitive to local point distribution.

5.1.3 Fitting of Quadric Surface Patches

After surface patches are extracted with the segmentation algorithm, the form quality of each patch and the relative positional error between them will be evaluated respectively. The shape of the patch is generally a simple geometry, e.g. planes, cones, spheres, cylinders etc. To recognize the shapes of the patches, each segment is firstly fitted with a general quadric function.

Recall the quadric surface fitting introduced in Subsection 2.3.1, the general function of a quadric surface is,

$$Q(\mathbf{x}) = Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + J = 0 \tag{5.15}$$

It can be rewritten into,

$$Q(\mathbf{x}) = \mathbf{x}^T \begin{bmatrix} A & D/2 & E/2 \\ D/2 & B & F/2 \\ E/2 & F/2 & C \end{bmatrix} \mathbf{x} + \begin{bmatrix} G & H & I \end{bmatrix} \mathbf{x} + J = 0 \tag{5.16}$$

with $\mathbf{x} = [x \quad y \quad z]^T$ and a normalization constraint $A^2 + B^2 + C^2 + D^2 + E^2 + F^2 + G^2 + H^2 + I^2 + J^2 = 1$. The equation is solved with the generalized eigenvector technique [Taubin 1991].

Now transform Equation (5.16) into a standard form so that the cross terms can be eliminated. According to the *Spectral Theorem*, the eigenvectors of a real symmetric matrix compose an orthogonal space [Halmos 1963]. So that we implement eigen-decomposition onto the quadric form,

$$\begin{bmatrix} A & D/2 & E/2 \\ D/2 & B & F/2 \\ E/2 & F/2 & C \end{bmatrix} = \mathbf{U}\mathbf{S}\mathbf{U}^T \tag{5.17}$$

In the equation, $\mathbf{S}$ is a diagonal matrix $\mathbf{S} = diag\{\sigma_1, \sigma_2, \sigma_3\}$ with its diagonal entries $\sigma_1 \geq \sigma_2 \geq \sigma_3$ being the eigenvalues. Here $\mathbf{U}$ is a 3×3 unitary matrix. We enforce its determinant be positive, so that $\mathbf{U}$ can be regarded as a rotation matrix in the 3-D Euclidean space and the coordinate system will not be reflected from right-handed to left-handed. Assume $\tilde{\mathbf{x}} = \mathbf{U}^T \mathbf{x}$, so that,

$$Q(\mathbf{x}) = \tilde{\mathbf{x}}^T \mathbf{S} \tilde{\mathbf{x}} + \begin{bmatrix} G & H & I \end{bmatrix} \mathbf{U} \tilde{\mathbf{x}} + J = 0 \tag{5.18}$$

It is rewritten as,

$$\begin{aligned} Q(\mathbf{x}) &= \tilde{\mathbf{x}}^T \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} \tilde{\mathbf{x}} + \begin{bmatrix} \tilde{G} & \tilde{H} & \tilde{I} \end{bmatrix} \tilde{\mathbf{x}} + J \\ &= \sigma_1 \tilde{x}^2 + \sigma_2 \tilde{y}^2 + \sigma_3 \tilde{z}^2 + \tilde{G}\tilde{x} + \tilde{H}\tilde{y} + \tilde{I}\tilde{z} + J \\ &= 0 \end{aligned} \tag{5.19}$$

- The following standard form emerges when $\sigma_1 \sigma_2 \sigma_3 \neq 0$,

$$
\begin{aligned}
Q(\mathbf{x}) &= \sigma_1\left(\tilde{x}+\frac{\tilde{G}}{2\sigma_1}\right)^2+\sigma_2\left(\tilde{y}+\frac{\tilde{H}}{2\sigma_2}\right)^2+\sigma_3\left(\tilde{z}+\frac{\tilde{I}}{2\sigma_3}\right)^2+\left(J-\frac{\tilde{G}^2}{4\sigma_1}-\frac{\tilde{H}^2}{4\sigma_2}-\frac{\tilde{I}^2}{4\sigma_3}\right) \\
&= \sigma_1(\tilde{x}-a_1)^2+\sigma_2(\tilde{y}-a_2)^2+\sigma_3(\tilde{z}-a_3)^2+a_4 \\
&= 0
\end{aligned} \tag{5.20}
$$

The parameters σ_1, σ_2, and σ_3 are used to recognize the surface shape. Without losing generality, we assume $\sigma_1\sigma_2\sigma_3 > 0$. If $a_4 \neq 0$, the function is normalized into $|a_4| = 1$. Otherwise the function is normalized with $\sigma_1^2+\sigma_2^2+\sigma_3^2 = 1$, so that for a given quadric function, a unique set of parameters $\{\sigma_1, \sigma_2, \sigma_3\}$ can be obtained.

• If any one of the three shape parameters vanishes, say $\sigma_3 = 0$, Equation (5.20) will be in the form of,

$$
\begin{aligned}
Q(\mathbf{x}) &= \sigma_1(\tilde{x}-a_1)^2+\sigma_2(\tilde{y}-a_2)^2+\tilde{I}\tilde{z}+\left(J-\frac{\tilde{G}^2}{4\sigma_1}-\frac{\tilde{H}^2}{4\sigma_2}\right) \\
&= \sigma_1(\tilde{x}-a_1)^2+\sigma_2(\tilde{y}-a_2)^2+\tilde{I}\tilde{z}+a_4 \\
&= 0
\end{aligned} \tag{5.21}
$$

Equation (5.21) is normalized into $\sigma_1 = 1$. We force the condition $\sigma_1+\sigma_3+\sigma_2 \geq 0$ are always satisfied here, so that $\sigma_1 \neq 0$. If $\sigma_2 = 0$, the function can be processed in the same manner.

• If only one of the three shape parameters is non-zero, it can only be $\sigma_1 \neq 0$ under the condition $\sigma_1+\sigma_3+\sigma_2 \geq 0$

$$
Q(\mathbf{x}) = \sigma_1(\tilde{x}-a_1)^2+\tilde{H}\tilde{y}+\tilde{I}\tilde{z}+a_4 = 0 \tag{5.22}
$$

The data will be rotated further about the $\tilde{x}$ axis with a matrix

$$
\mathbf{V} = \begin{bmatrix}
1 & 0 & 0 \\
0 & \dfrac{\tilde{H}}{\sqrt{\tilde{H}^2+\tilde{I}^2}} & \dfrac{\tilde{I}}{\sqrt{\tilde{H}^2+\tilde{I}^2}} \\
0 & -\dfrac{\tilde{I}}{\sqrt{\tilde{H}^2+\tilde{I}^2}} & \dfrac{\tilde{H}}{\sqrt{\tilde{H}^2+\tilde{I}^2}}
\end{bmatrix} \tag{5.23}
$$

So that the new data is $\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \mathbf{V} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix}$ and the standard function becomes,

$$Q(\mathbf{x}) = \sigma_1 (\hat{x} - a_1)^2 + \sqrt{\tilde{H}^2 + \tilde{I}^2}\, \hat{y} + a_4 = 0 \tag{5.24}$$

The only non-zero shape parameter σ_1 is normalized into +1. The relationships between surface shapes and shape parameters in different cases are summarized in Table 5.1.

σ_1	σ_2		σ_3		Shape
$\sigma_1 > 0$	$\sigma_1 = \sigma_2$		$\sigma_2 = \sigma_3$		sphere
			$\sigma_2 > \sigma_3 > 0$		oblate spheroid
			$\sigma_3 = 0$	$\tilde{I} \neq 0$	circular paraboloid
				$\tilde{I} = 0$	cylinder
	$\sigma_1 > \sigma_2 > 0$		$\sigma_2 = \sigma_3$		prolate spheroid
			$\sigma_2 > \sigma_3 > 0$		ellipsoid
			$\sigma_3 = 0$	$\tilde{I} \neq 0$	elliptic paraboloid
				$\tilde{I} = 0$	elliptic cylinder
	$\sigma_2 = 0$	$\tilde{H} \neq 0$	$\sigma_3 = 0$		parabolic cylinder
			$\sigma_3 < 0$		hyperbolic paraboloid
		$\tilde{H} = 0$	$\sigma_3 = 0$		two parallel planes
			$\sigma_3 < 0$	$a_4 \neq 0$	hyperbolic cylinder
				$a_4 = 0$	two intersecting planes
	$\sigma_2 < 0$		$\sigma_2 = \sigma_3$	$a_4 > 0$	one-sheet hyperboloid of revolution
				$a_4 = 0$	cone
				$a_4 < 0$	two-sheet hyperboloid of revolution
			$\sigma_2 > \sigma_3$	$a_4 > 0$	one-sheet hyperboloid
				$a_4 = 0$	elliptic cone
				$a_4 < 0$	two-sheet hyperboloid
$\sigma_1 = 0$	$\sigma_2 = 0$		$\sigma_3 = 0$		plane

Table 5.1 Determine the shape of quadrics according to the shape parameters

Once all the shape parameters have been obtained, the specific shape of each surface segment can be decided. The correspondence relationship between the segments of the measurement data and patches on the design template is thereby established.

In practice the fitted parameters may not be exactly equal to the theoretical values due to measurement noise and computational errors. Hence a small tolerance ε is set accordingly. An actual shape parameter will be regarded to be zero if it is very small or

thought to be equal to the nominal one if they are very close to each other within a tolerance ε, i.e. $|\sigma_3| < \varepsilon \Rightarrow \sigma_3 = 0$ and $|\sigma_1 - \sigma_1'| < \varepsilon \Rightarrow \sigma_1 = \sigma_1'$.

As shown in Figure 5.1, segments of structured surfaces may be very small and narrow, yielding very large uncertainty and bias in the fitted parameters. Thus each surface patch shall be fitted further with a type specific algorithm to work out their exact positional and form errors. Orthogonal distances can be employed in the error metric and outliers will be handled separately, so that the measurement data can be aligned with the design model according to the positional parameters, e.g. the centre of a sphere or the axis of a cylinder. In this way the shape quality of a structured surface can be evaluated. The orthogonal distance fitting and overcoming outliers are in the scope of final fitting, which will be discussed further in Chapter 6.

5.2 Structured Region Signature Method

If the measurement surface is a general smooth free-form surface, it will have no shape or positional parameters straightforwardly available as quadric surfaces do. Moreover, salient features or reference datums may not exist to be used for aligning the measurement data with the design template. Motivated by the *point signature technique* by Chua and Jarvis [Chua 1997], we propose a generalized feature called *Structured Region Signature* (SRS) for partial matching of smooth free-form surfaces.

It is assumed that the template consists of discrete points and it should be a smooth free-form surface, i.e. its normal vectors are continuous and there are no occlusions.

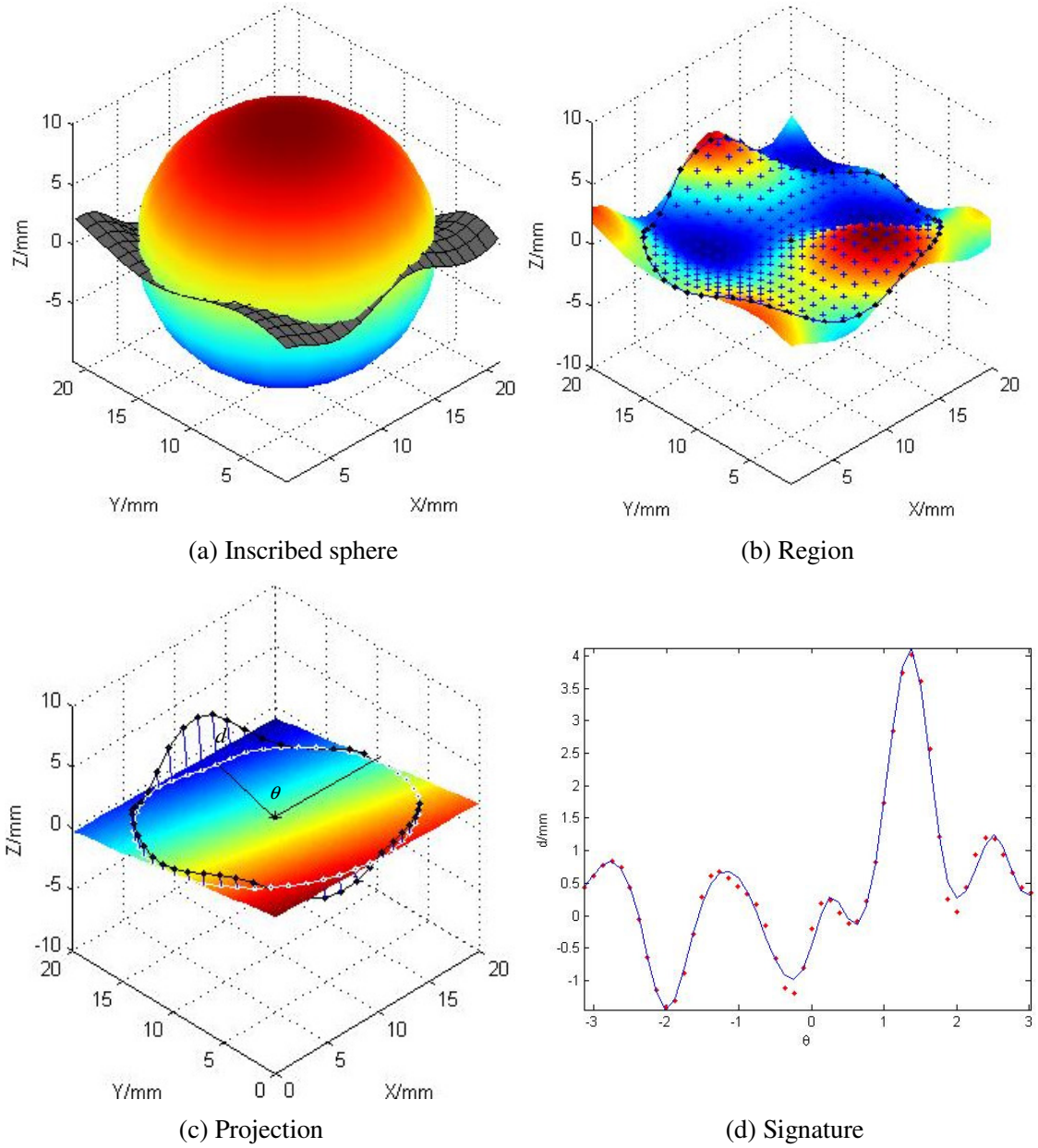
5.2.1 Definition of SRS

Firstly, given measurement data $\mathbf{P} = \{\mathbf{p}_i \mid i=1,2,\ldots, N\}$, a point $\mathbf{c}_m = [x_c, y_c, z_c]^T$ is chosen at the centre of the measurement surface and an inscribed sphere with radius R_m is placed with its centre at $\mathbf{c}_m$. Here R_m should be as large as possible, while the sphere should be always contained within the boundary of the measurement surface. In practice R_m is taken to be the smallest distance from $\mathbf{c}_m$ to the boundary, as illustrated in Figure 5.6 (a).

The measurement points lying within the sphere (termed *region points*, denoted with crosses in Figure 5.6 (b)) constitute a region REG_m. A plane $ax + by + cz + d = 0$ is fitted through the region, such that,

$$(a,b,c,d) = \arg \min \sum_{\mathbf{p}_i \in REG_m} \frac{(ax_i + by_i + cz_i + d)^2}{a^2 + b^2 + c^2} \tag{5.25}$$

where $\mathbf{p}_i = [x_i, y_i, z_i]^T$ is an arbitrary point within the region REG_m.



(a) Inscribed sphere (b) Region

(c) Projection (d) Signature

Figure 5.6 Creating a signature

Practically, this equation is solved with the generalized eigenvector method [Taubin 1991].

The gravity centre of the region is,

$$\mathbf{p}_c = \frac{1}{N_R} \sum_{\mathbf{p}_i \in REG_m} \mathbf{p}_i = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$$

where N_R is the number of the region points.

A 3×3 symmetric covariance matrix is thereby constructed,

$$\mathbf{A} = \sum_{\mathbf{p}_i \in REG_m} (\mathbf{p}_i - \mathbf{p}_c)(\mathbf{p}_i - \mathbf{p}_c)^T \tag{5.26}$$

The normal vector of the plane $\mathbf{n}_m = [a,b,c]^T$ is taken to be the normalized eigenvector associated with the smallest eigenvalue of the matrix $\mathbf{A}$, assuming it is $\mathbf{v}_3$ [Taubin 1991]. If the z component of $\mathbf{v}_3$ is negative, $-\mathbf{v}_3$ will be adopted instead, so that the representation of the fitted plane can always be guaranteed to be unique.

A new plane $\mathbf{P}_m$ is defined by moving the fitted plane to go through the centre point $\mathbf{c}_m$, without changing its orientation. The function of $\mathbf{P}_m$ is,

$$a(x - x_c) + b(y - y_c) + c(z - z_c) = 0$$

or rewritten as

$$ax + by + cz + d' = 0 \tag{5.27}$$

with $d' = -(ax_c + by_c + cz_c)$.

Then an appropriate number N_C of region points lying nearest to the sphere surface will be selected to constitute a circle, see Figure 5.6(b).

These circle points $\{a_j, b_j, c_j\}$, j=1, …, N_c are projected onto the plane $\mathbf{P}_m$ and the signed projection distances are

$$d_j = ax_j + by_j + cz_j + d', \; j = 1,2,\cdots,N_C$$

yielding N_C projection points,

$$\begin{bmatrix} x_j' \\ y_j' \\ z_j' \end{bmatrix} = \begin{bmatrix} x_j - ad_j \\ y_j - bd_j \\ z_j - cd_j \end{bmatrix} \tag{5.28}$$

To make the signature independent of the orientation and position of the surface, a local coordinate system is defined by setting the signature centre $\mathbf{c}_m$ as the origin $[0 \quad 0 \quad 0]^T$ and defining the normal vector of the plane as the positive z axis $[0 \quad 0 \quad 1]^T$, consequently yielding the projection plane $\mathbf{P}_m$ to be the *X*-*Y* plane.

For simplicity, it is implemented in an equivalent way as follows. Firstly the unit radial vectors $\{\bar{\mathbf{r}}_j\}$ from the signature centre $\mathbf{c}_m$ to the projection points are calculated,

$$\bar{\mathbf{r}}_j = \mathbf{r}_j \big/ \left\| \mathbf{r}_j \right\| \text{ with } \mathbf{r}_j = \begin{bmatrix} x_j' - x_c \\ y_j' - y_c \\ z_j' - z_c \end{bmatrix}$$

An auxiliary vector $\mathbf{n}_a$ is defined as the cross product of $\mathbf{n}_m$ and $\mathbf{n}_z = [0 \quad 0 \quad 1]^T$,

$$\mathbf{n}_a = \mathbf{n}_m \times \mathbf{n}_z \big/ \left\| \mathbf{n}_m \times \mathbf{n}_z \right\|$$

Thus two orthonormal frames $[\mathbf{n}_m \quad \mathbf{n}_a \quad \mathbf{n}_1]$ and $[\mathbf{n}_z \quad \mathbf{n}_a \quad \mathbf{n}_2]$ are constructed with $\mathbf{n}_1 = \mathbf{n}_m \times \mathbf{n}_a \big/ \left\| \mathbf{n}_m \times \mathbf{n}_a \right\|$ and $\mathbf{n}_2 = \mathbf{n}_z \times \mathbf{n}_a \big/ \left\| \mathbf{n}_z \times \mathbf{n}_a \right\|$.

Then the pointing vectors $\{\bar{\mathbf{r}}_j\}$ can be rotated onto the *X*-*Y* plane by [Chua 1996],

$$\mathbf{r}_j' = [\mathbf{n}_z \quad \mathbf{n}_a \quad \mathbf{n}_2] \times [\mathbf{n}_m \quad \mathbf{n}_a \quad \mathbf{n}_1]^T \bar{\mathbf{r}}_j \tag{5.29}$$

Set $\mathbf{r}_j' = [x_{rj} \quad y_{rj} \quad 0]^T$, its corresponding polar angle is,

$$\theta_j = \arctan\left(\frac{y_{rj}}{x_{rj}} \right), \; -\pi < \theta_j \le \pi \tag{5.30}$$

Thus the signed projection distances $\{d_j\}$ of all the circle points form a one dimensional function with respect to the polar angles $\{\theta_j\}$, as shown in Figure 5.6(d). This distance profile is called the *structured region signature* S_m of the measurement surface. If this surface is smooth, the theoretical signature curve will be smooth as well. In fact, the selected circle points are not exactly lying on the sphere surface and the resulting signature curve will contain perturbations. In addition the intervals between the adjacent polar angles are not uniform. Therefore a signature curve is modelled from these signature points with smoothing techniques, e.g. least squares cubic splines.

5.2.2 Matching Strategy

The measurement surface is usually only one part of the template, and the best matching position of the measurement data is not supplied. Thus N_s plausible candidate locations can be selected uniformly on the template with appropriate spacing.

Then signatures are similarly generated on the template surface, centred at the sampled plausible locations, employing the same sphere radius R_m as the measurement signature. The template signatures are indicated as $\{S_{Tk} \mid k = 1, 2, \cdots, N_s\}$.

The similarity between a template signature and the measurement signature is evaluated by the structure function,

$$Err_k = \int_{-\pi}^{\pi} [S_m(\theta) - S_{Tk}(\theta)]^2 \, d\theta \tag{5.31}$$

Practically the two coordinate systems of the measurement surface and the template are probably misaligned, hence there may be relative angle-shift between their signature profiles.

