# Algorithms for Three-Dimensional Free-Form Object Matching

by

Kwang Hee Ko

B.S. in Naval Architecture and Ocean Engineering (1995)
Seoul National University, Republic of Korea
M.S. in Naval Architecture and Marine Engineering (2001)
M.S. in Mechanical Engineering (2001)
Massachusetts Institute of Technology

Submitted to the Department of Ocean Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2003

Author . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Department of Ocean Engineering
March 13, 2003

Certified by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Nicholas M. Patrikalakis, Kawasaki Professor of Engineering
Thesis Co-Supervisor

Certified by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Takashi Maekawa, Principal Research Scientist
Thesis Co-Supervisor

Accepted by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Michael S. Triantafyllou, Professor of Ocean Engineering
Chairman, Departmental Committee on Graduate Students
Department of Ocean Engineering

# Algorithms for Three-Dimensional Free-Form Object Matching

by

Kwang Hee Ko

Submitted to the Department of Ocean Engineering
on March 13, 2003, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

## Abstract

This thesis addresses problems of free-form object matching for the point vs. NURBS surface and the NURBS surface vs. NURBS surface cases, and its application to copyright protection. Two new methods are developed to solve a global and partial matching problem with no a priori information on correspondence or initial transformation and no scaling effects, namely the KH and the umbilic method. The KH method establishes a correspondence between two objects by utilizing the Gaussian and mean curvatures. The umbilic method uses the qualitative properties of umbilical points to find correspondence information between two objects. These two methods are extended to deal with uniform scaling effects. The umbilic method is enhanced with an algorithm for scaling factor estimation using the quantitative properties of umbilical points. The KH method is used as a building block of an optimization scheme based on the golden section search which recovers iteratively an optimum scaling factor. Since the golden section search only requires an initial interval for the scaling factor, the solution process is simplified compared to iterative optimization algorithms, which require good initial estimates of the scaling factor and the rigid body transformation. The matching algorithms are applied to problems of copyright protection. A suspect model is aligned to an original model through matching methods so that similarity between two geometric models can be assessed to determine if the suspect model contains part(s) of the original model. Three types of tests, the weak, intermediate and strong tests, are proposed for similarity assessment between two objects. The weak and intermediate tests are performed at node points obtained through shape intrinsic wireframing. The strong test relies on isolated umbilical points which can be used as fingerprints of an object for supporting an ownership claim to the original model. The three tests are organized in two decision algorithms so that they produce systematic and statistical measures for a similarity decision between two objects in a hierarchical manner. Based on the systematic statistical evaluation of similarity, a decision can be reached whether the suspect model is a copy of the original model.

Thesis Co-Supervisor: Nicholas M. Patrikalakis, Kawasaki Professor of Engineering

Thesis Co-Supervisor: Takashi Maekawa, Principal Research Scientist

# Acknowledgments

First of all, I want to thank my wife, Suyeon, for her love and emotional support, and my family for their love and understanding during my study at MIT.

I would like to thank my thesis supervisors, Professor Nicholas M. Patrikalakis and Dr. Takashi Maekawa, for their expert advice on my research work and instructive guidance on my academic studies, and Professors D. C. Gossard, H. Masuda, S. Sarma and F.-E. Wolter for their helpful advice as members of my doctoral thesis committee.

Thanks also go to Professor Takis Sakkalis for his comments, Dr. Constantinos Evangelinos for helpful discussions and efforts for stable hardware environment for my thesis work, Mr. Fred Baker for efforts in the laboratory management, Design Laboratory fellows Dr. Wonjoon Cho, Ms. Hongye Liu, Mr. Da Guo and Mr. Harish Mukundan for making a good laboratory environment, and Dr. Yonghwan Kim, Mr. Jaehyeok Auh, Mr. Sungjoon Kim, Mr. Youngwoong Lee for their help and good advice.

Funding for this research was obtained from the National Science Foundation (NSF), under grant number DMI-0010127.

# Contents

# List of Figures

# List of Tables

# List of Symbols

$\mathbf{r}$ : surface
$\mathbf{p}$ : point
$u, v, t$ : parameters
$s$ : arc length
$I$ : first fundamental form
$II$ : second fundamental form
$E, F, G$ : first fundamental form coefficients
$L, M, N$ : second fundamental form coefficients
$\mathbf{t}$ : tangent vector
$N$ : unit normal vector
$\kappa$ : normal curvature
$K$ : Gaussian curvature
$H$ : mean curvature
$\kappa_g$ : geodesic curvature
$\Gamma^i_{jk}$ : Christoffel symbols
$\kappa_{1,2}$ : maximum and minimum principal curvatures
$\Gamma_1, \;\; \Gamma_2$ : characteristic lines
$\omega$ : complex number for an umbilical point
$V(x, y)$ : cubic form
$H_e(x, y)$ : Hessian
$w$ : positive weight in a NURBS representation
$B_{i,j}$ : Bernstein basis function
$N_{i,j}$ : B-spline basis function
$\delta$ : tolerance
$\mathbf{t}_T$ : translation vector
$\mathbf{R}$ : rotational matrix
$\hat{\mathbf{q}}$ : unit quaternion
$d_e$ : distance between two points
$d_{st}$ : minimum distance between a point and a surface
$\Phi$ : global distance function
$\sigma$ : scaling factor
$I_{ij}$ : moment of inertia

# Chapter 1

# Introduction

## 1.1 Background and Motivation

Rapid advance of computer technology has revolutionized design and manufacture of products in various fields. Almost all product data are created and stored in digital form using computer systems, and directly provided as input to computer aided manufacturing systems to produce physical products. As these data models are expensive and a significant part of the production process, there is a growing need to protect the ownership of these data models against unauthorized use by malicious parties [61]. Moreover, the ubiquitous nature of the Internet along with the World Wide Web and related technologies make it possible to rapidly exchange information electronically all over the world with no extra cost, which enables design and production to be performed at remote design and production sites. Easy data exchange, on the other hand, poses serious concerns to the owner of valuable data since such important data may be duplicated by unauthorized parties without losing any details when they are exposed to the Internet. Therefore, copyright protection for digital product models has become a major issue, and protecting intellectual property of digital information has emerged as an important research topic.

In the design and manufacturing fields, product models are typically represented in *Non-Uniform Rational B-Spline (NURBS)* form which is a standard format in industry [93, 89]. In the past, 3D model descriptions had been represented with a fairly restrictive shape variety, for example, 2D drawings (blueprints). Today they are typically described with CAD systems using digital data. Here the richest shape variety can be modeled by free-form surfaces that are typically defined as NURBS. Hence, the most important and fundamental part of the value creation process for a 3D model consists in creating the digital 3D free-form model.

Two types of feasible protection methods for 3D free-form objects can be considered: one is to embed *watermark* information in the object and check the watermark for illegal duplication. The other method is to align two objects as accurately as possible and check them for similarity. Several methods have been reported on digital watermarking for 3D models. Most of the methods are designed for models represented via a triangular mesh or via range data. These techniques, however, are not appropriate for 3D CAD data models usually represented in NURBS form. There has been an attempt to embed user-defined information to the NURBS representation by Ohbuchi *et al.* [86]. But the embedded data can be easily destroyed by reparametrization or reapproximation. Since embedding robust user-defined watermarks in the NURBS representation is difficult, the similarity checking method can be adopted for protection of the ownership for digital objects represented in NURBS form.

Matching is a key step in the similarity checking method. The purpose of matching is to minimize the geometric discrepancy caused by translation, rotation and scaling. Three dimensional object matching has been one of the most important topics in computer vision, computer graphics and inspection, and there have been many significant contributions in developing matching methods for various representation forms such as NURBS surface patches, polyhedral surfaces and range data. Campbell and Flynn [20] regarded *free-form* as "a general characterization of an object whose surfaces are not of a more easily recognized class such as planar and/or natural quadric surfaces." Another interpretation was given by Besl [9]: "a free-form surface has a well defined surface normal that is continuous almost everywhere except at vertices, edges, and cusps." Many surfaces such as ship hulls, automobile bodies, aircraft fairing surfaces and organs are typical examples of free-form surfaces, which can be represented in various forms, such as NURBS surface patches, polyhedral surfaces and range data. Matching is used in various applications. The manufacturing process mostly uses matching techniques for automatic inspection, and in computer vision, matching is used for scene integration and object recognition. When matching is used in the context of computer aided inspection, it is referred to as *localization* [90], whereas when it is used in the context of computer vision it is referred to as *registration* [10].

Many methods have been proposed for free-form object matching. In computer aided design, matching through minimization of a squared distance metric objective function is widely used since it is conceptually easy and shows good performance. But, such an approach cannot be used for a case that no *a priori* information for correspondence or initial transformation is provided, which commonly happens in practice. Correspondence information can be provided by the user and then an iterative search method can be employed to find the best transformation. In this case, however, the matching process is not automated. Partial matching and uniform scaling effects need to be considered in the context of the matching problem as well. Here, only uniform scaling is discussed since non-uniform scaling generally destroys the functionality of an object. A global method such as the moment method cannot handle partial object matching and most of matching methods fail to recover the scaling factor. A problem containing partial matching and scaling effects with no a priori information on correspondence is the most general form of the matching problem, which has not been studied so far.

## 1.2 Research Objectives

A primary objective of this thesis is to develop a global and partial matching method with scaling effects to handle the point vs. NURBS surface and the NURBS surface vs. NURBS surface cases when no a priori information on correspondence or initial transformation is provided. Two approaches are considered in this thesis: one involves use of isolated generic umbilical points and the other use of the Gaussian and mean curvatures and a 1D optimization scheme. Using either one of them, two objects are aligned as closely as possible so that the differences between two objects caused by the rigid body transformation including scaling are minimized.

Efficient construction of shape intrinsic wireframe is another topic of this thesis. A shape intrinsic wireframe is a representation method of a surface using shape intrinsic properties, such as lines of curvature and geodesic curves which are independent of parametrization as well as the rigid body transformation. Robust calculation of the intrinsic properties is an important issue that needs to be considered in this thesis. Extraction of umbilical points deserves more attention because complete information on them is critical in construction of shape intrinsic wireframe, matching using umbilical points and decision algorithms. An efficient and robust algorithm is developed to extract umbilical points from NURBS surfaces, which can effectively find not only isolated umbilics but also non-isolated umbilics forming curves or regions such as planar or spherical.

The assessment of matching is a topic that is also discussed in the thesis. Three hierarchical tests, the $\epsilon$-offset, principal curvature and direction test, and the umbilical point test, are proposed, and a quantitative evaluation method of similarity is developed along with two similarity decision algorithms. The algorithms consist of the three hierarchical tests, and produce systematic and statistical measures that can be used for a similarity decision.

Application of the proposed algorithms to copyright protection is demonstrated with examples. The decision algorithms are primarily used to determine if a suspect model is a copy of an original model. After matching the two models, a systematic and statistical assessment of the similarity between two models is performed, from which a decision can be made if one object is a copy of the other in a hierarchical manner.

## 1.3 Thesis Organization

The thesis is structured as follows:

In Chapter 2, differential geometry of surfaces, and the NURBS representation for curves and surfaces and their properties are reviewed. This is followed by mathematical treatment of lines of curvature, geodesics and umbilical points. The classification of umbilical points is presented in detail.

In Chapter 3, computation methods and algorithms for lines of curvature, geodesic curves, orthogonal projection and extraction of umbilical points are presented. Using all proposed calculation methods, a quasi-automatic system to create surface intrinsic wireframes is introduced.

Chapter 4 is devoted to matching algorithms. Two methods to establish a correspondence between two objects are proposed, and expanded to deal with partial matching problems with scaling when no a priori information on correspondence is given. One method involves use of umbilical points and the other use of an optimization scheme. Accuracy, complexity and actual performance analyses of the proposed algorithms are presented.

Two similarity decision algorithms are presented in Chapter 5, which are used for copyright protection. Both algorithms are based on the hierarchical tests proposed in Chapter 4. One algorithm uses the maximum values for a decision and the other provides statistical data for a decision.

Chapter 6 presents examples of the proposed matching algorithms and applications to protection of intellectual property.

Chapter 7 concludes the thesis with recommendations for future work.

Finally, detailed mathematical treatment of the classification of umbilical points is presented in Appendix A followed by brief presentation of the formulation of Gaussian and mean curvature functions for Bézier surface patches in Appendix B.

# Chapter 2

# Theoretical Background

In this chapter, mathematical definitions and concepts which will be used throughout this thesis are presented. Relevant aspects of differential geometry of surfaces are summarized [113, 32], and properties and classification of umbilical points are discussed in detail [95, 96, 17, 8, 79, 74, 91]. This chapter concludes with a brief discussion of definitions and properties of NURBS curves and surfaces [93, 50, 91].

## 2.1 Review of Differential Geometry

### 2.1.1 Basic Theory of Surfaces

A parametric surface can be defined as a subset of 3D space, $\mathbf{R}^3$, obtained by mapping a 2D parametric *uv* domain to $\mathbf{R}^3$

$$\mathbf{r}(u,v) = [x(u,v), y(u,v), z(u,v)]^T, \tag{2.1}$$

where usually $(u,v) \in [0,1] \times [0,1]$. A surface is *regular* if $\frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} \neq \mathbf{0}$. The regularity condition implies that a unique unit normal vector $\mathbf{N}$ is defined at every point on the surface and $\mathbf{N} = \frac{\mathbf{r}_u \times \mathbf{r}_v}{|\mathbf{r}_u \times \mathbf{r}_v|}$. A curve on a surface $\mathbf{r}$ can be represented in the form $\mathbf{r}(u(t), v(t))$ where $t$ is a parameter, usually in a range $0 \leq t \leq 1$. The *first fundamental form*, $I$, which is a distance measure on a surface, is defined as follows:

$$I = d\mathbf{r} \cdot d\mathbf{r} = Edu^2 + 2Fdudv + Gdv^2, \tag{2.2}$$

where $d\mathbf{r}$ is infinitesimal displacement of a curve on a surface $\mathbf{r}(u(t), v(t))$, and $E = \frac{\partial \mathbf{r}}{\partial u} \cdot \frac{\partial \mathbf{r}}{\partial u}$, $F = \frac{\partial \mathbf{r}}{\partial u} \cdot \frac{\partial \mathbf{r}}{\partial v}$ and $G = \frac{\partial \mathbf{r}}{\partial v} \cdot \frac{\partial \mathbf{r}}{\partial v}$ are the first fundamental form coefficients.

The *second fundamental form*, $II$, a measure of the curvature of a surface, is defined as follows:

$$II = -d\mathbf{r} \cdot d\mathbf{N} = Ldu^2 + 2Mdudv + Ndv^2, \tag{2.3}$$

where $\mathbf{N}$ is the unit normal vector of a surface, and $L = \mathbf{N} \cdot \frac{\partial^2 \mathbf{r}}{\partial u^2}$, $M = \mathbf{N} \cdot \frac{\partial^2 \mathbf{r}}{\partial u \partial v}$ and $N = \mathbf{N} \cdot \frac{\partial^2 \mathbf{r}}{\partial v^2}$ are the second fundamental form coefficients. The negative sign ensures that if the normal curvature $\kappa$ is positive, the center of curvature lies opposite to the direction of the surface normal. Here, $\mathbf{N} \cdot \frac{\partial \mathbf{r}}{\partial u} = \mathbf{N} \cdot \frac{\partial \mathbf{r}}{\partial v} = 0$ are used for the derivation.

The unit tangent vector $\mathbf{t}$ of a curve on a surface $\mathbf{r}(u(t), v(t))$ at $\mathbf{p}(\in \mathbf{r})$ is obtained by differentiating $\mathbf{r}$ with respect to the arc length $s$, i.e. $\mathbf{t} = \frac{d\mathbf{r}}{ds}$. Then, the curvature vector $\boldsymbol{\kappa}$ of the curve $\mathbf{r}(u(t), v(t))$ at $\mathbf{p}$ can be calculated as the second derivative with respect to $s$, i.e.

$$\boldsymbol{\kappa} = \frac{d\mathbf{t}}{ds} = \frac{d^2\mathbf{r}}{ds^2}. \tag{2.4}$$

The normal component of $\boldsymbol{\kappa}$, or $\kappa = -\boldsymbol{\kappa} \cdot \mathbf{N}$ is the normal curvature of a surface $\mathbf{r}$ in the direction of $\mathbf{t}$ at $\mathbf{p}$. Equation (2.4) can be rewritten as [113]

$$\kappa = -\boldsymbol{\kappa} \cdot \mathbf{N} = -\frac{d\mathbf{t}}{ds} \cdot \mathbf{N} = \frac{d\mathbf{r}}{ds} \cdot \frac{d\mathbf{N}}{ds} = -\frac{II}{I} = -\frac{L + 2M\lambda + N\lambda^2}{E + 2F\lambda + G\lambda^2}, \tag{2.5}$$

where $\lambda = \frac{dv}{du}$. The extrema of $\kappa$ are obtained from $\frac{d\kappa}{d\lambda} = 0$. This yields

$$(E + 2F\lambda + G\lambda^2)(N\lambda + M) - (L + 2M\lambda + N\lambda^2)(G\lambda + F) = 0, \tag{2.6}$$

which can be rewritten as:

$$(E + F\lambda)(M + N\lambda) = (L + M\lambda)(F + G\lambda). \tag{2.7}$$

Therefore, equation (2.5) becomes

$$\kappa = -\frac{L + 2M\lambda + N\lambda^2}{E + 2F\lambda + G\lambda^2} = -\frac{M + N\lambda}{F + G\lambda} = -\frac{L + M\lambda}{E + F\lambda}, \tag{2.8}$$

from which we can set up two simultaneous equations to find the maximum and minimum principal curvatures and their directions:

$$\begin{aligned} (L + \kappa E)du + (M + \kappa F)dv &= 0, \\ (M + \kappa F)du + (N + \kappa G)dv &= 0. \end{aligned} \tag{2.9}$$

Equations (2.9) have non-trivial solutions if and only if

$$\begin{vmatrix} L + \kappa E & M + \kappa F \\ M + \kappa F & N + \kappa G \end{vmatrix} = 0. \tag{2.10}$$

Two distinct real roots of (2.10), $\kappa_1$ and $\kappa_2$ are the *principal curvatures*, and their corresponding directions $\lambda_1$ and $\lambda_2$ the *principal directions* [113]. The principal directions are orthogonal to each other. A double root $\kappa$ is obtained at an *umbilical point*. The *Gaussian curvature*, $K = \kappa_1\kappa_2$, and the *mean curvature*, $H = \frac{\kappa_1+\kappa_2}{2}$, are obtained from the first and second fundamental form coefficients as follows [113]:

$$\begin{aligned} K &= \frac{LN - M^2}{EG - F^2}, \\ H &= \frac{1}{2}\left(\frac{2FM - EN - GL}{EG - F^2}\right). \end{aligned} \tag{2.11}$$

### 2.1.2 Lines of Curvature

A line of curvature is a curve on a surface whose tangents are principal directions at all points on the curve. At a point on a surface away from umbilical points, two orthogonal principal directions are uniquely determined [113]. Hence, two lines of curvature (maximum and minimum) intersect orthogonally at such a point. Lines of curvature passing through an umbilical point are explained in Section 2.1.4.

Suppose that $u = u(s)$ and $v = v(s)$, where $s$ is the arc length parameter. Then we obtain [113]

$$\frac{du}{ds} = \eta(M + \kappa F), \quad \frac{dv}{ds} = -\eta(L + \kappa E), \tag{2.12}$$

if the first equation of (2.9) is used. Similarly, the second equation of (2.9) yields

$$\frac{du}{ds} = \mu(N + \kappa G), \quad \frac{dv}{ds} = -\mu(M + \kappa F). \tag{2.13}$$

Parametric values of lines of curvature are calculated by solving the first order differential equations (2.12) or (2.13).

### 2.1.3 Geodesics

Let us define a unit vector $\mathbf{u} = \mathbf{N} \times \mathbf{t}$ at a point $\mathbf{p}$ on a surface, where $\mathbf{t}$ is the unit tangent vector of a curve $\mathbf{c}$ on the surface at $\mathbf{p}$. Then $\mathbf{u}$ is perpendicular to $\mathbf{N}$ and $\mathbf{t}$, and is contained in the tangent plane of the surface at $\mathbf{p}$. The $\mathbf{u}$ component of the curvature vector $\boldsymbol{\kappa}$ of $\mathbf{c}$, which is obtained by

$$\boldsymbol{\kappa}_g = (\boldsymbol{\kappa} \cdot \mathbf{u})\,\mathbf{u}, \tag{2.14}$$

is called the *geodesic curvature vector*, and the magnitude of $\boldsymbol{\kappa_g}$ is the *geodesic curvature* in the direction of $\mathbf{t}$ at $\mathbf{p}(\in \mathbf{r})$ [113]. Using (2.4) and the Christoffel symbols $\Gamma^i_{jk}$, $(i, j, k = 1, 2)$ [113],

$$\begin{aligned} \Gamma^1_{11} &= \frac{GE_u - 2FF_u + FE_v}{2(EG - F^2)}, \quad \Gamma^2_{11} = \frac{2EF_u - EE_v + FE_u}{2(EG - F^2)}, \\ \Gamma^1_{12} &= \frac{GE_v - FG_u}{2(EG - F^2)}, \qquad\quad\;\; \Gamma^2_{12} = \frac{EG_u - FE_v}{2(EG - F^2)}, \\ \Gamma^1_{22} &= \frac{2GF_v - GG_u + FG_v}{2(EG - F^2)}, \quad \Gamma^2_{22} = \frac{EG_v - 2FF_v + FG_u}{2(EG - F^2)}, \end{aligned} \tag{2.15}$$

we can derive the geodesic curvature $\kappa_g$ as follows:

$$\begin{aligned}\kappa_g &= \left[\Gamma^2_{11}\left(\frac{du}{ds}\right)^3 + (2\Gamma^2_{12} - \Gamma^1_{11})\left(\frac{du}{ds}\right)^2\frac{dv}{ds} + (\Gamma^2_{22} - 2\Gamma^1_{12})\frac{du}{ds}\left(\frac{dv}{ds}\right)^2\right.\\ &\left. -\Gamma^1_{22}\left(\frac{dv}{ds}\right)^3 + \frac{du}{ds}\frac{d^2v}{ds^2} - \frac{d^2u}{ds^2}\frac{dv}{ds}\right]\sqrt{EG - F^2}. \qquad (2.16)\end{aligned}$$

The equation of a geodesic curve can be obtained by setting $\kappa_g = 0$ in equation (2.16) according to the definition of the geodesics in [113]. Considering that the surface normal $\mathbf{N}$ has the direction of a normal $\pm\mathbf{n}$ to the geodesic curve, an alternative form of (2.16) can be obtained from equations $\mathbf{n}\cdot\mathbf{r}_u = 0$ and $\mathbf{n}\cdot\mathbf{r}_v = 0$ using the Christoffel symbols $\Gamma^i_{jk}$, $(i, j, k = 1, 2)$ as follows [113]:

$$\frac{d^2u}{ds^2} + \Gamma^1_{11}\left(\frac{du}{ds}\right)^2 + 2\Gamma^1_{12}\frac{du}{ds}\frac{dv}{ds} + \Gamma^1_{22}\left(\frac{dv}{ds}\right)^2 = 0, \qquad (2.17)$$

$$\frac{d^2v}{ds^2} + \Gamma^2_{11}\left(\frac{du}{ds}\right)^2 + 2\Gamma^2_{12}\frac{du}{ds}\frac{dv}{ds} + \Gamma^2_{22}\left(\frac{dv}{ds}\right)^2 = 0. \qquad (2.18)$$

Equations (2.17) and (2.18) can be rewritten as a system of four first order differential equations [113]:

$$\begin{aligned}\frac{du}{ds} &= p, &(2.19)\\ \frac{dv}{ds} &= q, &(2.20)\\ \frac{dp}{ds} &= -\Gamma^1_{11}p^2 - 2\Gamma^1_{12}pq - \Gamma^1_{22}q^2, &(2.21)\\ \frac{dq}{ds} &= -\Gamma^2_{11}p^2 - 2\Gamma^2_{12}pq - \Gamma^2_{22}q^2. &(2.22)\end{aligned}$$

### 2.1.4 Umbilics

An *umbilic* is a point on a surface where the normal curvatures in all directions are equal and the principal directions are indeterminate. The principal curvature functions are represented in terms of the Gaussian ($K$) and the mean ($H$) curvature functions as follows [113]:

$$\kappa_{1,2}(u, v) = H(u, v) \pm \sqrt{H^2(u, v) - K(u, v)}. \qquad (2.23)$$

Let $W(u, v) = H^2 - K$. The principal curvatures, $\kappa_{1,2}$ are real valued functions so that $W \geq 0$ must hold. From the definition of the umbilical point we have $W(u, v) = 0$. With these two conditions combined, we can infer that at an umbilical point, $W(u, v)$ has a global minimum [72, 74]. Here, we assume that $W$ is at least $C^2$ smooth. Then, the condition that $W$ has a global minimum at an umbilic implies that $\nabla W = 0$.

Therefore, at an umbilic the following equations hold [74]:

$$W(u,v) = 0, \quad \frac{\partial W(u,v)}{\partial u} = 0, \quad \frac{\partial W(u,v)}{\partial v} = 0. \tag{2.24}$$

Given a polynomial parametric surface patch such as a rational Bézier surface patch, we can set $W = \frac{P_N}{P_D}$, where $P_N$ and $P_D$ are polynomials in both $u$ and $v$. With the condition $W \geq 0$, $P_N \geq 0$ is assured since $P_D > 0$ is always true under the regularity condition of the surface [113]. The equation $W = 0$ is equivalent to $P_N = 0$. The first derivative of $W$ is $\frac{\partial W}{\partial x_i} = (\frac{\partial P_N}{\partial x_i} P_D - P_N \frac{\partial P_D}{\partial x_i})/P_D^2 (i = 1, 2)$, where $x_1 = u$ and $x_2 = v$, which is reduced to $\frac{\partial W}{\partial x_i} = (\frac{\partial P_N}{\partial x_i})/P_D$ using $P_N = 0$. Therefore, equations (2.24) are reduced to [74]

$$P_N(u,v) = 0, \quad \frac{\partial P_N}{\partial u} = 0, \quad \frac{\partial P_N}{\partial v} = 0. \tag{2.25}$$

### Classification of Umbilical Points

Umbilical points are classified into two types: *generic* and *non-generic*. Generic umbilical points maintain their properties under small perturbations of the surface, while non-generic umbilical points may lose their qualitative properties under small perturbations [8, 104, 74, 91]. They can be isolated or form lines or regions. Isolated generic umbilical points are further classified into three types: *star*, *monstar* and *lemon* as shown in Figure 2-1. Star type umbilical points are further classified into the *hyperbolic star* and the *elliptical star* type umbilical points. The umbilical diagram shown in Figure 2-2 [96] is an easy way to distinguish the type of an isolated generic umbilical point. In order to use this diagram, the local surface near an umbilical point has to be represented as a height function or the Monge form with respect to a local coordinate system as follows [74]:

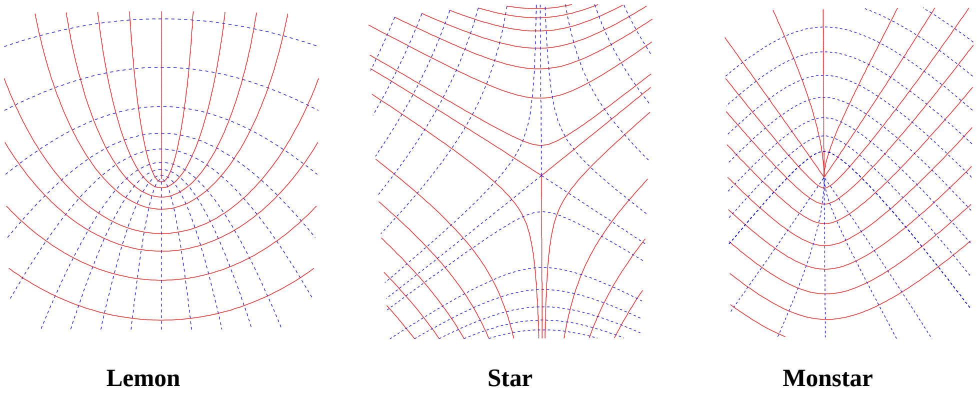


Figure 2-1: Three generic umbilics adapted from [91]

$$\mathbf{r} = (x, y, h(x,y)). \tag{2.26}$$

The height function $h(x, y)$ is Taylor expanded at the origin of the local coordinate system. Then we have

$$\begin{aligned} h(x, y) &= -\frac{\kappa}{2}(x^2 + y^2), \\ &\quad +\frac{1}{6}(ax^3 + 3bx^2y + 3cxy^2 + dy^3) + O(4), \end{aligned} \tag{2.27}$$

where $\kappa$ is the normal curvature at the umbilical point. Let us set $C(x, y) = ax^3 + 3bx^2y + 3cxy^2 + dy^3$. This expression implies that the local structure of a surface near an umbilical point is dominated by the coefficients of $C(x, y)$, i.e. by $a, b, c, d$, which determine the type of umbilical points [79, 96]. It is convenient to represent the cubic part $C(x, y)$ in the complex plane for analysis purposes. If we set $\zeta = x + iy$, then $C(x, y)$ becomes

$$\hat{C}(\zeta) = \alpha\zeta^3 + 3\overline{\beta}\zeta^2\overline{\zeta} + 3\beta\zeta\overline{\zeta}^2 + \overline{\alpha}\overline{\zeta}^3, \tag{2.28}$$

with

$$\begin{aligned} \alpha &= \frac{1}{8}[(a - 3c) + i(d - 3b)], \\ \beta &= \frac{1}{8}[(a + c) + i(b + d)], \end{aligned} \tag{2.29}$$

where $\alpha \neq 0$. We can express (2.28) in a coordinate system rotated about the normal vector without losing any essential features to make the coefficient of $\zeta^3$ equal to 1 [96]. Using $\xi = \alpha^{\frac{1}{3}}\zeta$, equation (2.28) becomes

$$\tilde{C}(\xi) = \xi^3 + 3\overline{\omega}\xi^2\overline{\xi} + 3\omega\xi\overline{\xi}^2 + \overline{\xi}^3, \tag{2.30}$$

where $\omega = \beta\alpha^{-\frac{1}{3}}\overline{\alpha}^{-\frac{2}{3}}$. This means that $C(x, y)$ is parametrized with respect to a single complex variable $\omega$ [17, 96]. Therefore, all variations of $C(x, y)$ can be mapped onto the complex plane [17, 79, 96]. When $\alpha = 0$, equation (2.28) is reduced to

$$\hat{C}(\zeta) = 3\zeta\overline{\zeta}(\overline{\beta}\zeta + \beta\overline{\zeta}). \tag{2.31}$$

This equation corresponds to infinity in the $\omega$ complex plane [96, 17, 79], which is not considered in this discussion.

Depending on the structure of $C(x, y)$ (or $\tilde{C}(\xi)$), three characteristic lines are determined as follows [96, 79]:

- $\Gamma_1 : \theta \rightarrow \frac{1}{3}(2e^{i\theta} + e^{-2i\theta})$,
- $|\omega| = 1$,
- $\Gamma_2 : \theta \rightarrow (2e^{i\theta} + e^{-2i\theta})$,

where $\Gamma_1$ and $\Gamma_2$ are maps from $\theta$ to the $\omega$ complex plane. They divide the $\omega$ complex plane into sub-regions as shown in Figure 2-2. Each sub-region corresponds to a specific type of an umbilical point. In Figure 2-2, *ES* means the *elliptic star*, *HS* the *hyperbolic star*, $MS$ the *monstar* and $L$ the *lemon*. If $\omega$ falls on a dividing curve, then the corresponding umbilical point is of non-generic type. The behavior of such an umbilical point can be analyzed with more higher order terms [74]. Using this diagram, the type of an umbilical point is easily determined, see [96, 17, 79]. A detailed discussion on the characteristic curves is presented in Appendix A.

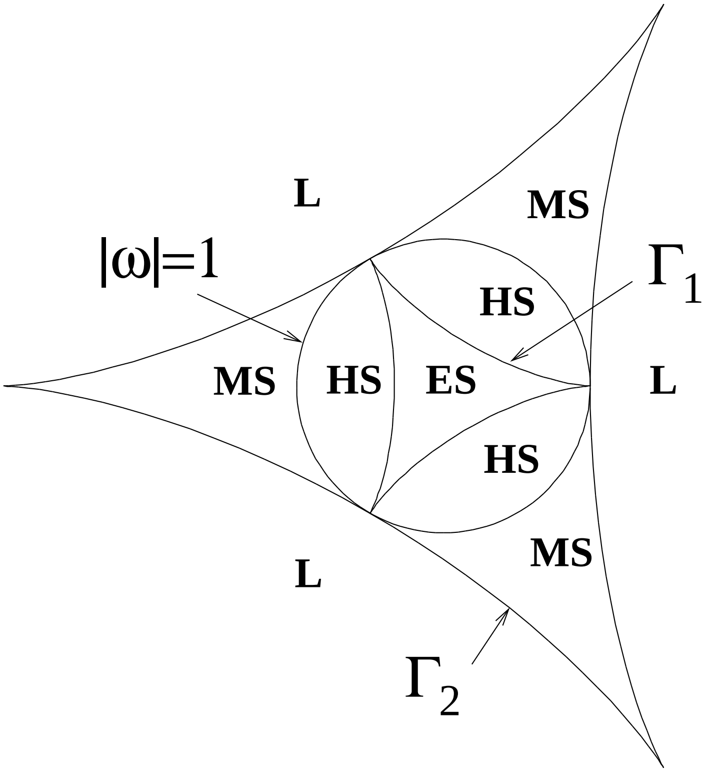


Figure 2-2: The umbilic diagram adapted from [96]

## 2.2 Review of NURBS Curves and Surfaces

A NURBS (Non-Uniform Rational B-Spline) representation is the most general form which includes integral B-spline and Bézier representations as special cases.

A NURBS curve $\mathbf{q}$ of order $k$ is defined as follows:

$$\mathbf{q}(u) = \frac{\sum_{i=0}^{m} w_i \mathbf{Q}_i N_{i,k}(u)}{\sum_{i=0}^{m} w_i N_{i,k}(u)}, \quad 0 \le u \le 1, \tag{2.32}$$

where $\mathbf{Q}_i$ are the control points, $m+1$ the number of control points, $w_i$ the positive weights, and $N_{i,k}(u)$ the B-spline basis functions defined by

$$\begin{aligned} N_{i,1}(u) &= \begin{cases} 1 & u_i \le u < u_{i+1} \\ 0 & otherwise \end{cases} \\ N_{i,k}(u) &= \frac{u-u_i}{u_{i+k-1}-u_i} N_{i,k-1}(u) + \frac{u_{i+k}-u}{u_{i+k}-u_{i+1}} N_{i+1,k-1}(u), \end{aligned} \tag{2.33}$$

with a non-uniform and non-periodic knot vector

$$\mathbf{U} = \{\underbrace{u_0, u_1, \cdots, u_{k-1}}_{k \text{ equal knots}}, \underbrace{u_k, u_{k+1}, \cdots, u_{p-1}, u_p}_{p-k+1 \text{ internal knots}}, \underbrace{u_{p+1}, \cdots, u_{p+k}}_{k \text{ equal knots}}\}, \tag{2.34}$$

which has $k+p+1$ elements. A NURBS curve has the following properties [93, 91]:

- Geometry invariance property
- End points geometric property
- Convex hull property
- Local support property
- Variation diminishing property

Similarly, a NURBS surface $\mathbf{r}$ of orders $k$ and $l$ and $(m+1) \times (n+1)$ control points is defined by

$$\mathbf{r}(u,v) = \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} w_{ij} \mathbf{R}_{ij} N_{i,k}(u) N_{j,l}(v)}{\sum_{i=0}^{m} \sum_{j=0}^{n} w_{ij} N_{i,k}(u) N_{j,l}(v)}, \quad 0 \le u, v \le 1 \tag{2.35}$$

with non-uniform and non-periodic knot vectors $\mathbf{U}$ and $\mathbf{V}$ for $u$ and $v$ respectively,

$$\begin{aligned} \mathbf{U} &= \{\underbrace{u_0, u_1, \cdots, u_{k-1}}_{k \text{ equal knots}}, \underbrace{u_k, u_{k+1}, \cdots, u_{p-1}, u_p}_{p-k+1 \text{ internal knots}}, \underbrace{u_{p+1}, \cdots, u_{p+k}}_{k \text{ equal knots}}\} \\ \mathbf{V} &= \{\underbrace{v_0, v_1, \cdots, v_{l-1}}_{l \text{ equal knots}}, \underbrace{v_l, v_{l+1}, \cdots, v_{q-1}, v_q}_{q-l+1 \text{ internal knots}}, \underbrace{v_{q+1}, \cdots, v_{q+l}}_{l \text{ equal knots}}\} \end{aligned}$$

where $\mathbf{R}_{ij}$ are the control points, $w_{ij}$ the non-zero weights, and $N_{i,k}$ and $N_{j,l}$ the B-spline basis functions defined in (2.33). Most of the properties of NURBS curves are also applied to NURBS surfaces. However, the variation diminishing property is not applicable to NURBS surface patches.

The derivative of a NURBS curve $\mathbf{q}(u)$ or a NURBS surface $\mathbf{r}(u,v)$, however, is complicated since denominators need to be considered in the derivative calculation. Suppose that a NURBS curve is denoted as $\frac{\mathbf{q}_N(u)}{q_D(u)}$ and a NURBS surface $\frac{\mathbf{r}_N(u,v)}{r_D(u,v)}$. Then, the first derivative of $\mathbf{q}(u)$ with respect to $u$ is given by

$$\frac{d\mathbf{q}(u)}{du} = \frac{\frac{d\mathbf{q}_N(u)}{du} q_D(u) - \mathbf{q}_N(u) \frac{dq_D(u)}{du}}{q_D^2(u)}. \tag{2.36}$$

The first derivative of $\mathbf{r}(u,v)$ with respect to $u$ is given by

$$\frac{\partial \mathbf{r}(u,v)}{\partial u} = \frac{\frac{\partial \mathbf{r}_N(u,v)}{\partial u} r_D(u,v) - \mathbf{r}_N(u,v)\frac{\partial r_D(u,v)}{\partial u}}{r_D^2(u,v)}. \tag{2.37}$$

Similarly, the first derivative of $\mathbf{r}(u,v)$ with respect to $v$ is given by

$$\frac{\partial \mathbf{r}(u,v)}{\partial v} = \frac{\frac{\partial \mathbf{r}_N(u,v)}{\partial v} r_D(u,v) - \mathbf{r}_N(u,v)\frac{\partial r_D(u,v)}{\partial v}}{r_D^2(u,v)}. \tag{2.38}$$

The positive weight provides an additional degree of freedom to control the shape. If all weights are one, then the NURBS representation reduces to the integral B-spline representation. Geometric entities that the integral B-spline formulation cannot represent such as conics (circle, ellipse and hyperbola) or quadrics, tori, cyclides, and surfaces of revolution [75, 93] can be modeled exactly using the NURBS representation.

# Chapter 3

# Mathematical and Computational Prerequisites

Intrinsic properties of a surface such as lines of curvature, umbilical points and geodesic curves, are full-fledged topics in differential geometry. Since they depend only on the geometry of a surface, they are independent of parametrization and representation methods, and invariant to the rigid body transformation such as translation and rotation. Because of such features, they have been widely used for matching and recognition purposes.

Accurate and robust evaluation of the intrinsic properties of a surface is important in order to use them in the applications. If a surface is represented in implicit, explicit or parametric form, then evaluation of the intrinsic properties is performed through analytic or numerical differentiation to yield exact values. In some cases, however, such evaluation alone is not enough depending on the applications. For example, calculation of umbilical points requires to solve a set of nonlinear polynomial equations which cannot be handled easily. In this case, the use of special tools is required.

This chapter is devoted to accurate and robust numerical calculation of various intrinsic properties which are used for matching and copyright protection explained in Chapters 4 and 5.

## 3.1 Literature Review

### 3.1.1 Umbilics

Umbilical points and the behavior of lines of curvature in the vicinity of umbilics have attracted the interest of many researchers. Berry and Hannay [8] classified the *generic* umbilics into three types, i.e. *star, lemon* and *monstar*, based on the coefficients of the cubic part in the Taylor expanded representation of a local surface and indices. They also showed the rarity of monstar patterns in the surfaces based on the *statistical singularity theory.* Porteous [95, 96] gave a thorough mathematical treatment of ridges and umbilics of surfaces, and Morris [79] studied ridges and subparabolic lines in particular. He also provided practical formulae for calculating ridges and sub-parabolic lines with a bumpy cube as an example. Sander and Zucker [103, 104] studied a problem of extraction of umbilical points from 3D images and provided a computational method to identify umbilical points. An iteration scheme for a functional minimization under the compatibility constraints is used to refine the principal direction field and curvature estimates. The indices of the umbilics are calculated in the direction field to classify them.

