ELSEVIER

Computer-Aided Design 34 (2002) 683–690

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

NURBS surface fitting using orthogonal coordinate transform for rapid product development

Seok-Hyung Bae, Byoung K. Choi*

Department of Industrial Engineering, KAIST, 373-1 Guseong-dong, Yuseong-gu, Daejeon, 305-701, South Korea

Accepted 13 November 2001

Abstract

Constructing a CAD model from a physical model plays a key role in some rapid product development processes. Presented in the paper is a method of fitting NURBS surfaces for *rotational freeform shapes*: (1) cloud-of-points data (COP-data) representing a rotational freeform shape are transformed into an *orthogonal coordinate* 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 COP-data represent a *true surface-of-revolution*, and allows an easy point-membership classification for NURBS-bounded solid objects. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Rapid technologies; Reverse engineering; Cloud-of-points data fitting; Orthogonal coordinate transform; Symbolic product operation; Rotation freeform shape; NURBS-representation

1. Introduction

Physical models play a key role in some rapid product development processes. When designing an aesthetic product, physical models (made of clay or chemical wood) are employed for aesthetic evaluation. Sometimes, designers start with existing products when designing a new product shape, which is known as *shape reverse engineering* [1]. 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.

Presented in the paper is an 'indirect' method of fitting NURBS surface from COP-data representing a *rotational freeform shape* that one-to-one maps to a sphere or cylinder. In order to construct a parametric surface $\mathbf{S}(u, v)$ 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 [2], but is not a trivial task for a rotational freeform shape. Another issue in handling a rotational freeform shape is how to recover *exact NURBS geometry* when the COP-data represent a 'true' *surface-of-revolution* shape (which includes sphere and cylinder) [3,4].

Various methods have been proposed for fitting sphere/cylinder [5] and surface-of-revolution shapes [6–8]. These 'specialized' methods are efficient when we know that the COP-data on hand represent a sphere, cylinder or surface-of-revolution shape. Also proposed in the literature are methods for fitting NURBS surface from general COP-data. Hoschek [9] employed a circular spline and Laurent-Gengoux and Mekhilef [10] treated NURBS surface fitting as a non-linear optimization problem. These 'general' methods focus on determining the weights of NURBS control points (so that the extra degrees-of-freedom provided by the weights are best utilized), but they are not quite suitable for COP-data representing rotational freeform shapes.

In this paper, we propose a NURBS surface fitting method for rotational freeform shapes that provides a natural means to parameterization, enables to recover exact NURBS geometry for a true surface-of-revolution, and allows an easy point-membership classification for NURBS-bounded solid objects. 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 function* 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* [11] with a *NURBS base-geometry*.

* Corresponding author. Tel.: +82-42-869-3115; fax: +82-42-869-3110.
E-mail address: bkchoi@vmslab.kaist.ac.kr (B.K. Choi).

PII: S0010-4485(01)00198-1

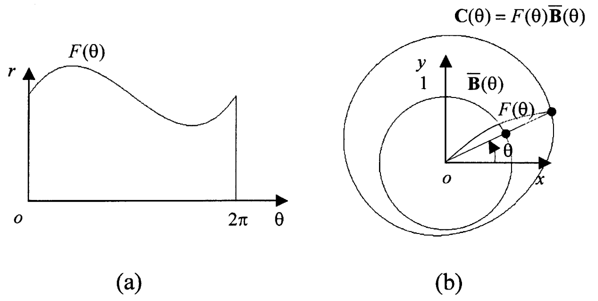


Fig. 1. (a) A single-valued function in polar coordinates, and (b) its corresponding 2D parametric curve in *xy*-coordinate system.

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 [12]. Point-membership classification for rotational freeform shapes has been studied quite extensively by Sanchez-Reyes [13–16]. The Sanchez-Reyes' method, which uses a *sinusoidal spline function* as the basis function, is similar to the method proposed in the paper 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 of the paper is as follows: described in Section 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 Section 3. Also presented in Section 3 are special treatments and a method for estimating local coordinate systems. Illustrative examples are given in Section 4, and Section 5 closes the paper with discussions and conclusions.

2. NURBS curve fitting procedure

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

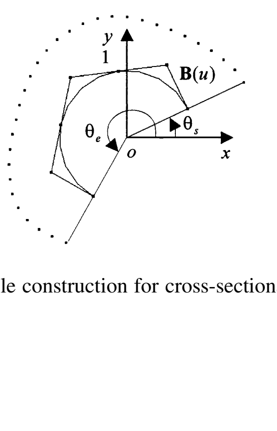


Fig. 2. Base circle construction for cross-sectional data points.

