Constrained 3D Shape Reconstruction Using a Combination of Surface Fitting and Registration

Yang Liu [a,*] Helmut Pottmann [b] Wenping Wang [a]

[a] *Department of Computer Science, The University of Hong Kong*
[b] *Geometric Modeling and Industrial Geometry, Vienna University of Technology*

Abstract

We investigate 3D shape reconstruction from measurement data in the presence of constraints. The constraints may fix the surface type or set geometric relations between parts of an object's surface, such as orthogonality, parallelity and others. It is proposed to use a combination of surface fitting and registration within the geometric optimization framework of squared distance minimization (SDM). In this way, we obtain a quasi-Newton like optimization algorithm, which in each iteration simultaneously registers the data set with a rigid motion to the fitting surface and adapts the shape of the fitting surface. We present examples to show the applicability of our method to constrained 3D shape fitting for reverse engineering of CAD models and to high accuracy fitting with kinematic surfaces, which include surfaces of revolution (reconstructed from fragments of archeological pottery) and spiral surfaces, which are fitted to 3D measurement data of shells. Our optimization algorithm can combine registration of multiple scans of an object and model fitting into a single optimization process which is shown to be superior to the traditional procedure, which first registers the data and then fits a model to it.

Key words: constrained surface fitting, registration, squared distance minimization, digital reconstruction

* corresponding author
  Email addresses: `yliu@cs.hku.hk` (Yang Liu),
`pottmann@geometrie.tuwien.ac.at` (Helmut Pottmann), `wenping@cs.hku.hk`
(Wenping Wang).

1 Introduction

The motivation for the present research comes from reconstruction of objects from 3D scanner data, where special kinematic surfaces (cones, cylinders, general surfaces of revolution, helical surfaces) appear frequently. Many reconstruction algorithms for the more general representatives of these surface classes require estimated surface normals [20,23]. Although these methods are quite efficient when good normal estimates are available, they lack the desired precision if it is difficult to obtain accurate normal estimation or the deviation of the data from the ideal shape model is relatively large; an example is the reconstruction of vessels from archeological findings. Moreover, in these methods the computation of the sweeping motion is separated from the computation of the swept profile, which is a further source of errors.

In the present paper, we extend recent work on improved reconstruction of surfaces of revolution [26] with a more generally applicable concept arising from the geometric optimization framework of *squared distance minimization (SDM)* [18,19,25,27]. Our new method combines the two types of optimization problems that have been solved so far with SDM, namely *curve/surface fitting* and *registration*. This new approach is not only applicable to surfaces of revolution but also to other classes of surfaces and to a number of surface reconstruction problems in reverse engineering in the presence of constraints.

1.1 Previous work

Since the focus of the present work is on *constrained* 3D shape reconstruction, we only review research in this direction. A constraint may fix the surface type: there have been a considerable number of contributions to fitting with special surfaces and thus we refer to [23] for a detailed survey. The existing methods are mainly taken from geometry (Gaussian image, line geometry, kinematical geometry), image processing (methods in extension of the Hough transform) and optimization (non-linear least squares problems). They are also used for surface type recognition (shape filters).

Fitting data with a surface of a given type that is determined with appropriate shape filters, while maintaining constraints between the individual elements of the surface, is a challenging problem [23]. Not only do we need to check the consistency of the constraints, we also need to fit the data simultaneously under these constraints. The work of Benkö et al. [2], Fisher [6], and Karniel et al. [10] can be considered to constitute the state of the art in this area. In the actual fitting part of the problem, most authors use a least squares formulation which embeds the constraints via penalty terms.

2

Our research is based on a combination of registration and fitting, and in this sense closely related to the work on knowledge based image segmentation via a combination of registration and active contours [17,24,15] and to deformable models introduced by Terzopoulos and Fleischer [21]. We also present a new solution for the simultaneous treatment of multiple view registration and model fitting, which extends prior work by Jin et al. [9] and Tubic et al. [22].

1.2 Contributions

Our contributions in this paper are:

- the extension of the SDM method to surface approximation with error measurement orthogonal to the fitting surface;
- the combination of registration and surface fitting within the SDM framework;
- refined algorithms for fitting with kinematic surfaces (rotational, helical and spiral surfaces) plus a demonstration of their efficiency for shape reconstruction from measurement data of archeological pottery, shells and engineering objects;
- a new way of incorporating constraints into 3D surface reconstruction for applications in reverse engineering of CAD models;
- an efficient optimization algorithm which combines multiple view registration and model fitting and in this way achieves higher accuracy than the traditional approach which first registers the data and then fits a model to it.

2 Fundamentals of SDM

Here we summarize a few basic facts about squared distance minimization (SDM). For more details and issues of efficient implementation we refer to [1,18,19,25,27]. Before entering this discussion we would like to point out that many authors have used the distance field [13,14] for registration and fitting; in fact, the concept of the distance field is so closely tied to the problem that it must occur in some way. However, most papers do not use the distance function in the same way as we are doing it: we use local quadratic approximants of the squared distance function and in this way obtain fast local convergence via algorithms of the Newton or quasi-Newton type.