Sinha [108] calculated differential properties of a surface represented in range data by using a global energy-minimizing thin-plate surface fit, and examined the effect of changing parameters in the surface fitting stage on the differential properties of the surface. Maekawa *et al.* [74] discussed a mathematical aspect of the generic features of free-form parametric surfaces, described a method to extract them, and investigated the generic features of umbilics and behaviors of lines of curvature around umbilical points on a parametric free-form surface. They presented novel and practical criteria which assure the existence of local extrema of principal curvature functions at umbilic points. The survey paper of Farouki [40] on various techniques for interrogation of free-form parametric surfaces based on differential geometry, reviews the theories of surface curvatures, and provides an integration method for lines of curvature. Maekawa *et al.* [74] discussed possible problems that can be encountered in the lines of curvature calculation and proposed a criterion to make the solution path not reverse its direction.

### 3.1.2 Principal Patches

Principal patches are defined as the patches whose sides are lines of curvature [75, 76]. They depend only on the shape of the surface and are independent of the parametrization or representation methods. These properties have encouraged researchers to study them for use in computational geometry and CAD applications. Martin [75, 76] proposed a method for creating surface patches whose boundaries are lines of curvature. Two constraints, i.e. the frame and the position matching equations, are imposed on the boundary curves, which ensure that the boundary curves are lines of curvature and the surface normals of two adjacent patches along the boundary are the same so that the surface continuity is preserved. As a practical example for principal patches, Dupin's cyclides, which have circular lines of curvature, are taken and discussed, and Dutta *et al.* [38] used cyclides in surface blending for solid modeling. Sinha and Besl [109] reviewed mathematical aspects on principal patches and presented an engineering solution to creating global principal patch networks. A meshing algorithm is proposed and the computational difficulties of building a quadrilateral mesh based on integrated lines of curvature are discussed such as isolation of planar and spherical regions, determination of directions of principal curvature, treatment of umbilics, etc. A concept similar to principal patches was proposed by Thirion [115]. He defined *the extremal mesh* as the graph whose vertices are the *extremal points* or umbilic points and whose edges are the *extremal lines*. The basic idea of the extremal mesh is the same as that of principal patches but he used a new local geometric invariant of 3D surface to resolve orientation problem arising in the construction of the mesh. Brady *et al.* [15] analyzed several classes of surface curves as a source of constraint on the surface and as a basis for describing it, such as bounding contours, surface intersections, lines of curvature and asymptotes. The surface is smoothed out with appropriate 2D Gaussian masks which is based on convolution of the input data with a Gaussian function and then operators to estimate the first and second derivatives are used, leading to computation of the principal curvatures. Umbilics are detected based on the principal curvatures, and lines of curvature are extracted by minimizing a closeness evaluation function.

An umbilical point can be calculated from the fact that the principal curvatures at that point are equal. For a surface represented in analytical form, this produces a set of nonlinear system of equations which can be solved by using software tools such as Matlab. When a surface is provided as range data, application of the condition for umbilical point detection is not robust. A different approach needs to be introduced such as the index calculation after refinement of principal curvature fields, see Sander and Zucker [103, 104].

## 3.2 Rotation and Translation

Suppose that we have two 3-tuples $\mathbf{m}_i$ and $\mathbf{n}_i$ $(i = 1, 2, 3)$, and correspondence information for each point. From these points, a translation vector and a rotation matrix can be calculated. The translation vector is easily obtained by using the centroids of each 3-tuple. The centroids $\mathbf{c}_m$ and $\mathbf{c}_n$ are given by

$$\mathbf{c}_m = \frac{1}{3}\sum_{i=1}^{3}\mathbf{m}_i, \quad \mathbf{c}_n = \frac{1}{3}\sum_{i=1}^{3}\mathbf{n}_i, \tag{3.1}$$

and the difference between $\mathbf{c}_m$ and $\mathbf{c}_n$ becomes the translation vector $\mathbf{t}_T = \mathbf{c}_n - \mathbf{c}_m$. A rotation matrix consists of three unknown components (the Euler angles). Since the two 3-tuples provide nine constraints, the rotation matrix may be constructed by using some of the constraints. But the results could be different if the remaining constraints are used for the rotation matrix calculation [48]. In order to use all the constraints equally, the least squares method may be employed [48]. The basic solution by Horn [48] is described below. Suppose that the translation has been performed. Then what is left is to find the rotation matrix $\mathbf{R}$ so that

$$\begin{aligned}\Phi' &= \sum_{i=1}^{3}|\mathbf{n}_i - (\mathbf{R}\mathbf{m}_i)|^2 \\ &= \sum_{i=1}^{3}|\mathbf{n}_i|^2 - 2\sum_{i=1}^{3}\mathbf{n}_i \cdot (\mathbf{R}\mathbf{m}_i) + \sum_{i=1}^{3}|\mathbf{R}\mathbf{m}_i|^2 \end{aligned} \tag{3.2}$$

is minimized. Here, $D = \sum_{i=1}^{3}\mathbf{n}_i \cdot (\mathbf{R}\mathbf{m}_i)$ has to be maximized to minimize $\Phi'$. The problem can be solved in the quaternion framework. A quaternion can be considered as a vector with four components, i.e. a vector part in 3D and a scalar part. A rotation can be equivalently defined as a unit quaternion $\check{\mathbf{q}} = \left[\cos\frac{\theta}{2}, a_x \sin\frac{\theta}{2}, a_y \sin\frac{\theta}{2}, a_z \sin\frac{\theta}{2}\right]$

which represents a rotation movement around $(a_x, a_y, a_z)$ by $\theta$ degree. In the quaternion framework, the problem is reduced to the eigenvalue problem of the $4 \times 4$ matrix $\mathbf{H}$ obtained from the correlation matrix $\mathbf{M}$:

$$\mathbf{H} = \left[ \begin{array}{cccc} s_{11} + s_{22} + s_{33} & s_{23} - s_{32} & s_{31} - s_{13} & s_{12} - s_{21} \\ s_{23} - s_{32} & s_{11} - s_{22} - s_{33} & s_{12} + s_{21} & s_{31} + s_{13} \\ s_{31} - s_{13} & s_{12} + s_{21} & s_{22} - s_{11} - s_{33} & s_{23} + s_{32} \\ s_{12} - s_{21} & s_{31} + s_{13} & s_{23} + s_{32} & s_{33} - s_{22} - s_{11} \end{array} \right], \tag{3.3}$$

where

$$\mathbf{M} = \sum_{i=1}^{3} \mathbf{n}_i \mathbf{m}_i^T = \left[ \begin{array}{ccc} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{array} \right]. \tag{3.4}$$

The eigenvector corresponding to the maximum positive eigenvalue is a quaternion which minimizes equation (3.2). An orthonormal rotation matrix $\mathbf{R}$ can be recovered from a unit quaternion $\check{q} = [q_0, q_1, q_2, q_3]$ by

$$\mathbf{R} = \left[ \begin{array}{ccc} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{array} \right]. \tag{3.5}$$

The procedure described above can also be applied to the case where more than three corresponding point pairs are provided.

## 3.3 Lines of Curvature

Depending on the size of the coefficients, $(L+\kappa E)$ and $(N+\kappa G)$, either (2.12) or (2.13) are selectively used to trace a line of curvature. Namely, if $|(L + \kappa E)| \leq |(N + \kappa G)|$, we solve (2.12). Otherwise, solve (2.13) to avoid numerical instability [3].

The factors $\eta$ and $\mu$ are determined by using the normalization condition of the first fundamental form. Since lines of curvature are arc length parametrized, the first fundamental form is reduced to

$$E\left(\frac{du}{ds}\right)^2 + 2F\frac{du}{ds}\frac{dv}{ds} + G\left(\frac{dv}{ds}\right)^2 = 1. \tag{3.6}$$

Substituting (2.12) into (3.6), we can obtain $\eta$ by

$$\eta = \frac{\pm 1}{\sqrt{E(M + \kappa F)^2 - 2F(M + \kappa F)(L + \kappa E) + G(L + \kappa E)^2}}. \tag{3.7}$$

Similarly, we can calculate $\mu$ by

$$\mu = \frac{\pm 1}{\sqrt{E(N + \kappa G)^2 - 2F(N + \kappa G)(M + \kappa F) + G(M + \kappa F)^2}}. \tag{3.8}$$

The choice of the sign of $\eta$ and $\mu$ is based on the following inequality [91] to maintain the direction of the solution path:

$$\begin{aligned}&\left|-\left(\frac{du^p}{ds}\frac{\partial \mathbf{r}^p}{\partial u}+\frac{dv^p}{ds}\frac{\partial \mathbf{r}^p}{\partial v}\right)-\left(\frac{du}{ds}\frac{\partial \mathbf{r}}{\partial u}+\frac{dv}{ds}\frac{\partial \mathbf{r}}{\partial v}\right)\right| \\ &< \left|\left(\frac{du^p}{ds}\frac{\partial \mathbf{r}^p}{\partial u}+\frac{dv^p}{ds}\frac{\partial \mathbf{r}^p}{\partial v}\right)-\left(\frac{du}{ds}\frac{\partial \mathbf{r}}{\partial u}+\frac{dv}{ds}\frac{\partial \mathbf{r}}{\partial v}\right)\right|,\end{aligned}$$

where $\mathbf{r} = \mathbf{r}(u(s), v(s))$ is a line of curvature on the surface $\mathbf{r}$ with respect to the arc length $s$ and the superscript $p$ means the previous step during the numerical integration of the governing equations. A well known numerical method such as Runge-Kutta method or Adams method is adopted as a solution scheme to the system of nonlinear ordinary differential equations, see [91].

## 3.4 Geodesics

There are two types of geodesic problems which may arise in real applications: one is the initial value problem (IVP) and the other is the boundary value problem (BVP). For IVP, the fourth order Runge-Kutta method can be applied to (3.39) through (3.43). However, in most cases, a geodesic problem comes in the form of BVP. It is well known that the solution to IVP is unique, whereas BVP may have many solutions or no solution at all. Two methods, the *shooting method* and the *relaxation method*, are available for the solution to BVP. The relaxation method is considered more stable than the shooting method [70]. What follows is a brief discussion about the relaxation method and its application to the orthogonal projection of geodesic curves.

### Relaxation Method [70, 98, 91]

The boundary value problem can be solved with the *relaxation* method. It approximates ordinary differential equations (ODEs) with finite difference equations (FDEs) on mesh points in the domain of interest [98, 70]. It starts with an initial guess and then iteratively converges to the solution, which is called to *relax* to the true solution [98]. Many schemes can be used to represent ODEs in the form of FDEs. In this work, the trapezoidal rule is adopted [70].

Let us assume that we have a first order differential equation as follows:

$$\frac{dy}{dx} = g(x, y). \tag{3.9}$$

At two consecutive points $k$ and $k-1$, the trapezoidal rule turns equation (3.9) into

$$y_k - y_{k-1} - (x_k - x_{k-1})\frac{1}{2}(g_k(x_k, y_k) + g_{k-1}(x_{k-1}, y_{k-1})) = 0. \tag{3.10}$$

A set of differential equations can be represented in vector form so that it can be treated as a single differential equation. Suppose that

$$\begin{aligned} \mathbf{y} &= (y_1, y_2, \cdots, y_n)^T, \qquad & \mathbf{g} &= (g_1, g_2, \cdots, g_n)^T, \\ \boldsymbol{\alpha} &= (\alpha_1, \alpha_2, \cdots, \alpha_n)^T, \qquad & \boldsymbol{\beta} &= (\beta_1, \beta_2, \cdots, \beta_n)^T. \end{aligned} \tag{3.11}$$

Then, a system of first order differential equations with two boundary conditions at **A** and **B** is given as follows:

$$\frac{d\mathbf{y}}{ds} = \mathbf{g}(s, \mathbf{y}), \quad \mathbf{y}(A) = \boldsymbol{\alpha}, \ \mathbf{y}(B) = \boldsymbol{\beta}, \tag{3.12}$$

where $s \in [A, B]$. Equivalently, the vector equation (3.12) can be represented in the finite difference form using the trapezoidal rule as follows:

$$\frac{\mathbf{Y}_k - \mathbf{Y}_{k-1}}{s_k - s_{k-1}} = \frac{1}{2}\left[\mathbf{G}_k + \mathbf{G}_{k-1}\right], k = 2, 3, \cdots, m, \tag{3.13}$$

with boundary conditions

$$\mathbf{Y}_1 = \boldsymbol{\alpha}, \quad \mathbf{Y}_m = \boldsymbol{\beta}, \tag{3.14}$$

where a mesh of points satisfying $A = s_1 < s_2 < \cdots < s_m = B$ is considered. Here, the $n$-vectors $\mathbf{Y}_k$ and $\mathbf{G}_k$ are discrete approximate values of $\mathbf{y}_k(s_k)$ and $\mathbf{g}_k(s_k)$.

Let us refer to equation (3.13) as

$$\mathbf{F}_k = (F_{1,k}, F_{2,k}, \cdots, F_{n,k})^T = \frac{\mathbf{Y}_k - \mathbf{Y}_{k-1}}{s_k - s_{k-1}} - \frac{1}{2}\left[\mathbf{G}_k + \mathbf{G}_{k-1}\right] = 0, k = 2, 3, \cdots, m \tag{3.15}$$

and equations (3.14) as

$$\begin{aligned} \mathbf{F}_1 &= (F_{1,1}, F_{2,1}, \cdots, F_{n,1})^T = \mathbf{Y}_1 - \boldsymbol{\alpha} = \mathbf{0}, \\ \mathbf{F}_{m+1} &= (F_{1,m+1}, F_{2,m+1}, \cdots, F_{n,m+1})^T = \mathbf{Y}_m - \boldsymbol{\beta} = \mathbf{0}. \end{aligned} \tag{3.16}$$

Then we have $mn$ nonlinear algebraic equations

$$\mathbf{F} = (\mathbf{F}_1^T, \mathbf{F}_2^T, \cdots, \mathbf{F}_{m+1}^T)^T = \mathbf{0}. \tag{3.17}$$

The vector equation (3.17) can be solved by the Newton's iteration scheme.

For a more stable solution, a step correction procedure can be adopted as follows [70]:

$$\mathbf{Y}^{(i+1)} = \mathbf{Y}^{(i)} + \mu \Delta \mathbf{Y}^{(i)}, \tag{3.18}$$

where $0 < \mu \leq 1$ is chosen so that $||\Delta \mathbf{Y}^{(i+1)}||_1 < ||\Delta \mathbf{Y}^{(i)}||_1$. Here $||\Delta \mathbf{Y}||_1$ is defined as:

$$||\Delta \mathbf{Y}^{(i)}||_1 = \sum_{k=1}^{m} \left( \frac{|\Delta u_k|}{M_u} + \frac{|\Delta v_k|}{M_v} + \frac{|\Delta p_k|}{M_p} + \frac{|\Delta q_k|}{M_q} \right), \tag{3.19}$$

where $M_u$, $M_v$, $M_p$ and $M_q$ are the scale factors for each variable. The values of $M_u = M_v = 1$ and $M_p = M_q = 10$ are used in the calculation, see [70, 91].

## 3.5 Orthogonal Projection of Points and Curves

Orthogonal projection is a well-known concept, and has many engineering and scientific applications. For example, in shipbuilding industry steel plates need to be trimmed off along pre-defined trimming lines before assembly [92], and in inspection of manufactured objects computation of orthogonal projection of measured points onto the CAD surface is a critical step [91].

The idea of the orthogonal projection of a point or a curve onto a surface proposed by Pegna and Wolter [92] is extended to include orthogonal projection of curves on a surface such as a line of curvature and a geodesic curve. Instead of discretizing lines of curvature or geodesic curves and then projecting the points, a set of differential equations is formulated, which directly trace the orthogonally projected curve using the concept of the orthogonal projection of a curve onto a surface by Pegna and Wolter [92].

In this section, a brief review of the orthogonal projection of a point and a curve by Pegna and Wolter [92] is provided followed by extension of the curve projection idea to the orthogonal projection of lines of curvature and geodesic curves.

### 3.5.1 Introduction

A few assumptions need to be made that a surface onto which a point or a curve is projected should be *regular* and second order continuous, a 3D curve remains close enough to the surface, and the projected point or curve lies in the interior of the surface. In this thesis, only a parametric surface $\mathbf{r} = \mathbf{r}(u, v)$, $(0 \leq u, v \leq 1)$ is considered.

### 3.5.2 Points

The orthogonal projection of a point $\mathbf{p}$ onto a surface $\mathbf{r}$ is defined as a set of points such that

$$\mathbf{Q}(\mathbf{p}, \mathbf{r}) = \left\{ \mathbf{q} | \mathbf{q} = \mathbf{r}(\mathbf{x}_q) \ \ s.t. \ \ (\mathbf{p} - \mathbf{q}) \cdot \frac{\partial \mathbf{r}(\mathbf{x}_q)}{\partial x^i} = 0; \mathbf{x}_q = (x_q^1, x_q^2), 0 \leq i \leq 2 \right\}. \tag{3.20}$$

The existence of the orthogonal projection is not always guaranteed when the surface $\mathbf{r}$ has boundary, and more than one projected points may exist [92].

#### Formulation

The computation of an orthogonally projected point is easily done by Newton's method. Let us assume that $\mathbf{p}$ is a given point and $\mathbf{q}$ a projected point on a surface $\mathbf{r}$.

From the definition (3.20), two equations with two unknowns are obtained as follows:

$$\begin{aligned} f_1 = (\mathbf{p} - \mathbf{q}) \cdot \frac{\partial \mathbf{r}}{\partial u} &= 0, \\ f_2 = (\mathbf{p} - \mathbf{q}) \cdot \frac{\partial \mathbf{r}}{\partial v} &= 0. \end{aligned} \tag{3.21}$$

Suppose that $\mathbf{F} = [f_1 \;\; f_2]^T$ and $\mathbf{x} = (u, v)^T$. The correction $\delta\mathbf{x}$ which directs each $f_i$ to zero can be obtained by

$$\delta\mathbf{x} = -\mathbf{J}^{-1}\mathbf{F}(\mathbf{x}), \tag{3.22}$$

where,

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial u} & \frac{\partial f_1}{\partial v} \\ \frac{\partial f_2}{\partial u} & \frac{\partial f_2}{\partial v} \end{bmatrix}. \tag{3.23}$$

Then, the vector $\mathbf{x}$ is adjusted with the correction $\delta\mathbf{x}$,

$$\mathbf{x}^{new} = \mathbf{x}^{old} + \delta\mathbf{x}. \tag{3.24}$$

Iteration continues until $|\delta\mathbf{x}|$ becomes smaller than the user-specified tolerance.

### 3.5.3 Curves

The orthogonal projection of a curve onto a surface is a direct extension of the orthogonal projection of a point (3.20). What follows is a summary of the work by Pegna and Wolter [92].

The orthogonal projection of a space curve onto a surface is defined as follows. Let $\mathbf{\Gamma}(t)$ be a space curve with $0 \le t \le 1$ and $\mathbf{p}$ is a point on $\mathbf{\Gamma}(t)$. Assuming that the projection of curve $\mathbf{r}(t)$ onto the surface $\mathbf{r}$ is a curve $\mathbf{R}$, then this curve is defined as follows:

$$\mathbf{R} = \mathbf{R}(\mathbf{\Gamma}(t), \mathbf{r}) = \{\mathbf{q} | \exists \mathbf{p} \in \mathbf{\Gamma}(t), \mathbf{q} \in \mathbf{Q}(\mathbf{p}, \mathbf{r})\}. \tag{3.25}$$

Suppose we have a parametric surface $\mathbf{r}(u, v)$, $(0 \le u, v \le 1)$ and a space curve $\mathbf{\Gamma}(t)$ with a parameter $t$, $(0 \le t \le 1)$. Then the equations of the projected curve of $\mathbf{\Gamma}(t)$ onto $\mathbf{r}(u, v)$, or $\boldsymbol{\gamma}(t) = \mathbf{r}(u(t), v(t))$ are derived from the definition (3.25):

$$\begin{aligned} (\boldsymbol{\gamma}(t) - \mathbf{\Gamma}(t)) \cdot \frac{\partial \mathbf{r}}{\partial u} = 0, \\ (\boldsymbol{\gamma}(t) - \mathbf{\Gamma}(t)) \cdot \frac{\partial \mathbf{r}}{\partial v} = 0. \end{aligned} \tag{3.26}$$

We take the derivative of equations (3.26) with respect to $t$. Then we obtain the following:

$$\begin{aligned} \frac{d\boldsymbol{\gamma}(t)}{dt} \cdot \frac{\partial \mathbf{r}}{\partial u} + (\boldsymbol{\gamma}(t) - \mathbf{\Gamma}(t)) \cdot \frac{d}{dt}\left(\frac{\partial \mathbf{r}}{\partial u}\right) &= \frac{d\mathbf{\Gamma}(t)}{dt} \cdot \frac{\partial \mathbf{r}}{\partial u}, \\ \frac{d\boldsymbol{\gamma}(t)}{dt} \cdot \frac{\partial \mathbf{r}}{\partial v} + (\boldsymbol{\gamma}(t) - \mathbf{\Gamma}(t)) \cdot \frac{d}{dt}\left(\frac{\partial \mathbf{r}}{\partial v}\right) &= \frac{d\mathbf{\Gamma}(t)}{dt} \cdot \frac{\partial \mathbf{r}}{\partial v}. \end{aligned} \tag{3.27}$$

Using

$$\frac{d\boldsymbol{\gamma}(t)}{dt} = \frac{\partial \mathbf{r}}{\partial u}\frac{du}{dt} + \frac{\partial \mathbf{r}}{\partial v}\frac{dv}{dt}, \tag{3.28}$$

and rearranging equations (3.27), we obtain the following:

$$\begin{bmatrix} \frac{du}{dt} \\ \frac{dv}{dt} \end{bmatrix} = \mathbf{K}^{-1} \begin{bmatrix} \frac{d\mathbf{p}}{dt} \cdot \frac{\partial \mathbf{r}}{\partial u} \\ \frac{d\mathbf{p}}{dt} \cdot \frac{\partial \mathbf{r}}{\partial v} \end{bmatrix}, \tag{3.29}$$

where

$$\mathbf{K} = \begin{bmatrix} \frac{\partial \mathbf{r}}{\partial u} \cdot \frac{\partial \mathbf{r}}{\partial u} + (\boldsymbol{\gamma}(t) - \boldsymbol{\Gamma}(t)) \cdot \frac{\partial^2 \mathbf{r}}{\partial u^2} & \frac{\partial \mathbf{r}}{\partial u} \cdot \frac{\partial \mathbf{r}}{\partial v} + (\boldsymbol{\gamma}(t) - \boldsymbol{\Gamma}(t)) \cdot \frac{\partial^2 \mathbf{r}}{\partial u \partial v} \\ \frac{\partial \mathbf{r}}{\partial u} \cdot \frac{\partial \mathbf{r}}{\partial v} + (\boldsymbol{\gamma}(t) - \boldsymbol{\Gamma}(t)) \cdot \frac{\partial^2 \mathbf{r}}{\partial u \partial v} & \frac{\partial \mathbf{r}}{\partial v} \cdot \frac{\partial \mathbf{r}}{\partial v} + (\boldsymbol{\gamma}(t) - \boldsymbol{\Gamma}(t)) \cdot \frac{\partial^2 \mathbf{r}}{\partial v^2} \end{bmatrix}. \tag{3.30}$$

We know that

$$\boldsymbol{\gamma}(t) - \boldsymbol{\Gamma}(t) = \rho \mathbf{N}, \quad \mathbf{N} = \frac{\frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v}}{\left| \frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} \right|}, \tag{3.31}$$

where $\rho = |\boldsymbol{\gamma}(t) - \boldsymbol{\Gamma}(t)|$. Then, the matrix $\mathbf{K}$ is reduced to

$$\mathbf{K} = \begin{bmatrix} E + \rho L & F + \rho M \\ F + \rho M & G + \rho N \end{bmatrix}, \tag{3.32}$$

where $E$, $F$ and $G$ are the first fundamental form coefficients, and $L$, $M$ and $N$ the second fundamental form coefficients of $\mathbf{r}$. The Runge-Kutta method can be used for the solution of equations (3.29).

### 3.5.4 Lines of Curvatures

The orthogonal projection of a line of curvature can be calculated using equations (3.29). Namely, the curve $\boldsymbol{\Gamma}$ in Section 3.5.3 is replaced by a line of curvature with the arc length parametrization.

Suppose we have two surfaces $\mathbf{r}_A = \mathbf{r}_A(u_A, v_A)$ and $\mathbf{r}_B = \mathbf{r}_B(u_B, v_B)$, where $0 \leq u_A, v_A \leq 1$ and $0 \leq u_B, v_B \leq 1$. $\mathbf{r}_B$ is a surface onto which a line of curvature is projected, and $\mathbf{r}_A$ is a surface on which the line of curvature is calculated. We denote a line of curvature on the surface $\mathbf{r}_A$ as $\boldsymbol{\Gamma}_A(s) = \mathbf{r}_A(u_A(s), v_A(s))$, where $s$ is the arc length parameter for $\mathbf{r}_A$. From equations (3.29), we obtain

$$\begin{bmatrix} \frac{du_B}{ds} \\ \frac{dv_B}{ds} \end{bmatrix} = \mathbf{K}_B^{-1} \begin{bmatrix} \frac{d\boldsymbol{\Gamma}_A}{ds} \cdot \frac{\partial \mathbf{r}_B}{\partial u_B} \\ \frac{d\boldsymbol{\Gamma}_A}{ds} \cdot \frac{\partial \mathbf{r}_B}{\partial v_B} \end{bmatrix}, \tag{3.33}$$

where

$$\mathbf{K}_B = \begin{bmatrix} E_B + \rho L_B & F_B + \rho M_B \\ F_B + \rho M_B & G_B + \rho N_B \end{bmatrix}. \tag{3.34}$$

Here, the subscript $_B$ indicates the quantities for the surface $\mathbf{r}_B$ and the subscript $_A$

those for the surface $\mathbf{r}_A$. By the chain rule, we have

$$\frac{d\mathbf{\Gamma}_A}{ds} = \frac{\partial \mathbf{r}_A}{\partial u_A}\frac{du_A}{ds} + \frac{\partial \mathbf{r}_A}{\partial v_A}\frac{dv_A}{ds}. \tag{3.35}$$

The differential equations for the line of curvature (2.12) or (2.13) are applied to the surface $\mathbf{r}_A$. Therefore, we have differential equations for the orthogonal projection of the line of curvature as follows:

$$\begin{aligned} \frac{du_A}{ds} &= \eta_A(M_A + \kappa_A F_A) \quad or \quad \mu_A(N_A + \kappa_A G_A), && (3.36)\\ \frac{dv_A}{ds} &= -\eta_A(L_A + \kappa_A E_A) \quad or \quad -\mu_A(M_A + \kappa_A F_A), && (3.37) \end{aligned}$$

and

$$\begin{bmatrix} \frac{du_B}{ds} \\ \frac{dv_B}{ds} \end{bmatrix} = \mathbf{K}_B^{-1} \begin{bmatrix} \left(\frac{\partial \mathbf{r}_A}{\partial u_A}\frac{du_A}{ds} + \frac{\partial \mathbf{r}_A}{\partial v_A}\frac{dv_A}{ds}\right) \cdot \frac{\partial \mathbf{r}_B}{\partial u_B} \\ \left(\frac{\partial \mathbf{r}_A}{\partial u_A}\frac{du_A}{ds} + \frac{\partial \mathbf{r}_A}{\partial v_A}\frac{dv_A}{ds}\right) \cdot \frac{\partial \mathbf{r}_B}{\partial v_B} \end{bmatrix}. \tag{3.38}$$

These equations (3.36) to (3.38) can be solved by using Runge-Kutta method as the initial value problem.

### 3.5.5 Geodesics

Suppose we have two surfaces $\mathbf{r}_A = \mathbf{r}_A(u_A, v_A)$ and $\mathbf{r}_B = \mathbf{r}_B(u_B, v_B)$, where $0 \le u_A, v_A \le 1$ and $0 \le u_B, v_B \le 1$, where $\mathbf{r}_B$ is a surface onto which a geodesic curve is projected, and $\mathbf{r}_A$ is a surface on which the geodesic curve is calculated. We denote a geodesic curve on the surface $\mathbf{r}_A$ as $\mathbf{\Gamma}_A(s) = \mathbf{r}_A(u_A(s), v_A(s))$, where $s$ is the arc length parameter on $\mathbf{r}_A$. From (2.19) to (2.22), we can calculate $\frac{du_A}{ds}$ and $\frac{dv_A}{ds}$. Combining all of them, we obtain two sets of differential equations as follows:

$$\begin{aligned} \frac{du_A}{ds} &= p, && (3.39)\\ \frac{dv_A}{ds} &= q, && (3.40)\\ \frac{dp}{ds} &= -\Gamma^1_{11}p^2 - 2\Gamma^1_{12}pq - \Gamma^1_{22}q^2, && (3.41)\\ \frac{dq}{ds} &= -\Gamma^2_{11}p^2 - 2\Gamma^2_{12}pq - \Gamma^2_{22}q^2, && (3.42) \end{aligned}$$

and

$$\begin{bmatrix} \frac{du_B}{ds} \\ \frac{dv_B}{ds} \end{bmatrix} = \mathbf{K}_B^{-1} \begin{bmatrix} \left(\frac{\partial \mathbf{r}_A}{\partial u_A}p + \frac{\partial \mathbf{r}_A}{\partial v_A}q\right) \cdot \frac{\partial \mathbf{r}_B}{\partial u_B} \\ \left(\frac{\partial \mathbf{r}_A}{\partial u_A}p + \frac{\partial \mathbf{r}_A}{\partial v_A}q\right) \cdot \frac{\partial \mathbf{r}_B}{\partial v_B} \end{bmatrix}, \tag{3.43}$$

where $\mathbf{K}_B$ is the matrix $\mathbf{K}$ of (3.32) for the surface $\mathbf{r}_B$.

Application of the relaxation method to the orthogonal projection of geodesics in BVP form is an extension of the work by Maekawa [70, 91]. Additional two differential equations (3.43) are obtained which trace orthogonally projected curves on the target surface $\mathbf{r}_B$ with the arc length parameter of the model surface $\mathbf{r}_A$. These two equations are turned into equivalent finite difference equations and provided for the solution scheme.

Six values are needed to solve the problem. Let us denote two boundaries $b_1$ and $b_2$, respectively. First, at both boundaries, we have four parametric values $u_{b_1}, v_{b_1}, u_{b_2}$ and $v_{b_2}$. These four parametric values are enough to determine a geodesic curve between $b_1$ and $b_2$ on the model surface. The remaining two values denoted as $u_{B0}$ and $v_{B0}$, which are initial values of the projected geodesic curve on the target surface are applied to the differential equations (3.43).

### 3.5.6 Calculation of Initial Values

Tracing a curve projected on a surface is, in general, formulated as the initial value problem. Therefore, selection of initial values is an important step for an accurate solution.

In the problem of the orthogonal projection of a curve onto a surface, the starting point of the curve is provided, and from this point an initial point for the orthogonally projected curve has to be calculated. Since we are dealing with the orthogonal projection, it would be a reasonable choice to use an orthogonally projected point of the starting point of the curve by using equations (3.21) which can be solved by Newton's method. However, it requires a good initial approximation for the solution to (3.21). In this work, the IPP algorithm is used to find an initial point by solving (3.21) robustly [125].

### 3.5.7 Examples

**Lines of Curvature**

Two B-spline surfaces with 16 (4×4) control points as shown in Figure 3-1 are used for demonstration of the orthogonal projection of lines of curvature. The upper surface is the model surface on which a line of curvature is calculated and the lower surface is the target surface where the line of curvature is projected. Four initial values are provided: two values are $u_0 = 0.3$ and $v_0 = 0.3$ for the line of curvature on the model surface, and the other two are $u_B = 0.3003$ and $v_B = 0.2995$ for the initial values of the orthogonally projected curve on the target surface. A line of curvature and its projection on the target surface are presented in Figure 3-1.

**Geodesic Curves**

Two B-spline surfaces as shown in Figure 3-2 are selected. The upper surface is the model surface with 16 $(4 \times 4)$ control points on which a geodesic is calculated and the lower surface with 16 $(4 \times 4)$ control points is the target surface where the geodesic curve is projected. Two end points are given so that the problem is treated as BVP. In total, six boundary values are necessary. Four parametric values of two boundary points $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, $u_{b_1} = 0.2$, $v_{b_1} = 0.3$, $u_{b_2} = 0.8$ and $v_{b_2} = 0.9$ are provided as input to the solution of the geodesic curve on the model surface. The parametric values

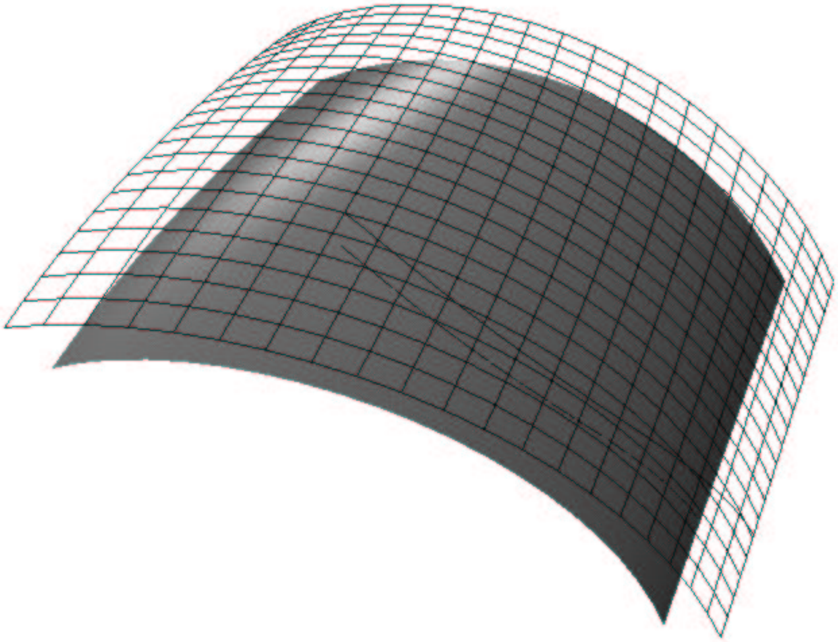


Figure 3-1: The orthogonal projection of a line of curvature

$u_{B0}$ and $v_{B0}$ are 0.2002 and 0.2987, which are used as initial values of the projected curve on the target surface. A straight line between $(u_{b_1}, v_{b_1})$ and $(u_{b_2}, v_{b_2})$ is used as an initial approximation of the solution. Thirty three discrete mesh points are used at each of which ODEs are turned into FDEs using the trapezoidal rule. Figure 3-2 shows the result which is obtained after five iterations under the tolerance $\epsilon = 10^{-5}$.

## 3.6 Extraction of Umbilical Points

Umbilical points appear in various forms on a free-form surface such as isolated points, lines or regions. Because of the singular behavior of the principal direction field around the umbilics, information on their locations is important in surface-intrinsic-wireframing (see Section 3.7), and in selection of reference points for matching. Moreover, isolated generic umbilical points need to be located and classified for the strong test of decision algorithms for similarity (see Chapter 5). Therefore, a robust and efficient algorithm for extraction of umbilical points and regions is necessary, and is presented below.

The quadtree decomposition is used to extract umbilical points or regions from a free-form NURBS surface. Such decomposition has been used for various purposes such as intersection problems [33] and computer graphics [102]. Combining the *convex*

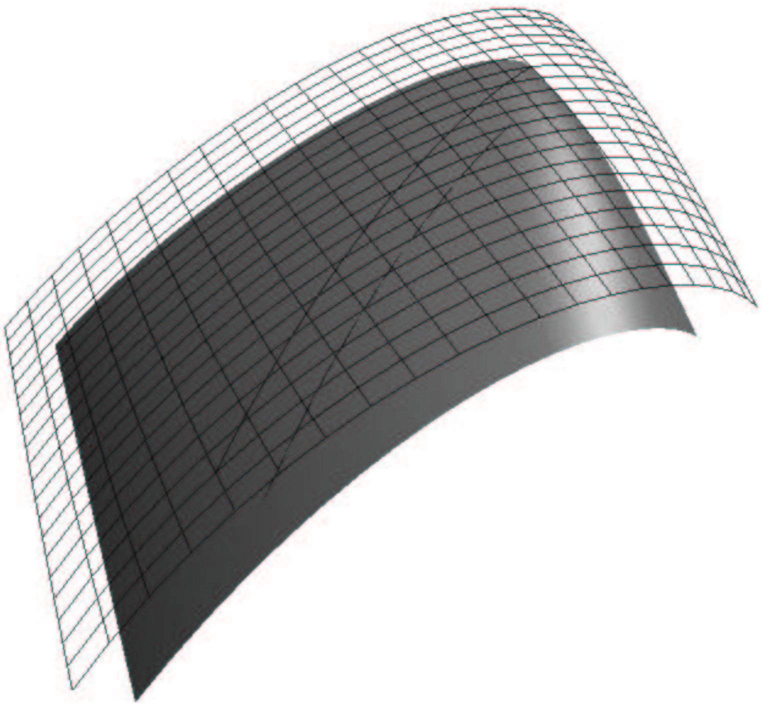


Figure 3-2: The orthogonal projection of a geodesic

*hull property* of the Bernstein polynomials, the quadtree decomposition provides an efficient method for extraction of umbilical points from the free-form NURBS surface, especially when the umbilics are not isolated.

A NURBS surface is first subdivided into rational Bézier surface patches by knot insertion and the governing equations (2.24) are formulated for each resulting rational Bézier surface patch, which are reduced to equations (2.25). The graph of $z = P_N(u,v)\ \ (0 \leq u,v \leq 1)$ is represented over the $uv$ parametric plane. The condition $P_N(u,v) \geq 0$ assures that no portion of $P_N$ lies below the $uv$ plane, i.e. $P_N$ has no negative value. Suppose that $\mathbf{P}_N = (u,v,P_N)$. Given an integral Bézier surface patch of degree $m$ and $n$ in $u$ and $v$, $\mathbf{P}_N(u,v)$ in the bivariate Bernstein form is given by [74]:

$$\mathbf{P}_N(u,v) = \sum_{i=0}^{10m-6} \sum_{j=0}^{10n-6} \mathbf{p}_{ij} B_{i,10m-6}(u) B_{j,10n-6}(v), \tag{3.44}$$

and for a rational Bézier surface patch of degree $m$ and $n$ in $u$ and $v$, $\mathbf{P}_N$ is given by [74]:

$$\mathbf{P}_N(u,v) = \sum_{i=0}^{24m-6} \sum_{j=0}^{24n-6} \mathbf{p}_{ij} B_{i,24m-6}(u) B_{j,24n-6}(v), \tag{3.45}$$

where $\mathbf{p}_{ij}$ are Bernstein coefficients. Then the detection problem is reduced to find a set of $(u, v)$ which satisfy $P_N(u, v) = 0$, or the zero-set of a bivariate Bernstein polynomial (3.44) or (3.45). An adaptive quadtree decomposition on the $uv$ domain is used to narrow down regions possibly containing umbilical points. A rectangular domain is subdivided into four rectangular domains at the mid points of $u$ and $v$ using the de Casteljau algorithm. This algorithm can be robustly executed in rounded interval arithmetic. At a depth $d$, there are at most $4^d$ nodes and each node has a domain of size $r2^{-d} \leq u \leq (r+1)2^{-d}$ and $s2^{-d} \leq v \leq (s+1)2^{-d}$ where $r$ and $s$ are integers of $0 \leq r \leq 4^{d-1}$ and $0 \leq s \leq 4^{d-1}$, respectively. Figure 3-3 shows an example of the quadtree decomposition. The depth of the quadtree depends on the user-specified size of subdivided regions.

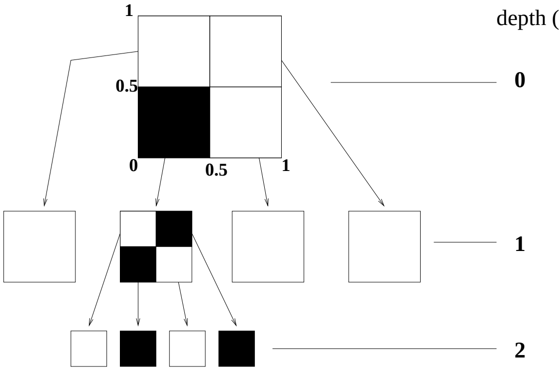


Figure 3-3: An example of the adaptive quadtree decomposition (The marked dark domains indicate those which possibly contain umbilical points.)

