박사학위논문

직교좌표변환을 이용한 **NURBS**곡면 생성방법 개발 및 응용

배석형 (裵 錫 亨)

산업공학과

한국과학기술원

2002

직교좌표변환을 이용한 **NURBS**곡면 생성방법 개발 및 응용

NURBS Surface Fitting Using Orthogonal Coordinate Transform and Applications

NURBS Surface Fitting Using Orthogonal Coordinate Transform and Applications

Advisor : Professor Byoung K. Choi
by
Seok-Hyung Bae

Department of Industrial Engineering
Korea Advanced Institute of Science and Technology

A thesis submitted to the faculty of the Korea Advanced Institute of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Industrial Engineering

Daejeon, Korea
2001. 11. 29.
Approved by

Professor Byoung K. Choi
Major Advisor

직교좌표변환을 이용한 **NURBS**곡면 생성방법 개발 및 응용

배석형

위 논문은 한국과학기술원 박사학위논문으로 학위논문심사위원회에서 심사 통과하였음.

2001 년 11 월 29 일

심사위원장 최 병 규 (인)

심사위원 염 봉 진 (인)

심사위원 신 하 용 (인)

심사위원 김 승 우 (인)

심사위원 김 덕 수 (인)

DIE 965181 배석형. Seok-Hyung Bae. NURBS Surface Fitting Using Orthogonal Coordinate Transform and Applications. 직교좌표변환을 이용한 NURBS곡면 생성방법 개발 및 응용. Department of Industrial Engineering. 2002. 73p. Advisor Prof. Byoung K. Choi. Text in English

Abstract

Constructing a CAD model from a physical model plays a key role in some product-development processes. To be presented is an “indirect” method of fitting NURBS-surfaces for *rotational-freeform shapes*: 1) Cloud-of-points data representing a rotational-freeform shape are transformed into an *orthogonal-coordinate* (spherical- or cylindrical-) system, 2) a single-valued B-spline surface is fitted to the transformed data, and 3) the resulting B-spline surface is converted to a 3D NURBS-surface by applying a *symbolic-product operation* with a *quadratic NURBS base-geometry*. Compared to the existing “direct” fitting methods, the proposed method has some distinctive advantages: It provides a natural means to *parameterization*, enables to recover “exact” NURBS geometry when the cloud-of-points data represent a “true” *surface-of-revolution*, and allows an easy *point-membership classification* for NURBS-bounded solid objects.

By combining with a “robust” local-coordinate system estimation method based on the *general-quadric surface approximation* for cloud-of-point data “arbitrarily” located in 3D space, the proposed method was successfully embedded to a research-purposed *reverse engineering software*, as one component of the *surface fitting engine*. In consequence, reverse engineering process becomes more efficient – the numbers of parametric surfaces and secondary-surface operations are reduced, and user interaction is also reduced. The proposed method can be used for an “intuitive” *scientific visualization* of empirical data acquired from spherical or cylindrical experiment set-ups. The basic idea of the proposed method – a rotational-freeform curve (or surface) can be interpreted as a product of a base geometry and a *scale function* – can be applied to some special shape (mechanical cam, pressure vessel, *etc.*) design.

Contents

List of Tables

List of Figures

1. Introduction

Physical models play a key role in some product-development processes. When designing an aesthetic product, physical models (made of clay or chemical wood) are employed for aesthetic evaluation [Yama1993]. Sometimes, designers start with existing products when designing a new product shape, which is known as (*shape*) *reverse engineering* [Vara1997]. In “custom-design” of sports wear and shoes, which is becoming quite popular, human body serves as a physical model. In these cases, the physical model is digitized to obtain *cloud-of-points data* (**COP-data**) and then a mathematical CAD model is constructed from the COP-data.

In order to construct a parametric surface from COP-data, a pair of parameter values have to be assigned first to each data point. This *parameterization* operation is critical to the quality of the fitted surface [Ma1995], but is not a trivial task for a *rotational-freeform shape* that one-to-one maps to a sphere or cylinder.

To be presented is an “indirect” method of fitting NURBS surface from COP-data representing a rotational-freeform shape, which provides a natural means to parameterization. The proposed method consists of three phases: 1) COP-data representing a rotational-freeform shape are transformed into an *orthogonal-* (spherical- or cylindrical-) *coordinate* system, 2) a *single-valued B-spline surface* of degree (*p, q*) is fitted to the transformed data, and 3) the single-valued B-spline surface is converted to a (parametric) NURBS surface by applying a *symbolic-product operation* [Pieg1997] with a *NURBS base-geometry*.

The NURBS-surface fitting method to be proposed enables to recover *exact NURBS geometry* for a true *surface-of-revolution* when the COP-data represent a “true” surface-of-revolution shape (which includes sphere and cylinder) [Pieg1987][Pieg1991]. There have been various proposed methods for fitting sphere/cylinder [Luka1997] and surface-of-revolution shapes [Pott1998a][Pott1998b][Lai2000]. These “specialized” methods are efficient only when we know that the COP-data on hand represent a sphere, cylinder or surface-of-revolution shape.

When the fitted NURBS-surface becomes a *face* of a *NURBS-bounded solid object*, the single-valued B-spline surface may be used as a point-membership classifier. Namely, a 3D point to be classified is transformed into the orthogonal-coordinate system of the fitted NURBS surface, and its “range” value is compared against the single-valued B-spline surface. *Point-membership classification* for a general NURBS-bounded solid object is not a trivial operation [Niel1993]. Point-membership classification for rotational freeform shapes has been studied quite extensively by Sanchez-Reyes [Sanc1990][Sanc1991][Sanc1992][Sanc1994]. The Sanchez-Reyes' method, which uses a *sinusoidal-spline function* as the basis function, is similar to the method proposed in this study and may be used as a NURBS-surface fitting method for a sphere-like shape (but not for cylinder-like shape because its single-valued function is not a tensor-product surface).

The organization is as follows: described in Chapter 2 is a NURBS-curve fitting method using polar-coordinate transform, which is the core procedure for the NURBS-surface fitting methods to be presented in Chapter 3. Also presented in Chapter 3 are special treatments including closed surfaces and pole degeneracy problem. As a main use of the proposed method, (shape) reverse engineering application is considered in Chapter 4 where the estimation of a local-coordinate system is especially focus on. Other applications including special shape designs and scientific visualization are given in Chapter 5. Finally, Chapter 6 closes this dissertation with discussions and conclusions.

2. NURBS-curve Fitting

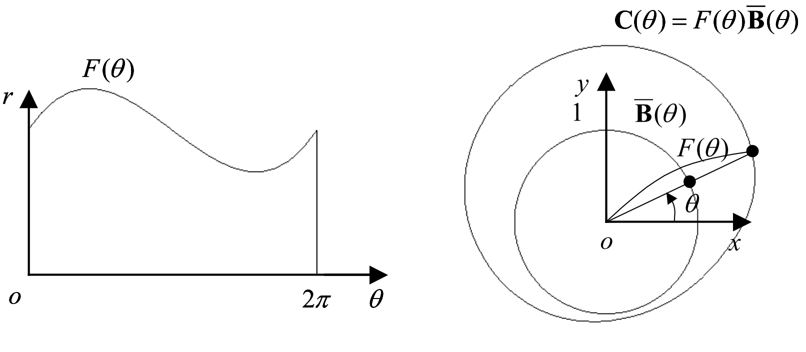
The NURBS-curve fitting method, which becomes the core part of the NURBS-surface fitting method of Chapter 3, is based on *polar-coordinate transformation* of 2D points representing a *circle-like curve* (*i.e.*, *rotational-freeform curve*).

2.1 Basic Idea

Let $\overline{\mathbf{B}}(\theta) = (\cos\theta, \sin\theta)$ denote a unit circle centered at the origin of a *xy*-coordinate system, then a *rotational-freeform curve* $\mathbf{C}(\theta)$ on the *xy*-plane may be expressed as the product of $\overline{\mathbf{B}}(\theta)$ and a *single-valued function* $F(\theta)$ as follows:

$$\mathbf{C}(\theta) = (x(\theta), y(\theta)) = F(\theta)\overline{\mathbf{B}}(\theta) = (F(\theta)\cos\theta, F(\theta)\sin\theta) \qquad (2.1)$$

where the unit circle $\overline{\mathbf{B}}(\theta)$ is called a (trigonometric) *base circle*. A construction for the above relationship is illustrated in Figure 2.1.



$F(\theta)$
r
o
2π
θ
$\mathbf{C}(\theta) = F(\theta)\overline{\mathbf{B}}(\theta)$
y
1
$\overline{\mathbf{B}}(\theta)$
$F(\theta)$
θ
o
x

(a) (b)

Figure 2.1. (a) Single-valued function in polar-coordinate system and (b) its corresponding 2D parametric curve in *xy*-coordinate system

A sequence of 2D points $\{\mathbf{\Psi}_k\} = \{(x_k, y_k)\}$ representing a rotational-freeform curve may be fitted to the curve $\mathbf{C}(\theta)$ in Equation (2.1) as follows:

1) $\{\mathbf{\Psi}_k\}$ is converted into a polar-coordinate point-sequence $\{\mathbf{\Theta}_k\} = \{(r_k, \theta_k)\}$

2) $\{\mathbf{\Theta}_k\}$ is fitted to a single-valued B-spline function $r = F(\theta)$

3) $F(\theta)$ is converted to a 2D curve $\mathbf{C}(\theta)$ using Equation (2.1).

By replacing the *trigonometric base-circle* $\overline{\mathbf{B}}(\theta)$ in Equation (2.1) with a *NURBS base-circle* $\mathbf{B}(u)$, we can obtain a NURBS curve $\mathbf{C}(u)$. However, for this replacement to work properly, we need a *parameter-synchronization* between θ and u. Because the NURBS base-circle $\mathbf{B}(u)$ may consist of four *rational-quadratic Bezier curves* (**RQB-curves**) [Till1983][Pieg1995] and it does not have a uniform flow rate [Faro1991] unlike the trigonometric base-circle $\overline{\mathbf{B}}(\theta)$. Now, this study presents a NURBS-curve fitting procedure using polar-coordinate transform.

2.2 Polar NURBS-curve Fitting Procedure

The overall procedure for fitting a 2D NURBS-curve from a sequence of 2D points representing a rotational-freeform curve, consists of five steps: 1) polar-coordinate transform, 2) base-circle construction, 3) curve-parameter synchronization, 4) single-valued B-spline curve fitting, and 5) NURBS-curve construction.

Step 1. Polar-coordinate Transform

A sequence of points $\{\mathbf{\Psi}_k\} = \{(x_k, y_k)\}$ in 2D Cartesian coordinates are converted into a *polar-coordinate point-sequence* $\{\mathbf{\Theta}_k\} = \{(r_k, \theta_k)\}$ as follows:

$$r_k = \sqrt{x_k^2 + y_k^2}$$
$$\theta_k = \begin{cases} \tan^{-1}(y_k/x_k) & \text{if } x_k > 0 \text{ and } y_k \geq 0 \\ \pi + \tan^{-1}(y_k/x_k) & \text{if } x_k < 0 \\ 2\pi + \tan^{-1}(y_k/x_k) & \text{if } x_k > 0 \text{ and } y_k < 0 \end{cases} \qquad (2.2)$$

Step 2. Base-circle Construction

A *base circle* $\mathbf{B}(u)$ having an *arc angle* of $\theta_e - \theta_s$ is easily obtained by using the *square-based method* [Till1983][Pieg1995], where a full circle having the *arc angle* of 2π is defined by four *rational-quadratic Bezier curves* (**RQB-curves**).

$$\mathbf{B}(u) = \frac{\sum_{i=0}^{n_B} N_{i,2}(u)\, w_i \mathbf{b}_i}{\sum_{i=0}^{n_B} N_{i,2}(u)\, w_i} \; ; \; U_B = \{\underbrace{u_s, \cdots, u_s}_{2+1}, u_{2+1}, \ldots, u_{n_B}, \underbrace{u_e, \cdots, u_e}_{2+1}\} \qquad (2.3)$$

where $\{N_{i,2}(u)\}$ are B-spline basis functions of degree 2; $\{w_i\}$ and $\{\mathbf{b}_i\}$ are the weights and control points of $\mathbf{B}(u)$, respectively.

Shown in Figure 2.2 is a sequence of 2D Cartesian-coordinate points $\{\mathbf{\Psi}_k\}$ inside an *angle range* of $[\theta_s, \theta_e]$, where the base circle $\mathbf{B}(u)$ consists of three RQB-curves since the arc angle is less than 3π/2 (but larger than π). Observe that the control polygon of the base circle $\mathbf{B}(u)$ is also shown in the figure.

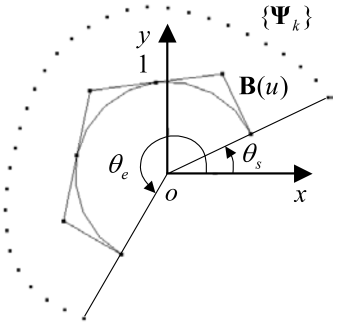


Figure 2.2. Base-circle construction for 2D data points

Step 3. Curve-parameter Synchronization

The *synchronized-parameter value* u_k of the base circle $\mathbf{B}(u)$ for a *polar-coordinate point* $\mathbf{\Theta}_k = (r_k, \theta_k)$ corresponds to the intersection point between $\mathbf{B}(u)$ and the line joining the origin and the *2D Cartesian-coordinate point* $\mathbf{\Psi}_k = (x_k, y_k)$ depicted in Figure 2.3.

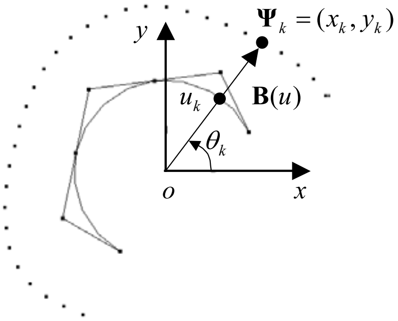


Figure 2.3. Geometric construction for parameter synchronization (*i.e.*, finding u_k)

To find u_k at the intersection-point, $\mathbf{B}(u)$ is first decomposed into individual RQB-curves $\{\mathbf{B}^j(\eta)\}$ for $u \in [u_s^j, u_e^j]$ corresponding to the angle range $[\theta_s^j, \theta_e^j]$ as follows:

$$\mathbf{B}^j(\eta) = \frac{\sum_{i=0}^{2} B_{i,2}(\eta)\, w_i^j \mathbf{b}_i^j}{\sum_{i=0}^{2} B_{i,2}(\eta)\, w_i^j}\, ; w_0^j = 1\, , \;\; w_1^j = \cos\{(\theta_e^j - \theta_s^j)/2\}\, , \;\; w_2^j = 1 \qquad (2.4)$$

where $\{B_{i,2}(\eta)\}$ are Bernstein polynomials of degree 2; $\{w_i^j\}$ and $\{\mathbf{b}_i^j\}$ are the weights and control points of each RQB-curve $\mathbf{B}^j(\eta)$, respectively; a *local parameter* η for each RQB-curve $\mathbf{B}^j(\eta)$ is defined as

$$\eta = (u - u_s^j)/(u_e^j - u_s^j)\, ; \;\; u \in [u_s^j, u_e^j]\, . \qquad (2.5)$$

Then, the local-parameter value η_k for a polar-coordinate point $\mathbf{\Theta}_k$ within the angle range $[\theta_s^j, \theta_e^j]$ may be obtained from the following relation:

$$(\cos\theta_k, \sin\theta_k) = \mathbf{B}^j(\eta) \qquad (2.6)$$

A solution to Equation (2.6) is given by [Blan1996]

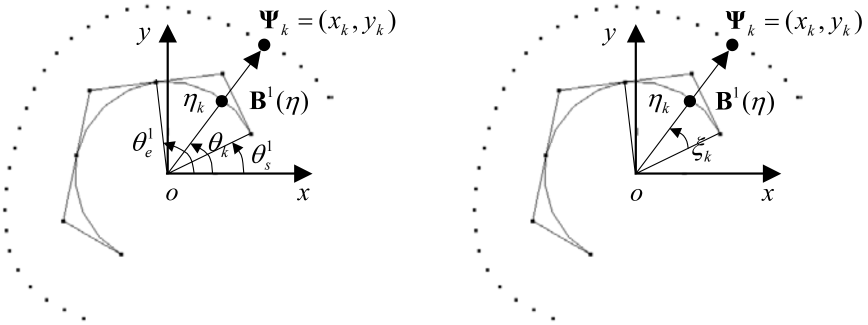
$$\eta_k = \frac{\tan\frac{\xi_k}{2}}{\sqrt{1-w^2} + (1-w)\tan\frac{\xi_k}{2}}\, ; \;\; \xi_k = \theta_k - \theta_s^j \qquad (2.7)$$

where w is the *weight* of the middle control point of the RQB-curve $\mathbf{B}^j(\eta)$. Finally, the synchronized-parameter value u_k for $\mathbf{\Theta}_k$ belonging to $\mathbf{B}^j(\eta)$ is obtained from the following equation:

$$u_k = (u_e^j - u_s^j)\eta_k + u_s^j \qquad (2.8)$$

In practice, an RQB-curve is defined over a unit interval so that $u_e^j = j$, $u_s^j = j-1$, in which case Equation (2.8) reduces to $u_k = \eta_k + (j-1)$.

Shown in Figure 2.4a is a construction for finding η_k corresponding to the intersection point between $\mathbf{B}^1(\eta)$ and the line joining the coordinate origin and $\mathbf{\Psi}_k$. The relation between η_k and ξ_k is depicted in Figure 2.4b.



(a) (b)

Figure 2.4. Geometric construction for local-parameter synchronization (*i.e.*, finding η_k)

Step 4. Single-valued B-spline Curve Fitting

A single-valued B-spline curve $F(u)$ of degree *p* (usually $p = 3$) is fitted to the *parameter-synchronized points* $\{\hat{\mathbf{\Theta}}_k\} = \{(r_k, u_k)\}$ by using standard fitting methods [deBo1978][Fari1988] [Choi1991][Pieg1995][Hosc1993].

$$F(u) = \sum_{i=0}^{n_F} N_{i,p}(u)\, f_i\ ;\quad U_F = \{\underbrace{u_s,\cdots,u_s}_{p+1}, u_{p+1},\ldots,u_{n_F}, \underbrace{u_e,\cdots,u_e}_{p+1}\} \qquad (2.9)$$

Step 5. NURBS-curve Construction

At the final step, a (parametric) NURBS curve is constructed from the single-valued B-spline curve $F(u)$ by applying a *symbolic-product operation* [Pieg1997] with the NURBS base-circle $\mathbf{B}(u)$. Namely, from $\mathbf{C}(u) = F(u)\mathbf{B}(u)$ we have

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n_C} N_{i,p+2}(u) w_i \mathbf{c}_i}{\sum_{i=0}^{n_C} N_{i,p+2}(u) w_i}; \quad U_C = \{\underbrace{u_s,\cdots,u_s}_{p+2+1}, u_{p+2+1},\ldots,u_{n_C}, \underbrace{u_e,\cdots,u_e}_{p+2+1}\} \tag{2.10}$$

where $\{N_{i,p+2}(u)\}$ are B-spline basis functions of degree $p+2$; $\{w_i\}$ are weights, and $\{\mathbf{c}_i\}$ are control points of the NURBS curve $\mathbf{C}(u)$, respectively. For the details, refer to the Appendix A. The above NURBS-curve construction concept is illustrated in Figure 2.5, where the degree of $F(u)$ is 3 and that of the resulting NURBS-curve is 5.

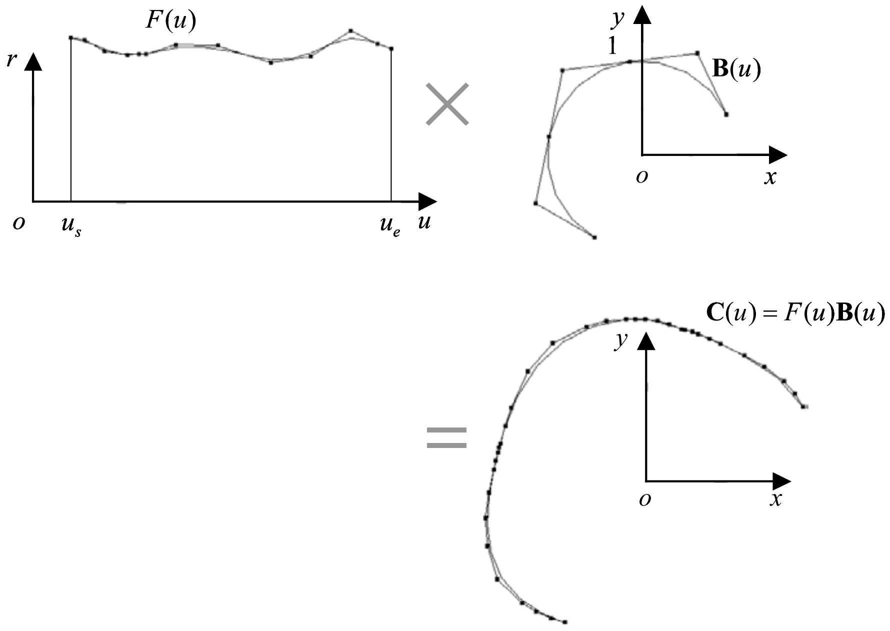


Figure 2.5. NURBS-curve construction via symbolic-product operation

If the 2D Cartesian-coordinate point-sequence $\{\mathbf{\Psi}_k\}$ represents a true circle of radius R, the single-valued B-spline curve will become a constant line, namely, $F(u) = R$. In this case, the degree of $F(u)$ is zero (*i.e.*, $p = 0$) and that of the resulting NURBS-curve becomes 2. That is, $\mathbf{C}(u) = R\,\mathbf{B}(u)$ becomes a true circle of radius *R*.

3. NURBS-surface Fitting

The procedure for fitting a NURBS surface from cloud-of-points (**COP**) data, is similar to the curve-fitting procedure given in the previous chapter, and consists of five steps: 1) orthogonal-coordinate transform, 2) base-geometry construction, 3) surface-parameter synchronization, 4) single-valued B-spline surface fitting, and 5) NURBS-surface construction. A *rotational-freeform shape* (*i.e.*, *sphere-like* or *cylinder-like shape*) may be one-to-one mapped to a sphere or a cylinder: When the shape is more naturally mapped to a sphere, a *spherical NURBS-surface* is fitted from the COP-data.