2.1 Squared distance function of a surface

Given a surface $\Phi \subset \mathbb{R}^3$, the squared distance function d^2 assigns to each point $\mathbf{x} \in \mathbb{R}^3$ the square of its shortest distance to Φ. The importance of this function for our algorithms lies in the fact that we want to compute a surface which minimizes the sum of squared distances to the data point cloud. Since several important optimization concepts require second order approximants of the objective function, we need to derive second order approximants of d^2.

Let us fix the notation. We consider a surface Φ with unit normal vector field $\mathbf{n}(\mathbf{s}) = \mathbf{n}_3(\mathbf{s})$, attached to points $\mathbf{s} \in \Phi$. At each point $\mathbf{s}$, we have a local Cartesian frame $(\mathbf{n}_1, \mathbf{n}_2, \mathbf{n})$, where the first two vectors $\mathbf{n}_1, \mathbf{n}_2$ determine the principal curvature directions. We will refer to this local frame as the *principal frame* $\Pi(\mathbf{s})$. Let κ_j be the (signed) principal curvature in the principal curvature direction $\mathbf{n}_j$, $j = 1, 2$, and let $\rho_j = 1/\kappa_j$.

Let $\mathbf{s} \in \Phi$ be the normal foot point of a point $\mathbf{p} \in \mathbb{R}^3$, i.e., $\mathbf{s}$ is the closest point on Φ to $\mathbf{p}$. Expressed in the principal frame at $\mathbf{s}$ the second order Taylor approximant F_d of the function d^2 at a point $\mathbf{x} \in \mathbb{R}^3$ in a neighborhood of $\mathbf{p}$ is

$$F_d(\mathbf{x}) = \frac{d}{d - \rho_1}[\mathbf{n}_1 \cdot (\mathbf{x} - \mathbf{s})]^2 + \frac{d}{d - \rho_2}[\mathbf{n}_2 \cdot (\mathbf{x} - \mathbf{s})]^2 + [\mathbf{n}_3 \cdot (\mathbf{x} - \mathbf{s})]^2. \tag{1}$$

Here, $[\mathbf{n}_j \cdot (\mathbf{x} - \mathbf{s})]^2$, $j = 1, 2, 3$, are the squared distances of $\mathbf{x}$ to the principal planes and tangent plane at $\mathbf{s}$, respectively.

In the important special case of $d = 0$ (i.e., $\mathbf{p} = \mathbf{s}$), the approximant F_d equals the squared distance function to the tangent plane of Φ at $\mathbf{s}$. Thus, if $\mathbf{p}$ is close to Φ, the squared distance function to the tangent plane at $\mathbf{p}$'s closest point on Φ is a good approximant of d^2.

In a Newton-like iteration it is important to employ nonnegative quadratic approximants; we obtain them by removing from the expression of $F_d(\mathbf{x})$ in (1) those terms with a negative coefficient $d/(d - \rho_j)$; see [25].

2.2 Registration using SDM

A set of points $X^0 = (\mathbf{x}_1^0, \mathbf{x}_2^0, \ldots) \subset \mathbb{R}^3$ is given in some coordinate system Σ^0. It will be rigidly moved (i.e., registered) to be in best alignment with a given surface Φ, represented in another system Σ. We view Σ^0 and Σ as a moving system and a fixed system, respectively. A position of X^0 in Σ is denoted by $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots)$. It is the image of X^0 under some rigid body motion α. Since

we identify positions with motions and the motions have to act on the same initial position, we write $X = \alpha(X^0)$, or $\mathbf{x}_i = \alpha(\mathbf{x}_i^0)$.

The *registration problem* is formulated in a least squares sense [3,5]: Compute a rigid body transformation α^*, which minimizes the sum of squared distances of $\alpha(\mathbf{x}_i^0)$ to Φ,

$$F(\alpha) = \sum_i d^2(\alpha(\mathbf{x}_i^0), \Phi). \qquad (2)$$

Starting from an appropriate initial position α^0, SDM performs a Newton-like iteration to minimize F [19]. We describe here a single iteration of the algorithm: Since F is the sum of squared distances of the data points $\mathbf{x}_i$ to the model shape Φ, a quadratic approximant is

$$G = \sum_i F_{d,i}, \qquad (3)$$

where the $F_{d,i}$ are the second order approximants of the squared distance functions of $\mathbf{x}_i$ to the model shape. These approximants have been described in Section 2.1. Then, by equation (1), a second order Taylor approximant of the squared distance function at $\mathbf{x}_i$ is written in the form

$$F_{d,i}(\mathbf{x}) = \sum_{j=1}^{3} \alpha_{i,j} [\mathbf{n}_{i,j} \cdot (\mathbf{x} - \mathbf{s}_i)]^2, \qquad (4)$$

