Geometric Fitting of Parametric Curves and Surfaces

Sung Joon Ahn*

Abstract: This paper deals with the geometric fitting algorithms for parametric curves and surfaces in 2-D/3-D space, which estimate the curve/surface parameters by minimizing the square sum of the shortest distances between the curve/surface and the given points. We identify three algorithmic approaches for solving the nonlinear problem of geometric fitting. As their general implementation we describe a new algorithm for geometric fitting of parametric curves and surfaces. The curve/surface parameters are estimated in terms of form, position, and rotation parameters. We test and evaluate the performances of the algorithms with fitting examples.

1. Introduction

1.1 Background

The curve/surface fitting to a set of given points is one of the common and frequently carried out tasks in many disciplines of science and engineering, e.g., computer vision, machine vision, coordinate metrology, reverse engineering, and computer-aided geometric design. This paper is concerned with the algorithms for least-squares fitting of parametric curves/surfaces $\mathbf{X}(\mathbf{a}, \mathbf{u})$ in 2-D/3-D space.

When we are particularly interested in fitting a set of measurement points in 2-D/3-D space, the shortest distance (known as geometric distance, Euclidean distance, or orthogonal distance in literature) between the model curve/surface and the measurement point is of practical meaning [1,13,15]. The least-squares model fitting with this error measure is called geometric fitting or orthogonal distance fitting in literature [2,3,4,5,8,10,11,19]. The geometric fitting problem is a nonlinear minimization problem to be solved through iteration and has been widely recognized as an analytically and computationally difficult problem. We can find in literature only a couple of original algorithms for geometric fitting of general curves/surfaces, [4] for explicit, [2,10,19] for implicit, and [7,11,18] for parametric curves/surfaces.

The analytical and computational difficulties in computing and minimizing the geometric distances blocked for a long time the development of geometric fitting algorithms for general curves/surfaces [4,5]. To bypass the difficulties

associated with the geometric fitting, most data processing software tools have been using the approximating measures of the geometric distance [6,8,9,14,17,20]. Nevertheless, when a highly accurate and reliable estimation of model parameters is essential to applications, we have no choice but to use the geometric distance as the error measure for least-squares model fitting, despite the high computing cost and the difficulties in developing the geometric fitting algorithms. In fact, the use of the geometric error measure is prescribed by an international standard for testing the data processing softwares for coordinate metrology [13].

1.2 Parametric Curves and Surfaces in Space

With applications of computer vision, motion analysis, robotics, and object visualization, the rigid body motion of objects is of great interest. For example, solving the bin-picking or obstacle-avoidance problem with robotics requires the object description in terms of shape, size, position, and orientation. Thus, a curve/surface fitting algorithm is highly advantageous for applications, if the curve/surface parameters $\mathbf{a}$ are grouped and estimated in terms of form $\mathbf{a}_g$, position $\mathbf{a}_p$, and rotation parameters $\mathbf{a}_r$ [2]

$$
\begin{aligned}
\mathbf{a}^\mathrm{T} &= (\mathbf{a}_g^\mathrm{T}, \mathbf{a}_p^\mathrm{T}, \mathbf{a}_r^\mathrm{T}) \\
&= (\underbrace{a_1, \ldots, a_l}_{l}, \underbrace{X_o, Y_o, Z_o}_{n}, \underbrace{\omega, \varphi, \kappa}_{s}) \\
&= (\underbrace{a_1, \ldots, a_p}_{p}).
\end{aligned}
$$

The *form parameters* $\mathbf{a}_g$ represent the shape and size of the standard (canonical) curve/surface defined in a model coordinate system xyz

$$\begin{cases} \mathbf{x}(\mathbf{a}_g, u) & : \text{curve(univariate)} \\ \mathbf{x}(\mathbf{a}_g, \mathbf{u}) & : \text{surface(bivariate)}. \end{cases} \tag{1}$$

The form parameters are invariant to the rigid body motion of curve/surface. The *position parameters* $\mathbf{a}_p$ and the *rotation parameters* $\mathbf{a}_r$ describe the rigid body motion of curve/surface in a machine coordinate system XYZ

$$\mathbf{X} = \mathbf{R}^{-1}\mathbf{x} + \mathbf{X}_o \quad \text{or} \quad \mathbf{x} = \mathbf{R}(\mathbf{X} - \mathbf{X}_o), \quad \text{where}$$
$$\mathbf{R} = \mathbf{R}_{\omega,\varphi,\kappa} = (\mathbf{r}_x \quad \mathbf{r}_y \quad \mathbf{r}_z)^T, \quad \mathbf{R}^{-1} = \mathbf{R}^T, \tag{2}$$
$$\mathbf{a}_p = \mathbf{X}_o = (X_o, Y_o, Z_o)^T, \quad \text{and} \quad \mathbf{a}_r = (\omega, \varphi, \kappa)^T.$$