3.1 Spherical NURBS-surface Fitting Procedure

Step 1. Spherical-coordinate Transform

The COP-data $\{\mathbf{\Psi}_k\} = \{(x_k, y_k, z_k)\}$ representing a sphere-like shape are converted into *spherical-coordinate points* $\{\mathbf{\Phi}_k\} = \{(\rho_k, \phi_k, \theta_k,)\}$ as follows:

$$\begin{aligned} \rho_k &= \sqrt{x_k^2 + y_k^2 + z_k^2} \\ \phi_k &= \cos^{-1}\left(z_k \Big/ \sqrt{x_k^2 + y_k^2 + z_k^2} \right) \\ \theta_k &= \begin{cases} \tan^{-1}(y_k/x_k) & \textit{if } x_k > 0 \textit{ and } y_k \geq 0 \\ \pi + \tan^{-1}(y_k/x_k) & \textit{if } x_k < 0 \\ 2\pi + \tan^{-1}(y_k/x_k) & \textit{if } x_k > 0 \textit{ and } y_k < 0 \end{cases} \end{aligned} \tag{3.1}$$

And, the *angle ranges* are computed as

$$\phi_s = \min\{\phi_k\};\ \ \phi_e = \max\{\phi_k\};\ \ \theta_s = \min\{\theta_k\};\ \ \theta_e = \max\{\theta_k\}. \tag{3.2}$$

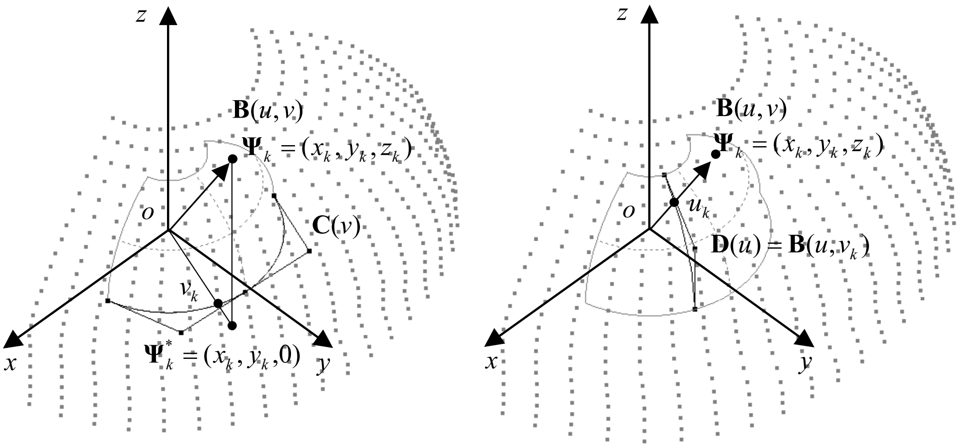
Step 2. Base-sphere Construction

A unit sphere defined over the angle ranges $[\phi_s, \phi_e]$ and $[\theta_s, \theta_e]$ is chosen as the *base sphere* $\mathbf{B}(u, v)$, which is a NURBS-surface consisting of up to (2×4) *rational-quadratic Bezier patches* (**RQB-patches**) depending on *arc angles* $\phi_e - \phi_s$ and $\theta_e - \theta_s$ [Pieg1987].

$$\mathbf{B}(u,v)=\frac{\sum_{i=0}^{n_B}\sum_{j=0}^{m_B}N_{i,2}(u)N_{j,2}(v)\,w_{i,j}\mathbf{b}_{i,j}}{\sum_{i=0}^{n_B}\sum_{j=0}^{m_B}N_{i,2}(u)N_{j,2}(v)\,w_{i,j}};$$
$$U_B=\{\underbrace{u_s,\cdots,u_s}_{2+1},u_{2+1},\ldots,u_{n_B},\underbrace{u_e,\cdots,u_e}_{2+1}\},\qquad (3.3)$$
$$V_B=\{\underbrace{v_s,\cdots,v_s}_{2+1},v_{2+1},\ldots,v_{m_B},\underbrace{v_e,\cdots,v_e}_{2+1}\}$$

Step 3. Surface-parameter Synchronization

Similar to the curve-fitting case, finding the *synchronized-parameter values* of the base sphere $\mathbf{B}(u,v)$ for a spherical-coordinate point $\mathbf{\Phi}_k=(\rho_k,\phi_k,\theta_k)$ is equivalent to finding an intersection point between $\mathbf{B}(u,v)$ and the line joining the origin and the *3D Cartesian-coordinate point* $\mathbf{\Psi}_k=(x_k,y_k,z_k)$. Since the base sphere $\mathbf{B}(u,v)$ is constructed by a surface-of-revolution method [Pieg1987], synchronized-parameter values u_k and v_k can be calculated separately by using the curve-parameter synchronization method (Step 3) described in Section 2.2. As depicted in Figure 3.1a, v_k is obtained from $\mathbf{C}(v)$ and $\mathbf{\Psi}_k^*=(x_k,y_k,0)$, where $\mathbf{C}(v)$ is a NURBS base-circle on the *xy*-plane with an arc angle $\theta_e-\theta_s$, and $\mathbf{\Psi}_k^*$ is the projection point of $\mathbf{\Psi}_k(x_k,y_k,z_k)$ on the *xy*-plane. Once v_k is determined, another base circle $\mathbf{D}(u)=\mathbf{B}(u,v_k)$ is defined as depicted in Figure 3.1b, and then, u_k is calculated from $\mathbf{D}(u)$ and $\mathbf{\Psi}_k$.



(a) (b)

Figure 3.1. Geometric construction for parameter synchronization for spherical NURBS-surface fitting (*i.e.*, finding v_k and u_k respectively)

Step 4. Single-valued B-spline Surface Fitting

A single-valued B-spline surface $F(u,v)$ of degree (p, q) is now fitted to the *parameter-synchronized points* $\{\hat{\mathbf{\Phi}}_k\} = \{(\rho_k, u_k, v_k)\}$ by using standard fitting methods [deBo1978][Fari1988][Choi1991][Pieg1995][Hosc1993].

$$\begin{aligned}
&F(u,v) = \sum_{i=0}^{n_F}\sum_{j=0}^{m_F} N_{i,p}(u)N_{i,q}(v)f_{i,j};\\
&U_F = \{\underbrace{u_s,\cdots,u_s}_{p+1}, u_{p+1},\ldots,u_{n_F},\underbrace{u_e,\cdots,u_e}_{p+1}\},\\
&V_F = \{\underbrace{v_s,\cdots,v_s}_{q+1}, v_{q+1},\ldots,v_{m_F},\underbrace{v_e,\cdots,v_e}_{q+1}\}
\end{aligned} \tag{3.4}$$

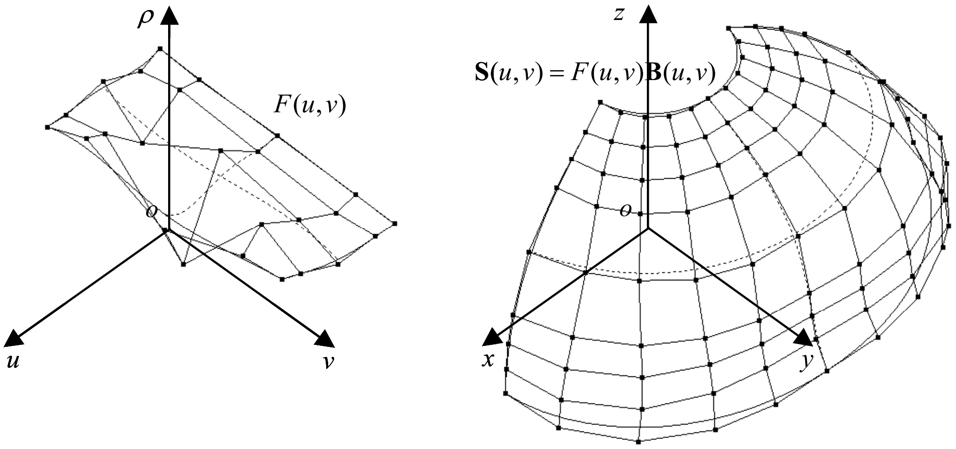
Step 5. NURBS-surface Construction

Finally, a (parametric) NURBS surface $\mathbf{S}(u,v)$ is constructed from the single-valued B-spline surface $F(u,v)$ by applying a symbolic-product operation with the NURBS base-sphere $\mathbf{B}(u,v)$. From $\mathbf{S}(u,v) = F(u,v)\mathbf{B}(u,v)$, we can obtain the resulting NURBS surface

$$\begin{aligned}
&\mathbf{S}(u,v) = \frac{\sum_{i=0}^{n_S}\sum_{j=0}^{m_S} N_{i,p+2}(u)N_{j,p+2}(v)\,w_{i,j}\mathbf{s}_{i,j}}{\sum_{i=0}^{n_S}\sum_{j=0}^{m_S} N_{i,p+2}(u)N_{j,p+2}(v)\,w_{i,j}};\\
&U_S = \{\underbrace{u_s,\cdots,u_s}_{p+2+1}, u_{p+2+1},\ldots,u_{n_S},\underbrace{u_e,\cdots,u_e}_{p+2+1}\},\\
&V_S = \{\underbrace{v_s,\cdots,v_s}_{q+2+1}, v_{q+2+1},\ldots,v_{m_S},\underbrace{v_e,\cdots,v_e}_{q+2+1}\}
\end{aligned} \tag{3.5}$$

of degree ($p+2$, $q+2$). For more details of above surface-case symbolic-production, see Appendix B.

Shown in Figure 3.2a is a *bi-cubic B-spline surface* (*i.e.*, $p=q=3$) for the parameter-synchronized points $\{\hat{\mathbf{\Phi}}_k\} = \{(\rho_k, u_k, v_k)\}$, and shown in Figure 3.2b is the resulting NURBS-surface of degree (5, 5) obtained from NURBS-surface conversion.



(a) (b)

Figure 3.2. NURBS surface obtained by spherical NURBS-surface fitting

3.2 Cylindrical NURBS-surface Fitting Procedure

When the rotational-freeform shape is more naturally mapped to a cylinder, a *cylindrical NURBS-surface* is fitted to the COP-data. The procedure is basically the same as that of the spherical case described in Section 3.1.

First, *3D Cartesian-coordinate points* $\{\mathbf{\Psi}_k\} = \{(x_k, y_k, z_k)\}$ are transformed into *cylindrical-coordinate points* $\{\mathbf{X}_k\} = \{(r_k, \theta_k, z_k)\}$ as

$$\begin{aligned} r_k &= \sqrt{x_k^2 + y_k^2} \\ \theta_k &= \begin{cases} \tan^{-1}(y_k/x_k) & \text{if } x_k > 0 \text{ and } y_k \geq 0 \\ \pi + \tan^{-1}(y_k/x_k) & \text{if } x_k < 0 \\ 2\pi + \tan^{-1}(y_k/x_k) & \text{if } x_k > 0 \text{ and } y_k < 0 \end{cases} \\ z_k &= z_k \end{aligned} \qquad (3.6)$$

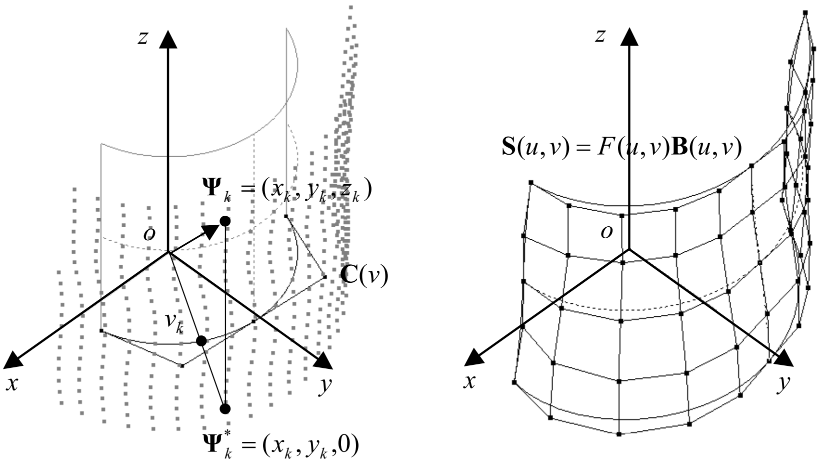
and a cylinder with unit radius is used as a *base cylinder* $\mathbf{B}(u,v)$ which is a NURBS surface of degree (1, 2) over the angle range $[\theta_s, \theta_e]$ and *height range* $[z_s, z_e]$.

$$\mathbf{B}(u,v)=\frac{\sum_{i=0}^{1}\sum_{j=0}^{m_B}N_{i,1}(u)N_{j,2}(v)\,w_{i,j}\mathbf{b}_{i,j}}{\sum_{i=0}^{1}\sum_{j=0}^{m_B}N_{i,1}(u)N_{j,2}(v)\,w_{i,j}};$$
$$U_B=\{\underbrace{u_s,u_s}_{1+1},\underbrace{u_e,u_e}_{1+1}\},\ V_B=\{\underbrace{v_s,\cdots,v_s}_{2+1},v_{2+1},\dots,v_{m_B},\underbrace{v_e,\cdots,v_e}_{2+1}\} \tag{3.7}$$

Surface-parameter synchronization is carried out as follows: As with the spherical case, v_k is obtained by using the curve-parameter synchronization method of Section 2.2 as depicted in Figure 3.3a, and u_k is given by $u_k=(z_k-z_s)/(z_e-z_s)$. And then, a single-valued B-spline surface $F(u,v)$ is fitted to parameter-synchronized points $\{\hat{\mathbf{X}}_k\}=\{(r_k,u_k,v_k)\}$ as with the spherical case. Finally, a 3D cylindrical NURBS-surface $\mathbf{S}(u,v)$ is given by:

$$\mathbf{S}(u,v)=\frac{\sum_{i=0}^{n_S}\sum_{j=0}^{m_S}N_{i,p+1}(u)N_{j,q+2}(v)\,w_{i,j}\mathbf{s}_{i,j}}{\sum_{i=0}^{n_S}\sum_{j=0}^{m_S}N_{i,p+1}(u)N_{j,q+2}(v)\,w_{i,j}};$$
$$U_S=\{\underbrace{u_s,\cdots,u_s}_{p+1+1},u_{p+1+1},\dots,u_{n_S},\underbrace{u_e,\cdots,u_e}_{p+1+1}\}, \tag{3.8}$$
$$V_S=\{\underbrace{v_s,\cdots,v_s}_{q+2+1},v_{q+2+1},\dots,v_{m_S},\underbrace{v_e,\cdots,v_e}_{q+2+1}\}$$

The resulting NURBS surface is constructed by using a "modified" symbolic-product operation, which is also derived in Appendix B. Shown in Figure 3.3b is the resulting NURBS-surface from the COP-data of Figure 3.3a.


z
$\boldsymbol{\Psi}_k = (x_k, y_k, z_k)$
o
$\mathbf{C}(v)$
v_k
x
y
$\boldsymbol{\Psi}_k^* = (x_k, y_k, 0)$
$\mathbf{S}(u,v) = F(u,v)\mathbf{B}(u,v)$
z
o
x
y

(a) (b)

Figure 3.3. NURBS surface obtained by cylindrical NURBS-surface fitting

When the degrees of $F(u,v)$ are (p, q), it can be shown that the degrees of the resulting NURBS-surface $\mathbf{S}(u,v)$ are ($\max\{p,1\}$, $q+2$) (See Appendix B.2). For COP-data representing a true *surface-of-revolution*, the degrees of $F(u,v)$ would be (3, 0). Thus, the degrees of $\mathbf{S}(u,v)$ are (3, 2). For a true cylinder shape, the degrees of $\mathbf{S}(u,v)$ become (1, 2). Shown in Figure 3.4a are COP-data for a traditional Korean ceramic. The fitted NURBS-surface together with its control points for the COP-data of Figure 3.4a is shown in Figure 3.4c. Observe from the control points that the degrees of the resulting NURBS surface are (3,2). This is expected since the ceramic is a “true” surface-of-revolution surface. Now, the COP-data of Figure 3.4a are scaled down (0.75) along the y-direction to obtain a rotational-freeform shape as shown in Figure 3.4b. The fitted NURBS surface of degrees (3,5) and its control points are shown in Figure 3.4d.

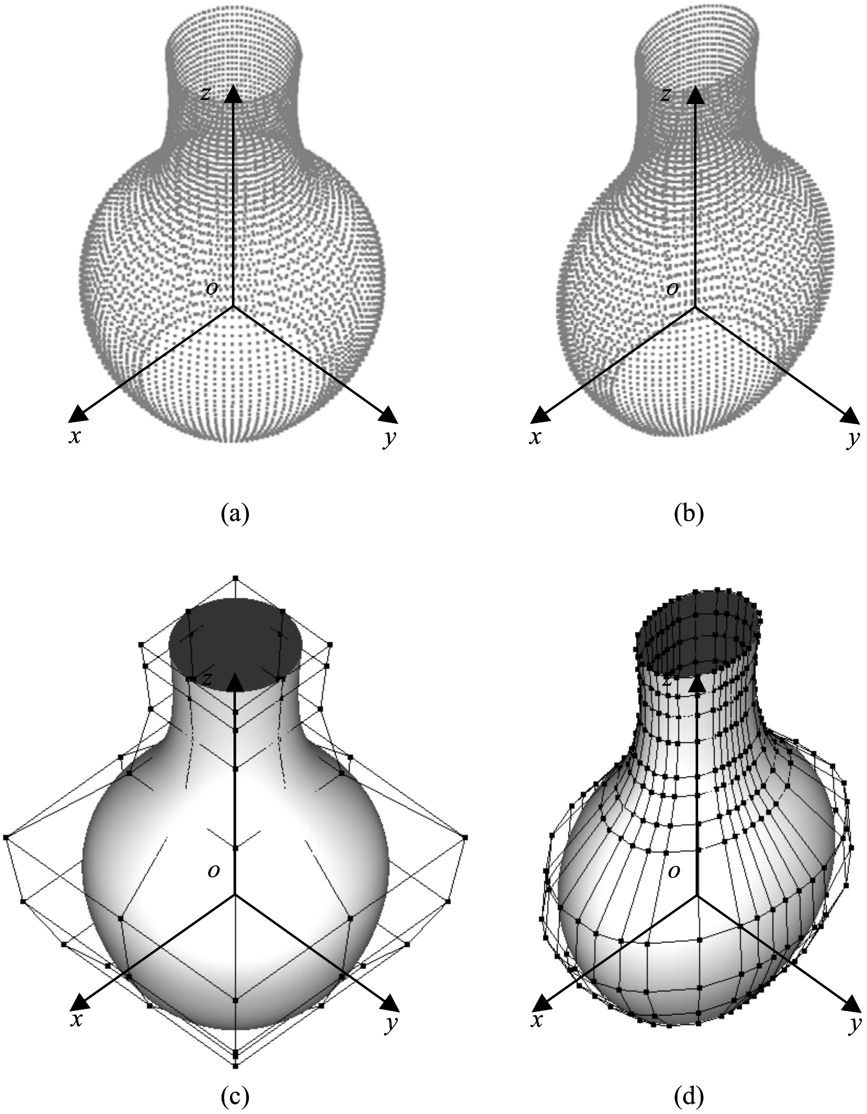


Figure 3.4. NURBS surface fitting for traditional Korean ceramic

3.3 Special Treatments

3.3.1 Closed Surfaces

When a "closed" surface is to be constructed, *periodic B-spline basis functions* have to be employed in the *v*-direction (corresponding to θ) during the single-valued B-spline surface fitting stage. A "closed" single-valued B-spline surface $\widetilde{F}(u,v)$ in *v*-direction is given by [Park1996]

$$\begin{aligned}
&\widetilde{F}(u,v)=\sum_{i=0}^{n_{\widetilde{F}}}\sum_{j=0}^{m_{\widetilde{F}}}N_{i,p}(u)N_{j,q}(v)\,\widetilde{f}_{i,j\,\mathrm{mod}\,q};\\
&U_{\widetilde{F}}=\{\underbrace{u_s,\cdots,u_s}_{p+1},u_{p+1},\ldots,u_{n_{\widetilde{F}}},\underbrace{u_e,\cdots,u_e}_{p+1}\},\\
&V_{\widetilde{F}}=\{v_0,\ldots,v_q,v_{q+1},\ldots,v_{m_{\widetilde{F}}},v_{m_{\widetilde{F}}+1},\ldots,v_{m_{\widetilde{F}}+q+1}\}
\end{aligned}\qquad(3.9)$$

where $v_{q-j+1}-v_{q-j}=v_{m_{\widetilde{F}}-j+2}-v_{m_{\widetilde{F}}-j+1}$, $v_{m_{\widetilde{F}}+j+1}-v_{m_{\widetilde{F}}+j}=v_{q+j}-v_{q+j-1}$; $j=1,\ldots,q$ and $v_q=v_s$, $v_{m_{\widetilde{F}}+1}=v_e$ (v_s and v_e are the *v*-directional start and end parameters of the base geometry, which corresponds to angle range $[0,2\pi]$). Because of the *v*-directional periodicity of $\widetilde{F}(u,v)$, finding control points (or ordinates) $\{\widetilde{f}_{i,j}\}$ is a kind of constrained fitting (See Appendix C) under following conditions:

$$\widetilde{f}_{i,j}=\widetilde{f}_{i,m_{\widetilde{F}}-q+j-1};\quad j=0,\ldots,q-1\,.\qquad(3.10)$$

Once $\widetilde{F}(u,v)$ is obtained, an additional step so called "clipping" is required before the NURBS-surface construction (Step 5) described in Section 3.1. A clipping step is for the conversion of a *v*-directionally "closed" single-valued B-splind surface $\widetilde{F}(u,v)$ defined over the *v*-directional "periodic" knot vector in Equation (3.9), into an "open" single-valued B-spline surface $F(u,v)$ defined over both "non-periodic" or "clamped" knot vectors as given in Equation (3.4). A clipping operation uses multiple-knot insertion with multiplicity q at the start and end parameters of base geometry so that the clipped $F(u,v)$ and base geometry $\mathbf{B}(u,v)$ have the same *v*-directional parameter span $[v_s,v_e]$ corresponding to angle range $[\theta_s,\theta_e]=[0,2\pi]$.

Shown in Figure 3.5 is a closed NURBS-surface fitting example: (a) COP-data of a cylinder-like part of a femur, $\{\mathbf{\Psi}_k\}$, (b) the corresponding base cylinder $\mathbf{B}(u,v)$, (c) parameter-synchronized cylindrical-coordinate points $\{\hat{\mathbf{X}}_k\}$, (d) a *v*-directionally closed single-valued B-spline surface $\widetilde{F}(u,v)$ fitted to $\{\hat{\mathbf{X}}_k\}$, (e) a open single-valued B-spline surface $F(u,v)$ obtained by clipping $\widetilde{F}(u,v)$, and (f) the resulting NURBS-surface constructed by applying a symbolic-product operation of $F(u,v)$ with $\mathbf{B}(u,v)$.

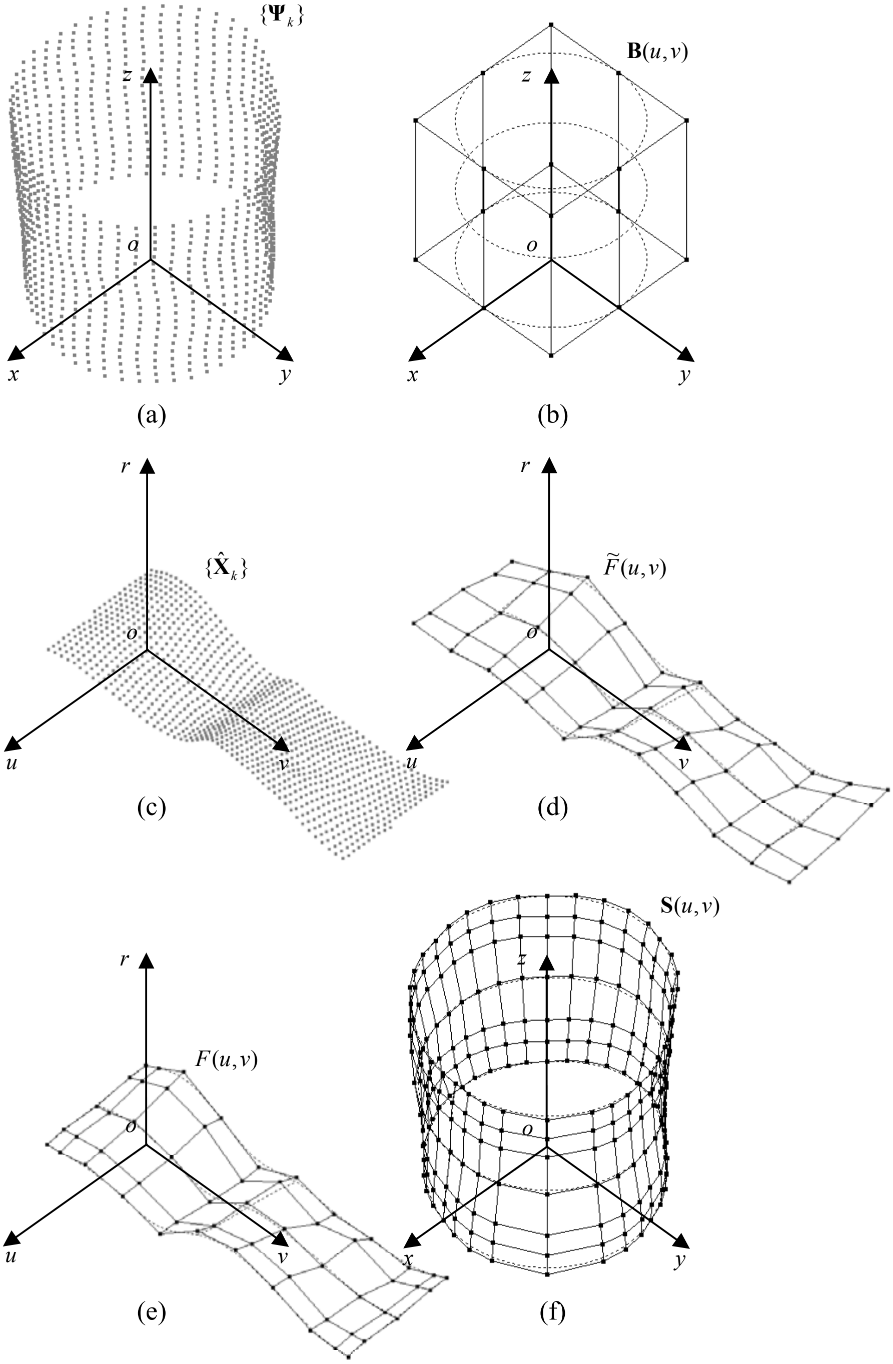


Figure 3.5. Closed NURBS-surface fitting

3.3.2 Degeneracy at Pole

When fitting a closed spherical-NURBS surface, an additional treatment is required to solve the "degeneracy problem", that is, to smoothly fill the "hole" near the pole (See Figure 3.6a).