2.1. Basic idea

Let $\bar{\mathbf{B}}(\theta) = (\cos\theta, \sin\theta)$ 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 a product of $\bar{\mathbf{B}}(\theta)$ and a *single-valued function* $F(\theta)$ as follows:

$$\mathbf{C}(\theta) \equiv (x(\theta), y(\theta)) = F(\theta)\bar{\mathbf{B}}(\theta) = (F(\theta)\cos\theta, F(\theta)\sin\theta) \tag{1}$$

where the unit circle $\bar{\mathbf{B}}(\theta)$ is called a (trigonometric) *base circle*. A construction for the above relationship is shown in Fig. 1.

A sequence of 2D points $\{(x_k, y_k)\}$ representing a rotational freeform curve may be fitted to the curve $\mathbf{C}(\theta)$ in Eq. (1) as follows:

1. $\{(x_k, y_k)\}$ is converted into a polar-coordinate point-sequence $\{(r_k, \theta_k)\}$.
2. $\{(r_k, \theta_k)\}$ 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 Eq. (1).

By replacing the *trigonometric base-circle* $\bar{\mathbf{B}}(\theta)$ in Eq. (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. The NURBS base-circle $\mathbf{B}(u)$ may consist of four *rational quadratic Bezier* curves (RQB-curves) [3] and it does not have a uniform flow rate [17]. Now we present our NURBS curve fitting procedure.

2.2. NURBS curve fitting procedure

The overall procedure for fitting a 2D NURBS curve from a sequence of 2D points $\{(x_k, y_k)\}$ consists of five steps: (1) polar coordinate transform, (2) base circle construction, (3) parameter synchronization, (4) single-valued B-spline curve fitting, and (5) NURBS curve construction.

Step 1: Polar coordinate transform

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

$$r_k = \sqrt{x_k^2 + y_k^2},$$

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

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* [18,19], where a full circle is defined by four RQB-curves. Shown in Fig. 2 is a sequence of polar coordinate points $\{(r_k, \theta_k)\}$ inside an

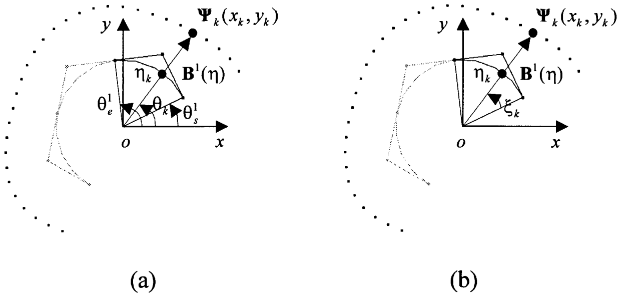


Fig. 3. Geometric construction for parameter synchronization (i.e. finding η_k).

angle range of [θ_s, θ_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.

Step 3: Curve parameter synchronization

The *synchronized parameter value* u_k of the base circle $\mathbf{B}(u)$ for a *polar-coordinate point* (r_k, θ_k) corresponds to the intersection point between $\mathbf{B}(u)$ and the line joining the origin and the *2D-Cartesian point* $\mathbf{\Psi}_k(x_k, y_k)$. To find u_k at the intersection-point, $\mathbf{B}(u)$ is first decomposed into individual RQB-curves $\{\mathbf{B}^j(\eta)\}$ for $u \in [u_s^j, u_e^j]$. Now, a *local parameter* for each RQB-curve is defined as

$$\eta = \left(u - u_s^j\right)/\left(u_e^j - u_s^j\right), \qquad u \in \left[u_s^j, u_e^j\right]. \tag{3}$$

Then, the local parameter value η_k for a polar-coordinate point (r_k, θ_k) within the angle range [θ_s^j, θ_e^j] may be obtained from the following relation:

$$(\cos\theta_k, \sin\theta_k) = \mathbf{B}^j(\eta) \tag{4}$$

A solution to Eq. (4) is given by [20]

$$\eta_k = \frac{\tan\dfrac{\xi_k}{2}}{\sqrt{1 - w^2} + (1 - w)\tan\dfrac{\xi_k}{2}}, \qquad \xi_k = \theta_k - \theta_s^j \tag{5}$$

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

$$u_k = \left(u_e^j - u_s^j\right)\eta_k + u_s^j \tag{6}$$

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 Eq. (6) reduces to $u_k = \eta_k + (j - 1)$.

Shown in Fig. 3(a) is a construction for finding the intersection parameter η_k between $\mathbf{B}^1(\eta)$ and the line joining the coordinate origin and $\mathbf{\Psi}_k(x_k, y_k)$. The relation between η_k and ξ_k is depicted in Fig. 3(b).