The strategy of the adaptive quadtree decomposition algorithm for extraction of umbilical points is to eliminate the regions which do not include any umbilical points using the convex hull property. The convex hull property determines if a subdivided region does not contain roots of $P_N(u, v) = 0$ or not. This property assures that if all Bernstein coefficients in a subdivided region are positive, then the Bézier surface patch in that region should lie above the $uv$ plane, which means that no umbilical point exists in the region. Note that the reverse is not necessarily true. The algorithm checks the signs of the Bernstein coefficients in a subdivided region. If all of them are positive, the node of the corresponding region is marked as a non-umbilic region. The algorithm stops at a depth $d_f$ where the size of the regions of the nodes at $d_f$ is less than the user defined tolerance. We traverse the tree and collect the unmarked nodes to produce the regions which possibly contain umbilical points.

It is not known a priori how many nodes will be generated in general. But because we can estimate the maximum depth required to achieve a given accuracy, we can perform the worst case analysis. Given a tolerance $\delta_{umb}$, the algorithm stops when the size of a subdivided domain at a depth $d$, or $2^{-d}$ is less than $\delta_{umb}$. Let us assume that $2^{-d} < \delta_{umb}$ and the algorithm has stopped. Subdivision happens twice at every node up to depth $d-1$. Therefore, the total number of applications of the de Casteljau algorithm becomes $2\sum_{j=0}^{d-1} 4^j$. For a Bézier surface patch of degree $m$ and $n$ in the $u$ and $v$, the application of the de Casteljau algorithm in $u$ parameter requires $O(m^2)$. Hence, the complexity of the worst case is reduced to $O(4^d(m^2 + n^2))$.

The IPP algorithm [74, 91, 106] can also be used for extraction of umbilics. But when the IPP algorithm encounters regions of umbilical points, it slows down dramatically. The quadtree decomposition method is efficient in dealing with lines or regions of umbilical points because it employs an adaptive subdivision scheme and only requires subdivision using the de Casteljau algorithm, whereas the IPP algorithm needs not only subdivision but also calculation of convex hulls and projection onto a hyperplane. However, since it belongs to the subdivision class of methods, the proposed method is unable to differentiate multiple roots [106, 91].

## Examples

An integral bicubic Bézier surface patch [74] is used for extraction of isolated umbilical points. The patch contains five umbilical points as shown in Figure 3-4. Table 3.1 summarizes the exact parametric values of each umbilical point adapted from [91, 74] and the boxes from the subdivision extraction algorithm. The calculation is performed with a tolerance of 0.005. Table 3.1 shows that the estimated boxes contain the exact location of the umbilical points computed via the IPP algorithm solving (2.25) at a higher precision of $10^{-12}$.

|  | Exact Location | Estimation |
|---|---|---|
| No. | $(u, v)$ | $[u_a, v_a] \times [u_b, v_b]$ |
| 1 | $(0.211, 0.052)$ | $[0.2109, 0.0508] \times [0.2148, 0.0547]$ |
| 2 | $(0.211, 0.984)$ | $[0.2109, 0.9807] \times [0.2148, 0.9844]$ |
| 3 | $(0.789, 0.052)$ | $[0.7852, 0.0508] \times [0.7891, 0.0547]$ |
| 4 | $(0.789, 0.984)$ | $[0.7852, 0.9807] \times [0.7891, 0.9844]$ |
| 5 | $(0.500, 0.440)$ | $[0.4961, 0.4375] \times [0.5039, 0.4414]$ |

Table 3.1: Comparison of positions of isolated umbilical points

The next example shows a line of umbilical points on a surface. The input surface is a developable cubic-linear surface adapted from [91, 71]. The surface has an inflection line at $u = 0.5754$ computed via the IPP algorithm with a precision of $10^{-12}$. The proposed algorithm with a tolerance 0.005 produces a series of boxes which contain 0.5754 as follows:

$$\begin{gathered}[0.5742, 0.0] \times [0.5781, 0.0039]\,,\\ [0.5742, 0.0039] \times [0.5781, 0.0078]\,,\\ \vdots\\ [0.5742, 0.9961] \times [0.5781, 1]\,.\end{gathered}$$

Those boxes are mapped onto the surface in a series of small boxes as shown in Figure 3-5.

An example of extraction of a planar region is presented in Figure 3-6. It is a bicubic B-spline surface which is partially planar, with 20 × 20 control points in the

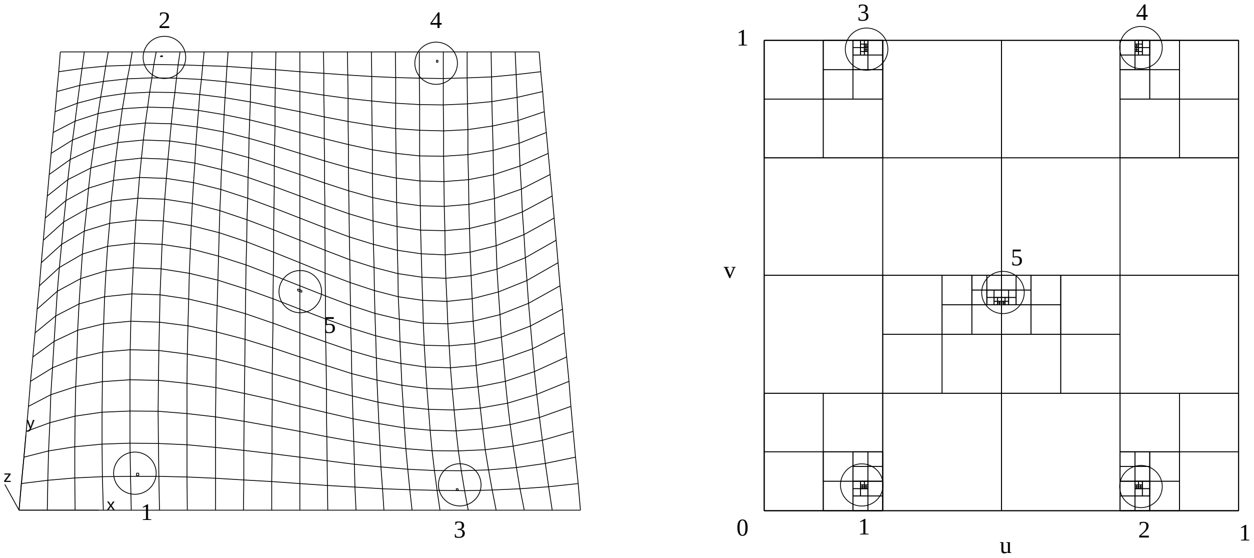


Figure 3-4: An example of isolated umbilical points on *uv* domain and the surface

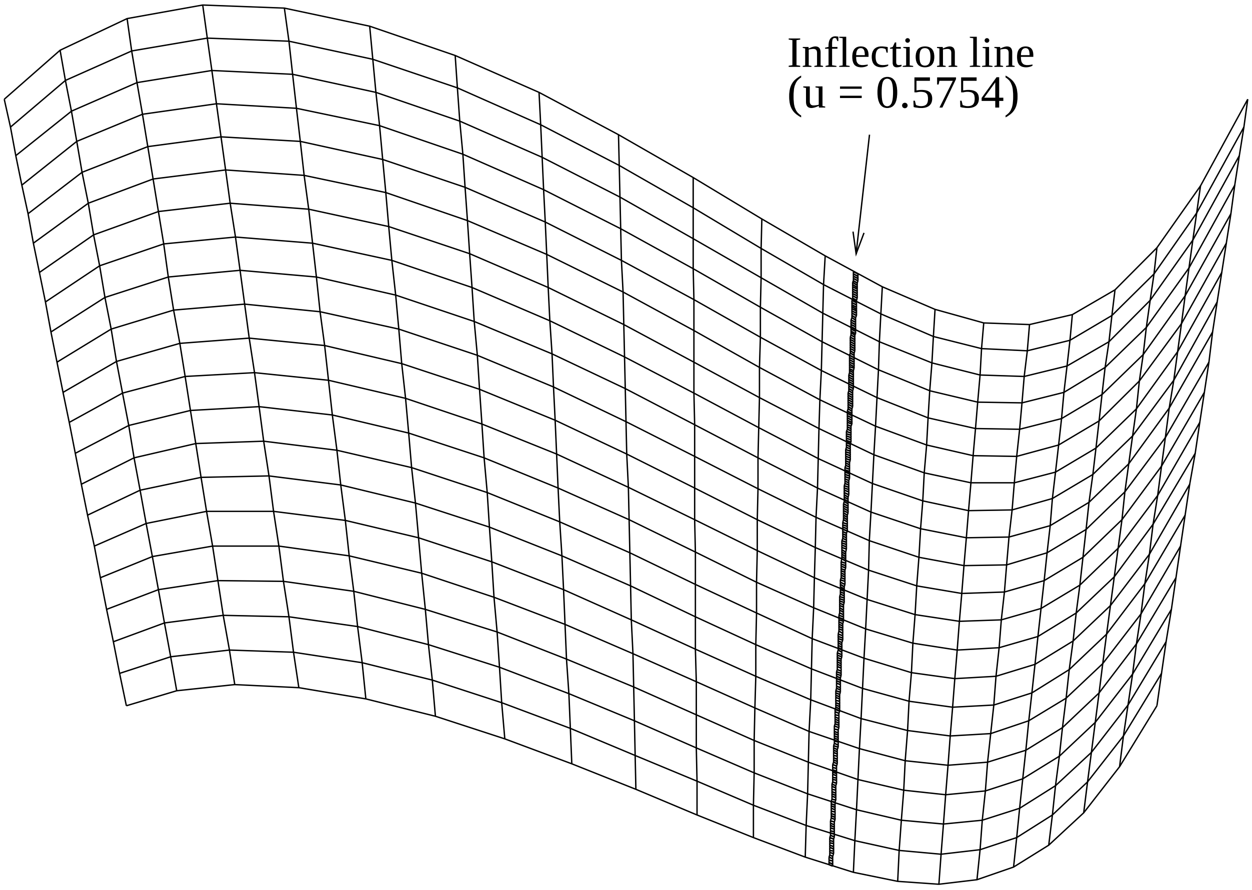


Figure 3-5: An example of a line of umbilical points

$u$ and $v$ directions. The extracted planar region of the surface is shown in Figure 3-6.

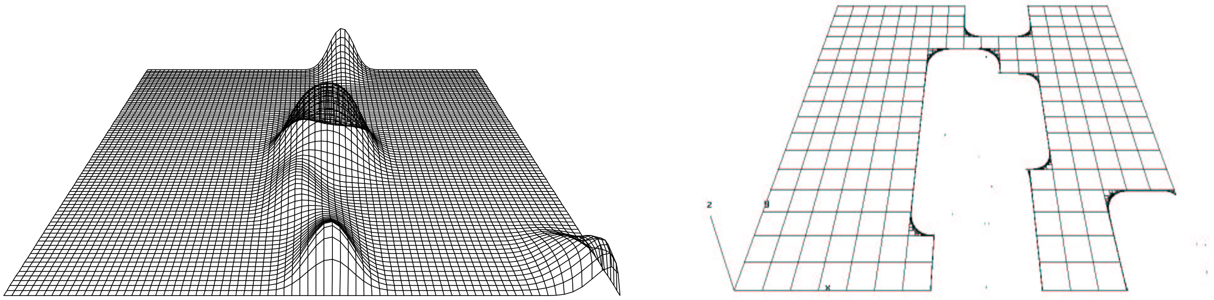


Figure 3-6: Extraction of planar region

## 3.7 Shape Intrinsic Wireframing

In this section, a method for surface-intrinsic-wireframing using lines of curvature and geodesic curves is explained, which will not be affected by parametrization, any rigid body transformation and representation methods. From this wireframe, reference points are selected for comparison and surface matching.

### 3.7.1 Overall Structure

An algorithm for the construction of intrinsic wireframing is shown in Figure 3-7. A NURBS surface is provided as input. As the principal directions are indeterminate at umbilical points, we need to locate such points prior to wireframing. Next a starting point is selected, from which wireframe creation begins. It continues until lines of curvature cannot be traced any longer. The remaining area is covered with geodesic curves. If the entire surface has been wireframed sufficiently densely according to appropriate user-specified thresholds, the algorithm stops.

**Input Surfaces and Umbilics (Steps 10 and 12)**

Information on exact locations of umbilical points is essential for wireframing using lines of curvature because the principal directions cannot be uniquely defined there so that the lines of curvature cannot be traced properly. Therefore, surfaces need to be provided as input to the algorithm with complete information of isolated umbilical points, spherical or planar regions, or lines of umbilical points extracted by using the quadtree decomposition method of Section 3.6.

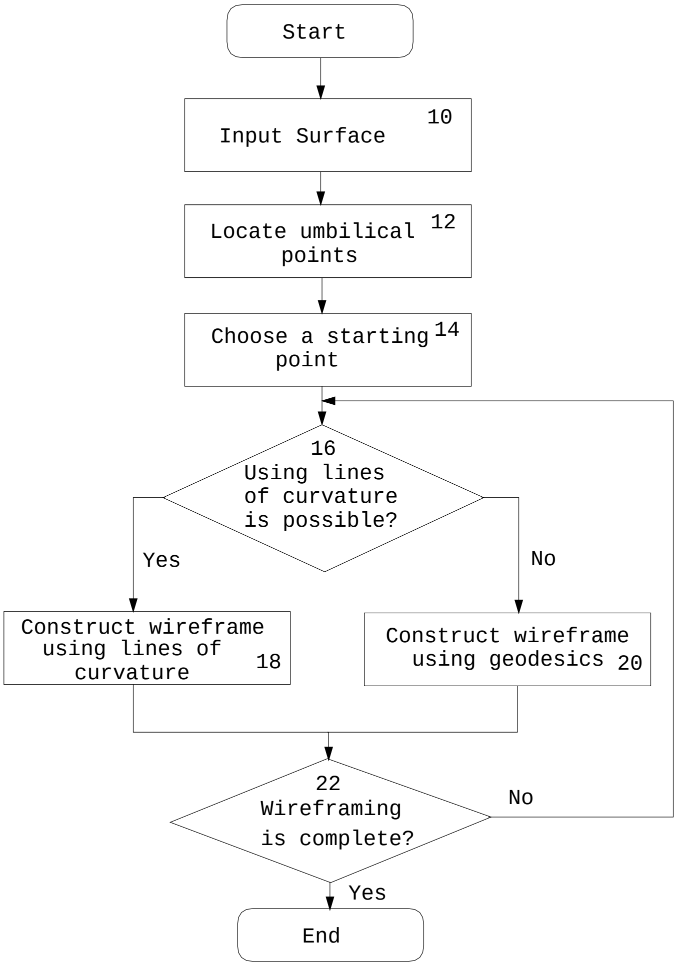


Figure 3-7: A diagram of the algorithm

### Starting Points (Step 14)

Either a star type umbilical point or a non-umbilical point can be chosen as a starting point for wireframing because the maximum and minimum lines of curvature radiate from the point in an alternating pattern so that a simple algorithm is sufficient and the resulting mesh is more well-proportioned. The other two types of umbilics, i.e. lemon and monstar, are not appropriate for this purpose. Three lines of curvature pass through a star umbilical point, each of which changes its attribute from the maximum line of curvature to the minimum line of curvature or *vice versa*. Therefore, at the star type umbilical point, we can imagine that six lines of curvature radiate from the umbilical point so that we can use up to six initial directions for tracing lines of curvature. When there is no umbilical point on the surface, a non-umbilical point is chosen as a starting point. The non-umbilical point has the maximum and minimum lines of curvature intersecting orthogonally. Thus, we can choose up to four directions for tracing the lines of curvature.

### Wireframing with Lines of Curvature (Step 18)

Lines of curvature are calculated by solving equations (2.12) or (2.13) using for example the fourth order Runge-Kutta method.

An intersection point of the maximum and minimum lines of curvature can be calculated accurately using Newton's method. Let us assume that a maximum line of curvature $L_1$ and a minimum line of curvature $L_2$ intersect as shown in Figure 3-8. Using the arc length $s_1$ as a parameter, $L_1$ can be represented as

$$u_1 = u_1(s_1), \quad v_1 = v_1(s_1). \tag{3.46}$$

Similarly, $L_2$ is

$$u_2 = u_2(s_2), \quad v_2 = v_2(s_2), \tag{3.47}$$

where $s_2$ is the arc length parameter. The problem can be stated to find $s_1$ and $s_2$ so that

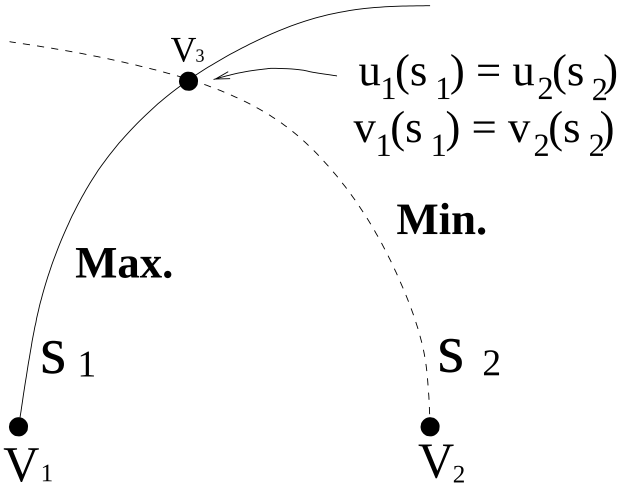


Figure 3-8: Intersection of lines of curvature

$$\mathbf{F}(\mathbf{x}) = \begin{pmatrix} f(s_1, s_2) \\ g(s_1, s_2) \end{pmatrix} = \begin{pmatrix} u_1(s_1) - u_2(s_2) \\ v_1(s_1) - v_2(s_2) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \tag{3.48}$$

where $\mathbf{x} = (s_1, s_2)^T$. The multivariate Newton's method can be adopted as a solution method. Equation (3.48) is Taylor expanded to obtain $\mathbf{F}(\mathbf{x}+\delta\mathbf{x}) = \mathbf{F}(\mathbf{x})+\mathbf{J}\cdot\delta\mathbf{x}$, where $\delta\mathbf{x} = (\delta_1, \delta_2)^T$ and $\mathbf{J}$ is the Jacobian matrix of $\mathbf{F}$ which can be calculated numerically using equations (2.12) or (2.13). Two tolerances $\delta_1$ and $\delta_2$ for the variables $s_1$ and $s_2$ are provided for termination of the iteration of Newton's method.

### Geodesic Wireframing (Step 20)

In a region where the algorithm using lines of curvature fails, i.e. in the neighborhood of an umbilical point (except the umbilical point used as a starting point) and near a boundary or in an umbilical region, a geodesic curve can be used to complete wireframing. A geodesic problem comes in the form of a boundary value or an initial value problem. However, the problem arising in wireframing is always formulated as a boundary value problem (BVP) and a solution method for BVPs is adopted. Two points are selected and a geodesic line is calculated which connects them. As an initial approximation, a straight line connecting two boundary points is used, from which a solution is obtained iteratively by using the *relaxation method*, see [91, 98, 70].

### 3.7.2 Algorithms for Constructing Quadrilateral Meshes

The proposed algorithm is semi-automatic so that in order to complete a wireframe of a free-form surface a user may need to work iteratively. An automatic routine handles wireframing using lines of curvature. After a starting point has been selected, the routine constructs quadrilateral meshes (mostly orthogonal meshes) radiating from the starting point. An outline of this procedure follows:

1. Advance $L_1$ and $L_2$ from $V_0$ by a user defined distance to reach $V_1$ and $V_2$ points as shown in Figure 3-9. Here, we assume that $L_1$ and $L_2$ are the maximum and the minimum lines of curvature, respectively.

2. Choose the minimum line of curvature ($L_1'$) at $V_1$ and the maximum lines of curvature ($L_2'$) at $V_2$.

3. Find an intersection point $V_3$ of $L_1'$ and $L_2'$.

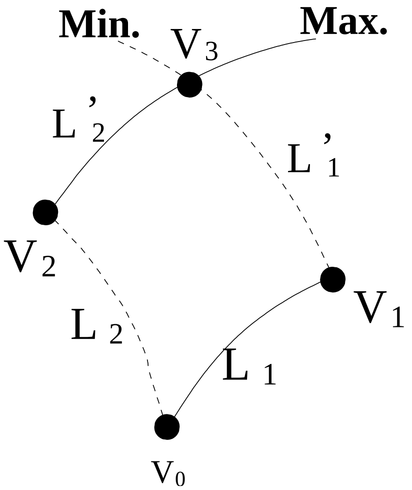


Figure 3-9: A diagram for meshing algorithm

This routine continues until no more intersection point can be obtained, i.e. when lines of curvature hit either boundary or umbilical points. In a region that lines of curvature cannot cover, a user inserts additional nodes and connects them using geodesic curves. Figure 3-10 shows an example of intrinsic wireframing. This example has 413 nodes, 356 quadrilateral and 14 triangular elements.

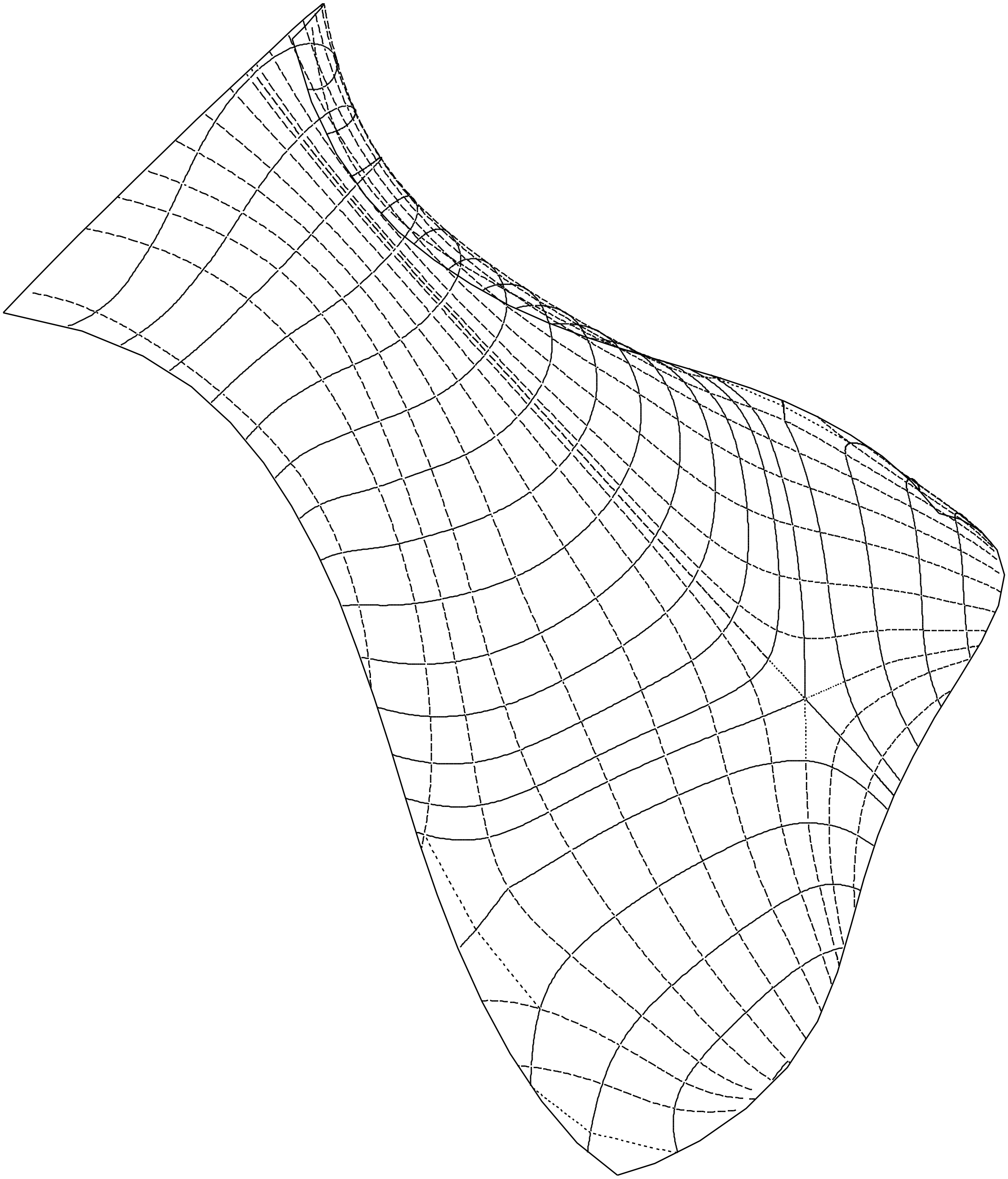


Figure 3-10: Intrinsic wireframe

### 3.7.3 Implementation

A computer program was developed to integrate all algorithms to generate wireframe of a NURBS surface. It consists of two windows. One is to visualize a surface and wireframe in 3D space, and the other, which is denoted as the control window, is to provide an interface for a user to interface with the wireframe when user input is needed. The visualization window is created based on OpenGL. It provides functions for rotation and scaling for better visualization. The control window has a parametric domain. It shows the current status of the parametric domain during wireframing. Several options are provided for interactive operation. The language used in the development is C++. A set of window managing libraries, Qt-2.3.1, are used for graphical user interface. Figures 3-11 and 3-12 show intermediate results of our algorithms.

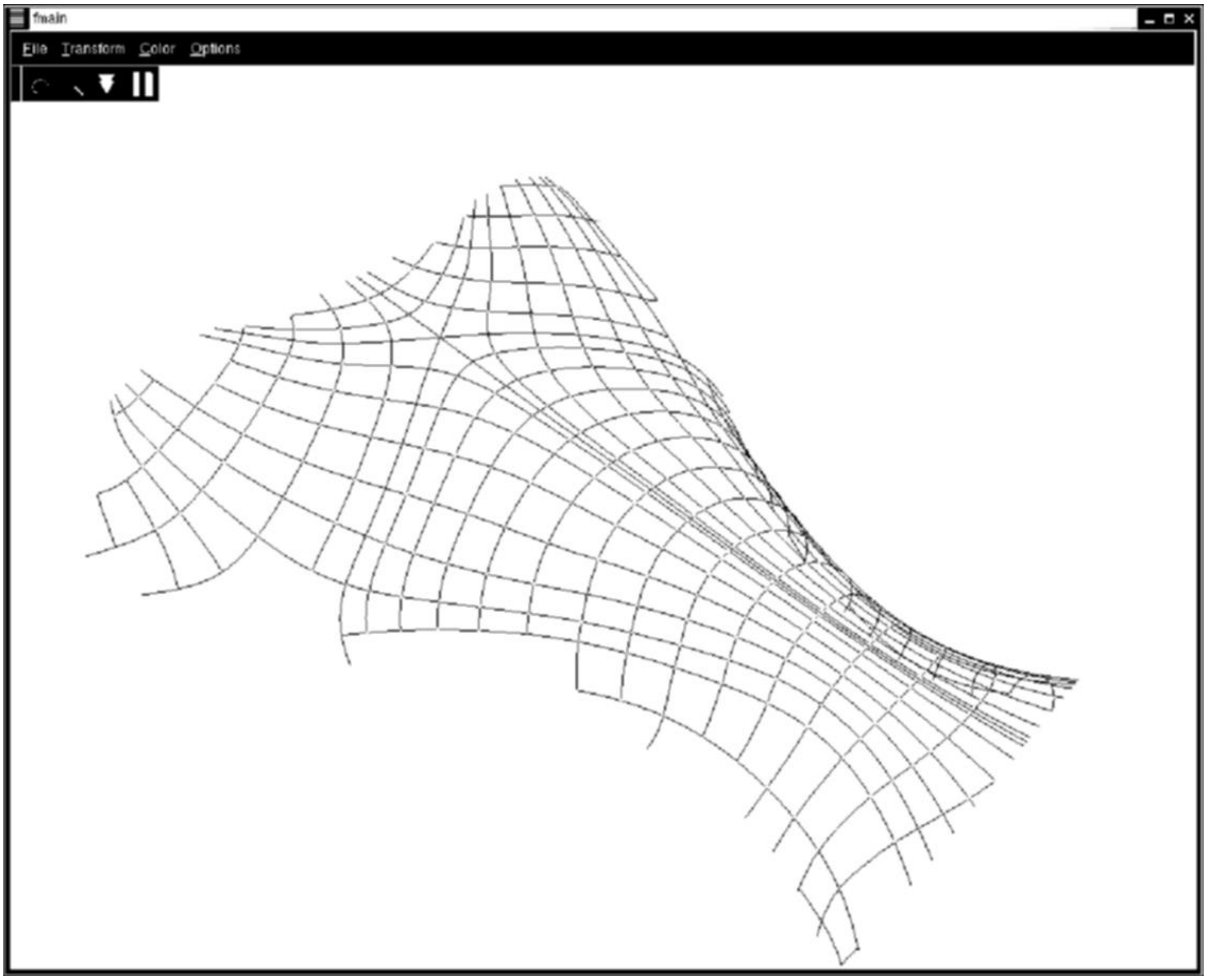
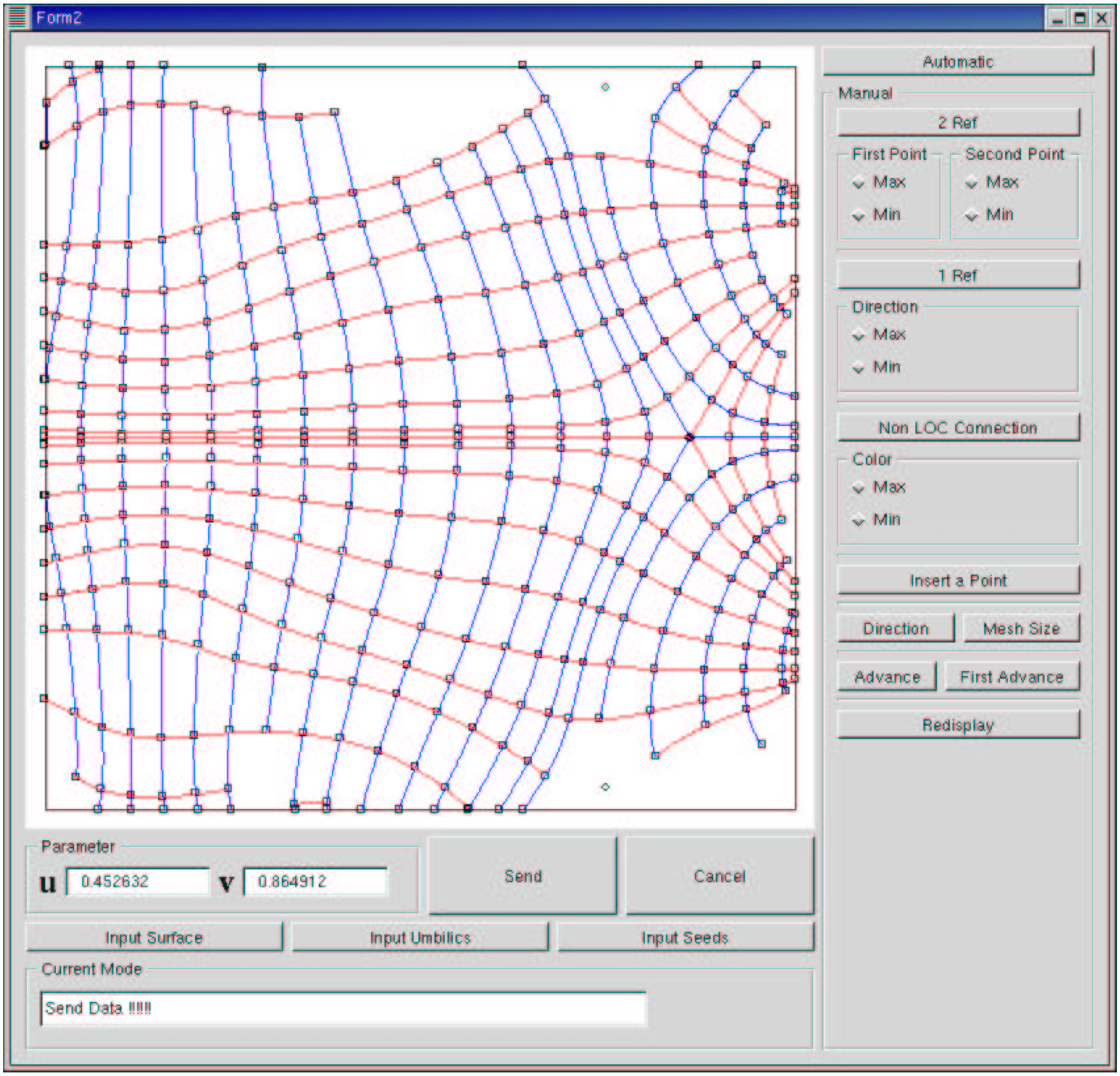


Figure 3-11: A visual window for wireframing



Parameter

**u** 0.452632 **v** 0.864912

Current Mode

Send Data !!!!!

Figure 3-12: A control window for wireframing

### 3.7.4 Analysis of the Algorithm

The overall time complexity of the automatic part of the wireframing algorithm is hard to determine because the total number of nodes that will be generated cannot be predicted in advance before execution. The only parameter which can control the density of the node points is the distance that the meshing algorithm advances from a starting point. The mesh creating algorithm consists of advancing lines of curvature and calculating a node which is obtained by locating the intersection point between a maximum and a minimum lines of curvature. The accuracy and elapsed time of advancing lines of curvature depends on the step size of the numerical integration routine such as Runge-Kutta method. Finding the intersection point is based on Newton's method so that a quadratic convergence is achieved.

## 3.8 Interval Projected Polyhedron Algorithm

### 3.8.1 Robustness in Numerical Computation

*Floating point arithmetic (FPA)* is commonly used in current CAD systems. However, arithmetic operations in FPA such as division, sometimes, result in serious errors during the computation. This numerical instability may cause a change of topological structure of a model or miss of roots in solving a system of equations. Such problems mainly arise due to the limited precision inherent to the internal representation of a floating point in digital form in the computer.

Most of commercial computer systems adopt ANSI/IEEE Std 754-1985 Standard for *Binary Floating Point Arithmetic* to represent a floating point value [4]. Due to the nature of digitization of the representation, some floating points cannot be represented exactly in the system. The floating point numbers that can be represented exactly are finite and distributed non-uniformly. Therefore, rounding methods defined in IEEE-754 have to be used to approximate the values that cannot be represented exactly, which means that the errors introduced in the rounding may propagate during the calculation so that the final results may end up containing significant errors.

To avoid such problems, all calculations can be performed in rational arithmetic. But it is memory intensive and time consuming [91]. On the other hand, one can use a special purpose library such as *GNU multiple precision arithmetic* [44] to overcome the limitation of precision. However, it also poses similar problems as those of rational arithmetic.

A different approach to get around the problems of floating point arithmetic is to use *rounded interval arithmetic (RIA)*. An interval is a representation unit which encloses an exact floating point value. Even though an exact value may not be obtained in rounded interval arithmetic, rounded interval arithmetic guarantees that an interval contains an exact floating point value during the calculation, and operations which are prone to errors such as division, can be performed robustly. The actual execution load for rounded interval arithmetic lies between rational arithmetic and floating point arithmetic.

### 3.8.2 Brief Review of Interval Projected Polyhedron Algorithm

The basic idea of the proposed work in Chapter 4 is to cast the matching problem to the solution of a nonlinear system of equations. Therefore, robust detection of all roots of the system is critical for the robustness of the proposed matching methods. A robust and efficient method, the *interval projected polyhedron (IPP)* technique is used in this work [106, 91, 72, 2].

The algorithm belongs to the subdivision-based class of techniques and makes use of the Bernstein polynomial properties: the convex hull property, the linear precision property and subdivision to narrow down the regions containing the real roots of systems of a nonlinear polynomial equation system. The key aspect of this algorithm is to change the algebraic problem of finding roots into the geometric problem of finding intersections between the axes and the hypersurfaces. A detailed description of the IPP algorithm can be found in [106, 72, 91] together with various applications.

In order to avoid the adverse effects of numerical error in floating point arithmetic, rounded interval arithmetic is employed for the implementation of the IPP algorithm.

Suppose that we have a set of $n$ nonlinear polynomials $f_1, f_2, \cdots, f_n$ and each $f_i$ is a function of $l$ independent variables $x_1, x_2, \cdots, x_l$ and degree $m_i$ for each variable. The problem is to find all points $\mathbf{x} = (x_1, x_2, \cdots, x_l)$ such that

$$f_1(\mathbf{x}) = f_2(\mathbf{x}) = \cdots = f_n(\mathbf{x}) = 0. \tag{3.49}$$

The total asymptotic computation time per step is of $O(nlm^{l+1})$ [106, 91], where $m = max(m_i)$. In this work, the system of equations (4.6) is used and during the formulation, the auxiliary variable method is also used to remove the square root terms. So the system becomes three dimensional ($n = 3$) and the number of independent variables is three ($l = 3$), i.e. $u$, $v$ and one auxiliary variable $n_d$. Therefore, the time complexity per step of the algorithm becomes $O(m^4)$.

## 3.9 Conclusions

In this chapter, robust numerical calculation methods for lines of curvature, geodesic curves, umbilical points and orthogonal projection are developed. Those algorithms are incorporated into the shape intrinsic wireframing algorithm to create shape intrinsic wireframe of a NURBS surface. Such a wireframe is independent of parametrization and the rigid body transformation, and they can be used as a unique representation of a given surface combined with the patterns of isolated generic umbilical points which are stable against small perturbations and behave like fingerprints of a surface.

The wireframe is used to select reference points for comparison. Since it is based on the shape intrinsic properties, the same reference points can be obtained all the times irrespective of parametrization and the rigid body transformation.

# Chapter 4

# Object Matching

A basic goal of matching is to find *the best* rigid body transformation which aligns two objects as closely as possible. The correspondence search between two surfaces is a key issue in finding the best transformation for matching. Correspondence can be established by calculating distinct features of one object and locating the same ones on the other object. Therefore, the features have to be carefully chosen such that they are robustly extracted and invariant with respect to geometric transformations. Among various features, intrinsic differential properties have been used for matching purposes. They are independent of parametrization and methods of representation, and only depend on the geometric shape of the object. Moreover, they are invariant under any rigid body transformations (rotation and translation).

Two types of matching can be considered: *global* and *partial*. Simply, the global matching is regarded as the matching for whole objects, while the partial matching is considered as the matching of part of objects. Matching problems can be further categorized based on the availability of correspondence or initial transformation information between two objects and the application of scaling. The classification of matching problems is summarized in Table 4.1. In the table, acronyms are used for simplification as follows:

- C : Correspondence information is provided.
- I : Initial information on correspondence is provided.
- N : No correspondence information is available.
- P : Partial matching.
- G : Global matching.
- WOS : Without scaling.
- WS : With scaling.

When correspondence information is provided, which is one of the types CGWOS or CPWOS, then a matching problem simply is reduced to calculation of the rigid body transformation [48, 49]. If no correspondence is known, but a good initial

| Criteria | Global Matching | | Partial Matching | |
|---|---|---|---|---|
| | Without Scaling | With Scaling | Without Scaling | With Scaling |
| Correspondence Information | CGWOS | CGWS | CPWOS | CPWS |
| Initial Information | IGWOS | IGWS | IPWOS | IPWS |
| No Information | NGWOS | NGWS | NPWOS | NPWS |

Table 4.1: Classification of matching problems

approximation for the transformation is available (IGWOS or IPWOS), then popular iterative schemes such as the *Iterative Closest Point (ICP)* algorithm [10] can be employed. However, when no prior clue for correspondence or transformation is given (NGWOS or NPWOS), the matching problem becomes more complicated. In this case, the solution process must provide a means to establish such correspondence information such as in [25].

Scaling is another factor that needs to be considered separately. If a matching problem involves scaling effects, then direct comparison of quantitative measures cannot be used any longer. For the global matching case, a scaling factor can be estimated by the ratio of surface areas and applied to resolve the scaling transformation. However, when it comes to partial matching, such area information becomes useless for the scaling factor estimation. When the correspondence information between two objects is known (CGWS or CPWS), the scaling factor between the objects can be easily obtained by using the ratio of Euclidean distances between two sets of corresponding points or areas. If an initial scaling value, as well as a good initial approximation is provided (IGWS or IPWS), the ICP algorithm by Besl [10] or other optimization schemes such as the *quasi-Newton method* [98] can be employed. The problem of NGWS type can be solved by the moment method using the principal moment of inertia and ratio of areas or volumes. No attempt, however, has been made to solve the problems of NPWS type.