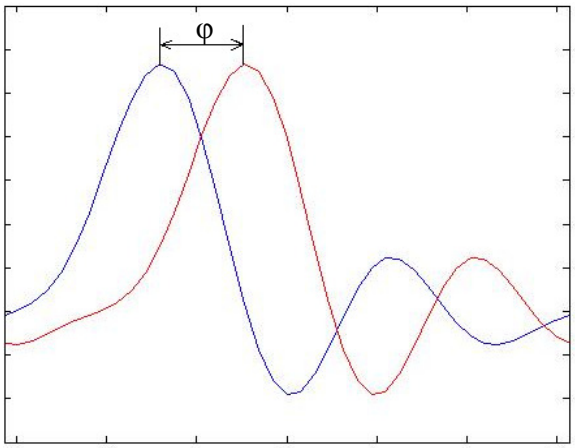


Figure 5.7 Relative shift between two signatures

Then the best-matching problem turns out to be a minimization task,

$$\{S_{Tk}, \varphi\} = \arg\min_{\substack{k=1,\cdots,N_s \\ -\pi<\varphi\le\pi}} = \int_{-\pi}^{\pi} [S_m(\theta+\varphi) - S_{Tk}(\theta)]^2 \, d\theta \tag{5.32}$$

However, this is very burdensome to calculate, so all the signature curves are resampled evenly with an appropriate interval π / n (n is a positive integer). The

signature curve S_{Tk} will become discrete date sets $\{S_{Tk}(\theta_l) \mid l = 1, 2, \cdots, 2n\}$ and the relative shift angles are discretized as well,

$$\varphi_r = \frac{\pi}{n}(r-n),\ r = 1,2,\cdots,2n$$

Therefore the best matching is the one which occupies the smallest dissimilarity,

$$\{S_{Tk}, \varphi_r\} = \underset{\substack{k=1,2,\cdots,N_s \\ r=1,2,\cdots,2n}}{\arg\min} \sum_{l=1}^{2n} \left[S_m(\theta_l + \varphi_r) - S_{Tk}(\theta_l)\right]^2 \tag{5.33}$$

The centre of the best-matching template signature S_{Tk} is denoted with $\mathbf{c}_T$, and the unit normal vector of the corresponding fitted plane is $\mathbf{n}_T$.

Then a rotation is performed on the measurement surface to align its normal vector $\mathbf{n}_m$ with $\mathbf{n}_T$. Three unit vectors are defined subsequently

$$\begin{cases} \mathbf{n}_0 = \mathbf{n}_m \times \mathbf{n}_T / \| \mathbf{n}_m \times \mathbf{n}_T \| \\ \mathbf{n}_1 = \mathbf{n}_m \times \mathbf{n}_0 / \| \mathbf{n}_m \times \mathbf{n}_0 \| \\ \mathbf{n}_2 = \mathbf{n}_T \times \mathbf{n}_0 / \| \mathbf{n}_T \times \mathbf{n}_0 \| \end{cases} \tag{5.34}$$

to construct orthonormal frames for the measurement and template signatures respectively. The rotation matrix to align the two orthonormal frames is,

$$\mathbf{R}_1 = \begin{bmatrix} \mathbf{n}_T & \mathbf{n}_0 & \mathbf{n}_2 \end{bmatrix} \times \begin{bmatrix} \mathbf{n}_m & \mathbf{n}_0 & \mathbf{n}_1 \end{bmatrix}^T \tag{5.35}$$

The measurement surface should be rotated an angle φ_r about its new normal vector $\mathbf{n}_T = [n_x \quad n_y \quad n_z]^T$ to eliminate the relative angle-shift between S_m and S_{Tk}. The corresponding rotation matrix is [Grimson 1984],

$$\mathbf{R}_2 = \begin{bmatrix} c + n_x^2(1-c) & n_z s + n_x n_y (1-c) & -n_y s + n_x n_z (1-c) \\ -n_z s + n_x n_y (1-c) & c + n_y^2 (1-c) & n_x s + n_y n_z (1-c) \\ n_y s + n_x n_z (1-c) & -n_x s + n_y n_z (1-c) & c + n_z^2 (1-c) \end{bmatrix} \tag{5.36}$$

with $s = \sin(\varphi_r)$ and $c = \cos(\varphi_r)$. Finally the measurement surface is translated to overlap the two signature centres, and the new measurement surface is,

$$\mathbf{P}' = \mathbf{R}_2 \mathbf{R}_1 (\mathbf{P} - \mathbf{c}_m) + \mathbf{c}_T \tag{5.37}$$

5.2.3 Further Discussion

(a) Multi-Circle Signature

The above SRS employs only one single circle to describe the shape of a surface. To improve the descriptive capability of the signatures, more shape-information could be involved. Actually several concentric circles can be defined at the same signature centre. The number of circles is determined by the point number within the region as well as the complexity of the surface shape. The ratios between the radii of these circles are set to be,

$$R_1 : R_2 : R_3 : \cdots = 1 : \sqrt{2} : \sqrt{3} : \cdots \tag{5.38}$$

so that the areas between adjacent circles are approximately the same. Figure 5.8 shows a two-circle signature.

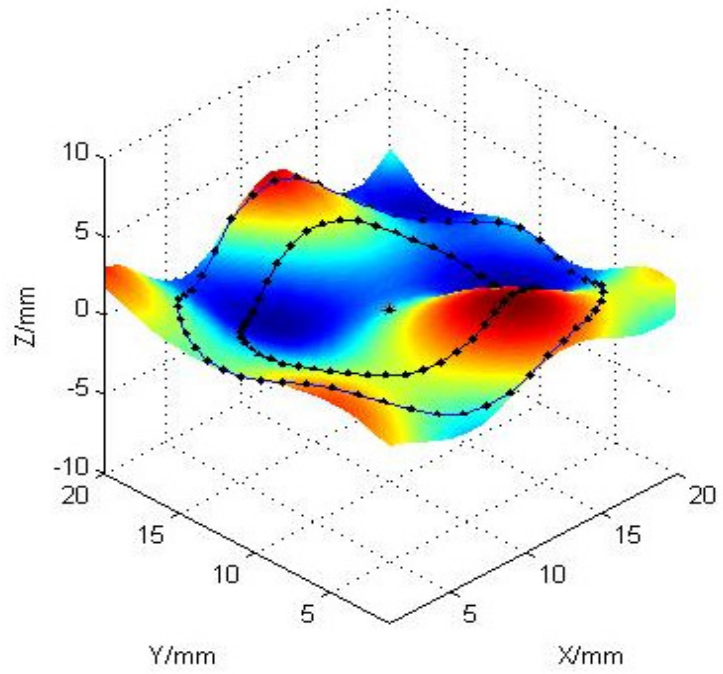


Figure 5.8 A two-circle signature

(b) Sampling Signature Centres on the Template

It has been mentioned in Subsection 5.2.2 that the candidate signature centres are uniformly selected on the template surface. For planar smooth surfaces, it is appropriate to sample centres in this manner. However, at some sharp areas of non-planar surfaces, the SRS may vary greatly even if they are located very near to each other. In this case the density of the centre points is determined by the local density of template points and the

local shape variation on the template surface, in another word, by the curvatures near the signature centres. Furthermore, if the rough position of the best matching is known already, signature centres will be placed with a higher density at this area.

The matching accuracy of translation is determined by the density of the signature centres. Using a higher centre density, the translation error can be restricted within a smaller area, thereby improving the matching accuracy. However, it yields more signatures to be created and compared, consequently reducing the matching efficiency. To overcome this problem, the signature centres can be sampled in a coarse-to-fine approach. Initially, the signature centres are selected using a larger distance (not greater than $R_m/2$). When a rough position $\mathbf{C}_T$ is found, new signature centres are placed around its neighbourhood with a smaller spacing, as shown in Figure 5.9. This is repeated until the spacing is small enough to give a sufficiently good matching result.

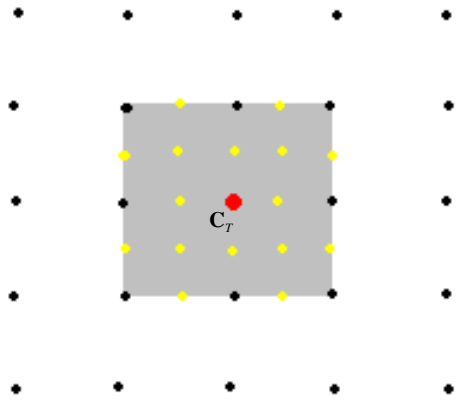


Figure 5.9 Sampling centres in a coarse-to-fine way

(c) Matching-Residual-Checking Strategy

For nearly symmetric surfaces, the SRS matching method fails because the signature's domain of interest is a small part of the measurement surface; there may be many locations occupying nearly the same signatures. As a result of measurement noise and numerical computation errors, the candidate location with the best-matching signature may be an incorrect one. That is to say, the correct matching location usually has a high signature similarity, but the location occupying the highest signature similarity is not necessarily the correct matching.

To overcome this problem, all the plausible locations are sorted in an order such that the locations occupying higher signature similarities are put at the front of the list. Then

the matching residual of each location is checked successively. Once the matching residual satisfies a user-set threshold, the checking process is terminated and a correct matching location is found. Here, the Root Mean Squared Error of the residual is adopted as a metric to assess the goodness of matching.

Figure 5.10 highlights the scheme of the SRS algorithm with residual check.

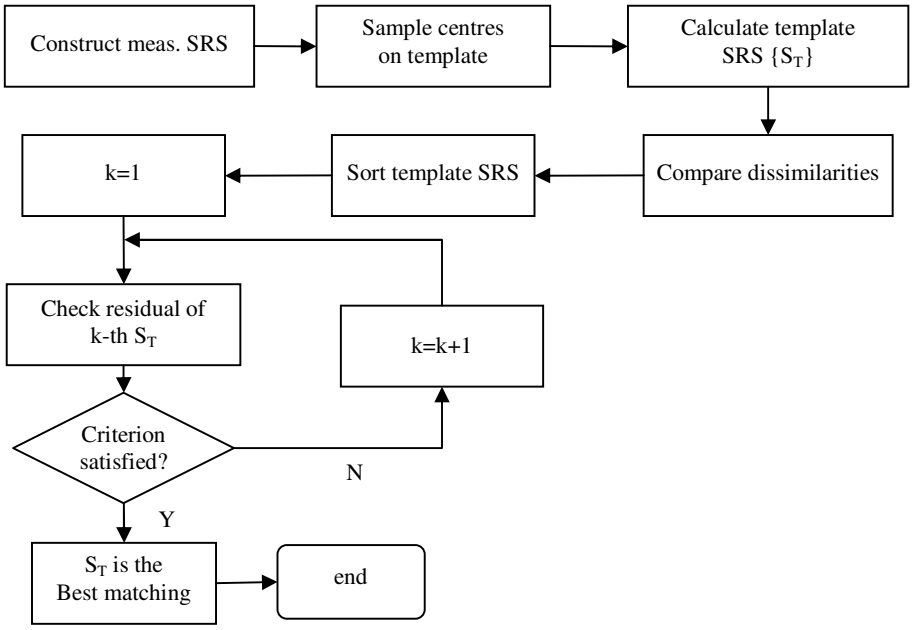


Figure 5.10 Flowchart of the SRS algorithm

5.3 Simulation and Experimental Results

Example 1 Segmentation Method

A micro Fresnel lens is measured with a Wyko NT 2000 Optical Profiler. It consists of three spherical sections.

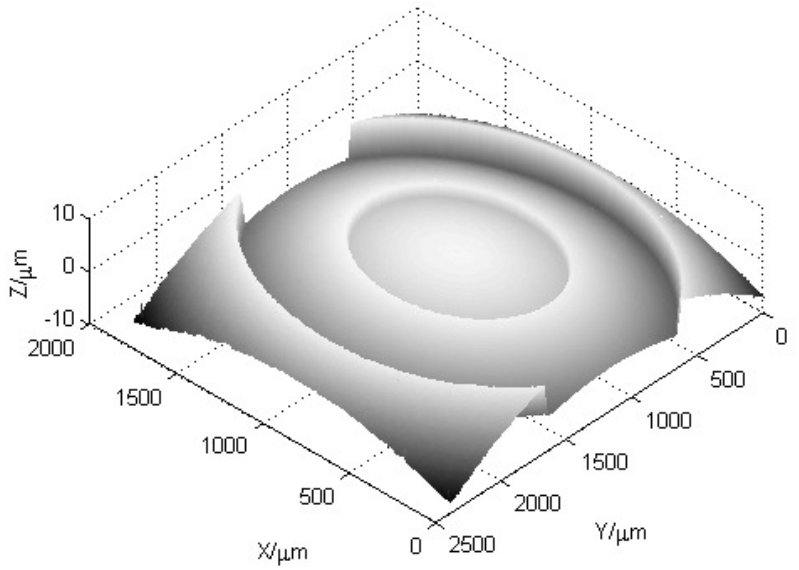
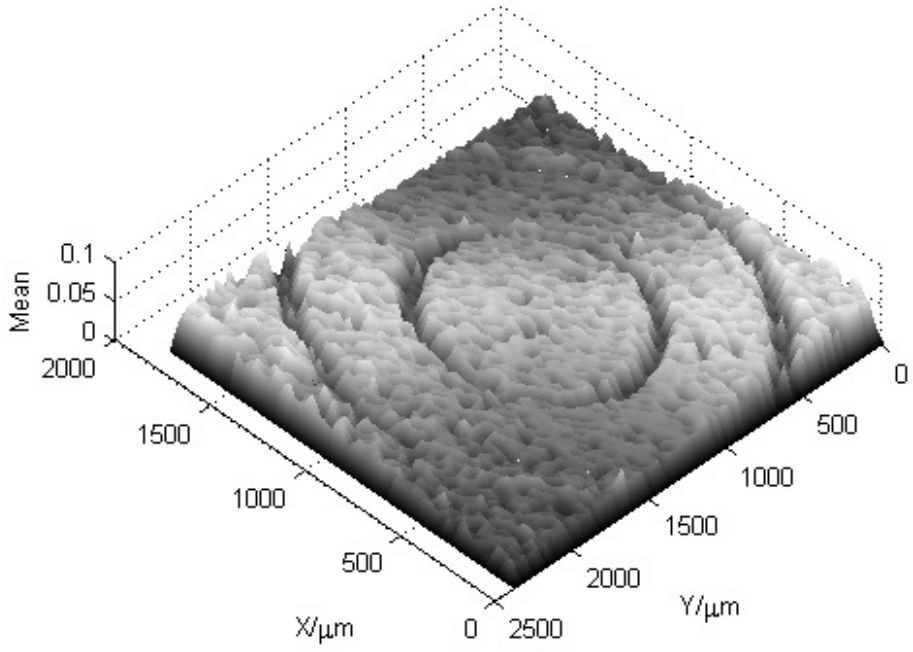
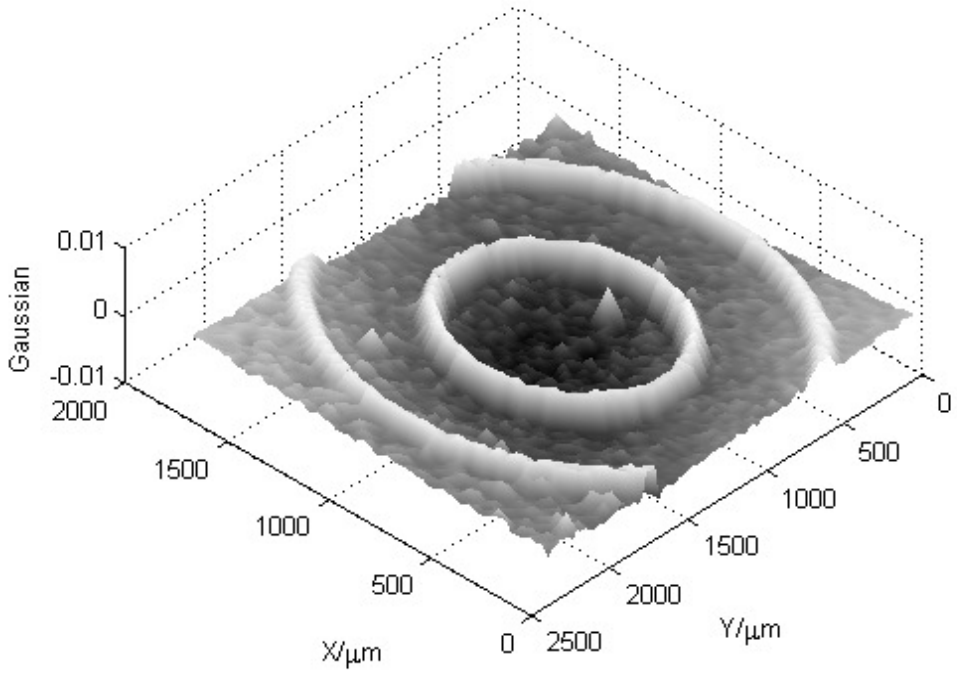


Figure 5.11 Fresnel lens

The measurement data are regularly distributed on a grid and it is straightforward to obtain the connection relationship between the data points. The mean and Gaussian curvatures are calculated for each vertex respectively, as shown in Figure 5.12.



(a) Mean curvatures



(b) Gaussian curvatures

Figure 5.12 Discrete curvatures

The three segments of this Fresnel surface are all spherical, thus all the vertices within each surface section have approximately the same curvature values, except at segment boundaries. In fact, the curvature values are polluted by the measurement error and missing data. When the measurement data are very noisy, local perturbations will dominate and submerge the real information of the surface shape. Therefore, pre-processing is required to deal with the missing data and outliers, and the data will be smoothed if necessary. Another approach is to sample less data points and use a larger spacing, so that the effect of measurement errors can be restrained. Additionally, the discrete curvatures can be post-processed with median filtering.

Then *k*-means clustering is implemented onto the curvatures and all the vertices are grouped into four clusters. It is obvious that the red dots in Figure 5.13 denote segment boundaries. Dispersed red dots at the outer region are caused by missing data and spikes, thus they will be neglected.

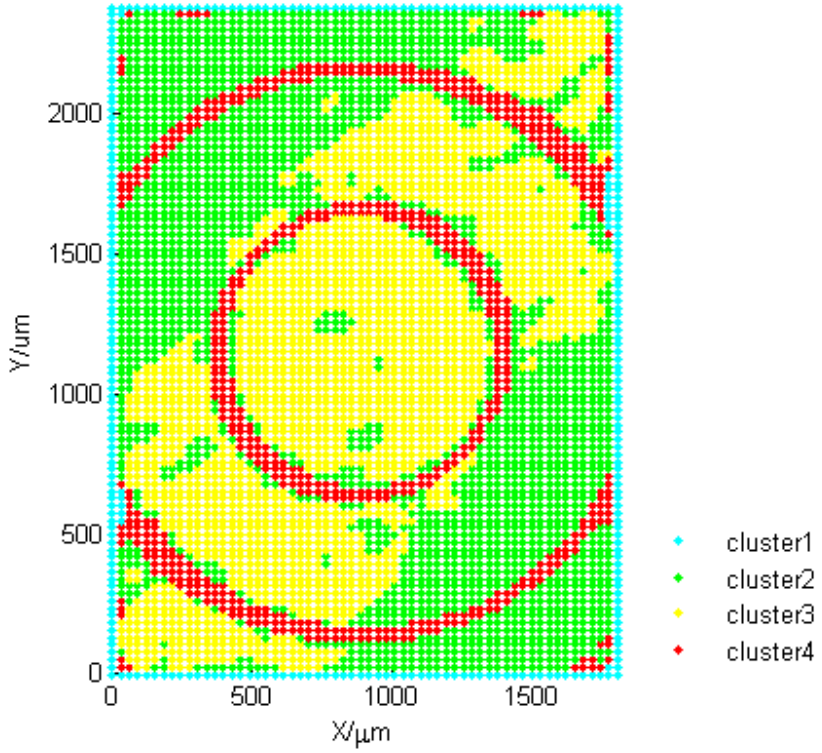


Figure 5.13 Clustering points based on the curvatures

The triangles within the clusters 1, 2, and 3 are regarded as seed triangles. They are organized into four regions using the region growing technique. We have known beforehand that the two outer sections are on the same spherical surface; hence they are combined into one segment, as depicted in Figure 5.14.

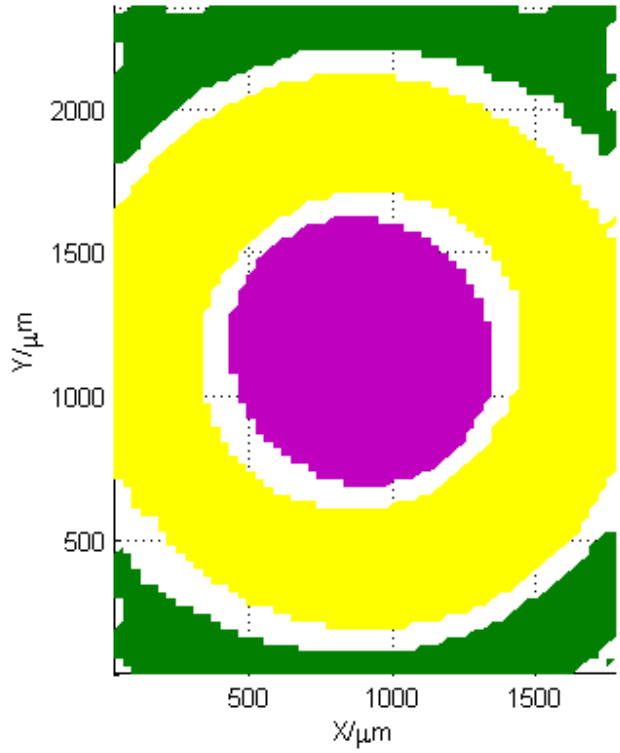


Figure 5.14 Surface segments

These three segments are fitted with functions of spheres respectively and the corresponding parameters are given in Table 5.2. Here regions I, II, and III refer to the central, medium and outer sections respectively. In fact, these parameters are not very accurate because of outliers and missing data. Taking these rough parameters as initial solutions, each section can be fitted further with robust and non-biased techniques. This will be discussed in detail in Chapter 6.

region	Region I	Region II	Region III
point number	901	2485	1162
sphere centre/mm	(0.8996, 1.1683, -54.8061)	(0.9017, 1.1663,-55.4256)	(0.9017, 1.1641,-56.0859)
Sphere radius/mm	54.810	55.431	56.098

Table 5.2 Parameters of the three segments

Figure 5.15 presents the residuals of the measurement data with respect to the fitted sphere surfaces.