The conditions for a level of geometric continuities at the degenerate point $\mathbf{S}(u_s, v)$ corresponding to the pole of the base sphere $\mathbf{B}(u_s, v)$ are:

1) for G^0 continuity: the collapse of the boundary of the surface to a single point, that is,

$$\mathbf{S}(u_s, v) = \mathbf{p}^* ; \tag{3.11}$$

2) for G^1 continuity: the existence of a tangent plane, that is,

$$\mathbf{n}(u_s, v) = \frac{\mathbf{S}_u(u_s, v) \times \mathbf{S}_{uv}(u_s, v)}{\left\| \mathbf{S}_u(u_s, v) \times \mathbf{S}_{uv}(u_s, v) \right\|} = \mathbf{n}^* \quad \text{[Faux1981];} \tag{3.12}$$

3) for G^2 continuity: for the normal curvature to follow Euler formula [Wolt1992], that is,

$$\kappa(u_s, v : \psi) = \kappa_M \cos^2 \psi + \kappa_m \sin^2 \psi \quad \text{[doCa1976]} \tag{3.13}$$

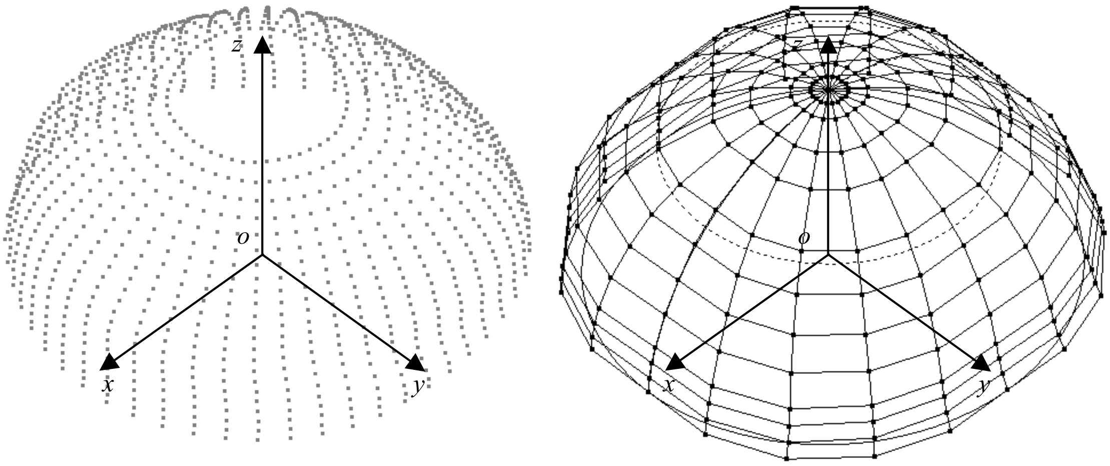
where κ_M, κ_m are the maximum- and minimum-principal curvatures, respectively, and ψ is the angle from maximum-principal direction on the tangent plane. However, Equation (3.13) is only met if the surface is locally a surface-of-revolution at the degenerate point [Sanc1994] – it is too restrictive to fit a NURBS surface to COP-data representing a general rotational-freeform shape.

One simple solution is to impose following control point (or ordinate) conditions when fitting a single-valued B-spline surface:

$$f_{0,j} = f_{0,0} ; \quad j = 1, \ldots, m_F \tag{3.14a}$$

$$f_{1,j} = f_{0,j} ; \quad j = 0, \ldots, m_F \tag{3.14b}$$

With above conditions, the resulting NURBS-surface has G^1 continuity at the degenerate point while the normal vector at the point is $\mathbf{k} = (0,0,1)$ (See Appendix D.1). Namely, the tangent plane at the degenerate point is horizontal. Shown in Figure 3.6b is the NURBS surface satisfying the G^1 continuity condition, which is fitted to COP-data of Figure 3.6a.



(a) (b)

Figure 3.6. COP-data of sphere-like part with degeneracy problem and resulting NURBS-surface with simple G^1 continuity at degenerate point

Above simple solution with a horizontal tangent-plane condition at the degenerate point works well in almost cases. However, sometimes, it may generate a NURBS-surface having a “wiggle” around the degenerate point as shown in Figure 3.7.

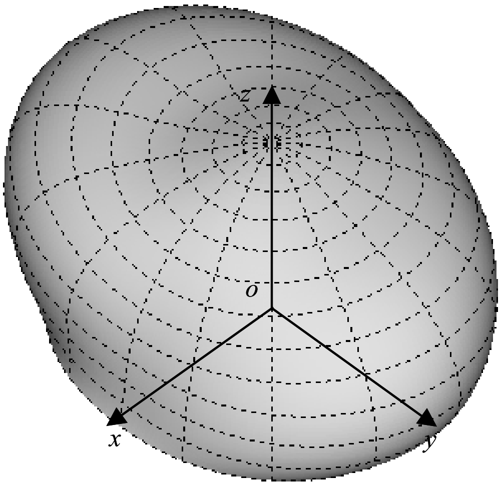


Figure 3.7. Spherical NURBS-surface with wiggle around degenerate point

Therefore, another solution is given here – to generate a "cap" surface smoothly being connected to an existing NURBS-surface $\mathbf{S}(u,v)$ with a hole, while closing up the hole with a non-horizontal tangent plane at the degenerate point.

To make a cap surface, first of all, it is required for the single-valued B-spline surface $F(u,v)$ for $\mathbf{S}(u,v)$ to be fitted over knot vectors

$$\begin{aligned} U_F &= \{\underbrace{u_s,\ldots,u_s}_{p+1},u_{p+1},\ldots,u_{n_F},\underbrace{u_e,\ldots,u_e}_{p+1}\}, \\ V_F &= \{\underbrace{v_s,\ldots,v_s}_{q+1},v_{p+1},\ldots,v_{q+H},\ldots,v_{m_F},\underbrace{v_e,\ldots,v_e}_{q+1}\} \end{aligned} \tag{3.15}$$

where $v_{q+H}=(v_s+v_e)/2$, $v_{j+H}=v_j+(v_e-v_s)/2$; $j=q+1,\ldots,q+H-1$, $H=K/2$, $K=m_F-q+1$ (K is assumed to be an even number). Whereas U_F is an ordinary clamped-knot vector, V_F is a special typed where the center knot v_{q+H} divides the parameter span in half, and the front part of knot spans are repeated periodically after v_{q+H}.

The procedure to construct a cap surface is composed of following four steps: 1) preparation of connecting curves, 2) determination of the degenerate point and its surface-normal vector, 3) connecting-curve splitting, and 4) construction of a cap surface.

Step 1. Preparation of Connecting Curves

Because of the periodicity of the v-directional knot spans (Refer to Equation (3.15)), it is possible to generate a plane curve ("*connecting curve*") connects one column of the control polygon of $\mathbf{S}(u,v)$ with its corresponding column on opposite side in the *meridian section* (or plane) depicted in Figure 3.8a. The connecting curves are Bezier curves of degree 5,

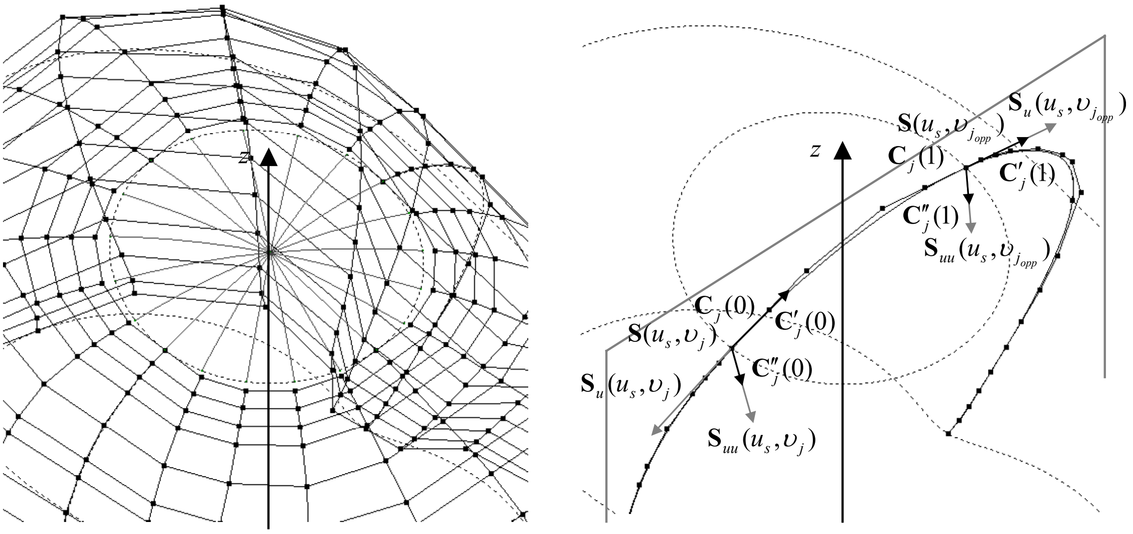
$$\mathbf{C}_j(t)=\sum_{i=0}^{5}B_{i,5}(t)\mathbf{c}_i^j\,;\quad j=0,\ldots,m_S/2-1. \tag{3.16}$$

For each connecting-curve to be smoothly connected to $\mathbf{S}(u,v)$, G^2 continuity condition is imposed as follows (See Figure 3.8b):

$$\mathbf{C}_j(0)=\mathbf{S}(u_s,\upsilon_j),\mathbf{C}'_j(0)=\alpha\,\mathbf{S}_u(u_s,\upsilon_j),\mathbf{C}''_j(0)=\beta\,\mathbf{S}_{uu}(u_s,\upsilon_j), \tag{3.17a}$$

$$\mathbf{C}_j(1)=\mathbf{S}(u_s,\upsilon_{j_{opp}}),\mathbf{C}'_j(1)=-\alpha\,\mathbf{S}_u(u_s,\upsilon_{j_{opp}}),\mathbf{C}''_j(1)=\beta\,\mathbf{S}_{uu}(u_s,\upsilon_{j_{opp}}) \tag{3.17b}$$

where $\upsilon_j = \sum_{k=1}^{q} v_{j+k} \big/ q$; $j = 0,\ldots,m_S/2-1$, $j_{opp} = j + m_S/2$ are the v-directional nodes [Ries1973] [Pieg1995] of $\mathbf{S}(u,v)$.



z
$\mathbf{S}_u(u_s,\upsilon_{j_{opp}})$
$\mathbf{S}(u_s,\upsilon_{j_{opp}})$
z
$\mathbf{C}_j(1)$
$\mathbf{C}'_j(1)$
$\mathbf{C}''_j(1)$
$\mathbf{S}_{uu}(u_s,\upsilon_{j_{opp}})$
$\mathbf{C}_j(0)$
$\mathbf{C}'_j(0)$
$\mathbf{S}(u_s,\upsilon_j)$
$\mathbf{C}''_j(0)$
$\mathbf{S}_u(u_s,\upsilon_j)$
$\mathbf{S}_{uu}(u_s,\upsilon_j)$

(a) (b)

Figure 3.8. Curves connect two columns of control polygon and G^2 continuity condition imposing for each connecting-curve

Step 2. Determination of Degenerate Point and Surface-normal Vector

First, let us find α^*, β^* minimize the sum of the squares of mean-deviations of connecting curves at $t = 0.5$,

$$E(\alpha,\beta) = \sum_{j=0}^{m_S/2-1} \{\mathbf{C}_j(\alpha,\beta:0.5) - \overline{\mathbf{C}}(\alpha,\beta:0.5)\}^2 \tag{3.18}$$

where

$$\mathbf{C}_j(\alpha,\beta:t) = \sum_{i=0}^{5} B_{i,5}(t)\mathbf{c}_i^j(\alpha,\beta), \tag{3.19a}$$

$$\overline{\mathbf{C}}(\alpha,\beta:t) = \frac{2}{m_S} \sum_{j=0}^{m_S/2-1} \mathbf{C}_j(\alpha,\beta:t) . \tag{3.19b}$$

Differentiating Equation (3.18) with respect to α, β and setting zero, that is, $\partial E(\alpha,\beta)/\partial\alpha = 0$, $\partial E(\alpha,\beta)/\partial\beta = 0$ yield optimal α^*, β^* (See Appendix D.2). Then,

$$\mathbf{p}^* = (p_x, p_y, p_z) = \overline{\mathbf{C}}(\alpha^*, \beta^* : 0.5)\,, \tag{3.20}$$

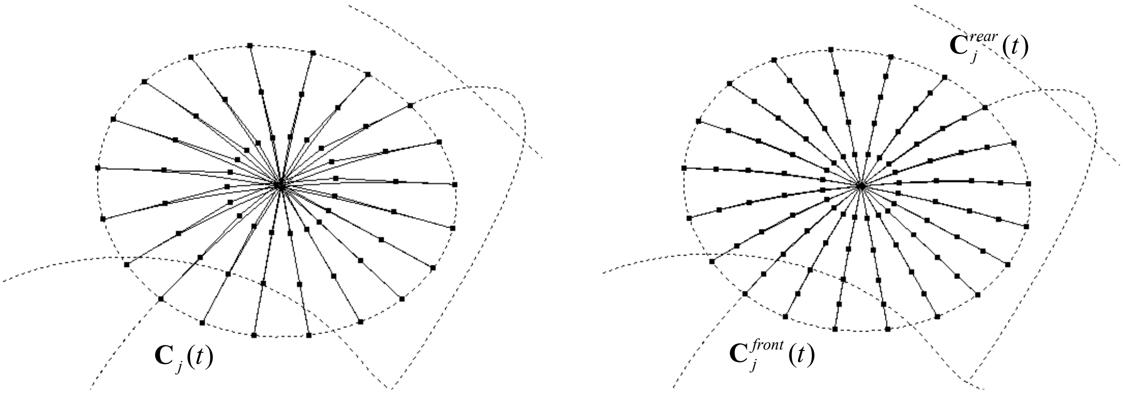
and

$$\mathbf{n}^* = (n_x, n_y, n_z) = \overline{\mathbf{n}} \big/ \left\|\overline{\mathbf{n}}\right\| \tag{3.21}$$

where $\overline{\mathbf{n}} = \left(\sum_{j=0}^{m_S/2-1} \varphi_j \overline{\mathbf{n}}_j\right) \Big/ \left(\sum_{j=0}^{m_S/2-1} \varphi_j\right)$; $\overline{\mathbf{n}}_j = \mathbf{n}_j \big/ \left\|\mathbf{n}_j\right\|$; $\mathbf{n}_j = \mathbf{C}'_j(0.5) \times \mathbf{C}'_{j+1}(0.5)$, $\varphi = \angle(\mathbf{C}'_j(0.5), \mathbf{C}'_{j+1}(0.5))$.

Step 3. Connecting-curve Splitting

Based on the deCasteljau algorithm [Boeh1984][deCa1986], each connecting-curve $\mathbf{C}_j(t)$ is split at $t = 0.5$ into two Bezier curves, $\mathbf{C}_j^{front}(t)$, $\mathbf{C}_j^{rear}(t)$. Next, the reversal of the parameter direction of $\mathbf{C}_j^{front}(t)$ is done to march the parameter direction with that of a cap surface to be constructed in the following Step 4. Then, the first control point of each split curve is moved to the $\mathbf{p}^*$ for the G^0 continuity, and the second point is also moved on the tangent plane, $n_x x + n_y y + n_z z = \mathbf{n}^* \bullet \mathbf{p}^*$ for the G^1 continuity. Finally, the third control point can be moved so that a fairness measure $\int \{\mathbf{C}''(t)\}^2 dt$ is minimized [Kjel1983][Meie1987] (See Figure 3.9).



(a) (b)

Figure 3.9. Connecting curves: (a) before splitting and (b) after splitting and control-point repositioning

Step 4. Construction of Cap Surface

Let us note the curves split and modified in the previous step using homogenous coordinates as

$$\begin{aligned} &\tilde{\mathbf{C}}_k^w(t) = \sum_{i=0}^{5} N_{i,5}(t)\,\tilde{\mathbf{c}}_i^{k,w};\ k = 0,\ldots,m_S; \\ &U_{\tilde{C}} = \{\underbrace{u_s,\ldots,u_s}_{5+1},\underbrace{u_e,\ldots,u_e}_{5+1}\} \end{aligned} \tag{3.22}$$

where $\tilde{\mathbf{c}}_i^{k,w} = (w_k\tilde{\mathbf{c}}_i^k, w_k)$ ($\{w_k\}$ are the weights of the control points of the boundary curve $\mathbf{S}(u_s,v)$). A cap surface can be written in the homogenous representation as

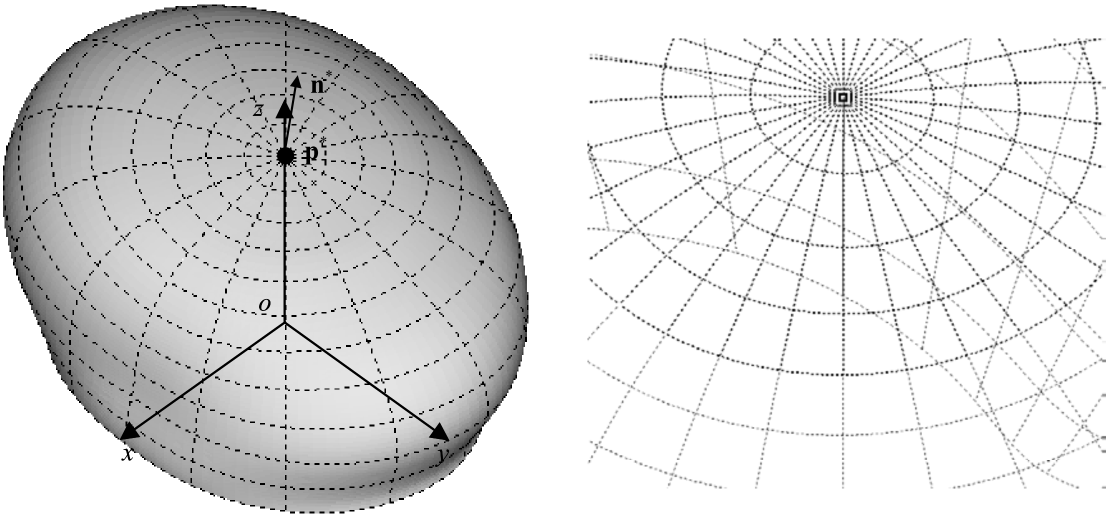
$$\begin{aligned} &\mathbf{S}_{cap}^w(u,v) = \sum_{i=0}^{5}\sum_{j=0}^{m_S} N_{i,5}(u)N_{j,q+2}(v)\,\mathbf{s}_{i,j}^{cap,w}; \\ &U_{S_{cap}} = U_{\tilde{C}} = \{\underbrace{u_s,\ldots,u_s}_{5+1},\underbrace{u_e,\ldots,u_e}_{5+1}\}, \\ &V_{S_{cap}} = V_S = \{\underbrace{v_s,\ldots,v_s}_{q+2+1},v_{q+2+1},\ldots,v_{m_S},\underbrace{v_e,\ldots,v_e}_{q+2+1}\} \end{aligned} \tag{3.23}$$

where V_S is the v-directional knot vector of $\mathbf{S}(u,v)$ (or the knot vector of the boundary curve $\mathbf{S}(u_s,v)$). Because $\mathbf{S}_{cap}^w(u,v)$ is a tensor-product surface, there are the relation between Equations (3.22) and (3.23) as given by

$$\tilde{\mathbf{c}}_i^{k,w} = \sum_{j=0}^{m_S} N_{j,q}(\upsilon_k)\,\mathbf{s}_{i,j}^{cap,w}\ ;\ \ k = 0,\ldots,m_S,\ \ i = 0,\ldots,m_S \tag{3.24}$$

Above relations can be represented in a matrix equation $\mathbf{C} = \mathbf{S}\,\mathbf{N}$ where $\mathbf{C}$, $\mathbf{S}$, $\mathbf{N}$ are $(5+1)\times(m_S+1)$, $(5+1)\times(m_S+1)$, $(m_S+1)\times(m_S+1)$ matrices. The control polygon $\{\mathbf{s}_{i,j}^{cap,w}\}$ is simply calculated by $\mathbf{S} = \mathbf{C}\,\mathbf{N}^{-1}$.

Shown in Figure 3.10 are the resulting surfaces (cap surface and adjacent surface) fitted to COP-data used in the fitting shown in Figure 3.7.



(a) (b)

Figure 3.10. Resulting surfaces of which degeneracy problem is resolved: (a) shaded image and (b) zoomed-in image for degenerate point and boundary between cap surface and adjacent surface

4. Reverse Engineering Application

4.1 Introduction

Whereas "conventional" engineering transforms engineering concepts and models into real parts, "reverse" engineering transforms real parts into engineering concepts and models [Vara1997]. In CAD/CAM and CG applications, the term of "(shape) reverse engineering" is commonly used for designation of the process that converts measured "shape" data obtained from a real part into a computer model, as a narrow interpretation of the term.

Reverse engineering have been traditionally used in several areas: 1) aesthetic design, 2) copy production of existing parts, 3) special custom design, *etc*. Recently, the need of reverse engineering is rapidly increasing because: 1) customers' respect is being transformed from functionality into design, and so, aesthetic surfaces are frequently used in the exterior design of industrial goods, 2) most of manufacturer offer effort to digitalize manufacturing information for high productivity, and 3) the requirement of 3D shapes in cultural industries including game, movie, advertisement, *etc*., is on the explosive increase. Also, advances in measuring technologies allow us to acquire "dense" measured data from real objects in reasonable time. Thus, to develop an efficient and "easy-to-use" *reverse engineering software* (**RE S/W**) is one of the needs of the times.

This Chapter is composed as follows: described in Section 4.2 is the overall process of reverse engineering. Given in Section 4.3 is how to apply the proposed NURBS-surface fitting method for COP-data arbitrary positioned in 3D space, especially focusing on the estimation of a local-coordinate system. The architecture of the RE S/W in which the proposed NURBS surface-fitting procedure is embedded is presented in Section 4.4 followed by reverse-engineering based surface modeling examples in Section 4.5.

4.2 Overall Process of Reverse Engineering

One of the most important characteristics of RE S/Ws is to actively utilize measured data with fine geometric details. For example, Imageware® (former Surfacer®) [Imag] provides various methods for generating section data, and ICEM Surf® [ICEM] *orthogonal-cross section* (**OCS**) data. These section data are used in surface modeling as reference data for a *generating curve*, *spine curve*, surface boundary curve, *trimming curve*, and so on. On the other hand, Krishnamurthy and Levoy [Kris1996] suggested a new surface-fitting pipeline where users paint boundary curves on "dense" and "irregular" polygonal mesh and fit a B-spline surface to each "segmented" *polygonal patch*. This *boundary-curve painting technology* is adopted by recently commercialized RE S/Ws, CySlice® [CySl], RapidForm® [Rapi], *etc*. Like above, by offering abilities to retrieve geometric information from measured data, RE S/Ws allows users (not even high-level experts of surface modeling) to reconstruct CAD models more quickly and easily than traditional *curve-based CAD modelers*.

Shown in Figure 4.1 is an overall process of reverse engineering, which consists of roughly four stages: 1) *structured-data* construction, 2) *segmentation*, 3) surface fitting, and 4) *skin-surface* modeling. The first stage is to construct an irregular polygonal mesh of arbitrary topology – so called a "triangular-net" – from measured data. Because whole data of real object can be obtained from many range scans, the structured-data are constructed by applying registration [Besl1992][Chen1992][Zhan1994][Gosh1997], integration [Turk1994][Curl1996] and 3D triangulation [Hopp1992][Baja1995][Amen1998][Bern1999] technologies. The second stage is to divide the triangular-net into sub-triangular nets so that each segmented sub-triangular net can be fitted to a single parametric-surface. Although much research on "automatic" segmentation has been done (also in computer vision and image processing field), there are a few can be applied to reverse engineering [Besl1988][Sapi1995][Eck1996][Milo1997]. Because segmentation requires the understanding of surfaces and also a kind of creativity, recently it looks common to shift the responsibility of segmentation on to users by offering "easy-to-use" segmentation tools, for example, a curve-network painting tool [Kris1996]. The third stage is to fit a parametric surface to each segmented sub-triangular net and the forth stage to make a "skin" surface by applying secondary-surface operations to fitted surfaces, such as *surface-surface intersection* (**SSI**), *blending*, *trimming*, *etc*.