In this chapter, algorithms are proposed to solve all types of matching problems in Table 4.1 with emphasis on the matching problem of NPWS type. Algorithms for establishing correspondences are proposed, which do not include scaling effects. They are extended to deal with scaling effects. Similarity is evaluated by tests using the Euclidean distance, principal curvatures and directions, and umbilical points, which are provided as input to decision algorithms described in Chapter 5.

This chapter is constructed as follows: in Section 4.1, previous work on matching is classified and reviewed, and mathematical problem statements used in this thesis are presented in Section 4.2. Section 4.3 briefly introduces surface fitting methods followed by three matching criteria used for similarity evaluation in Section 4.4. In Section 4.5, a global method, the moment method, is explained which is used for matching solids. In Section 4.6, two methods for establishing correspondences between two objects are proposed followed by two algorithms and their theoretical analysis, which include uniform scaling effects in Section 4.7. Section 4.8 discusses performance issues involved in the algorithms and Section 4.9 concludes this chapter.

## 4.1 Literature Review

### 4.1.1 Moment Theory

Moment theories for 2D or 3D objects have been extensively employed for object matching, recognition, classification, etc. Both invariant and non-invariant features of moments of an object have been of primary interest to researchers, and there is extensive literature on this topic.

First order moments (center of mass) and second order moments (moments of inertia) can be used for pose estimation. Novotni and Klein [83] used geometric moments up to second order to bring inspected objects into a standard pose. They resolved ambiguity of aligning principal directions with a reference coordinate system by voxelizing the objects and choosing one of all the possible directions which gives the minimum difference of volume between the two objects (XOR (eXclusive OR) operation between the volumes occupied by the objects which produces the voxels which belong to each object, but not both). Galvez and Canton [43] used the center of mass and moments of inertia of a 3D shape for shape recognition. A heuristic procedure to avoid the ambiguity in matching principal axes for a 3D object was presented. They calculated the distance from the centroid to the point determined by the intersection of each coordinate axis with the bounding object surface and orient the eigenvector based on the distance values.

The method of principal axes, however, is not robust for rotationally symmetric objects, i.e. objects with no unique set of principal axes. Many efforts have been made to develop a technique to achieve the moment invariants with respect to not only scale and position but also rotation. The first significant contribution was made by Hu [51], who proposed seven moment invariants for 2D images based on the invariant algebra theory. The concept of 2D moment invariants was extended to 3D by Sadjadi and Hall [100].

Several attempts were made to improve Hu's work. Some fast algorithms were proposed for computing moment invariants by Li *et al.* [67] and Zakaria *et al.* [122]. Li *et al.* used an iterative scheme for moment calculation which needs no multiplication and Zakaria *et al.* presented *delta method* for fast invariant moment computation of contiguous images. Modified moment invariants were proposed by Chen [22], which involve the calculation along the shape boundary to reduce the amount of computation. Lia and Pawlak [68] carried out the discretization error analysis for moment computation and proposed a new technique to improve the accuracy and efficiency of moment calculation. They also examined the inverse moment problem of reconstruction of an image.

Various moment invariants and use of orthogonal moments such as Zernike moments and Legendre moments have been investigated. Prokop and Reeves [99] performed a comprehensive survey on moment-based techniques for object representation and recognition. They reviewed fundamental theories on moments and assessed various moment-based approaches.

### 4.1.2 Principal Component Analysis

The *principal component analysis* known as the *Karhunen-Loève transformation* [42] is based on the orthogonal set of eigenvectors of an image set which allows a compact representation of an object for various purposes such as object recognition, pose estimation, image compression, etc. One or more multidimensional spaces are constructed based on the eigenvectors which are obtained from a statistical analysis of the ensemble of training images. An object is projected onto the eigenspace and stored as a vector with respect to the eigenspace. By comparing model and image vectors the best matched object is obtained in the database. For more detailed information, see [20]. The eigenspace approach is popular in recognition of human faces [118, 59], and much research for its application to 3D object recognition has been performed up to now. Murase and Nayar [81] proposed a method for recognition of 3D objects, which is based on the eigenspace. Two eigenspaces are used: one is the *universal eigenspace* computed by using all different kinds of training images of interest for recognition to identify an object in the input scene, while the other is the *object eigenspace* obtained from various images of an object for pose estimation. Each of the training images is projected to eigenspace so that each object falls on a *manifold* parametrized by pose and illumination (called *parametric eigenspace*). Recognition of an object is done by checking on which manifold the object is placed, and using the cubic interpolation on the manifold, the object's pose can be determined. Murase and Nayar [80] also used the same parametric eigenspace approach to determine the illumination condition which enhances differentiability in appearance and Nene and Nayar [82] improved performance of the nearest neighbor search in high dimension, which is used in finding the closest vector in eigenspace. The parametric eigenspace for object recognition by Murase and Nayar has, however, potential problems with background clutter and partial occlusions. These problems were discussed by Krumm [63], who proposed an algorithm that uses several small rectangular image patches which constitute features of each object, rather than one large patch. The parametric eigenspace idea is applied to the features and they are matched using a median distance measure. The *Eigen Window* method which is similar to Krumm's approach was proposed by Ohba and Ikeuchi [84] for recognition of partially occluded objects. Small windows which contain multiple partial appearance are applied to the original images and projected onto the eigenspace. The selection of the optimal set of eigen windows is discussed based on three measures: detectability, reliability and uniqueness to reduce memory requirements. The occlusion problem was also discussed by Leonardis and Bischof [65] which was extended in [66]. A novel approach was adopted for calculation of the coefficients of eigenimages to obtain robustness in coefficient calculation, which leads to cope with the problems arising in occlusion and segmentation. A number of trials that lead to a set of hypotheses with respect to both error and compatibility of points are made by randomly selecting image points from the scene (hypothesize-and-test paradigm) and a *good* hypothesis is chosen based on the *Minimum Description Length Principle*. This method demonstrates the ability to reconstruct unseen portions of the object. In [11], the application of Leonardis and Bischof's method [65] to large and scaled images through a multi resolution approach was discussed, and problems for missing pixels in the image were dealt with by Skocaj and Leonardis [110]. Black and Jepson [13] proposed another robust estimation techniques of the coefficients in the eigenspace. A robust error norm is used for estimation of the coefficients in the eigenspace method to reduce large residual errors which undermines traditional eigenspace methods. Campbell and Flynn [19] extended the eigenspace recognition approach to deal with 3D objects in range images. A set of *eigensurfaces* are created which capture the shapes of objects in various orientations. Recognition of an object is done by finding the nearest vector in the eigenspace. 3D rotational pose estimation is discussed in the paper. Huang *et al.* [52] used the eigenspace approach to represent parts of an object and their relations. The parts are collected based on the *minimum description length* principle (MDL) and each part is encoded by constructing parametrized manifolds in eigenspaces. Additionally, using more than one part establishes eigenspaces representing spatial relations. This representation is robust to occlusion. Camps *et al.* [21] extended their previous work [52] by enhancing the discriminatory power and the method of organization of large databases.

### 4.1.3 Contour and Silhouette Matching

2D contours which are extracted from 3D objects have been used for recognition. Mokhtarian [78] developed an object recognition system based on the silhouettes of the 3D object using *Curvature Scale Space (CSS)*. The physical boundaries are obtained from the silhouette images and then classified as either convex or concave from which CSS representations are computed. CSS maxima are used to find the best matching silhouettes which are verified by registering them and measuring errors. Josh *et al.* [55] proposed a *HOT curve* representation of smooth curved 3D shapes. HOT curves (where High Order Tangential contact with a surface exist) are a non parametric, discrete representation which determines the structure of image contours and its catastrophic changes. Five categories of HOT curves are maintained such that their position, direction of the surface normal and direction of the tangent are recorded. The scale independent parameters coming from angles between the tangents line and the ratios of distances between features are estimated and used to index a recognition table. Chen and Stockman [23] proposed a method to recognize a free-form 3D object from a single 2D view of a 3D scene. Edge features are extracted for generating and verifying hypotheses from the silhouette curves in the intensity images, and part invariant attributes are used for indexing against the model database, which leads to matches between possible model parts and observed parts through the test of hypotheses. More accurate estimation is obtained by using internal edges and silhouettes.

### 4.1.4 New Representation Scheme

Delingette *et al.* [31, 47] proposed a new surface representation scheme called *spherical attribute image (SAI)* for recognition of 3D objects. They pointed out the drawbacks of the earlier spherical representation such as EGI and CEGI [58, 57], and developed a new approach combining the point set matching and the original EGI approach.

An initial mesh under the local regularity condition is defined to represent the input object as closely as possible by deforming a standard shaped mesh under the guidance of forces which make the constraints satisfied. Meshes are created on the unit sphere which have the same number of nodes as the meshes on the object, and a canonical map is defined between the two mesh representations. A new curvature measure, the *simplex angle* is calculated and stored at every node of the mesh, which is invariant in terms of rotation, translation and scaling. Matching and recognition is achieved by comparing the spherical model of a reference object with that of the input object. Shum *et al.* [107] dealt with the problem of similarity between two 3D surfaces and proposed a shape similarity metric based on spherical attribute images. The similarity measure is also used for multi-scale classification of 3D objects by Zhang and Hebert [123] which can be applied to hierarchical recognition scheme.

Stein and Medioni [111] proposed a method for matching which uses two different types of primitives: the 3D curves from boundary edges and differential properties called the *splash* in smooth areas. The 3D curves are represented based on a polygonal approximation, and curvature and torsion angles are estimated together with the number of segments of the polygon characterize the *3D super segment*. For smooth areas, splashes are calculated. A reference normal is determined at a point on the surface and points are sampled along a circular slice around the reference normal with a pre-defined geodesic radius and surface normals are calculated at the sampled points, which comprises a *super splash* with a set of geodesic radii. The splashes are mapped to periodical polygonal curves based on spherical coordinates which are encoded as 3D super segments. A hash table combined with indexing scheme is used for efficient storage and access in the database which records all the features from which hypotheses for matching are retrieved. A set of hypotheses are used for pose and recognition and a least squares match on all corresponding features is used for the precise pose estimation.

Several new representation schemes for 3D object recognition have been used in matching. Not only the shape of an object but also additional information which is invariant to various operations such as rotation, translation, etc. is incorporated to represent the object for recognition purposes. A new representation scheme for recognition of 3D free-form objects, called *COSMOS*, was proposed by Dorai and Jain [34, 35, 36]. An object is represented by maximally sized surface patches each of which has constant *shape index* characterizing local shapes of the object. The areas of the surface of each shape index value are represented as a histogram, called *shape spectrum* which gives a distribution of shape components of the surface on the object. *Curvedness* is used to capture the scale differences between objects which shows the amount of curvatures in a region. Connectivity information is maintained for the relative spatial arrangement of the surface patches along with the orientations of them using *the Gauss Patch Map* and *the Surface Connectivity List*. A graph based multi-level matching scheme based on shape spectral analysis and COSMOS representation was studied by Dorai and Jain [35]. Campbell and Flynn [18] discussed the use of curvatures as a descriptive feature for recognition for polygonal mesh approximation and range data. The location of curvature extrema for polygonal representation is discussed. They also proposed a curvature/edge detector which uses a hysteresis-based approach and non-maxima suppression as an enhancement to the hysteresis method and showed that the use of different methods and measures to estimate surface curvature depend on the type of surface polygonization. Chua and Jarvis [26] used additional one dimensional information as well as data points on the surface. The information, called *the point signature* is calculated at a point, which describes the local surface structure in the vicinity of the point. A point signature at a point $p$ is obtained by the distance profile of the intersection curve of the sphere whose center is the point $p$ with the object surface against a plane determined by the intersection curve and translated to the point $p$. The matching procedure is easily done by comparing the one-dimensional point signatures. One the other hand, Johnson and Hebert [53] proposed a new representation scheme based on *spin images*. An *oriented point* defined as a 3D vertex position with a surface normal on an object defines a 2D basis i.e. a local coordinate system, around which a function that maps 3D points to the 2D coordinates of the 2D basis is evaluated. The function produces a 2D histogram of the surface, which is called the *spin image*. It contains the relative position of 3D points on the surface so that it describes the shape of the object. Comparing spin images is done with linear correlation coefficients from which a point correspondence is established. Several point correspondences are then used to estimate the transformation for matching and verification.

### 4.1.5 Matching Through Localization/Registration

Object recognition can be accomplished by selecting the candidate model in the database which gives the minimum distance metric with the object in question. This procedure needs to be preceded by object alignment i.e. registration. Registration of images has previously been studied by various researchers [16]. Finding corresponding features or points between two sets of objects is a critical step to estimate the transformation for registration. A solution scheme for establishing correspondence depends on the availability of initial correspondence information. If such correspondence is known, then the calculation of the transformation is easily obtained [5, 48, 49, 119]. On the other hand, with an initial transformation or initial correspondence information, an optimum transformation can be found through iteration by minimizing a mean square distance metric object function which involves six degrees of freedom for registration (three translations and three rotations). Besl and McKay [10] used *the Iterative Closest Point* (*ICP*) algorithm for registration of 3D shapes. It updates the correspondence between the objects based on the closest distance at each iteration and iteratively converges to a local minimum. The ICP algorithm is generalized to include Euclidean invariant features to provide a more effective algorithm by Sharp *et al.* [105]. A similar idea was proposed by Zhang [124]. He dealt with outliers, occlusion, appearance and disappearance using a statistical method based on distance distribution. Tucker and Kurfess [116] used Newton's method to solve an optimization problem of registration. They derived analytical expressions for Jacobian and Hessian which are used in Newton's method to improve efficiency and accuracy of the registration for parametric surfaces. Patrikalakis and Bardis [90] provided an efficient method for accurate localization of free-form rational B-spline surfaces given an initial estimate of the localization parameters. This iteration approach, however, cannot guarantee to produce an optimum result in the global sense and it requires a good initial state to find a global minimum which is not easily obtained. When no prior clue for correspondence or transformation is available, the localization problem becomes more difficult. In this case, correspondence information between objects must be established to estimate the transformation. A few attempts have been made to solve a problem of this kind. Bergevin *et al.* [7] proposed a method to estimate the 3D rigid transformation between two range views of a complex object. They used a hierarchical surface triangulation representation through an iterative process. Some pairs of matching triangles in the overlap between two views are calculated. From the pairs a set of hypothetical rigid transformations can be obtained. Additional restrictions are imposed to reduce the number of initial estimations. They used the iterative least-square computation of an incremental transformation proposed by Chen and Medioni [24] to find an optimum transformation based on the estimated hypothetical transformations. Chua and Jarvis [25] developed a method to align two objects through registration assuming no prior knowledge of correspondence between two range data sets. They use a bi-quadratic polynomial to fit data points in the local area in the least squares sense and calculate the principal curvatures and Darboux frames. Then, they construct a list of sensed data points based on the fit error. Three seed points are selected to form the list such that the area of the triangle represented by the seed points becomes maximized to reduce any mismatch error. Three constraints (curvature, distance and direction) are imposed to sort out possible corresponding points out of the model data set. Then a list of transformations can be obtained from the candidate points and an optimum transformation is selected. Various searching algorithms are described and demonstrated in [25].

### 4.1.6 Miscellaneous Approaches

Elber and Kim [39] formulated intrinsic conditions to recognize specific shapes. They developed scalar fields based on symbolic computation for shape recognition such as collinearity, circularity, etc. For practical purposes, point sampling at finite, well-selected locations is suggested to determine the shape of a curve or a surface. Point sampling is also the basis of Piegl and Tiller [94]'s approach for shape recognition. They first detected special shapes such as circles, planes, spheres, etc and used them to get offset curves or surfaces. Surazhsky and Elber [114] proposed a matching algorithm by using the unit normal fields of the two surfaces. The similarity of the normal fields is the criterion for matching, which is formulated as an optimization problem. The algorithm, however, deals with simple surfaces that are topologically equivalent to a disk and only local solutions are discussed. Chuang [27] proposed a potential-based approach for 2D shape matching and recognition and Chuang *et al.* [28] extended it to three dimensions using a generalized potential model. The matching process involves the minimization of a scalar function, the potential which depends on shape and interaction between two objects. One object is translated and rotated so that the potential is minimized, which is considered as the best match. Novotni and Klein [83] proposed a geometric comparison of 3D objects by using 3D discrete distance fields. They matched two objects by first aligning the principal moments and constructed the distance histograms from the distance fields which measure geometric similarity of the objects. Osada *et al.* [88] proposed the idea to reduce the shape matching problem to the comparison of probability distributions which can be used as a signature of an object. Different *shape functions* representing global geometric properties are selected to produce the signatures of the objects and the similarity between the distributions is investigated. An issue of approximate equality of two B-rep models of solids is discussed in Sakkalis *et al.* [101]. Motivated by the robustness in representation of solids, they provided sufficient conditions that an *interval solid* represented using a finite collection of boxes is homeomorphic to the solid for a sufficiently small resolution. An outline of construction of interval solid based on an ISI algorithm is presented and it is shown that an interval solid constructed by the algorithm based on interval arithmetic is an equivalent representation of the exact solid. The theoretical results are applied to verification of geometric consistency and rectification of B-rep solids. In addition, they provide mathematical properties which are readily applicable to surface or B-rep solid matching problem.

Most of matching or recognition literature is dealing with objects in range data or polygonal mesh representation. This may be justified because the input device such as the laser scanner usually produces range data or a cloud of points, which also can be converted to polygonal meshes or STL's [1], so that matching or recognition of objects in such formats becomes a main interest and is directed to use such representation formats. However, the discretization loses some local surface information, which brings up the difficulty in extracting surface intrinsic information useful for matching or recognition purposes. Especially, the errors introduced in the course of digitization due to the nature of input devices, may deteriorate the situation.

The moment based approach is simple and useful for object matching and recognition, but unfortunately, its sensitivity to noise remains a major shortcoming of the approach. Especially, when the moment invariants involve coefficients of higher order, the sensitivity to noise is intensified so that its effect becomes dominant. The invariant properties of the moment based approach are well preserved for noise free objects. But they become unreliable when they are dealing with objects with noise. Principal component analysis has a similar problem. If noise is not removed before the analysis, then only the components of low frequency are used for matching so as to minimize the effect of the noise, which may undermine the ability to discriminate between objects with relatively small differences in shape.

Matching through localization may require segmentation as a preliminary step, which is also difficult and prone to errors. Without such segmentation, the localization approach fails to recognize an object when more than two objects are involved. There are a few iterative registration methods which do not require segmentation before matching. They, however, suffer from the weaknesses that the optimum transformation is not guaranteed and the result heavily depends on the initial value of the transformation.

Recognition from boundary curves or silhouettes is useful as long as the boundary curves or silhouettes are robustly extracted. Also they may lack overall shape information which can lead to false matching. Internal edges as well as silhouettes are used for more accurate recognition [23].

Some global representation methods for recognition such as SAI and EGI are proposed. They only can be applied to objects which are topologically equivalent to the sphere because a new mesh representation is generated from the unit sphere through deformation that satisfies some constraint conditions [57, 58, 88].

Local descriptors such as the splash and super segments or point signatures are used for matching and recognition purposes. They are able to capture the local surface information. However, extraction of such local properties is sensitive to noise and the quality of meshes of the local surface.

## 4.2 Problem Statement

### 4.2.1 Matching Objects

Two matching cases are considered in this chapter: the point vs. NURBS surface and the NURBS surface vs. NURBS surface. The base object has to be a NURBS surface, and the objects that are transformed to match the base surface can be represented in range data or mathematical surfaces such as NURBS surfaces and implicit surfaces. Since the surface vs. surface case can be regarded as a special case of the point vs. surface case, the point vs. surface case is mainly considered in this chapter. When solids are provided, their bounding surfaces are considered for matching and similarity evaluation.

### 4.2.2 Distance Metric

The Euclidean distance between two points $\mathbf{p}_1$ and $\mathbf{p}_2$ is defined as

$$d_e(\mathbf{p}_1, \mathbf{p}_2) = |\mathbf{p}_1 - \mathbf{p}_2|. \tag{4.1}$$

We also define the minimum distance between a surface $\mathbf{r}$ and a point $\mathbf{p}$ as follows:

$$d_{sp}(\mathbf{r}, \mathbf{p}) = min\{d_e(\mathbf{p}_i, \mathbf{p}), \forall \mathbf{p}_i \in \mathbf{r}\}. \tag{4.2}$$

### 4.2.3 Distance between a Point and a Parametric Surface

Let us assume that we have a point $\mathbf{p}$ in the 3D space and a parametric surface $\mathbf{r} = \mathbf{r}(u, v)$, $0 \le u, v \le 1$. Then the squared distance function is defined as follows:

$$\begin{aligned} D(u,v) &= |\mathbf{p} - \mathbf{r}(u,v)|^2, \\ &= (\mathbf{p} - \mathbf{r}(u,v)) \cdot (\mathbf{p} - \mathbf{r}(u,v)). \end{aligned} \tag{4.3}$$

Finding the minimum distance between $\mathbf{p}$ and $\mathbf{r}$ is reduced to minimizing (4.3) within the square $0 \le u, v \le 1$. Therefore, the problem needs to be broken up into several sub-problems which consider the behavior of the distance function at the boundary and in the interior of the bound [91]. The sub-problems are summarized as follows:

Find the minimum distances (1) in the interior domain, (2) along the boundary curves and (3) from the corner points. Among those minimum distances, the smallest one is chosen as the minimum distance between the point $\mathbf{p}$ and the surface $\mathbf{r}$. A robust calculation of the minima of the distance function (4.3) can be achieved by the *Interval Projected Polyhedron (IPP)* algorithm [106, 91, 125].

### 4.2.4 Distance Metric Function

Suppose that we have a NURBS surface $\mathbf{r}_B$ and an object $\mathbf{r}_A$ represented in discrete points or surfaces. Then, the matching problem can be stated as finding the rigid body transformation (a translation vector $\mathbf{t}$ and a rotation matrix $\mathbf{R}$) so that a global distance metric function

$$\Phi = \sum_{\forall \mathbf{p} \in \mathbf{r}_A} d_{sp}(\mathbf{r}_B, (\sigma \mathbf{R}\mathbf{p} + \mathbf{t})) \tag{4.4}$$

becomes minimal, where $\sigma$ is a scaling factor.

## 4.3 Surface Fitting

Two cases are dealt with for matching in this paper: the point vs. surface and the surface vs. surface matching cases. Since the proposed algorithms are based on differential properties for matching, robust and accurate extraction of them is important. When the surface vs. surface case is considered, the differential properties can be accurately calculated. However, estimation of umbilical points or curvatures from range data itself is a difficult problem. A surface fitting method such as a least squares fitting method or a method in [108] is preferred here because high order derivative properties can be calculated analytically from the fitted surface.

In this thesis, we use a NURBS surface fitting method in the least squares sense [93] using the singular value decomposition method. When data points are arranged as a grid, then the chord length parametrization method can be adopted for parametrization of the data points, and the control points of a NURBS surface are obtained using the standard least squares method. If data points are unorganized, one can use the base surface method for parametrization proposed by Ma and Kruth [69]. A Gaussian (low pass) filter may be used to eliminate high frequency noise in the data points to reduce the effects from the noise in the differential property calculation [87].

## 4.4 Matching Criteria

Three tests are proposed for evaluation of similarity. The $\epsilon$-offset test, the principal curvature test and the umbilic test are performed at the node points of the intrinsic wireframe.

### 4.4.1 $\epsilon$-Offset Test

How close surface $\mathbf{r}_B$ is to surface $\mathbf{r}_A$ in terms of the Euclidean distance is the objective of this test. The squared minimum distances between $\mathbf{r}_A$ and $\mathbf{r}_B$ are calculated and checked if all of them are within an $\epsilon$-distance bound or one of the surfaces is within an **$\epsilon$-offset** of the other. In [125], the squared distance function and its stationary points between two variable points located on two different geometric entities are investigated. Based on this technique, the distances between two surfaces $\mathbf{r}_A$ and $\mathbf{r}_B$ are calculated.

### 4.4.2 Principal Curvature and Direction

The principal curvatures and their directions are used in this similarity test. The differences of the principal curvatures and the directions between two surfaces at the reference points obtained from the shape intrinsic wireframe, are calculated and used for a similarity decision criterion.

### 4.4.3 Umbilic Test

Every closed orientable surface in $\mathbf{R}^3$ (of class $C^3$) with genus different from one (hence being topologically different from the torus) has at least one umbilic [14] and various free form surfaces may contain umbilical points. However, the availability of this test depends on the existence of isolated generic umbilics. This test is based on the fact that generic isolated umbilics and the patterns of lines of curvature around them are stable to perturbations so that their qualitative properties are preserved, and determines whether their locations and patterns for surface $\mathbf{r}_A$ match those for surface $\mathbf{r}_B$.

### 4.4.4 Assessment of Matching

Let us denote $k$ node points of the surface intrinsic wireframe from surface $\mathbf{r}_B$ as $\mathbf{P}_i$ $(i = 1, 2, \cdots, k)$. Here surface intrinsic sampling methods using geodesics or lines of curvature are preferred because they are independent of parametrization. Next, find the minimum distance foot-points $\mathbf{Q}_i$ on surface $\mathbf{r}_A$ of $\mathbf{P}_i$. The IPP algorithm can be used to find these minimum distance footpoints robustly as in [125]. After finding the footpoints $\mathbf{Q}_i$ on $\mathbf{r}_A$, we calculate the following quantities between $\mathbf{P}_i$ and $\mathbf{Q}_i$ $(i = 1, 2, \cdots, k)$.

- Euclidean distance of $|\mathbf{P}_i - \mathbf{Q}_i|$ : $\epsilon_{0i}$
- The second derivative properties
  - Difference of principal curvatures : $\epsilon_{1i}, \epsilon'_{1i}$
  - Difference of principal directions : $\epsilon_{2i}$

Maximum values, average values and standard deviations can be calculated for each $\epsilon_{0i}$, $\epsilon_{1i}$, $\epsilon'_{1i}$ and $\epsilon_{2i}$ to provide quantitative statistical measures to determine how similar the two surfaces are in a global manner.

Local similarity can be assessed with $\epsilon_{ji}$ at corresponding positions. Each $\epsilon_{ji}$ $(j = 0, 1, 2)$ is normalized with respect to the maximum value of $max_i(\epsilon_{ji})$. Tolerances $\delta_j$ $(j = 0, 1, 2)$, corresponding to $\epsilon_{ji}$, are used to extract the regions of interest. Namely, the regions in which $\epsilon_{ji} > \delta_j$ are those where the two surfaces are different. As an extension of this idea, the similarity between two surfaces can be provided as a percentage value. First, the difference values $\epsilon_{ji}$ are located over the $uv$ plane. Then the $uv$ plane is subdivided into a set of square grids of size $(\delta_s \times \delta_s)$ where $\delta_s$ is a user defined value. The total number of the square meshes is denoted as $D_T$. Given a tolerance $\delta_j$, the number of the squares $D_\epsilon$, which contain at least one point satisfying $\epsilon_{ji} > \delta_j$, is found. Then, $\left(1 - \frac{D_\epsilon}{D_T}\right) \times 100$ becomes a percentage value of similarity. The squares which do not contain points satisfying the condition indicate the regions where the two surfaces are equivalent under a given test with a tolerance $\delta_j$.

## 4.5 Moment Method

Integral properties are used for global matching of solids. The integral properties for solid $A$ and $B$, i.e. centroids (centers of volume) and moments of inertia, are calculated using Gauss's theorem or the divergence theorem which reduces volume integrals to surface integrals. The inertia tensor of solid $A$ and $B$ is constructed. It consists of a $3 \times 3$ square matrix whose diagonal terms $I_{xx}, I_{yy}$ and $I_{zz}$ are called the moments of inertia and the remaining terms $(I_{xy}, I_{yx}, I_{xz}, I_{zx}, I_{yz}$ and $I_{zy})$ the products of inertia. By the definition of products of inertia we have $I_{xy} = I_{yx}$, $I_{xz} = I_{zx}$ and $I_{yz} = I_{zy}$, and the tensor is symmetric. The principal moments of inertia and their directions are obtained by solving an eigenvalue problem [99].

Once the centroids and the principal directions of both solids are calculated, solid $A$ and $B$ are translated and rotated so that their centroids and principal axes of inertia coincide. If necessary, solid $B$ is uniformly scaled based on the relative volumes of the two solids.

There are cases where ambiguity in matching arises when the principal moments of inertia are used for matching. Since for each principal axis, two opposite directions are possible and thus a total of eight matching cases can be obtained. A right-handed coordinate frame assumption can reduce the eight matching cases to four [43]. Such ambiguity in matching the directions of the principal moments of inertia can be resolved by evaluating the sum of the squared distances calculated at the reference points for comparison and choosing one direction which yields the minimum value.

## 4.6 Correspondence Search

The main purpose of this section is to find a correspondence between two objects (points and a NURBS surface, or a NURBS surface and a NURBS surface) using surface intrinsic properties. Let us assume that we have two objects $\mathbf{r}_A$ and $\mathbf{r}_B$. The object $\mathbf{r}_B$ is a base surface and the object $\mathbf{r}_A$ is a target object which is positioned in a different pose with no scaling effect and is to be transformed for alignment to the base surface.

### 4.6.1 Algorithm using Umbilical Points

The availability of this method depends on the existence of umbilical points on an object. In order to locate umbilical points from range data, any kind of surface fitting method is used to get a NURBS surface, which is provided as input to the umbilical point detection algorithm [74, 91]. Let us assume that we have an approximated surface $\mathbf{r}_A$. When isolated generic umbilical points exist on both surfaces $\mathbf{r}_A$ and $\mathbf{r}_B$, we can use the $\omega$-plane to establish correspondence of the umbilical points. Let us assume that $\mathbf{r}_A$ and $\mathbf{r}_B$ have $n_A$ and $n_B$ umbilical points, respectively. Then we calculate $\omega$ in equation (2.30) for each umbilical point, i.e. $\omega_{Aj} \quad (j = 1, \cdots, n_A)$ for the umbilics on $\mathbf{r}_A$ and $\omega_{Bk} \quad (k = 1, \cdots, n_B)$ for the umbilics on $\mathbf{r}_B$. Since the number of isolated generic umbilical points on a surface is generally small, a brute force search scheme can be employed without loss of performance.

The searching process is to find pairs of $(\omega_{Aj},\ \omega_{Bk})$ of the same type which satisfy

$$|\omega_{Aj} - \omega_{Bk}| < \delta_\omega, \tag{4.5}$$

where $\delta_\omega$ is a user defined tolerance.

### 4.6.2 Algorithm using Curvatures

In this section, a new method is presented to find an optimal transformation, which minimizes a global distance metric function between two objects, when no prior knowledge on correspondence or transformation is provided [60]. Our work is motivated by Chua and Jarvis [25], and is applied to the points vs. surface and surface vs. surface cases. Our method uses intersection between iso-curvature lines of the Gaussian and the mean curvatures to establish correspondence, which is computed robustly by a subdivision algorithm, the *Interval Projected Polyhedron (IPP)* algorithm [106, 91]. This algorithm requires no initial estimate and avoids converging to a suboptimal local minimum, as it is a robust global algorithm which exhaustively isolates all solutions. Moreover, depending on the convergence tolerance used in the IPP algorithm, an accurate result can be obtained.

The overall diagram of the *KH method* is shown in Figure 4-1. The input of the process includes two objects and three pairs of the Gaussian and the mean curvatures at three different non-collinear locations on a target object $\mathbf{r}_A$. The algorithm yields the minimum value of $\Phi$ in equation (4.4), and the corresponding rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$. Since no scaling effect is involved, the scaling factor $\sigma = 1$.

The first step is to select three non-collinear points $\mathbf{m}_1$, $\mathbf{m}_2$ and $\mathbf{m}_3$ on $\mathbf{r}_A$ where each point has different Gaussian $K$ and mean curvature $H$ values. More than three

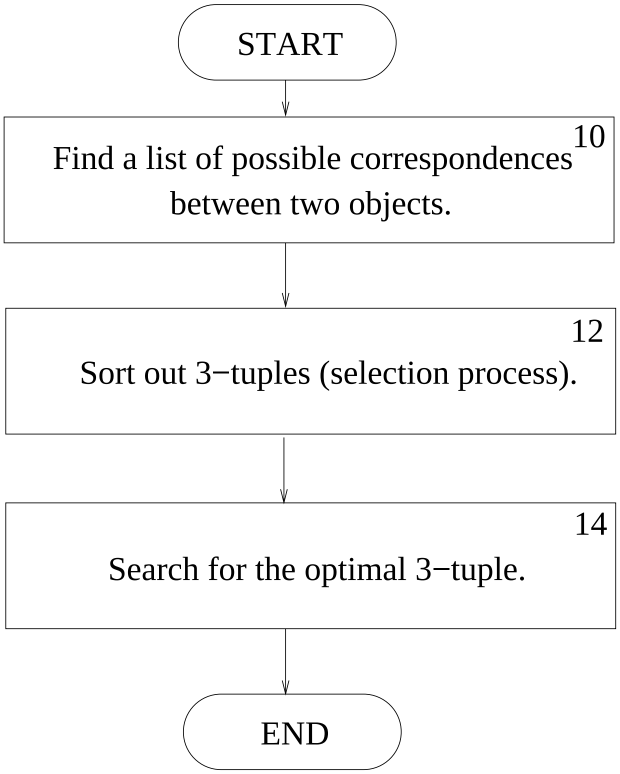


Figure 4-1: A diagram of the KH method

points can be used in this process as long as they are not collinear. But for performance reasons, three non-collinear points are chosen in this step. They are the minimum number of points from which the rigid body transformation can be obtained as shown in Section 3.2. If surface $\mathbf{r}_A$ is described in terms of discrete data points, we have to estimate the curvatures at these points as accurately as possible. This can be achieved by various methods such as the circle fitting method by Martin [77], the CP method by Stokely and Wu [112], the CT method by Sander and Zucker [104] and surface fitting methods described in Section 4.3. At $\mathbf{m}_i$, we assume we have computed estimates for the $K_i$ and $H_i$, where $i = 1, 2, 3$. Next, we subdivide $\mathbf{r}_2$ into rational Bézier surface patches $\mathbf{B}_j \quad (j = 1, \cdots, n)$ by inserting appropriate knots [50, 93]. Then, for each rational Bézier surface patch $\mathbf{B}_j$, we express $K_j$ and $H_j$ in the bivariate rational Bernstein polynomial basis form using equations (2.11) in either rational arithmetic or rounded interval arithmetic, as described in [72], to maintain accuracy in the calculation, see Appendix B for details. This allows us to use the Interval Projected Polyhedron (IPP) algorithm [91, 106] for solving nonlinear polynomial systems.

For each pair $(K_i, H_i)$, we solve the following system of equations by the IPP algorithm,

$$\begin{aligned} K_j(u,v) &= K_i \pm \delta_K, \\ H_j(u,v) &= H_i \pm \delta_H, \quad (j = 1, \cdots, n \ \ and \ \ i = 1, 2, 3), \end{aligned} \tag{4.6}$$

where $\delta_K$ and $\delta_H$ quantify the uncertainty in curvature estimation from discrete point sets.

Estimation of curvature values from a set of discrete data points with noise is a difficult problem, because the curvature is a second order differential property and sensitive to perturbations of the data points. The curvature values estimated by any method involve uncertainty, represented by $\delta_K$ and $\delta_H$ as in equations (4.6). The values of $\delta_K$ and $\delta_H$ are set to be zero when accurate curvatures are available. The accuracy of estimation depends on the accuracy of the digitization device, the estimation methods, etc. Thus, $\delta_K$ and $\delta_H$ need to be determined on a case-by-case basis.

During the formulation of equations (4.6) in the Bernstein basis, square root terms are present because of the normalization of the normal vector. To find the solution of system (4.6) with the IPP algorithm, we use the *auxiliary variable method* to handle the square root terms in the formulation [91, 72]. In this manner a nonlinear polynomial system of higher dimensionality is obtained, see Appendix B.

Two versions of equations (4.6) have been formulated: one is for integral and the other for rational Bézier surface patches. The latter is a general expression so that it can deal with the former as a special case. However, the formulation for the rational Bézier representation contains higher degrees than the other. Hence, depending on the input representation form, more appropriate formulation needs to be used. Let us assume that a given surface has degrees $m$ and $n$ in the $u$ and $v$ directions, respectively. The degrees of the equations which are provided to the IPP algorithm are summarized in Table 4.2, where $n_d$ is an auxiliary variable used in the formulation, see Appendix B. Since interval arithmetic is used in the IPP algorithm, any concern about the floating point error caused from high-degree polynomials can be avoided [91, 106, 72]. Hence, no roots of equations (4.6) are missed due to floating point error. The IPP algorithm has been used in a variety of multivariate nonlinear polynomial systems with great success.

| | Integral Bézier | | | Rational Bézier | | |
|---|---|---|---|---|---|---|
| Equation | $n_d$ | u | v | $n_d$ | u | v |
| (B.7) | 2 | 4m-2 | 4n-2 | 2 | 8m-2 | 8n-2 |
| (B.8) | 4 | 6m-4 | 6n-4 | 4 | 16m-4 | 16n-4 |
| (B.9) | 3 | 5m-3 | 5n-3 | 3 | 12m-3 | 12n-3 |

Table 4.2: Degrees for integral and rational Bézier representations

When interval curvature values for $K_i$ and $H_i$ are used in equations (4.6), the intersection of the Gaussian and mean curvature lines forms regions, i.e. a series of boxes on the $uv$ domain. The center points of the boxes are chosen as the roots of equations (4.6). Depending on the size of $\delta_K$ and $\delta_H$, the size of intersection boxes can be large. If the size of a box is larger than $\delta_b \times \delta_b$, where $\delta_b$ is a user-defined value, then they are subdivided into smaller ones and then the new center points are taken as the candidate points.

For each $K_i$ and $H_i$, a list of roots $L_i = (u_{ik}, v_{ik}),\quad (k = 1, \cdots, d_i)$ is obtained, from which one 3-tuple $(\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3)$

$$\mathbf{n}_1 = \mathbf{r}_2(u_{1k}, v_{1k}),\ \ \mathbf{n}_2 = \mathbf{r}_2(u_{2k}, v_{2k}),\ \ \mathbf{n}_3 = \mathbf{r}_2(u_{3k}, v_{3k}), \tag{4.7}$$

is selected to satisfy the following Euclidean distance constraints simultaneously,

$$\begin{aligned} ||\mathbf{m}_1 - \mathbf{m}_2| - |\mathbf{n}_1 - \mathbf{n}_2|| &< \delta, \\ ||\mathbf{m}_2 - \mathbf{m}_3| - |\mathbf{n}_2 - \mathbf{n}_3|| &< \delta, \\ ||\mathbf{m}_3 - \mathbf{m}_1| - |\mathbf{n}_3 - \mathbf{n}_1|| &< \delta, \end{aligned} \tag{4.8}$$

where $\delta$ is a user-defined tolerance. After this step, correspondence between each point $\mathbf{m}_i$ on $\mathbf{r}_A$ and $\mathbf{n}_i$ on $\mathbf{r}_B$ is established, from which a translation vector and a rotation matrix can be obtained.

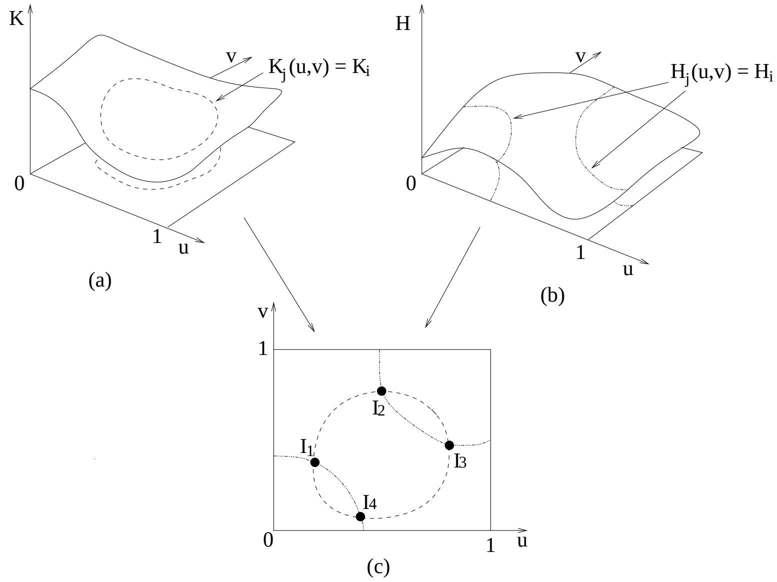


Figure 4-2: (a) The Gaussian curvature (b) The mean curvature (c) The pre-images of the Gaussian and mean curvatures on the uv domain.