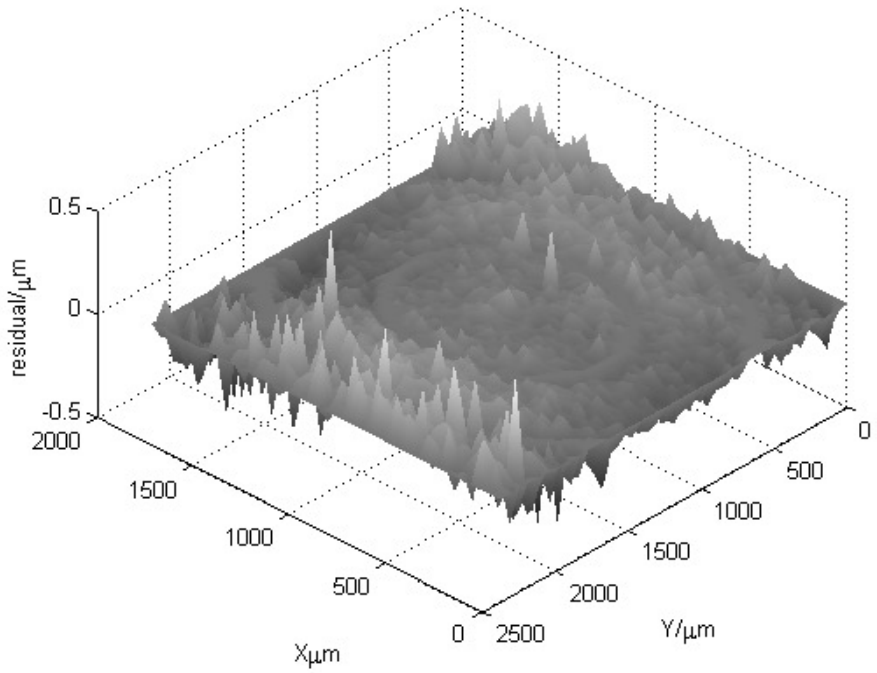


Figure 5.15 Fitting residuals

The program is coded with Matlab 2007a and run on the NEC PC. The running time for calculating curvatures, clustering, region growing, and fitting spheres is 6.903 s, 0.377 s, 0.6763 s, and 0.003 s respectively.

As we know, Fresnel zones are located on a spherical or an aspheric surface, with different offsets. Table 5.2 indicates that the fitted radii and the x and y coordinates of the sphere centres are approximately the same, only their z values are distinctively different.

It can be seen in Figure 5.15 that the fitting residuals are very small and planar, which suggests that the fitted spheres faithfully represent the real shapes of the three spherical sections. From the measurement data we see that the borders between sections are not strictly vertical. Instead, there are data points located on the steep slopes of the interim parts. That is why we can see apparent gaps between sections in Figure 5.14. But in fact they are not so wide. This is caused by the median-filtering of the discrete curvatures. Therefore, post-processing is required to carefully put sharp points into appropriate sections if necessary according to the relative heights between these points and their neighbourhoods.

Another distinct advantage of this curvature-based segmentation algorithm is that more geometric information can be involved. For example, if two adjacent regions with different shapes are tangent to each other, with no obvious height-step between them, in

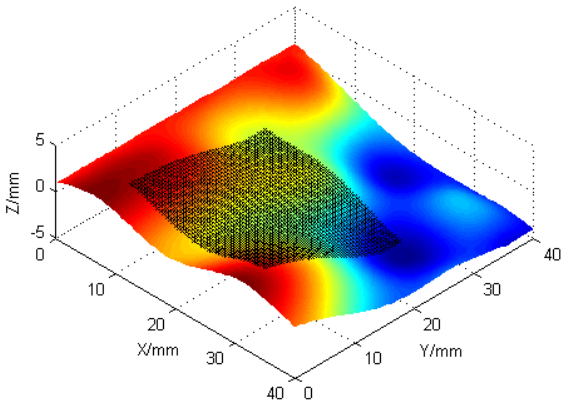
this case it is not sufficient to check only the relative height differences between neighbour points, but this segmentation algorithm can still apply.

Example 2 Simulation of the Structured Region Signature Method

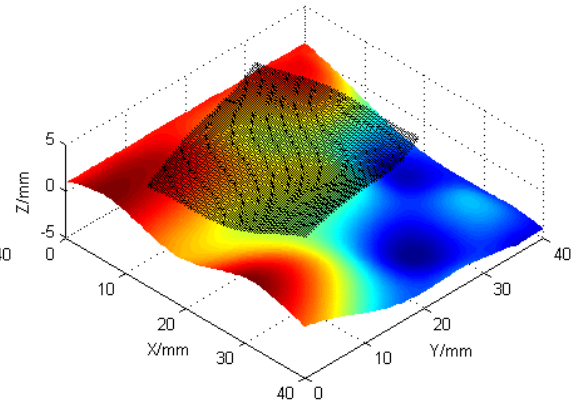
A simulation is given for the Structured Region Signature method. A free-form template surface is simulated with the function,

$$z = \cos(xy/240) + \cos(y^2/200)\sin(x/4) - xy/320 \qquad \begin{matrix} 0 < x \le 40 \\ 0 < y \le 40 \end{matrix} \tag{5.39}$$

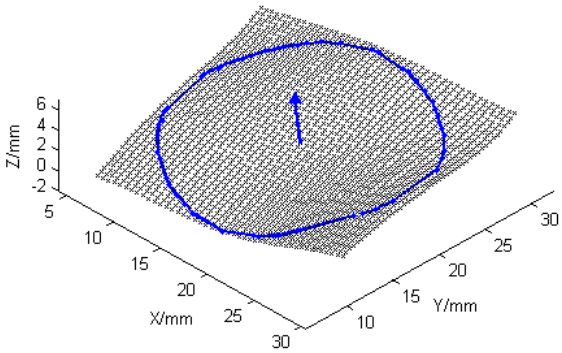
A small part of 22.5 mm×22.5 mm is taken from this template as measurement data (Figure 5.16(a)) and Gaussian noise $N(0,2\mu m)$ is added as measurement error. The measurement surface is moved to an arbitrary location as the initial position before matching (Figure 5.16(b)).



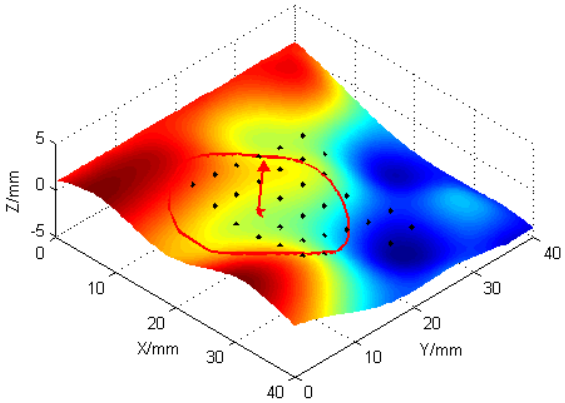
(a) Ideal position



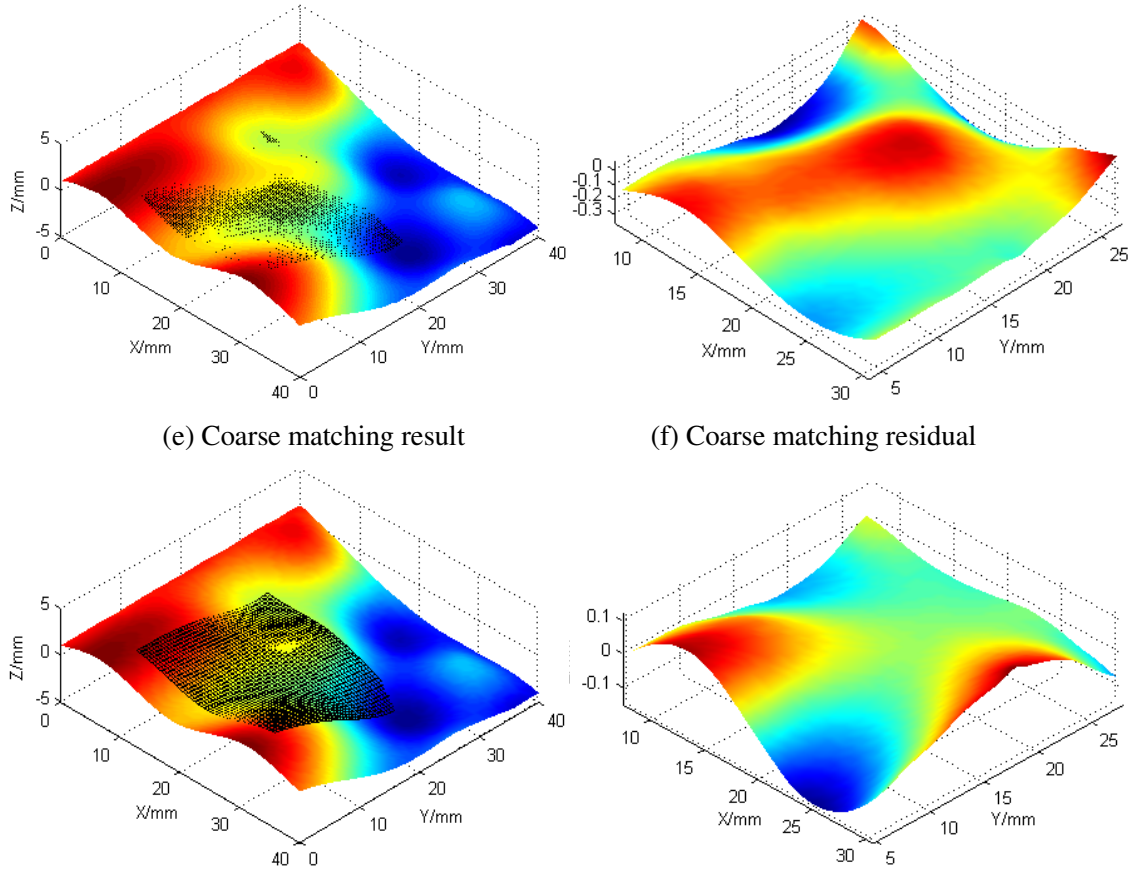
(b) Transformed to a new position



(c) Measurement SRS



(d) Best-matching template SRS



(e) Coarse matching result (f) Coarse matching residual

(g) Fine matching result (h) Fine matching residual

Figure 5.16 Simulation of SRS matching

A SRS is firstly constructed for the measurement surface, and its radius is set to be R_m=11 mm, as shown in Figure 5.16 (c). If candidate signature centres are sampled uniformly with a distance $D=R_m/3$ on the template, the best-matching SRS on the template is given in Figure 5.16 (d). The matching result and residuals are shown in Figure 5.16(e) and (f) respectively. For the matching residual, its Root-Mean-Squared Error S_q is 0.1396 mm and the Max-Min Error S_z is 0.4206 mm. Compared with the ideal position, the translational error is $[0.2550, -0.8939, -0.0423]^T$ mm and the rotational angle error is $[-0.0523, -0.7345, 1.5388]^T$ °. The Matlab program ran 1.703 s to find the best matching position.

If the spacing between the template signature centres is decreased down to $D=R_m/8$, the matching result and residuals are illustrated in Figure 5.16(g) and (h). In this circumstance, S_q is 0.0499 mm and S_z is 0.2761 mm. The errors of translation and rotation are $[0.2562, -0.0240, -0.0344]^T$ mm and $[-0.0216, -0.3991, 1.6537]^{T\circ}$ respectively.

It can be seen that adopting a coarse-to-fine approach, the translation accuracy is greatly improved, but the errors in the rotation angles are still very large. Its reason is apparent: the sampling interval on each signature curve is unchanged. In order to get higher rotation accuracy, with increasing the density of the template signature centres, the sample density on each signature curve should also be increased simultaneously.

This simulation demonstrates the validity of the SRS method. The matching accuracy of translation is restricted by the sampling density of signature centres, whilst the rotation accuracy is controlled by the sampling density of the angles from signature curves.

Example 3 Experimental Result of the SRS Method

Figure 5.17 presents the bearing surface of a total knee joint replacement bearing couple.

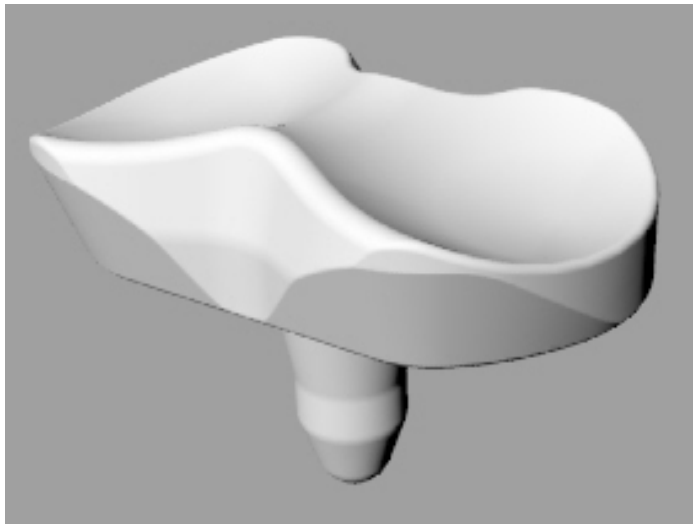
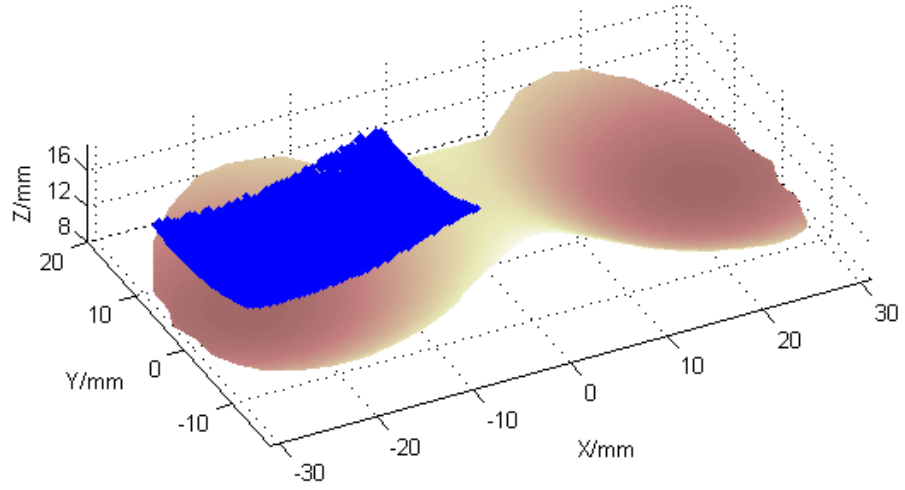


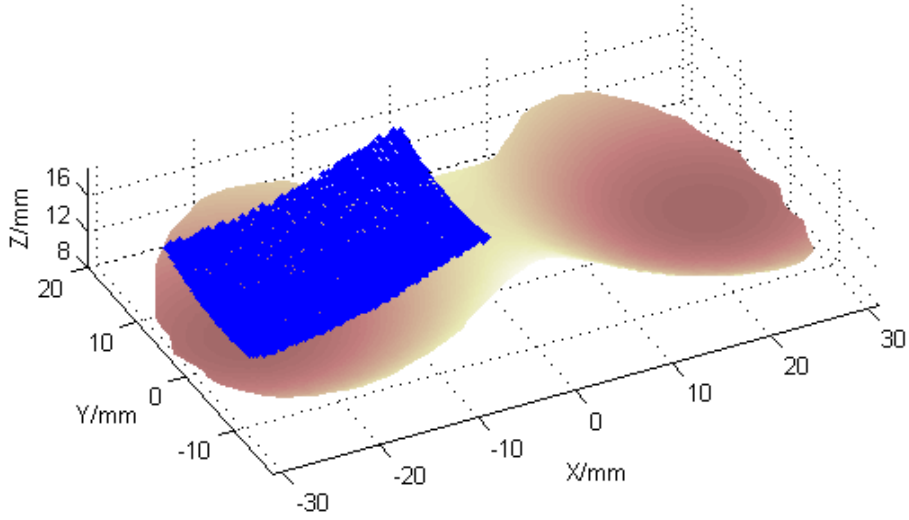
Figure 5.17 Total knee joint replacement model

A nearly spherical part is measured on this joint replacement with spacing d=0.5mm using a Carl Zeiss PRISMO CMM. Radial basis functions are employed to represent the design template and the SRS algorithm is used to match the measurement data with the template. In order to reject false matching caused by the spherical symmetry, the residual checking strategy is applied. The spacing between signature centres on the template is adopted to be $D=R_m/8$. Figure 5.18 (b) plots the situation which has the most similar SRS with the measurement surface. Obviously it is a false matching. In fact the real correspondence position is found to have the eighth best-matching signature. The matching result and residual error are shown in Figure (c) and (d). S_q and S_z parameters

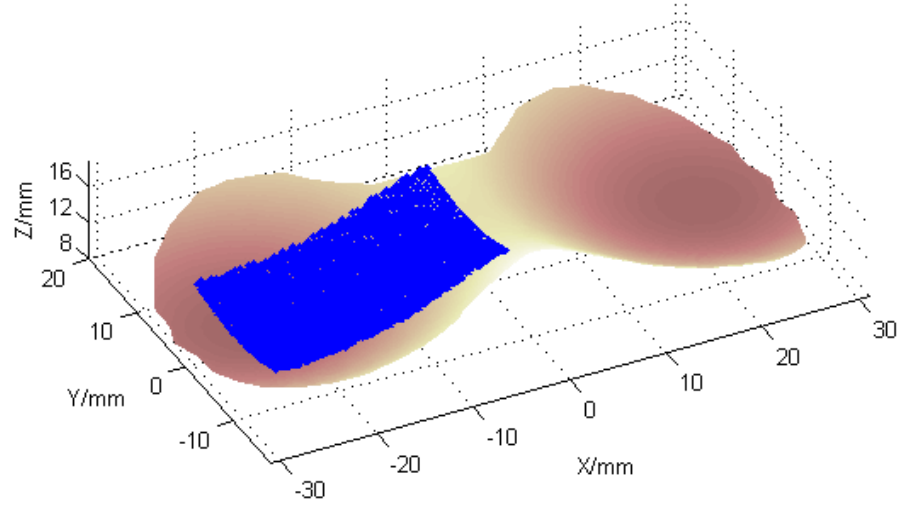
are 36.30 μm and 159.53 μm respectively. The running time of the Matlab program is 2.861 s.



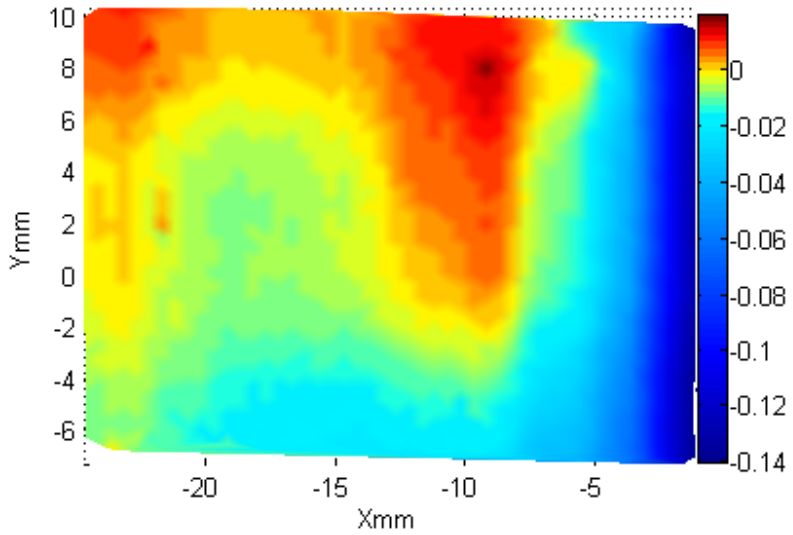
(a) Relative position before matching



(b) False matching result



(c) Correct matching result



(d) Matching residual (mm)

Figure 5.18 Matching a knee joint replacement

This example shows the necessity and validity of the residual checking strategy. For nearly symmetric surfaces, the location with the most similar signature is not necessarily the best matching. Sorting the candidate locations by their signature similarities, the best matching location is usually at the front of the list and not many template signatures need to be checked. Therefore the efficiency of the matching algorithm will not be influenced much, but the reliability of the matching result can be greatly improved.

5.4 Summary

To improve its accuracy and efficiency, the whole fitting procedure is divided into two phases, initial matching and final fitting.

If the measurement surface is structured and composed of simple geometries, the surface qualities of different sections cannot be assessed globally; hence a *segmentation approach* will be applied.