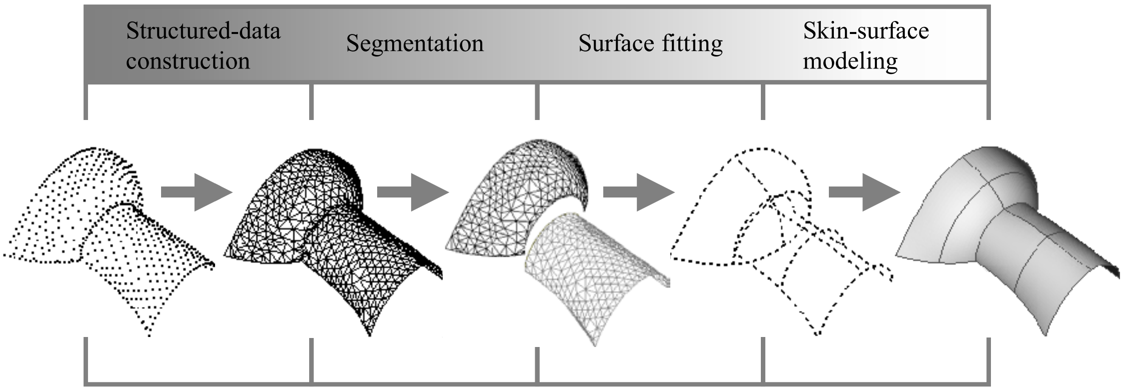


Figure 4.1. Overall process of reverse engineering

Segmentation and surface-fitting stages are inevitably connected to each other because the purpose of segmentation is to provide a sub-triangular net (or COP-data) appropriate to the surface-fitting method chosen by RE S/Ws. So far RE S/Ws have considered general B-spline surface methods, where 4-sided *base-surfaces* (including a plane) are generated by user-interaction [Ma1995][Kris1996] or automatically [Eck1995] for the parameterization purpose. However, these kinds of surface-fitting methods are not appropriate for a (sub-) triangular-net (or COP-data) representing a rotational-freeform shape, where the triangular-net (or COP-data) has to be divided again into several 4-sided polygonal-patches and each polygonal-patch is fitted to a parametric surface respectively.

Therefore, how to apply the NURBS surface-fitting method proposed in Chapter 3 to reverse engineering is to be presented in the following sub-section. If the NURBS surface-fitting modules for rotational-freeform shapes is embedded in RE S/Ws, users can choose the surface fitting methods (general B-spline surface fitting methods or the proposed NURBS-surface fitting method) and perform the segmentation process according to the partial shape of the whole triangular-net. In consequence, it can be expected for surface modeling with the RE S/W to be more efficient – the numbers of surfaces and secondary-surface operations can be smaller and user’s interaction can be reduced, too.

4.3 NURBS-surface Fitting Procedure for Reverse Engineering

The procedure to apply the NURBS-surface fitting method proposed in the previous Chapter, consists of following four steps: 1) local-coordinate system estimation, 2) local-coordinate transform, 3)

NURBS-surface fitting, and 4) global-coordinate transform of the NURBS surface.

Step 1. Local-coordinate System Estimation

To apply the proposed NURBS-surface fitting method for reverse engineering, most of all, it is required to define a *local Cartesian-coordinate system* where the orthogonal coordinates are defined. Some studies dealt with finding the rotational axis of surfaces-of-rotation from COP-data [Elsa1996] [Pott1998a][Pott1998b][Lai2000], however, they are not appropriate for COP-data from general rotational-freeform shapes. On the other hand, the *principal component analysis* (**PCA**), which is widely used in the field of image processing and computer vision [Jain1989][Prat1991][Lay1996] (it is also known as *Karhunen-Loeve transform*), may be used to define a local-coordinate system. PCA works well for COP-data of the whole of a rotational freeform shape, but not for those of a part, because it finds the oriented-bounding box of COP-data [Gott1996] (PCA looks more appropriate for general B-spline surface fitting rather than the proposed NURBS-surface fitting method). The local-coordinate system estimation method to be presented is based on the *general-quadric surface approximation*.

A general-quadric surface is a 2^{nd} degree algebraic surface as follows:

$$q(x,y,z) = \mathbf{x}^T \mathbf{Q}_u \mathbf{x} + \mathbf{v}^T \mathbf{x} + d = 0;$$
$$\mathbf{Q}_u = \begin{bmatrix} q_1 & q_4/2 & q_6/2 \\ q_4/2 & q_2 & q_5/2 \\ q_6/2 & q_5/2 & q_3 \end{bmatrix}, \mathbf{v} = \begin{bmatrix} q_7 \\ q_8 \\ q_9 \end{bmatrix}, d = q_{10}, \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4.1a}$$

or

$$q(x,y,z) = \tilde{\mathbf{x}}^T \mathbf{Q}\, \tilde{\mathbf{x}} = 0;$$
$$\mathbf{Q} = \begin{bmatrix} q_1 & q_4/2 & q_6/2 & q_7/2 \\ q_4/2 & q_2 & q_5/2 & q_8/2 \\ q_6/2 & q_5/2 & q_3 & q_9/2 \\ q_7/2 & q_8/2 & q_9/2 & q_{10} \end{bmatrix}, \tilde{\mathbf{x}} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4.1b}$$

The coefficients of Equation (4.1) $\{q_i\}$ can be determined so that the quadric surface is fitted to given COP-data in the least-squares sense [Faug1986][Mars1992]. The fitted quadric surface may be classified into different types based on its coefficients [Levi1976][Zwil1996] as listed on Table 4.1.

Table 4.1. Quadric-surface classification table [Zwil1996]

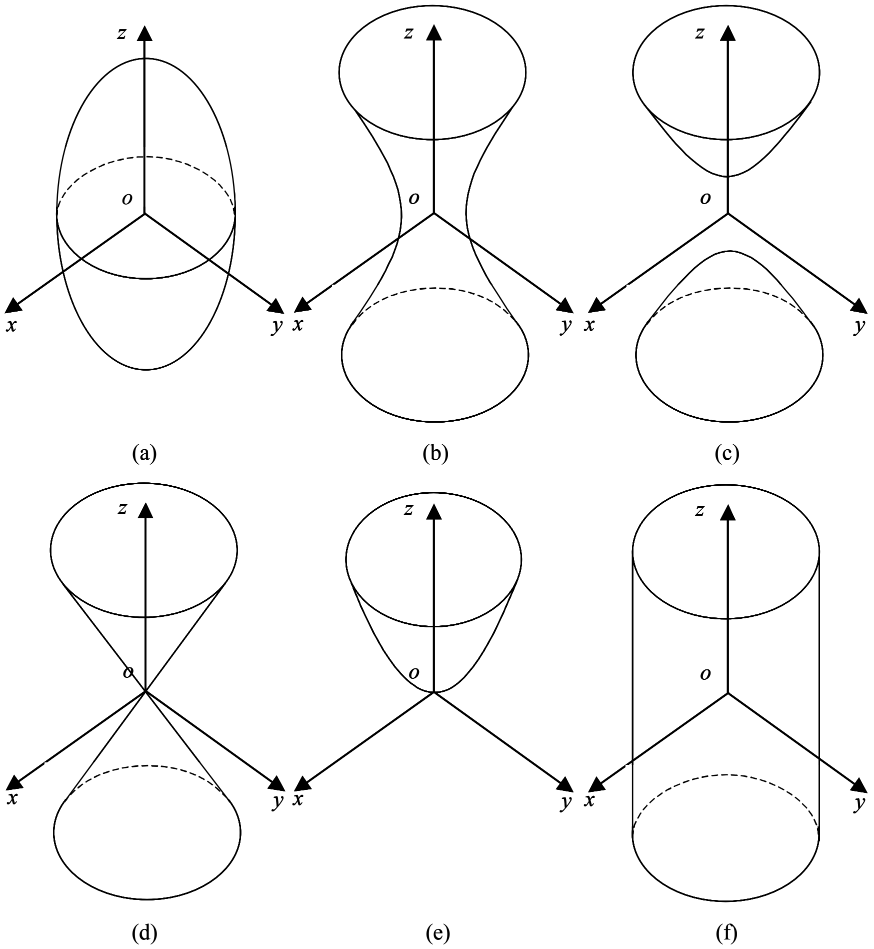
$r(\mathbf{Q}_u)$	$r(\mathbf{Q})$	$d(\mathbf{Q})$	k	K	Type of quadric surface	Canonical form
3	4	-			Real ellipsoid	$x^2/a^2 + y^2/b^2 + z^2/c^2 = 1$
3	4	+	=		Imaginary ellipsoid	$x^2/a^2 + y^2/b^2 + z^2/c^2 = -1$
3	4	+	≠		Hyperboloid of 1 sheet	$x^2/a^2 + y^2/b^2 - z^2/c^2 = 1$
3	4	-	≠		Hyperboloid of 2 sheets	$x^2/a^2 + y^2/b^2 - z^2/c^2 = -1$
3	3		≠		Real quadric cone	$x^2/a^2 + y^2/b^2 - z^2/c^2 = 0$
3	3		=		Imaginary quadric cone	$x^2/a^2 + y^2/b^2 + z^2/c^2 = 0$
2	4	-	=		Elliptic paraboloid	$x^2/a^2 + y^2/b^2 + 2z = 0$
2	4	+	≠		Hyperbolic paraboloid	$x^2/a^2 - y^2/b^2 + 2z = 0$
2	3		=	≠	Real elliptic cylinder	$x^2/a^2 + y^2/b^2 = 1$
2	3		=	=	Imaginary elliptic cylinder	$x^2/a^2 + y^2/b^2 = -1$
2	3		≠		Hyperbolic cylinder	$x^2/a^2 - y^2/b^2 = 1$
2	2		≠		Real intersecting planes	$x^2/a^2 - y^2/b^2 = 0$
2	2		=		Imaginary intersecting planes	$x^2/a^2 + y^2/b^2 = 0$
1	3				Parabolic cylinder	$x^2 + 2y = 0$
1	2			≠	Real parallel planes	$x^2 = 1$
1	2			=	Imaginary parallel planes	$x^2 = -1$
1	1				Coincident planes	$x^2 = 0$

$r(\mathbf{Q}_u) \equiv rank\ \mathbf{Q}_u$, $d(\mathbf{Q}_u) \equiv det\ \mathbf{Q}_u$,

k : signs of non-zero eigenvalues of $\mathbf{Q}_u$, K : signs of non-zero eigenvalues of $\mathbf{Q}$,

$=$: same, $\neq$: opposite

Among 17 types listed on Table 4.1, the following six types are relevant to the local-coordinate system estimation: real ellipsoid, hyperboloid of one sheet, hyperboloid of two sheets, real quadric cone, elliptic paraboloid, real elliptic cylinder (See Figure 4.2). Namely, when COP-data are fitted to one of the six quadric surface types, the local-coordinate system $\omega = \{\mathbf{o}, \mathbf{i}, \mathbf{j}, \mathbf{k}\}$ can be determined.



(a) (b) (c)

(d) (e) (f)

Figure 4.2. Six quadric surfaces relevant to local-coordinate system estimation: (a) real ellipsoid, (b) hyperboloid of one sheet, (c) hyperboloid of two sheets, (d) real quadric cone, (e) elliptic paraboloid, (f) real elliptic cylinder

The origin $\mathbf{o}$ of the local-coordinate system can be obtained from the following relation:

$$2\mathbf{Q}_u\mathbf{x}+\mathbf{v}=\mathbf{0} \tag{4.2}$$

In the case of $rank\,\mathbf{Q}_u=3$ (real ellipsoid, hyperboloid of one sheet, hyperboloid of two sheets, and real quadric cone), the origin is given by [Faug1986][Mars1992]:

$$\mathbf{o}=-\frac{1}{2}\mathbf{Q}_u^{-1}\mathbf{v} \tag{4.3}$$

For $rank\,\mathbf{Q}_u = 2$ (elliptic paraboloid and real elliptic cylinder), the origin of the local coordinate is given by:

$$\mathbf{o} = \mathbf{p} + t_o\mathbf{q} \tag{4.4}$$

where $\mathbf{p}$, $\mathbf{q}$, t_o can be referred to Appendix E.

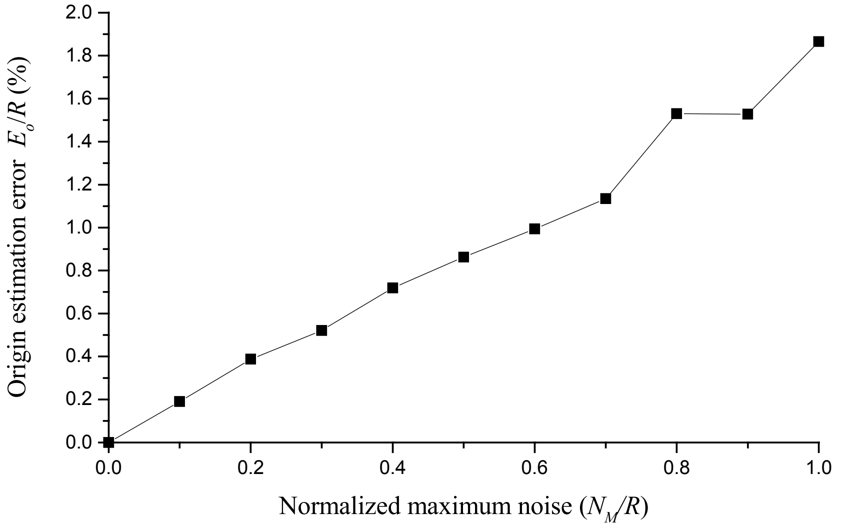
The axes of the local coordinate system are related to the eigensystem of *subdiscriminant matrix* $\mathbf{Q}_u$. There are three cases to assign the eigenvectors to $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$.

First, in the case of $rank\,\mathbf{Q}_u = 2$ (elliptic paraboloid and real elliptic cylinder), the *z-axis unit vector* $\mathbf{k}$ is set to the eigenvector having zero (smallest) eigenvalue. The length of each semi-axis of a real ellipsoid is reciprocal to the squares of corresponding eigenvalue [Pett1970]. Thus, the rotational axis of a surface-of-revolution shape becomes the eigenvector with zero (smallest) eigenvalue.

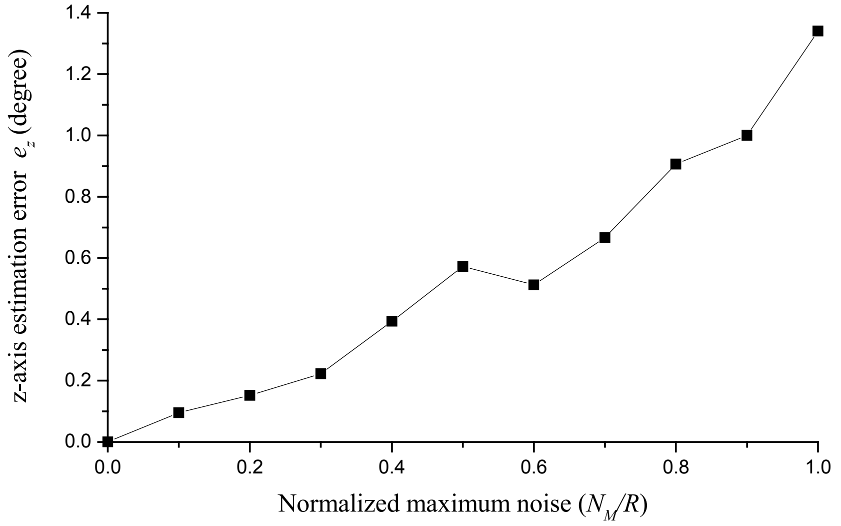
Second, in the case that $rank\,\mathbf{Q}_u = 3$ and all the three eigenvalues have not the same sign (hyperboloid of one sheet, hyperboloid of two sheets, real quadric cone), the only one eigenvector having the opposite sign is assigned to $\mathbf{k}$, which can work well as the rotational axis of a surface-of-revolution shape.

Finally, for a real ellipsoid, every eigenvector can be a candidate for $\mathbf{k}$. In many cases, it is desirable to set $\mathbf{k}$ to the eigenvector having the largest eigenvalue because it makes the shape near the degenerate point of the resulting NURBS surface natural.

The local-coordinate estimation method was applied to a set of COP-data obtained by “virtually digitizing” spheres and cylinders with different values of radius R (cylinders’ length is set to $2R$). Random noises sampled from normal distribution $N(0,\sigma^2)$ were added to the *x*-, *y*-, *z*-components of the COP-data points. Shown in Figure 4.3a are *relative origin-estimation errors* (in %) plotted against the *normalized maximum noise* N_M/R (for sphere COP-data), where $N_M = 3\sigma$. The *z-axis estimation errors* e_z (in degrees) plotted against N_M/R are shown in Figure 4.3b (for cylinder COP-data).



(a)



(b)

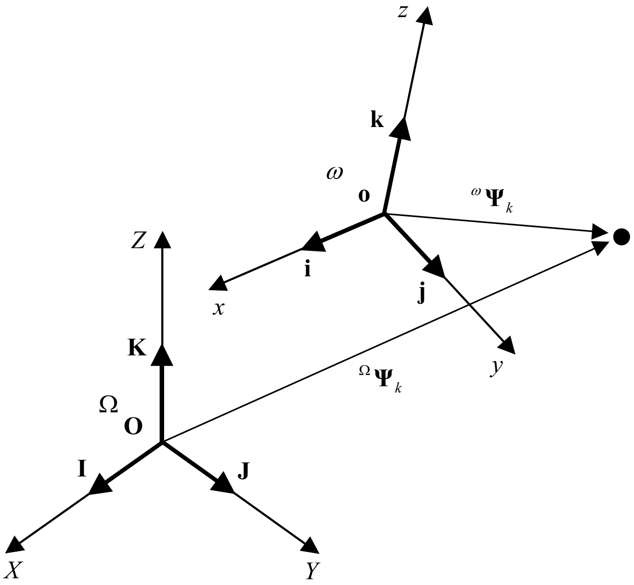
Figure 4.3. Empirical results of local-coordinate system estimation

Step 2. Local-coordinate Transform

Once the local-coordinate system is determined, the each point of COP-data is transformed from global coordinates ${}^{\Omega}\boldsymbol{\Psi}_k$ to local coordinates ${}^{\omega}\boldsymbol{\Psi}_k$ using following relation (See Figure 4.4):

$$ {}^{\omega}\boldsymbol{\Psi}_k = \mathbf{R}({}^{\Omega}\boldsymbol{\Psi}_k - \mathbf{T})\,; \quad \mathbf{R} = \begin{bmatrix} \mathbf{i}\bullet\mathbf{I} & \mathbf{i}\bullet\mathbf{J} & \mathbf{i}\bullet\mathbf{K} \\ \mathbf{j}\bullet\mathbf{I} & \mathbf{j}\bullet\mathbf{J} & \mathbf{j}\bullet\mathbf{K} \\ \mathbf{k}\bullet\mathbf{I} & \mathbf{k}\bullet\mathbf{J} & \mathbf{k}\bullet\mathbf{K} \end{bmatrix}, \quad \mathbf{T} = \mathbf{o} - \mathbf{O} \tag{4.5} $$

where Ω denotes the global-coordinate system with $\{\mathbf{O},\mathbf{I},\mathbf{J},\mathbf{K}\}$ and ω the local-coordinate system with $\{\mathbf{o},\mathbf{i},\mathbf{j},\mathbf{k}\}$.



z
k
ω
o
${}^{\omega}\boldsymbol{\Psi}_k$
Z
i
x
j
K
y
${}^{\Omega}\boldsymbol{\Psi}_k$
Ω
O
I
J
X
Y

Figure 4.4. Global- and local-coordinate systems

Step 3. NURBS-surface Fitting

By applying the NURBS-surface fitting method proposed in the Chapter 3, a spherical or cylindrical NURBS-surface is fitted to local 3D Cartesian-coordinate points $\{{}^{\omega}\boldsymbol{\Psi}_k\}$.

Step 4. Global-coordinate Transform of NURBS surface

Based on the well-known "affine-transform invariance" property [Pieg1995], the NURBS surface fitted in the local-coordinate system can be retransformed into the global-coordinate system by transforming each control-point from local coordinates ${}^{\omega}\mathbf{s}_{i,j}$ to global coordinates ${}^{\Omega}\mathbf{s}_{i,j}$ using following relation:

$$ {}^{\Omega}\mathbf{s}_{i,j} = \mathbf{R}^{-1}\,{}^{\omega}\mathbf{s}_{i,j} + \mathbf{T} \tag{4.6} $$

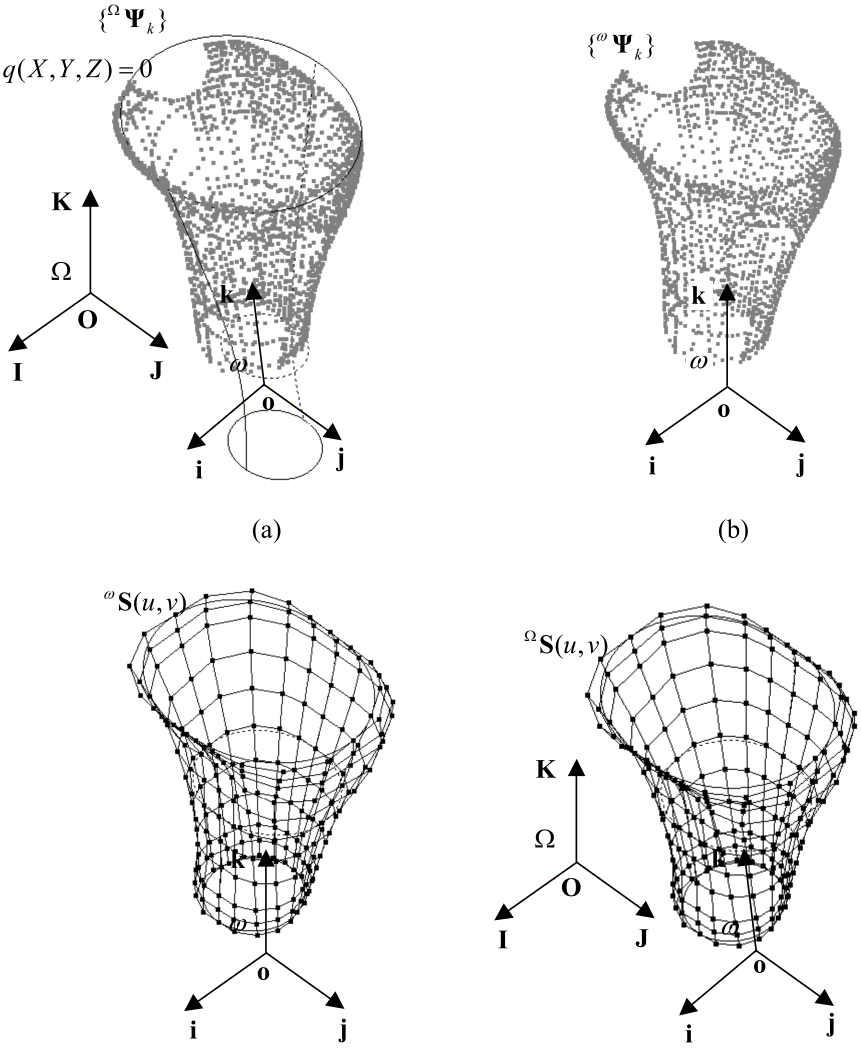
Shown in Figure 4.5 is an example of cylindrical NURBS-surface fitting for reverse engineering. Depicted in Figure 4.5a are given COP-data $\{{}^{\Omega}\mathbf{\Psi}_k\}$ and the quadric surface approximated having a general form as:

$$ \begin{aligned} q(X,Y,Z) = {} & 0.808678\,X^2 + 0.574794\,Y^2 - 0.045684\,Z^2 \\ & + 0.015879\,XY + 0.002311\,YZ - 0.163925\,ZX \\ & + 49.971136\,X - 43.447695\,Y - 9.216195\,Z \\ & + 1443.009728 \\ & = 0 \end{aligned} \tag{4.7} $$

The type of the quadratic surface is a hyperboloid of one sheet ($r(\mathbf{Q}_u) = 3$, $r(\mathbf{Q}) = 4$, $d(\mathbf{Q}) = 0.023747$, $k :\neq$, $K :\neq$). The estimated local-coordinate system ω is determined as follows:

$$ \begin{aligned} \mathbf{o} &= (\text{-}35.028812, 38.352442, \text{-}37.053139), \\ \mathbf{i} &= (0.995002, 0.032202, \text{-}0.094521), \\ \mathbf{j} &= (\text{-}0.031771, 0.999477, 0.006060), \\ \mathbf{k} &= (0.094667, \text{-}0.003027, 0.995504) \end{aligned} \tag{4.8} $$

where $\mathbf{k}$ is set to the eignevector whose its eigenvalue has the minus-sign alone.


$\{{}^{\Omega}\mathbf{\Psi}_k\}$
$q(X,Y,Z)=0$
K
Ω
O
I
J
k
ω
o
i
j
$\{{}^{\omega}\mathbf{\Psi}_k\}$
k
ω
o
i
j
${}^{\omega}\mathbf{S}(u,v)$
k
ω
o
i
j
${}^{\Omega}\mathbf{S}(u,v)$
K
Ω
O
I
J
k
ω
o
i
j

(a) (b)

(c) (d)

Figure 4.5. Cylindrical NURBS-surface fitting for reverse engineering

4.4 Reverse-engineering Software Development

The procedure of NURBS surface-fitting method using orthogonal-coordinate transform for reverse engineering presented in the previous sub-section is implemented as one component of the *surface fitting engine* for a “research-purposed” RE S/W whose the system architecture is shown in Figure 4.6 (the data flow in the system is given in Figure 4.7).