Geometrically, solving the system of equations (4.6) is equivalent to finding the intersection points on $uv$ parameter space of $\mathbf{r}_B$ between the constant Gaussian curvature lines and the constant mean curvature lines. For each Bézier surface patch, $K_j$ and $H_j$ may be visualized as an explicit surface over the same parameter domain. Figure 4-2 shows a schematic diagram of the method. One of equations in (4.6), $K_j(u, v) = K_i$ indicates the iso-curvature lines of $K_i$ in Figure 4-2(a) and $H_j(u, v) = H_i$ the iso-curvature line of $H_i$ on the surface $H_j$ in Figure 4-2(b). Finding the roots of system (4.6) is equivalent to locating the intersection points in the parameter space, i.e. $I_1$, $I_2$, $I_3$ and $I_4$ as shown in Figure 4-2(c). It may happen that the curves of $K_j$ and $H_j$ on the parameter space overlap so that an infinite number of roots exist. In such a case, the IPP algorithm produces a finite number of boxes which contain these roots and whose size depends on the tolerance used to terminate the

IPP algorithm, and each box is used as a candidate for the subsequent step. When $K_i$ and $H_i$ are provided with non-zero $\delta_K$ and $\delta_H$, then $I_1$, $I_2$, $I_3$ and $I_4$ are obtained as rectangular boxes, and the center points of each of these boxes are used as candidate points for the next step.

A prune search algorithm using the constraints (4.8) is employed to find candidate points from the list of roots $L_i$. The first constraint is used for lists $L_1$ and $L_2$ to sort out the candidate points $(u_{1s_1}, v_{1s_1})$ and $(u_{2s_2}, v_{2s_2})$. Then, the second constraint is applied to $(u_{2s_2}, v_{2s_2})$ and $L_3$ to yield $(u_{3s_3}, v_{3s_3})$. The last condition is used between $(u_{1s_1}, v_{1s_1})$ and $(u_{3s_3}, v_{3s_3})$ to determine the 3-tuple of $(u, v)$, i.e.

$$F_s = \{(u_{1s_1}, v_{1s_1}), (u_{2s_2}, v_{2s_2}), (u_{3s_3}, v_{3s_3})\},$$

which satisfies all three constraints (4.8). Let us assume that $L_i$ has $d_i$ points. Then applying equation (4.8) involves searching a 3-tuple in $L_i$ that satisfies the constraints. Therefore, the time complexity in the worst case is $O(d_1 d_2 d_3)$. If $F_s$ has more than one tuple as a member, we have to choose one that minimizes a distance metric function.

Unfortunately, the mean curvature value changes its sign when the directions of parametrization are reversed. Therefore, two mean curvature values (positive and negative) are possible for the mean curvature values in (4.6) so that the whole process needs to be performed twice to cover all possible cases. Then, two sets of $F_s$ are generated from which a tuple has to be selected which minimizes a distance metric function.

After two 3-tuples with the correspondence information have been found, a rotation matrix and a translation vector can be calculated by standard techniques, e.g. using a unit quaternion [48].

## 4.7 Algorithms with Uniform Scaling Effects

In this section, matching algorithms involving scaling are discussed. Two methods are proposed. One is using umbilical points for matching and the other is formulating a matching problem based on optimization.

The KH method described in Section 4.2 is designed to solve a matching problem when no scaling effect is included by establishing correspondence, and in this section it is extended to resolve scaling effects in the matching problem. Here, only uniform scaling is considered because non-uniform scaling may distort the geometry of a surface so that its functionality may be destroyed.

In case of global matching, a scaling value may be recovered by calculating the ratio of areas between two surfaces or volumes between two solids. However, for partial matching, comparison of any type of quantitative measures does not make sense. Only qualitative feature matching can be considered. The other possible approach is to search for a good match out of many possible solutions. Both methods are explained in the following subsections.

### 4.7.1 Use of Umbilical Points

The correspondence search explained in Section 4.6.1 only deals with qualitative aspects. Since the $\omega$-plane is not affected by scaling, only qualitative correspondence can be established in the process. This implies that without a scaling factor applied, the rigid body transformation cannot be obtained for aligning two surfaces. Therefore, a scaling factor has to be estimated before any transformation is considered.

#### Method 1

Let us assume that we have two surfaces $\mathbf{r}_A$ and $\mathbf{r}_B$, where $\mathbf{r}_A$ is an approximated surface of input data points. The overall procedure is shown in Figure 4-3.

In step 100, all generic umbilical points are located on both surfaces $\mathbf{r}_A$ and $\mathbf{r}_B$ using the IPP algorithm [74, 91]. Non-generic umbilical points are not used in this process. If a generic umbilical point does not exist, this procedure cannot be applied.

In step 102, the correspondence search described in Section 4.6.1 is performed. The value $\omega$ in the complex plane is scale-independent so that qualitative correspondence can be built from this step. Suppose that matched pairs are denoted as $m_k$, $(k = 1, \cdots, n_k)$, where $n_k$ is the number of matched pairs. Then, when at least one pair is found, the next step 104 is performed. If no correspondence is established, then the algorithm stops, implying that the umbilical point method cannot be used in this case.

Step 104 resolves the scaling transformation. To calculate a scaling factor, the normal curvatures are evaluated at the corresponding umbilical points on both surfaces $\mathbf{r}_A$ and $\mathbf{r}_B$. Then the ratio between them is obtained as a scaling factor. Suppose that a surface $\mathbf{r}$ is scaled with a scaling factor $\sigma$, denoted as $\mathbf{r}_\sigma$. Then the normal curvature $\kappa$ on $\mathbf{r}$ is scaled to be $\frac{\kappa}{\sigma}$ on $\mathbf{r}_\sigma$. Therefore using this relation, the scaling factor can be recovered.

In step 106, after sorting out candidate points, the rigid body transformation is estimated by using the unit quaternion method [48]. Since the number of matched pairs is more than three, the problem is reduced to finding the rigid body transformation with three known corresponding pairs which goes to step 112. Using the methods in [48] a rotation matrix and a translation vector can be calculated. If the transformation does not exist, then the algorithm goes to step 108 which deals with the matching process using less than three matched pairs.

In step 108, the orientations of the normal vectors at the corresponding umbilical points are aligned. First, translate the scaled surface $\mathbf{r}_A$ by the difference between the positions of the matched umbilical points. Then, align the normal vectors at the umbilical points. The alignment of the normal vectors can be achieved by using the unit quaternion method [48]. Let us assume that we have two normal vectors $\mathbf{n}_A$ and $\mathbf{n}_B$ at the corresponding umbilical points for $\mathbf{r}_A$ and $\mathbf{r}_B$, respectively. The problem of matching the normal vectors can be stated as: rotate $\mathbf{n}_A$ around the vector $\mathbf{N}_n = \frac{\mathbf{n}_A \times \mathbf{n}_B}{\|\mathbf{n}_A \times \mathbf{n}_B\|}$ by an angle $\theta$ formed by $\mathbf{n}_A$ and $\mathbf{n}_B$. The angle $\theta$ can be calculated by $\theta = \arccos(\mathbf{n}_A, \mathbf{n}_B)$; see [48] for details of rotation in the quaternion frame.

In step 110, matching of lines of curvature emanating from an umbilical point is performed. Depending on the type of the umbilical point, one (*lemon*) or three (*star and monstar*) lines of curvature pass through the umbilical point as shown in Figure 2-1, and each direction can be determined by the structure of the cubic terms $C(x, y)$ as summarized in Section 2.1.4. The directions can be obtained by calculating angles of the lines of curvature with respect to the local coordinate system at the umbilical point [74, 91]. Using the angles, vectors which indicate the directions of the lines of curvature at the umbilical point can be obtained. These vectors are calculated at the matched umbilical points on $\mathbf{r}_A$ and $\mathbf{r}_B$. Suppose that the number of the direction vectors on $\mathbf{r}_A$ is $n_{vA}$ and the number of the direction vectors on $\mathbf{r}_B$ is $n_{vB}$. Choose one vector from $\mathbf{r}_B$ and align all of the vectors on $\mathbf{r}_A$. This process produces $n_{vA}$ matched cases among which one match is chosen that minimizes equation (4.4). This alignment is achieved by rotation around the normal vector in the tangent plane at the matched umbilical point. Therefore, the rotation method using the unit quaternion can be used in this process [48].

#### Method 2

The rigid body transformation can also be obtained by using the KH method described in Section 4.6.2 after the scaling transformation is resolved. The algorithm is the same as in Figure 4-3 from step 100 through step 104. After the scaling transformation is resolved, the KH method can be used to find the rigid body transformation between two objects.

### 4.7.2 Optimization

A matching problem with scaling effects can be solved with an optimization technique. Since there is no quantitative measure that can be used to estimate a scaling value, the solution scheme has to resort to an optimum search method which can narrow down the best estimate from the possible set of candidate solutions.

The KH method can be treated as a function of the scaling factor which yields a value of $\Phi$ in equation (4.4) when a scaling factor is given. Namely, steps 10, 12 and 14 in the diagram of Figure 4-1 are grouped as a function $f$ such that

$$f = \Phi(\sigma, \mathbf{R}, \mathbf{t}), \tag{4.9}$$

where $\Phi$ is the expression given in equation (4.4), $\sigma$ the scaling factor, $\mathbf{R}$ the rotation matrix and $\mathbf{t}$ the translation vector. Since the rotation matrix and the translation vector are obtained from the KH method, we can reduce equation (4.9) to a function of one variable, that is $f = \Phi(\sigma)$. Therefore, when $\sigma$ is given as input, then $f$ produces the best rigid body transformation (a translation vector and a rotation matrix) as well as the value of the object function defined in equation (4.4). When the selection process fails, the tolerance $\delta_{select}$ is iteratively increased so that any triplet can be obtained. When no triplet is found, then the function $f$ is penalized to yield a very large value.

Using the function $f(\sigma)$, the problem can be formulated as a one dimensional

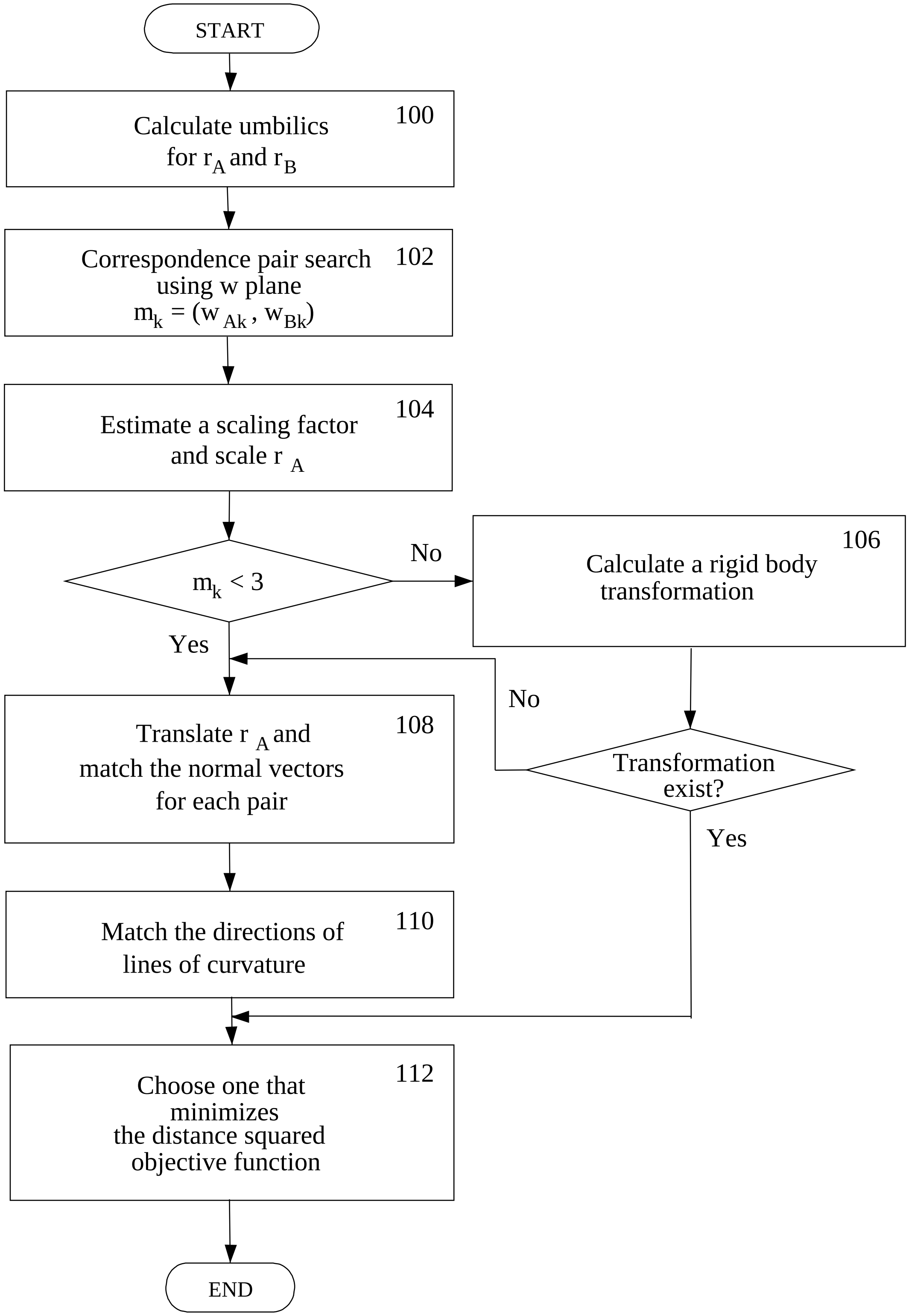


Figure 4-3: A diagram for matching algorithm using umbilics.

optimization problem to find a scaling value which yields the minimum of $f$. A popular one dimensional optimization scheme, the *golden section search* [98] can be employed to solve it.

An initial bracket $[a, b, c]$ of the scaling factor is provided which contains an optimum value, and satisfies $f(a) > f(b)$ and $f(c) > f(b)$. Suppose this bracket is $I_0$. The golden section search starts with $I_0$ and continues while the size of an interval containing the optimum value $\sigma$ is larger than a user defined tolerance which determines the size of the interval.

### 4.7.3 Complexity Analysis

In this section, the proposed algorithms are analyzed in terms of time complexity and accuracy. The convergence of the optimization method is also discussed.

#### Surface Fitting

The least squares fitting method requires solving a system of linear equations. Suppose that there are $r$ input data points and the number of the control points of an approximated NURBS surface patch is $c$. Then the solution algorithm of the singular value decomposition takes $O(rc^2 + c^3)$ multiplications [12, 45]. In general, the singular value decomposition method is slower than solving the normal equations. However, it is more stable and reliable [98].

#### IPP Algorithm

The total asymptotic computation time per step is $O(nlm^{l+1})$ [106, 91], where $n$ is the number of the nonlinear equations that need to be solved, $l$ the number of independent variables, $m$ the maximum degree of the variables. Since the total number of steps depends on a user-defined tolerance, it is hard to predict how many steps will occur in advance. But for analysis purposes, we can use a coefficient $\alpha$ to indicate the total number of steps performed in the algorithm. Then, the asymptotic time complexity becomes $O(\alpha nlm^{l+1})$.

#### Calculation of Umbilical Points

Calculating isolated umbilical points from a Bézier surface patch depends on the degree of the surface. The governing equations that have to be solved are equations (2.25), and the IPP algorithm is employed for the solution to the system of the nonlinear equations. Let us assume that the degrees of a Bézier surface are $d_u$ and $d_v$ in $u$ and $v$ directions. In this case, the maximum degrees of each governing equation in $u$ and $v$ directions are proportional to the degrees of the input surface. The number of governing equations is three and the number of independent variables is two. Therefore, the time complexity for the calculation of umbilical point using the IPP algorithm is reduced to $O(m^3)$ per step, where $m = max(d_u, d_v)$.

In general, a surface is represented in NURBS form. Therefore, in order to apply the IPP algorithm to locate umbilical points, the surface needs to be subdivided into

Bézier surface patches by the knot insertion algorithm [93, 91]. Suppose, a surface has $c_u \times c_v$ control points. Then the total number of subdivided Bézier surface patches is proportional to $c_u c_v$. Therefore, it is concluded that given a NURBS surface of degree $d_u$ and $d_v$ in $u$ and $v$ directions with $c_u \times c_v$ control points, the total complexity is $O(c_u c_v \alpha m^3)$ where $m = max(d_u, d_v)$ and $\alpha$ is the number of steps performed in the IPP algorithm.

### Umbilical Method

The time complexity of the matching process of the umbilical point method depends on the number of isolated umbilical points on the surfaces. Suppose that $n_1$ and $n_2$ are the number of umbilical points on the model and target surfaces $\mathbf{r}_1$ and $\mathbf{r}_2$. Then the search of correspondence takes $O(n_1 n_2)$ time in the worst case. However, in general, $n_1$ and $n_2$ are small integers. Therefore, the elapsed time is typically negligible.

### Optimization Method

The elapsed time of the optimization method relies on the number of iterations in the IPP algorithm and the tolerance of the golden section search. Therefore, it is reasonable to consider the time complexity of one iteration of the optimization method. The complexity of the KH method consists of two parts: the IPP algorithm and the selection process. The IPP algorithm solves equations (4.6) using the auxiliary variable method [91]. Therefore, the number of equations and the number of independent variables is three. Suppose that the degrees of a NURBS surface are $d_u$ and $d_v$ and denote the total number of steps performed by the IPP algorithm by $\alpha$ which depends on the tolerance provided by the user. Since the maximum degree of each equation is proportional to the degree of the surface, the total complexity for the IPP algorithm becomes $O(\alpha m^4)$, where $m = max(d_u, d_v)$. The selection process takes $O(d_1 d_2 d_3)$ where $d_i$ is the number of points of $L_i$ in the KH method. Therefore, the overall time complexity is $O(\alpha l m^4 + d_1 d_2 d_3)$ per iteration of the golden section search.

## 4.7.4 Accuracy Analysis

### Umbilical Point Method

The accuracy of the umbilic method depends on that of the computation in locating umbilical points. The IPP algorithm requires a tolerance $\delta_{umb}$ which limits the size of intervals containing roots. Therefore, the tolerance $\delta_{umb}$ becomes the expected maximum accuracy bound of the umbilic method.

The quantitative estimation of the normal curvature at an umbilical point is important to recover a scaling factor. Since, in general, the curvature value is hard to be estimated from a set of data points, the umbilical matching method may not yield a satisfactory result. However, the result can be used as an initial estimate for an optimization method such as the ICP or the optimization method proposed in this thesis.

**Optimization Method**

The accuracy of this method depends on various tolerances that need to be provided as input to the algorithm. First, the KH method requires two different tolerances $\delta_{IPP}$ and $\delta_{select}$. The tolerance $\delta_{IPP}$ is provided as input to the IPP algorithm which limits the size of intervals of roots. The tolerance $\delta_{select}$ is used in the selection process of the KH method. However, the tolerance $\delta_{select}$ does not affect the accuracy of the final result because it is used as a value to sort out 3-tuples from a set of candidate points. Therefore, the tolerance $\delta_{select}$ can be a quite large number compared to $\delta_{IPP}$. The additional tolerance $\delta_G$, which is used by the golden section search, is the one that affects the accuracy of the result, and it restricts the size of an interval which contains an optimum scaling factor.

The tolerance $\delta_G$ needs to be carefully chosen [98]. The algorithm is designed to stop when the size of a bracket becomes smaller than $\delta_G$ during the iteration. But how small the tolerance can be is an important issue that needs to be clarified. As Press *et al.* [98] indicate, the smallest tolerance that can be used for the golden section search is $\sqrt{\epsilon}$, where $\epsilon$ is the machine precision. A smaller tolerance than $\sqrt{\epsilon}$ results in unnecessary subdivisions of the bracket so that the overall performance deteriorates.

### 4.7.5 Convergence of the Optimization Method

The function $f = f(\sigma)$ is not necessarily smooth so that it may not be suitable for optimization methods such as the *parabolic interpolation in one dimension* [98], that can find an optimum value more efficiently when a function is smooth. Instead, a slow but robust method, the golden section search in one dimension is employed for this optimization problem. This optimization technique is designed to cope with the worst possible case [98] and narrow down the interval which is guaranteed to contain an optimum value. It is known that the golden section search converges linearly to an interval of user-defined size which is guaranteed to contain an optimum (local minimum) value. The strategy of the proof is to show that the interval in the subsequent step which contains an optimum value decreases by a factor of $\gamma$ $(< 1)$. An assumption is made that three points $a$, $b$ and $c$ in a bracket $[a, b, c]$ satisfy the following conditions.

$$f(a) > f(b), \quad f(c) > f(b). \tag{4.10}$$

In addition, suppose that $w$ is $\frac{3-\sqrt{5}}{2}$. This is called *the golden section*[1] which is used in the subdivision of the interval. We have an initial bracket $[a_0, b_0, c_0]$ which contains an optimum value and satisfies (4.10). At the $n$-th step, suppose that we have an interval $[a_n, b_n, c_n]$ and at $a_n$, $b_n$ and $c_n$ the conditions (4.10) are satisfied. Moreover, we have $\frac{b_n - a_n}{c_n - a_n} = w$. The interval $l_n$ is calculated as $l_n = |c_n - a_n|$. At the next step, i.e. the $(n+1)$-th step, a value $x_n$ is selected that satisfies the ratio $\frac{x_n - a_n}{c_n - a_n} = 1 - w$ and then the function $f(x_n)$ is evaluated there. If $f(x_n) > f(b_n)$, then the interval

---

[1] The number $\frac{3-\sqrt{5}}{2}$ is a root of $w^2 - 3w + 1 = 0$, which supposedly has aesthetic properties dating back to the ancient Pythagoreans [98].

$[a_n, b_n, x_n]$ is chosen. If $f(x_n) < f(b_n)$, then the other interval $[b_n, x_n, c_n]$ is taken. These two conditions guarantee that the selected interval encloses the optimum value all the time. First, let us assume that the interval $[a_n, b_n, x_n]$ is selected. The size of the interval is $l_{n+1} = |x_n - a_n|$ which is equivalent to $(1-w)l_n$. Since $1 - w < 1$, $l_n$ is decreased to $l_{n+1}$ by a factor of $1 - w$. Next, consider the interval $[b_n, x_n, c_n]$. The size of the interval is $l_{n+1} = |c_n - b_n|$ which is also equal to $(1-w)\,|c_n - a_n|$, i.e. $(1-w)l_n$. Therefore, in both cases, the size of the interval $l_n$ is reduced by a factor of $1 - w \ < 1$.

## 4.8 Performance Considerations

The performance of the KH method depends on the degree of equations (4.6). As shown in Table 4.2, when a bicubic integral Bézier surface is used, the degrees of equations (B.7), (B.8) and (B.9) are 10, 14 and 12 for both $u$ and $v$, respectively. For a bicubic rational Bézier surface case, the degrees are 22, 44 and 33 for $u$ and $v$, respectively. Therefore, solving equations (4.6) using the IPP algorithm takes substantial time because the IPP algorithm has $O(m^4)$ complexity per step for a bicubic Bézier surface patch, where $m$ is the maximum degree of the variables in the equations. For a rational surface patch, the degrees are exceptionally high so that the computation time is much larger for practical applications.

The overall performance can be improved by using low-degree approximation of the Gaussian and the mean curvature functions. The Gaussian and the mean curvature functions are sampled over the parametric domain, and using the approximation techniques such as the multilevel B-spline approximation by Lee *et al.* [64] or the low degree approximation of high degree B-spline surfaces by Tuohy and Bardis [117], those two point sets can be approximated or interpolated with bicubic B-spline surfaces. Such approximation can be carefully performed to represent the sampled points as closely as possible and this process can be done off-line. Then, the low degree curvature functions are used as input to the algorithm.

The surface shown in Figure 4-4 is used for test purposes. The surface is a bicubic B-spline surface patch with $20 \times 20$ control points. For this comparison, the surface vs. surface localization case is considered. In this example, for the Gaussian and the mean curvature approximation, 10,000 points are sampled uniformly over the parametric domain, and the Gaussian and the mean curvatures are calculated at each parametric value. Then, the curvature functions are represented as a height function over the parametric domain. They are approximated by using the hierarchical B-spline method by Lee *et al.* [64], which produces two bicubic B-spline surfaces of $259 \times 259$ control points for the Gaussian and the mean curvatures as shown in Figure 4-5, respectively. The elapsed time for approximation is 7.1 seconds for each curvature. The curvature surfaces are subdivided into cubic Bézier surfaces, which are provided as input to the IPP algorithm. In this case, the algorithm needs to be executed $t_n$ times, where $t_n$ is the total number of the subdivided Bézier patches. In this example, the number, $t_n$ is 67081 $(259 \times 259)$, a rather large number. However, since the maximum degree of the input surface is three, the total elapsed time of the IPP algorithm per step decreases

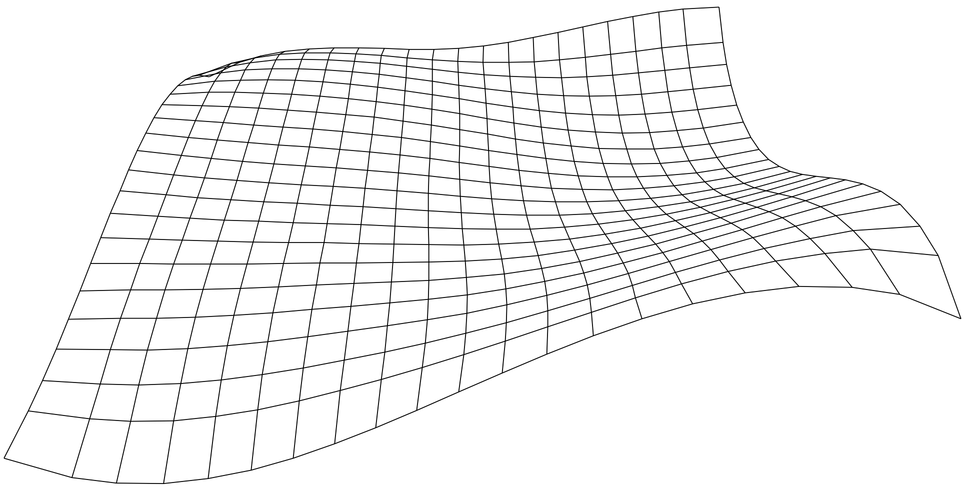
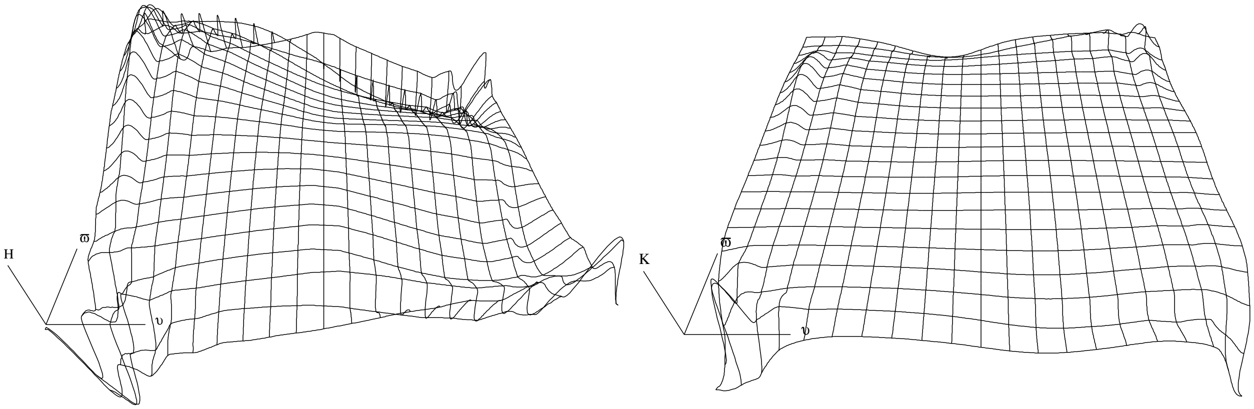


Figure 4-4: The surface used for the peformance test



**Mean Curvature** **Gaussian Curvature**

Figure 4-5: The approximated Gaussian and the mean curvature function graphs

drastically so that the time increase due to the increased number of executions of the IPP algorithm is compensated by the use of low degree polynomials in the algorithm.

The quantitative comparison of the total elapsed execution time between the method using approximation of the curvature functions and the IPP algorithm on the high degree function based on the integral B-spline formulation is made in Table 4.3. A linux machine with 1.6 GHz CPU and 512 Mbytes memory is used for this test. The approximation method is three times faster than the IPP algorithm with small error difference. The difference between the two methods will be even more pronounced for the rational B-spline.

The approximation can be made as accurate as possible using more control points. If an adaptive approximation method is used, then the accurate approximation can be achieved with less control points so that the total execution time can be reduced.

| | Approximation | IPP |
|---|---|---|
| Time (sec) | 225 | 725 |
| Relative Error | 0.0008 | 0.0001 |

Table 4.3: Time comparison of two methods

## 4.9 Conclusions

We have addressed a global/partial matching problem of free-form objects with no prior clue on correspondence or initial transformation, and extended it to include scaling effects. We demonstrated that correspondence between surface features can be optimally established using the constant Gaussian and mean curvature lines and generic umbilical points, and a scaling factor can be estimated by using the normal curvatures at the generic umbilical points or an optimization method. The proposed methods can be applied to free-form object matching with scaling and do not require initial estimates for optimal matching.

With rough tolerances, the KH method can be used to produce a good initial estimate for iterative algorithms such as the ICP algorithm and its variants. When tight tolerances and accurately estimated curvatures are used, the method yields accurate transformations. The KH method works well for the case when no information on correspondence or transformation exists. With no good initial estimate, the iterative algorithms may end up being trapped in a local minimum so that a successful matching cannot be obtained. The KH method overcomes this problem. Moreover, through approximation of the Gaussian and mean curvature functions with low degree B-spline functions, the overall performance can be improved to compete with standard iterative algorithms such as the ICP by Besl and McKay [10].

In the optimization approach for partial matching with scaling effects, an initial interval for the golden section search is required. But compared to the iterative methods such as the ICP [10] which needs not only an initial scaling value but also a good rigid body transformation to avoid converging to a local minimum, the proposed method in this chapter is much simpler in the sense that only an approximate interval of the scaling factor is necessary, since the algorithm is able to find a rigid body transformation with no prior clue on correspondence or initial transformation.

The moment method, which is one of the global matching methods, is used only for matching solids. Solid matching can be covered by the proposed methods such as the KH method or the umbilical point method. However, for global matching of solids, the moment method may be preferred for efficiency reasons.

# Chapter 5

# Shape Intrinsic Fingerprints

## 5.1 Introduction

Several methods have been reported on digital watermarking for 3D polygonal models which are widely used for virtual reality and computer graphics. Fornaro and Sanna [41] developed a *public* watermarking technique for authentication of CSG (or boundary representation) models. An extractable watermark is built using a hash function and a public key algorithm is used for encryption and decryption of the watermark. Two places are considered for storing the watermark: solids and comments. To store the watermark in the solid, a new watermark node is created and linked to the original CSG tree. In the comments, the watermark information can be added without changing the model. 3D polygonal models are popular in computer graphics and multimedia. Ohbuchi *et al.* [85] proposed data embedding methods for 3D polygonal models. Three attributes for data embedding are studied: geometrical quantity, topology and non-geometrical quantity. Dimensionless quantity pairs such as ratio between an edge and a height in a triangular mesh or tetrahedral volume (defined by three vertices of each triangle and a common apex vertex) ratios are used for data embedding. Topological modification is another way to embed information. The adjacency (connectivity) of triangles in a triangle strip can be used, or by stripping off a pattern out of a given mesh, desired information can be embedded. For non-geometrical quantity, change of texture coordinates is used for data embedding. Yeo and Yeung [120] proposed a fragile watermarking method, which detects unauthorized alterations of 3D models. Watermarks that are sensitive to slight modification are embedded, which allows easy detection, localization and visualization of the modification. Vertex coordinates are slightly altered for watermark embedding such that for every vertex, the bit value from the key function should match the watermark bit. Benedens [6] proposed a method to embed watermarks by altering surface normal distribution, which is achieved by displacing vertices in the mesh, inducing alteration adjacent face normals. Normals are grouped into distinct sets called bins which are the entities for embedding one bit of watermark data. Each bin's property is changed to embedding watermark information. Cox *et al.* [29] proposed a spread spectrum watermarking for multimedia data such as audio and video. Discrete cosine transform is used for Fourier domain method and a watermark is embedded in the data's perceptually significant frequency components. The spread spectrum watermarking for 3D polygonal models was proposed by Kanai *et al.* [56]. It is based on wavelet transformation and multiresolution representation of the polygonal model. The watermark is embedded in the large wavelet coefficient vectors at one or more resolution levels of detail. On the other hand, Yin *et al.* [121] used Guskov's multiresolution signal processing method for meshes and constructed a Burt-Adelson pyramid for the mesh using the 3D non-uniform relaxation operator. Through pyramid construction, the low frequency components are found, where the watermark information is embedded. Praun *et al.* [97] extended the spread-spectrum approach to work for the watermarking of arbitrary triangle meshes. A set of scalar basis functions over the mesh vertices from multiresolution analysis is constructed and used to insert the watermark by disturbing the original vertices based on the scalar function and watermark information.

Despite the popularity of NURBS curves and surfaces, watermarking for NURBS representations is relatively new to engineering CAD. Ohbuchi *et al.* [86] proposed a new data embedding algorithm for NURBS curves and surfaces, which are reparameterized using rational linear functions whose coefficients are manipulated for encoding data. The exact geometric shape is preserved.

Watermark for verification such as [41] or [120] is fragile to a malicious attack. The existence of the watermark determines whether the model is original or not. Its purpose is only for authentication so that the watermark may or may not be visible or extractable by a third party. On the other hand, robust watermark has a somewhat different purpose. It is designed to exist against any kind of malicious attacks to erase it. The hope is that if the attacker wants to get rid of it, he/she needs to degrade the object significantly enough so that the object contains different functional quality. Various mesh watermarking methods have been proposed but most of them suffer from vulnerability from some kind of attack. The change of topology of mesh or use of geometrical or non-geometrical quantities by Ohbuchi *et al.* [85] is not robust against remeshing, simplification and the addition of noise. Surface normals are used to overcome the weaknesses by Benedens [6] but its application is limited. Praun *et al.* [97] proposed a robust watermarking method against such attacks by using the extended spread-spectrum method and multiresolution analysis. Watermark for NURBS curves and surfaces was proposed by Ohbuchi *et al.* [86]. The watermark information, however, can be easily destroyed by reparametrization or reapproximation of the surface.

Since it is difficult to insert any form of robust user-defined watermark in the NURBS representation, the similarity checking method is adopted in this work. Every surface or solid via the definition of its shape has its own shape identity in determining the geometry of the object through a Euclidean motion. This shape identity might be called the *shape intrinsic fingerprint*, as it is not artificial but is related merely to the definition of the object's shape. One example of these shape intrinsic features are umbilics. Therefore, the fundamental issue involved here is to compare two objects (solids or NURBS surfaces) using the shape intrinsic fingerprints and determine whether one object is a copy of the other.

Since digital fingerprints are salient features of an object, extracting these fingerprints does not disturb the object as opposed to watermarking which embeds additional information resulting in some modification of the object. Digital fingerprints have been widely used for 2D images and multimedia data. Recognition, digital indexing and tracking of images or multimedia data [54, 37] are main applications of the digital fingerprints. Some researchers refer to digital fingerprints as *passive watermarks*. They can also be used as an alternative to digital watermarks. However, digital fingerprints for 3D CAD models have not been discussed so far.

Before comparing two free-form objects, they first have to be aligned so that the effects of the rigid body transformation (rotation and translation) and scaling are minimized. This alignment problem is equivalent to *localization* or *registration*. The proposed matching algorithms are used for localization/registration. For checking whole solids the moment method is preferred due to efficiency. The proposed algorithms in this chapter are employed to compare the whole or part of surfaces or solids.

## 5.2 Algorithms

Two similarity decision algorithms are proposed in this section. The decision algorithms consist of three tests as described in Section 4.4 and provide quantitative results with which one can determine whether one object is a copy of another or not. One assumption is made that the objects which the algorithms are applied to are either NURBS surfaces or solids bounded by NURBS surfaces. For a solid, the bounding NURBS surfaces of the solid are used for comparison.

Algorithm 1 uses the maximum deviation value at each test for a decision, while Algorithm 2 employs statistical methods for a decision. Each algorithm produces hierarchical results for similarity between two surfaces. In the subsequent sub-sections, it is assumed that $\mathbf{r}_A$ and $\mathbf{r}_B$ are matched, wireframed, and all umbilical points are detected.

### 5.2.1 Algorithm 1

Two surfaces are provided as input to the first test or a weak test ($\epsilon$-offset test) as shown in Figure 5-1. Then a decision is made that surfaces $\mathbf{r}_A$ and $\mathbf{r}_B$ are within or out of tolerance $\epsilon_d$ based on the Euclidean distance. If the maximum distance between corresponding points on surfaces $\mathbf{r}_A$ and $\mathbf{r}_B$ is within $\epsilon_d$, then surface $\mathbf{r}_B$ is considered to have passed the weak test and determined to be a copy of $\mathbf{r}_A$ under the weak test. On the other hand, if the distance is greater than tolerance $\epsilon_d$, the test fails. In such case, there are two possible courses of action. If $\epsilon_d$ is not large with respect to the size of surfaces, the user may decide to increase it and retry the weak test. If $\epsilon_d$ is large, then the user may decide to stop the process and decide that $\mathbf{r}_B$ is not derived from $\mathbf{r}_A$.

If the weak test is passed, then an intermediate test (principal curvature and direction test) may be performed. The procedure is similar to that of the weak test.

If the test succeeds, $\mathbf{r}_B$ is considered to be a copy of $\mathbf{r}_A$ under the intermediate test. If it fails, the user may decide to stop the process and conclude that $\mathbf{r}_B$ is derived from $\mathbf{r}_A$ with respect only to the weak test or try the test again with a new $\epsilon_a$.

If no isolated generic umbilical point exists, the process stops and it is concluded that $\mathbf{r}_B$ is derived from $\mathbf{r}_A$ under the intermediate test. If an umbilic exists, the strong test (umbilic test) may be performed. If the test succeeds, it is concluded that $\mathbf{r}_B$ is derived from $\mathbf{r}_A$ under the strong test. Otherwise, it is decided that $\mathbf{r}_B$ is a copy of $\mathbf{r}_A$ under the intermediate test.

### 5.2.2 Algorithm 2

The overall procedure is the same as Algorithm 1 except that no iteration is involved; see Figure 5-2. In this algorithm, a decision is made based on statistical information obtained from each test. From the weak test, statistics of the distance function are computed and evaluated by the user or a computer program. If the statistics pass a set of threshold tests, then it is concluded that $\mathbf{r}_B$ is derived from $\mathbf{r}_A$ under the weak test, and then the intermediate test begins. Otherwise, it is concluded that $\mathbf{r}_B$ is not derived from $\mathbf{r}_A$.

The intermediate test constructs statistical information of intrinsic properties, such as angle differences of the principal directions. A determination is made as to whether the statistics pass a set of threshold tests. If the threshold tests are negative, it is concluded that $\mathbf{r}_B$ is derived from $A$ under the weak test. Otherwise, it is concluded that $\mathbf{r}_B$ is derived from $\mathbf{r}_A$ under the intermediate test.

Similarly, depending on the existence of umbilical points, the strong test may be performed. Statistics of position differences of the locations between corresponding isolated generic umbilics are considered in this test. A decision is made as to whether the statistics pass a set of threshold tests. If the tests are negative, $\mathbf{r}_B$ is concluded to be derived from $\mathbf{r}_A$ under the intermediate test. Otherwise, it is decided that $\mathbf{r}_B$ is derived from $\mathbf{r}_A$ under the strong test.

## 5.3 Conclusions

In this chapter, two decision algorithms are proposed which provide systematic and statistical measures for a user to check two surfaces for similarity using the three tests discussed in Section 4.4. They can be applied to copyright protection of NURBS surfaces or solids bounded by NURBS surfaces. Algorithm 1 uses the maximum value at each test for a decision. Therefore, Algorithm 1 cannot supply partial similarity information between two objects. On the other hand, since Algorithm 2 is based on statistical methods for a decision, we can investigate which part is similar under a user-specified tolerance. Such partial similarity information can be used to determine whether part of an object has been stolen or not.