After establishing the connectivity relation between the data points, discrete curvatures can be calculated for each vertex of the data mesh. Then these vertices are grouped into several clusters based on their curvatures and organized into several segments using the region growing method. Then each segment can be fitted with a quadric function and processed individually.

When the measurement data are very dense or noisy, measurement errors will have serious influence on the discrete curvatures. Thus pre-processing is required to eliminate local perturbations. Sparser data points are sometimes preferred to restrain the errors in curvatures caused by the measurement noise.

Global features can be defined to match smooth free-form surfaces. A new initial matching technique called *Structured Region Signature* (SRS) is proposed. Compared with Chua and Jarvis' *Point Signature* [Chua 1997], it does not need to calculate intersection curves between the spheres and surfaces, so that the computational cost is greatly decreased. More importantly, no reference vectors are employed to indicate the zero polar angles, which are prone to false matching. The similarity between two signatures is assessed by successively shifting the polar angles from $-\pi$ to π, so that the relative rotation about the normal vector can be worked out.

Compared with the well known *Spin Image Method* [Johnson 1997], the SRS method does not need to construct lots of spin images, which are very computationally expensive and memory consuming. Additionally, the spin image is a local feature of surfaces. Even for non-symmetric surfaces, it may still lead to false correspondences. If the points are not dense enough or the point density varies greatly on the template, the spin images will not be sufficiently descriptive and may lead to an incorrect matching. On the contrary, SRS is a global feature of the surface and applies an approximate approach to select circle points, even when the region points are not uniformly distributed and the number of the points within a signature sphere is reduced down to 100, this algorithm is still able to find a correct matching location.

To improve the descriptive capability of SRS, several concentric circles can be defined at one signature centre. The translational accuracy is restricted by the sampling spacing of signature centres on the template and the rotational accuracy is determined by the sampling density of the polar angles from signature curves. Candidate locations can be sampled in a coarse-to-fine way on the template surface. To reject false matching, a residual checking approach can be employed. It works well for nearly symmetric surfaces.

If the measurement surface is a long and narrow patch, the radius of the signature will be relatively very small and not much information is involved in the sphere region of the signature, so that SRS cannot represent the surface shape very well. Fortunately, this case rarely occurs in practice.

5.5 References

Chua, C. and Jarvis, R. 1996 3D free form surface registration and object recognition. *Int J of Comput Vis*.17(1):77-99

Chua, C. S. and Jarvis, R. 1997 Point signatures: a new representation for 3D object recognition, *Int J Comput Vis.* 25 (1): 63–85

Delaunay, B. 1934 Sur la sphère vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*. 7(6):793-800

Desbrun, M., Meyer, M., Schroder, P. and Barr, A. 2000 *Discrete Differential-Geometry Operators in nD*. Technical Report. California Institute of Technology

Grimson, W. E. L. and Lozano-Perez, T. 1984 Model-based recognition and localization from sparse range or tactile data. *Int J of Robotics Research*, 3(3):3–35

Guillaume, L., Florent, D. and Atilla, B. 2004 Curvature tensor based triangle mesh segmentation with boundary rectification. *Proc Comp Graphics Int* 10-17

Halmos, P. 1963 What does the spectral theorem say? *American Mathematical Monthly.* 70(3): 241-247

ISO/DIS 25178-2: 2007 *Geometrical Product Specifications-Surface Texture: Areal*-Part 2: Terms, definitions and surface texture parameters

Johnson, A. E. 1997 *Spin-Images: A Representation for 3-D Surface Matching*. Ph.D Thesis. Carnegie Mellon University, USA

MacQueen, J. B. 1967 Some methods for classification and analysis of multivariate observations. *Proc of 5th Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, USA, 281-297

Shore, P., Morantz, P., Lee, D. and McKeown, P. A. 2006 Manufacturing and measurement of the MIRI spectrometer optics for the James Webb Space Telescope. *CIRP Annals-Manuf Technol*. 55(1): 543-546

Struik, D. J. 1950 *Lectures on Classical Differential Geometry*. Addison-Wesley. Cambridge, MA

Taubin, G. 1991 Estimation of planar curves, surfaces and nonplanar spaces curves defined by implicit equation with applications to edge and range image segmentation. *IEEE Trans on Patt Anal and Mach Intell*. 13(11): 1115-1138

CHAPTER 6 FINAL FITTING OF FREE-FORM SURFACES

When a rough matching between the measurement data and template is provided, final fitting follows subsequently to improve the matching accuracy. Two kinds of final fitting methods are explored in this thesis, the *Iterative Closest Point* (ICP) method and *derivative based methods*.

ICP has become the most popular technique for registration. It has no particular requirement on surface shape and works well for various data formats, e.g. continuous functions, discrete point clouds, triangular meshes etc [Jost 2002]. However, it has also some serious drawbacks: local minimum problem and high computational cost.

As a result, the derivative based methods will be adopted when the template is represented with a continuous function or it is easy to be reconstructed. These techniques can efficiently achieve very high fitting accuracy through only several iterations. The reason is evident: more information is incorporated in the templates of continuous formats than discrete ones.

6.1 The Iterative Closest Point Method

We assume that the template $\mathbf{Q} = \{\mathbf{q}_j \mid j = 1,2,\cdots,M\}$ and the measurement data $\mathbf{P} = \{\mathbf{p}_i \mid i = 1,2,\cdots,N\}$ are all constituted of discrete points. The ICP algorithm establishes correspondences between the data and template points, and then gets an optimal transformation to match these point pairs [Besl 1992]. This procedure is repeated until the motion parameters converge, as depicted below.

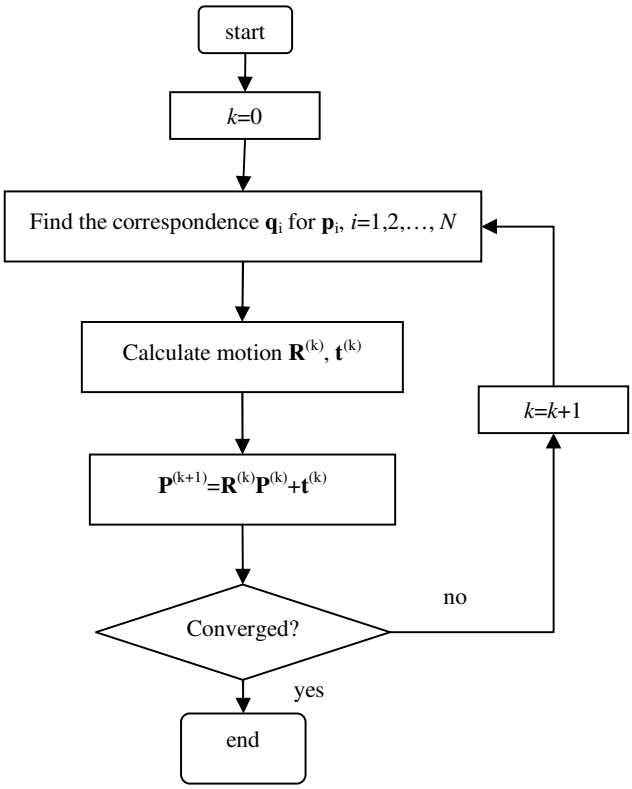


Figure 6.1 Flowchart of ICP

6.1.1 Closest Point Searching with K-D Tree

For each measurement point $\mathbf{p}_i$, the closest template point $\mathbf{q}_i$ is taken as its correspondence point. As mentioned in Subsection 2.3.2, it will be very inefficient if directly searching the closest points all over the whole template, with complexity $O(MN)$, where M and N are the point numbers of the template and measurement data respectively. In order to speed up the closest point searching, the k-D tree technique is adopted [Bentley 1990].

K-D tree is a multidimensional binary search tree constructed by dividing the elements at the median on an axis where the elements have the highest variance. The division of median is repeated until the number of data in each node is less than a given threshold.

Since the measurement data are usually measured in the X-Y plane and the ranges of the x and y coordinates are much greater than that of the z coordinates, a 2-D tree in X-Y plane is sufficient in most occasions. Thus the template points are divided with the medians of the x and y coordinates alternately. The tree nodes are arranged in such an

order that the nodes with smaller x and y coordinates have smaller indices. A 2-D tree example with eight nodes is given in Figure 6.2.

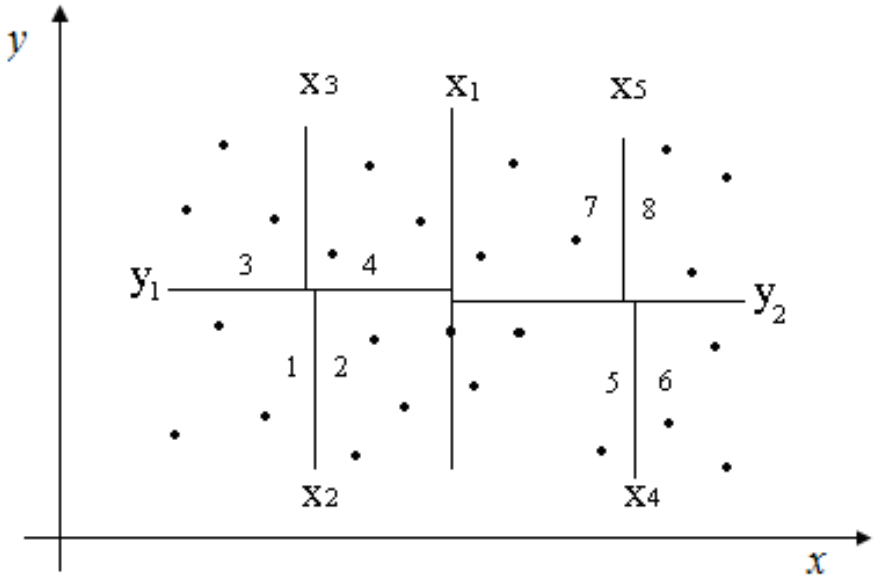


Figure 6.2 Constructing a 2-D tree

If the point number in each node is set no greater than a user-set threshold n, the searching complexity is,

$$O\left(Nn\log\left(\frac{M}{n}\right)\right)=O(N\log M) \tag{6.1}$$

Theoretically, it is fastest to set the node size to be n=1, i.e. each node contains only one point. However, it will make many nodes be dull in practice; as a consequence the back-tracing problem arises. Following the suggestion of Greenspan [Greenspan 2003], the node size is set to be 20.

Once a k-D tree is constructed for the template surface, the corresponding node is sought for each measurement point. For the sake of simplicity, the 2-D tree in Figure 6.2 is taken as an example. The query process to find the corresponding node for an arbitrary measurement point $\mathbf{p}_i(x, y, z)$ is illustrated in Figure 6.3.

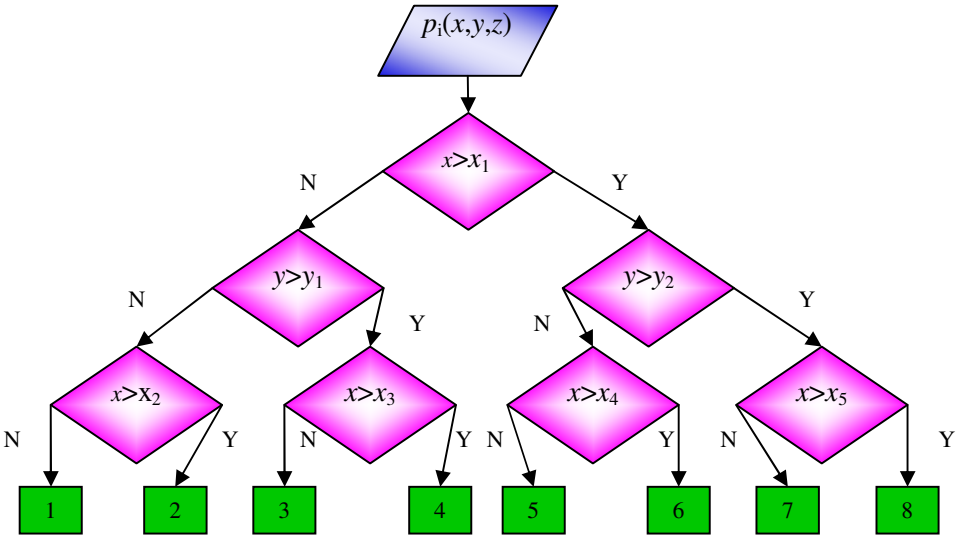


Figure 6.3 2-D tree query process

The point $\mathbf{p}_i$ is assumed to lie in node 2, then all the template points in node 2 are checked and the closest one to $\mathbf{p}_i$ can be normally regarded as its correspondence. But the correspondence $\mathbf{q}_i$ is not always located in the same node with $\mathbf{p}_i$, especially when $\mathbf{p}_i$ is very near to the node boundaries. Therefore, the template points in the neighbour nodes should be checked as well. The searching procedure for the closest point is,

```
Find the closest point p_i in node 2
if |x − x_2| < ||p_i − q_i||
% the distance from p_i to node 1's boundary is nearer than to
q_i, the real correspondence may be in node 1
    find the closest point q_i1 in node 1
    if ||p_i − q_i1|| < ||p_i − q_i||
        q_i ← q_i1
        %q_i1 is the real correspondence point
    end
end
if |x − x_1| < ||p_i − q_i||
% the distance from p_i to node 5's boundary is nearer than to
q_i, the real correspondence may be in node 5
    find the closest point q_i5 in node 5
    if ||p_i − q_i5|| < ||p_i − q_i||
        q_i ← q_i5
        %q_i5 is the real correspondence point
```

```
    end
end
if  |y − y1| < ‖pi − qi‖
% the distance from pi to node 4's boundary is nearer than to
qi, the real correspondence may be in node 4
    find the closest point qi4 in node 4
    if  ‖pi − qi4‖ < ‖pi − qi‖
        qi ← qi4
        %qi4 is the real correspondence point
    end
end
```

It can be seen that node 7 also shares a piece of common boundary with node 2, but the probability that the nearest point lies in node 7 is very low. For simplicity, only the "main" neighbour nodes 1, 4 and 5 are considered.

6.1.2 Calculating Motion Parameters

When the correspondence relationship between the point pairs has been established, optimal motion parameters $\mathbf{R}(\theta_x, \theta_y, \theta_z) \in \Re^{3\times3}$ and $\mathbf{t} = [t_x, t_y, t_z]^T$ are then calculated to minimize an error metric which is used to measure the quality of match. The most widely used error metric is the sum of squared Euclidean distances between correspondence pairs,

$$\min \sum_{i=1}^{N} \left\| \mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i \right\|^2 \tag{6.2}$$

Due to the nonlinearity of this problem, it seems natural to solve the motion parameters using recursive techniques, such as the Newton algorithm, but these methods are somewhat onerous. Some closed-form solution techniques have been developed for this particular purpose. They show superiorities over recursive algorithms in term of efficiency and stability. David W. Eggert et al compared four closed-form solutions and asserted that the *Singular Value Decomposition* (SVD) method achieves the highest matching accuracy [Eggert 1997]. Therefore this technique is adopted here.

To solve the nonlinear problem in Equation (6.2), firstly the centroids of the two point sets are moved to the origin,

$$\begin{cases} \underline{\mathbf{p}}_i = \mathbf{p}_i - \mathbf{p}_c \\ \underline{\mathbf{q}}_i = \mathbf{q}_i - \mathbf{q}_c \end{cases} \tag{6.3}$$

with
$$\begin{cases} \mathbf{p}_c = \dfrac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i \\ \mathbf{q}_c = \dfrac{1}{N}\sum_{i=1}^{N}\mathbf{q}_i \end{cases} \tag{6.4}$$

Hence the translation vector $\mathbf{t}$ can be neglected and Equation (6.2) becomes,

$$\min_{\mathbf{R}} \sum_{i=1}^{N} \left\| \mathbf{R}\underline{\mathbf{p}}_i - \underline{\mathbf{q}}_i \right\|^2$$

$$= \min_{\mathbf{R}} \sum_{i=1}^{N} \left(\underline{\mathbf{p}}_i^T \underline{\mathbf{p}}_i + \underline{\mathbf{q}}_i^T \underline{\mathbf{q}}_i - 2\underline{\mathbf{q}}_i^T \mathbf{R}\underline{\mathbf{p}}_i \right)$$

because the orthogonal matrix $\mathbf{R}$ satisfies $\mathbf{R}^T\mathbf{R}=\mathbf{I}$. This is called the *orthogonal procrustes problem* [Golub 1996]. It is demonstrated to be equivalent to maximizing the trace of $\mathbf{RH}$, where $\mathbf{H} \in \Re^{3\times 3}$ is the correlation matrix,

$$\mathbf{H} = \sum_{i=1}^{N} \underline{\mathbf{p}}_i \, \underline{\mathbf{q}}_i^T \tag{6.5}$$

If the singular value decomposition of $\mathbf{H}$ is $\mathbf{H} = \mathbf{USV}^T$, the optimal rotation matrix will be,

$$\mathbf{R} = \mathbf{VU}^T \tag{6.6}$$

It is evident that the optimal translation vector is,

$$\mathbf{t} = \mathbf{q}_c - \mathbf{R}\mathbf{p}_c \tag{6.7}$$

6.1.3 Convergence Rate of ICP

Suppose the ideal motion parameters are $\mathbf{m}^* = [\theta_x^*, \theta_y^*, \theta_z^*, t_x^*, t_y^*, t_z^*]$ and the solution at the k-th iteration is $\mathbf{m}^{(k)}$. It is demonstrated that ICP exhibits a linear convergence rate [Pottmann 2006],

$$\left\| \mathbf{m}^{(k+1)} - \mathbf{m}^* \right\| \le C \left\| \mathbf{m}^{(k)} - \mathbf{m}^* \right\| \tag{6.8}$$

The positive decay constant is given locally by

$$C = \frac{\cos^2 \phi}{(1 - d\kappa_t^n)(1 - d\kappa_m^n)} \tag{6.9}$$

In the equation, ϕ is the relative angle between two correspondence normal vectors, d is the distance between the two correspondence points, and κ_t^n and κ_m^n are the local normal curvatures at the template and the measurement surface respectively.

When the residual is zero and the minimiser is approached tangentially, we have the worst case C=1. A tangential approach occurs in an exact way only for surfaces which are invariant under a uniform motion. That is to say, the solution will be trapped at a local minimum, and a false matching result will be caused. The false matching shown in Figure 2.9 is for the same reason.

On the other hand, if the two surfaces are planar, which is common for smooth free-form surfaces, the normal curvatures will be relatively small. So that the convergence rate will be very slow if the relative angle ϕ between the two surfaces is small as well. Unfortunately, the Structured Region Signature rough matching will lead to such a situation, i.e. the two surfaces have an apparent relative lateral shift and a small relative angle. Therefore, ICP is not suited for final matching of two planar smooth surfaces which have only relative lateral shift between them. If such a case is encountered, the template surface will be reconstructed into a continuous function and the derivative based algorithms will be adopted.

6.2 Derivative Based Methods

Due to the slow convergence rate and local minimum problem of ICP, the *derivative based algorithms* are instead employed to fit smooth free-form surfaces. Here derivative information is needed for calculating the increment of the solution, therefore a continuous representation should be provided for the nominal template. If the template is in a form of discrete points or a mesh, a reconstruct procedure will be undertaken.

6.2.1 The Levenberg-Marquardt Algorithm

Definition

If the analytical function of a template is

$$z = f(\mathbf{x}), \mathbf{x} = [x, y]^T \tag{6.10}$$

it is intuitive to calculate the optimal motion parameters $\mathbf{m} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z]^T$ by minimising,

$$E = \mathbf{e}^T \mathbf{e} = \sum_{i=1}^{N} [z_i - f(\mathbf{x}_i)]^2 \tag{6.11}$$

where $\mathbf{e} \in \Re^{N\times 1}$ is the residual vector and $\mathbf{x}_i$ is the abscissae of an arbitrary measurement point $\mathbf{p}_i$.

A local minimum can be obtained via $\frac{\partial E}{\partial \mathbf{m}}\Big|_{\mathbf{m}^*} = 2\left(\frac{\partial \mathbf{e}}{\partial \mathbf{m}}\right)^T \mathbf{e}\Big|_{\mathbf{m}^*} = 0$. It is then expanded with the Taylor series,

$$\frac{\partial E}{\partial \mathbf{m}}\Big|_{\mathbf{m}^*} = 2\left(\frac{\partial \mathbf{e}}{\partial \mathbf{m}}\right)^T \mathbf{e} + 2\left[\left(\frac{\partial \mathbf{e}}{\partial \mathbf{m}}\right)^T \frac{\partial \mathbf{e}}{\partial \mathbf{m}} + \mathbf{e}^T \frac{\partial^2 \mathbf{e}}{\partial \mathbf{m}^2}\right](\mathbf{m}^* - \mathbf{m}) + O\left[(\mathbf{m}^* - \mathbf{m})^2\right] = 0 \tag{6.12}$$

Ignoring higher order terms, the *Newton algorithm* (also known as *Newton-Raphson* or *Newton-Fourier* algorithm) iteratively updates the solution by [Fletcher 2000]

$$\delta\mathbf{m} = -\left(\mathbf{J}^T \mathbf{J} + \mathbf{S}\right)^{-1} \mathbf{J}^T \mathbf{e} \tag{6.13}$$

In the equation, $\mathbf{J} = \frac{\partial \mathbf{e}}{\partial \mathbf{m}}$ is the *Jacobian matrix* and $\mathbf{S} \in \Re^{6\times 6}$ with $S_{ij} = \mathbf{e}^T \frac{\partial^2 \mathbf{e}}{\partial m_i \partial m_j}$.