2. Least-Squares Geometric Fitting

The goal of the geometric fitting of curves/surfaces to a set of given points is the determination of the model parameters that *minimize* the square sum of the *minimum* distances between the given points and the model curve/surface. We have to solve two subproblems that not only do the square sum have to be minimized but also every single distance between the given points and the curve/surface. In order to solve the geometric fitting problems we introduce two performance indices (cost functions or energy functions) representing – in two different ways – the square sum of the weighted distances between the given points and the curve/surface [2]:

$$\sigma_0^2 = \left\| \mathbf{P}(\mathbf{X} - \mathbf{X}') \right\|^2$$
$$= (\mathbf{X} - \mathbf{X}')^T \mathbf{P}^T \mathbf{P}(\mathbf{X} - \mathbf{X}') \tag{3}$$

and

$$\sigma_0^2 = \left\| \mathbf{Pd} \right\|^2$$
$$= \mathbf{d}^T \mathbf{P}^T \mathbf{Pd}, \tag{4}$$

where

$\mathbf{X}$: coordinate column vector of the m given points,
$\quad \mathbf{X}^T = (\mathbf{X}_1^T, \ldots, \mathbf{X}_m^T)$ with $\mathbf{X}_i^T = (X_i, Y_i, Z_i)$

$\mathbf{X}'$: coordinate column vector of the m corresponding points on curve/surface, $\mathbf{X}'^T = (\mathbf{X}_1'^T, \ldots, \mathbf{X}_m'^T)$ with
$\quad \mathbf{X}_i'^T = (X_i', Y_i', Z_i')$

$\mathbf{d}$: distance column vector, $\mathbf{d} = (d_1, \ldots, d_m)^T$ with
$\quad d_i = \left\| \mathbf{X}_i - \mathbf{X}_i' \right\|$

$\mathbf{P}^T\mathbf{P}$: weighting matrix with a size of $nm \times nm$ for (3) and $m \times m$ for (4).

The definition of the performance indices (3)–(4) supposes neither the type of curve/surface nor how to determine the corresponding points $\{\mathbf{X}_i'\}_{i=1}^m$ on curve/surface to the given points $\{\mathbf{X}_i\}_{i=1}^m$. Thus, based on (3)–(4), we can derive a variety of general algorithms for geometric fitting. We characterize the probable algorithms minimizing the performance indices (3) and (4) as *coordinate-based algorithm* and *distance-based algorithm*, respectively [2]. In addition, in order to minimize the performance indices (3)–(4) with curve/surface fitting, we have to determine not only $\mathbf{a}$ but also $\{\mathbf{X}_i'\}_{i=1}^m$ on curve/surface S. With the *total method* [4, 18], $\mathbf{a}$ and $\{\mathbf{X}_i'\}_{i=1}^m$ are simultaneously determined

$$\min_{\mathbf{a} \in \mathbb{R}^p, \{\mathbf{X}_i'\}_{i=1}^m \in S} \sigma_0^2 \left(\{\mathbf{X}_i'(\mathbf{a})\}_{i=1}^m \right), \tag{5}$$

while they are alternately determined by the *variable-separation method* [2,7,10,11,19] in a nested iteration scheme

$$\min_{\mathbf{a} \in \mathbb{R}^p} \min_{\{\mathbf{X}_i'\}_{i=1}^m \in S} \sigma_0^2 \left(\{\mathbf{X}_i'(\mathbf{a})\}_{i=1}^m \right). \tag{6}$$

There could be four combinations from the two performance indices (3)–(4) and the two iteration schemes (5)–(6) (Table 1). One of the four combinations results in an underdetermined linear equation system for iteration, hence it is unpractical for solving the geometric fitting problems. The classification of Algorithms I–III was addressed first in [3]. The known algorithms for geometric fitting can be classified as Algorithm I [4,18] or II [7,10,11,19], while Algorithm III is a new class of geometric fitting algorithms [2].

Table 1. Algorithms for geometric fitting of curves and surfaces

Algorithmic approach	Distance-based algorithm	Coordinate-based algorithm
Total method	Underdetermined: $\min\limits_{\mathbf{a} \in \mathbb{R}^p, \{\mathbf{X}_i'\}_{i=1}^m \in S} \left\| \mathbf{Pd} \right\|^2$	Algorithm I: $\min\limits_{\mathbf{a} \in \mathbb{R}^p, \{\mathbf{X}_i'\}_{i=1}^m \in S} \left\| \mathbf{P}(\mathbf{X} - \mathbf{X}') \right\|^2$
Variable-separation method	Algorithm II: $\min\limits_{\mathbf{a} \in \mathbb{R}^p} \min\limits_{\{\mathbf{X}_i'\}_{i=1}^m \in S} \left\| \mathbf{Pd} \right\|^2$	Algorithm III: $\min\limits_{\mathbf{a} \in \mathbb{R}^p} \min\limits_{\{\mathbf{X}_i'\}_{i=1}^m \in S} \left\| \mathbf{P}(\mathbf{X} - \mathbf{X}') \right\|^2$

Section 3 describes in detail a robust and efficient algorithm for locating the shortest distance point on parametric curve/surface from a given point, a subproblem of (6) (Algorithms II and III). In this section we briefly describe the two numerical algorithms for minimizing (3) and (4), respectively [2].