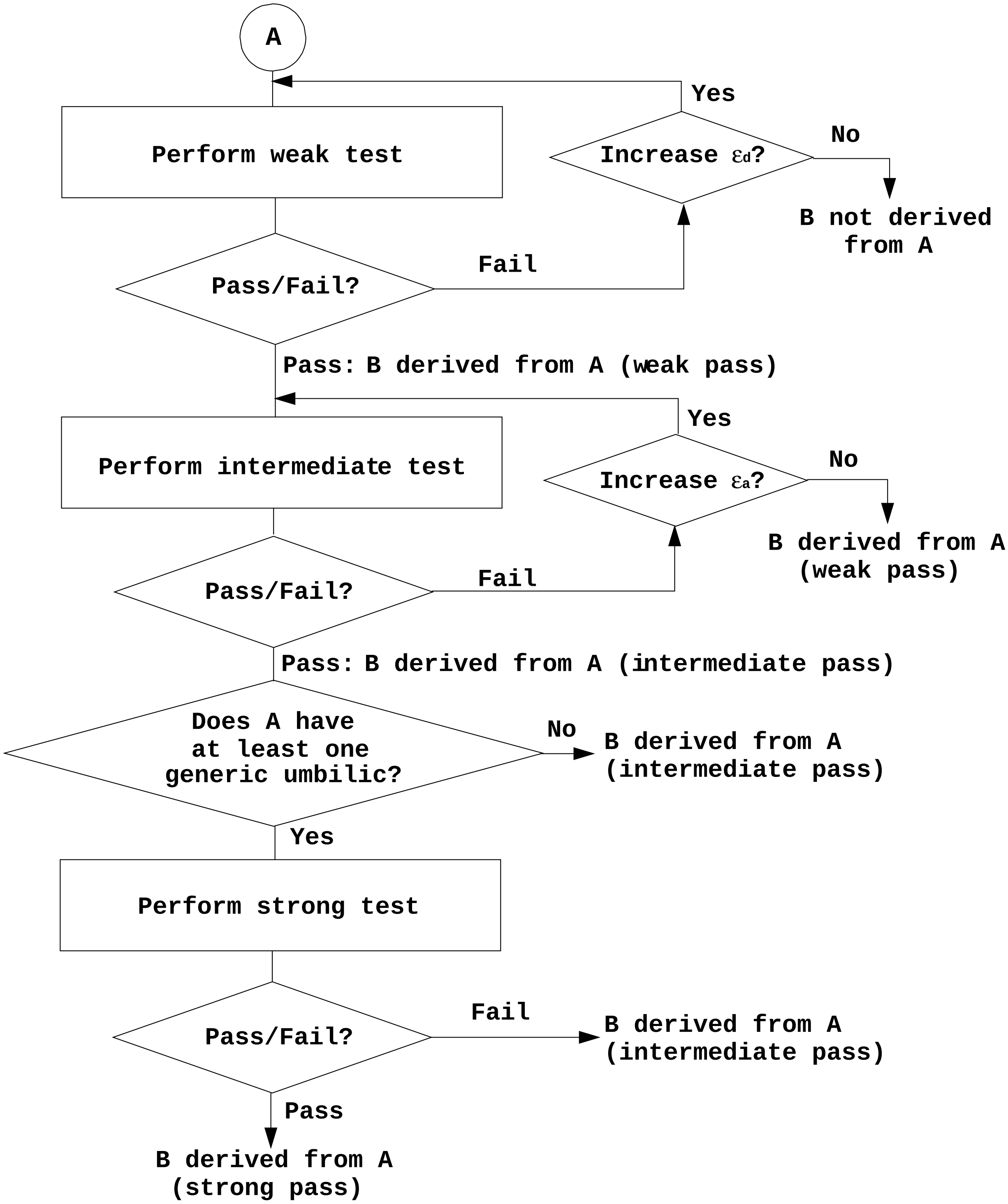


Figure 5-1: Algorithm 1 for similarity decision

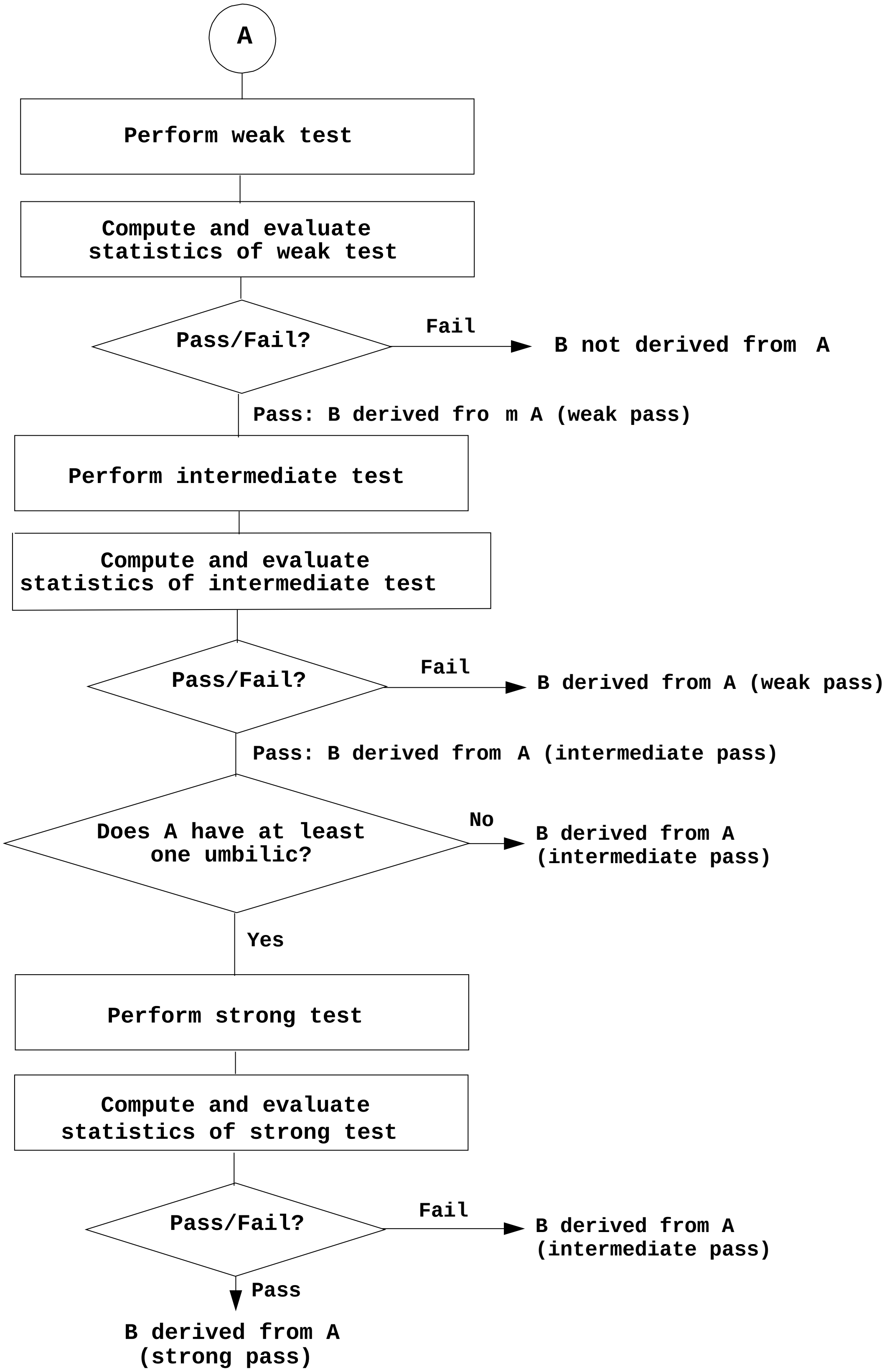


Figure 5-2: Algorithm 2 for similarity decision

# Chapter 6

# Examples and Applications

## 6.1 Object Matching

The proposed methods are tested with several numerical examples. Equations (2.25) and (4.6) are solved by using the IPP algorithm in interval arithmetic for robustness, and the rest of the calculation is done in double precision. A linux machine with 1.6 GHz CPU and 512 Mbytes was used for the calculations.

### 6.1.1 Moment Method

Solids bounded by bicubic integral B-spline surfaces, $A$ and $B$ are used. Solid $A$ is enclosed in a rectangular box of $25.0mm \times 23.48mm \times 11.0mm$. Here, the height of solid A is $25.0mm$. Figure 6-1 shows a sequence of operations for matching of the two solids using the principal moments of inertia of input solids. In this example, for better visualization, only part of the boundary surfaces of the solids is displayed. The smaller solid is reparameterized, translated, rotated and uniformly scaled. In Figure 6-1-(A), two boundary surfaces of the input solids are shown with their control points. Those two bounding surfaces have similar shape but different number of control points and parametrization. Matching the centroids of the two solids is performed by translating the small solid by the position difference between the centroids, which is demonstrated in Figure 6-1-(B). The orientation of the largest principal moment of inertia of the solid $A$ is aligned to that of the largest one of the solid $B$. Similarly, the remaining two orientations are aligned based on the values of the principal moments of inertia. After matching the orientations of the principal moments of inertia, the two solids are aligned in their orientations as shown in Figure 6-1-(C). Figure 6-1-(D) shows that

| Items | Solid A | Solid B |
|---|---|---|
| Volume ($mm^3$) | 83.794 | 18.007 |
| Center of Volume ($mm$) | $(2.689 \times 10^{-5}, 7.016 \times 16^{-5}, 16.907)$ | $(12.628, -8.354, 17.216)$ |
| Principal Moments of Inertia ($mm^5$) and direction cosines (x,y,z) | 1951.228, $(0.0, 0.0, 1.0)$ | 150.437, $(0.259, -0.330, 0.908)$ |
| | 4867.277, $(1.0, 0.0, 0.0)$ | 373.973, $(0.951, -0.076, -0.299)$ |
| | 6046.463, $(0.0, 1.0, 0.0)$ | 464.746, $(0.168, 0.941, 0.295)$ |

Table 6.1: Integral properties of solids A and B

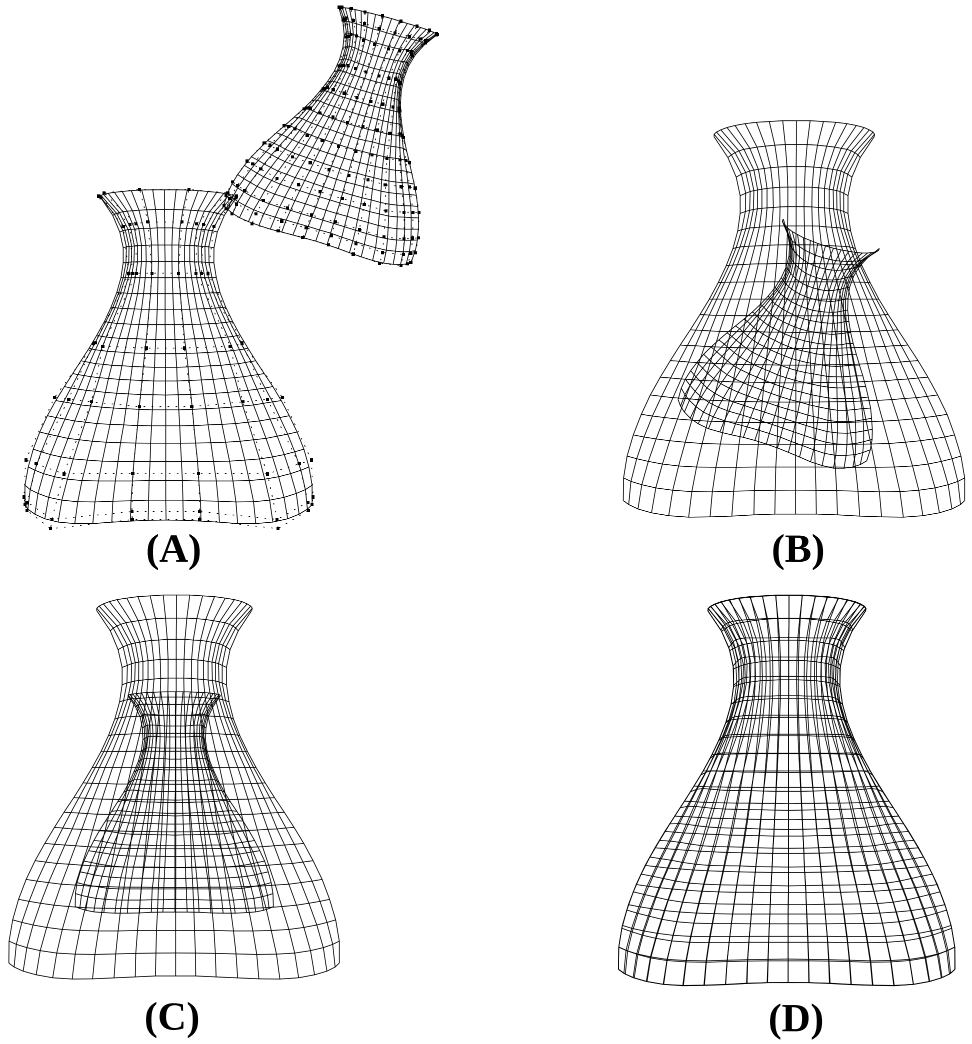


Figure 6-1: Matching via integral properties

the two solids match after a uniform scaling factor, 4.651, is applied to the small solid. The scaling factor is obtained from the ratio between the volumes of the two solids. The centroids, volumes, principal moments and their directions are summarized in Table 6.1.

### 6.1.2 Matching using Umbilics with Scaling Effects

In this section, matching through umbilical points is demonstrated with an example. Suppose we have a set of data points $\mathbf{r}_A$ and a surface $\mathbf{r}_B$. The surface $\mathbf{r}_B$ shown in Figure 6-2 is a bicubic B-spline surface with 64 $(8 \times 8)$ control points enclosed in a box of $25mm \times 23.48mm \times 11mm$. It has three star type umbilical points as shown in Figure 6-2, and the parametric values of the umbilical points in interval arithmetic from the IPP algorithm are summarized in Table 6.2. The elapsed time to calculate the umbilical points is 81 seconds. The center values of the interval roots representing umbilical points are given in Table 6.2 and the corresponding $\omega$ values in Table 6.3.

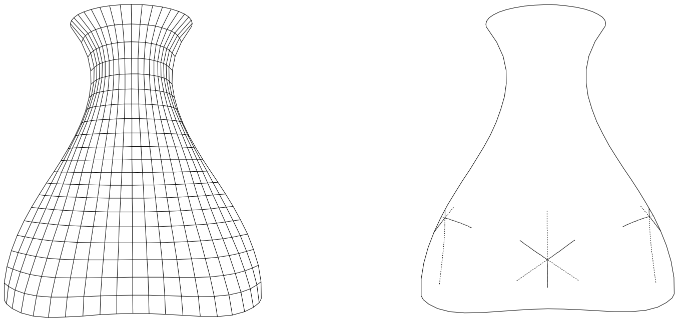


Figure 6-2: Surface $\mathbf{r}_B$ and its umbilics

| Calculated umbilics for Surface $\mathbf{r}_B$ using the IPP algorithm | |
|---|---|
| No. | (u,v) |
| 1 | ([0.748157559043998, 0.748157794338109], [0.0280931620263166, 0.0280933620263166]) |
| 2 | ([0.860717315099411, 0.860717550393523], [0.4999999, 0.5000001]) |
| 3 | ([0.748157559043993, 0.748157794338105], [0.971906637973675, 0.971906837973676]) |

Table 6.2: Umbilical points in interval arithmetic

The point set $\mathbf{r}_A$ shown in Figure 6-3 is approximated with a bicubic B-spline surface patch of 256 ($16 \times 16$) control points. It takes 20 seconds to obtain the approximated surface. It has one umbilical point of star type as shown in Figure 6-3. The root is ([0.207059775021701, 0.207059851944778], [0.684685549876914, 0.684685626799991]) in interval arithmetic and the elapsed time is 524 seconds. The center value of the interval and $\omega$ values are given in Table 6.4.

Each $\omega$ value is plotted in the complex plane as shown in Figure 6-4. We can find out that the umbilical point of $\mathbf{r}_B$ matches the number 2 umbilical point of $\mathbf{r}_A$ by comparing their complex $\omega$ values. Since correspondence has been found, a scaling factor can be estimated by using the normal curvatures at the corresponding umbilical points on both surfaces. The normal curvatures at the corresponding umbilical points of $\mathbf{r}_A$ and $\mathbf{r}_B$ are $\kappa_A = 0.334 \times 10^{-2}$ and $\kappa_B = 0.113 \times 10^{-2}$, respectively. So the scaling factor $\sigma$ can be calculated as $s = \frac{\kappa_A}{\kappa_B} = 2.941$. This scaling factor is applied to $\mathbf{r}_A$ and the surface is translated by the difference of the positions between the two corresponding umbilical points to get $\mathbf{r}'_A$. The next step is to align the normal vectors $\mathbf{n}_1$ for $\mathbf{r}'_A$ and $\mathbf{n}_2$ for $\mathbf{r}_B$. The alignment can be done by rotating $\mathbf{r}'_A$ by the angle between $\mathbf{n}_A$ and $\mathbf{n}_B$ around a vector $\mathbf{N}_\mathbf{n} = \frac{\mathbf{n}_A \times \mathbf{n}_B}{\|\mathbf{n}_A \times \mathbf{n}_B\|}$. The angle is 0.239 (rad)

| Surface $\mathbf{r}_B$ | | |
|---|---|---|
| No. | (u,v) | $\omega = (x + iy)$ |
| 1 | (0.748, 0.028) | $0.094 - 0.069i$ |
| 2 | (0.861, 0.5) | $0.151 - 0.261i$ |
| 3 | (0.748, 0.972) | $0.094 + 0.069i$ |

Table 6.3: Umbilics and $\omega$ values for $\mathbf{r}_B$

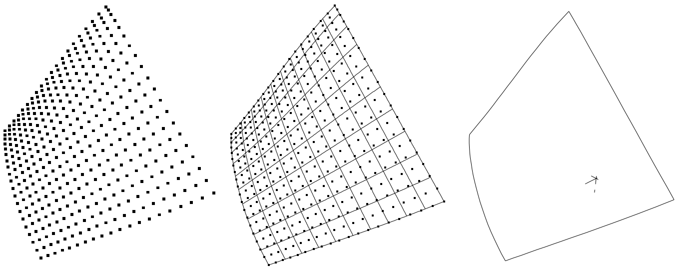


Figure 6-3: Approximated surface $\mathbf{r}_A$ and its umbilic

| Surface $\mathbf{r}_A$ | | |
|---|---|---|
| No. | (u,v) | $\omega = (x + iy)$ |
| 1 | (0.207, 0.685) | $0.151 - 0.261i$ |

Table 6.4: An umbilic and $\omega$ value for $\mathbf{r}_A$

for this example. In order to match lines of curvature passing umbilical points, one direction of the lines of curvature from the surface $\mathbf{r}_B$ is selected. Let us denote the selected direction vector as $\mathbf{v}_B$

$$\mathbf{v}_B = (-0.821, -0.021, -0.570). \tag{6.1}$$

Then three rotation angles between $\mathbf{v}_B$ and the directions in Table 6.5 at the umbilical points on $\mathbf{r}'_A$ are calculated as in Table 6.6. Here, the angles are measured from $\frac{\partial \mathbf{r}'_A}{\partial u}$ at the umbilical point. Matching the directions of lines of curvature is done by rotating $\mathbf{r}'_A$ around the normal vector in the tangent plane at the umbilical point. The rotation of $\mathbf{r}'_A$ by the first angle 1.105 *rad* yields the best match as shown in Figure 6-5. The relative measure of the maximum error can be calculated by dividing the maximum distance error by a square root of the surface area of the surface, which is 0.011.

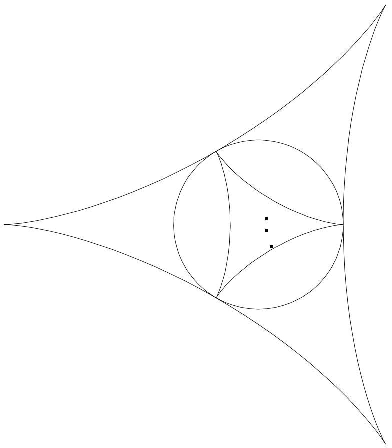


Figure 6-4: Umbilical points on the $\omega$-plane

| No. | angle (rad) | direction |
|---|---|---|
| 1 | 0.964 | (0.141, -0.001, -0.990) |
| 2 | -0.964 | (0.879, 0.017, 0.477) |
| 3 | $1.669 \times 10^{-15}$ | (0.893, 0.014, -0.450) |

Table 6.5: Angles and directions of lines of curvatures

|  | 1 | 2 | 3 |
|---|---|---|---|
| angle (rad) | 1.105 | 3.033 | 2.069 |

Table 6.6: Rotation angles for matching lines of curvature

### 6.1.3 Matching using Curvatures

#### Surface vs. Surface

In these examples, exact curvatures can be calculated so that $K_i$ and $H_i$ are provided with $\delta_K = \delta_H = 0$. Figure 6-6 shows an example of global matching. Two bicubic B-spline surfaces (enclosed in a box of $25.0mm \times 23.48mm \times 11.0mm$) are given, one of which is transformed through rotation and translation of the other. The surfaces have knot vectors

$$\begin{aligned} \mathbf{u} &= \{0,0,0,0,0.235,0.471,0.706,0.941,1,1,1,1\}, \\ \mathbf{v} &= \{0,0,0,0,0.2,0.4,0.6,0.8,1,1,1,1\}, \end{aligned} \tag{6.2}$$

and 64 ($8 \times 8$) control points. Three seed points are chosen on surface $\mathbf{r}_A$ in Figure 6-6. At those points, we get the result of Table 6.7 in double precision. Here, the center values of the interval roots are shown. In Table 6.7, $(u, v)$ are the parametric values, $(x, y, z)$ the Cartesian coordinates of the three points on $\mathbf{r}_A$, $K$ the Gaussian curvature, and $H$ the mean curvature. A surface $\mathbf{r}_B$ is subdivided into 25 Bézier surface patches, each of which is labeled by $\mathbf{r}_{Bi}$ $(i = 1, \cdots, 25)$. Four, ten and six candidate points are obtained for the point $\mathbf{n}_1$, $\mathbf{n}_2$ and $\mathbf{n}_3$. A value of $10^{-4}$ is used as the tolerance to the IPP algorithm [106, 91].

| Index | (u,v) | (x,y,z) ($mm$) | K ($mm^{-2}$) | H ($mm^{-1}$) |
|---|---|---|---|---|
| $\mathbf{n}_1$ | (0.20,0.45) | (-44.26,17.75,-15.10) | $-9.46 \times 10^{-3}$ | $-1.80 \times 10^{-1}$ |
| $\mathbf{n}_2$ | (0.58,0.68) | (-52.32,13.36,-17.36) | $-1.94 \times 10^{-3}$ | $-4.30 \times 10^{-2}$ |
| $\mathbf{n}_3$ | (0.91,0.14) | (-50.37,10.78,-31.83) | $1.04 \times 10^{-3}$ | $-6.14 \times 10^{-2}$ |

Table 6.7: Gaussian and mean curvature values for example 1

The next step is to sort out 3-tuples which have the same Euclidean distance constraints (4.8). After the correspondence between two 3-tuples on $\mathbf{r}_A$ and $\mathbf{r}_B$ is established using the equations (4.8), the rotation matrix and the translation vector

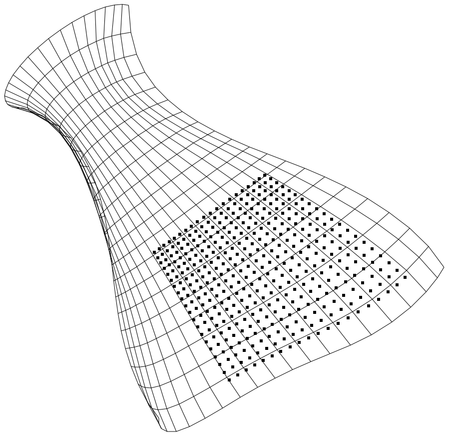


Figure 6-5: Localized points on the surface

are obtained as follows:

$$\begin{aligned}\mathbf{R} &= \begin{bmatrix} 0.62132 & 0.09854 & -0.77734 \\ -0.33704 & 0.92921 & -0.15159 \\ 0.70737 & 0.35618 & 0.61054 \end{bmatrix}, \\ \mathbf{t}_T &= [14.54754 \quad -29.14947 \quad 43.98560]^T. \end{aligned} \tag{6.3}$$

Applying $\mathbf{R}$ and $\mathbf{t}_T$ to $\mathbf{r}_A$ results in Figure 6-6(C), which shows a well localized surface. The relative measure of the maximum error can be calculated by dividing the maximum distance error by a square root of the surface area, which is 0.00533.

In Figure 6-7, a problem of partial matching is demonstrated. Two surface patches in B-spline form are extracted from surface C and transformed through rotation and translation, respectively. We can match both patches A and B onto the surface C as shown in Figure 6-7 through the proposed method. In Figure 6-7, the dark areas are the patches A and B after matching.

Two additional partial matching examples using the KH method are presented in Figures 6-8 and 6-9. Figure 6-8 is half of a car hood and Figure 6-9 is a traditional Japanese mask. Both are represented as bicubic B-splines. The hood has 64 ($8 \times 8$) control points (enclosed in a rectangular box of $13mm \times 12mm \times 6mm$) and the mask is constructed from $2\frac{1}{2}$D sensed data through the hierarchical approximation method by Lee *et al.* [64] with 1024 ($32 \times 32$) control points (enclosed in a rectangular box of $158mm \times 204mm \times 36mm$). The estimated rotation matrices and translation vectors as well as the relative maximum errors, i.e. the maximum distance divided by the square root of the surface area, are given in Tables 6.8 and 6.9.

## Point vs. Surface without Scaling Effects

Half of a fictitious car hood which is enclosed in a box of $10mm \times 12mm \times 6mm$ shown in Figure 6-10 is used for demonstration of the proposed method to solve a partial matching problem for the point vs. surface case. A set of data points is placed as shown in Figure 6-10. To imitate the behavior of a 3D scanner, the data points $\mathbf{p}$ are

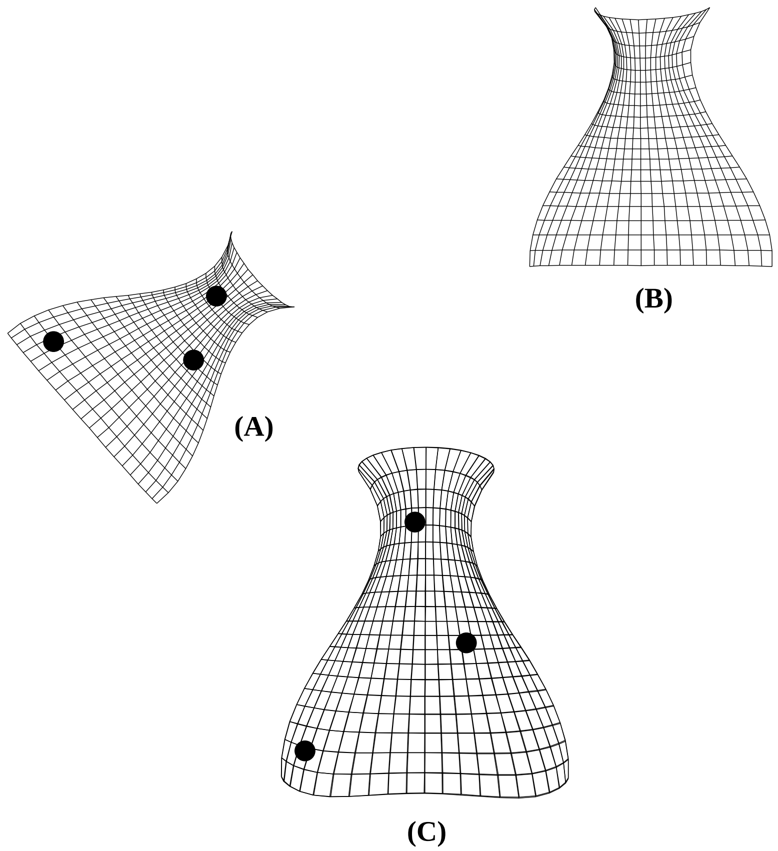


Figure 6-6: An example for global matching

sampled and perturbed randomly by the following equation with $\xi = 0.01$ [74, 91]:

$$\mathbf{p}' = \mathbf{p} + \xi \frac{(e_x, e_y, e_z)^T}{\sqrt{e_x^2 + e_y^2 + e_z^2}}, \tag{6.4}$$

where $e_x, e_y$ and $e_z$ are randomly chosen numbers which vary from -1 to 1 and $\xi$ is a constant that determines the amount of perturbations in the control points. Three points are selected, and three pairs of the Gaussian and the mean curvatures are estimated from the data points. Various curvature estimation methods can be adopted here. In this example, a bicubic B-spline surface with $16 \times 16$ control points is fitted over the points and then the curvatures are calculated. Values of 0.004 and 0.04

| Hood | Patch B ($\mathbf{r}_B$) |
|---|---|
| $\mathbf{R}$ | $\begin{bmatrix} 0.49457 & -0.68014 & -0.54113 \\ -0.28685 & 0.45999 & -0.84032 \\ 0.82044 & 0.57081 & 0.03240 \end{bmatrix}$ |
| $\mathbf{t}_T$ | $[1.28624 \quad -2.04411 \quad -0.27958]$ |
| Relative Max. Error | 0.00484 |

Table 6.8: Rotation matrix and translation vector for the hood

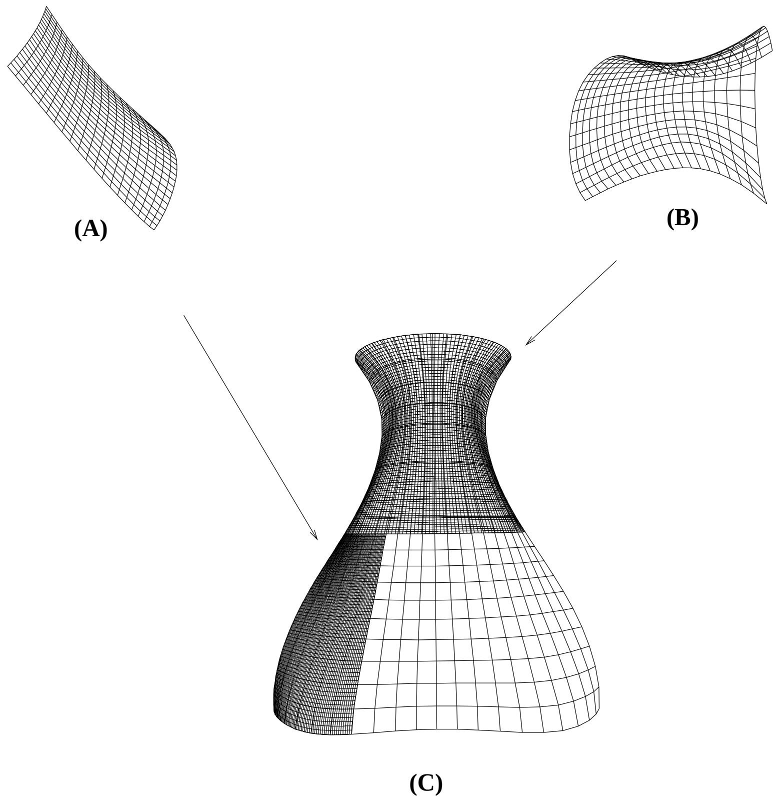


Figure 6-7: Examples for partial matching

| Mask | Patch A ($\mathbf{r}_A$) |
|---|---|
| $\mathbf{R}$ | $\begin{bmatrix} 0.32140 & -0.14531 & 0.93573 \\ -0.11698 & 0.97449 & 0.19151 \\ -0.93969 & -0.17101 & 0.29620 \end{bmatrix}$ |
| $\mathbf{t}_T$ | $[-6.73206 \quad -9.95848 \quad 7.44983]$ |
| Relative Max. Error | 0.00001 |

Table 6.9: Rotation matrix and translation vector for the mask

are used for the tolerances, $\delta_K$ and $\delta_H$, and a value of 0.01 is provided as a tolerance for the IPP algorithm. The tolerances $\delta_K$ and $\delta_H$ are selected by investigating errors between true values and estimated ones over the surface under the given perturbation. The numbers of candidate points are 2050, 1722 and 2870, for each selected point, respectively, and one triplet which yields a minimum value of the objective function is selected. The localized data points are shown in Figure 6-11. The relative maximum error, i.e. the maximum distance divided by the square root of the surface area is 0.0174.

This example is also tested via the ICP algorithm by Besl and McKay [10] for comparison purposes. The current position in Figure 6-10 is used as a starting state. The ICP algorithm converges to a local minimum which is shown in Figure 6-12. Obviously, the figure shows that the ICP algorithm fails in this situation, which

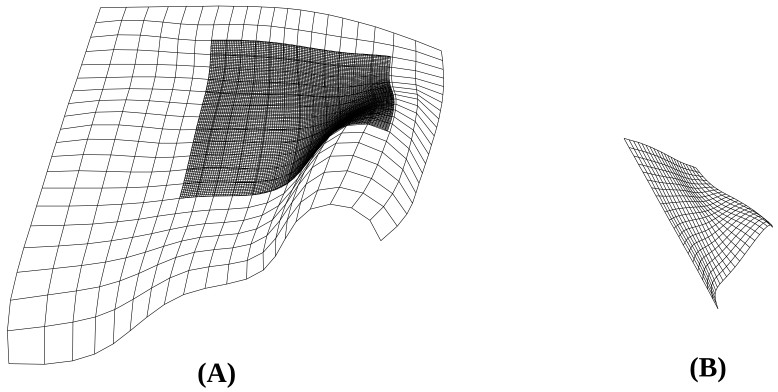


Figure 6-8: A localization example of a hood

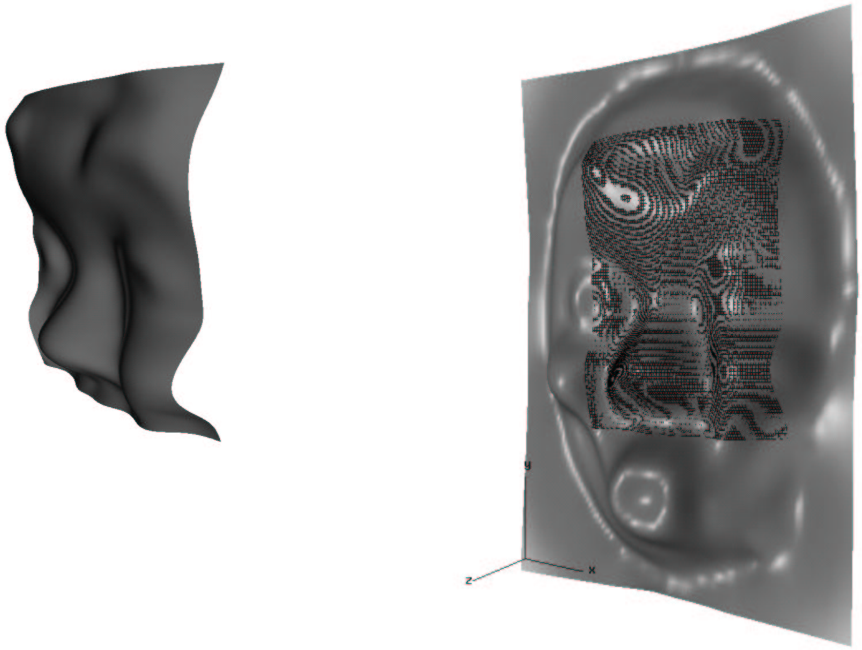
means that the success of the ICP algorithm heavily depends on the starting position or initial transformation. No such problems are encountered in the proposed method, which does not require good initial estimates.

## Point vs. Surface with Scaling Effects

A few examples are presented for demonstration of the optimization method for global and partial matching. The first example is a partial matching problem as shown in Figure 6-13. Data points in Figure 6-13-(B) have been scaled and transformed. A bicubic B-spline surface with 324 ($18 \times 18$) control points is used to approximate the data points to calculate the Gaussian and the mean curvatures at three seed points. The fitting step takes 99 seconds. The bottle surface shown in Figure 6-13-(A) is a bicubic B-spline surface with 64 ($8 \times 8$) control points, enclosed in a box of $25mm \times 23.48mm \times 11mm$. The problem here is finding a scaling factor and the rigid body transformation which make the two objects match as closely as possible. In this example, $\delta_{IPP} = 0.5$ was used for the IPP algorithm tolerance and $\delta_G = 0.001$ for the golden section search tolerance. For an initial interval for the golden section search, $[0.3, 1.0]$ was used. The Gaussian and the mean curvature functions are approximated by a bicubic B-spline surface with $256 \times 256$ control points and provided as input to the algorithm. The tolerances $\delta_K$ and $\delta_H$ of 0.0001 and 0.001 are used, respectively. After the optimization stops, the estimated scaling factor is 0.364, and the rotation matrix and the translation vector are summarized in Table 6.10. The matched surfaces are shown in Figure 6-13-(A) and the darker portion is surface B scaled and transformed. The relative measure of the maximum error can

| Rotation Matrix | Translation Vector |
|---|---|
| $\begin{bmatrix} 0.039 & 0.921 & 0.386 \\ -0.169 & 0.388 & -0.906 \\ -0.985 & -0.030 & 0.171 \end{bmatrix}$ | $[-32.090 - 53.595 - 12.391]$ |

Table 6.10: Estimated rigid body transformation for the first example



**(A)** **(B)**

Figure 6-9: A matching example of a mask

be calculated by dividing the maximum distance error by a square root of the surface area of the bottle, which is 0.0068.

The second example is a global matching problem. An artificial surface shown in Figure 6-14 is created and used for demonstration of the proposed algorithm. The surface shown in Figure 6-14-(A) is a bicubic B-spline surface with 400 ($20 \times 20$) control points enclosed in a box of $10mm \times 10mm \times 2.5mm$. The Gaussian and the mean curvature values are estimated at three selected seed points. An interval of $[0.3, 1.0]$ is used for an initial bracket for the optimization routine, and a value of 0.001 is used for the tolerance of the golden section search. The recovered scale value is 0.652, and the relative maximum error is 0.002. The localized result is presented in Figure 6-14-(A).

The third example is a partial matching problem using half of a fictitious automobile hood surface enclosed in a box of $13mm \times 12mm \times 6mm$. To imitate the behavior of a 3D scanner, the points $\mathbf{P}$ in Figure 6-15-(B) are disturbed by the equation (6.4) with $\xi = 0.01$ [91, 74]. An interval of $[0.3, 1.0]$ is used for an initial bracket for the optimization routine, and a value of 0.01 is used for the tolerance of the golden section search. The recovered scaling factor is 0.708, and the relative maximum error is 0.0005. The localized result is presented in Figure 6-15-(A).

The elapsed time of the optimization method (excluding the surface fitting step) depends on the tolerances of the golden section search and the IPP algorithm. The elapsed times for the examples under the given tolerances in this subsection are summarized in Table 6.11.