The Newton algorithm exhibits a quadratic convergence rate, which is the fastest among all the iterative algorithms [Fletcher 2000],

$$\left\|\mathbf{m}^{(k+1)} - \mathbf{m}^*\right\| \le C\left\|\mathbf{m}^{(k)} - \mathbf{m}^*\right\|^2 \tag{6.14}$$

In spite of its remarkable convergence rate, the Newton algorithm has also some serious drawbacks. One shortcoming is that the second order derivatives need to be calculated at each iteration, which is very expensive when the function of the surface is rather complicated. As a consequence the term **S** in Equation (6.13) is sometimes ignored, and this leads to the *Gauss-Newton* (G-N) algorithm [Chong 2001],

$$\delta\mathbf{m} = \left(\mathbf{J}^T \mathbf{J}\right)^{-1} \mathbf{J}^T \mathbf{e} \tag{6.15}$$

The validity of the G-N method depends on the accuracy of the second order approximation. Given an initial guess of the variables sufficiently close to the solution,

the G-N method has a super-linear convergence rate. That is to say, the G-N method behaves similarly with the Newton algorithm when **S** is very small. However, a poor starting value may lead to divergence.

Another iterative technique is the *steepest gradient descent method* (SGD) [Chong 2001]. This method regards the objective function *E* as a scalar field in the space of the variables. The solution is incremented recursively along the direction of the negative gradient,

$$\delta\mathbf{m} = -\lambda^{-1}\mathbf{J}^T\mathbf{e} \tag{6.16}$$

The parameter λ controls the step-length at each iteration. This method can guarantee to reduce *E* each time, providing the step-length is sufficiently small, i.e. λ is sufficiently large. However, near the optimum, the convergence rate will become very slow.

Based on a suggestion of Kenneth Levenberg, Donald Marquardt developed a new method, called the *Levenberg-Marquardt* (L-M) algorithm [Marquardt 1963].

This method combines the G-N and SGD methods together, and updates the solution iteratively by

$$\delta\mathbf{m} = -(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{D})^{-1}\mathbf{J}^T\mathbf{e} \tag{6.17}$$

where λ is a damping factor and $\mathbf{D}$ is a diagonal matrix with entries equal to the diagonal elements of $\mathbf{J}^T\mathbf{J}$. In practice, it is feasible to set $\mathbf{D}$ as an identity matrix.

When the damping factor λ changes, this algorithm smoothly switches between the Gauss-Newton and the steepest gradient descent method. A large value of λ corresponds to a small safe gradient descent step, and when $\lambda \to 0$, this algorithm moves towards the Gauss-Newton method and allows faster convergence near the minimum.

A common technique to select λ is as the following hypothesize-and-test paradigm [Press 2002],

(a) Calculate the current fitting error $E(\mathbf{m})$.
(b) Initialize λ, e.g. $\lambda = 0.001$.
(c) Calculate $\delta\mathbf{m}$ using Equation (6.17), and recalculate the error $E(\mathbf{m}+\delta\mathbf{m})$.
(d) If $E(\mathbf{m}+\delta\mathbf{m}) \geq E(\mathbf{m})$, $\lambda \leftarrow \lambda \times k$, reject this update and return to (c).
(e) If $E(\mathbf{m}+\delta\mathbf{m}) < E(\mathbf{m})$, $\lambda \leftarrow \lambda / k$, accept this update and return to (c).

Here k is a user-defined factor, e.g. $k = 5$.

Convergence Region

Now we discuss the convergence domains of the above algorithms.

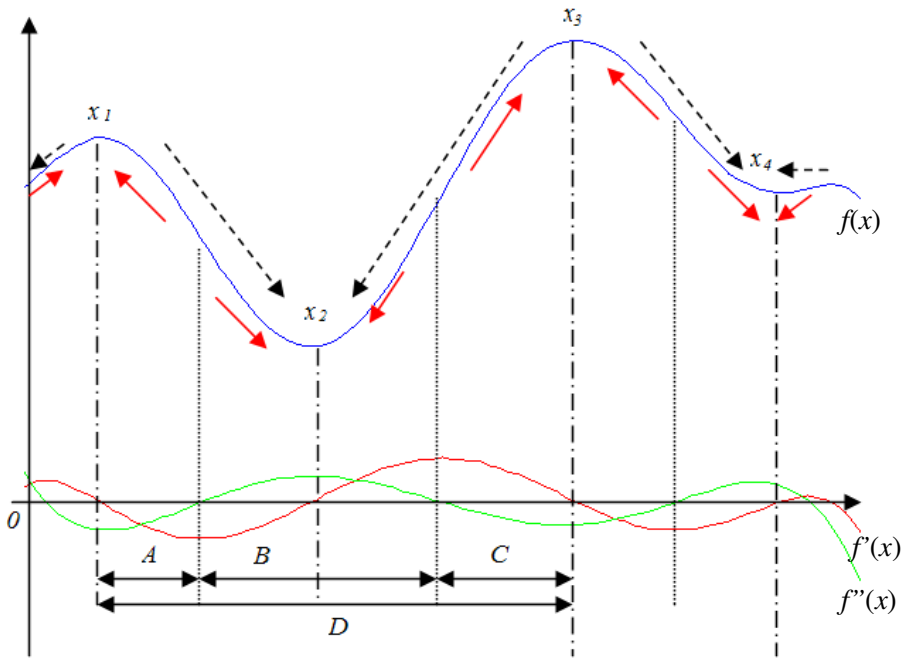


Figure 6.4 Convergence regions of recursive methods

In this figure, the blue, red and green curves indicate the functions $f(x)$, $f'(x)$, and $f''(x)$ respectively. The solid and dashed arrows denote the convergence directions of the Newton and the steepest gradient descent (or L-M) methods.

For simplicity, a 1-D minimization problem $\min f(x)$ is considered here. The Newton algorithm updates the solution by $\delta x = -\frac{\partial f}{\partial x} \Big/ \frac{\partial^2 f}{\partial x^2}$. Its incremental directions at different regions are shown in Figure 6.4 with red solid arrows [Ahn 2004]. If the current solution x lies at the region B, the solution of the Newton method moves toward the global minimum x_2. However, a local maximum will be caused at the regions A and C where $\frac{\partial^2 f}{\partial x^2} < 0$. By contrast, the SDG method is always capable of updating the solution along the downhill directions and thus its convergence domain is as large as D. As for the

L-M algorithm, $\frac{\partial^2 f}{\partial x^2} + \lambda > 0$ can always hold true as long as the damping factor λ is properly selected. Therefore, its convergence region is *D* as well.

It is worth noting that the L-M method converges at a local minimum x_4 when the current solution $x > x_3$. To overcome this problem, several initial solutions can be supplied at different regions, and the solution which yields the smallest error metric is regarded to be the optimal solution. A correct global minimum can certainly be obtained when at least one initial solution is located at the convergence domain of the L-M algorithm.

For multi-variable minimization problems, such as the six-variable problem of 3-D fitting in this thesis, a necessary condition $\frac{\partial^2 f}{\partial x^2} + \lambda > 0$ for a local minimum becomes: the second order derivative matrix $\mathbf{A} = \mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}$ should be positive definite [Fletcher 2000].

Implementation

Now we go back to the six-variable problem of free-form fitting in Equation (6.11). The L-M algorithm is adopted. At each iteration, the measurement points and motion parameters are updated by

$$\begin{cases} \mathbf{p} \leftarrow \mathbf{R}_1\mathbf{p} + \delta\,\mathbf{t} \\ \mathbf{R} \leftarrow \mathbf{R}_1\mathbf{R} \\ \mathbf{t} \leftarrow \mathbf{R}_1\mathbf{t} + \delta\,\mathbf{t} \end{cases} \tag{6.18}$$

with $\mathbf{R}_1 = \begin{bmatrix} \cos\delta\theta_z & \sin\delta\theta_z & 0 \\ -\sin\delta\theta_z & \cos\delta\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos\delta\theta_y & 0 & -\sin\delta\theta_y \\ 0 & 1 & 0 \\ \sin\delta\theta_y & 0 & \cos\delta\theta_y \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\delta\theta_x & \sin\delta\theta_x \\ 0 & -\sin\delta\theta_y & \cos\delta\theta_y \end{bmatrix}$ and $\delta\,\mathbf{t} = [\delta t_x, \delta t_y, \delta t_z]^T$. The key part of the programme is to calculate the Jacobian matrix,

$$\begin{cases} J_{i1} = \dfrac{\partial(z_i - f_i)}{\partial \theta_x} = y_i + z_i \dfrac{\partial f_i}{\partial y_i} \\ J_{i2} = \dfrac{\partial(z_i - f_i)}{\partial \theta_y} = -x_i - z_i \dfrac{\partial f_i}{\partial x_i} \\ J_{i3} = \dfrac{\partial(z_i - f_i)}{\partial \theta_z} = y_i \dfrac{\partial f_i}{\partial x_i} - x_i \dfrac{\partial f_i}{\partial y_i} \\ J_{i4} = \dfrac{\partial(z_i - f_i)}{\partial t_x} = -\dfrac{\partial f_i}{\partial x_i} \\ J_{i5} = \dfrac{\partial(z_i - f_i)}{\partial t_y} = -\dfrac{\partial f_i}{\partial y_i} \\ J_{i6} = \dfrac{\partial(z_i - f_i)}{\partial t_z} = 1 \end{cases} \tag{6.19}$$

In the equation $[x_i, y_i, z_i]^T = \mathbf{p}_i$ is an arbitrary measurement point and $f_i = f(x_i, y_i)$.

Since $\mathbf{J}^T\mathbf{J}$ is a positive semi-definite Hermitian matrix, its singular value decomposition result is the same with the eigen-decomposition $\mathbf{J}^T\mathbf{J} = \mathbf{U}\mathbf{S}\mathbf{U}^T$, where $\mathbf{U} \in \Re^{6\times6}$ is a unitary matrix, $\mathbf{S} = diag\{\sigma_1, \sigma_2, \cdots, .\sigma_6\}$ is a diagonal matrix, and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_6 \geq 0$ are the singular values [Golub 1996]. According to the matrix theory, $\mathbf{J}^T\mathbf{J}$ is positive definite if and only if $\sigma_6 > 0$ [Chong 2001]. It is evident that,

$$\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I} = \mathbf{U}(\mathbf{S} + \lambda\mathbf{I})\mathbf{U}^T \tag{6.20}$$

Therefore SVD does not need to be performed twice and the new singular values are $\sigma_i' = \sigma_i + \lambda$. So that the damping factor λ can be properly selected to guarantee $\sigma_6' = \sigma_6 + \lambda > 0$. A very large λ will decrease the step length of the motion parameters, thereby reducing the convergence rate; whilst a very small singular value will make the solution unstable. Hence λ is selected according to the smallest singular value σ_6. If $\sigma_6 < \varepsilon$, where ε is a user-defined threshold, e.g. 10^{-5}, set $\lambda = \varepsilon - \sigma_6$; otherwise set $\lambda = 0$ [Hansen 1998]. Figure 6.5 highlights the fitting procedure.

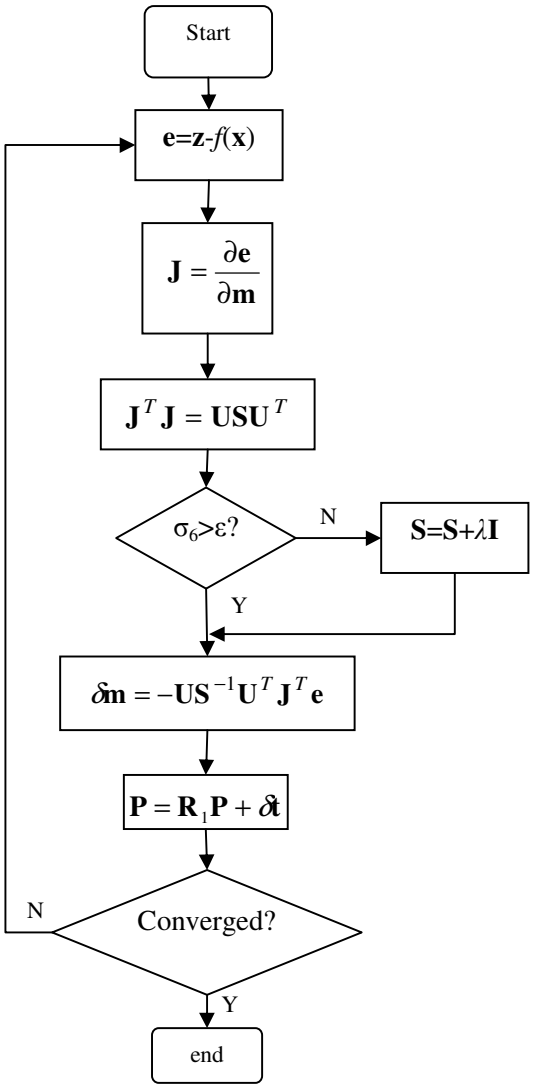
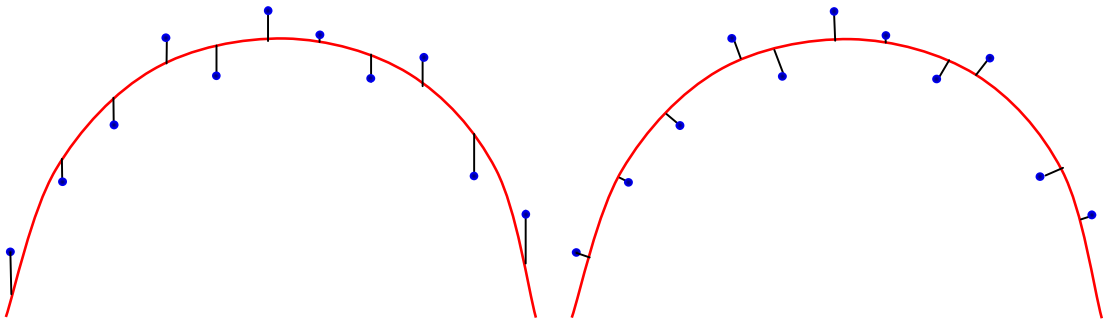


Figure 6.5 Scheme of the L-M fitting

6.2.2 The Orthogonal Distance Fitting of Explicit Surfaces

In the previous subsection, only the deviation in the *z* direction is considered, which is called *algebraic fitting* [Ahn 2001]. This approach is extensively applied in the metrology field because of its ease of implementation. However, its definition of error-distance does not coincide with measurement guidelines. The estimated fitting parameters will be biased, especially in the case there exist errors in the explanatory variables [DIN 1986, Ahn 2001, Sun 2007]. Consequently researchers have developed the *orthogonal distance fitting* (also termed *geometric fitting*) method. This technique intends to minimize the sum of the squared orthogonal distances from the measurement points to the nominal surface.



(a) Algebraic fitting (b) orthogonal distance fitting

Figure 6.6 Comparison of algebraic and geometric fitting

It can effectively overcome the bias problem of the algebraic fitting. Most researchers paid attention to the fitting of simple geometries like quadric surfaces, whose orthogonal distances can be directly calculated via closed-form methods [Ahn 2004, Sun 2007]. But the orthogonal distances are not so straightforward to find for free-form surfaces, and the computational cost may be dramatically increased if calculating the orthogonal projection points with recursive techniques.

Suppose that the explicit function of a template is given as $z = f(x, y)$. We aim to find an optimal rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$ to minimize the sum of the squared orthogonal distances from all the measurement points to the template,

$$E = \sum_{i=1}^{N} \left\| \mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i \right\|^2 = \sum_{i=1}^{N} \left\| \mathbf{p}'_i - \mathbf{q}_i \right\|^2 \tag{6.21}$$

Here $\mathbf{p}'_i = \mathbf{R}\mathbf{p}_i + \mathbf{t} = [x'_i, y'_i, z'_i]^T$ is an arbitrary measurement point after motion and $\mathbf{q}_i$ is its corresponding closest point on the template. If the geometry model of the template has been known, some intrinsic characteristics (shape parameters) $\mathbf{a} \in \Re^{p\times 1}$ may need to be fitted as well.

The coordinates of $\mathbf{q}_i$ are represented as $\mathbf{q}_i = [x'_i + \xi_i, y'_i + \zeta_i, f(x'_i + \xi_i, y'_i + \zeta_i; \mathbf{a})]^T$ and the weighting technique is incorporated for the sake of robustness. Then the error metric in Equation (6.21) becomes,

$$E = \sum_{i=1}^{N} w_i^2 [z'_i - f(x'_i + \xi_i, y'_i + \zeta_i)]^2 + \sum_{i=1}^{N} (u_i^2 \xi_i^2 + v_i^2 \zeta_i^2) = \mathbf{g}^T \mathbf{g} + \mathbf{h}^T \mathbf{h} \tag{6.22}$$

with $\mathbf{g} \in \Re^{N\times 1}: g_i = w_i [z'_i - f(x'_i + \xi_i, y'_i + \zeta_i)], i = 1, \cdots, N$

and $\mathbf{h} \in \Re^{2N\times 1} : h_i = \begin{cases} -u_{(i+1)/2}\xi_{(i+1)/2} & i = 1,3,\cdots,2N-1 \\ -v_{i/2}\zeta_{i/2} & i = 2,4,\cdots,2N \end{cases}$

So that $\{\xi_i\}$ and $\{\zeta_i\}$ can be taken as unknown variables and solved simultaneously with the six motion parameters $\theta_x, \theta_y, \theta_z, t_x, t_y, t_z$ and the intrinsic characteristics **a**.

Paul T. Boggs et al [Boggs 1987] proposed an efficient method to solve this nonlinear least squares problem based on the Levenberg-Marquardt algorithm.

Denoting the shape and motion parameters with a vector $\mathbf{m} \in \Re^{(p+6)\times 1}$ and denoting $\{\xi_i\}$ and $\{\zeta_i\}$ with a vector $\boldsymbol{\beta} \in \Re^{2N\times 1}$, Equation (6.22) can be solved iteratively by,

$$\begin{bmatrix} \mathbf{J} & \mathbf{V} \\ 0 & \mathbf{D} \\ \lambda^{1/2}\mathbf{S} & 0 \\ 0 & \lambda^{1/2}\mathbf{T} \end{bmatrix} \begin{bmatrix} \delta\mathbf{m} \\ \delta\boldsymbol{\beta} \end{bmatrix} = -\begin{bmatrix} \mathbf{g} \\ \mathbf{h} \\ 0 \\ 0 \end{bmatrix} \tag{6.23}$$

with $\mathbf{J} \in \Re^{N\times(p+6)} : J_{ij} = \dfrac{\partial g_i}{\partial m_j} = w_i \dfrac{\partial z_i'}{\partial m_j} - w_i \dfrac{\partial f_i}{\partial m_j}, i = 1,\cdots,N,\ \ j = 1,\cdots,p+6,$

$\mathbf{V} \in \Re^{N\times 2N} : V_{ij} = \dfrac{\partial g_i}{\partial \beta_j} = -w_i \dfrac{\partial f_i}{\partial \beta_j}$, $i = 1,\cdots,N, j = 1,\cdots,2N$. In fact, **V** has a 'staircase' structure,

$$\mathbf{V} = \begin{bmatrix} -w_1\dfrac{\partial f_1}{\partial \xi_1}, & -w_1\dfrac{\partial f_1}{\partial \zeta_1} & & & \\ & & -w_2\dfrac{\partial f_2}{\partial \xi_2}, & -w_2\dfrac{\partial f_2}{\partial \zeta_2} & \\ & & & & \ddots \\ & & & & -w_N\dfrac{\partial f_N}{\partial \xi_N}, \quad -w_N\dfrac{\partial f_N}{\partial \zeta_N} \end{bmatrix}$$

$\mathbf{D} \in \Re^{2N\times 2N} : \mathbf{D} = -diag\{u_1, v_1, u_2, v_2, \cdots, u_N, v_N\}$.

Here $\lambda > 0$ is a damping factor and $\mathbf{S} \in \Re^{(p+6)\times(p+6)}$ and $\mathbf{T} \in \Re^{2N\times 2N}$ are two scaling matrices related with the Levenberg-Marquardt algorithm. In practice, we set $\mathbf{S} = \mathbf{I}^{(p+6)\times(p+6)}$ and $\mathbf{T} = \mathbf{0}$.

The normal function of Equation (6.23) is,

$$\begin{bmatrix} \mathbf{J}^T\mathbf{J} + \lambda\mathbf{I} & \mathbf{J}^T\mathbf{V} \\ \mathbf{V}^T\mathbf{J} & \mathbf{V}^T\mathbf{V} + \mathbf{D}^2 \end{bmatrix} \begin{bmatrix} \delta\mathbf{m} \\ \delta\boldsymbol{\beta} \end{bmatrix} = -\begin{bmatrix} \mathbf{J}^T\mathbf{g} \\ \mathbf{V}^T\mathbf{g} + \mathbf{D}\mathbf{h} \end{bmatrix} \tag{6.24}$$

So that,

$$\delta\mathbf{m} = -(\mathbf{J}^T\mathbf{M}\mathbf{J} + \lambda\mathbf{I})^{-1}\mathbf{J}^T\mathbf{M}(\mathbf{g} - \mathbf{V}\mathbf{D}^{-1}\mathbf{h}) \tag{6.25}$$

and $$\delta\boldsymbol{\beta} = -\mathbf{D}^{-2}[\mathbf{V}^T\mathbf{M}(\mathbf{g} + \mathbf{J}\delta\mathbf{m} - \mathbf{V}\mathbf{D}^{-1}\mathbf{h}) + \mathbf{D}\mathbf{h}] \tag{6.26}$$

with $\mathbf{M} = \mathbf{I} - \mathbf{V}(\mathbf{V}^T\mathbf{V} + \mathbf{D}^2)^{-1}\mathbf{V}^T$. In fact, it is proved to be a diagonal matrix,

$$\mathbf{M} = diag\left\{ 1 \Big/ \left[1 + \left(\frac{w_i}{u_i} \frac{\partial f_i}{\partial \xi_i} \right)^2 + \left(\frac{w_i}{v_i} \frac{\partial f_i}{\partial \zeta_i} \right)^2 \right] \right\}, i = 1, \cdots, N . \tag{6.27}$$

This algorithm avoids calculating orthogonal projections successively, and the correspondence points $\{\mathbf{q}_i\}$ can be straightforwardly obtained from the measurement points. Thus the computational cost of this method is in the same order with the algebraic fitting.