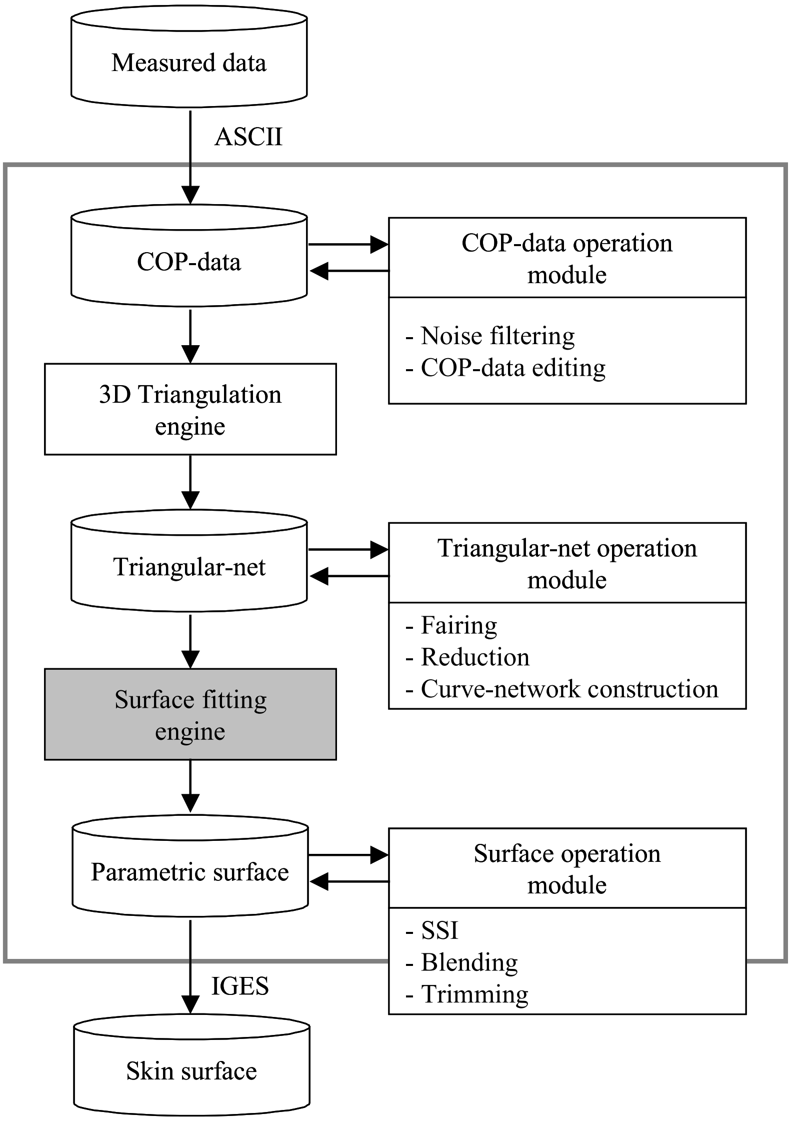


Figure 4.6. System architecture of RE S/W in which proposed NURBS-surface fitting procedure is embedded

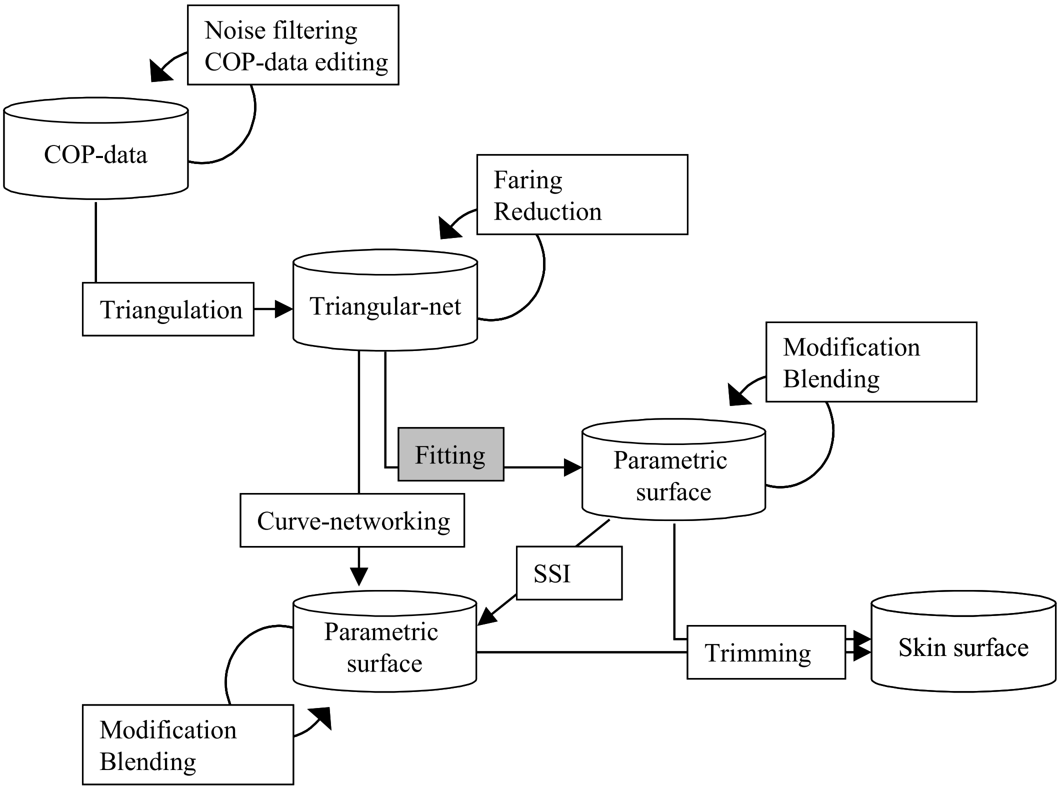
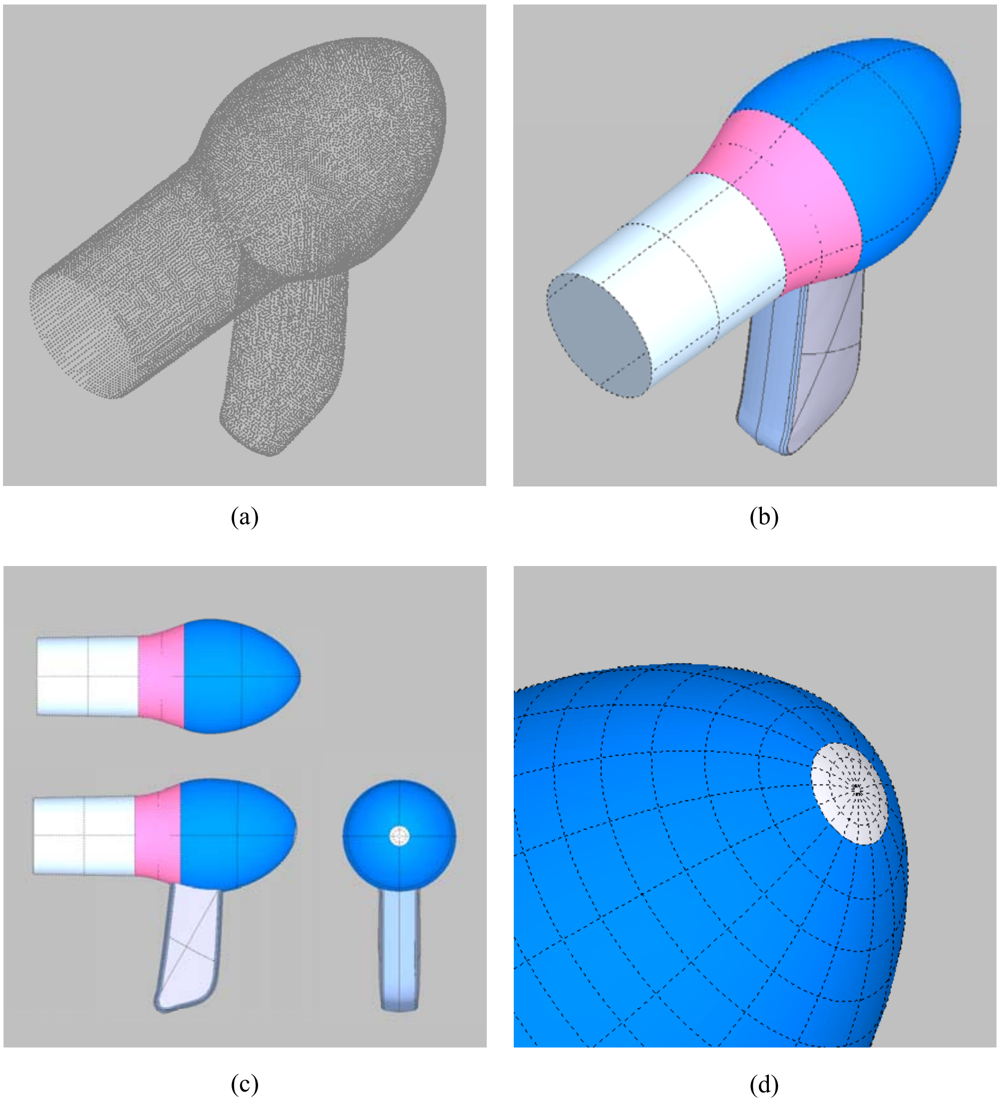


Figure 4.7. Data flow in RE S/W

4.5 Examples

Presented in this sub-section are examples of surface modeling for various kinds of COP-data with the RE S/W in which the proposed NURBS surface-fitting procedure is implemented. Shown in Figure 4.8a and Figure 4.8b are COP-data sampled from a hair-drier and the resulting CAD model, respectively. The "head" part of COP-data is fitted to NURBS surfaces by applying the proposed NURBS surface-fitting procedure: The "mouth" part, which is a true surface-of-revolution, was fitted to a cylindrical NURBS-surface, and the "occiput" part, which is a rotational-freeform shape (sphere-like shape) as depicted in Figure 4.8c, was fitted to a spherical NURBS-surface. The degenerate problem the closed spherical NURBS-surface fitting has was resolved with a cap surface as shown in Figure 4.8d.



(a)

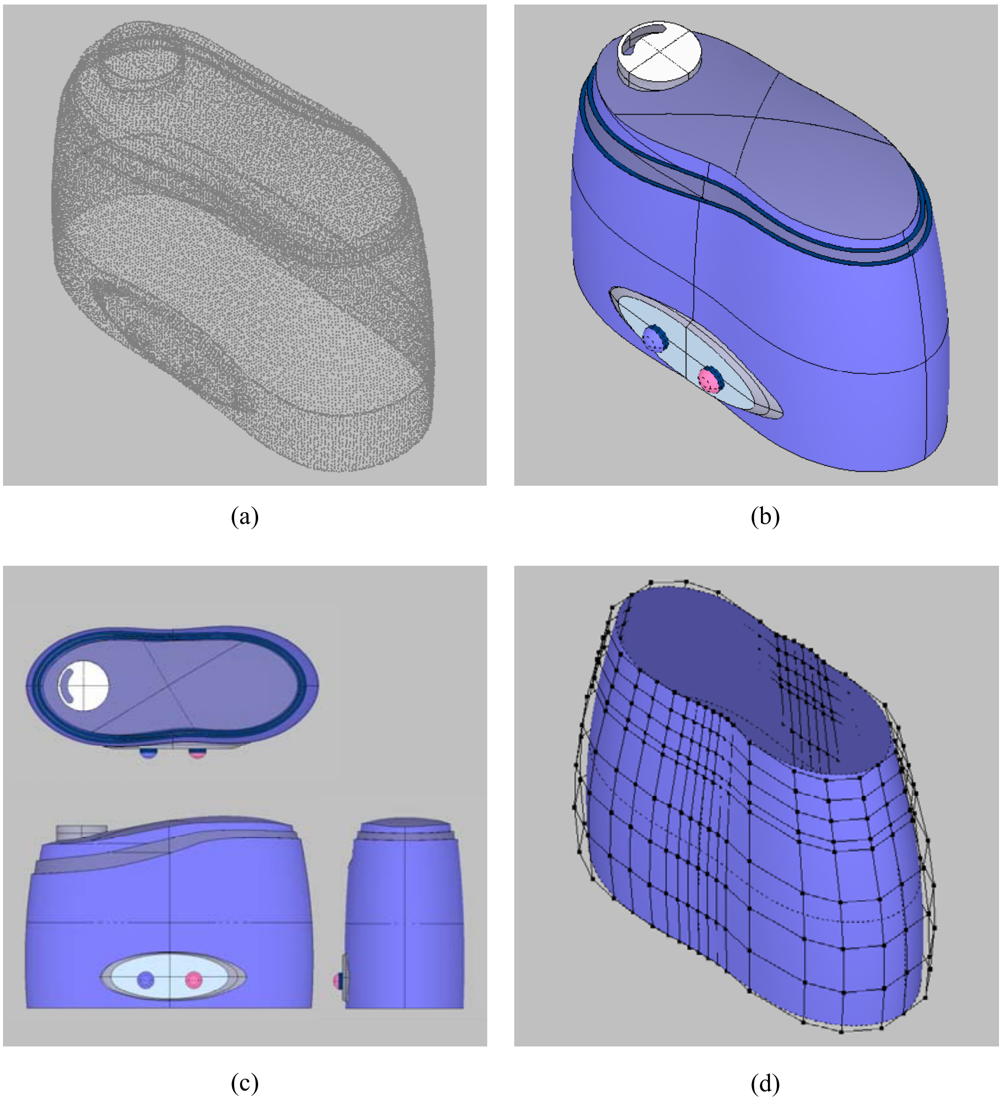
(b)

(c)

(d)

Figure 4.8. Surface modeling for COP-data from hair-drier

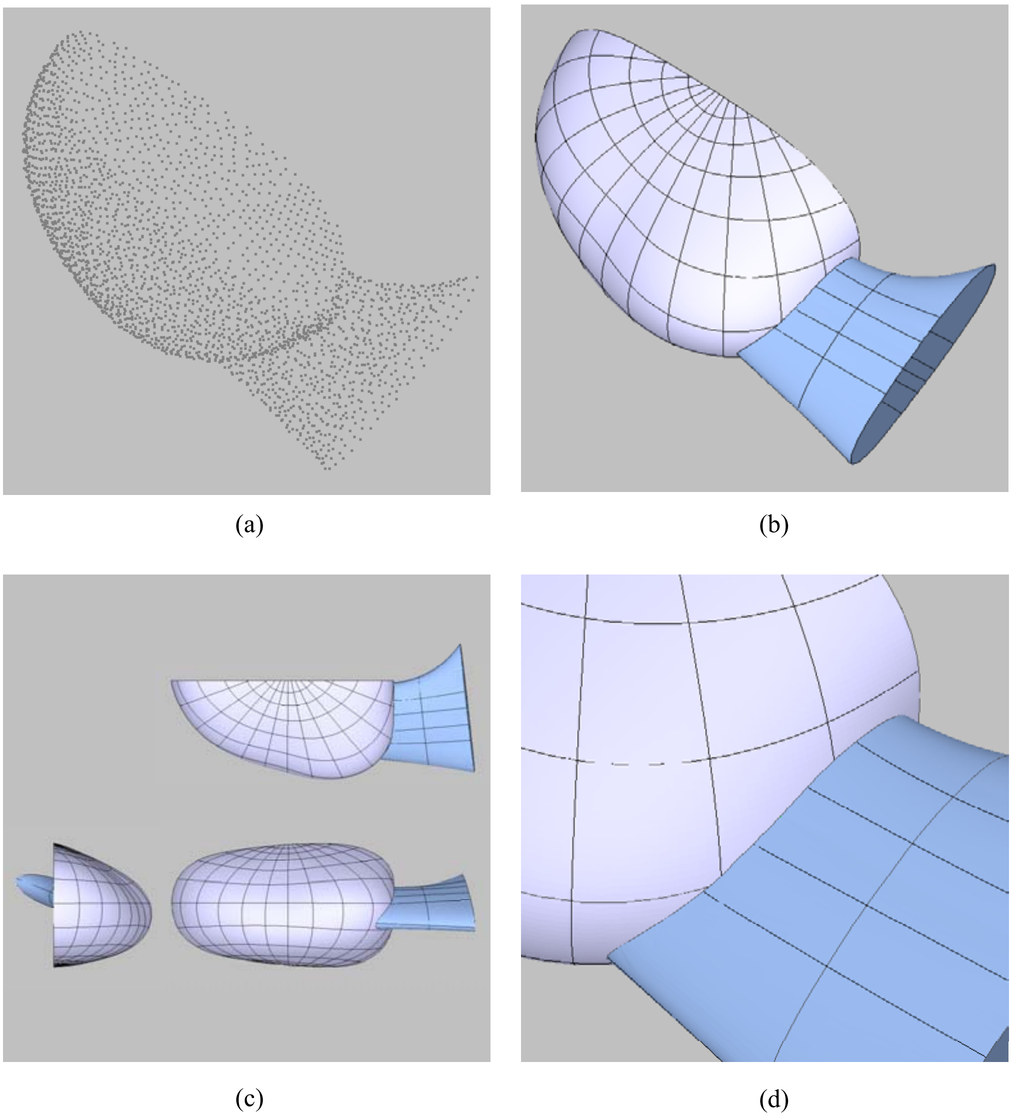
Given in Figure 4.9a and Figure 4.9b are COP-data sampled from a humidifier and the resulting CAD model, respectively. The “body” surface is wholly constructed by applying the cylindrical NURBS-surface fitting method (Observe the control net of the surface depicted in Figure 4.9d).



(a) (b) (c) (d)

Figure 4.9. Surface modeling for COP-data from humidifier

Figure 4.10 shows the surface-modeling example for an outside rear-view mirror. Figure 4.10a and Figure 4.10b are COP-data and the resulting CAD model, respectively. Because both the "housing" and "bracket" surfaces were constructed as one NURBS surface, respectively, the final CAD model was obtained by applying only one secondary-surface operation (or *trimming*). Figure 4.10d shows the *trimmed-surface boundary* between two NURBS *faces* (or *trimmed surface*).



(a)

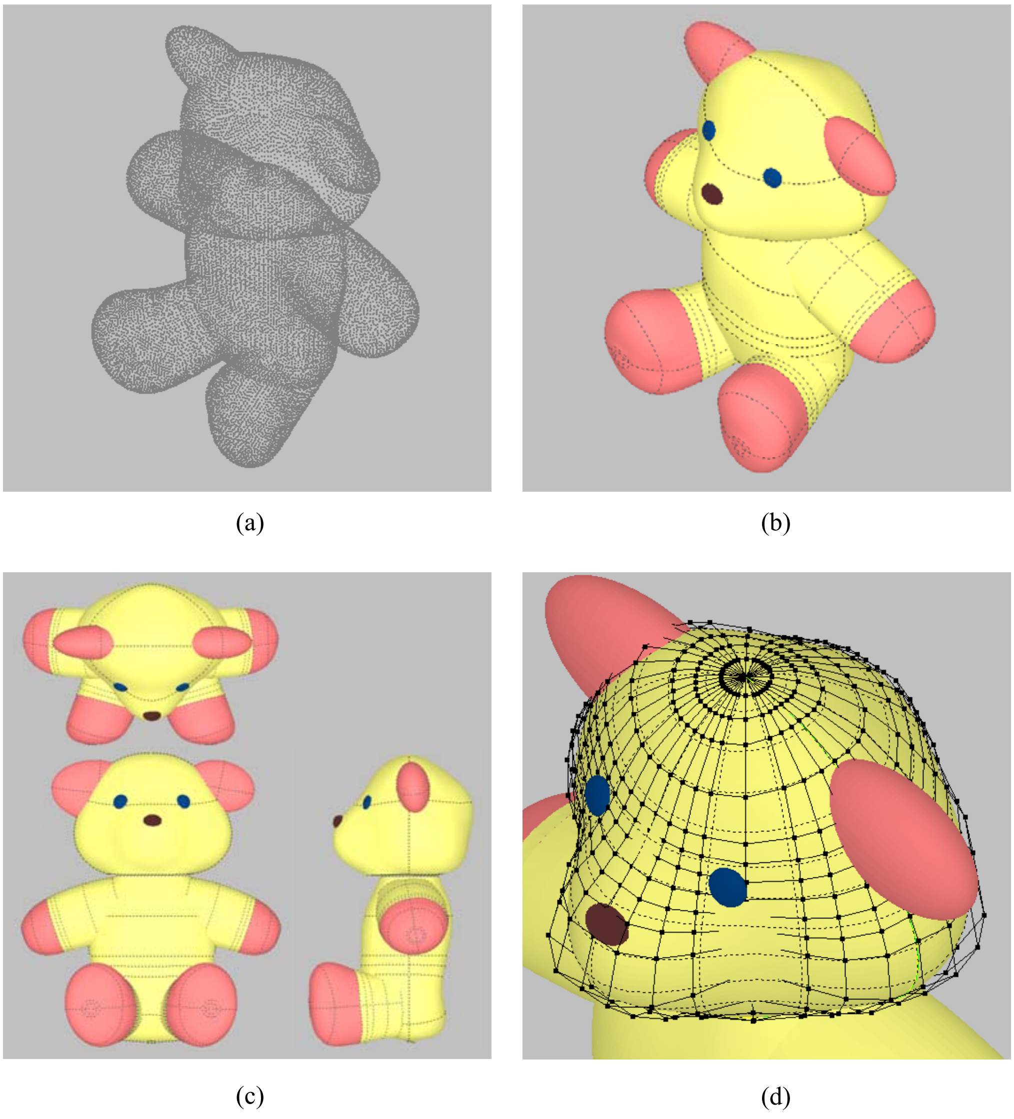
(b)

(c)

(d)

Figure 4.10. Surface modeling for COP-data from outside rear-view mirror

The proposed NURBS-surface fitting procedure can be applied to “realistic rendering” and/or “character modeling” purposed S/Ws [3dsM][Rhin]. The most standard modeling method provided by these kinds of S/Ws is as follows: 1) construction of a NURBS sphere or cylinder, 2) modification of the NURBS surface (sphere or cylinder before modification) by relocating the control points until users are satisfied with the resulting shape. The process is very tedious and time-consuming. If COP-data from *physical maquettes* can be imported to the S/W, the proposed NURBS-surface fitting procedure may provide a good starting point for artistic works. Shown in Figure 4.11a are COP-data sampled from a teddy bear. The almost parts of the resulting 3D model are constructed by applying the proposed NURBS-surface fitting procedure (See Figure 4.11b and Figure 4.11c). The "head" part of the teddy bear is composed of only one spherical NURBS-surface as shown in Figure 4.11d – it is very similar to the resulting surface modeled by the commercial S/Ws such as 3ds Max® [3dsM] and Rhino® [Rhin], *etc*.


(a)

(b)

(c)

(d)

Figure 4.11. Surface modeling for COP-data from teddy bear

When modifying the resulting NURBS-surface by relocating control points, a special consideration is required. Because the resulting NURBS-surface has “mathematically” C^0 continuity at the knots of NURBS base geometry (“geometrically” G^1) – it is the characteristic of NURBS, for smooth shape modification, the control-points corresponding to these knot positions must be moved together with adjacent control-points so that the tangent-plane condition is kept.

5. Other Applications

5.1 Mechanical-cam Design

A mechanical cam, one of the most "versatile" mechanical element, is used to transform one motion into another. Among various types of cams, a plate (or disk) cam and cylindrical cam with translating follower, which are the most commonly used, are conventionally designed in a polar-coordinate system. The design of these kinds of the cam is how to synthesize a *follower-displacement curve*, $R(\theta)$, which is composed of the *rise*, *dwell*, and *return* periods (See Figure 5.1). With considerations of dynamics of cam motion, each rise- or return-curve segment has to be constrained under endpoint conditions (See the rise period in Figure 5.1):

$$R(\theta_s) = R_s\,;\quad R'(\theta_s) = 0\,;\quad R''(\theta_s) = 0 \tag{5.1a}$$

$$R(\theta_e) = R_e\,;\quad R'(\theta_e) = 0\,;\quad R''(\theta_e) = 0 \tag{5.1b}$$

To satisfy above conditions, a polynomial function [Roth1956] or B-spline function [Ange1983] [MacC1988] is commonly employed for synthesis of $R(\theta)$.

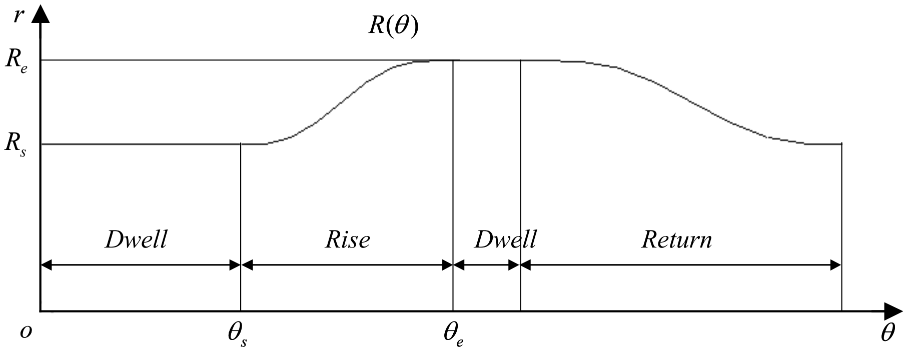


Figure 5.1. Follower-displacement curve

Once a follower-displacement curve is synthesized, so the shape of the plate or cylindrical cam is perfectly determined. However, there is one problem – how to express the shape of the cam in a Cartesian-coordinate system in the form of NURBS, which is the most standard curve or surface representation for CAD/CAM/CAE systems. As so far, there have not been any solutions of the exact transformation. In this sub-section, an approximate approach is presented, which is based on the basic idea given in Section 2.1.