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* $\{(r_k, u_k)\}$ by using standard fitting methods [19,21–24].

$$F(u) = \sum_{i=0}^{n_F} N_{i,p}(u) f_i;$$
$$U_F = \{\underbrace{u_s, \ldots, u_s}_{p+1}, u_{p+1}, \ldots, u_{n_F}, \underbrace{u_e, \ldots, u_e}_{p+1}\} \tag{7}$$

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* [11] with the NURBS base circle $\mathbf{B}(u)$. Namely, from $\mathbf{C}(u) = F(u)\mathbf{B}(u)$ we have

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n_C} N_{i,p+2}(u) w_i \mathbf{P}_i}{\sum_{i=0}^{n_C} N_{i,p+2}(u) w_i};$$
$$U_C = \{\underbrace{u_s, \ldots, u_s}_{p+2+1}, u_{p+2+1}, \ldots, u_{n_C}, \underbrace{u_e, \ldots, u_e}_{p+2+1}\} \tag{8}$$

where $N_{i,p+2}(u)$ is a B-spline basis function of degree $p + 2$, w_i is a weight, and $\mathbf{P}_i$ is a control point of the NURBS curve $\mathbf{C}(u)$. The above NURBS curve construction concept is illustrated in Fig. 4, where the degree of $F(u)$ is 3 and that of the resulting NURBS curve is 5.

If the 2D point sequence $\{(x_k, y_k)\}$ 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 ($p = 0$) so that the degree of the resulting NURBS curve becomes 2. That is, $\mathbf{C}(u) = R\mathbf{B}(u)$ becomes a true circle of radius R.

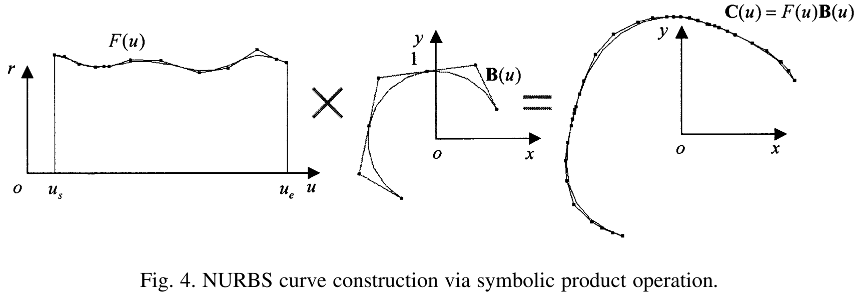


Fig. 4. NURBS curve construction via symbolic product operation.

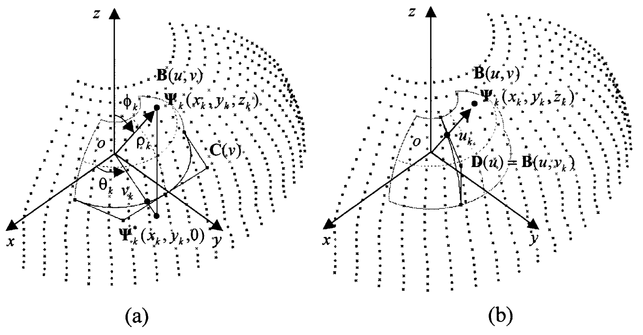


Fig. 5. Geometric construction for parameter synchronization for spherical NURBS surface fitting (i.e. finding v_k and u_k).

3. NURBS surface fitting procedure

The procedure for fitting a NURBS surface from COP-data is similar to the curve fitting procedure given in Section 2.2, and consists of five steps: (1) orthogonal coordinate transform, (2) base geometry construction, (3) parameter synchronization, (4) single-valued B-spline surface fitting, and (5) NURBS surface construction. A rotational freeform 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

Step 1: Spherical coordinate transform

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

$$\rho_k = \sqrt{x_k^2 + y_k^2 + z_k^2}, \qquad \phi_k = \cos^{-1}\left(z_k/\sqrt{x_k^2 + y_k^2 + z_k^2}\right),$$

$$\theta_k = \begin{cases} \tan^{-1}(y_k/x_k) & \text{if } x_k > 0 \text{ and } y_k \geq 0 \\ \pi + \tan^{-1}(y_k/x_k) & \text{if } x_k < 0 \\ 2\pi + \tan^{-1}(y_k/x_k) & \text{if } x_k > 0 \text{ and } y_k < 0 \end{cases} \tag{9}$$

And, the *angle ranges* are computed as

$$\phi_s = \min\{\phi_k\}, \qquad \phi_e = \max\{\phi_k\}, \qquad \theta_s = \min\{\theta_k\},$$

$$\theta_e = \max\{\theta_k\}.$$

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) RQB patches depending on *arc angles* $\phi_e - \phi_s$ and $\theta_e - \theta_s$ [3].