6.2.3 The Orthogonal Distance Fitting of Parametric Surfaces

The above algorithm works well for the orthogonal distance fitting (ODF) of explicit surfaces. However, explicit functions are not always available for free-form surfaces. Parametric representations are more common, for instance NURBS surfaces. In most situations, the Cartesian coordinates are nonlinear with respect to the location parameters $\{u_i\}$ and $\{v_i\}$. Additionally, the number of measurement points in practice is probably very large, sometimes over a million points, which makes the size of the observation matrix increasing dramatically. Then the computational cost and memory usage will be rather tedious. As a consequence it is practical to use a nested iteration scheme— to solve the foot-point parameters alternately with the motion parameters,

$$\min_{\mathbf{m}} \sum_{i=1}^{N} \min_{u_i, v_i} \left\| \mathbf{p}_i - \mathbf{q}_i \right\|^2 \tag{6.28}$$

That means, firstly find the closest template point (projection point) corresponding to each measurement point in the inner iteration, and then work out the optimal motion parameters and intrinsic characteristics at the outer iteration.

Some closed form techniques have been developed to calculate the orthogonal projection points for simple geometries [Ahn 2004]. Whereas for general shaped parametric functions, the iterative Newton-Raphson algorithm can be adopted. For example, the two-stage approach stated in Subsection 2.3.2 can be employed to solve the point-projection problem of NURBS surfaces.

After all the projection points $\{\mathbf{q}_i\}$ are obtained, the motion and shape parameters will be updated subsequently. We define $\mathbf{g} \in \Re^{3N\times 1}$,

$$g_k = \begin{cases} x_i' - X_i & k = 3i-2 \\ y_i' - Y_i & k = 3i-1 \\ z_i' - Z_i & k = 3i \end{cases}, \quad i = 1, \cdots, N$$

where $[X_i, Y_i, Z_i]^T = \mathbf{q}_i$ is the projection point associated with $\mathbf{p}_i$.

Hence Equation (6.28) can be rewritten as,

$$\min_{\mathbf{m}} \sum_{k=1}^{3N} g_k^2 = \min_{\mathbf{m}} \mathbf{g}^T \mathbf{g} \tag{6.29}$$

It can be solved with the Levenberg-Marquardt algorithm,

$$\left[\left(\frac{\partial \mathbf{g}}{\partial \mathbf{m}} \right)^T \left(\frac{\partial \mathbf{g}}{\partial \mathbf{m}} \right) + \lambda \mathbf{I} \right] \delta \mathbf{m} = - \left(\frac{\partial \mathbf{g}}{\partial \mathbf{m}} \right)^T \mathbf{g} \tag{6.30}$$

The three rows of the Jacobian matrix $\mathbf{J} = \partial \mathbf{g} / \partial \mathbf{m} \in \Re^{3N\times(p+6)}$ associated with the point $\mathbf{p}_i$ are

$$\begin{bmatrix} J_{3i-2} \\ J_{3i-1} \\ J_{3i} \end{bmatrix} = \frac{\partial \mathbf{p}_i'}{\partial \mathbf{m}} - \frac{\partial \mathbf{q}_i}{\partial \mathbf{u}_i} \frac{\partial \mathbf{u}_i}{\partial \mathbf{m}} \tag{6.31}$$

Here $\mathbf{u}_i = [u_i, v_i]^T$ are the foot-point parameters of $\mathbf{q}_i$. When the measurement point $\mathbf{p}_i$ moves, its closest point $\mathbf{q}_i$ moves as well, so that the corresponding foot-point parameters $\mathbf{u}_i$ shall be updated simultaneously. That means the foot-point parameters of $\{\mathbf{q}_i\}$ are relevant with the motion parameters $\mathbf{m}$ in each iteration. Here the parameter dependency $\partial \mathbf{u}_i / \partial \mathbf{m}$ will be derived as follows. Each pair of points are nearest to each other and the following relation always holds true [Ahn 2004],

$$\frac{\partial}{\partial \mathbf{u}_i}(\mathbf{p}_i - \mathbf{q}_i)^T(\mathbf{p}_i - \mathbf{q}_i) = -2\left(\frac{\partial \mathbf{q}_i}{\partial \mathbf{u}_i}\right)^T(\mathbf{p}_i - \mathbf{q}_i) = \mathbf{0} \tag{6.32}$$

For the sake of clarity, the subscript *i* is omitted and the partial derivatives $\partial\mathbf{q}/\partial\mathbf{u}$ is written as $\mathbf{q}_\mathbf{u}$, so that,

$$\frac{\partial}{\partial \mathbf{m}}\left(-\mathbf{q}_\mathbf{u}^T(\mathbf{p}-\mathbf{q})\right)$$

$$= \mathbf{q}_\mathbf{u}^T\mathbf{q}_\mathbf{u}\mathbf{u}_\mathbf{m} - \begin{bmatrix} \mathbf{q}_{uu}^T(\mathbf{p}-\mathbf{q}) & \mathbf{q}_{uv}^T(\mathbf{p}-\mathbf{q}) \\ \mathbf{q}_{uv}^T(\mathbf{p}-\mathbf{q}) & \mathbf{q}_{vv}^T(\mathbf{p}-\mathbf{q}) \end{bmatrix}\mathbf{u}_\mathbf{m} - \mathbf{q}_\mathbf{u}^T\mathbf{p}_\mathbf{m}$$

$$= \mathbf{0}$$

We obtain,

$$\mathbf{u}_\mathbf{m} = \left\{\mathbf{q}_\mathbf{u}^T\mathbf{q}_\mathbf{u} - \begin{bmatrix} \mathbf{q}_{uu}^T(\mathbf{p}-\mathbf{q}) & \mathbf{q}_{uv}^T(\mathbf{p}-\mathbf{q}) \\ \mathbf{q}_{uv}^T(\mathbf{p}-\mathbf{q}) & \mathbf{q}_{vv}^T(\mathbf{p}-\mathbf{q}) \end{bmatrix}\right\}^{-1}\mathbf{q}_\mathbf{u}^T\mathbf{p}_\mathbf{m} \tag{6.33}$$

Substituting Equation (6.33) into (6.31), then the increment of the solution in Equation (6.30) will be obtained.

A necessary condition for Equation (6.29) to converge at a local minimum is that the observation matrix $\mathbf{A} = \mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}$ is positive definite, so that the damping factor λ can be selected according to the smallest singular value of the matrix $\mathbf{J}^T\mathbf{J}$.

For a uniform NURBS or B-spline surface, the explicit representations of the second order derivatives in Equation (6.33) can be obtained. They are even simpler to calculate than the first order derivatives, and will be reserved when the template is represented as a NURBS or B-spline surface, thus leading to a damped Newton minimization. However, for most general-shaped parametric surfaces, the second order derivatives are rather tedious to be derived from the complex surface functions. Therefore, they will be neglected when the residuals are very small and the surface is very smooth, i.e. $\begin{bmatrix} \mathbf{q}_{uu}^T(\mathbf{p}-\mathbf{q}) & \mathbf{q}_{uv}^T(\mathbf{p}-\mathbf{q}) \\ \mathbf{q}_{uv}^T(\mathbf{p}-\mathbf{q}) & \mathbf{q}_{vv}^T(\mathbf{p}-\mathbf{q}) \end{bmatrix}$ is much smaller than $\mathbf{q}_\mathbf{u}^T\mathbf{q}_\mathbf{u}$. To guarantee convergence in this case, the damping factor needs to be adapted carefully. In the program we update it in a hypothesize-and-test scheme as introduced in Subsection 6.2.1.

6.3 Robust Fitting

The sum of the squared distances between corresponding point pairs is used above as the error metric of fit. It is easy to implement and, more importantly, unbiased when the error is normally distributed [Barker 2004]. However, its solution is very sensitive to large errors. Measurement data may contain outliers or missing data due to improper operation, poor reflectivity of the specimen or environmental noise. The workpiece can also have manufacturing defects, such as pits and troughs involved in honed surfaces. As a result, the fitting result will be distorted or even break down.

To improve the robustness of the fitted results, various techniques have been proposed [Rey 1983]. Among these methods, the l_1 norm pays less attention to the wild points and concentrates on the vast majority of the data points; therefore it has attracted extensive attention. But it has discontinuous derivatives and thus is difficult to solve. Hunter and Lange proposed an algorithm based on the *Majorize-Minimize theory* [Hunter 2000]. A continuous surrogate function is adopted to approximate the initial l_1 norm objective function, which is easy to code and shows distinctive computational superiority. Therefore it is adopted here.

Suppose we want to minimize an objective function $f(\mathbf{m})$ with $\mathbf{m} \in \Re^{p\times 1}$. If the current solution at the k-th iteration is $\mathbf{m}^{(k)}$, the majorize-minimize theory defines a surrogate function $g(\mathbf{m} \mid \mathbf{m}^{(k)})$ satisfying

$$\begin{cases} g(\mathbf{m}^{(k)} \mid \mathbf{m}^{(k)}) = f(\mathbf{m}^{(k)}) \\ g(\mathbf{m} \mid \mathbf{m}^{(k)}) \geq f(\mathbf{m}) \text{ for all } \mathbf{m} \end{cases} \tag{6.34}$$

Here $g(\mathbf{m} \mid \mathbf{m}^{(k)})$ is said to majorize $f(\mathbf{m})$ at $\mathbf{m}^{(k)}$. In the next iteration, a new solution $\mathbf{m}^{(k+1)}$ is found to minimize $g(\mathbf{m} \mid \mathbf{m}^{(k)})$. Since the surrogate function $g(\mathbf{m} \mid \mathbf{m}^{(k)})$ can be selected much simpler than the initial objective function $f(\mathbf{m})$, thus the complexity of the optimization problem can be greatly reduced.

For the l_1 norm fitting problem,

$$E = \sum_{i=1}^{N} \left\| \mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i \right\| = \sum_{i=1}^{N} r_i \tag{6.35}$$

The cost function for each point is $\rho_i = r_i$. A small perturbation $\varepsilon > 0$ is introduced into the error metric,

$$\rho_i^{\varepsilon} = \rho_i - \varepsilon \ln(\varepsilon + r_i) = r_i - \varepsilon \ln(\varepsilon + r_i)$$

The resultant change in the objective function is

$$\left| E^{\varepsilon} - E \right| = \left| \sum_{i=1}^{N} (\rho_i^{\varepsilon} - \rho_i) \right| = \left| -\sum_{i=1}^{N} \varepsilon \ln(\varepsilon + r_i) \right| \le -\varepsilon N \ln \varepsilon = \tau$$

thus the constant ε can be properly selected based on the overall error threshold defined on the change of the objective function τ.

If the surrogate function is chosen to be,

$$g^{\varepsilon}(r_i \mid r_i^{(k)}) = \frac{1}{2} \frac{r_i^2}{\varepsilon + r_i^{(k)}} + c_i$$

where $r_i^{(k)}$ is the current residual at the *k*-th iteration and the constants $\{c_i\}$ are selected properly so that $g^{\varepsilon}(r_i \mid r_i^{(k)}) = \rho_i^{\varepsilon}$. Then the new objective function turns out to be,

$$E^{\varepsilon} = \sum_{i=1}^{N} g^{\varepsilon}(r_i \mid r_i^{(k)}) = \sum_{i=1}^{N} \left(\frac{1}{2} \frac{r_i^2}{\varepsilon + r_i^{(k)}} + c_i \right) \tag{6.36}$$

It is evident that the fixed constants $\{c_i\}$ and coefficient 1/2 do not influence the solution, and Equation (6.36) is equivalent to the reweighted least squares problem,

$$E^{\varepsilon} = \sum_{i=1}^{N} w_i r_i^2 \quad \text{with} \quad w_i = \frac{1}{\varepsilon + r_i^{(k)}} \tag{6.37}$$

Therefore, all the algorithms in this chapter can be modified accordingly. Firstly we consider the SVD technique of the ICP algorithm. The centroids of the two surfaces are now calculated in this way,

$$\begin{cases} \mathbf{p}_c = \sum_{i=1}^{N} w_i \mathbf{p}_i \Big/ \sum_{i=1}^{N} w_i \\ \mathbf{q}_c = \sum_{i=1}^{N} w_i \mathbf{q}_i \Big/ \sum_{i=1}^{N} w_i \end{cases} \tag{6.38}$$

and the correlation matrix of Equation (6.5) becomes,

$$\mathbf{H} = \sum_{i=1}^{N} w_i \mathbf{p}_i \mathbf{q}_i^T \tag{6.39}$$

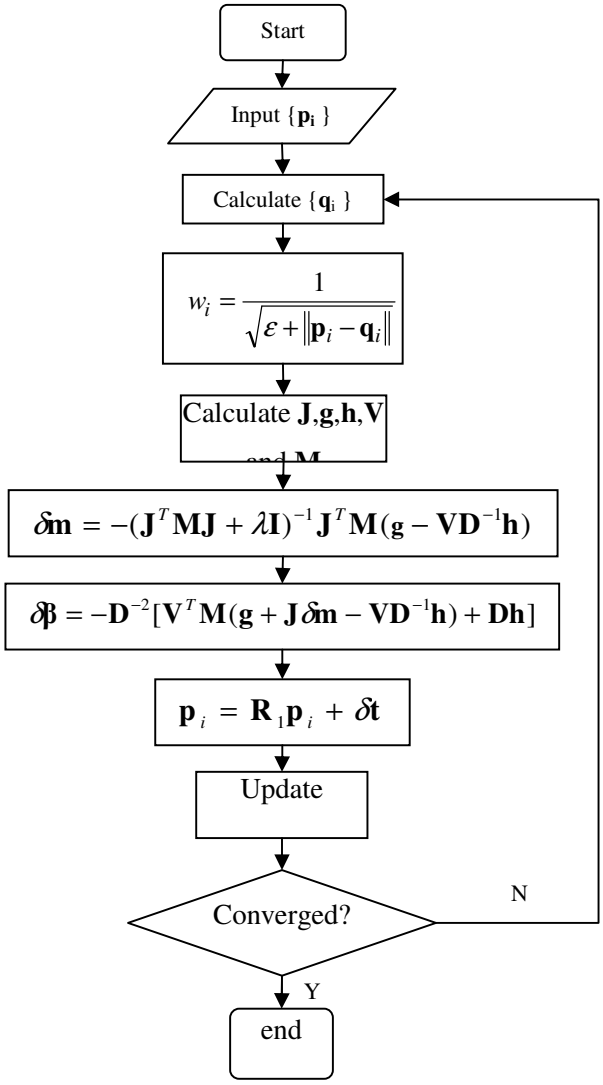


Figure 6.7 Flowchart of robust ODF of explicit surfaces

As regards the Levenberg-Marquardt algorithm, only (6.17) is changed into a weighted form,

$$\delta\mathbf{m} = -(\mathbf{J}^T\mathbf{W}\mathbf{J} + \lambda\mathbf{D})^{-1}\mathbf{J}^T\mathbf{W}\mathbf{e} \tag{6.40}$$

with $\mathbf{W} = diag\{w_1, w_2, \cdots, w_N\}$ and $w_i = \dfrac{1}{\varepsilon + |z_i - f(x_i, y_i)|}$

It is worth noting that the weighting factors in the Subsection 6.2.2 are a little different. They should be calculated as,

$$w_i = \frac{1}{\sqrt{\varepsilon + \|\mathbf{p}_i - \mathbf{q}_i\|}} \tag{6.41}$$

$\{u_i\}$ and $\{v_i\}$ are chosen according to $\{w_i\}$ and the quotient between the lateral and vertical errors. The l_1 norm ODF fitting of explicit surfaces is shown in Figure 6.7,

The solution of the motion parameters in Equation (6.30) appears similar with the explicit-surface fitting, but they are not the same. The residual vector is $\mathbf{g} \in \Re^{3N\times 1}$, therefore the weighting matrix is

$$\mathbf{W} = diag\{w_1, w_1, w_1, w_2, w_2, w_2, \cdots, w_N, w_N, w_N\}, w_i = \frac{1}{\varepsilon + \|\mathbf{p}_i - \mathbf{q}_i\|} \tag{6.42}$$

The fitting procedure of the l_1 norm ODF of parametric surfaces is presented in Figure 6.8,

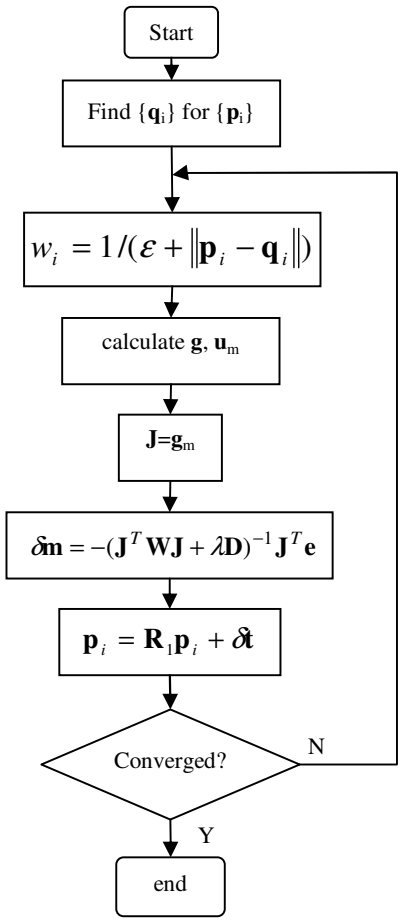


Figure 6.8 Flowchart of robust ODF of parametric surfaces

It needs to be emphasized that this leads to an approximate l_1 norm regression, i.e. the cost function behaves like least squares and approaches l_1 norm when the residuals get larger. This technique is not the only way to improve the robustness of the solution. The l_1 norm corresponds to the maximum likelihood estimation for a double exponential error distribution (Laplace distribution) [Norton 1984]. However, the actual distribution of the measurement data is not easy to define in practice. We do not intend to model the specific distribution of error, but to overcome the influence of outliers and defects. Of course, if the actual distribution of error has been known beforehand, the corresponding optimal error metric will be adopted. In this case the technique introduced here still works, as long as the error metric can be transferred into a reweighted least squares form, and appropriate weights can be assigned accordingly.

6.4 Simulation and Experimental Results

Example 1 Comparison of ICP and L-M Methods

Firstly we compare the performance of the Iterative Closest Point technique and the Levenberg-Marquardt algorithm.

A femoral knee joint replacement is taken as an example. The coordinate system of this model is given in Figure 6.9. The directions of the three axes are defined based on the planes of the brass support, thus they can be aligned very well. But there is no salient reference datum to localize the origin of the coordinate. In the CAD model, the origin of the x-axis is defined as the central point of the inter-condylar notch between the two condyles of femur, and the origin of the z axis is defined at the ultimate point of the workpiece. When establishing the measurement coordinate system manually, it is very difficult to find the exact position of the notch's mid-point and the ultimate point of the condyles. Actually we measured five points at the arc of the notch by the Carl Zeiss PRISMO CMM, and fitted a circle with CALYPSO. This circle's centre was applied to define the x origin. Then a cloud of points were scanned at the top of the lateral condyle of femur. The point with the greatest z coordinate was employed to localize the z origin.

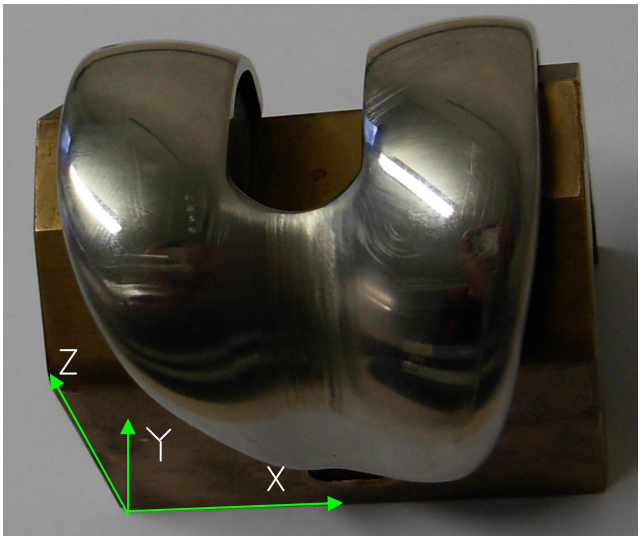


Figure 6.9 CoCr femoral knee joint

2642 data points were measured with spacing d=0.5 mm at the top of the lateral condyle of this joint. CMM evaluates its form error using the software HOLOS, as plotted in Figure 6.10.