The curve/surface parameters $\mathbf{a}$ minimizing the performance index (3) can be estimated by the Gauss-Newton method

$$\mathbf{PJ}\big|_k \Delta\mathbf{a} = \mathbf{P}(\mathbf{X} - \mathbf{X}')\big|_k, \qquad \mathbf{a}_{k+1} = \mathbf{a}_k + \alpha\Delta\mathbf{a}, \quad \text{with} \quad \mathbf{J} = \frac{\partial\mathbf{X}'}{\partial\mathbf{a}}. \qquad (7)$$

The following unfolded form of (7) is helpful for implementing the algorithm:

$$\mathbf{P}\underbrace{\begin{pmatrix} \mathbf{J}_{\mathbf{X}'_1,\mathbf{a}} \\ \vdots \\ \mathbf{J}_{\mathbf{X}'_m,\mathbf{a}} \end{pmatrix}}_{nm \times p} \Delta\mathbf{a} = \mathbf{P}\underbrace{\begin{pmatrix} \mathbf{X}_1 - \mathbf{X}'_1 \\ \vdots \\ \mathbf{X}_m - \mathbf{X}'_m \end{pmatrix}}_{nm \times 1} \qquad (8)$$

with

$$\mathbf{J}_{\mathbf{X}'_i,\mathbf{a}} = \frac{\partial\mathbf{X}'_i}{\partial\mathbf{a}} = \frac{\partial\mathbf{X}}{\partial\mathbf{a}}\bigg|_{\mathbf{X}=\mathbf{X}'_i} = \frac{\partial(\mathbf{R}^{-1}\mathbf{x} + \mathbf{X}_o)}{\partial\mathbf{a}}\bigg|_{\mathbf{u}=\mathbf{u}'_i}.$$

Also, the curve/surface parameters $\mathbf{a}$ minimizing the performance index (4) can be estimated by the Gauss-Newton method

$$\mathbf{PJ}\big|_k \Delta\mathbf{a} = -\mathbf{Pd}\big|_k, \qquad \mathbf{a}_{k+1} = \mathbf{a}_k + \alpha\Delta\mathbf{a}, \qquad \text{with} \qquad \mathbf{J} = \frac{\partial\mathbf{d}}{\partial\mathbf{a}}. \qquad (9)$$

The unfolded form of (9) is

$$\mathbf{P}\underbrace{\begin{pmatrix} \mathbf{J}_{d_1,\mathbf{a}} \\ \vdots \\ \mathbf{J}_{d_m,\mathbf{a}} \end{pmatrix}}_{m \times p} \Delta\mathbf{a} = -\mathbf{P}\underbrace{\begin{pmatrix} d_1 \\ \vdots \\ d_m \end{pmatrix}}_{m \times 1} \qquad (10)$$

with

$$\mathbf{J}_{d_i,\mathbf{a}} = \frac{\partial d_i}{\partial\mathbf{a}} = \frac{\partial\|\mathbf{X}_i - \mathbf{X}'_i\|}{\partial\mathbf{a}} = -\frac{(\mathbf{X}_i - \mathbf{X}'_i)^{\mathrm{T}}}{\|\mathbf{X}_i - \mathbf{X}'_i\|}\frac{\partial\mathbf{X}'_i}{\partial\mathbf{a}}.$$

3. Minimum Distance Point

The time-consuming element of the variable-separation method (6) is the inner iteration finding the location parameters $\{\mathbf{u}'_i\}_{i=1}^m$ of the minimum distance points $\{\mathbf{X}'_i\}_{i=1}^m$ on the current curve/surface from each given point $\{\mathbf{X}_i\}_{i=1}^m$. The resulting values $\{\mathbf{X}'_i\}_{i=1}^m$ and $\{d_i\}_{i=1}^m$ are to be supplied for (8) and (10), respectively.

For a given point $\mathbf{x}_i = \mathbf{R}(\mathbf{X}_i - \mathbf{X}_o)$ in frame xyz, we determine the location parameters $\mathbf{u}'_i$ of the minimum distance point $\mathbf{x}'_i$ on the standard curve/surface (1) defined in xyz. The minimum distance point $\mathbf{X}'_i$ in XYZ from the given point $\mathbf{X}_i$ can be obtained through the

backward transformation of $\mathbf{x}'_i$ into XYZ by using $\mathbf{X}'_i = \mathbf{R}^{-1}\mathbf{x}'_i + \mathbf{X}_o$.