Step 3: Parameter synchronization

Similar to the curve fitting case, finding the *synchronized parameter values* of the base surface $\mathbf{B}(u,v)$ for a spherical coordinate point $(\rho_k, \phi_k, \theta_k)$ is equivalent to finding an intersection point between $\mathbf{B}(u,v)$ and the line joining the origin and the *3D-Cartesian point* $\boldsymbol{\Psi}_k(x_k, y_k, z_k)$. Since the base sphere $\mathbf{B}(u,v)$ is constructed by a surface-of-revolution method [3], 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 Fig. 5(a), v_k is obtained from $\mathbf{C}(v)$ and $\boldsymbol{\Psi}^*_k(x_k, y_k, 0)$, where $\boldsymbol{\Psi}^*_k(x_k, y_k, 0)$ is the projection point of $\boldsymbol{\Psi}_k(x_k, y_k, z_k)$ on the *xy*-plane and $\mathbf{C}(v)$ is a NURBS base circle on the *xy*-plane with an arc angle $\theta_e - \theta_s$. Once v_k is determined, another base circle $\mathbf{D}(u) = \mathbf{B}(u, v_k)$ is defined as depicted in Fig. 5(b), and then, u_k is calculated from $\mathbf{D}(u)$ and $\boldsymbol{\Psi}_k(x_k, y_k, z_k)$.

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* $\{(\rho_k, u_k, v_k)\}$ by using standard fitting methods [19,21–24].

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

$$U_F = \{ \underbrace{u_s, \ldots, u_s}_{p+1}, u_{p+1}, \ldots, u_{n_F}, \underbrace{u_e, \ldots, u_e}_{p+1} \}, \tag{10}$$

$$V_F = \{ \underbrace{v_s, \ldots, v_s}_{q+1}, v_{q+1}, \ldots, v_{m_F}, \underbrace{v_e, \ldots, v_e}_{q+1} \}$$

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 $\mathbf{B}(u,v)$. That is, $\mathbf{S}(u,v) = F(u,v)\mathbf{B}(u,v)$ having a tensor product form of the NURBS curve in Eq. (8), where the degrees of the resulting NURBS surface are $(p+2, q+2)$. Shown in Fig. 6(a) is a *bi-cubic B-spline function* (i.e. $p = q = 3$) for the parameter-synchronized points and the resulting NURBS surface of degree (5,5) obtained from NURBS surface conversion is shown in Fig. 6(b).

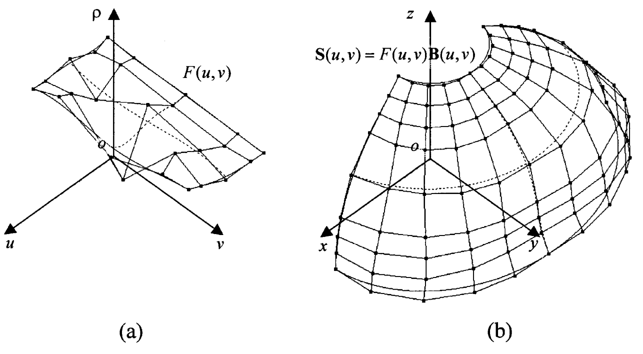


Fig. 6. NURBS surface obtained by spherical NURBS surface fitting.

3.2. Cylindrical NURBS surface fitting

When the rotational freeform shape is more naturally mapped to a cylinder, a *cylindrical NURBS surface* is fitted

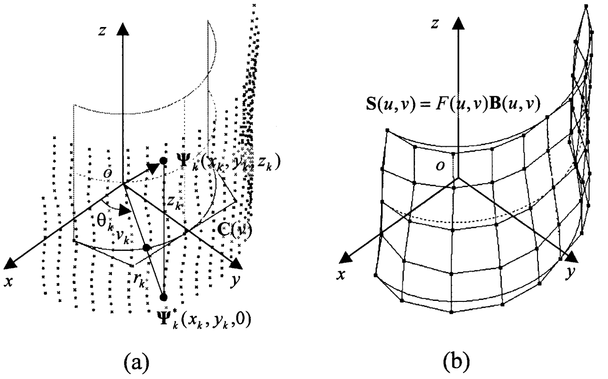


Fig. 7. NURBS surface obtained by cylindrical NURBS surface fitting.

from the COP-data. The procedure is basically the same as that of the spherical case described in Section 3.1.

First, *3D points* $\{(x_k, y_k, z_k)\}$ are transformed into *cylindrical coordinate points* $\{(r_k, \theta_k, z_k)\}$ 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). 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 Fig. 7(a), and u_k is given by $u_k = (z_k - z_s)/(z_e - z_s)$. And then, a single-valued bi-cubic B-spline surface $F(u,v)$ is fitted as with the spherical case. Finally, a 3D NURBS surface $\mathbf{S}(u,v)$ is given by:

$$\mathbf{S}(u,v) = \frac{\sum_{i=0}^{n_S}\sum_{j=0}^{m_S} N_{i,3}(u)N_{j,5}(v)w_{i,j}\mathbf{P}_{i,j}}{\sum_{i=0}^{n_S}\sum_{j=0}^{m_S} N_{i,3}(u)N_{j,5}(v)w_{i,j}};$$

$$U_S = \{\underbrace{u_s, \dots, u_s}_{3+1}, u_{3+1}, \dots, u_{n_S}, \underbrace{u_e, \dots, u_e}_{3+1}\}, \tag{11}$$

$$V_S = \{\underbrace{v_s, \dots, v_s}_{5+1}, v_{5+1}, \dots, v_{m_S}, \underbrace{v_e, \dots, v_e}_{5+1}\}$$

Shown in Fig. 7(b) is the resulting NURBS surface of degree (3,5) for the COP-data of Fig. 7(a).