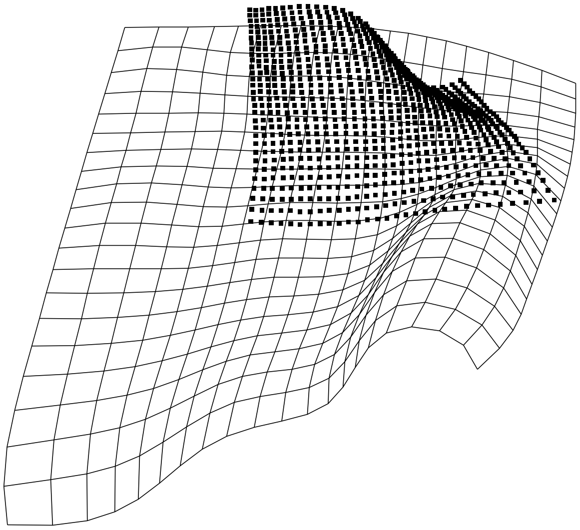


Figure 6-10: Initial position of the data points

| Examples | Times (sec) |
|---|---|
| Bottle Surface (Figure 6-13) | 3449 |
| Test Surface (Figure 6-14) | 1567 |
| Automobile Hood (Figure 6-15) | 907 |

Table 6.11: Elapsed times for the examples

## 6.2 Copyright Protection

In this section, the two similarity decision algorithms proposed in Chapter 5 are demonstrated with the bottle example in Section 6.1.1. After aligning the two solids $A$ and $B$ shown in Figure 6-1, we are ready to assess the similarity between them. Here, part of the bounding surfaces are used for similarity checking. The surfaces are represented as bicubic B-splines and one surface has 64 ($8 \times 8$) and the other 144 ($12 \times 12$) control points. Both surfaces are enclosed in a rectangular box of $25.0mm \times 23.48mm \times 11.0mm$. The 429 node points are used from the wireframe given in Figure 3-10. The statistical quantities are calculated and summarized in Table 6.12, and all umbilical points on the two surfaces are located as shown in Figure 6-16.

| Criteria | Max | Average | Standard Deviation |
|---|---|---|---|
| $\epsilon$-offset ($mm$) | 0.03456 | 0.00814 | 0.00665 |
| Maximum principal curvature ($mm^{-1}$) | 0.07872 | 0.01572 | 0.01530 |
| Minimum principal curvature ($mm^{-1}$) | 0.10577 | 0.01411 | 0.02165 |
| Principal direction (rad) | 0.70052 | 0.05657 | 0.11385 |

Table 6.12: Statistical quantities for matching tests

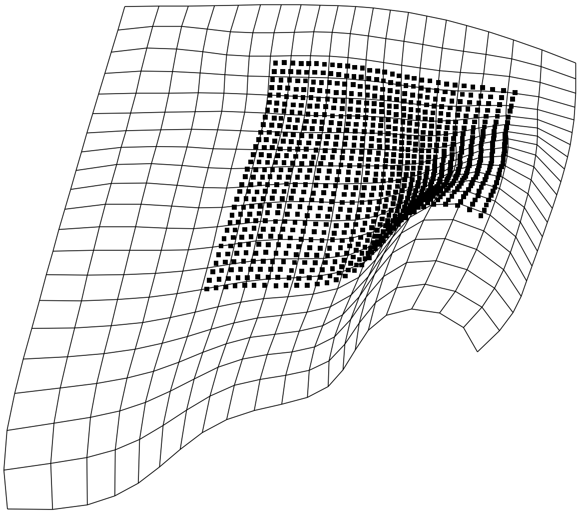


Figure 6-11: The localized data points

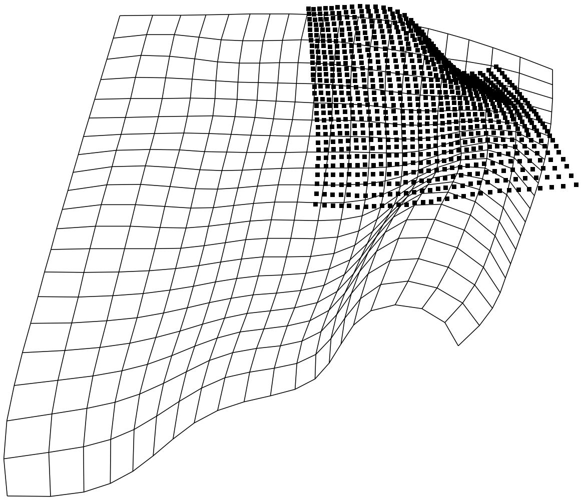


Figure 6-12: An example showing that the ICP algorithm may fail.

The Euclidean distances of the corresponding umbilical points are summarized in Table 6.13. In order to use Algorithm 1, we need to provide tolerances for $\epsilon_0$, $\epsilon_1$, $\epsilon'_1$ and $\epsilon_2$. Depending on each tolerance, we can determine which test has passed or failed.

Statistical information given in Table 6.12 is obtained for Algorithm 2. Unlike Algorithm 1 which uses the maximum value for each criterion, Algorithm 2 considers not only maximum values but also averages and standard deviations. Moreover, at each test, under a given tolerance, we can examine local similarity over the surface so that we can quantify the similarity between two surfaces and we can see which part is different. Suppose that we have 0.01 as a tolerance for the weak test. We subdivide the *uv* region into 400 square sub-regions (each box size of $0.05 \times 0.05$). The total number of sub-regions which contain footpoints $P_i$ satisfying $\epsilon_i > 0.01$ is 31. Therefore, we can conclude that two surfaces are similar by 92.25% under the weak

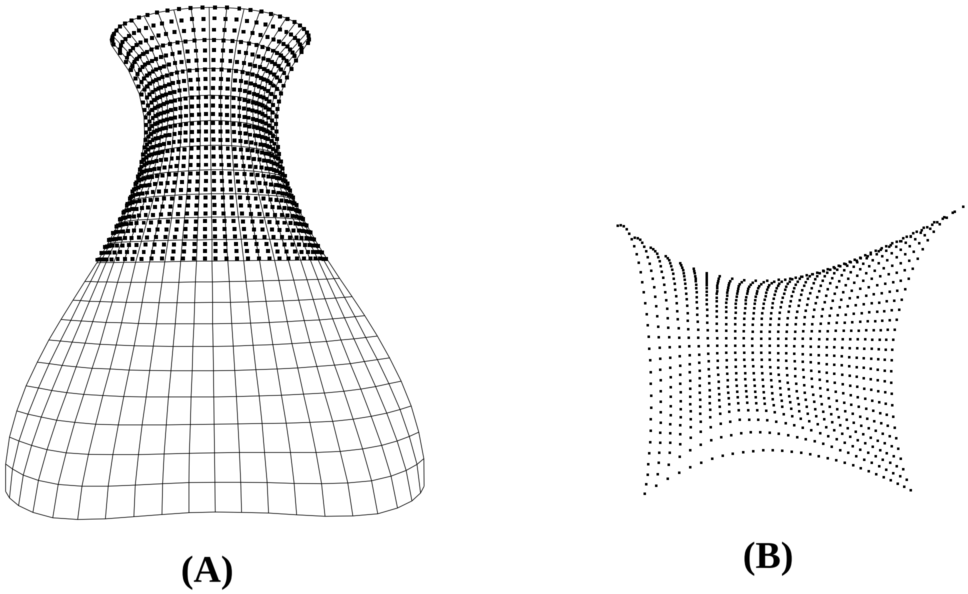


Figure 6-13: An example for partial surface matching with scaling effects using the optimization method

| | 1 | 2 | 3 |
|---|---|---|---|
| Distances ($mm$) | 0.08099 | 0.02954 | 0.08115 |

Table 6.13: Euclidean distances between the corresponding umbilics

test with a tolerance 0.01 and sub-region of size 0.05 × 0.05. This can be visualized as in Figure 6-17-(A). Here, the boxes indicate the regions which have more than one point with deviation larger than the tolerance 0.01. The intermediate test using the maximum principal curvature is visualized in Figure 6-17-(B). Under the intermediate test for the maximum principal curvature with a tolerance 0.03, the similarity between two surfaces is calculated to be at 91.25%. The results for the intermediate tests of the minimum principal curvature and the principal direction are shown in Figure 6-18-(A) and 6-18-(B). The similarity values for each case are 96.0% with a tolerance 0.03 and 95.50% with a tolerance 0.06.

The strong test can also be performed based on the umbilical points for both surfaces as shown in Figure 6-16. Three star type umbilical points are identified for each surface, and the Euclidean distances between the corresponding umbilical points are calculated as in Table 6.13. The types of the corresponding umbilical points match, and the position differences are small compared to the size of the object. Therefore, we may conclude that solid $B$ is derived from solid $A$ under the strong test.

The next example shows a case that one surface has been deformed so significantly that the strong test fails. Two surfaces in Figure 6-19-(A) and (B) are represented as a bicubic B-spline surface patch with 64 (8 × 8) control points, and enclosed in a rectangular box of $25mm \times 23.48mm \times 11mm$, respectively. The control points of surface $A$ as shown in Figure 6-19-(A) has been changed such that the difference of

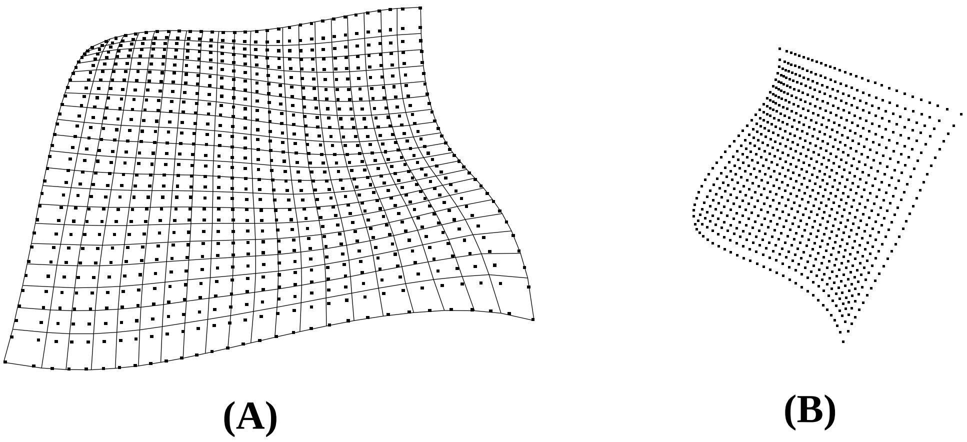


Figure 6-14: A test surface and target points

the bottom shape of surfaces $A$ and $B$ is noticeable as shown in Figure 6-19-(C). The surfaces are matched using the KH method with three seed points selected around the neck of surface $A$. From the wireframe of surface $A$ as shown in Figure 6-20, the 366 node points are obtained for similarity tests. The statistics information for the similarity tests are calculated and summarized in Table 6.14.

| Criteria | Max | Average | Standard Deviation |
|---|---|---|---|
| $\epsilon$-offset ($mm$) | 2.13435 | 0.26950 | 0.47164 |
| Maximum principal curvature ($mm^{-1}$) | 0.19392 | 0.00222 | 0.03114 |
| Minimum principal curvature ($mm^{-1}$) | 0.10979 | 0.01265 | 0.02135 |
| Principal direction (rad) | 1.54548 | 0.15166 | 0.27892 |

Table 6.14: Statistics for the matching tests

To assess local similarity, the $uv$ region is subdivided into 400 square boxes. Under the user-specified tolerances for each test, the quantitative similarity measures are calculated as in Table 6.15. Figures corresponding to each test are shown in Figure 6-21. The boxes indicate the regions where the condition for each test is not satisfied

| Test | Tolerance | Similarity (%) | Figure |
|---|---|---|---|
| $\epsilon$-offset ($mm$) | 0.48 | 86.50 | 6-21-(A) |
| Maximum principal curvature ($mm^{-1}$) | 0.04 | 88.50 | 6-21-(B) |
| Minimum principal curvature ($mm^{-1}$) | 0.02 | 89.00 | 6-21-(C) |
| Principal direction (rad) | 0.30 | 92.50 | 6-21-(D) |

Table 6.15: Quantitative similarity values

All umbilical points have been calculated using the IPP algorithm as shown in Figure 6-22, which are provided as input to the strong test. Surface $B$ has three star type umbilical points, whereas surface $A$ has two umbilics of star type. Because this is a global matching case and the number of umbilical points is different, it is concluded that the strong test fails.

The matching cases, Figure 6-7-(B), Figure 6-5, Figure 6-8 are used for partial surface similarity assessment. Let us call Figure 6-7-(B) case M1, Figure 6-5 case M2 and Figure 6-8 case M3. The wireframes of cases M1, M2 and M3 are shown in Figure 6-26. The statistics information and quantitative similarity values for case of the three cases are summarized in Table 6.16 and Table 6.19, respectively. No umbilical point exist for case M1. Therefore, the strong test cannot be applied to case M1. Umbilical points and lines of curvature passing through them are shown in Figure 6-27. The types and distances of the corresponding umbilical points for each case are summarized in Table 6.17 and Table 6.18. These cases are dealing with partial matching. Therefore, the comparsion region is limited to the region of the smaller surface. For case M2, the model surface has three star type isolated generic umbilical points and the transformed part has one star type isolated generic umbilical point. For case M3, the entire hood surface has 12 isolated generic umbilical points (7 star and 5 lemon umbilics) and there are 6 isolated generic umbilics (4 star and 2 lemon umbilics) on the transformed surface.

| Case M1 | | | |
|---|---|---|---|
| Criteria | Max | Average | Standard Deviation |
| $\epsilon$-offset ($mm$) | 0.00218 | 0.00071 | 0.00039 |
| Maximum principal curvature ($mm^{-1}$) | 0.00096 | 0.00007 | 0.00013 |
| Minimum principal curvature ($mm^{-1}$) | 0.00029 | 0.00003 | 0.00004 |
| Principal direction (rad) | 0.00055 | 0.00025 | 0.00012 |

| Case M2 | | | |
|---|---|---|---|
| Criteria | Max | Average | Standard Deviation |
| $\epsilon$-offset ($mm$) | 0.23407 | 0.10283 | 0.07036 |
| Maximum principal curvature ($mm^{-1}$) | 0.01378 | 0.00300 | 0.00264 |
| Minimum principal curvature ($mm^{-1}$) | 0.00939 | 0.00166 | 0.00181 |
| Principal direction (rad) | 0.09192 | 0.02701 | 0.01886 |

| Case M3 | | | |
|---|---|---|---|
| Criteria | Max | Average | Standard Deviation |
| $\epsilon$-offset ($mm$) | 0.01018 | 0.00472 | 0.00285 |
| Maximum principal curvature ($mm^{-1}$) | 0.00938 | 0.00153 | 0.00147 |
| Minimum principal curvature ($mm^{-1}$) | 0.00295 | 0.00085 | 0.00069 |
| Principal direction (rad) | 0.28397 | 0.00717 | 0.02630 |

Table 6.16: Statistics for the matching tests for case M1, M2 and M3

| No | Type | Distance |
|---|---|---|
| 1 | star | 0.15237 |

Table 6.17: The strong test for case M2

| No | Type | Distance |
|---|---|---|
| 1 | star | 0.01494 |
| 2 | lemon | 0.01491 |
| 3 | star | 0.01392 |
| 4 | star | 0.01426 |
| 5 | star | 0.014953 |
| 6 | lemon | 0.01494 |

Table 6.18: The strong test for case M3

| Case M1 | | | |
|---|---|---|---|
| Test | Tolerance | Similarity (%) | Figure |
| $\epsilon$-offset ($mm$) | 0.0014 | 98.75 | 6-23-(A) |
| Maximum principal curvature ($mm^{-1}$) | 0.00014 | 93.75 | 6-23-(B) |
| Minimum principal curvature ($mm^{-1}$) | 0.00006 | 96.00 | 6-23-(C) |
| Principal direction (rad) | 0.0004 | 96.00 | 6-23-(D) |

| Case M2 | | | |
|---|---|---|---|
| Test | Tolerance | Similarity (%) | Figure |
| $\epsilon$-offset ($mm$) | 0.19 | 94.25 | 6-24-(A) |
| Maximum principal curvature ($mm^{-1}$) | 0.0059 | 94.25 | 6-24-(B) |
| Minimum principal curvature ($mm^{-1}$) | 0.003 | 96.00 | 6-24-(C) |
| Principal direction (rad) | 0.05 | 93.25 | 6-24-(D) |

| Case M3 | | | |
|---|---|---|---|
| Test | Tolerance | Similarity (%) | Figure |
| $\epsilon$-offset ($mm$) | 0.009 | 98.00 | 6-25-(A) |
| Maximum principal curvature ($mm^{-1}$) | 0.003 | 96.25 | 6-25-(B) |
| Minimum principal curvature ($mm^{-1}$) | 0.0016 | 96.00 | 6-25-(C) |
| Principal direction (rad) | 0.012 | 98.50 | 6-25-(D) |

Table 6.19: Quantitative similarity values for case M1, M2 and M3

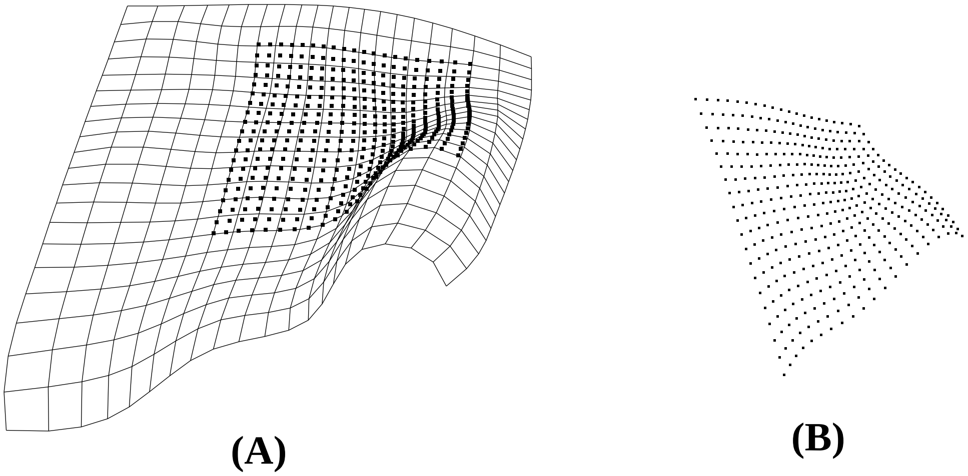


Figure 6-15: Matching of a fictitious automobile hood surface

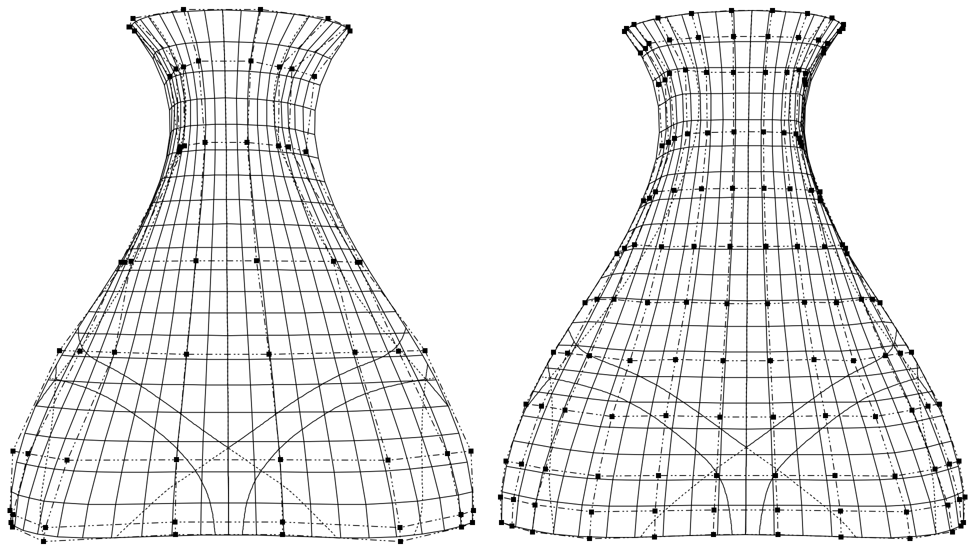


Figure 6-16: Comparison of lines of curvatures and umbilical points

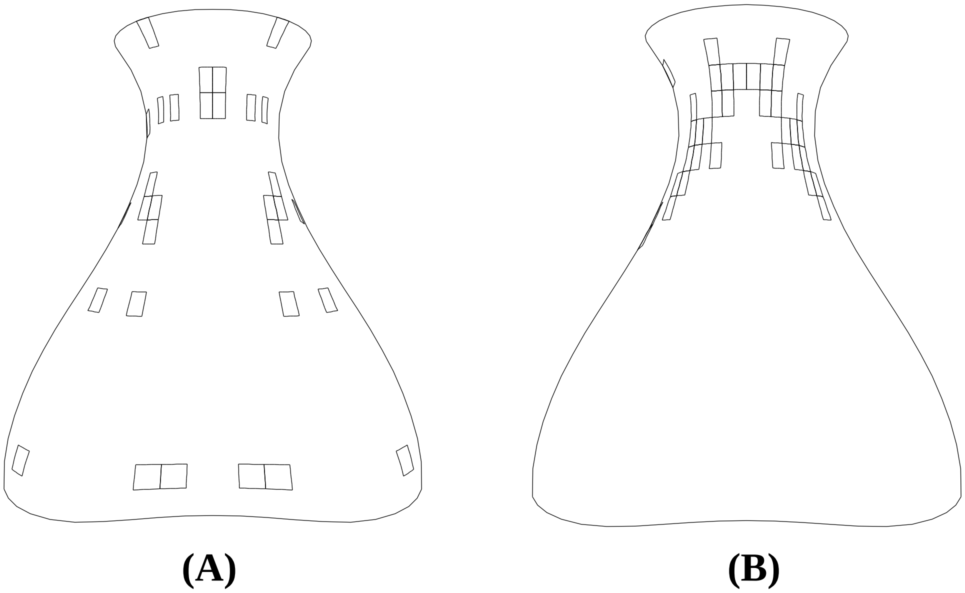


Figure 6-17: (a) Weak test ($\epsilon$-offset) and (b) Intermediate test (maximum principal curvature) based on Algorithm 2

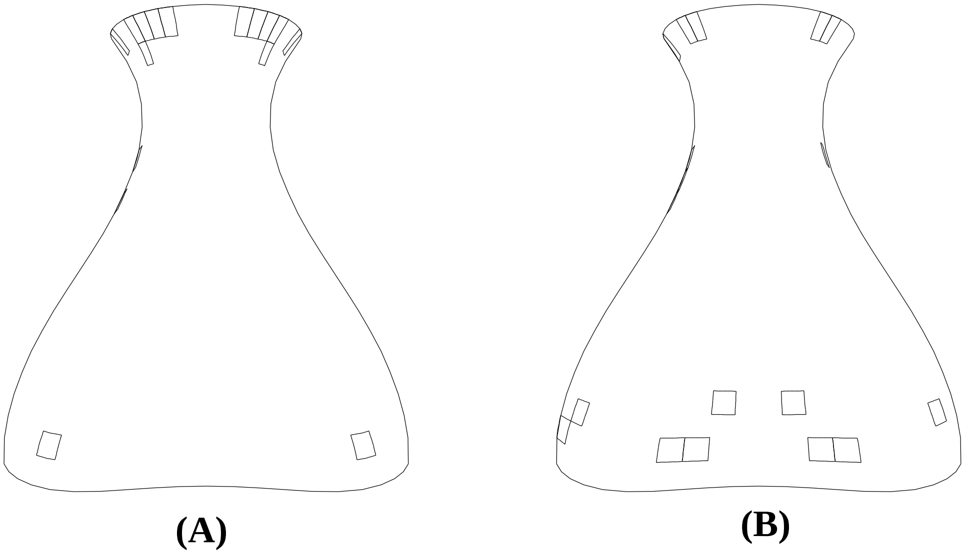


Figure 6-18: (a) Intermediate test (minimum principal curvature) based on Algorithm 2 and (b) Intermediate test (principal direction) based on Algorithm 2

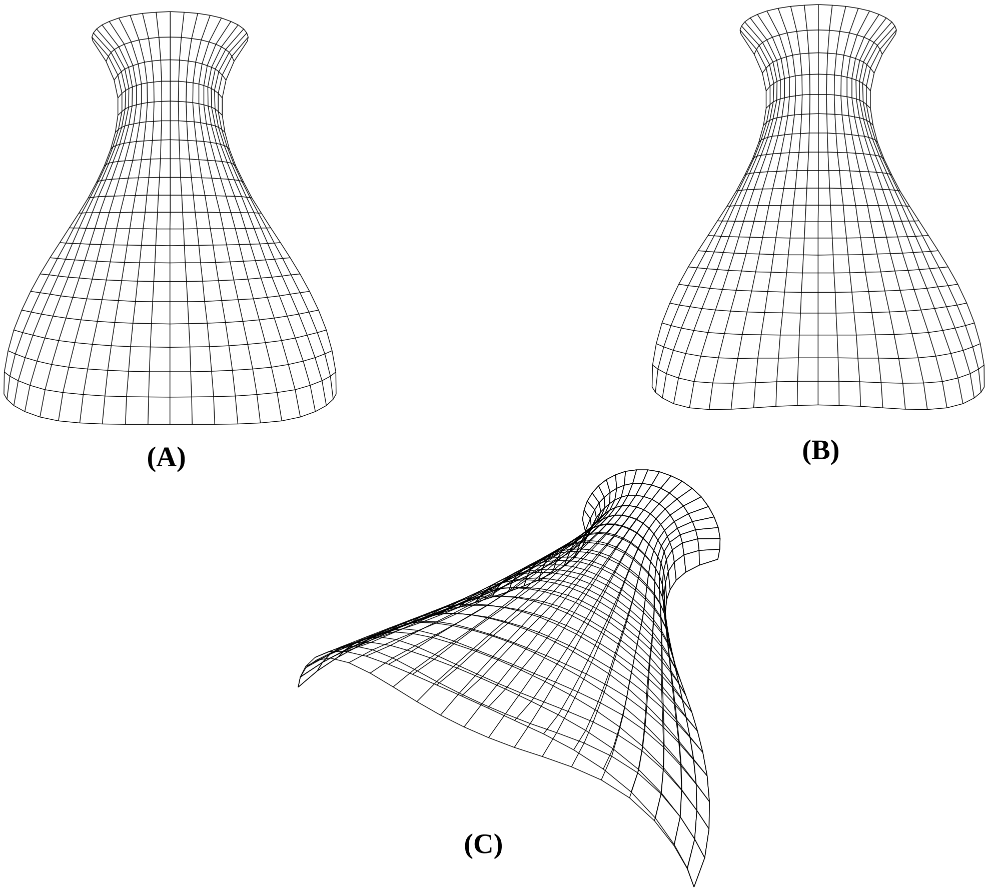


Figure 6-19: Surfaces for the failure case

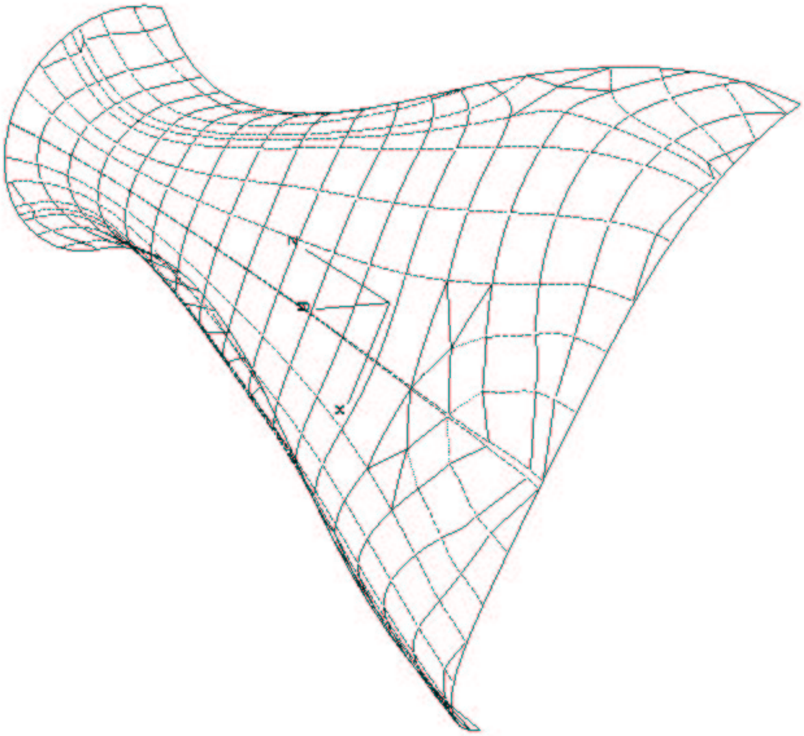


Figure 6-20: Wireframe of surface $A$

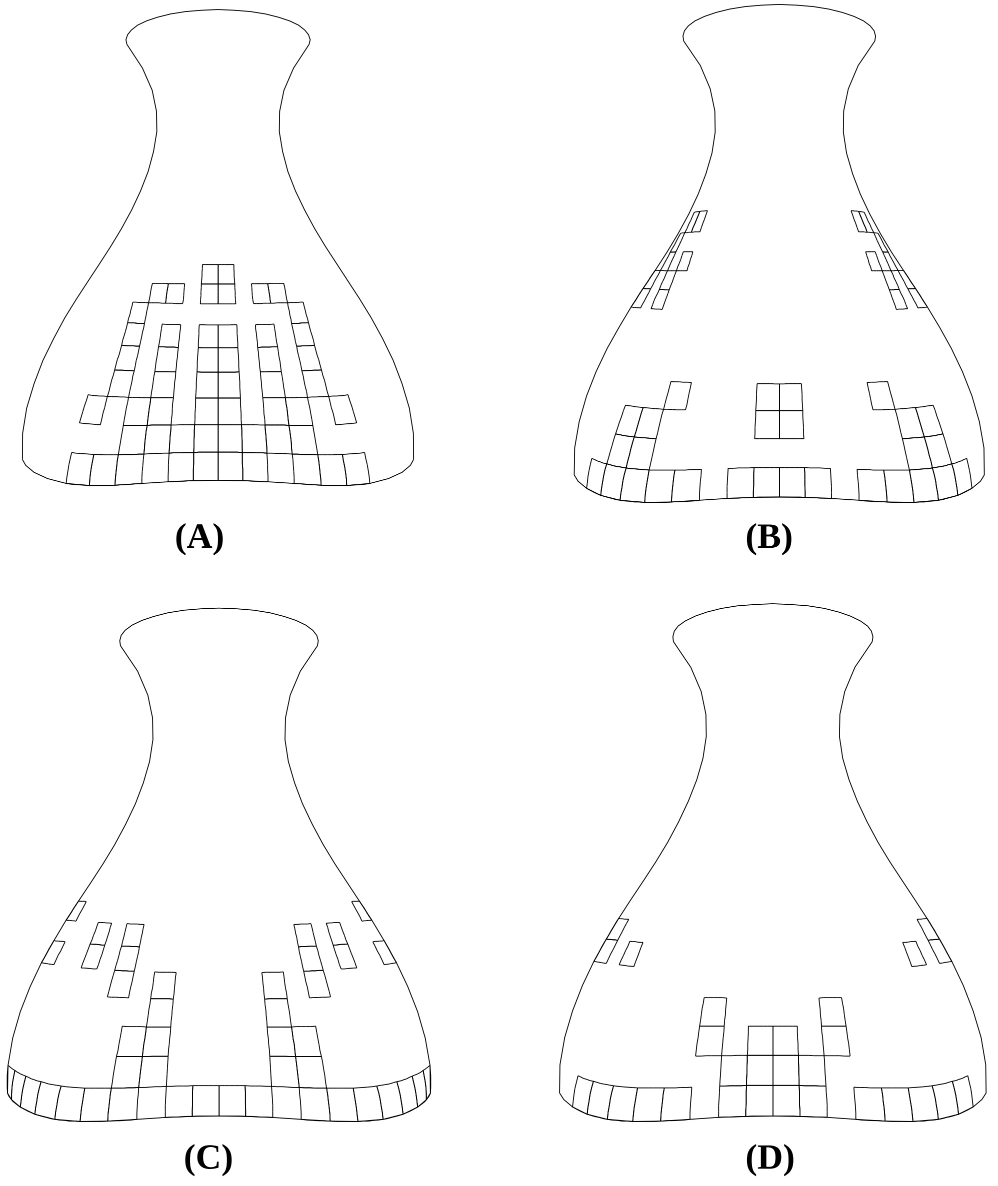


Figure 6-21: (A) $\epsilon$-offset (B) Maximum principal curvature (C) Minimum principal curvature (D) Principal direction

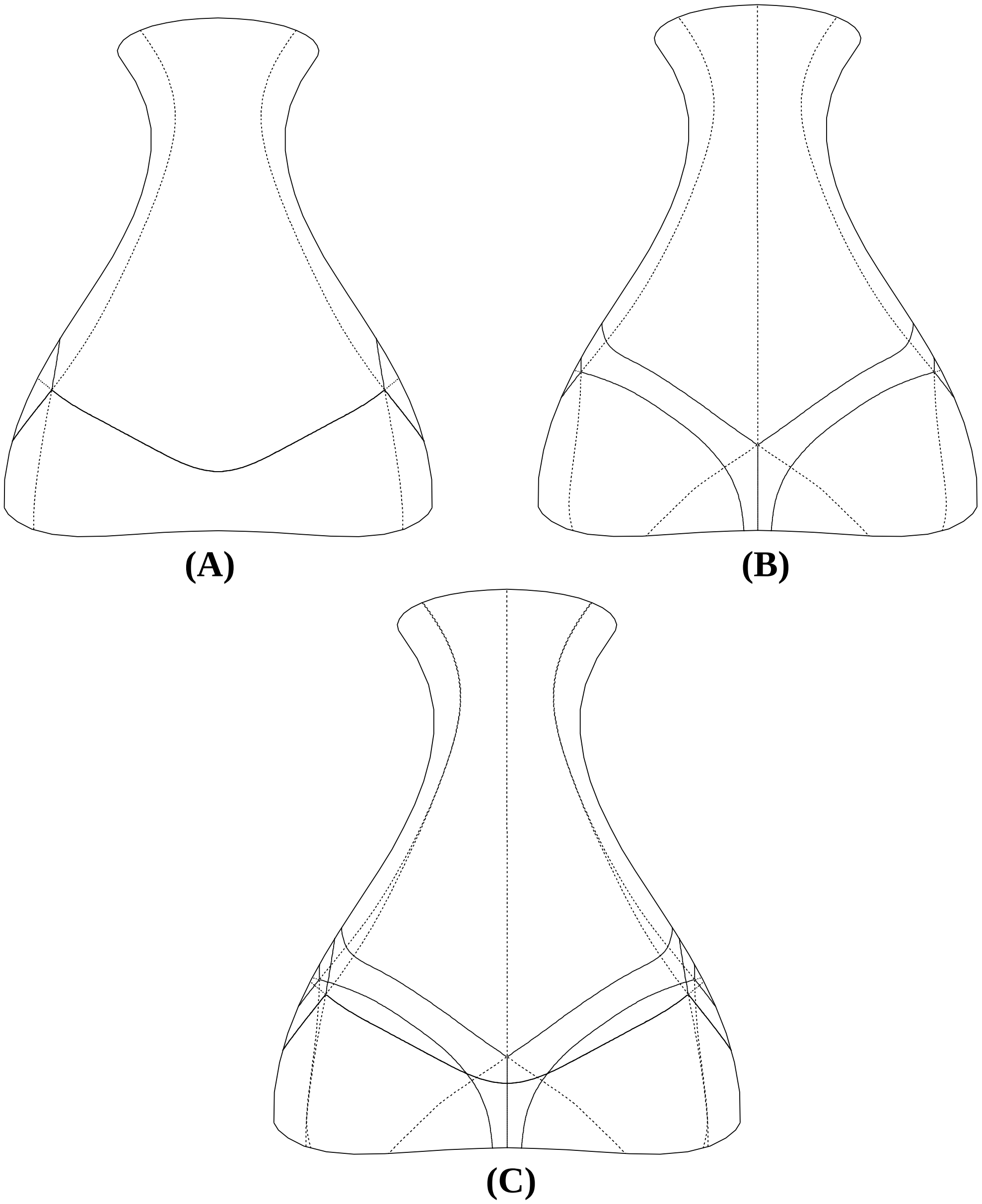


Figure 6-22: Umbilical points and lines of curvature

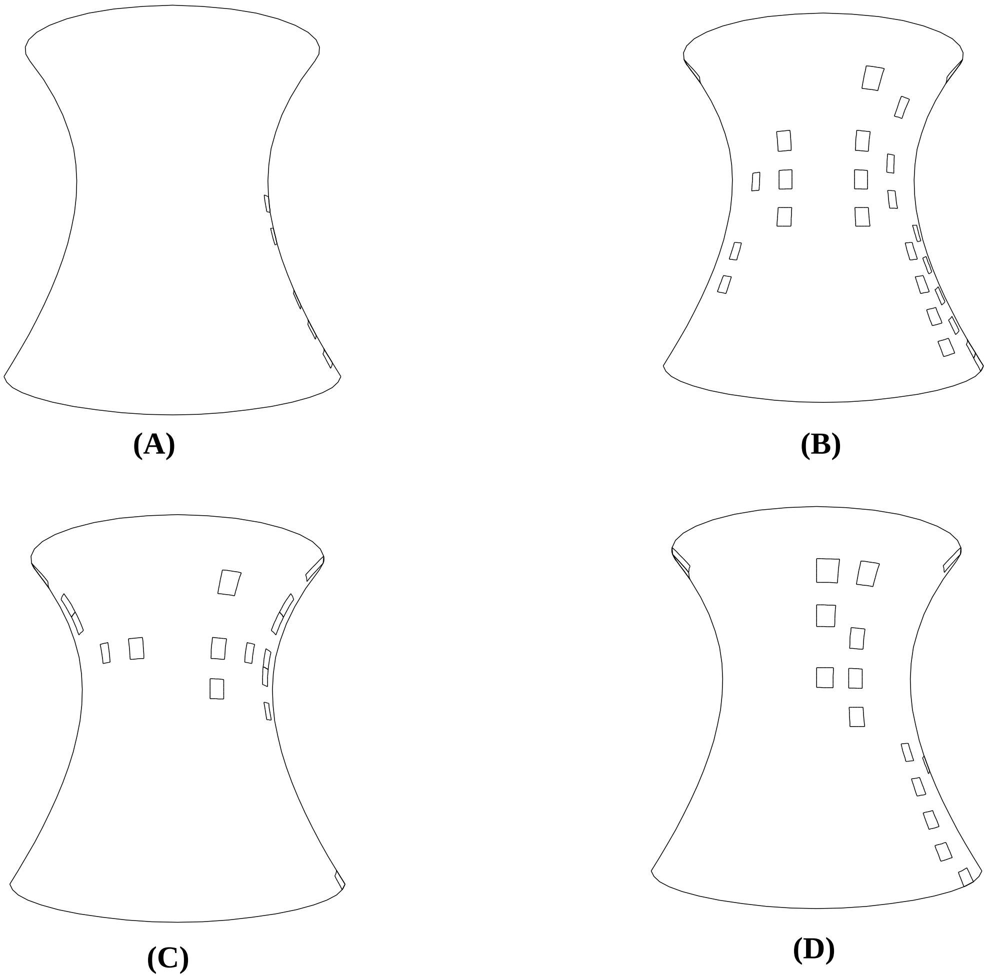


Figure 6-23: Case M1 : (A) $\epsilon$-offset (B) Maximum principal curvature (C) Minimum principal curvature (D) Principal direction

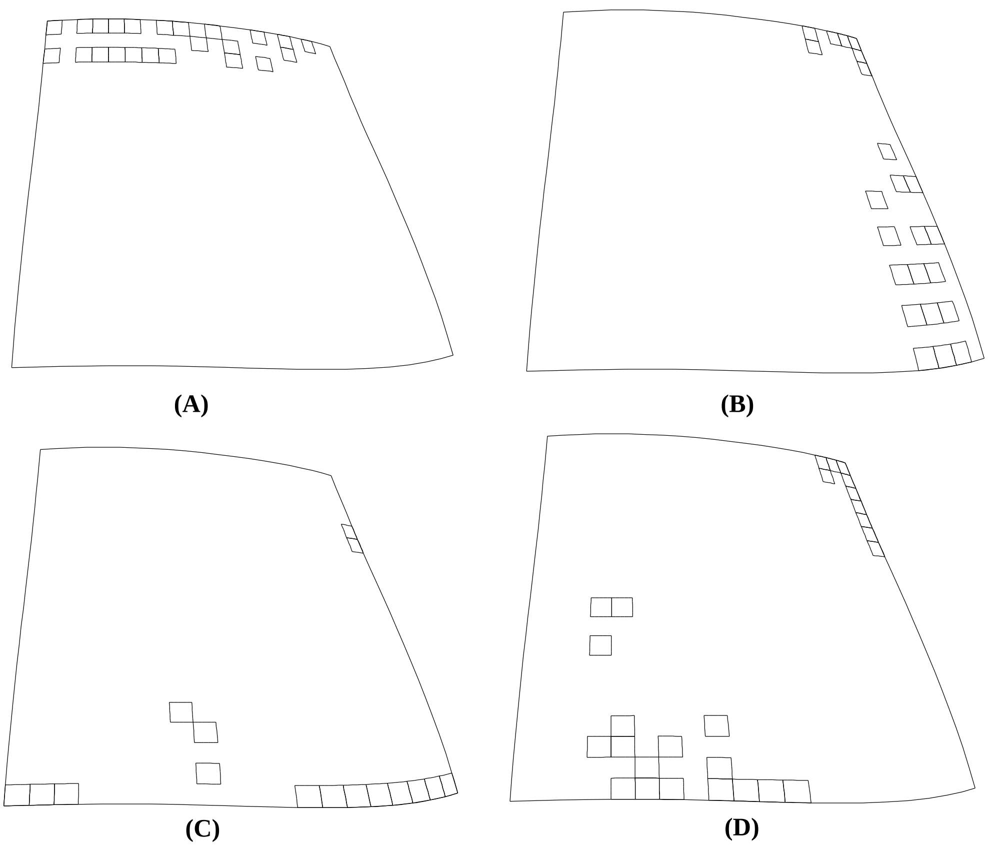
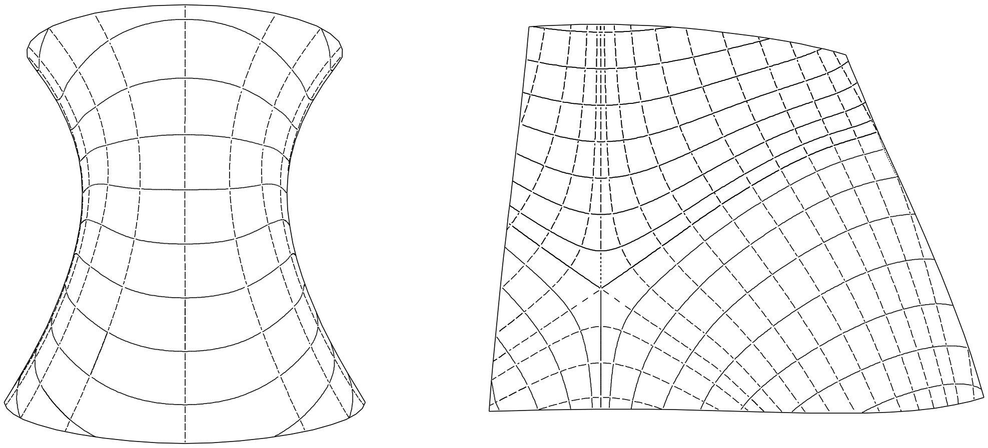


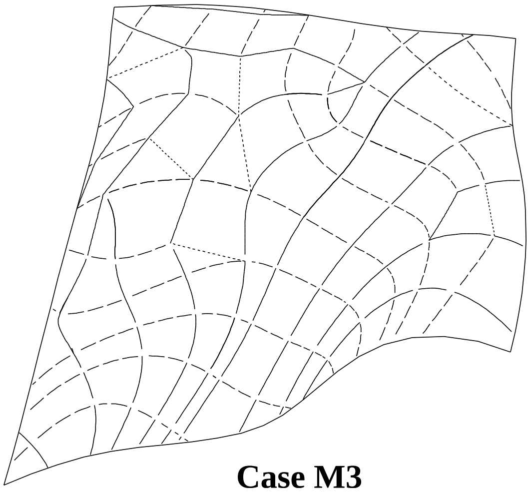
Figure 6-24: Case M2 : (A) $\epsilon$-offset (B) Maximum principal curvature (C) Minimum principal curvature (D) Principal direction

Figure 6-25: Case M3 : (A) $\epsilon$-offset (B) Maximum principal curvature (C) Minimum principal curvature (D) Principal direction



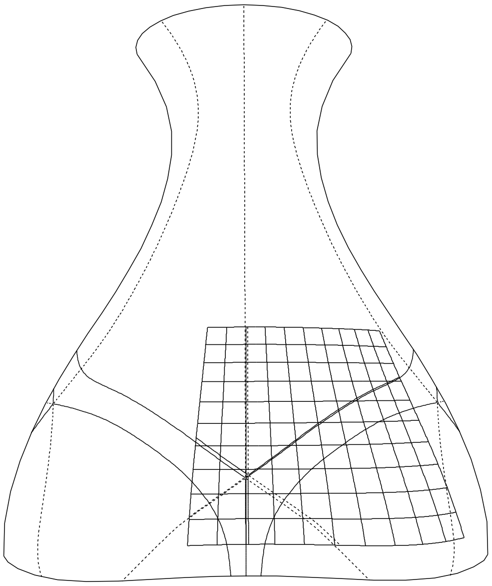
**Case M1**

**Case M2**

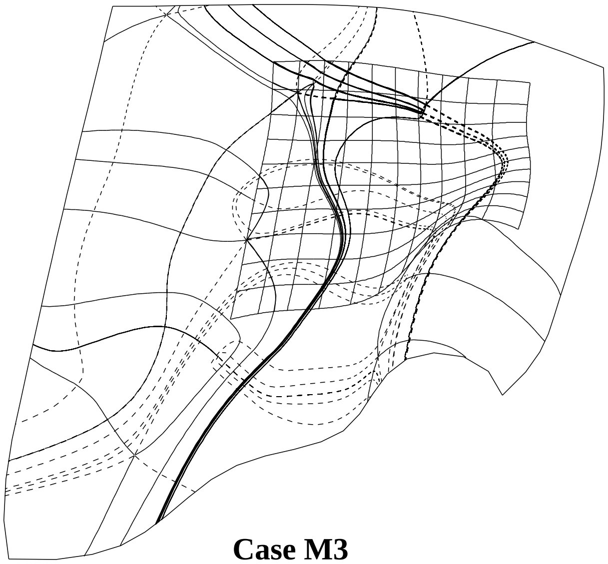


**Case M3**

Figure 6-26: Wireframes for case M1, M2 and M3



**Case M2**



**Case M3**

Figure 6-27: Umbilics for case M2 and M3

# Chapter 7

# Conclusions and Recommendations

## 7.1 Conclusions

In this thesis, we have addressed problems of free-form matching, assessment of similarity, and their applications to copyright protection. A matching problem includes global and partial object matching for the point vs. NURBS surface and the NURBS surface vs. NURBS surface cases with uniform scaling effects, and no *a priori* information on correspondence or initial transformation. The assessment of similarity consists of three tests (weak, intermediate and strong tests) which are organized in such a way that the two decision algorithms produce systematic and hierarchical results for a decision. The weak and intermediate tests compare the Euclidean distance, and the principal curvature and direction, respectively, at the node points obtained from the shape intrinsic wireframe. The strong test is performed when isolated generic umbilical points are available, which can be robustly calculated by the IPP algorithm.