As mentioned earlier, it is impossible to obtain the 2D NURBS curve by applying the direct symbolic-product operation of $R(\theta)$ with the base circle $\mathbf{B}(u)$ because the parameters, θ and u, are not linearly related. For the curve and surface fitting presented in Chapter 2 and Chapter 3, respectively, so-called the parameter synchronization is introduced to solve the problem. However, for the problem of this sub-section, it cannot be done to re-parameterize $R(\theta)$ along $\mathbf{B}(u)$. So, inversely, a NURBS base-circle whose the flow rate is “almost” uniform is looked for.

Blanc and Schlick [Blan1996] suggested a quartic NURBS-circle with its flow rate is almost uniform. This *quasi-uniform NURBS circle* $\tilde{\mathbf{B}}(\theta)$ is composed of three, four, or six rational Bezier curves of degree 4, $\{\tilde{\mathbf{B}}^j(\eta)\}$. Each rational Bezier curve having arc angle 2ϕ is given by:

$$\tilde{\mathbf{B}}^j(\eta) = \frac{\sum_{i=0}^{4} B_{i,4}(\eta)\,\tilde{w}_i^j \tilde{\mathbf{b}}_i^j}{\sum_{i=0}^{4} B_{i,4}(\eta)\,\tilde{w}_i^j}; \tag{5.2}$$

$$\tilde{w}_0^j = 1,\ \tilde{w}_1^j = \frac{1+w}{2}\alpha,\ \tilde{w}_2^j = \frac{\alpha^2 + \alpha^2 w + w}{3},\ \tilde{w}_3^j = \frac{1+w}{2}\alpha,\ \tilde{w}_4^j = 1;$$

$$\tilde{\mathbf{b}}_0^j = \mathbf{b}_0^j,\ \tilde{\mathbf{b}}_1^j = \frac{\mathbf{b}_0^j + w\mathbf{b}_1^j}{1+w},\ \tilde{\mathbf{b}}_2^j = \frac{\alpha^2\,\mathbf{b}_0^j + 2w(1+\alpha^2)\mathbf{b}_1^j + \alpha^2\,\mathbf{b}_2^j}{2(\alpha^2 + \alpha^2 w + w)},\ \tilde{\mathbf{b}}_3^j = \frac{\mathbf{b}_2^j + w\mathbf{b}_1^j}{1+w},\ \tilde{\mathbf{b}}_4^j = \mathbf{b}_2^j;$$

$$\alpha = \frac{4 - 2\cos^3(\phi/5) + \cos(\phi/5)}{-1 + 8\cos^3(\phi/5) - 4\cos(\phi/5)}$$

where $\{w_i^j\} = \{w_0^j, w_1^j, w_2^j\} = \{1, w, 1\}$, $\{\mathbf{b}_i^j\}$ are the weights and control points of the corresponding RQB-curve $\mathbf{B}^j(\eta)$ having the same arc angle 2ϕ, respectively.

First, let us consider how uniform the flow rate of the quasi-uniform arc is. Shown in Figure 5.2 is the angle deviation of the quasi-uniform arc from a uniform arc, $\Delta(\eta) = \tilde{\xi}(\eta) - 2\phi\eta$ where $\tilde{\xi}(\eta)$ is given by [Blan1996]

$$\tilde{\xi}(\eta) = 2\tan^{-1}\left(\frac{s(\eta)\sin\phi}{1-s(\eta)(1-\cos\phi)}\right);$$
$$s(\eta) = \frac{\alpha\eta+(1-\alpha)\eta^2}{1-2(1-\alpha)\eta+2(1-\alpha)\eta^2} \tag{5.3}$$

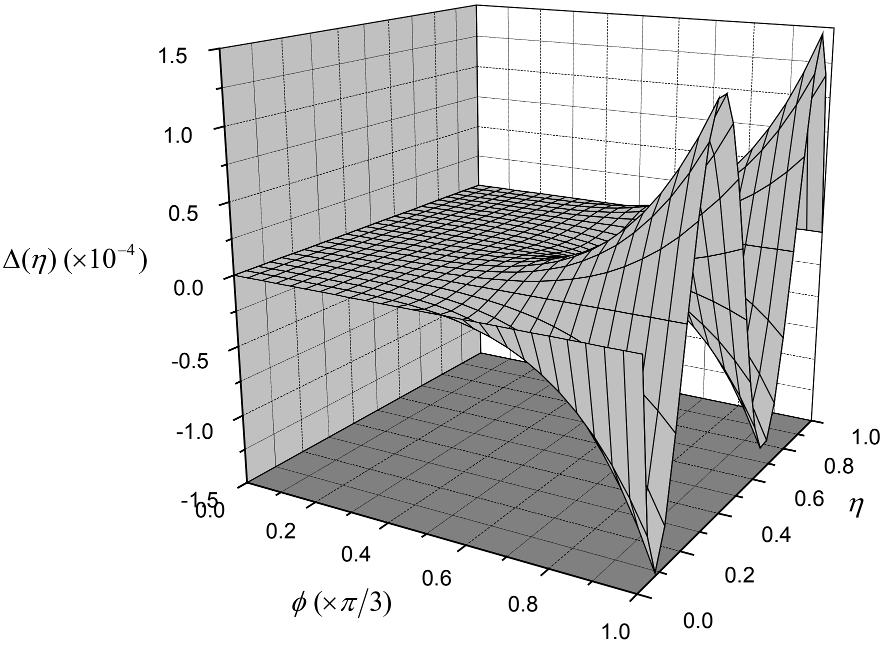
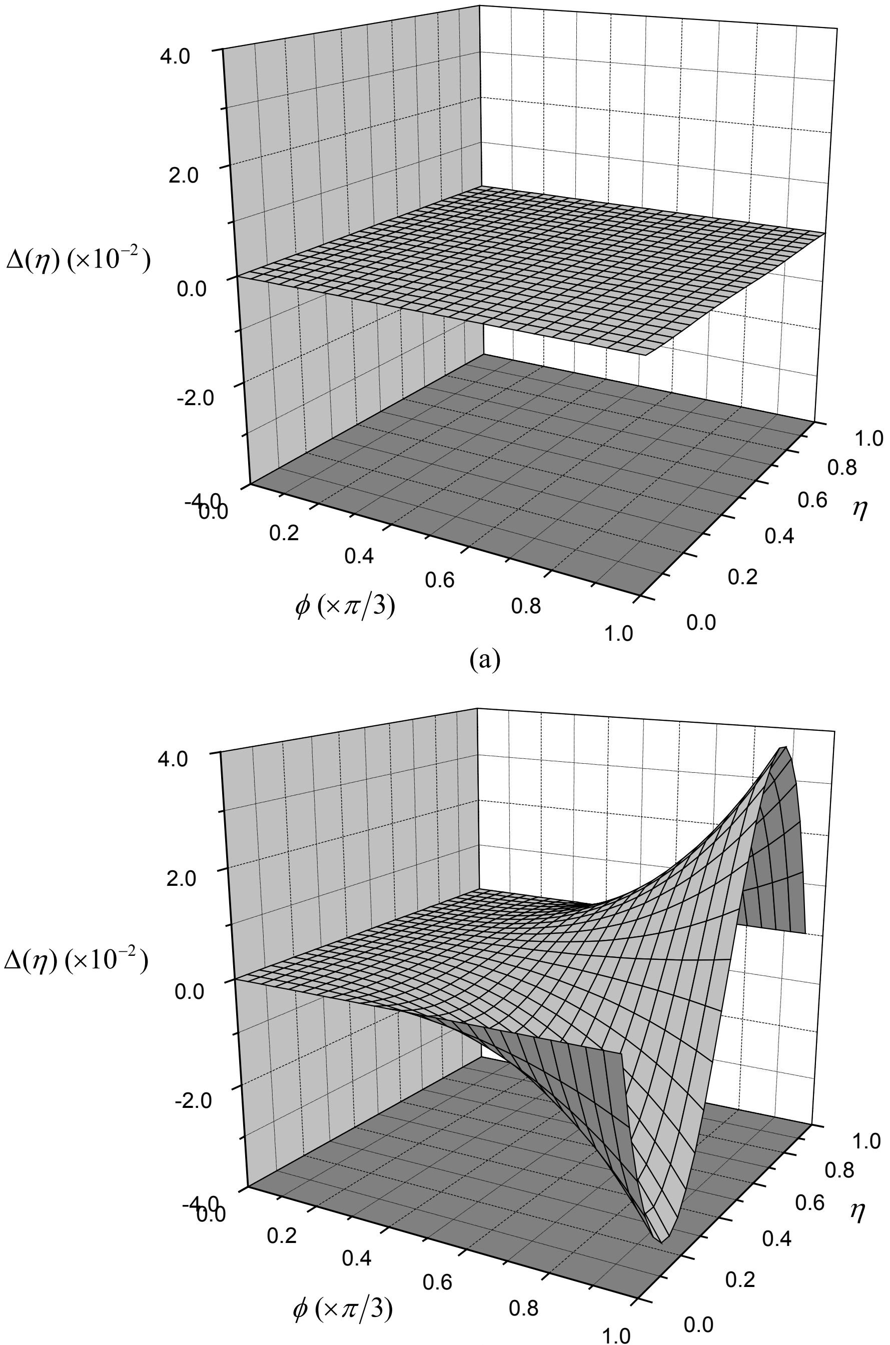


Figure 5.2. Angle deviation of quasi-uniform arc

With comparison to ordinary RQB-arc on the same scale (See Figure 5.3), the uniformity of the quasi-uniform arc is more noticeable – the ratio of the maximum-angle deviation of the ordinary RQB-arc to that of the quasi-uniform arc is more than hundreds. The ratios are about 250, 500, 1000 for the triangle-based circle $(2\phi = 2\pi/3)$ [Vers1975], square-based circle $(2\phi = \pi/2)$, hexagon-based circle $(2\phi = \pi/3)$ [Till1983], respectively.



(a)

(b)

Figure 5.3. Angle deviation: (a) quasi-uniform arc and (b) RQB arc

The radial deviation of the NURBS curve obtained by a symbolic-product operation of $R(2\phi\eta)$ with a quasi-uniform circle $\widetilde{\mathbf{B}}(\eta)$ is calculated approximately,

$$\begin{aligned}\left|\Delta R(\eta)\right| &= \left|R(\widetilde{\xi}(\eta)) - R(2\phi\eta)\right| = \left|R(2\phi\eta + \Delta(\eta)) - R(2\phi\eta)\right| \\ &\cong \left|\{R(2\phi\eta) + R'(2\phi\eta)\Delta(\eta)\} - R(2\phi\eta)\right| = \left|R'(2\phi\eta)\Delta(\eta)\right|\end{aligned} \tag{5.4}$$

Given a conversion tolerance ε, the number of the arc segments of which the quasi-uniform base-circle consists, can be calculated for successful conversion so that $\left|R'(2\phi\eta)\Delta(\eta)\right| \le \varepsilon$. Because $\left|R'(2\phi\eta)\Delta(\eta)\right| \le \left|R'(2\phi\eta)\right|\left|\Delta(\eta)\right| \le \left|R'(2\phi\eta)\right|\left|\Delta(\eta)\right|_{\max}$, once $\left|R'(\theta)\right|_{\max}$ is calculated, the arc angle $2\phi^*$ satisfies $\left|\Delta(\eta)\right|_{\max} = \varepsilon / \left|R'(\theta)\right|_{\max}$, can be obtained from the relation between ϕ and $\left|\Delta(\eta)\right|_{\max}$ shown in Figure 5.4. Then, the 2D NURBS curve of the cam is constructed from $\mathbf{C}(\theta) = R(\theta)\widetilde{\mathbf{B}}(\theta)$ where $\widetilde{\mathbf{B}}(\theta)$ is a base-circle composed of $\lfloor \pi/\phi^* \rfloor$ quasi-uniform arcs.

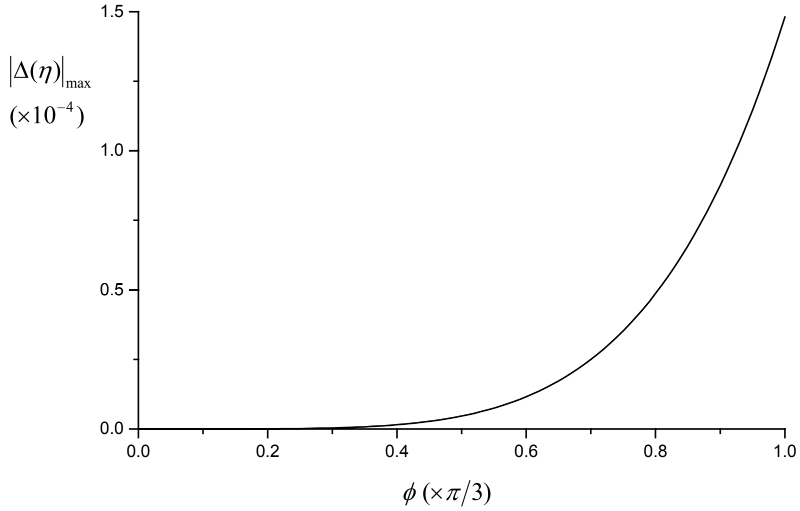


Figure 5.4. Maximum angle deviation vs. arc angle

Figure 5.5 gives an example of the construction of the 2D NURBS cam-profile.

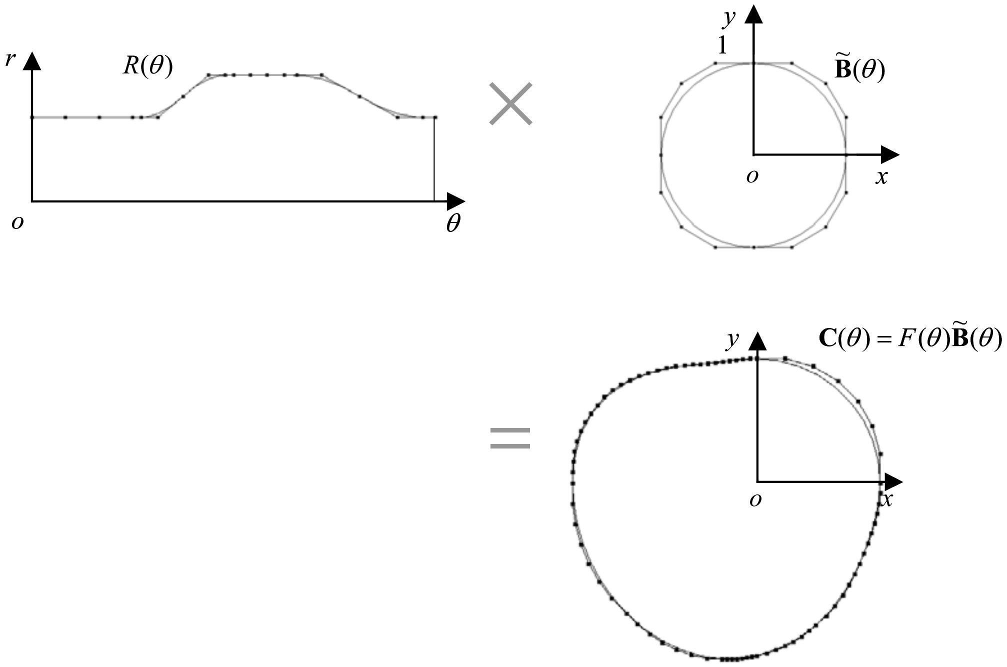

r
$R(\theta)$
o
θ
y
1
$\widetilde{\mathbf{B}}(\theta)$
o
x
y
$\mathbf{C}(\theta) = F(\theta)\widetilde{\mathbf{B}}(\theta)$
o
x

Figure 5.5. Construction of 2D NURBS cam-profile

5.2 Pressure-vessel Design

Almost pressure vessels are designed in the form of sphere- or cylinder-like “shell” shapes for smooth pressure distribution. Shown in Figure 5.6 is an axisymmetric-cylindrical pressure vessel.

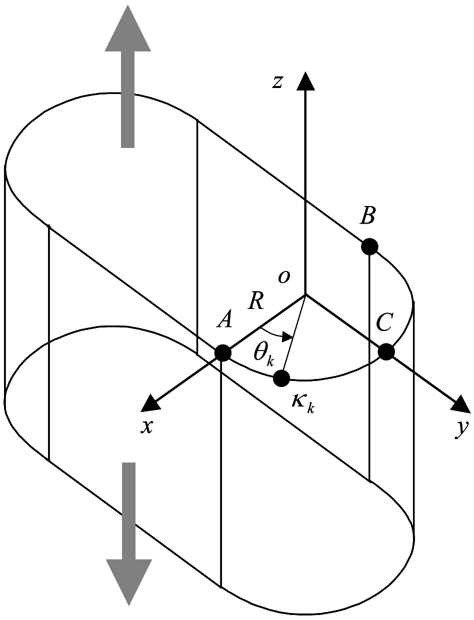


Figure 5.6. Axissymmetric-cylinder pressure vessel

To design the right (or left) side of wall is to find a single-valued function $R(\theta)$ from $\{(\kappa_k, \theta_k)\}$ where κ_k is curvature. For the curvature to be continuous at points A, B, following conditions are imposed when solving the nonlinear system [Fran1975][Ange1983].

$$R(0) = R\,;\quad R'(0) = 0\,;\quad R''(0) = 0 \tag{5.5a}$$

$$R(\pi) = R\,;\quad R'(\pi) = 0\,;\quad R''(\pi) = 0 \tag{5.5b}$$

Once $R(\theta)$ is obtained, the corresponding NURBS-curve (or surface) can be constructed by applying the conversion method suggested in Section 5.1.

5.3 Scientific Visualization

In much scientific and/or engineering research, spherical or cylindrical experimental set-ups are employed, and data in the form of single-valued functions ($\{(p_{k,l}, \phi_k, \theta_l)\}$ or $\{(p_{k,l}, \theta_k, z_l)\}$) are generated. These kinds of empirical data can be very effectively visualized by applying the proposed NURBS-surface fitting method. One example is shown in the Figure 5.7, which is "simulated" sound-pressure distribution caused by a noise source (monopole).

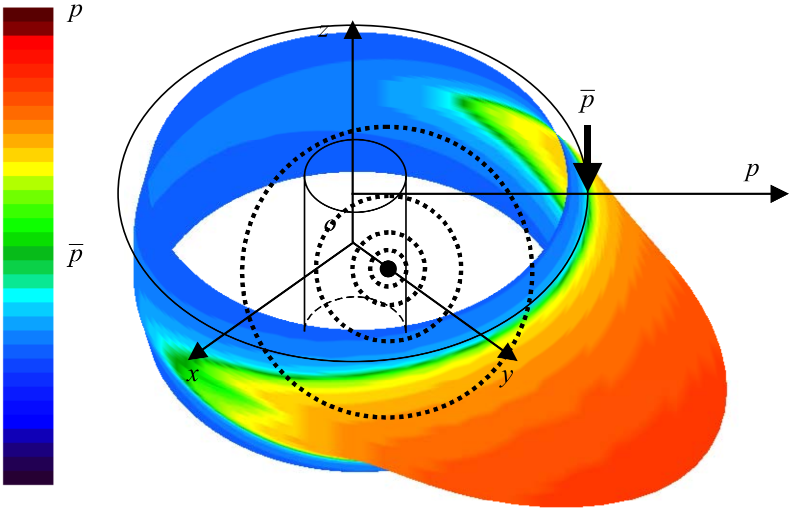


Figure 5.7. Sound-pressure distribution

The noise source is located at arbitrary position near the origin, and sound pressure is measured at $m \times n$ grid points on the cylindrical microphone array. The sound-pressure distribution can be shown in the form of a NURBS surface

$$\mathbf{S}(u,v) = \{s \cdot F(u,v) + d\}\,\mathbf{B}(u,v) \tag{5.6}$$

where $F(u,v)$ is a single-valued function fitted to $\{(p_{k,l}, \theta_k, z_l)\}$, s and d are scale factors for satisfactory display.

6. Discussions and Conclusions

The proposed NURBS-surface fitting method may be dubbed as an "indirect" method because it differs from the existing "direct" methods where a 3D B-spline or NURBS surface is constructed "directly" from cloud-of-points data (**COP-data**). Whereas "general" methods for fitting NURBS surface from COP-data in the literature – Hoschek [Hosc1993] employed a *circular spline* and Laurent-Gengoux and Mekhilef [Laur1993] treated NURBS-surface fitting as a non-linear optimization problem – focus on determining the weights of NURBS control-points so that the extra degrees-of-freedom provided by the weights are best utilized, in the proposed method first a single-valued B-spline surface is fitted to the data transformed into orthogonal- (spherical- or cylindrical-) coordinate system and second the fitted surface is converted to a NURBS surface by applying a symbolic-product operation with a NURBS base-geometry.

The proposed method provides a natural means to assign parameter values to COP-data representing a "rotational-freeform" shape, and may be embedded in a laser scanner having a spherical- or cylindrical-kinematics structure (because the laser scanner generates *range data* in its orthogonal-coordinate system). The proposed method enables to recover an "exact" NURBS surface for COP-data sampled from a "true" surface-of-revolution shape. In the proposed method the "real" fitting process is done in the form of a single-valued function, thus, the *point inversion* and approximation error calculation are trivial. The calculated approximation error is a kind of *parametric error* e_p, which has the relation with *geometric error* (or 3D Euclidean distance) e_g as given by $e_g \cong e_p \cos\psi$ where ψ is the angle between the surface-normal vector of the resulting NURBS surface and symbolic-product (or scaling) direction ($\overline{\mathbf{B}}(\phi,\theta)$ for a sphere and $\overline{\mathbf{B}}(\theta,0)$ for a cylinder). Because geometric error cannot be greater than parametric error, if the single-valued function is approximated within the pre-defined tolerance, the resulting NURBS-surface is also satisfied with the tolerance. However, it is a somewhat conservative manner. Thus, for the resulting NURBS-surface to be fitted tighter, it can be possible to directly use the "approximately" estimated geometric error ($e_g \cong e_p \cos\psi$) when the single-valued B-spline surface is fitted. On the other hand, because measured COP-data inevitably include noise, *faring* operation to the corresponding structured-data is essential before surface fitting. However, the repetitive application of *difference fairing* [Renz1982][Choi1998][Cho2001], which is one of the most widely used fairing methods, causes the shrinkage of the structured data. Thus, shrinkage avoidance is an important research issue related to fairing [Taub1995][Cho2000]. Whereas, if the fairing operation is performed in the orthogonal-coordinate system (before single-valued B-spline surface fitting), it does not cause the severe shrinkage of the resulting NURBS-surface unlike the fairing "directly" performed in 3D Cartesian-coordinate system. Besides, the proposed method allows the NURBS surface to be stored as a single-valued function and base geometry separately, which requires less memory and, more importantly, it allows an easier point-membership classification for NURBS-bounded solid objects. For the purpose of point-membership classification, a "general" B-spline surface representing a rotational-freeform shape can be "roughly" factorized into a single-valued B-spline surface and base geometry as follows: 1) a local-coordinate system $\omega = \{\mathbf{o}, \mathbf{i}, \mathbf{j}, \mathbf{k}\}$ for the B-spline surface ${}^{\Omega}\mathbf{S}(u,v)$ is estimated; 2) ${}^{\Omega}\mathbf{S}(u,v)$ is transformed into local-coordinate surface ${}^{\omega}\mathbf{S}(u,v)$; 3) a base geometry $\mathbf{B}(u,v)$ for ${}^{\omega}\mathbf{S}(u,v)$ is constructed; 4) a single-valued surface is extracted as given by $F(u,v) = {}^{\Omega}\mathbf{S}(u,v) \bullet \mathbf{B}(u,v)$. However, above "rough" factorization cannot be directly applied to "precise" surface operations such as *surface-surface intersection* (**SSI**), *trimming*, *etc.* because it does not consider any kind of parameter synchronization.

By offering a "robust" local-coordinate system estimation method for COP-data "arbitrarily" located in 3D Cartesian space, the proposed method was successfully embedded to a research-purposed RE S/W, as one component of the *surface fitting engine*. In consequence, surface modeling with the RE S/W is more efficient – the numbers of surfaces and secondary-surface operations are reduced, and user interaction is also reduced. Besides, the proposed method can be used for an "intuitive" scientific visualization of empirical data acquired from spherical or cylindrical experiment set-ups. The basic idea of the proposed method – a curve (or surface) can be interpreted as a product of a base geometry and a *scale function* – can be applied to some special shape (mechanical cam, pressure vessel, *etc.*) design.