The minimum distance problem of parametric curves/surfaces can be solved by the Newton method [12, 16, 21]. The aim is to find the location parameters $\mathbf{u}$ that minimize the error distance d_i between the given point $\mathbf{x}_i$ and the corresponding point $\mathbf{x}$ on the standard curve/surface (1)

$$\begin{aligned} D(\mathbf{u}) &= d_i^2 \\ &= \|\mathbf{x}_i - \mathbf{x}(\mathbf{a}_g, \mathbf{u})\|^2 \\ &= (\mathbf{x}_i - \mathbf{x}(\mathbf{a}_g, \mathbf{u}))^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}(\mathbf{a}_g, \mathbf{u})). \end{aligned} \qquad (11)$$

The first order necessary condition for a minimum of (11) as a function of $\mathbf{u}$ is

$$\begin{aligned} \mathbf{f}(\mathbf{x}_i, \mathbf{x}(\mathbf{a}_g, \mathbf{u})) &= \frac{1}{2}\begin{pmatrix} \partial D(\mathbf{u})/\partial u \\ \partial D(\mathbf{u})/\partial v \end{pmatrix} \\ &= -\begin{pmatrix} (\mathbf{x}_i - \mathbf{x}(\mathbf{a}_g, \mathbf{u}))^{\mathrm{T}}\mathbf{x}_u \\ (\mathbf{x}_i - \mathbf{x}(\mathbf{a}_g, \mathbf{u}))^{\mathrm{T}}\mathbf{x}_v \end{pmatrix} = \mathbf{0}. \end{aligned} \qquad (12)$$

Equation (12) denotes that the distance vector $(\mathbf{x}_i - \mathbf{x})$ and the tangent vectors $\partial\mathbf{x}/\partial\mathbf{u}$ at $\mathbf{x}$ should be orthogonal. We solve (12) for $\mathbf{u}$ by using the Newton method (how to derive the Hessian matrix $\partial\mathbf{f}/\partial\mathbf{u}$ is shown in Section 4)

$$\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\bigg|_k \Delta\mathbf{u} = -\mathbf{f}(\mathbf{u})\big|_k, \qquad \mathbf{u}_{k+1} = \mathbf{u}_k + \alpha\Delta\mathbf{u}. \qquad (13)$$

4. Geometric Fitting Algorithm

This section describes in detail the Algorithms III (Table 1) for geometric fitting of parametric curves/surfaces. With the implementations, the model parameters $\mathbf{a}$ are grouped and simultaneously estimated in terms of form $\mathbf{a}_g$, position $\mathbf{a}_p$, and rotation parameters $\mathbf{a}_r$ (see (1)–(2)), which is a highly desirable algorithmic feature for applications. Algorithm III is based on the performance index (3) and alternately estimates the parameters $\mathbf{a}$ and $\{\mathbf{u}_i\}_{i=1}^m$ (variable-eparation method (6)). The location parameters $\{\mathbf{u}'_i\}_{i=1}^m$ of $\{\mathbf{X}'_i\}_{i=1}^m$ on the current curve/surface are found by the algorithm described in Section 3 (inner iteration of (6)). In this section, we generally implement the coordinate-based algorithm (8) (outer iteration of (6)) which updates $\mathbf{a}$ by minimizing (3).

From (1)–(2), we derive the Jacobian matrices of each minimum distance point $\{\mathbf{X}'_i\}_{i=1}^m$

$$
\mathbf{J}_{\mathbf{X}_i',\mathbf{a}} = \left. \frac{\partial \mathbf{X}}{\partial \mathbf{a}} \right|_{\mathbf{X}=\mathbf{X}_i'}
$$

$$
= \left. \left(\mathbf{R}^{\mathrm{T}} \left(\frac{\partial \mathbf{x}}{\partial \mathbf{a}} + \frac{\partial \mathbf{x}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{a}} \right) + \frac{\partial \mathbf{R}^{\mathrm{T}}}{\partial \mathbf{a}} \mathbf{x} + \frac{\partial \mathbf{X}_{\mathrm{o}}}{\partial \mathbf{a}} \right) \right|_{\mathbf{u}=\mathbf{u}_i'} \tag{14}
$$

$$
= \underbrace{\mathbf{R}^{\mathrm{T}} \frac{\partial \mathbf{x}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{a}}}_{n \times p} \bigg|_{\mathbf{u}=\mathbf{u}_i'} + \left(\underbrace{\mathbf{R}^{\mathrm{T}} \frac{\partial \mathbf{x}}{\partial \mathbf{a}_{\mathrm{g}}}}_{n \times l} \bigg|_{\mathbf{u}=\mathbf{u}_i'} \bigg| \underbrace{\mathbf{I}}_{n \times n} \bigg| \underbrace{\frac{\partial \mathbf{R}^{\mathrm{T}}}{\partial \mathbf{a}_{\mathrm{r}}} \mathbf{x}_i'}_{n \times s} \right).
$$