It is well known that umbilical points are stable with respect to small noise, and their qualitative properties are maintained irrespective of scaling [74]. Therefore, umbilical points may behave as fingerprints for a 3D surface, which in turn can be used to uniquely identify the object for various purposes such as object recognition, tracking and copyright protection.

Accurate and robust calculation of various intrinsic properties is very important in this work. When a NURBS surface is provided, calculation of intrinsic properties can be done analytically. However, estimation of shape intrinsic properties from discrete points such as range data or piecewise linear polyhedral surface models is a difficult problem since complete local surface information cannot be recovered from the discrete points. For calculation of intrinsic properties such as umbilical points or lines of curvature, a least squares NURBS surface fitting method is used to obtain an approximated NURBS surface which is provided as input to the algorithms after high frequency noise has been smoothed out by a low pass filter. However, since there is no perfect method which can cover all cases, a different method needs to be employed for the estimation of differential properties depending on the quality of input data.

In the KH method, user-defined tolerances are provided in the governing equations (4.6) to deal with the uncertainty inherent to the estimation of curvatures from discrete data points. Accurate estimation of curvatures can improve the accuracy and performance of the KH method. If rough tolerances are used in equations (4.6), then the area satisfied by the equations is large so that more candidate points are generated, which leads to poor performance of the selection process in the KH algorithm. With more accurate curvature values, smaller tolerances can be used for equations (4.6) to improve the overall performance of the KH method. When a 3D scanner is used to digitize a physical surface, more dense data points can be obtained at three selected locations for the KH method by scanning the regions around those locations several times to estimate more accurate curvatures. Since the KH method does not require data of the entire surface, the expected performance loss due to the increased number of data points in the curvature estimation may be negligible and will not affect the performance of the KH method. On the other hand, rough tolerances for the equations (4.6) and the IPP algorithm may produce a result which can be effectively used as an initial state for iterative algorithms such as the ICP algorithm and its variants.

The optimization method proposed in this paper requires an initial interval of the scaling factor which has to be supplied by a user. However, compared to any iterative algorithms such as the ICP algorithm which needs 7 values (3 from translation, 3 from rotation and 1 from scaling), only one initial interval is enough for the optimization method because it incorporates the KH method which can handle a matching problem of NGWOS or NPWOS type. Therefore, the solution process to the matching problem can be simplified.

The overall performance of the optimization method can be improved by combining the golden section search and the secant method [30]. Near an optimum point where the size of the interval from the golden section search is small, the secant method may be used to reduce execution time using its higher convergence rate.

The proposed matching and similarity checking techniques can be used for copyright protection of NURBS surfaces or solids bounded by NURBS surface patches as demonstrated in Chapter 6. A user can compare a suspicious object with an object registered in an independent repository to check if the suspect is a copy of the copyrighted one. The partial matching technique may provide a way to determine whether or not part of the copyrighted object has been stolen.

## 7.2 Recommendations for Future Work

There are many cases where free-form objects represented by discrete data points have to be matched and compared. The proposed matching algorithms can deal with the point vs. surface or surface vs. surface matching cases so that they cannot be directly used for free-form data objects in discrete data points. One solution is to approximate the discrete data point with a surface and then use the surface for matching. However, surface approximation of data points itself is a difficult problem. Therefore, a method needs to be developed to use discrete data points directly in the matching process. Similarity assessment methods for free-form objects represented by discrete data points is another topic that needs to be discussed for future work.

Unlike the point vs. surface or surface vs. surface cases, not only the minimum distances but also intrinsic properties between two objects in discrete data points have to be estimated reasonably. Such extensions to cover the point vs. point or polygon vs. polygon case are necessary for the proposed matching algorithms to have more applications in computer vision, recognition and digital library.

The similarity decision algorithms require various user-defined tolerances for a decision. Those tolerances are very critical in making a decision if an object is a copy of another object. Therefore, the theoretical basis for determination of those tolerances needs to be established and evaluated by an independent party for unbiased results of the algorithms.

Currently, the algorithm for shape intrinsic wireframing is semi-automatic. In the areas near umbilical points where lines of curvature show singular behaviors and near the boundary of a surfaces where lines of curvature cannot form quadrilateral meshes, manual operations are required to complete the wireframe of the surface. One promising approach for the automation of shape intrinsic wireframing is to use geodesic curves exclusively. Construction of the geodesic curve is more flexible than that of the line of curvature because the geodesic curve depends on the choice of two boundary points, while the line of curvature is calculated from the principal direction field which is determined by the shape of the surface. Therefore, the algorithm for shape intrinsic wireframing can be made automatic by using geodesic curves, which is recommended for future work.

Analysis of non-generic umbilical points is recommended for a future research topic. In the current work, non-generic umbilical points are classified but are not used in the process. Even though non-generic umbilical points are unstable under small perturbations, understanding their properties and behaviors is significant to deal with them properly in applications. In relation to non-generic umbilical points, a study of the birth/death of generic umbilical points and its applications to matching is recommended for a future research topic.

In this work, two different types of arithmetic are used: rounded interval arithmetic and floating point arithmetic. Rounded interval arithmetic is exclusively used for the formulation and solution to the systems of nonlinear equations in the calculation of umbilical points and the KH method using the IPP algorithm. Floating point arithmetic is used for the other procedures. Since the current floating point representation method used by most of computer systems has inherent problems in representing floating point values, some operations in floating point arithmetic are not robust. Therefore, in order to maintain robustness and consistency in the calculation, it is recommended that the entire proposed algorithms be implemented in rounded interval arithmetic. There may exist many mathematical properties that are not clearly defined when interval arithmetic is used for analysis of umbilical points, calculation of geodesic curves and lines of curvature, orthogonal projection and the golden section search. Such uncertainties need to be resolved and properly handled in the context of rounded interval arithmetic.

# Appendix A

# Classification of Umbilical Points

In this appendix, theoretical analysis on classification of umbilical points is presented based on the work by Porteous [96], Bruce *et al.* [17], Giblin [46] and Maekawa *et al.* [74]. As described in Section 2.1.4, three characteristic lines are significant in understanding the behavior of local surface structure in relation to umbilical points.

## A.1 Cubic Form

In order to understand the relations between the characteristic lines and the cubic part $C(x, y)$ described in Section 2.1.4, we have to take a look at the structures and characteristics of a cubic form $V(x, y)$ defined by

$$V(x, y) = ax^3 + 3bx^2y + 3cxy^2 + dy^3. \tag{A.1}$$

We define the *Hessian* of (A.1) as follows:

$$H_e(x, y) = (ac - b^2)x^2 + (ad - bc)xy + (bd - c^2)y^2. \tag{A.2}$$

The *root line* of (A.1) is the set of points $\{(x, y)|V(x, y) = 0\}$. Similarly, the root line of (A.2) can be defined as the set of points $\{(x, y)|H_e(x, y) = 0\}$. The type of the cubic form (A.1) can be distinguished based on the root lines of (A.1) and (A.2) as in Table A.1 [96]. When the cubic form (A.1) has three distinct real roots or the Hessian has two complex roots, then the cubic form is classified as elliptic. Similarly, the cubic form is of parabolic or hyperbolic type depending on the root lines of the cubic form or the Hessian as summarized in Table A.1. The Hessian is important in the study of umbilical points since the root line directions of the Hessian of a cubic form close to an

| Type | Root lines of $V(x, y) = 0$ | Root lines of $H_e(x, y) = 0$ |
|---|---|---|
| Elliptic | three real distinct | Complex Conjugate |
| Parabolic | three real (two coincide) | real and coincide |
| Hyperbolic | one real and two complex | real distinct |

Table A.1: Classification of the cubic form adapted from [96] umbilical point approximate the principal directions near the umbilical point [46, 96] so that we can investigate the behavior of the lines of curvature without calculating the exact principal directions.

## A.2 Characteristic Lines vs. Cubic Form

### A.2.1 $\Gamma_1 : \theta \to \frac{1}{3}(2e^{i\theta} + e^{-2i\theta})$

The cubic form (2.30) is parabolic on the deltoid $\Gamma_1$ [96]. This implies that there are three root lines of (2.30) but two of them coincide. Inside the deltoid, the cubic form (2.30) is elliptic [96]. It is hyperbolic outside the deltoid. This classification is directly related to the number of ridge lines passing through an umbilical point, and the existence of extrema of the principal curvatures near the umbilical point [17, 46, 74]. Here, a ridge point is defined as an extremum point of a line of curvature, and a ridge line is a set of such points [46, 79]. A ridge line is also identified as a pre-image of a cuspidal edge on the focal surface [79]. Inside the deltoid, the number of ridge curves is three, the extrema of principal curvatures exist, and an umbilical point is of elliptical star type [17, 46]. On the other hand, outside the deltoid only one ridge curve passes through an umbilical point, no extremum of a principal curvature exists, and the umbilical point is of hyperbolic star type [17, 46].

### A.2.2 $|\omega| = 1$

On the circle $|\omega| = 1$, the cubic form (2.30) is *right-angled* [96]. When the root line directions of the Hessian (A.2) are mutually orthogonal, the cubic form (A.1) is *right-angled* [96]. This implies that the maximum and minimum lines of curvature are orthogonal at an umbilical point and form approximately a plain rectangular grid pattern [96]. This circle is related to the *index* [46]. The index is defined as an amount of rotation that a straight line segment tangent to lines of curvature experiences when rotating in the counterclockwise direction along a small closed path around an umbilical point [8, 74]. Inside the circle, the index is $-\frac{1}{2}$, and an umbilical point is of star type. Outside the circle, the index is $\frac{1}{2}$, and the umbilical point is classified as the lemon or monstar type [46, 74]. On the circle, the index value is zero, and the umbilical point is of non-generic type. This circle is also related to the birth/death of generic umbilical points under the evolution of a surface [96].

### A.2.3 $\Gamma_2 : \theta \to (2e^{i\theta} + e^{-2i\theta})$

Another cubic form, called the *Jacobian* cubic form, is defined to explain the deltoid $\Gamma_2$ as follows:

$$U(x, y) = bx^3 - (2c - a)x^2y - (2b - d)xy^2 - cy^3, \tag{A.3}$$

whose root lines are tangent to the lines of curvature near an umbilical point [46]. On the deltoid $\Gamma_2$, the cubic form (A.3) becomes parabolic [96]. The Jacobian cubic form (A.3) is related to the number of extrema of the cubic form (A.1) [96]. The cubic form (A.1) can be represented as $C(r, \theta)$ in polar coordinates with $x = r\cos\theta$ and $y = r\sin\theta$, and the expression of the direction in which the local extrema of the cubic form $C(r, \theta)$ occur, i.e. $\frac{dC(\theta)}{d\theta} = 0$ [74] is reduced to the Jacobian cubic form (A.3). Inside the deltoid $\Gamma_2$, there are three real root lines of the Jacobian cubic form (A.3) or three directions of the extrema of the cubic form (A.1), which implies that three lines of curvature converge to an umbilical point [46, 96]. This umbilical point is classified as star or monstar type [46, 96]. Outside the deltoid $\Gamma_2$, there is one root line of the Jacobian cubic form (A.3), and no extremum of the cubic form (A.1) exists. An umbilical point of this case is of the lemon type [46, 96].

## A.3 Inverse Transformation

One interesting aspect of the $\omega$ plane representation is the inverse process of mapping the cubic form (A.1) onto the $\omega$ plane. With a given $\omega$ value, we cannot recover the original coefficients of the cubic form (A.1) since the rotation angle and the scaling factor which are used in the map of the cubic form (A.1) onto the $\omega$ plane cannot be obtained. Instead, we can obtain the coefficients of the cubic form (A.1) which yield the same qualitative nature of an umbilical point and surface structure.

# Appendix B

# Formulation of Gaussian and Mean Curvatures

Suppose that we have an integral Bézier surface patch of degree $m$ and $n$ in $u$ and $v$ directions, respectively, as follows:

$$\mathbf{r}(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} \mathbf{p}_{ij} B_{i,m}(u) B_{j,n}(v), \tag{B.1}$$

where $u, v \in [0,1]$ and $\mathbf{p}_{ij}$ are the control points. Equations (4.6) can be represented by Bernstein polynomials using equations (2.11), i.e.

$$\begin{aligned} LN - M^2 - K_i(EG - F^2) &= 0, && \text{(B.2)} \\ 2FM - EN - GL - 2H_i(EG - F^2) &= 0, && \text{(B.3)} \end{aligned}$$

where $i = 1, 2, 3$. The first fundamental form coefficients can be obtained in Bernstein polynomial form using the first derivatives of $\mathbf{r}$ with respect to $u$ and $v$. The unit normal vector $\mathbf{N} = \frac{\frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v}}{\| \frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} \|}$, however, contains a square root term in the denominator. We can substitute the numerator for $\mathbf{n}_d$ and the denominator for $n_d$ which is equivalent to $\sqrt{EG - F^2}$. Then the second fundamental form coefficients can be expressed as follows:

$$\begin{aligned} L &= \mathbf{N} \cdot \frac{\partial^2 \mathbf{r}}{\partial u^2} = \frac{1}{n_d}(\mathbf{n}_d \cdot \frac{\partial^2 \mathbf{r}}{\partial u^2}) = \frac{l_s}{n_d}, && \text{(B.4)} \\ M &= \mathbf{N} \cdot \frac{\partial^2 \mathbf{r}}{\partial u \partial v} = \frac{1}{n_d}(\mathbf{n}_d \cdot \frac{\partial^2 \mathbf{r}}{\partial u \partial v}) = \frac{m_s}{n_d}, && \text{(B.5)} \\ N &= \mathbf{N} \cdot \frac{\partial^2 \mathbf{r}}{\partial v^2} = \frac{1}{n_d}(\mathbf{n}_d \cdot \frac{\partial^2 \mathbf{r}}{\partial v^2}) = \frac{n_s}{n_d}. && \text{(B.6)} \end{aligned}$$

Using the auxiliary variable method [91], we introduce $n_d$ as an auxiliary variable in the formulation. Then, the system of equations (B.2) and (B.3) becomes

$$\begin{aligned} n_d^2 - (EG - F^2) &= 0, && \text{(B.7)} \\ l_s n_s - m_s^2 - K_i n_d^4 &= 0, && \text{(B.8)} \\ 2m_s F - E n_s - G l_s - 2H_i n_d^3 &= 0. && \text{(B.9)} \end{aligned}$$

Equations (B.7)-(B.9) can be rewritten using the linear precision property as follows:

$$\begin{aligned} \sum_{i=0}^{4m-2} \sum_{j=0}^{4n-2} \sum_{k=0}^{2} \alpha_{ijk} B_{i,4m-2}(u) B_{j,4n-2}(v) B_{k,2}(\sigma) &= 0, && \text{(B.10)} \\ \sum_{i=0}^{6m-4} \sum_{j=0}^{6n-4} \sum_{k=0}^{4} \beta_{ijk} B_{i,6m-4}(u) B_{j,6n-4}(v) B_{k,4}(\sigma) &= 0, && \text{(B.11)} \\ \sum_{i=0}^{5m-3} \sum_{j=0}^{5n-3} \sum_{k=0}^{3} \gamma_{ijk} B_{i,5m-3}(u) B_{j,5n-3}(v) B_{k,3}(\sigma) &= 0, && \text{(B.12)} \end{aligned}$$

where $\sigma = \frac{n_d - a}{b - a}$ $(0 \leq \sigma \leq 1)$ is a scaled parameter of $n_d$ with $a \leq n_d \leq b$. Here $a$ and $b$ are square roots of the minimum and the maximum Bernstein coefficients of the expression $EG - F^2$, respectively. When the minimum of the Bernstein coefficient of $EG - F^2$ is negative, then we just set $a = 0$. The coefficients of equations (B.10)-(B.12) are provided as input to the IPP algorithm.

A rational Bézier surface patch is given as follows:

$$\mathbf{r}(u, v) = \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{p}_{ij} w_{ij} B_{i,m}(u) B_{j,n}(v)}{\sum_{i=0}^{m} \sum_{j=0}^{n} w_{ij} B_{i,m}(u) B_{j,n}(v)}, \tag{B.13}$$

where $u, v \in [0, 1]$, $\mathbf{p}_{ij}$ are the control points and $w_{ij}$ are the non-negative weights. If $w_{ij} = 1$, then equation (B.13) is reduced to (B.1). Let $\mathbf{r}$ be $\frac{\mathbf{r}_n}{r_d}$. Then we can derive equations (4.6) in a similar manner.

In the computer implementation, we evaluate the coefficients of the governing nonlinear polynomial equations in multivariate Bernstein form starting from the given input surface using the arithmetic operations in Bernstein form [91] executed either in rational or in rounded interval arithmetic.

# Bibliography

[1] 3D Systems, Inc. *Stereolithography Interface Specification*, June 1988.

[2] S. L. Abrams, W. Cho, C.-Y. Hu, T. Maekawa, N. M. Patrikalakis, E. C. Sherbrooke, and X. Ye. Efficient and reliable methods for rounded-interval arithmetic. *Computer-Aided Design*, 30(8):657–665, July 1998.

[3] P. G. Alourdas. *Shape Creation, Interrogation and Fairing Using B-Splines*. Engineer's thesis, Massachusetts Institute of Technology, Department of Ocean Engineering, Cambridge, Massachusetts, 1989.

[4] ANSI/IEEE Std 754–1985. *IEEE Standard for Binary Floating–Point Arithmetic*. IEEE, New York, 1985. Reprinted in *ACM SIGPLAN Notices*, 22(2):9-25, February 1987.

[5] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[6] O. Benedens. Geometry-based watermarking of 3D models. *IEEE Computer Graphics and Applications*, 19(1):46–55, 1999.

[7] R. Bergevin, D. Laurendeau, and D. Poussart. Estimating the 3D rigid transformation between two range views of a complex object. In *Proceedings of 11th IAPR International Conference on Pattern Recognition. Conference A : Computer Vision and Applications*, volume 1, pages 478–482, The Hague, The Netherlands, Aug. - Sep. 1992. IEEE Computer Society Press.

[8] M. V. Berry and J. H. Hannay. Umbilic points on Gaussian random surfaces. *Journal of Physics A.*, 10(11):1809–1821, 1977.

[9] P. J. Besl. The free–form surface matching problem. In H. Freeman, editor, *Machine Vision for Three–Dimensional Scenes*, pages 25–71. Academic Press, 1990.

[10] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[11] H. Bischof and A. Leonardis. Robust recognition of scaled eigenimages through a hierarchical approach. In *CVPR '98, Proceedings of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 664–670, Santa Barbara, CA, June 1998.

[12] A. Bjorck. *Numerical Methods for Least Squares Problems.* Society for Industrial and Applied Mathematics, Philadelphia, 1996.

[13] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

[14] W. Blaschke and K. Leichtweiss. *Elementare Differential Geometrie.* Springer-Verlag, Berlin, Heidelberg, New York, 1973.

[15] M. Brady, J. Ponce, A. Yuille, and H. Asada. Describing surfaces. *Computer Vision, Graphics and Image Processing*, 32(1):1–28, October 1985.

[16] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, December 1992.

[17] J. W. Bruce, P. J. Giblin, and F. Tari. Ridges, crests and sub-parabolic lines of evolving surfaces. *International Journal of Computer Vision*, 18(3):195–210, 1996.

[18] R. Campbell and P. J. Flynn. Model and range image features for free-form object recognition. In *Vision Interface 98*, pages 173–180, Vancouver, BC, June 1998.

[19] R. J. Campbell and P. J. Flynn. Eigenshapes for 3D object recognition in range data. In *CVPR '99, Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol 2*, pages 505–510, Fort collins, Colorado, June 1999.

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

[21] O. I. Camps, C.-Y. Huang, and T. Kanungo. Hierarchical organization of appearance-based parts and relations for object recognition. In *CVPR '98, Proceedings of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 685–691, Santa Barabra, CA, June 1998.

[22] C. C. Chen. Improved moment invariants for shape discrimination. *Pattern Recognition*, 26(5):683–686, 1993.

[23] J. L. Chen and G. C. Stockman. 3D free-form object recognition using indexing by contour features. *Computer Vision and Image Understanding*, 71(3):334–355, 1998.

[24] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *The Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 2724–2729, Sacramento, California, April 1991.

[25] C. S. Chua and R. Jarvis. 3D free-form surface registration and object recognition. *International Journal of Computer Vision*, 17(1):77–99, 1996.

[26] C. S. Chua and R. Jarvis. Point signatures: A new representation for 3D object recognition. *International Journal of Computer Vision*, 25(1):63–85, 1997.

[27] J. H. Chuang. A potential-based approach for shape matching and recognition. *Pattern Recognition*, 29(3):463–470, 1996.

[28] J. H. Chuang, J. F. Sheu, C. C. Lin, and H. K. Yang. Shape matching and recognition using a physically based object model. *Computers and Graphics*, 25:211–222, 2001.

[29] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, December 1997.

[30] G. Dahlquist and Å. Björck. *Numerical Methods.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1974.

[31] H. Delingette, M. Hebert, and K. Ikeuchi. A spherical representation for the recognition of curved objects. In *IEEE Proceedings, Fourth International Conference on Computer Vision*, pages 103–112, Berlin, Germany, April 1993. IEEE Computer Society.

[32] P. M. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1976.

[33] T. Dokken. Finding intersections of B-spline represented geometries using recursive subdivision techniques. *Computer Aided Geometric Design*, 2(1-3):189–195, September 1985.

[34] C. Dorai and A. K. Jain. View organization and matching of free-form objects. In *Proceedings of the IEEE International Symposium on Computer Vision*, pages 25–30, Coral Gable, Florida, November 1995.

[35] C. Dorai and A. K. Jain. Recognition of 3D free-form objects. In *Proceedings of the 13th International Conference on Pattern Recognition, Vol 1*, pages 697–701, Vienna, Austria, August 1996.

[36] C. Dorai and A. K. Jain. COSMOS-a representation scheme for 3D free-form objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1115–1130, October 1997.

[37] Z. Duric, N. F. Johnson, and S. Jajodia. Recovering watermarks from images. *Technical Report: ISE-TR-99-04*, the Center for Secure Information Syhstems, George Mason University, 1999.

[38] D. Dutta, R. R. Martin, and M. J. Pratt. Cyclides in surface and solid modeling. *IEEE Computer Graphics and Applications*, 13(1):53–59, January 1993.

[39] G. Elber and M.-S. Kim. Geometric shape recognition of freeform curves and surfaces. *Graphical Models & Image Processing*, 59(6):417–433, November 1997.

[40] R. T. Farouki. Graphical methods for surface differential geometry. In R. R. Martin, editor, *The Mathematics of Surfaces II*, pages 363–385. Clarendon Press, 1987.

[41] C. Fornaro and A. Sanna. Public key watermarking for authentication of CSG models. *Computer Aided Design*, 32(12):727–735, 2000.

[42] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, London, 1990.

[43] J. M. Galvez and M. Canton. Normalization and shape recognition of three-dimensional objects by 3d moments. *Pattern Recognition*, 26(5):667–681, 1993.

[44] GNU. *GNU MP : The GNU Multiple Precision Arithmetic Library, Edition 4.1*, May, 2002.

[45] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solution. *Numerische Mathematik*, 14:403–420, 1970.

[46] P. W. (editor) Hallinan. *Two and Three Dimensional Patterns of the Face.* A.K. Peters, Ltd., Wellesley, Massachusetts, 1999.

[47] M. Hebert, K. Ikeuchi, and H. Delingette. A spherical representation for recognition of free-form surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):681–689, July 1995.

[48] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America, Series A*, 4(4):629–642, 1987.

[49] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America, Series A*, 5(7):1127–1135, 1988.

[50] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design.* A. K. Peters, Wellesley, MA, 1993. Translated by L. L. Schumaker.

[51] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:179–187, 1962.

[52] C. Y. Huang, O. I. Camps, and T. Kanungo. Object recognition using appearance-based parts and relations. In *CVPR '97, Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 877–883, San Juan, Puerto Rico, June 1997.

[53] A. E. Johnson and M. Hebert. Recognizing objects by matching oriented points. In *CVPR '97, Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 684–689, San Juan, Puerto Rico, June 1997.

[54] N. F. Johnson, Z. Duric, and S. Jajodia. A role for digital watermarking in electronic commerce. *ACM Computing Survey*, To appear.

[55] T. Joshi and J. Ponce. HOT curves for modelling and recognition of smooth curved 3D objects. In *CVPR '94, Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 876–880, Seattle, Washington, June 1994.

[56] S. Kanai, H. Date, and T. Kishinami. Digital watermarking for 3D polygons using multiresolution wavelet decomposition. In *Proceedings of the Sixth IFIP WG 5.2/GI International Workshop on Geometric Modelling: Fundamentals and Applications, Tokyo, December, 1998*, pages 296–307, 1998.

[57] S. B. Kang and K. Ikeuchi. Determining 3-D object pose using the complex extended Gaussian image. In *CVPR '91, Proceedings of the 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 580–585, Lahaina, Maui, Hawaii, June 1991.

[58] S. B. Kang and K. Ikeuchi. The complex EGI: A new representation for 3-D pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7):707–721, July 1993.

[59] M. Kirby and L. Sirovich. Application of the Karhunen-Loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, January 1990.

[60] K. H. Ko, T. Maekawa, and N. M. Patrikalakis. An algorithm for optimal free-form object matching. *Computer Aided Design*, 35(10):913–923, September 2003.

[61] K. H. Ko, T. Maekawa, N. M. Patrikalakis, H. Masuda, and F.-E. Wolter. Shape intrinsic watermarks for 3D solids. In *Proceedings of the 2003 NSF Design, Service and Manufacturing Grantees and Research Conference*, Birmingham, Alabama, January 2003.

[62] K. H. Ko, T. Maekawa, N. M. Patrikalakis, H. Masuda, and F.-E. Wolter. Shape intrinsic fingerprints for free-form object matching. In G. Elber and V. Shapiro, editors, *8th ACM Symposium on Solid Modeling and Applications*, Seattle, WA, June 2003, In press.

[63] J. Krumm. Eigenfeatures for planar pose measurement of partially occluded objects. In *CVPR '96, Proceedings of the 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 55–60, San Francisco, CA, June 1996.

[64] S. Lee, G. Wolberg, and Y. S. Shin. Scattered data interpolation with multilevel B-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):228–244, 1997.

[65] A. Leonardis and H. Bischof. Dealing with occlusions in the eigenspace approach. In *CVPR '96, Proceedings of the 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 453–458, San Francisco, CA, June 1996.

[66] A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78(1):99–119, 2000.

[67] B. C. Li and J. Shen. Fast computation of moment invariants. *Pattern Recognition*, 20(8):807–813, 1991.

[68] S. X. Liao and M. Pawlak. On image analysis by moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):254–266, March 1996.

[69] Y. Ma and J. P. Kruth. Parametrization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Computer-Aided Design*, 27(9):663–675, 1995.

[70] T. Maekawa. Computation of shortest paths on free-form parametric surfaces. *Journal of Mechanical Design, Transactions of the ASME*, 118(4):499–508, December 1996.

[71] T. Maekawa and J. S. Chalfant. Computation of inflection lines and geodesics on developable surfaces. *Mathematical Engineering in Industry*, 7(2):251–267, 1998.

[72] T. Maekawa and N. M. Patrikalakis. Interrogation of differential geometry properties for design and manufacture. *The Visual Computer*, 10(4):216–237, March 1994.

[73] T. Maekawa, N. M. Patrikalakis, F.-E. Wolter, and H. Masuda. Shape-intrinsic watermarks for 3-D solids. MIT Technology Disclosure Case 9505S, September 2001. Patent Attorney Docket No. 0050.2042-000. Application pending, January 7, 2002.

[74] T. Maekawa, F.-E. Wolter, and N. M. Patrikalakis. Umbilics and lines of curvature for shape interrogation. *Computer Aided Geometric Design*, 13(2):133–161, March 1996.

[75] R. R. Martin. *Principal Patches for Computational Geometry.* PhD thesis, Pembroke College, Cambridge University, 1982.

[76] R. R. Martin. Principal patches - a new class of surface patch based on differential geometry. In P. J. W. Ten Hagen, editor, *Eurographics '83, Proceedings of the 4th Annual European Association for Computer Graphics Conference and Exhibition, Zagreb, Yugoslavia*, pages 47–55. Amsterdam: North-Holland, September 1983.

[77] R. R. Martin. Estimation of principal curvatures from range data. *International Journal of Shape Modeling*, 4(3 & 4):99–109, 1998.

[78] F. Mokhtarian. Silhouette-based isolated object recognition through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):539–544, May 1995.

[79] R. Morris. The sub-parabolic lines of a surface. In G. Mullineux, editor, *The Mathematics of Surfaces VI, Proceedings of the 6th IMA Conference on Mathematics of Surfaces VI*, pages 79–102, Oxford, UK, 1996. Clarendon Press.

[80] H. Murase and S. K. Nayar. Illumination planning for object recognition using parametric eigenspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1219–1227, December 1994.

[81] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[82] S. A. Nene and S. K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):989–1003, September 1997.

[83] M. Novotni and R. Klein. A geometric approach to 3D object comparison. In *International Conference on Shape Modeling and Applications, Shape Modeling International (SMI 2001)*, pages 167–175, Genova, Italy, May 2001. IEEE.

[84] K. Ohba and K. Ikeuchi. Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):1043–1048, September 1997.

[85] R. Ohbuchi, H. Masuda, and M. Aono. Watermarking three-dimensional polygonal models through geometric and topological modifications. *IEEE Journal on Selected Areas in Communications*, 16(14):551–560, May 1998.

[86] R. Ohbuchi, H. Masuda, and M. Aono. A shape-preserving data embedding algorithm for NURBS curves and surfaces. In *Proceedings of Computer Graphics International, CGI '99, June 1999*, pages 180–187. IEEE Computer Society, 1999.

[87] A. V. Oppenheim. *Applications of Digital Signal Processing*. Prentice Hall, Inc., Englewood Cliffs, N.J. 07632, 1978.

[88] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In *International Conference on Shape Modeling and Applications, Shape Modeling International (SMI 2001)*, pages 154–166, Genova, Italy, May 2001. IEEE.

[89] J. Owen. *STEP: An Introduction*. Information Geometers, Winchester, UK, 1993.

[90] N. M. Patrikalakis and L. Bardis. Localization of rational B-spline surfaces. *Engineering with Computers*, 7(4):237–252, 1991.

[91] N. M. Patrikalakis and T. Maekawa. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer-Verlag, Heidelberg, February 2002.

[92] J. Pegna and F.-E. Wolter. Surface curve design by orthogonal projection of space curves onto free-form surfaces. *Journal of Mechanical Design, ASME Transactions*, 118(1):45–52, March 1996.

[93] L. A. Piegl and W. Tiller. *The NURBS Book*. Springer, New York, 1995.

[94] L. A. Piegl and W. Tiller. Computing offsets of NURBS curves and surfaces. *Computer-Aided Design*, 31(2):147–156, February 1999.

[95] I. R. Porteous. Ridges and umbilics of surfaces. In R. Martin, editor, *The Mathematics of Surfaces II*, pages 447–458. Oxford University Press, 1987.

[96] I. R. Porteous. *Geometric Differentiation for the intelligence of curves and surfaces*. Cambridge University Press, Cambridge, 1994.

[97] E. Praun, H. Hoppe, and A. Finkelstein. Robust mesh watermarking. In *Proceedings of SIGGRAPH '99, Los Angeles, August 8-13, 1999*, pages 49–56. ACM, 1999.

[98] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1988.

[99] R. J. Prokop and A. P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *Graphical Models and Image Processing*, 54(2):438–460, 1992.

[100] F. A. Sadjadi and E. L. Hall. Three-dimensional moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2:127–136, 1980.

[101] T. Sakkalis, G. Shen, and N. M. Patrikalakis. Topological and geometric properties of interval solid models. *Graphical Models*, 63(3):163–175, 2001.

[102] H. Samet. Hierarchical data structures and algorithms for computer graphics. I. Fundamentals. *IEEE Computer Graphics and Applications*, 8:48–68, 1988.

[103] P. T. Sander and S. W. Zucker. Singularities of principal direction fields from 3-D images. In *IEEE Second International Conference on Computer Vision, Tampa Florida*, pages 666–670, 1988.

[104] P. T. Sander and S. W. Zucker. Inferring surface trace and differential structure from 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):833–854, September 1990.

[105] G. C. Sharp, S. W. Lee, and D. K. Wehe. ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):90–102, January 2002.

[106] E. C. Sherbrooke and N. M. Patrikalakis. Computation of the solutions of nonlinear polynomial systems. *Computer Aided Geometric Design*, 10(5):379–405, October 1993.

[107] H. Y. Shum, M. Hebert, and K. Ikeuchi. On 3D shape similarity. In *CVPR '96, Proceedings of the 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 526–531, San Francisco, CA, June 1996.

[108] S. S. Sinha. Differential properties from adaptive thin-plate splines. In B. C. Vemuri, editor, *Geometric Methods in Computer Vision*, pages 64–74, San Diego, CA, July 1991. SPIE, SPIE. Volume 1570.

[109] S. S. Sinha and P. J. Besl. Principal patches: A viewpoint-invariant surface description. In *IEEE International Robotics and Automation, Cincinnati, Ohio*, pages 226–231, May 1990.

[110] D. Skocaj and A. Leonardis. Robust recognition and pose determination of 3-D objects using range images in eigenspace approach. In *IEEE Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling*, pages 171–178, Québec, Canada, May 2001.

[111] F. Stein and G. Medioni. Structural indexing: Efficient 3-D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, February 1992.

[112] E. M. Stokely and S. Y. Wu. Surface parameterization and curvature measurement of arbitrary 3-D objects: Five practical methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):833–840, August 1992.

[113] D. J. Struik. *Lectures on Classical Differential Geometry.* Addison-Wesley, Cambridge, MA, 1950.

[114] T. Surazhsky and G. Elber. Matching free-form surfaces. *Computers & Graphics*, 25:3–12, 2001.

[115] J.-P. Thirion. The extremal mesh and the understanding of 3D surfaces. *International Journal of Computer Vision*, 19(2):115–128, 1996.

[116] T. M. Tucker and T. R. Kurfess. Newton methods for parametric surface registration. part i. theory. *Computer-Aided Design*, 35(1):107–114, 2003.

[117] S. T. Tuohy and L. Bardis. Low degree approximation of high degree B-spline surfaces. *Engineering with Computers*, 9(4):198–209, Winter 1993.

[118] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *CVPR '91, Proceedings of the 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, Lahaina, Maui, Hawaii, June 1991.

[119] M. W. Walker, L. Shao, and R. A. Volz. Estimating 3-D location parameters using dual number quaternions. *Computer Vision, Graphics and Image Processing: Image Understanding*, 54(3):358–367, 1991.

[120] B. L. Yeo and M. M. Yeung. Watermarking 3D objects for verification. *IEEE Computer Graphics and Applications*, 19(1):36–45, 1999.

[121] K. Yin, Z. Pan, J. Shi, and D. Zhang. Robust mesh watermarking based on multiresolution processing. *Computers & Graphics*, 25:409–420, 2001.

[122] M. F. Zakaria, L. J. Vroomen, P. J. A. Zsombor-Murray, and J. M. H. M. Van Kessel. Fast algorithm for the computation of moment invariants. *Pattern Recognition*, 20(6):639–643, 1987.

[123] D. Zhang and M. Hebert. Multi-scale classification of 3-D objects. In *CVPR '97, Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 864–869, San Juan, Puerto Rico, June 1997.

[124] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[125] J. Zhou, E. C. Sherbrooke, and N. M. Patrikalakis. Computation of stationary points of distance functions. *Engineering with Computers*, 9(4):231–246, Winter 1993.