A major drawback of the proposed NURBS-surface fitting method is that the degrees of the resulting NURBS-surface are somewhat high. For example, if a bi-cubic B-spline surface is used in the single-valued surface fitting stage, the degrees of the spherical NURBS-surface are (5,5).

要 約 文

NURBS Surface Fitting Using Orthogonal Coordinate Transform and Applications

실물모델로부터 CAD모델을 생성하는 작업은, 많은 제품개발과정에서 중요한 역할을 하고 있다. 본 논문에서는 회전자유형상의 측정데이터에 대하여 다음과 같은 과정을 가지는 "간접적인" NURBS곡면 생성방법을 제안한다: 1) 회전자유형상의 측정데이터에 대한 직교좌표 (구면좌표 또는 원주좌표) 변환, 2) 좌표변환된 데이터에 대한 일가함수 (B-스플라인 함수) 생성, 3) 생성된 일가함수와 2차 NURBS기저형상과의 곱연산을 통한 3D NURBS곡면 생성. 기존의 "직접적인" 곡면생성 방법에 비하여 본 논문에서 제안하는 방법은 다음과 같은 특징적인 장점을 가지고 있다: 1) 회전자유형상의 측정데이터에 대한 자연스러운 매개변수화 제공, 2) 실제 회전형상으로부터 측정된 데이터의 경우 정확한 회전형상을 나타내는 NURBS곡면 생성, 3) NURBS곡면으로 구성된 입체에 대한 점구분 용이.

제안하는 NURBS곡면 생성방법은, 임의의 3D 공간상의 회전자유형상 측정데이터에 대한 국소좌표계 정의 방법 (2차곡면 근사에 근거) 과 결합하여, 역공학소프트웨어의 곡면생성엔진의 한 부분으로 성공적으로 구현되었으며, 그 결과, 회전자유형상을 포함하는 형상에 대하여, 곡면 개수 및 이차곡면연산작업 감소, 사용자 개입의 감소 등을 통하여 보다 효율적인 역공학프로세스를 기대할 수 있게 되었다. 또한, 제안하는 방법은 구면 또는 원주 형상의 실험장치를 이용하여 측정한 실험데이터의 효과적인 시각화에 적합하다. 한편, 제안하는 방법의 기본개념 즉, 회전자유곡선 (또는 곡면) 은 기저형상과 스케일 함수로 해석 가능하다는 사실은, 캠 또는 압력용기 등의 특별한 형상설계에도 이용될 수 있다.

References

[Amen1998] Amenta, N., Bern, M., and Kamvysselis, M., “ A new voronoi-based surface reconstruction algorithm,” *The Proceedings of ACM SIGGRAPH 1995*, pp. 415-422

[Ange1983] Angeles, J., “Synthesis of plane curves with prescribed local geometric properties using periodic splines,” *Computer-Aided Design*, Vol. 15, No. 3, 1983, pp.147-155

[Baja1995] Bajaj, C., Bernardini, F., and Xu, G., “Automatic reconstruction of surfaces and scalar fields from 3D scans,” *The Proceedings of ACM SIGGRAPH 1995*, pp. 109-118

[Bern2000] Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G., “The ball-pivoting algorithm for surface reconstruction,” *IEEE Transactions on Visualization and Computer Graphics*, Vol.5, No.4, 2000, pp. 349-359

[Besl1988] Besl, P. J. and Jain, R. C., "Segmentation through variable-order surface fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 3, 1988, pp. 167-192

[Besl1992] Besl, P. J. and McKay, N. D., “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, 1992, pp. 239-256

[Blan1996] Blanc, C. and Schlick, C., “Accurate parameterization of conics by NURBS,” *IEEE Computer Graphics and Applications*, Vol. 16, No. 6, 1996, pp. 64-71

[Boeh1984] Boehm, W., Farin, G. and Kahmann, J., “A survey of curve and surface methods in CAGD,” *Computer Aided Geometric Design*, Vol. 1, 1984, pp. 1-60

[Chen1992] Chen, Y. and Medioni, G., “Object modelling by registration of multiple range images,” *Image and Vision Computing*, Vol. 10, No. 3, 1992, pp. 145-155

[Cho2000] Cho, S. K., *Point-set data fairing using difference operator*, PhD Dissertation, KAIST, 2000

[Cho2001] Cho, S. K. and Choi, B. K., “Analysis of difference fairing based on DFT-filter,” *Computer-Aided Design*, Vol. 33, No. 1, 2001, pp. 45-56

[Choi1991] Choi, B. K., *Surface Modeling for CAD/CAM*, Elsevier, 1991

[Choi1996] Choi, B. K. and Jerard, R. B., *Sculptured Surface Machining: Theory and Practice*, Kluwer Academic Publisher, 1998

[Curl1996] Curless, B. and Levoy, M., "A volumetric method for building complex models from range images," *The Proceedings of ACM SIGGRAPH 1996*, pp. 303-312

[CySl] http://www.cyberware.com/products/cyslice.html

[deBo1978] de Boor, C., *A Practical Guide to Splines*, Springer-Verlag, 1978

[deCa1993] de Casteljau, P., "Polar forms for curve and surface modeling as used at Citroem," in *Fundamental Developments of Computer Aided Geometric Modeling*, Edited by Piegl, L., Academic Press, 1993, pp. 1-12

[doCa1976] do Camo, M. P., *Differential Geometry of Curves and Surfaces*, Prentice-Hall, 1976

[Eck1996] Eck, M. and Hoppe, H., "Automatic reconstruction of B-spline surfaces of arbitrary topological type," *The Proceedings of ACM SIGGRAPH 1996*, pp.325-334

[Elbe1993] Elber, G., and Cohen, E., "Second-order surface analysis using hybrid symbolic and numeric operations," *ACM Transactions on Graphics*, Vol. 12, No. 2, 1993, pp.160-178

[Elsa1996] Elsasser, B. and Hoscheck, J., "Approximation of digitized points by surfaces of revolution," *Computers & Graphics*, Vol. 20, No. 1, 1996, pp. 85-94

[Fari1988] Farin, G., *Curves and Surfaces for Computer Aided Geometric Design: A Practice Guide*, Academic Press, 1988

[Faro1988] Farouki, R. T. and Rajan, V. T., "Algorithms for polynomials in Bernstein form," *Computer Aided Geometric Design*, Vol. 3, 1988, pp. 15-43

[Faro1991] Farouki, R. T. and Sakkalis, T., "Real rational curves are not 'unit speed'," *Computer Aided Geometric Design*, Vol. 8, 1991, pp. 151-158

[Faug1986] Faugeras, O. D. and Hebert, M., "The representation, recognition and locating of 3-D objects," *International Journal of Robotics Research*, Vol. 5, No. 3, 1986, pp. 27-52

[Faux1981] Faux, I. D. and Pratt, M. J., *Computational Geometry for Design and Manufacture*, Ellis Horwood, 1981

[Fran1975] Fracavilla, A., Ramakrishnan, C. V. and Zienkiewicz, O., "Optimization of shape to minimize stress concentration," *The Journal of Strain Analysis*, Vol. 10, No. 2, 1975, pp. 63-70

[Gosh1997] Goshtasby, A., Nambala, S., DeRijk, W. and Campbell, S., "A range scanner for digital reconstruction for stone dental casts," *IEEE Transactions of Medical Imaging*, Vol. 16, No. 5, 1997, pp. 664-674

[Gott1996] Gottschalk, S., Lin, M. C. and Manocha, D., "OBBTree: a hierarchical structure for rapid interference detection," *The Proceedings of ACM SIGGRAPH 1996*, pp. 171-180

[Hopp1992] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W., "Surface reconstruction from unorganized points," *The Proceedings of ACM SIGGRAPH 1992*, pp. 71-78

[Hosc1992] Hoschek, J., "Circular splines," *Computer-Aided Design*, Vol. 24, No. 11, 1992, pp. 611-618

[Hosc1993] Hoschek, J. and Lasser, D., *Fundamentals of Computer Aided Geometric Design*, A K Peters, 1993

[ICEM] http://www.ptc.com/icem/

[Imag] http://www.plmsol-eds.com/ideas/imageware.dshtml

[Jain1989] Jain, A. K., *Fundamentals of Digital Image Processing*, Prentice Hall, 1989

[Kjel1983] Kjellander, J. A. P., "Smoothing of cubic parametric splines," *Computer-Aided Design*, Vol. 15, No. 3, 1983, pp. 175-179

[Kris1996] Krishnamurthy, V. and Levoy, M., "Fitting smooth surfaces to dense polygon meshes," *The Proceedings of ACM SIGGRAPH 1996*, pp. 313-324

[Lai2000] Lai, J. Y. and Ueng, W. D., "Reconstruction of surfaces of revolution from measured points," *Computers in Industry*, Vol. 41, 2000, pp. 147-161

[Laur1993] Laurent-Gengoux, P. and Mekhilef, M., "Optimization of a NURBS representation," *Computer-Aided Design*, Vol. 25, No. 11, 1993, pp. 699-710

[Lay1996] Lay, D. C., *Linear Algebra and Its Applications*, Addison-Wesley, 1996

[Levi1976] Levin, J., "A parametric algorithm for drawing pictures of solid objects composed of quadric surfaces," *Communications of the ACM*, Vol. 19, No. 10, 1976, pp. 555-563

[Luka1997] Lukacs, G., Marshall, A. D. and Martin, R. R., "Geometric least-squares fitting of spheres, cylinders, cones and tori," in: Martin, R. R. and Varady, T. (Ed.), *RECCAD Deliverable Documents 2 and 3 Copernicus Project No. 1068: Reports on basic geometry and geometric model creation, etc.*, Report GML 1997/5, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1997

[Ma1995] Ma, W. and Kruth, J. P., "Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces," *Computer-Aided Design*, Vol. 27, No. 9, 1995, pp. 663-675

[MacC1988] MacCarthy, B. L., "Quintic splines for kinematic design," *Computer-Aided Design*, Vol. 20, No. 7, 1988, pp. 406-415

[Mars1992] Marshall, A. D. and Martin, R. R., *Computer Vision, Models and Inspection*, World Scientific, 1992

[Meie1987] Meier, H. and Nowacki, H., "Interpolating curves with gradual changes in curvature," *Computer Aided Geometric Design*, No. 4, 1987, pp. 297-305

[Milo1997] Milroy, M. J., Bradley, C. and Vickers, G. W., "Segmentation of a wrap-around model using an active contour," *Computer-Aided Design*, Vol. 29, No. 4, 1997, pp. 299-320

[Niel1993] Nielson, G., "CAGD's top ten: what to watch," *IEEE Computer Graphics & Application*, Vol. 13, No. 1, 1993, pp.35-37

[Park1996] Park, H. and Kim, K., "Smooth surface approximation to serial cross-sections," *Computer-Aided Design*, Vol. 28, No. 12, 1996, pp.995-1005

[Pett1970] Pettofrezzo, A. J. and Lacatena, M. M., *Analytic Geometry with Vectors*, Scott-Foresman, 1970

[Pieg1987] Piegl, L. and Tiller, W., “Curve and surface constructions using rational B-splines,” *Computer-Aided Design*, Vol. 19, No. 9, 1987, pp. 485-498

[Pieg1991] Piegl, L., “On NURBS: A survey,” *IEEE Computer Graphics and Applications*, Vol. 10, No. 1, 1991, pp. 55-71

[Pieg1995] Piegl, L. and Tiller, W., *The NURBS Book*, Springer-Verlag, 1995

[Pieg1997] Piegl, L. and Tiller, W., “Symbolic operators for NURBS,” *Computer-Aided Design*, Vol. 29, No. 5, 1997, pp. 361-368

[Pott1998a] Pottman, F. and Randrup, T., “Rotational and helical surface approximation for reverse engineering,” *Computing*, Vol. 60, 1998, pp. 307-322

[Pott1998b] Pottman, F., Chen, H. Y. and Lee, I. K., “Approximation by profile surface,” in *The Mathematics of Surfaces VIII*, Edited by Cripps, R., Information Geometers, 1998, pp. 17-36

[Prat1991] Pratt, W. K., *Digitial Image Processing 2nd Edition*, John Wiley & Sons, 1991

[Pres1994] Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P., *Numerical Recipes in C: The Art of Scientific Computing 2nd Edition*, Cambridge University Press, 1994

[Rapi] http://www.rapidform.com/English/Home.htm

[Renz1982] Renz, W., “Interactive smoothing of digitized point data,” *Computer-Aided Design*, Vol. 14, No. 5, 1982, pp.267-269

[Rhin] http://www.rhino3d.com

[Ries1973] Riesenfeld, R. F., *Applications of B-spline Approximation to Geometric Problems of Computer-aided Design*, Ph.D Dissertation, Syracuse University, 1973

[Roth1956] Rothbart, H. A., *Cams-Design, Dynamics, and Accuracy*, John Wiley & Sons, 1956

[Sanc1990] Sanchez-Reyes, J., “Single-valued curves in polar coordinates,” *Computer-Aided Design*, Vol. 22, No. 1, 1990, pp. 19-26

[Sanc1991] Sanchez-Reyes, J., “Single-valued surfaces in cylindrical coordinates,” *Computer-Aided Design*, Vol. 23, No. 8, 1991, pp. 561-568

[Sanc1992] Sanchez-Reyes, J., “Single-valued spline curves in polar coordinates,” *Computer-Aided Design*, Vol. 24, No. 6, 1992, pp. 307-315

[Sanc1994] Sanchez-Reyes, J., “Single-valued surfaces in spherical coordinates,” *Computer Aided Geometric Design*, Vol. 11, 1994, pp. 491-517

[Sapi1995] Sapidis, N. S. and Besl, P. J., “Direct construction of polynomial surfaces from dense range images through region growing,” *ACM Transactions on Graphics*, Vol. 14, No. 2, 1995, pp. 171-200

[Taub1995] Taubin, G., “Curve and surface smoothing without shrinkage,” *The proceedings of Fifth International Conference on Computer Vision*, 1995, pp. 852-857

[Till1983] Tiller, W., “Rational B-splines for curve and surface representation,” *IEEE Computer Graphics and Applications*, Vol. 3, No. 6, 1983, pp. 61-69

[Turk1994] Turk, G. and Levoy, M., “Zippered polygon meshes from range images,” *The Proceedings of ACM SIGGRAPH 1994*, pp. 311-318

[Vara1997] Varady, T., Martin, R. R. and Cox, J., “Reverse engineering of geometric models–an introduction,” *Computer-Aided Design*, Vol. 29, No. 4, 1997, pp. 255-268

[Vers1975] Versprille, K., *Computer Aided Design Applications of the Rational B-spline Approximation Form*, PhD Dissertation, University of Syracuse, 1975

[Wolt1992] Wolter, F. E. and Tuohy, S. T., “Curvature computations for degenerate surface patches,” *Computer Aided Geometric Design*, No. 9, 1992, pp. 241-270

[Yama1993] Yamada, Y., *Clay Modeling: Techniques for Giving Three-dimensional Form to Idea*, *Car Styling Extra Issues*, Vol. 93 1/2, 1993

[Zhan1994] Zhang, Z., “Iterative point matching for registration of free-form curves and surfaces,” *International Journal of Computer Vision*, Vol. 13, No. 2, 1994, pp. 119-152

[Zwil1996] Zwillinger, D. and Kellogg, B., *CRC Standard Mathematical Tables and Formulae 30th Edition*, CRC Press, 1996

[3dsM] http://www.discreet.com

Appendix A. NURBS-curve Construction Using Symbolic-product Operation

A symbolic-product operation of a B-spline function and NURBS curve, is more efficient when the NURBS curve is represented using homogeneous coordinates [Pieg1997]. Thus, to construct a parametric NURBS-curve from $\mathbf{C}(u)=F(u)\mathbf{B}(u)$, let the NURBS base-circle $\mathbf{B}(u)=(X(u),Y(u))$ be written in the form of a homogeneous representation as

$$\mathbf{B}^w(u)=(X^w(u),Y^w(u),W(u)) \tag{A.1}$$

where $X^w(u)=\sum_{i=0}^{n_B} N_{i,2}(u)\,x_i^w$, $Y^w(u)=\sum_{i=0}^{n_B} N_{i,2}(u)\,y_i^w$, $W(u)=\sum_{i=0}^{n_B} N_{i,2}(u)\,w_i$; $x_i^w=w_i x_i$, $y_i^w=w_i y_i$. Then, the homogeneous form of the resulting NURBS-curve can be obtained by applying the symbolic-product operations of a single-valued B-spline function $F(u)$ with the first two (non-rational) B-spline functions as

$$\mathbf{C}^w(u)=(F(u)X^w(u),F(u)Y^w(u),W(u))\,. \tag{A.2}$$

For the details of symbolic-product operation of two non-rational B-spline functions, see Appendix A.1. By applying a perspective mapping of $\mathbf{C}^w(u)$ with respect to the origin, the resulting NURBS-curve is obtained in turn:

$$\begin{aligned}\mathbf{C}(u)&=H\{\mathbf{C}^w(u)\}=H\{(F(u)X^w(u),F(u)Y^w(u),W(u))\}\\&=\left(\frac{F(u)X^w(u)}{W(u)},\frac{F(u)Y^w(u)}{W(u)}\right)=(F(u)X(u),F(u)Y(u))\\&=F(u)\mathbf{B}(u)\end{aligned} \tag{A.3}$$

Appendix A.1 Symbolic-product Operation of Two Non-rational B-spline Functions

Given two non-rational B-spline functions

$$\begin{aligned} F(u) &= \sum_{i=0}^{n_F} N_{i,p}(u)\, f_i; \\ U_F &= \{\underbrace{u_s,\ldots,u_s}_{p+1}, u_{p+1},\ldots,u_{n_F}, \underbrace{u_e,\ldots,u_e}_{p+1}\} \\ &= \{\underbrace{u_s,\ldots,u_s}_{p+1}, \underbrace{u_1,\ldots,u_1}_{m_F^1},\ldots,\underbrace{u_i,\ldots,u_i}_{m_F^i},\ldots,\underbrace{u_e,\ldots,u_e}_{p+1}\} \end{aligned} \tag{A.4}$$

$$\begin{aligned} G(u) &= \sum_{j=0}^{n_G} N_{j,q}(u)\, g_j; \\ U_G &= \{\underbrace{v_s,\ldots,v_s}_{q+1}, v_{q+1},\ldots,v_{n_G}, \underbrace{v_e,\ldots,v_e}_{q+1}\} \\ &= \{\underbrace{v_s,\ldots,v_s}_{q+1}, \underbrace{v_1,\ldots,v_1}_{m_G^1},\ldots,\underbrace{v_j,\ldots,v_j}_{m_G^j},\ldots,\underbrace{v_e,\ldots,v_e}_{q+1}\} \end{aligned} \tag{A.5}$$

of degree p and q, respectively, the symbolic product of them is also a non-rational B-spline function

$$H(u) = \sum_{k=0}^{n_H} N_{k,p+q}(u)\, h_k \tag{A.6}$$

of degree $p+q$ over the knot vector

$$\begin{aligned} U_H &= \{\underbrace{w_s,\ldots,w_s}_{p+q+1}, w_{p+q+1},\ldots,w_{n_H}, \underbrace{w_e,\ldots,w_e}_{p+q+1}\} \\ &= \hat{U}_F \cup \hat{U}_G \end{aligned} \tag{A.7}$$

where

$$\hat{U}_F = \{\underbrace{u_s,\ldots,u_s}_{(p+1)+q}, \underbrace{u_1,\ldots,u_1}_{m_F^1+q},\ldots,\underbrace{u_i,\ldots,u_i}_{m_F^i+q},\ldots,\underbrace{u_e,\ldots,u_e}_{(p+1)+q}\},$$

$$\hat{U}_G = \{\underbrace{v_s,\ldots,v_s}_{(q+1)+p}, \underbrace{v_1,\ldots,v_1}_{m_G^1+p},\ldots,\underbrace{v_j,\ldots,v_j}_{m_G^j+p},\ldots,\underbrace{v_e,\ldots,v_e}_{(q+1)+p}\}.$$

The procedure of determining control points $\{h_k\}$ is as follows:

1) Decompose $F(u)$ and $G(u)$ into Bezier functions $\{F^b(u)\}$ and $\{G^b(u)\}$, respectively, using knot refinements with the distinct knot vector of U_H;

2) Compute the product of each paired Bezier functions (See Appendix A.2);

3) Recompose the Bezier product functions into a B-spline form using a full knot vector of for degree $p+q$, namely,

$$\hat{U}_H = \{\underbrace{w_s,\ldots,w_s}_{p+q+1},\underbrace{w_1,\ldots,w_1}_{p+q},\ldots,\underbrace{w_k,\ldots,w_k}_{p+q},\ldots,\underbrace{w_e,\ldots,w_e}_{p+q+1}\}; \tag{A.8}$$

4) Remove all knots of $U_H^C = \hat{U}_H - U_H$.

Above procedure to find U_H and $\{h_k\}$, is conceptually the same as Piegl and Tiller's [Pieg1997], which is summarized more intuitively.

Appendix A.2 Symbolic-product Operation of Two Bezier Functions

Given two Bezier functions

$$F(u) = \sum_{i=0}^{p} B_{i,p}(u)\, f_i \tag{A.9}$$

$$G(u) = \sum_{j=0}^{q} B_{j,q}(u)\, g_j \tag{A.10}$$

of degree p and q, respectively, the Bezier product function is given by [Faro1988][Elbe1993][Pieg1997]

$$H(u) = \sum_{k=0}^{p+q} B_{k,p+q}(u)\, h_k \tag{A.11}$$

where

$$h_k = \sum_{l=\max(0,k-q)}^{\min(p,k)} \frac{\binom{p}{l}\binom{q}{k-l}}{\binom{p+q}{k}} f_l g_{k-l}; \quad k = 0,\ldots,p+q.$$

Appendix B. NURBS-surface Construction Using Symbolic-Product Operation

As similar to the curve case given in Appendix A, a symbolic-product operation of a single-valued B-spline function $F(u,v)$ and NURBS base-geometry $\mathbf{B}(u,v)=(X(u,v),Y(u,v),Z(u,v))$, is more efficient when the NURBS base-geometry (*i.e.*, base sphere or cylinder) is in the form of a homogeneous representation,

$$\mathbf{B}^{w}(u,v)=(X^{w}(u,v),Y^{w}(u,v),Z^{w}(u,v),W(u,v)) \tag{B.1}$$

where $X^{w}(u,v)=\sum_{i=0}^{n_B}\sum_{j=0}^{m_B}N_{i,2}(u)N_{j,2}(v)\,x_{i,j}^{w}$, $Y^{w}(u,v)=\sum_{i=0}^{n_B}\sum_{j=0}^{m_B}N_{i,2}(u)N_{j,2}(v)\,y_{i,j}^{w}$, $Z^{w}(u,v)=\sum_{i=0}^{n_B}\sum_{j=0}^{m_B}N_{i,2}(u)N_{j,2}(v)\,z_{i,j}^{w}$, $W(u,v)=\sum_{i=0}^{n_B}\sum_{j=0}^{m_B}N_{i,2}(u)N_{j,2}(v)\,w_{i,j}$; $x_{i,j}^{w}=w_{i,j}x_{i,j}$, $y_{i,j}^{w}=w_{i,j}y_{i,j}$, $z_{i,j}^{w}=w_{i,j}z_{i,j}$ for a spherical NURBS-surface fitting. The component functions for a cylindrical NURBS-surface fitting, can be written in the similar form where $\{N_{i,1}(u)\}$ are substituted for all $\{N_{i,2}(u)\}$.

The homogeneous form of the resulting NURBS-surface is given by

$$\mathbf{S}^{w}(u,v)=(F(u,v)X^{w}(u,v),F(u,v)Y^{w}(u,v),F(u,v)Z^{w}(u,v),W(u,v)) \tag{B.2}$$

for the spherical NURBS-surface fitting, or

$$\mathbf{S}^{w}(u,v)=(F(u,v)X^{w}(u,v),F(u,v)Y^{w}(u,v),Z^{w}(u,v),W(u,v)) \tag{B.3}$$

for the cylindrical NURBS-surface fitting.

A symbolic-product operation of two non-rational "bi-variate" B-spline functions (let us denote $F(u,v)$ and $G(u,v)$) can be performed by a simple extended procedure of the curve case given in Appendix A.1 as follows: 1) determine u- and v-directional knot vectors (U_H and V_H) for the resulting non-rational bi-variate B-spline function $H(u,v)$, respectively by Equations (A.4)-(A.7); 2) decompose $F(u,v)$ and $G(u,v)$ into bi-variate Bezier functions $\{F^{b}(u,v)\}$ and $\{G^{b}(u,v)\}$ using knot refinements with the distinct knot vectors of U_H and V_H; 3) compute the product of the each paired Bezier functions (See Appendix B.1); 4) recompose the Bezier-product functions into a B-spline form using knot removal with U_H and V_H. The degrees of the resulting NURBS-surface are $(p+2,q+2)$ for the spherical NURBS-surface fitting and $(\max\{p,1\},q+2)$ for the cylindrical NURBS-surface fitting (See Appendix B.2).