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 $(\max\{p,1\}, q+2)$. 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 cylindrical shape, the degrees of $\mathbf{S}(u,v)$ become (1,2).

3.3. Special treatments

When a 'closed' surface is to be constructed, *periodic B-spline basis functions* [25] have to be employed in the v-direction (corresponding to θ) of $F(u,v)$ during the single-valued B-spline surface fitting stage.

A special treatment is also required to solve the 'degeneracy problem' when fitting a spherical NURBS surface. In order to smoothly fill the 'hole' near the pole of the base sphere (see Fig. 8(a)), the following *continuity conditions* must be imposed on the control points of the B-spline function

$$f_{0,j} = f_{0,0}; \qquad j = 1, \dots, m_F \tag{12}$$

$$f_{1,j} = f_{0,j}; \qquad j = 0, \dots, m_F \tag{13}$$

Shown in Fig. 8(b) is the resulting NURBS surface satisfying G^1 continuity condition.

3.4. Local coordinate system estimation

The proposed NURBS surface fitting method requires that the COP-data in a global Cartesian coordinate system be transformed to a *local Cartesian coordinate system* where the orthogonal coordinates are defined. This in turn requires a robust estimation of local coordinate system from the original COP-data. For the purpose, *principal component analysis* (PCA) [26,27], which is also known as *Karhunen–Loeve transform*, is widely used in the field of image processing and computer vision. 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 local coordinate system defining the oriented bounding box of the COP-data. Presented in this sub-section is a local coordinate system

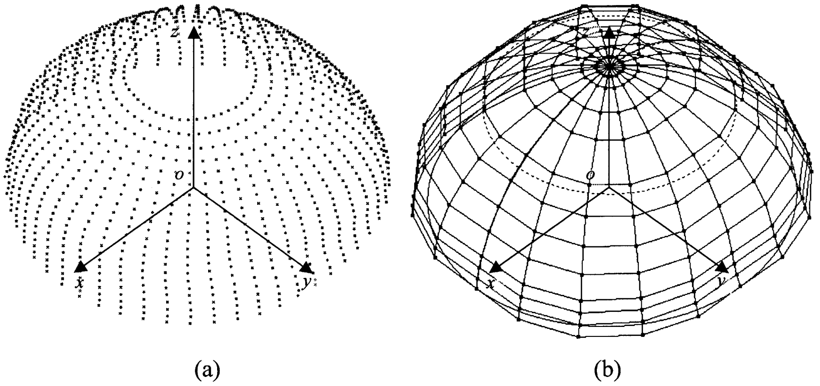


Fig. 8. Cloud-of-points data of sphere-like part with degeneracy problem and resulting NURBS surface with G^1 continuity at degenerate point.

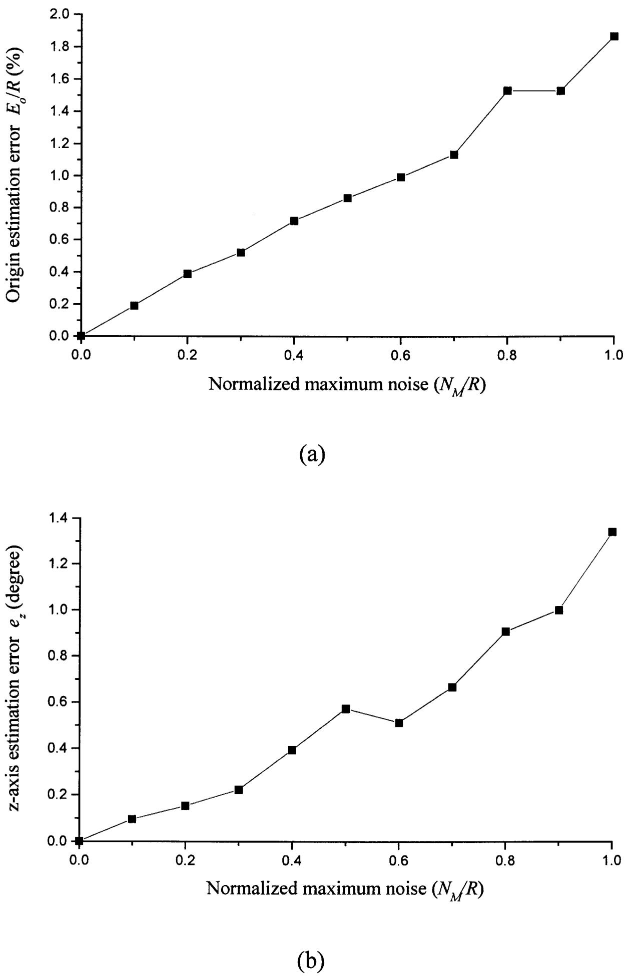


Fig. 9. Empirical results of local coordinate system estimation.

estimation method based on the *least squares approximation* method proposed by Faugeras and Hebert [28].