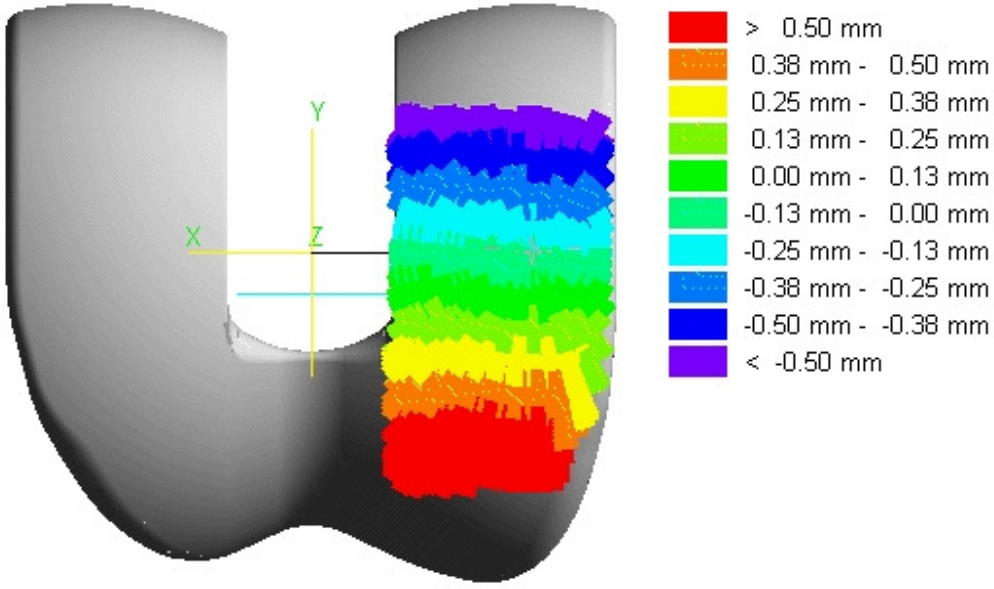


Figure 6.10 Residual map plotted by HOLOS

In this figure the peak-to-valley error is greater than 1.0 mm. It is obvious that a misalignment exists between the two coordinate systems. To work out the correct form quality of this workpiece, we will transform the measurement data properly to find a best matching between the data and the template. 58×90 points are uniformly sampled with spacing D=0.4 mm at the same part of the CAD model. Of course, the sampled area of

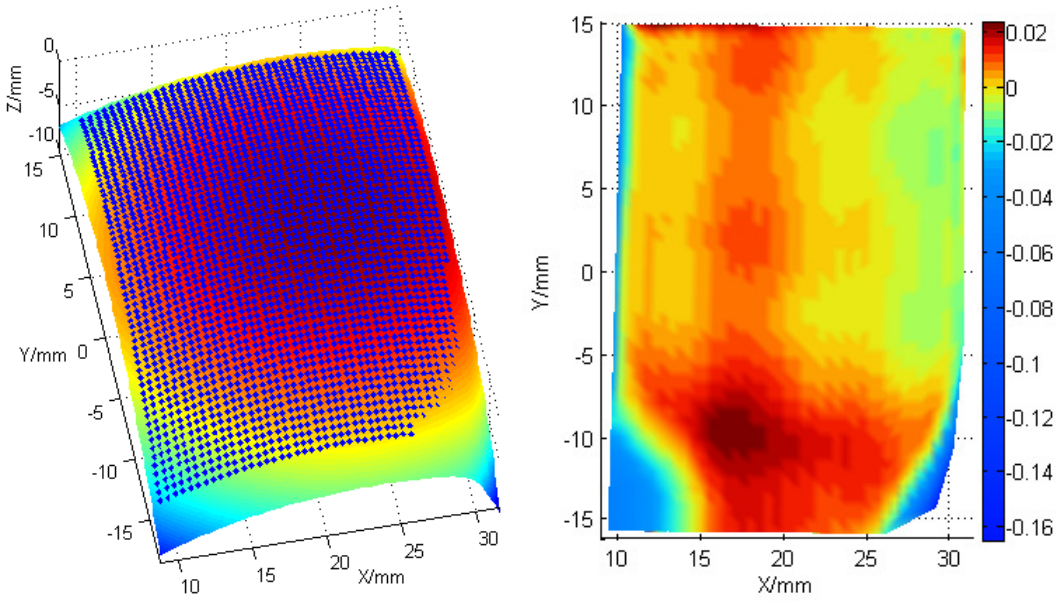
the template needs to be greater than the measurement data. The model points are reconstructed into a uniform NURBS surface with reconstruction error within ±2.0 μm. In this NURBS system 32×22 control points are applied.

Then we fit the measurement data with this NURBS template using the L-M algorithm presented in Section 6.2.1. The residual's amplitude parameters S_a, S_q, and S_z are employed to measure the goodness-of-fit. The fitting programme is coded in Matlab and run on a NEC PC with Intel Pentium 4 CPU 3.00GHz, 2.00GB of RAM. It runs eight iterations and takes 3.362 seconds. The increments of the six motion components and the resultant error parameters at each iteration are given in Table 6.1.

Iterat Num	$\delta\theta_x$ / °	$\delta\theta_y$ / °	$\delta\theta_z$ / °	δT_x/μm	δT_y/μm	δT_z/μm	S_a/μm	S_q/μm	S_z/μm
0	0	0	0	0	0	0	287.96	336.91	1364.7
1	2.4801	-1.0719	2.4348	-28.560	-42.723	-45.343	16.05	20.76	226.06
2	-6.4676	0.9543	-0.1090	29.883	130.598	51.004	227.45	227.98	236.85
3	1.5000	0.8889	-2.7419	25.760	6.526	18.347	22.26	25.36	245.86
4	0.8735	-0.2046	0.0202	-6.877	-18.710	-13.365	11.05	15.25	195.20
5	0.2672	-0.1345	-0.1778	-3.871	-3.016	-7.945	8.76	14.24	190.39
6	0.0253	-0.0819	-0.2353	-2.372	2.652	-4.504	8.709	14.22	189.00
7	0.0116	-0.0397	-0.1035	-1.133	1.161	-2.155	8.70	14.22	188.31
8	-0.0015	-0.0096	-0.0107	-0.272	0.176	-0.518	8.69	14.22	188.18

Table 6.1 Parameter update of the L-M algorithm

This L-M algorithm converges after eight iterations. The relative deviation between the two surfaces is reduced by more than one order of magnitude. Actually the residual map contains the reconstruction error of the NURBS surface. Since the reconstruction error is relatively much smaller and the manufacturing error of the workpiece dominates in the fitting residual, thus it is acceptable to evaluate the form quality of the knee joint replacement via the fitting residual map.



(a) fitting result (b) residual

Figure 6.11 Fitting result and error map of the L-M method

In contrast, the ICP method directly matches two sets of discrete points, consequently causing no reconstruction error. But we need to gain the residuals so as to assess the fitting quality, thus a continuous representation of the template is also required. It needs to be clarified that surface reconstruction is implemented here to calculate the final matching residual and it is not necessary in the ICP matching procedure.

We sampled template points uniformly with spacing D varying from 0.2 mm to 0.8 mm. The k-D tree was applied to accelerate the closest-point searching and the SVD technique was adopted to update the motion vector in each iteration. The Matlab matching programme ran 20 iterations in each case. Table 6.2 lists the obtained error parameters of the residual and the positional transformations with respect to the initial location. Here N indicates the number of the template points and *Time* refers to the running time of the Matlab programme.

It can be seen that the matching result is not significantly affected by the density of the template points, and a very small spacing does not necessarily lead to a better matching result, whilst yielding lower efficiency. Thus we recommend to adopt $d < D < 2d$. Here d and D are the densities of the data and template points respectively.

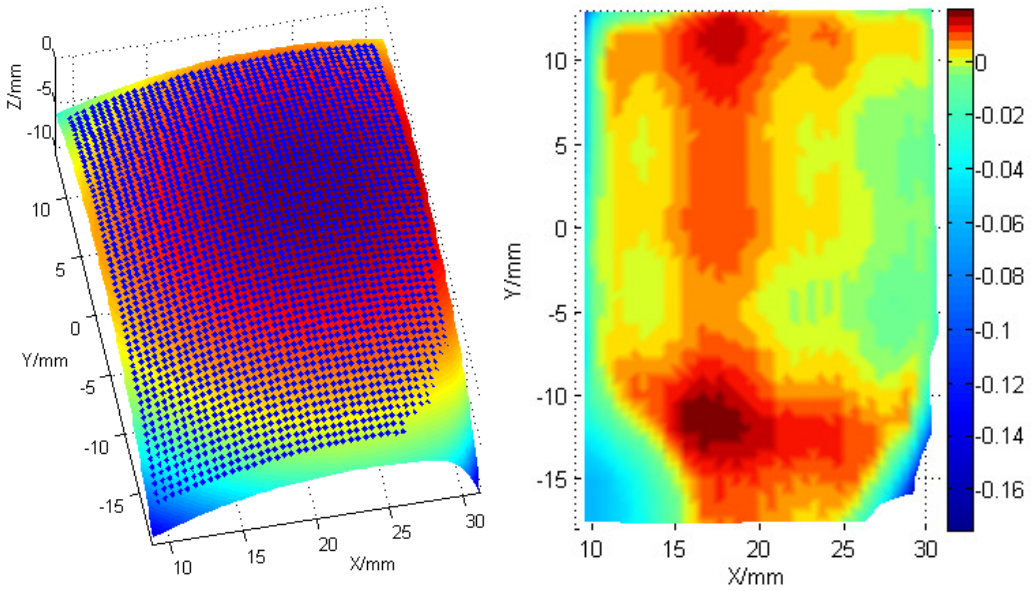
D/mm	N	θ_x / °	θ_y / °	θ_z / °	T_x/μm	T_y/μm	T_z/μm	S_a/μm	S_q/μm	S_z/μm	*Time*/sec
0.2	20406	1.899	-0.036	0.240	47.22	97.39	27.44	8.56	14.88	192.80	3.713
0.3	9044	1.974	-0.119	0.216	-2.32	55.86	21.44	9.00	14.82	192.65	3.668

0.4	5130	1.864	-0.028	0.237	48.14	111.88	29.93	9.29	14.88	193.22	3.377
0.5	3312	2.194	-0.449	0.204	-197.72	-120.73	-16.65	15.82	20.31	187.77	3.479
0.6	2280	1.998	-0.071	0.681	44.69	21.38	23.02	8.47	15.13	194.57	2.965
0.7	1683	1.953	-0.117	0.306	7.56	59.61	20.73	8.52	14.98	190.15	2.795
0.8	1305	1.920	-0.024	0.195	49.98	96.40	27.51	8.50	15.06	195.09	2.801
0.9	1040	1.947	-0.131	0.348	-8.18	62.07	15.57	9.13	15.61	194.00	2.791
1.0	828	1.753	-0.001	0.206	56.11	127.74	30.15	16.99	21.62	191.54	2.790

Table 6.2 ICP matching results with different model densities

The matching result seems unsatisfactory when D=0.5 mm, i.e. when the densities of the measurement data and template points are equal. This is not hard to understand. ICP intends to draw the measurement points toward their correspondences so as to minimize the Euclidean distances between them. In practice, the actual positions of the two points in one pair are rarely coincident with each other. If the densities of the two point sets are different, the pull force exerted on these measurement points is averaged, so that the lateral force caused by the relative X-Y shifts in point pairs can be cancelled. Hence their overall effect is: the measurement surface is moved toward its correct matching location. However, when the two point sets have the same density, the lateral shifts between correspondence pairs are along the same direction. It was made worse, as the coordinate variation in z direction is less than x and y, so the X-Y deviations will play a main role in the Euclidean distances. Therefore the ICP turns out to overlap the X-Y coordinates of the point pairs, instead of their correct positions in accordance with the surface shape, i.e. a local minimum is caused. To avoid wrong matching results, the template points should be in a different distribution scheme with the measurement data (e.g. one raster, and the other circular), or at least have different sampling densities.

Figure 6.12 shows the ICP fitting result with D=0.6 mm. Its error map is almost the same with the L-M technique.



(a) fitting result (b) residual

Figure 6.12 Fitting result and error map of the ICP method

This example demonstrates that ICP may be trapped at a local minimum. It converges slowly and usually needs more than 15 iterations. In contrast, the L-M algorithm is able to get a more accurate and more stable fitting result with a faster convergence rate. Its main shortcoming is the template surface requires a continuous representation, which is essential to construct a Jacobian matrix. For a smooth surface, even if the template is provided as a discrete point set, it is still recommended to firstly reconstruct it into continuous functions, and then fit the data using the Levenberg-Marquardt algorithm, instead of directly matching the two surfaces with the ICP method.

Example 2 Verification of the ODF Algorithm for Explicit Surfaces

In the previous example, only the noise in the z direction is considered, i.e. the x and y coordinates of the measurement data are taken as ideal values. But this does not hold true in practice; instead, some instruments have larger uncertainty in the lateral directions than the vertical one. If only taking the z deviation into the error metric, the fitted parameters may be biased, so that the orthogonal distance fitting (ODF) algorithm can be adopted.

In order to make the added noise more 'real', the *Fractional Brownian Motion* is employed [Mandelbrot 1968]. A normalized fractional Brownian motion (fBm) $B^H(x)$ on $[0,T], T \in \Re$ is a continuous-time Gaussian process starting at zero, with mean zeros and a correlation function of,

$$E\left[B^H(x)B^H(y)\right]=\frac{1}{2}\left(|x|^{2H}+|y|^{2H}-|x-y|^{2H}\right) \tag{6.47}$$

Here $H\in[0,1]$ is called the *Hurst index* or *Hurst parameter*. In this example, it is set to be H=0.5.

Figure 6.13 illustrates the topography of random noise calculated by fBm. Its standard deviation is σ=3.0 μm.

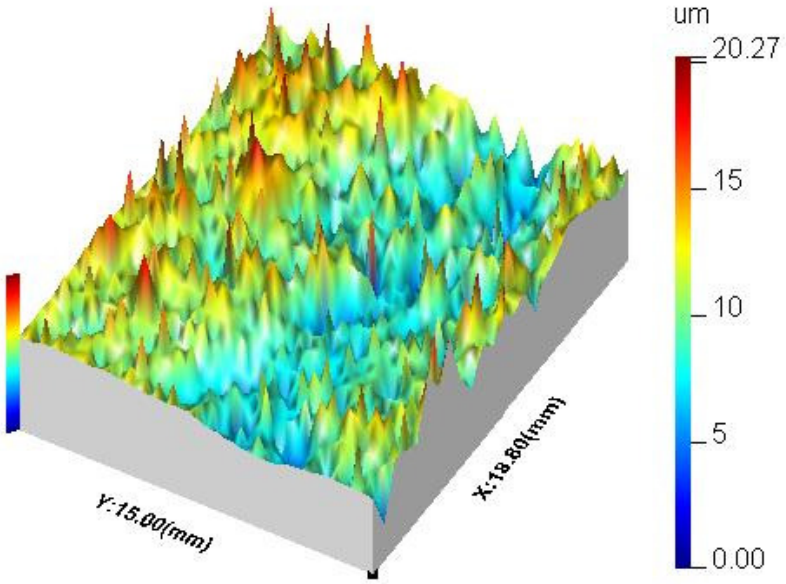


Figure 6.13 Adding fractal Brownian motion as measurement noise

The upper part of a cylinder with axis r=10.0 mm and length l=15.0 mm is adopted to verify the ODF algorithm. The width of this section is set to be w=18.8 mm, as plotted in Figure 6.14. The steepest slope is $\alpha=\arcsin(18.8/2/10.0)=70.05°$.

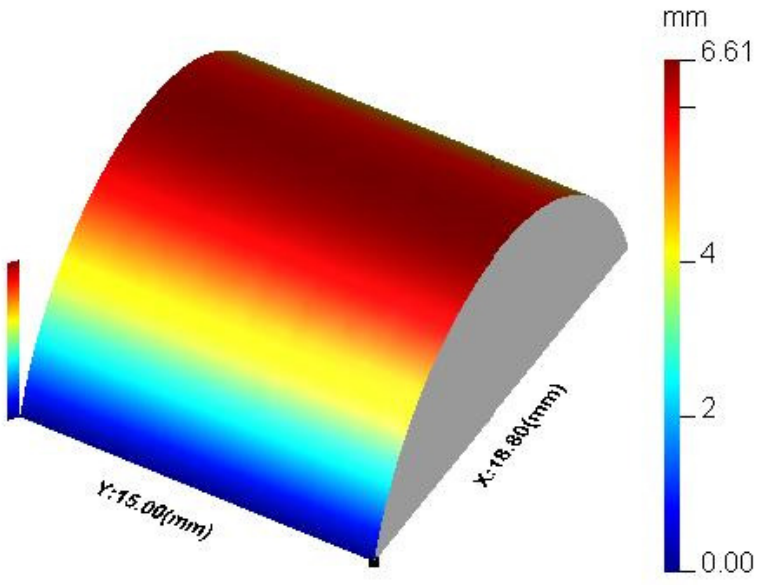


Figure 6.14 A cylinder model

Data points are sampled on the cylinder with spacing 0.4 mm in the x direction and 0.3 mm in the y direction. Then fBm noise with σ=3.0 μm is introduced into the x, y, and z coordinates of the data to simulate the measurement error. The data is written into a SDF file and fitted with the standard commercial metrology software Talymap. It is known that Talymap fits geometries using the nonlinear algebraic least-squares algorithm. A radius $\hat{r}$=9.997 mm is recovered from the noisy data. If we fit the same data using the ODF algorithm introduced in Section 6.2.2, a better result of $\hat{r}$=9.998 mm is obtained. Then we change the standard deviation of the noise into 10.0 μm and 30.0 μm respectively. The obtained radii of AF and ODF methods are listed below.

Method	σ/μm	$\hat{r}$ /mm	Vertical residuals/μm		
			S_a	S_q	S_z
AF	3.0	9.997	2.374	3.289	35.67
	10.0	9.990	7.919	10.975	118.92
	30.0	9.970	23.814	33.045	356.92
ODF	3.0	9.998	2.368	3.255	34.20
	10.0	9.994	7.894	10.846	114.41
	30.0	9.982	23.712	32.565	341.35

Table 6.3 Comparison of AF with ODF

It can be seen that the fitted radii of ODF are always better than AF. This effectively demonstrates the capability of the ODF algorithm on overcoming the bias in the fitted parameters. In order to examine the quality of fit more completely, we calculate the

residuals of these two algorithms $e=z-\sqrt{r^2-x^2}$. Their S_a, S_q, and S_z values are also given in Table 6.3. It is evident that the ODF technique attempts to minimize the orthogonal errors, instead of the vertical residuals, thus ODF does not show distinctive superiority over AF on minimizing the residual errors.

Talymap does not develop programs to fit cylinders at non-standard positions. Our ODF programs can straightforwardly solve this problem. Now we fix the magnitude of the noise to be σ=3.0 μm and rotate the cylinder along the x and z axes with different angles. It is worth noting that this cylinder is translationally and rotationally symmetric about the y axis, hence only the remaining four degrees of freedom are considered in the fitting programme. The fitted results in different cases are presented in Table 6.4.

The quality of the fitted radius is not significantly influenced by the initial position of the cylinder. Since the uncertainty of the fitted result is mainly affected by the magnitude of the measurement noise, whilst the rotation angles determine the convergence property of the fitted result. It is proved that even if the rotation angle is as large as 20˚, the ODF algorithm can still obtain a correct result.

θ_x, θ_z	$\hat{r}$ /mm	$\hat{\theta}_x$ /˚	$\hat{\theta}_z$ /˚	Vertical residuals/μm			Orthogonal errors/μm
				S_a	S_q	S_z	*Peak-to-valley*
θ_x =-0.5˚, θ_z=0.8˚	9.9980	-0.501	0.799	2.373	3.258	34.173	20.090
θ_x =-5.0˚, θ_z=3.0˚	9.9980	-5.001	2.998	2.406	3.302	34.163	19.757
θ_x =20˚, θ_z=-4.5˚	10.0009	20.018	-4.497	2.573	3.829	88.090	21.567

Table 6.4 ODF fitting results of the cylinder at non-standard positions

This ODF algorithm is a general-purposed method and works for any smooth shapes with explicit and differentiable functions. To fit standard geometries like cylinders or spheres, some specific algorithms are recommended. Here an example of cylinder is given only to validate the effectiveness and non-biasedness of this ODF algorithm. When the measurement data is highly curved and contains some rather steep regions, this fitting technique is preferable if we are interested in restoring the shape parameters. But if our purposes are only to remove the form from the data and to analyze the micro-topography, or the surface is sufficiently planar, or the explanatory coordinates of the data are much more accurate than the z values, in such circumstances the traditional algebraic fitting method is preferred.

Example 3 Simulation of the ODF Algorithm for Parametric Surfaces

This numerical simulation is the robust orthogonal distance fitting of a parametric surface. The Matlab built-in function *peaks* mentioned in Section 4.4 is adopted again as a template surface,

$$z = 4\,peaks(\frac{x}{20},\frac{y}{20}),\ -20\ mm \le x < 20\ mm,\ 0 \le y < 40\ mm, \tag{6.48}$$

It is represented using a bi-cubic NURBS surface with 18×18 control points. 60×60 points are sampled on the template surface with spacing h=0.3mm as measurement data. They are transformed with $\theta_x = -2°, \theta_y = 2.5°, \theta_z = 1.5°$ and $\mathbf{t}$=[1, -0.8, 1.5]$^{\mathrm{T}}$ mm as the initial position, i.e. to indicate the misalignment between the two coordinate systems, as shown in Figure 6.15.