Appendix B.1 Symbolic-product Operation of Two Bi-variate Bezier Functions

Given two bi-variate Bezier functions

$$F(u,v)=\sum_{i=0}^{p_F}\sum_{j=0}^{q_F}B_{i,p_F}(u)B_{i,q_F}(v)\,f_{i,j} \tag{B.4}$$

$$G(u,v)=\sum_{i=0}^{p_G}\sum_{j=0}^{q_G}B_{i,p_G}(u)B_{j,q_G}(v)\,g_{i,j} \tag{B.5}$$

of degree (p_F, q_F) and (p_G, q_G), respectively, the bi-variate Bezier-product function is given by [Faro1988][Elbe1993][Pieg1997]

$$H(u,v)=\sum_{k=0}^{p_F+p_G}\sum_{l=0}^{q_F+q_G}B_{k,p_F+p_G}(u)B_{l,q_F+q_G}(v)\,h_{k,l} \tag{B.6}$$

where

$$h_{k,l}=\sum_{i=\max(0,k-p_G)}^{\min(p_F,k)}\ \sum_{j=\max(0,l-q_G)}^{\min(q_F,l)}\frac{\binom{p_F}{i}\binom{p_G}{k-i}\binom{q_F}{j}\binom{q_G}{l-j}}{\binom{p_F+p_G}{k}\binom{q_F+q_G}{l}}f_{i,j}g_{k-i,l-j};$$

$$k=0,\ldots,p_F+p_G,\quad l=0,\ldots,q_F+q_G.$$

Appendix B.2 Degrees of NURBS Surfaces

First, let us consider a spherical NURBS-surface fitting. A base sphere can be represented in a homogeneous form as Equation (B.1), where the degrees of the first three component functions, $X^w(u,v)$, $Y^w(u,v)$, $Z^w(u,v)$, are $(0,2)$, $(0,2)$, $(2,0)$, respectively (because the base sphere is constructed by a surface-of-revolution method with respect to *z*-axis). Thus, the degrees of the first three product-component functions in Equation (B.2), $F(u,v)X^w(u,v)$, $F(u,v)Y^w(u,v)$, $F(u,v)Z^w(u,v)$, can be easily calculated as $(p,q+2)$, $(p,q+2)$, $(p+2,q)$, respectively. Then, the degrees of the resulting NURBS surface are

$$(\max\{p,p,p+2\},\max\{q+2,q+2,q\})=(p+2,q+2)\,. \tag{B.7}$$

On the other hand, the degrees of first three component functions in Equation (B.1) for a base cylinder, are $(0,2)$, $(0,2)$, $(1,0)$, respectively, and the degrees of the first three functions in Equation

(B.3), $F(u,v)X^w(u,v)$, $F(u,v)Y^w(u,v)$, $Z^w(u,v)$, are $(p,q+2)$, $(p,q+2)$, $(1,0)$, respectively. Therefore, the degrees of the resulting cylindrical NURBS-surface are given by:

$$(\max\{p,p,1\},\max\{q+2,q+2,0\}) = (\max\{p,1\},q+2) \tag{B.8}$$

Appendix C. Constrained-surface Fitting

Assume that $\{G_k\} = \{F(u_k, v_k)\}$; $k = 0, \ldots, K$ are given. We are to seek a single-valued B-spline surface

$$F(u,v) = \sum_{i=0}^{n_F} \sum_{j=0}^{m_F} N_{i,p}(u) N_{j,q}(v) f_{i,j} \tag{C.1}$$

of degrees (p,q) over knot vectors U_F and V_F under the constraints

$$H_l = \sum_{i=0}^{n_F} \sum_{j=0}^{m_F} c_{i,j}^l f_{i,j} \; ; \; l = 0, \ldots, L \, . \tag{C.2}$$

The unconstrained equations are written in a matrix form as

$$\mathbf{G} = \mathbf{N}\,\mathbf{F} \tag{C.3}$$

and constrained equations as

$$\mathbf{H} = \mathbf{C}\,\mathbf{F} \tag{C.4}$$

where

$$\mathbf{N} = \begin{bmatrix} N_{0,p}(u_0)N_{0,q}(v_0) & \cdots & N_{0,p}(u_0)N_{m_F,q}(v_0) & \cdots & N_{n_F,p}(u_0)N_{0,q}(v_0) & \cdots & N_{n_F,p}(u_0)N_{m_F,q}(v_0) \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ N_{0,p}(u_K)N_{0,q}(v_K) & \cdots & N_{0,p}(u_K)N_{m_F,q}(v_K) & \cdots & N_{n_F,p}(u_K)N_{0,q}(v_K) & \cdots & N_{n_F,p}(u_K)N_{m_F,q}(v_K) \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} c_{0,0}^0 & \cdots & c_{0,m_F}^0 & \cdots & c_{n_F,0}^0 & \cdots & c_{n_F,m_F}^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{0,0}^L & \cdots & c_{0,m_F}^L & \cdots & c_{n_F,0}^L & \cdots & c_{n_F,m_F}^L \end{bmatrix},$$

$$\mathbf{F} = \begin{bmatrix} f_{0,0} \\ \vdots \\ f_{0,m_F} \\ \vdots \\ f_{m_F,0} \\ \vdots \\ f_{n_F,m_F} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} G_0 \\ \vdots \\ G_K \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} h_0 \\ \vdots \\ h_L \end{bmatrix}.$$

If the sum of the numbers of above equations is the same as the numbers of degrees-of-freedom of $F(u,v)$, that is, $(K+1)+(L+1)=(n_F+1)(m_F+1)$, the given problem becomes a general interpolation problem as

$$\tilde{\mathbf{G}} = \tilde{\mathbf{N}}\,\mathbf{F} \tag{C.5}$$

where

$$\tilde{\mathbf{G}} = \begin{bmatrix} \mathbf{G} \\ \mathbf{H} \end{bmatrix}, \quad \tilde{\mathbf{N}} = \begin{bmatrix} \mathbf{N} \\ \mathbf{C} \end{bmatrix}.$$

If $\tilde{\mathbf{N}}$ is invertible, there is a unique solution as given by

$$\mathbf{F} = \tilde{\mathbf{N}}^{-1}\tilde{\mathbf{G}} \tag{C.6}$$

On the other hand, if $(K+1)+(L+1)>(n_F+1)(m_F+1)$, the given problem becomes a constrained least-squares approximation to minimize the sum of squares of errors subject to constraints

$$(\mathbf{G}-\mathbf{N}\,\mathbf{F})^T(\mathbf{G}-\mathbf{N}\,\mathbf{F})+\mathbf{L}^T(\mathbf{C}\,\mathbf{F}-\mathbf{H}) \tag{C.7}$$

where $\mathbf{L}$ is a column vector of *Lagrange multipliers*. Differentiating Equation (C.7) with respect to unknowns $\mathbf{F}$ and $\mathbf{L}$, and setting zero yield two matrix equations

$$\mathbf{N}^T\mathbf{N}\,\mathbf{F}+\mathbf{C}^T\mathbf{L}=\mathbf{N}^T\mathbf{G} \tag{C.8a}$$

$$\mathbf{C}\,\mathbf{F}=\mathbf{H} \tag{C.8b}$$

By substituting $\mathbf{F}=(\mathbf{N}^T\mathbf{N})^{-1}(\mathbf{N}^T\mathbf{G}-\mathbf{C}^T\mathbf{L})$ (from Equation C.8a) into Equation (C.8b) if $\mathbf{N}^T\mathbf{N}$ is invertible, $\mathbf{L}$ can be obtained if $\mathbf{C}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{C}^T$ is also invertible as follows:

$$\mathbf{L}=\{\mathbf{C}\,(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{C}^T\}^{-1}\{\mathbf{C}\,(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{G}-\mathbf{H}\} \tag{C.9}$$

Finally, $\mathbf{F}$ can be calculated by substituting Equation (C.9) into Equation (C.8a) as

$$\mathbf{F}=(\mathbf{N}^T\mathbf{N})^{-1}[\mathbf{N}^T\mathbf{G}-\mathbf{C}^T\{\mathbf{C}\,(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{C}^T\}^{-1}\{\mathbf{C}\,(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{G}-\mathbf{H}\}]. \tag{C.10}$$

Appendix D. G-continuity Conditions at Degenerate Point

Appendix D.1 Proof of Simple G^1 Continuity Condition

With Equation (3.14a), the boundary values of the single-valued B-spline surface at the pole yield,

$$F(u_s, v) = \sum_{j=0}^{m_F} N_{j,q}(v) f_{0.j} = \sum_{j=0}^{m_F} N_{j,q}(v) f_{0,0} = f_{0,0} \sum_{j=0}^{m_F} N_{j,q}(v) = f_{0,0} \,. \tag{D.1}$$

Then, the G^0 continuity condition of Equation (3.11) becomes satisfied:

$$\mathbf{S}(u_s, v) = F(u_s, v)\mathbf{B}(u_s, v) = f_{0,0}\mathbf{k} \tag{D.2}$$

Another constraints given by Equation (3.14b) yields,

$$F_u(u_s, v) = \sum_{j=0}^{m_F} \frac{p\, N_{j,q}(v)}{u_{p+1} - u_1}(f_{1,j} - f_{0,j}) = 0 \tag{D.3a}$$

$$F_{uv}(u_s, v) = \sum_{j=0}^{m_F - 1} \frac{pq\, N_{j,q-1}(v)}{(u_{p+1} - u_1)(v_{j+q+1} - v_{j+1})}(f_{1,j+1} - f_{0,j+1} - f_{1,j} + f_{0,j}) = 0 \tag{D.3b}$$

Then,

$$\mathbf{S}_u(u_s, v) \times \mathbf{S}_{uv}(u_s, v) = \{F(u_s, v)\}^2 \{\mathbf{B}_u(u_s, v) \times \mathbf{B}_{uv}(u_s, v)\} \tag{D.4}$$

where $\mathbf{B}_u(u_s, v) \times \mathbf{B}_{uv}(u_s, v)$ is in the direction of $\mathbf{k}$, which makes the condition of Equation (3.12) for the G^1 continuity be satisfied.

Appendix D.2 Optimal Scale Factors to Find Degenerate Point and Surface-normal Vector

The derivatives of a p^{th} degree Bezier curve $\mathbf{C}(t) = \sum_{i=0}^{p} B_{i,p}(t)\mathbf{c}_i$ at endpoints are easily obtained as:

$$\mathbf{C}(0) = \mathbf{c}_0, \mathbf{C}'(0) = p(\mathbf{c}_1 - \mathbf{c}_0), \mathbf{C}''(0) = p(p-1)(\mathbf{c}_2 - 2\mathbf{c}_1 + \mathbf{c}_0) \tag{D.5a}$$

$$\mathbf{C}(1) = \mathbf{c}_p, \mathbf{C}'(1) = p(\mathbf{c}_p - \mathbf{c}_{p-1}), \mathbf{C}''(1) = p(p-1)(\mathbf{c}_p - 2\mathbf{c}_{p-1} + \mathbf{c}_{p-2}) \tag{D.5b}$$

Simple arithmetic manipulations with using Equations (3.17) and (D.5) give the control points of connecting curves $\mathbf{C}_j(t)=\sum_{i=0}^{5} B_{i,5}(t)\,\mathbf{c}_i^j$ in terms of α, β as follows:

$$\begin{aligned}
\mathbf{c}_0^j &= \mathbf{S}(u_s,\upsilon_j)\\
\mathbf{c}_1^j &= \mathbf{S}(u_s,\upsilon_j)+(1/5)\,\alpha\,\mathbf{S}_u(u_s,\upsilon_j)\\
\mathbf{c}_2^j &= \mathbf{S}(u_s,\upsilon_j)+(2/5)\,\alpha\,\mathbf{S}_u(u_s,\upsilon_j)+(1/20)\,\beta\,\mathbf{S}_{uu}(u_s,\upsilon_j)
\end{aligned}\tag{D.6a}$$

$$\begin{aligned}
\mathbf{c}_3^j &= \mathbf{S}(u_s,\upsilon_{j_{opp}})+(2/5)\,\alpha\,\mathbf{S}_u(u_s,\upsilon_{j_{opp}})+(1/20)\,\beta\,\mathbf{S}_{uu}(u_s,\upsilon_{j_{opp}})\\
\mathbf{c}_4^j &= \mathbf{S}(u_s,\upsilon_{j_{opp}})+(1/5)\,\alpha\,\mathbf{S}_u(u_s,\upsilon_{j_{opp}})\\
\mathbf{c}_5^j &= \mathbf{S}(u_s,\upsilon_{j_{opp}})
\end{aligned}\tag{D.6b}$$

Then, the mean curve $\overline{\mathbf{C}}(t)=(2/m_S)\sum_{j=0}^{m_S/2-1}\mathbf{C}_j(t)$ also can be expressed in terms of α, β as follows:

$$\overline{\mathbf{C}}(t)=\frac{2}{m_S}\sum_{j=0}^{m_S/2-1}\mathbf{C}_j(t)=\frac{2}{m_S}\sum_{j=0}^{m_S/2-1}\sum_{i=0}^{5}B_{i,5}(t)\,\mathbf{c}_i^j=\sum_{i=0}^{5}B_{i,5}(t)\left(\frac{2}{m_S}\sum_{j=0}^{m_S/2-1}\mathbf{c}_i^j\right)\tag{D.7}$$

The mean-deviations of connection curves at $t=0.5$ in Equation (3.18) are

$$\begin{aligned}
\hat{\mathbf{C}}_j(\alpha,\beta:0.5) &= \mathbf{C}_j(\alpha,\beta:0.5)-\overline{\mathbf{C}}_j(\alpha,\beta:0.5)\\
&=\sum_{i=0}^{5}B_{i,5}(t)\left[\mathbf{c}_i^j-\left(\frac{2}{m_S}\sum_{j=0}^{m_S/2-1}\mathbf{c}_i^j\right)\right]=\sum_{i=0}^{5}B_{i,5}(t)\,\hat{\mathbf{c}}_i^j
\end{aligned}\tag{D.8}$$

where

$$\hat{\mathbf{c}}_0^j=\hat{\mathbf{S}}^j,\ \hat{\mathbf{c}}_1^j=\hat{\mathbf{S}}^j+(1/5)\,\alpha\,\hat{\mathbf{S}}_u^j,\ \hat{\mathbf{c}}_2^j=\hat{\mathbf{S}}^j+(2/5)\,\alpha\,\hat{\mathbf{S}}_u^j+(1/20)\,\beta\,\hat{\mathbf{S}}_{uu}^j,$$

$$\mathbf{c}_3^j=\hat{\mathbf{S}}^{j_{opp}}+(2/5)\,\alpha\,\hat{\mathbf{S}}_u^{j_{opp}}+(1/20)\,\beta\,\hat{\mathbf{S}}_{uu}^{j_{opp}},\ \mathbf{c}_4^j=\hat{\mathbf{S}}^{j_{opp}}+(1/5)\,\alpha\,\hat{\mathbf{S}}_u^{j_{opp}},\ \mathbf{c}_5^j=\hat{\mathbf{S}}^{j_{opp}};$$

$$\begin{cases}
\hat{\mathbf{S}}^j=\mathbf{S}(u_s,\upsilon_j)-\overline{\mathbf{S}}\\
\hat{\mathbf{S}}_u^j=\mathbf{S}_u(u_s,\upsilon_j)-\overline{\mathbf{S}}_u\\
\hat{\mathbf{S}}_{uu}^j=\mathbf{S}_{uu}(u_s,\upsilon_j)-\overline{\mathbf{S}}_{uu}
\end{cases},\quad
\begin{cases}
\hat{\mathbf{S}}^{j_{opp}}=\mathbf{S}(u_s,\upsilon_{j_{opp}})-\overline{\mathbf{S}}^{opp}\\
\hat{\mathbf{S}}_u^{j_{opp}}=\mathbf{S}_u(u_s,\upsilon_{j_{opp}})-\overline{\mathbf{S}}_u^{opp}\\
\hat{\mathbf{S}}_{uu}^{j_{opp}}=\mathbf{S}_{uu}(u_s,\upsilon_{j_{opp}})-\overline{\mathbf{S}}_{uu}^{opp}
\end{cases};$$

$$\begin{cases}
\overline{\mathbf{S}}=\dfrac{2}{m_S}\displaystyle\sum_{j=0}^{m_S/2-1}\mathbf{S}(u_s,\upsilon_j)\\
\overline{\mathbf{S}}_u=\dfrac{2}{m_S}\displaystyle\sum_{j=0}^{m_S/2-1}\mathbf{S}_u(u_s,\upsilon_j)\\
\overline{\mathbf{S}}_{uu}=\dfrac{2}{m_S}\displaystyle\sum_{j=0}^{m_S/2-1}\mathbf{S}_{uu}(u_s,\upsilon_j)
\end{cases},\quad
\begin{cases}
\overline{\mathbf{S}}^{opp}=\dfrac{2}{m_S}\displaystyle\sum_{j=0}^{m_S/2-1}\mathbf{S}(u_s,\upsilon_{j_{opp}})\\
\overline{\mathbf{S}}_u^{opp}=\dfrac{2}{m_S}\displaystyle\sum_{j=0}^{m_S/2-1}\mathbf{S}_u(u_s,\upsilon_{j_{opp}})\\
\overline{\mathbf{S}}_{uu}^{opp}=\dfrac{2}{m_S}\displaystyle\sum_{j=0}^{m_S/2-1}\mathbf{S}_{uu}(u_s,\upsilon_{j_{opp}})
\end{cases}.$$

Now, Equation (D.8) is explicitly expressed in terms of α, β as:

$$\hat{\mathbf{C}}_j(\alpha,\beta:0.5)=\mathbf{A}_0^j+\mathbf{A}_1^j\alpha+\mathbf{A}_2^j\beta \tag{D.9}$$

where

$$\begin{aligned}
\mathbf{A}_0^j &= \{B_{0,5}(0.5)+B_{1,5}(0.5)+B_{2,5}(0.5)\}(\hat{\mathbf{S}}^j+\hat{\mathbf{S}}^{j_{opp}}),\\
\mathbf{A}_1^j &= (1/5)\{B_{1,5}(0.5)+2B_{2,5}(0.5)\}(\hat{\mathbf{S}}_u^j+\hat{\mathbf{S}}_u^{j_{opp}}),\\
\mathbf{A}_2^j &= (1/20)B_{2,5}(0.5)(\hat{\mathbf{S}}_{uu}^j+\hat{\mathbf{S}}_{uu}^{j_{opp}}).
\end{aligned}$$

Finally, the sum of the squares of Equation (D.9)

$$E(\alpha,\beta)=\sum_{j=0}^{m_S/2-1}(\mathbf{A}_0^j+\mathbf{A}_1^j\alpha+\mathbf{A}_2^j\beta)^2 \tag{D.10}$$

is minimized when $\partial E/\partial\alpha=0$, $\partial E/\partial\beta=0$. Optimal-scale factors α^*, β^* can be calculated by solving following a matrix equation

$$\begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix}\begin{bmatrix} \alpha \\ \beta \end{bmatrix}=-\begin{bmatrix} B_{0,1} \\ B_{0,2} \end{bmatrix} \tag{D.11}$$

where $B_{k,l}=\sum_{j=0}^{m_S/2-1}\mathbf{A}_k^j\bullet\mathbf{A}_l^j$; $k=0,1,2$, $l=1,2$.

Appendix E. Origin Estimation of Local-coordinate System for *rank* Q$_u$= 2

Let $\mathbf{A} = 2\mathbf{Q}_u$ and $\mathbf{b} = -\mathbf{v}$, then Equation (4.2) can be written in a matrix form as follows:

$$\mathbf{A}\,\mathbf{x} = \mathbf{b} \tag{E.1}$$

For *rank* $\mathbf{A} = 2$, a general solution of Equation (E.1) is given by

$$\mathbf{x} = \mathbf{p} + t\mathbf{q}\,; \quad t \in R \tag{E.2}$$

where $\mathbf{p}$ is a *particular solution* of $\mathbf{A}\,\mathbf{x} = \mathbf{b}$, and the unit vector $\mathbf{q}$ is a solution set of $\mathbf{A}\,\mathbf{x} = \mathbf{0}$ ($\mathbf{p}$, $\mathbf{q}$ are calculated using the *singular-value decomposition* method [Pres1994]). In order to obtain the origin **o** of the local-coordinate system for the input COP-data $\{\mathbf{\Psi}_k\}$; $k = 1, \ldots, N$, t is determined so that the following objective function is minimized:

$$f(t) = \sum_{k=1}^{N} (\mathbf{\Psi}_k - \mathbf{o})^2 = \sum_{k=1}^{N} \{\mathbf{\Psi}_k - (\mathbf{p} + t\mathbf{q})\}^2 \tag{E.3}$$

Thus, the value of *t* satisfying $df(t)/dt = 0$ is obtained as follows:

$$t_o = \frac{1}{N}\mathbf{q} \bullet \sum_{k=1}^{N} (\mathbf{\Psi}_k - \mathbf{p}) \tag{E.4}$$

감사의글

본 논문이 완성되기까지 도와주신 모든 분들에게 감사의 말씀을 전합니다.

최 병규 교수님은 CAD/CAM 분야에 대한 막연한 동경과 갈망만이 모두였던 저에게 새로운 연구의 장을 허락하여 주셨고, 지난 6년간 각별한 관심으로 학문과 연구에 임하는 자세와 방법을 가르쳐 주셨습니다. 또한, 본 논문을 진행하는데 있어서도 온갖 어려움이 있을 때 마다 현명한 방향을 제시하여 주셨습니다. 교수님께서 몸소 실천하시는 열정적인 생활 태도와 연구 자세를 언제까지나 기억하고 인생의 본보기로 삼겠습니다. 진심으로 감사합니다.

논문심사를 맡아 주신 염 봉진 교수님, 김 승우 교수님, 이 태억 교수님, 김 덕수 교수님, 신 하용 교수님께 감사 드립니다. 교수님들의 깊이 있는 조언은 본 논문이 바른 방향으로 나아갈 수 있는 든든한 길잡이가 되어 주었습니다.

조 수경 선배님은 논리적인 사고와 유연한 대화로 학문적인 토론의 즐거움을 가르쳐 주었습니다. 본 논문의 기본 개념 중 많은 부분은 조 수경 선배님과 함께한 수학적 검토 끝에 도출되었습니다. 같이한 5년 동안의 수많은 분야의 질문과 대답은 언제 어디서나 꺼내어 쓸 수 있는 귀중한 자산이 되리라고 생각합니다.

유 석규 형님은 많은 시간과 노력을 들여 본 논문 4장의 곡면모델링 예제들을 만들어 주었습니다. 또한, 많은 부분에 있어 저의 모지라는 부분을 자상히 챙겨 주었고, 연구와 생활에 있어서의 어려움들을 극복해 나가는데 큰 힘이 되어 주었습니다.

정 원형 씨는 본 논문 5장의 과학적 가시화 기능을 구현하여 주었습니다. 탁월한 프로그래밍 실력은 그 동안의 연구 생활에 있어서 많은 도움이 되었습니다.

연구실 생활을 같이 한 모두 – 김 대현 선배님, 한 관희 선배님, 박 정현 선배님, 황 문호 선배님, 김 보현 선배님, 김 병희 선배님, 박 준철 선배님, 박 상철, 고 기훈 씨, 윤 진민, 박 범철, 김 성철, 차 은성 씨, 유 남규, 양 병운, 정 구환, 서 정철, 정 창영 씨, 황 현철, 황 동환, 김 종한, 류 호열, 임 정희, 김 혁래, 이 재원, 노 성철 – 에게 감사 드립니다.

정말 오랫동안 같은 길을 걸어 가며 의지할 수 있었던 경남과학고등학교 동기들, 선배님들, 그리고 후배들에게 감사 드립니다. 낯선 땅에서 만나서 다양한 경험을 함께 하였

던 대학교 동아리 친구들, 선배님들, 그리고 후배들에게 감사 드립니다.

10년이 넘도록 기쁨과 슬픔을 나누면서 사랑을 키워 왔던 이 은진에게도 감사 드립니다. 거친 세상을 함께 헤쳐 나갈 절친한 친구이자 평생의 연구 동료입니다.

마지막으로, 항상 지켜 보아 주시고 용기를 북돋아 주시는 집안 어른들과 친지들, 언제나 저를 좋아하고 잘 따르는 동생 배 진형, 아낌없이 사랑해 주시는 부모님께 감사 드립니다.

위의 모두에게 쉽지 않게 이룬 이 작은 결실을 바칩니다.

이력서

성명: 배석형 (裵錫亨)

생년월일: 1972년 7월 19일

출생지: 경상남도

본적: 경상남도 마산시 진동면 인곡리 217

학력

1990-1994: 한국과학기술원 기계공학과 (B.S.)

1994-1996: 한국과학기술원 기계공학과 (M.S.)

1996-2002: 한국과학기술원 산업공학과 (Ph.D.)