A general quadric surface is a second-degree algebraic surface having the following matrix form:

$$q(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{Q}_u\mathbf{x} + \mathbf{v}^{\mathrm{T}}\mathbf{x} + d = 0; \qquad \mathbf{x} \equiv \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

$$\mathbf{Q}_u \equiv \begin{bmatrix} q_1 & q_4/2 & q_6/2 \\ q_4/2 & q_2 & q_5/2 \\ q_6/2 & q_5/2 & q_3 \end{bmatrix}, \qquad \mathbf{v} \equiv \begin{bmatrix} q_7 \\ q_8 \\ q_9 \end{bmatrix}, \qquad d = q_{10} \tag{14}$$

Quadric surface may be classified into different types based on its coefficients [29,30] and the following six types are relevant to local coordinate system estimation: real ellipsoid, hyperboloid of one sheet, hyperboloid of two sheets, real quadric cone, elliptic paraboloid, real elliptic cylinder. Namely, when COP-data are fitted to one of the six quadric surface types, its coordinate frame (i.e. *z-axis vector* **k** and origin **o**) can be uniquely determined from the subdiscriminant matrix $\mathbf{Q}_u$. The *z-axis vector* **k** is set to the eigenvector having the smallest eigenvalue [31]. Thus, the rotational axis of a surface-of-revolution shape becomes the eigenvector with the smallest eigenvalue.

And, the *origin* **o** of the local coordinate system is determined from the following relation:

$$2\mathbf{Q}_u\mathbf{x} + \mathbf{v} = 0 \tag{15}$$

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 **o** is given by [28,32]

$$\mathbf{o} = -\frac{1}{2}\mathbf{Q}_u^{-1}\mathbf{v} \tag{16}$$

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

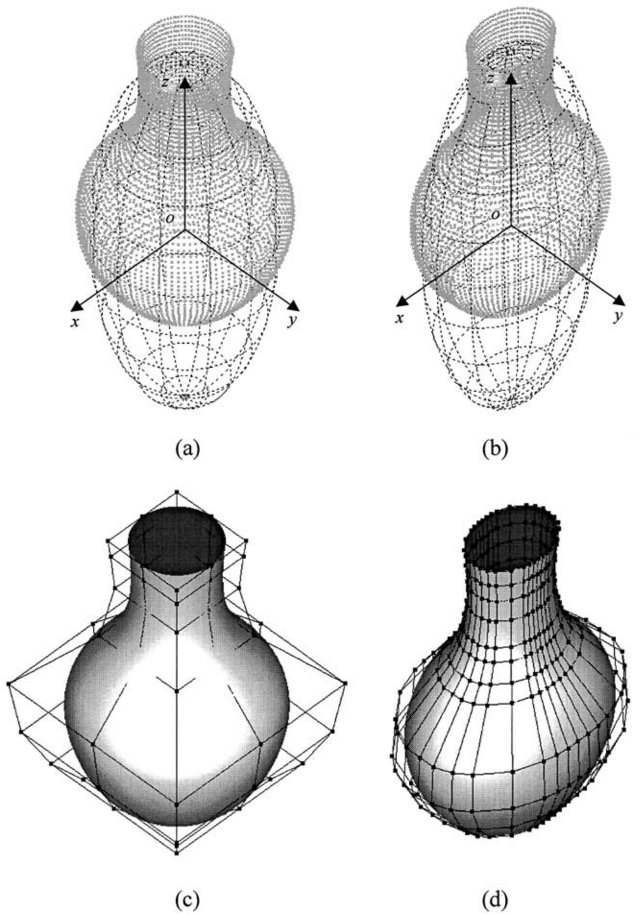
$$\mathbf{o} = \mathbf{p} + t_0\mathbf{q} \tag{17}$$

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 Fig. 9(a) are *relative origin estimation errors* (in %) plotted against the *normalized maximum noise* N_{M}/R (for the sphere COP-data), where $N_{\mathrm{M}} = 3\sigma$. The z-axis estimation errors (in degrees) plotted against N_{M}/R are shown in Fig. 9(b) (for the cylinder COP-data).