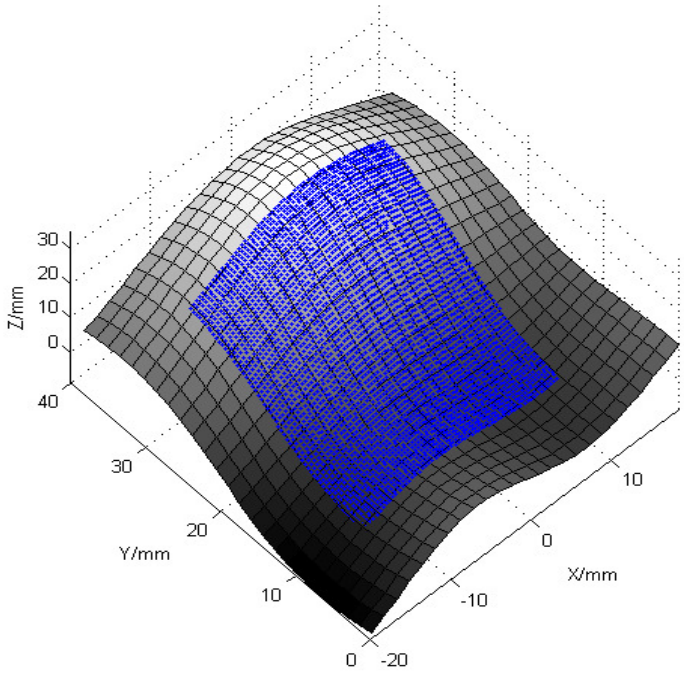


Figure 6.15 Template and data

Gaussian noise of $N(0,(0.6\mu m)^2)$ is introduced into the z coordinates as measurement errors. To simulate measurement outliers, 200 points are randomly sampled and Gaussian error of $N(0,(6\mu m)^2)$ is added onto these points. Defects in the order of millimetre are also involved as illustrated in Figure 6.16. The errors in the x and y coordinates are supposed to be $N(0,(0.9\mu m)^2)$.The Monte-Carlo simulation is employed and the fitting procedure is run 15 iterations 300 times.

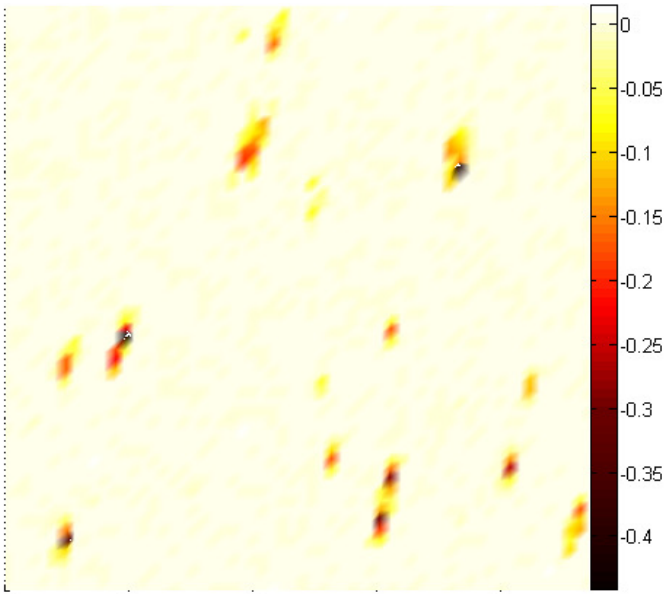


Figure 6.16 Defects and noise

In Section 6.2.3, the dependency between the foot-point parameters and motion parameters is derived from the closest-point constraints. If we ignore it, the motion parameters are much easier to calculate. This problem thereby becomes matching two fixed point sets in each iteration, so that the singular value decomposition technique for ICP discussed in Subsection 6.1.2 will be applied.

Three algorithms are compared here: the robust Orthogonal Distance Fitting, the robust Singular Value Decomposition and the l_2 norm Orthogonal Distance Fitting. The corresponding fitting bias and uncertainty in the rotation angles and translation components are listed in Table 6.5. It is obvious that the SVD method obtains the worst result. At each iteration, it endeavours to minimize the distances between the corresponding point pairs. However, the projection point is already the closest one on the template associated with each measurement point. Thus this algorithm will be trapped at a local minimum and lead to an incorrect result. Therefore it is not proper to directly neglect the dependency between the projection points and the transformation parameters. The ordinary least squares technique is also biased, especially for the rotation angles. Adopting the robust estimator, the influence of the defects can be greatly reduced and the fitting accuracy of the motion parameters may be two orders higher. It can be seen that the uncertainty is roughly in the same order for the three algorithms, since it is mainly determined by the amplitude of the introduced random noise. This simulation clearly validates the high accuracy and reliability of the proposed robust ODF method.

Method		Robust ODF	Robust Norm SVD	l_2 Norm ODF
Bias	θ_x	3.068×10^{-5} °	0.2562°	-2.5834×10^{-3}°
	θ_y	-2.802×10^{-6} °	0.5211°	-5.6011×10^{-3} °
	θ_z	5.259×10^{-5} °	2.2824°	-4.2877×10^{-3} °
	t_x	-0.04909μm	0.8329mm	-1.8408μm
	t_y	-0.01835μm	-0.2093mm	-2.1993μm
	t_z	0.10678μm	0.5293mm	2.3557μm
Uncertainty (σ)	θ_x	1.371×10^{-4} °	6.1952×10^{-3} °	2.496×10^{-4} °
	θ_y	2.687×10^{-4} °	8.0340×10^{-3} °	4.921×10^{-4} °
	θ_z	3.400×10^{-4} °	9.2670×10^{-3}°	6.625×10^{-4} °
	t_x	0.0957μm	1.4528μm	0.1843μm
	t_y	0.0617μm	0.7378μm	0.0982μm
	t_z	0.0559μm	1.6689μm	0.0777μm
Running time		47.2057s	42.9947s	45.7937s

Table 6.5 Comparison of three fitting methods

6.5 Summary

After providing a rough guess for the relative position between the data and template, final fitting is implemented to optimize the solution.

The Iterative Closest Point (ICP) technique is widely adopted for the purpose of registration. It applies for different formats of data and has no special restrictions on the surface shape. The *k*-D tree technique can be utilized to reduce the computational cost of closest-point searching and the singular value decomposition method is applied to update the motion parameters.

However, ICP has a very slow convergence rate, and usually needs more than 15 iterations to make the solution achieve a good result. When the surface is relatively planar, it does not work well and a lateral translation error may exist in the final result. The matching accuracy is influenced by the densities of the template and data points, as well as their distribution modes. It is recommended to sample the template points in a different distribution scheme to the measurement data.

Due to its high computational cost and poor accuracy, ICP is not preferred for final fitting in precision metrology. The Levenberg-Marquardt (L-M) algorithm can be adopted. If the template is provided as a discrete point set, appropriate reconstruction techniques like NURBS or RBF will be employed to obtain a continuous representation for the nominal surface.

The L-M algorithm combines the advantages of the Gauss-Newton and the steepest gradient descent algorithms. If setting the damping factor properly, the design matrix can be guaranteed to be positive definite. Then the solution will always increment towards a local minimum with a super-linear convergence rate. Usually only several iterations are sufficient to get a very accurate result for a smooth free-form surface.

When the explanatory coordinates of the measurement data also contain errors, the fitted result will be biased if only considering the z deviations in the error metric, especially at steep areas of a surface. Hence the orthogonal distance fitting (ODF) algorithm is utilized in this circumstance. The motion parameters (sometimes shape parameters are involved as well) will be updated simultaneously with the correspondence points, so that the computational complexity is in the same order with the algebraic fitting.

The previous algorithm needs an explicit function for the template surface, which is not always available. If the representation is in a parametric format, the foot-point parameters of the template correspondences will be updated in the inner iterate, and the transformation is calculated at the outer iterate. This nested procedure is performed alternately so that a very accurate solution can be achieved. The dependency between the foot-point parameters and the motion parameters is derived from the closest-point constraint between each correspondence pair.

The error metric of least squares is widely applied for its ease of implementation and unbiasedness for the normally distributed errors. However, it is not robust against outliers and missing data. The l_1 norm behaves better under such conditions. But it is not differentiable at zero, so that the l_1 norm problem cannot be solved using conventional derivative-based algorithms such Gauss-Newton or the Levenberg-Marquardt algorithm. Here it is transferred into a reweighted least squares problem based on the majorize-minimize theory. This technique behaves well and is easy to implement in the programme.

It needs to be emphasized that the practical situation should always be analyzed with extreme caution, and the objective function and optimization algorithm be adopted accordingly, instead of blindly attempting to minimize the deviation between the data and the nominal template. If a region of the measured free-form surface has higher manufacturing quality than other areas, larger weights should be assigned onto the data of this area; If some parts of the surface has been worn, known as a *priori* or analyzed from the micro-topography, alignment will be implemented based on the unworn region, then the wear volume of the whole surface can be obtained from the fitted result. When most

of the surface has been worn or the unworn part is not straightforward to be found, weights will be assigned separately for the positive and negative residuals, so that all the fitted residuals are guaranteed to be consistent with the actual situation.

6.6 References

Ahn, S. J, Rauh, W. and Warnecke, H. J. 2001 Least-squares orthogonal distances fitting of circle sphere, ellipse, hyperbola and parabola. *Patt Recog*. 34(12): 2283-2303

Ahn, S. J. 2004 *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*. Springer

Barker, R. M., Cox, M. G., Forbes, A. B. and Harris, P. M. 2004 *Discrete Modelling and Experimental Data Analysis*. Ver 2. NPL Report

Bentley, J. L. 1990 K-D trees for semidynamic point sets. *Proc of the 6th Annual Symposium on Computational Geometry*. 187-197

Besl, P. J. and McKay, N. D. 1992 A method for registration of 3-D shapes. *Trans Patt Anal and Mach Intell*.14(2):239-256

Boggs, P. T., Byrd, R. H. and Schnabel, R. B. 1987 A stable and efficient algorithm for nonlinear orthogonal distance regression. *SIAM J Stat Comput*. 8(6): 1052-1078

Chong, E. K. P. and Żak, S. H. 2001 *An Introduction to Optimization*. Wiley

DIN 32880-1:1986 Coordinate Metrology; Geometrical Fundamental Principles, Terms and Definitions, German Standard, Beuth Verlag, Berlin

Eggert, D. W., Lorusso, A. and Fisher, R. B. 1997 Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine Vis and Appl*.9(5-6): 272-290

Fletcher, R. 2000 *Practical Methods of Optimization*. 2nd Ed, John Wiley & Sons, LTD

Golub, G. H, and van Loan, C. F. 1996 *Matrix Computations*. 3rd Ed. John Hopkins University Press

Greenspan, M. and Yurick, M. 2003 Approximate k-d tree search for efficient ICP. *Proc 4th Int Conf 3DIM*, 442-448

Hansen, P. C. 1998 *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM

Hunter, D. R. and Lange, K. 2000 Quantile regression via an MM Algorithm. *J of Comput and Graph Stat*. 9(1): 60-77

Jost, T. 2002 Fast *Geometric Matching for Shape Registration*. Ph.D Thesis. Université de Neuchâtel. Switzerland

Mandelbrot, B. B. and Van Ness, J. W. 1968 Fractional Brownian motions, fractional noises and applications. *SIAM Review*.10: 422-437

Marquardt, D. W. 1963 An algorithm for least squares estimation of nonlinear parameters. *J of the Soc for Industrial and Appl Math*. 11(2): 431-441

Norton, R. M. 1984 The double exponential distribution: using calculus to find a maximum likelihood estimator. *The American Statistician*.38(2): 135-136

Pottmann, H., Huang, Q. X., Yang, Y. L. and Hu, S. M.2006 Geometry and convergence analysis of algorithms for registration of 3D shapes. *Int J of Computer Vision*. 67(3): 277-296

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. 2002 *Numerical Recipes in C++*. 2nd Ed. Cambridge University Press

Rey, W. J. J. 1983 *Introduction to Robust and Quasi-Robust Statistical Methods*. Springer-Verlag

Sun, W. 2007 *Precision Measurement and Characterisation of Spherical and Aspheric Surfaces*. Ph.D Thesis, University of Southampton, UK

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

7.1 Concluding Remarks

The aim of this dissertation is to investigate and develop appropriate algorithms to fit the measurement data with the design templates and to evaluate the form qualities of free-form surfaces.

In practice the area of the measurement surface is usually smaller than the template, so that the best-matching position of the measurement surface needs to be found on the template surface. Additionally, the coordinate systems of these two surfaces are not exactly identical; hence an optimal transformation (a rotational matrix and a translational vector) is to be determined.

The research work accomplished in this thesis is listed below.

1. Surface Reconstruction

In order to calculate the derivatives and to assess the precise relative deviation between the measurement data and the nominal template, a continuous representation needs to be reconstructed for the design template from discrete points.

(a) We adopt the *Non-Uniform Rational B-Spline* (NURBS) for regular lattice data. To model the NURBS surface as a tensor product, the data points are firstly parameterized into a regular grid. As normally points are sampled uniformly in the domain of interest, it is appropriate to reconstruct the free-form template using a uniform bi-cubic B-spline surface. The explicit representations of the basis functions and their first and second order derivatives are all worked out.

Due to the parametric form of NURBS surfaces, point inversion and projection are required when doing interpolation and fitting. It is very difficult to determine the correct parameter spans associated with the projection point. We insert multiple knots simultaneously using a fast algorithm to decompose a whole NURBS surface into cubic Bézier patches, so that the convex-hull property can be applied. A ‘jumping’ approach is proposed to find the correct incremental direction of the foot-point parameters if the current parameter-span is not a correct one.

(b) The *Radial Basis Function* (RBF) technique is explored to reconstruct a continuous surface patch from scattered data points. It does not require the data points to distribute regularly and applies to multi-dimensional approximation problems.

To reduce the size of the observation matrix and overcome the over-fitting problem of RBF, a sparser set of centres is selected for a given point cloud. If the surface is relatively smooth and the input points are uniformly distributed, centres can also be uniformly located in the domain of interest; otherwise, the *orthogonal least squares basis hunting* technique is utilized to sample centres subsequently from a candidate point set.

The reconstruction quality near the surface boundary is usually much worse compared with the inner region. In order to solve this problem, a new circle of auxiliary centre points are proposed to be added outside the region of interest. The *Graham scan* technique is modified to find the boundary points for a point cloud and these boundary points are extended outward to form a circle of new centres.

When the number of data points exceeds several thousand, the observation matrix tends to be ill-conditioned. Thereby the *Rank-Revealing QR Decomposition* is utilized to solve the weighting vector for a medium or large sized problem, whilst the *Truncated Singular Value Decomposition method* is adopted for a small-sized problem.

2. Initial Matching

Initial matching is implemented first to supply a rough guess for the relative position between the measured data and the design template if an approximate position is not provided.

(a) When the surface is structured and each section is of a simple geometry, a discrete-curvature-based *segmentation* technique is introduced to divide the measurement data into smooth patches.

Each section is fitted by a general quadric function using the linear least squares. A shape-recognition approach is developed to obtain the shape parameters of a general quadric surface and the function is then transformed into a standard form. Subsequently a specific quadric function is fitted through these data to work out the accurate intrinsic characteristics of different sections and the correspondence relationship between the data and template patches.

(b) A rough matching algorithm called *Structured Region Signature* is proposed. One single signature is generated for the measurement surface and many candidate signatures are calculated at sampled locations on the template. The plausible location which possesses the most similar signature with the data is regarded as the best matching position. For the sake of simplicity and efficiency, the signature profiles are sampled uniformly so that the difference between signatures can be calculated by summation, instead of integration. The accuracy of rotation is determined by the sampling density of points on each signature profile, whereas the accuracy of translation is restricted by the sampling spacing of the candidate signature centres on the template. For nearly symmetric surfaces, the residual checking strategy is adopted to avoid false matching.

This method has some remarkable benefits,

- It represents the surface shape with a one-dimensional signature profile, thus is very efficient and straightforward to implement.
- The signature is calculated from the intersection curve between the surface and its inscribed sphere, and is invariant under rotation and translation.
- The signature is a global feature of a surface, and not sensitive to measurement noise and local surface variation.
- This technique is very versatile. It has no particular restriction regarding the surface shape and format. A continuous representation is not required for the surface.
- It is flexible in practice. According to the specific surface shape, multi-circle features can be employed so that more information is involved in one signature.

3. Final Fitting

The purpose of final fitting is to improve the accuracy and reliability of the matching result.

(a) When the template's shape is very complicated or it is difficult to be reconstructed, the *Iterative Closest Point* (ICP) algorithm is adopted.

In order to save time spent on searching for closest points, a *k*-D tree is constructed for the design template. The motion parameters are updated iteratively using the *Singular Value Decomposition* technique.

(b) Due to the slow convergence rate and high computational expense of ICP, a derivative based approach is carried out. The discrete template is reconstructed into a

continuous representation as a nominal surface and then the *Levenberg-Marquardt algorithm* is applied to calculate the optimal motion. By adjusting the damping factor, this method switches between the Gauss-Newton and the steepest gradient descent methods. Its design matrix can be guaranteed to be positive definite, i.e. the solution always converges toward a local minimum.

Compared with ICP, this method converges much faster. The solution can be particularly good through only several iterations.

(c) If the shape of a free-form surface is highly curved, and its intrinsic characteristics are of our particular concern, the *orthogonal distance fitting* is accomplished to overcome the bias problem of the traditional algebraic fitting. An efficient algorithm is adopted to update the correspondence points simultaneously with the motion parameters (and intrinsic characteristics if necessary). The computational complexity of this method is in the same order with the algebraic fitting.

(d) When the function of the design template is supplied in a parametric form and moreover, the coordinates are nonlinear with respect to the foot-point parameters, it will be unacceptably tedious to solve the projection points simultaneously with the motion parameters. Hence the solution can be updated alternately in a *nested approach*. Firstly a closest point is found on the template for each measurement point, and then the motion parameters are incremented. This procedure is repeated until the solution converges.

With the measurement points moving, the projection points will be changed at the same time. That is to say, the foot-point parameters of the projection points are related with the motion parameters. The dependency relationship is derived from the closest-point constraint.

(e) The *least squares method* is extensively applied for its ease of implementation. The solution is unbiased when the error obeys the Gaussian distribution. However, the solution of least squares is not robust enough, and outliers may make the solution distorted, or even break down.

The l_1 norm (least absolute deviation) pays less attention to large errors and thus is much more stable. But it has discontinuous derivatives and is very difficult to solve. Here it is transferred into a reweighted least squares problem based on the *majorize-minimum theory*.

The fitting strategy of free-form surfaces is summarised here. Appropriate methods need to be adopted according to the shapes and applications of the free-form components.

(a) **Structured Surfaces** If a free-form surface is structured and composed of small sections, the entire surface can be firstly segmented into continuous sections and each section is individually fitted with a simple quadric function. Thus alignment can be established by comparing the intrinsic characteristics (shape parameters) and overlapping the correspondence features (centre points, rotational-symmetric axes etc).

(b) **Non-Smooth Surfaces** Non-smooth surfaces are very difficult to represent using analytical functions. Hence the design template is sampled with discrete points. The iterative closest point technique can be utilized to match the nominal points with measurement data. It is worth noting that the distribution modes and sampling densities of the two point sets should not be the same, otherwise a false local-minimum matching result will be caused.

(c) **Smooth Surfaces** Here a continuous representation is required for the reference template. NURBS and RBFs are adopted for regular and scattered point sets respectively if the design template is supplied as a CAD model. In order to avoid false fitting results, the whole fitting procedure is divided into two stages: initial matching and final fitting. Firstly the structured region signature technique is applied to find a rough guess for the relative position between the template and measurement data. Then the solution is refined with the Levenberg-Marquardt algorithm. If the fitted shape parameters (intrinsic characteristics) and motion parameters are of special importance, the orthogonal distance fitting can be applied to reduce the bias in the solution. Additionally, an appropriate robust estimator can be used to overcome the outliers and missing data.

7.2 Future Work

In this dissertation effective fitting algorithms have been proposed and proper techniques have been developed to evaluate the form qualities of free-form surfaces. However, there still exist some problems to be solved. Here we point out some directions for future research.

1. The reconstruction accuracy of RBF depends heavily on the distribution of the input data points. Large oscillations may arise between the data because of over-fitting. Practical and reliable regularization techniques will be developed for surface

reconstruction of RBF, so that the resultant surface is guaranteed to be smooth and close to the ideal surface.

2. For structured surfaces, the segmentation technique cannot properly divide the boundary points between sections. New techniques are needed to recognize the boundary points more rigidly. Wavelet or morphological algorithms can be adopted to divide structured surfaces.

3. When the measurement surface is relatively planar, and the template and data points are not properly distributed, the ICP algorithm will converge very slowly and obtain a poor result. Hence special techniques shall be developed to speed up its convergence rate and to overcome its local-minimum problem.

4. A free-form surface may be represented with an implicit function, rather than explicit or parametric forms, although this is not common in practice. Some special fitting algorithms will be developed to fit this particular type of surfaces. These methods are required to be sufficiently efficient and accurate.

5. The quality of the fitted result is closely related to the error distribution. Hence the relationship between the fitting accuracy and the distribution of measurement noise will be investigated carefully. Proper error metrics will be adopted for different error models, and appropriate solution techniques will be applied accordingly.

PUBLICATION LIST

1. Xiangchao Zhang, Xiangqian Jiang, and Paul J Scott. Orthogonal distance fitting of precision free-form surfaces based on the l_1 norm. *8th Conference on Advanced Mathematical and Computational Tools in Metrology and Testing*. Paris, 2008
2. Xiangchao Zhang and Xiangqian Jiang. Numerical analyses of the boundary effect of radial basis functions in 3D surface reconstruction. *Numerical Algorithms*. 2008; 47(4): 327-339
3. X Zhang, X Jiang, and P J Scott. A new free-form surface fitting method for precision coordinate metrology. *Wear*. 2009; 266(5-6): 543-547