The derivative matrix $\partial \mathbf{u}/\partial \mathbf{a}$ at $\mathbf{u} = \mathbf{u}_i'$ in (14) represents the variational behavior of $\mathbf{u}_i'$ relative to the differential changes of $\mathbf{a}$. Purposefully, we extract the information $\partial \mathbf{u}/\partial \mathbf{a}$ from (12) that constrains $\mathbf{u}$ to represent the minimum distance point. Because (12) has an implicit form, its derivatives lead to

$$
\underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{u}}}_{q \times q} \underbrace{\frac{\partial \mathbf{u}}{\partial \mathbf{a}}}_{q \times p} + \underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{x}_i}}_{q \times n} \underbrace{\frac{\partial \mathbf{x}_i}{\partial \mathbf{a}}}_{n \times p} + \underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{a}}}_{q \times p} = \mathbf{0} \qquad \text{with} \qquad q = \begin{cases} 1 \ : \ \text{curve} \\ 2 \ : \ \text{surface,} \end{cases}
$$

thus,

$$
\left. \frac{\partial \mathbf{u}}{\partial \mathbf{a}} \right|_{\mathbf{u}=\mathbf{u}_i'} = -\left. \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^{-1} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}_i} \frac{\partial \mathbf{x}_i}{\partial \mathbf{a}} + \frac{\partial \mathbf{f}}{\partial \mathbf{a}} \right) \right|_{\mathbf{u}=\mathbf{u}_i'}. \tag{15}
$$

The derivative matrix $\partial \mathbf{x}_i/\partial \mathbf{a}$ in (15) represents the coordinate variation of the given point $\mathbf{x}_i$ in frame xyz relative to the variation of $\mathbf{a}$. We derive $\partial \mathbf{x}_i/\partial \mathbf{a}$ from the coordinate transformation equation $\mathbf{x}_i = \mathbf{R}(\mathbf{X}_i - \mathbf{X}_{\mathrm{o}})$ (2),

$$
\frac{\partial \mathbf{x}_i}{\partial \mathbf{a}} = \frac{\partial \mathbf{R}}{\partial \mathbf{a}} (\mathbf{X}_i - \mathbf{X}_{\mathrm{o}}) - \mathbf{R} \frac{\partial \mathbf{X}_{\mathrm{o}}}{\partial \mathbf{a}}
$$

$$
= \left(\mathbf{0} \ \bigg| \ -\mathbf{R} \ \bigg| \ \frac{\partial \mathbf{R}}{\partial \mathbf{a}_{\mathrm{r}}} (\mathbf{X}_i - \mathbf{X}_{\mathrm{o}}) \right).
$$

The variation $\partial \mathbf{x}_i/\partial \mathbf{a}$ is independent of the size or shape parameters $\mathbf{a}_{\mathrm{g}}$ of the model curve/surface. The other three matrices $\partial \mathbf{f}/\partial \mathbf{u}$, $\partial \mathbf{f}/\partial \mathbf{x}_i$, and $\partial \mathbf{f}/\partial \mathbf{a}$ in (13) and (15) can be directly derived from (12):

$$
\frac{\partial \mathbf{f}}{\partial \mathbf{u}} = (\mathbf{x}_u \ \mathbf{x}_v)^{\mathrm{T}} (\mathbf{x}_u \ \mathbf{x}_v) - \begin{pmatrix} (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}} \mathbf{x}_{uu} & (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}} \mathbf{x}_{uv} \\ (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}} \mathbf{x}_{vu} & (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}} \mathbf{x}_{vv} \end{pmatrix},
$$

$$
\frac{\partial \mathbf{f}}{\partial \mathbf{x}_i} = -(\mathbf{x}_u \ \mathbf{x}_v)^{\mathrm{T}}, \quad \frac{\partial \mathbf{f}}{\partial \mathbf{a}} = \begin{pmatrix} \mathbf{x}_u^{\mathrm{T}} \mathbf{G}_0 - (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}} \mathbf{G}_1 \\ \mathbf{x}_v^{\mathrm{T}} \mathbf{G}_0 - (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}} \mathbf{G}_2 \end{pmatrix} \bigg| \ \mathbf{0} \ \bigg| \ \mathbf{0} \end{pmatrix},
$$

with (XHG matrix)

$$
\frac{\partial \mathbf{x}}{\partial \mathbf{u}} = (\mathbf{x}_u \ \mathbf{x}_v), \quad \mathbf{H} = \begin{pmatrix} \mathbf{x}_{uu} & \mathbf{x}_{uv} \\ \mathbf{x}_{vu} & \mathbf{x}_{vv} \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \mathbf{G}_0 \\ \mathbf{G}_1 \\ \mathbf{G}_2 \end{pmatrix} = \frac{\partial}{\partial \mathbf{a}_{\mathrm{g}}} \begin{pmatrix} \mathbf{x} \\ \mathbf{x}_u \\ \mathbf{x}_v \end{pmatrix}. \tag{16}
$$

Now, (15) can be solved for $\partial \mathbf{u}/\partial \mathbf{a}$ at $\mathbf{u} = \mathbf{u}_i'$, then (14) and (8) can be completed and solved for $\Delta \mathbf{a}$.