4. Illustrative examples

The proposed NURBS surface fitting methods were applied to different types of COP-data. Shown in Fig. 10(a) are COP-data for a traditional Korean ceramic. Also shown are the approximated quadric surface (real ellipse) and the estimated local coordinate system (x-, y-axis are arbitrarily chosen). The fitted NURBS surface together with its control points for the COP-data of Fig. 10(a) is shown in Fig. 10(c). 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 shape. Now, the COP-data of Fig. 10(a) are scaled down (0.75) along the y-direction to obtain a rotational freeform shape as shown in Fig. 10(b). Also shown in Fig. 10(b) are the approximated quadric surface and its local coordinate system. The fitted NURBS surface of degrees (3,5) and its control points are shown in Fig. 10(d).

Shown in Fig. 11(a) are COP-data for an auto-gearshift knob obtained from a coordinate measuring machine (CMM). The initial COP-data were split into two data sets (defined on the same coordinate system of Fig. 11(a)): upper (head) data and lower (neck) data. The 'head' COP-data were fitted to spherical NURBS surface and the 'neck'



(a) (b)

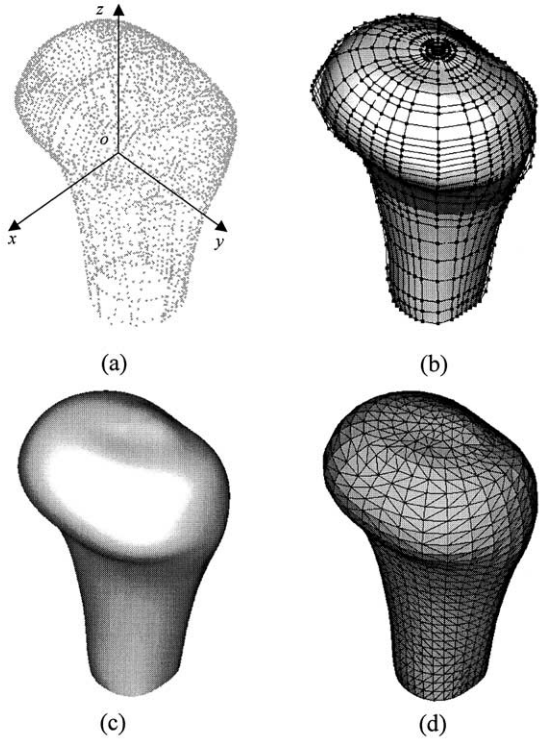
(c) (d)

Fig. 10. NURBS surface fitting for traditional Korean ceramic.

COP-data to cylindrical NURBS surface, and then the two NURBS surfaces are joined together as shown in Fig. 11(b). In order to achieve G^1 continuity between the two surfaces, the control points near the common boundary have to be 'corrected' (this is an important issue by itself, but will not be discussed in this paper). A shaded image and facet model (for rapid prototyping) of the joined NURBS surface are shown in Fig. 11(c) and (d), respectively.

5. 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 surface is constructed 'directly' from COP-data. 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. The proposed method allows the NURBS surface to be stored as a single-valued function, which requires less memory and, more importantly, it allows an easier point-membership classification for NURBS-bounded solid objects.



(a) (b)

(c) (d)

Fig. 11. Surface modeling for auto-gearshift knob data.

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

Acknowledgements

The Ministry of Science and Technology of Korean government has supported the research.

Appendix A. Origin of local coordinate system when rank $\mathbf{Q}_u = 2$

Let $\mathbf{A} = 2\mathbf{Q}_u$ and $\mathbf{b} = -\mathbf{v}$, then Eq. (15) can be written in a matrix form as follows:

$$\mathbf{Ax} = \mathbf{b} \tag{A1}$$

When rank $\mathbf{A} = 2$, a general solution of Eq. (A1) is given by

$$\mathbf{x} = \mathbf{p} + t\mathbf{q}; \qquad t \in R \tag{A2}$$

where $\mathbf{p}$ is a *particular solution* of $\mathbf{Ax} = \mathbf{b}$, and the unit vector $\mathbf{q}$ is a solution set of $\mathbf{Ax} = 0$. $\mathbf{p}$, $\mathbf{q}$ are calculated using the *singular value decomposition* method [33]. In order to obtain the origin $\mathbf{o}$ of the local coordinate system for the input COP-data $\{\boldsymbol{\Psi}_k(x_k, y_k, z_k);\ k = 1, \ldots, N\}$, t is determined so that the following objective function is minimized:

$$f(t) = \sum_{k=1}^{N} (\boldsymbol{\Psi}_k - \mathbf{o})^2 = \sum_{k=1}^{N} \{\boldsymbol{\Psi}_k - (\mathbf{p} + t\mathbf{q})\}^2 \tag{A3}$$

Thus, the value of t satisfying $\mathrm{d}f(t)/\mathrm{d}t = 0$ is obtained as follows:

$$t_0 = \frac{1}{N}\mathbf{q}\cdot \sum_{k=1}^{N} (\boldsymbol{\Psi}_k - \mathbf{p}) \tag{A4}$$

References

[1] Varady T, Martin RR, Cox J. Reverse engineering of geometric models—an introduction. Comput-Aided Des 1997;29(4):255–68.

[2] Ma W, Kruth JP. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. Comput-Aided Des 1995;27(9):663–75.

[3] Piegl L, Tiller W. Curve and surface constructions using rational B-splines. Comput-Aided Des 1987;19(9):485–98.

[4] Piegl L. On NURBS: a survey. IEEE Comput Graphics Appl 1991; 10(1):55–71.

[5] Lukacs G, Marshall AD, Martin RR. Geometric least-squares fitting of spheres, cylinders, cones and tori. In: Martin RR, Varady T, editors, 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.

[6] Pottmann H, Randrup T. Rotational and helical surface approximation for reverse engineering. Computing 1998;60:307–22.

[7] Pottmann H, Chen HY, Lee IK. Approximation by profile surface. In: Cripps R, editor. The mathematics of surfaces VIII, Information Geometers, 1998. p. 17–36.

[8] Lai JY, Ueng WD. Reconstruction of surfaces of revolution from measured points. Comput Ind 2000;41:147–61.

[9] Hoschek J. Circular splines. Comput-Aided Des 1992;24(11):611–8.

[10] Laurent-Gengoux P, Mekhilef M. Optimization of a NURBS representation. Comput-Aided Des 1993;25(11):699–710.

[11] Piegl L, Tiller W. Symbolic operators for NURBS. Comput-Aided Des 1997;29(5):361–8.

[12] Nielson G. CAGD's top ten: what to watch. IEEE Comput Graphics Appl 1993;13(1):35–7.

[13] Sanchez-Reyes J. Single-valued curves in polar coordinates. Comput-Aided Des 1990;22(1):19–26.

[14] Sanchez-Reyes J. Single-valued surfaces in cylindrical coordinates. Comput-Aided Des 1991;23(8):561–8.

[15] Sanchez-Reyes J. Single-valued spline curves in polar coordinates. Comput-Aided Des 1992;24(6):307–15.

[16] Sanchez-Reyes J. Single-valued surfaces in spherical coordinates. Comput Aided Geom Des 1994;11:491–517.

[17] Farouki RT, Sakkalis T. Real rational curves are not unit speed. Comput Aided Geom Des 1991;8:151–8.

[18] Tiller W. Rational B-splines for curve and surface representation. IEEE Comput Graphics Appl 1983;3(6):61–9.

[19] Piegl L, Tiller W. The NURBS book. Berlin: Springer, 1995.

[20] Blanc C, Schlick C. Accurate parameterization of conics by NURBS. IEEE Comput Graphics Appl 1996;16(6):64–71.

[21] Choi BK. Surface modeling for CAD/CAM. Amsterdam: Elsevier, 1991.

[22] de Boor C. A practical guide to splines. Berlin: Springer, 1978.

[23] Farin G. Curves and surfaces for computer aided geometric design: a practice guide. London: Academic Press, 1988.

[24] Hoschek J, Lasser D. Fundamentals of computer aided geometric design. AK Peters, 1993.

[25] Park H, Kim K. Smooth surface approximation to serial cross-sections. Comput-Aided Des 1996;28(12):995–1005.

[26] Jain AK. Fundamentals of digital image processing. Englewood Cliffs, NJ: Prentice Hall, 1989.

[27] Pratt WK. Digitial image processing. 2nd ed. New York: Wiley, 1991.

[28] Faugeras OD, Hebert M. The representation, recognition and locating of 3-D objects. Int J Robotics Res 1986;5(3):27–52.

[29] Levin J. A parametric algorithm for drawing pictures of solid objects composed of quadric surfaces. Commun ACM 1976;19(10):555–63.

[30] Zwillinger D, Kellogg B. CRC standard mathematical tables and formulae. 30th ed. Boca Raton: CRC Press, 1996.

[31] Pettofrezzo AJ, Lacatena MM. Analytic geometry with vectors. Glenview, IL: Scott, Foresman, 1970.

[32] Marshall AD, Martin RR. Computer vision, models and inspection. Singapore: World Scientific, 1992.

[33] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical recipes in C: the art of scientific computing. 2nd ed. Cambridge: Cambridge University Press, 1994.

Seok-Hyung Bae is a PhD candidate in the Department of Industrial Engineering at Korea Advanced Institute of Science and Technology. He received a BS and an MS in Mechanical Engineering from KAIST in 1994 and 1996, respectively. His research interests are in the area of surface modeling, CAD/CAM, and reverse engineering.

Byoung K. Choi is a Professor of Manufacturing Systems Engineering in the Department of Industrial Engineering at Korea Advanced Institute of Science and Technology since he joined KAIST in 1983. He received a BS from Seoul National University, an MS from KAIST, and a PhD from Purdue University, all in Industrial Engineering. His research interests are in the area of sculptured surface modeling, die-cavity machining, CAPP, system modeling and simulation, and virtual manufacturing.