To note is that only the standard curve/surface description (1) without involvement of the position/rotation parameters is required in (16). All that is necessary for a new curve/surface is to derive the XHG matrix (16) from (1) of the new curve/surface and to supply a proper set of initial parameter values $\mathbf{a}_0$ for (8). The storage space usage and computing cost are proportional to the number of points. In comparison with Algorithms I and II, a disadvantage of Algorithm III is the involvement of the second derivatives $\partial^2 \mathbf{x}/\partial u \partial \mathbf{a}_{\mathrm{g}}$ and $\partial^2 \mathbf{x}/\partial v \partial \mathbf{a}_{\mathrm{g}}$ ($\mathbf{G}_1$ and $\mathbf{G}_2$ in (16)) demanding an extra implementation cost.

5. Fitting Examples

To compare the convergence and computing cost of surface fitting by Algorithms I–III, we fit ellipsoids to 10–200 points emulating the optical 3-D measurement of an ellipsoid (Fig. 1). An ellipsoid is to be defined in space through 9 parameters, i.e. 3 form parameters of a, b, c (half lengths), 3 position parameters of $X_{\mathrm{o}}, Y_{\mathrm{o}}, Z_{\mathrm{o}}$, and 3 rotation parameters of ω, φ, κ. The standard description (1) of an ellipsoid can be defined in xyz frame as below:

$$
\mathbf{x}(a,b,c,\mathbf{u}) = \begin{pmatrix} a \cos u \cos v \\ b \sin u \cos v \\ c \sin v \end{pmatrix}
$$

with

$$
-\pi < u \leq \pi, \quad -\pi/2 \leq v \leq \pi/2.
$$

There is no significant difference of the convergence performance between Algorithms II and III (Fig. 1c). Algorithm I shows poor convergence, as long as the current (initial) parameter values are not close enough to the final estimation values. The computing cost of Algorithm I increases very rapidly with the number of data points. For $m = 160$, the computing cost of Algorithm I is about 1,000 times higher than that of Algorithm III. For $m = 80$, it is 150 times high (Fig. 1d). Algorithm III demands a higher computing cost than Algorithm II, because the number of equation rows of (8) with surface fitting ($3m$, Algorithm III) is three times large relatively to that of (10) (m, Algorithm II). Compared with solving (8) or (10) for $\Delta \mathbf{a}$ in Section 4, the time-consuming element is the determination of $\{\mathbf{u}_i'\}_{i=1}^m$ in Section 3 which is shared by Algorithms II and III.
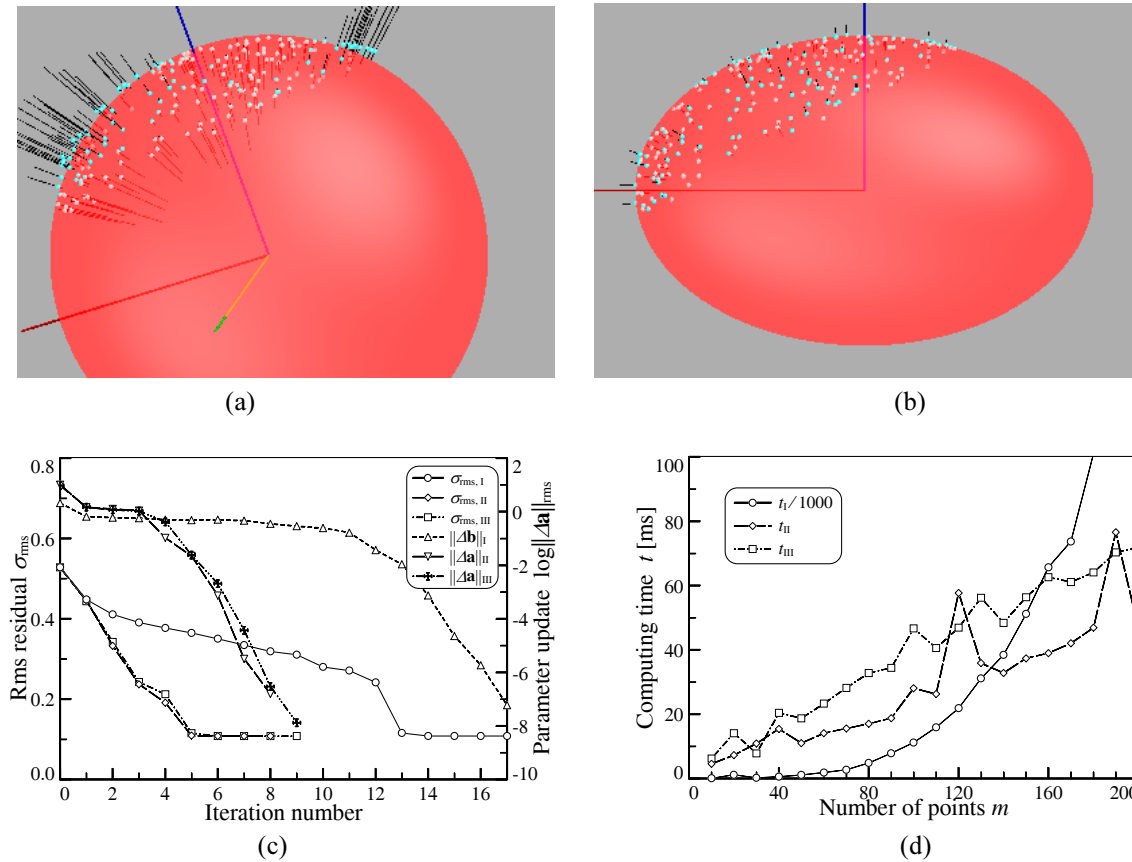
Fig. 1. Ellipsoid fit to 10–200 points randomly distributed on an ellipsoid of $\mathbf{a}_g = (30,25,20)^T$, $\mathbf{a}_p = (10,-10,10)^T$, and $\mathbf{a}_r = (0.3,-0.3,0.3)^T$. The point coordinates are contaminated by Gaussian random errors with an rms distance of 0.1. The initial parameter values with $\mathbf{a}_r = (0,0,0)^T$ are taken from the sphere fittings to each point set: (a) Sphere fit (initial ellipsoid) and (b) ellipsoid fit to 200 points. Distance error bars are elongated ten times. (c) Convergence of Algorithms I–III with ellipsoid fit to 100 points. (d) Computing time of ellipsoid fit to 10–200 points with a Pentium 2.4 GHz PC.

6. Summary

The geometric fitting estimates the parameters of a curve/surface by minimizing the square sum of the shortest distances (*geometric distances*) between the curve/surface and the given points. The geometric fitting is a *nonlinear minimization problem* and has been widely recognized as an analytically and computationally difficult problem. The development of efficient and flexible algorithms for geometric fitting has been eagerly pursued in the last two decades. Due to the applied error measure, the geometric fitting is a substantial mathematical tool for many disciplines of science and engineering, e.g., computer/machine vision, pattern recognition, coordinate metrology, motion analysis, astronomy, particle physics and computer-aided drug design with which coordinate data or dimensional object models are processed.

In this paper, we concisely formulated the square sum of the geometric error distances into two cost functions (3)–(4). Then, there are two numerical methods, the *coordinate-based algorithm* (8) and the *distance-based algorithm* (10) for minimizing the two cost functions (3) and (4), respectively. In addition, there are two strategies for minimizing the cost functions through iteration, the *total method* (5) and the *variable-separation method* (6). From the two numerical methods and the two minimization strategies, we identified the following three realistic algorithmic approaches for generally solving the nonlinear problems of geometric fitting through iteration:

- Algorithm I: combination of the total method with the coordinate-based algorithm
- Algorithm II: combination of the variable-separation method with the distance-based algorithm
- Algorithm III: combination of the variable-separation method with the coordinate-based algorithm.

This paper gives a general implementation of Algorithm III to parametric curves and surfaces. With the implementation,

the curve/surface parameters $\mathbf{a}$ are grouped and simultaneously estimated in terms of form $\mathbf{a}_g$, position $\mathbf{a}_p$, and rotation parameters $\mathbf{a}_r$, which is a helpful algorithmic feature for applications.

With the experimental examples for geometric fitting of ellipsoid, the performances of Algorithms I-III are tested and compared in terms of convergence, computing cost. The efficient and robust algorithm described in Section 3 for locating the minimum distance point is essential to Algorithms I–III. Algorithm I is recommended for a one-time instant application. Algorithm II is advantageous for certain applications as long as 3-D curves are not dealt with. Otherwise, due to its excellent overall performance on convergence and computing cost, Algorithm III is highly recommended for general applications of geometric fitting.

Reference

[1] R.J. Adcock, "Note on the method of least squares, " The Analyst, Des Moines, Iowa, vol.4, pp.183~184, 1877.

[2] S.J. Ahn, W. Rauh, H.S. Cho, H.-J. Warnecke, "Orthogonal Distance Fitting of Implicit Curves and Surfaces," IEEE Trans. Pattern Analy. Mach. Intell, vol.24, no.5, pp.620~638, 2002.

[3] S.J. Ahn, E. Westkämper, W. Rauh, "Orthogonal Distance Fitting of Parametric Curves and Surfaces, " Proc. 4th Int'l Symp. Algorithms for Approximation, J. Levesley et al. (eds.), U.K.: Univ. of Huddersfield, pp.122~129, 2002.

[4] P.T. Boggs, R.H. Byrd, R.B. Schnabel, "A stable and efficient algorithm for nonlinear orthogonal distance regression," SIAM J. Sci. Stat. Compt., vol.8, pp. 1052~1078, 1987.

[5] P.T. Boggs, J.R. Donaldson, R.H. Byrd, R.B. Schnabel, "Algorithm 676 – ODRPACK: Software for Weighted Orthogonal Distance Regression," ACM Trans. Math. Softw., vol.15, no.4, pp.348~364, 1989.

[6] F.L. Bookstein, "Fitting conic sections to scattered data," Comput. Graph. Image Process., vol.9, no.1, pp.56~71, 1979.

[7] B.P. Butler, A.B. Forbes, P.M. Harris, "Algorithms for Geometric Tolerance Assessment," Report no. DITC 228/94, Teddington, U.K.: NPL, 1994.

[8] X. Cao, N. Shrikhande, G. Hu, "Approximate orthogonal distance regression method for fitting quadric surfaces to range data," Pattern Recogn. Lett., vol.15, no.8, pp. 781~796, 1994.

[9] A. Fitzgibbon, M. Pilu, R.B. Fisher, "Direct Least Square Fitting of Ellipses," IEEE Trans. Pattern Analy. Mach. Intell, vol.21, no.5, pp.476~480, 1999.

[10] H.-P. Helfrich and D. Zwick, "A trust region method for implicit orthogonal distance regression," Numer. Algorithms, vol.5, pp.535~545, 1993.

[11] H.-P. Helfrich and D. Zwick, "A trust region algorithm for parametric curve and surface fitting," J. Comput. Appl. Math., vol.73, pp.119~134, 1996.

[12] J. Hoschek, F.-J. Schneider, P. Wassum, "Optimal approximate conversion of spline surfaces", Comput. Aided Geom. Design, vol.6, pp.293~306, 1989.

[13] ISO 10360-6, "Geometrical Product Specifications (GPS) – Acceptance and reverification test for coordinate measuring machines (CMM) – Part 6: Estimation of errors in computing Gaussian associated features," Geneva, Switzerland: ISO, Dec. 2001.

[14] D. Marshall, G. Lukacs, R. Martin, "Robust Segmentation of Primitives from Range Data in the Presence of Geometric Degeneracy," IEEE Trans. Pattern Analy. Mach. Intell, vol.23, no.3, pp.304~314, 2001.

[15] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," The Philos. Mag., ser.6, vol.2, no.11, pp.559~572, 1901.

[16] L. Piegl and W. Tiller, "The NURBS Book," 2nd Ed., Berlin: Springer, 1997.

[17] F. Solina and R. Bajcsy, "Recovery of Parametric Models from Range Images: The Case for Superquadrics with Global Deformations," IEEE Trans. Pattern Analy. Mach. Intell, vol.12, no.2, pp.131~147, 1990.

[18] D. Sourlier, "Three Dimensional Feature Independent Bestfit in Coordinate Metrology," Ph.D. Thesis, no. 11319, Zurich, Switzerland: ETH Zurich, 1995.

[19] S. Sullivan, L. Sandford, J. Ponce, "Using Geometric Distance Fits for 3-D Object Modeling and Recognition," IEEE Trans. Pattern Analy. Mach. Intell, vol.16, no.12, pp.1183~1196, 1994.

[20] G. Taubin, "Estimation of Planar Curves, Surfaces, Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation," IEEE Trans. Pattern Analy. Mach. Intell, vol.13, no. 11, pp.1115~1138, 1991.

[21] D.A. Turner, "The approximation of Cartesian coordinate data by parametric orthogonal distance regression," Ph.D. Thesis, U.K.: Univ. of Huddersfield, 1999.

Sung Joon Ahn

He received the BS degree in Mechanical Design and Production Engineering from the Seoul National University in 1985, and the MS degree in Production Engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 1987. In 2004 he received the Dr.-Ing. degree in Mechanical Engineering from the Stuttgart University, Germany. He worked as a junior research scientist at the research center of LG Electronics from 1987 to 1990. During 1990–2004, he was a member of research scientist at the Fraunhofer Institute for Manufacturing Engineering and Automation (IPA) in Stuttgart, Germany. Since 2005 he has been an Assistant Professor of the School of Information and Communication Engineering, Sungkyunkwan University. His research interests include Pattern Recognition, Optical 3-D Measurement, and 3-D Information Processing.