where $\mathbf{n}_{i,j} \cdot (\mathbf{x} - \mathbf{s}_i) = 0$, $j = 1, 2, 3$, denote the coordinate planes of the principal frame at the foot point $\mathbf{s}_i \in \Phi$ of the point $\mathbf{x}_i^0$, and the $\alpha_{i,j}$ can readily be read from equation (1). The same form holds for a nonnegative modification, i.e., terms with negative coefficients will be discarded. One now approximates the displacement of the data point $\mathbf{x}_i$ up to the first order by,

$$\mathbf{x}_i' = \mathbf{x}_i^0 + \overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i^0, \qquad (5)$$

where $\overline{\mathbf{c}} = (\bar{c}_1, \bar{c}_2, \bar{c}_3)$ and $\mathbf{c} = (c_1, c_2, c_3)$ represent the translational and rotational components of a velocity field.

Plugging $\mathbf{x}_i'$ into G in equation (3) gives a local quadratic model of the objective function,

$$F_2(\mathbf{c}, \overline{\mathbf{c}}) = \sum_i \sum_{j=1}^{3} \alpha_{i,j} [\mathbf{n}_{i,j} \cdot (\mathbf{x}_i^0 + \overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i^0 - \mathbf{s}_i)]^2.$$

Since $\mathbf{n}_{i,j} \cdot (\mathbf{x}_i^0 - \mathbf{s}_i)$ is the distance of $\mathbf{x}_i^0$ to the j-th coordinate plane of the principal frame, it vanishes for $j = 1, 2$; and it equals the oriented distance d_i of $\mathbf{x}_i^0$ to the model surface Φ for $j = 3$. Therefore we may rewrite F_2 as

$$F_2(\mathbf{c}, \overline{\mathbf{c}}) = \sum_i \sum_{j=1}^{2} \alpha_{i,j} [\mathbf{n}_{i,j} \cdot (\overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i^0)]^2 + \tilde{F}_2(\mathbf{c}, \overline{\mathbf{c}}). \qquad (6)$$

Here, $\tilde{F}_2$ denotes the part arising from the squared distances to the tangent planes at the foot points, given by

$$\tilde{F}_2(\mathbf{c}, \overline{\mathbf{c}}) = \sum_i [\mathbf{n}_i \cdot (\overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i^0) + d_i]^2, \tag{7}$$

where $\mathbf{n}_i = \mathbf{n}_{i,3}$. Since F_2 is a quadratic function in $(\mathbf{c}, \overline{\mathbf{c}})$, the unique minimizer $(\mathbf{c}', \overline{\mathbf{c}}')$ can be given explicitly by solving a system of linear equations.

Remark 1: In the above application of SDM we measure the squared distance errors from the moving points $\mathbf{x}_i$ orthogonal to the fixed model surface Φ. The moving points $\mathbf{x}_i$ are functions of the motion parameters $(\mathbf{c}, \overline{\mathbf{c}})$ to be optimized.

So far we have estimated the displacement of the data point cloud with help of the velocity field $(\mathbf{c}, \overline{\mathbf{c}})$. We now apply an appropriate helical motion which is determined by this velocity field: The Plücker coordinates $(\mathbf{g}, \overline{\mathbf{g}})$ of the axis, the rotational angle ϕ and the pitch p of the helical motion (including special cases) are given by

$$p = (\mathbf{c} \cdot \overline{\mathbf{c}})/(\mathbf{c})^2, \ \ \phi = \|\mathbf{c}\|, \ \ (\mathbf{g}, \overline{\mathbf{g}}) = (\mathbf{c}, \overline{\mathbf{c}} - p\mathbf{c}). \tag{8}$$

Recall that the *Plücker coordinates* of a line G consist of a direction vector $\mathbf{g}$ and the moment vector $\overline{\mathbf{g}} = \mathbf{p} \times \mathbf{g}$, where $\mathbf{p}$ represents an arbitrary point on G. Altogether, the desired motion is the superposition of a rotation about some axis A through an angle of $\phi = \|\mathbf{c}\|$ and a translation parallel to A by the distance of $p \cdot \phi$. For the explicit formulae we refer to the literature [20].

2.3   SDM for B-spline surface fitting

In this subsection, we describe the basic idea of B-spline curve fitting according to [25]. Different from the SDM introduced in Section 2.1 and 2.2 (ref. Remark 1, Section 2.2), this method measures the fitting error orthogonal to a moving fitting surface. But, since the fitting error is also given by a quadratic approximation of the squared distance to the fitting surface, we shall also refer to the method as *squared distance minimization*, or *SDM*. Although the method has been presented in [25] for B-spline curves only, its generalization to fitting a B-spline surface to a point cloud $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ is straightforward and outlined below.

The main steps are as follows.

(1) Specify a proper initial shape of a B-spline fitting surface.
(2) Compute squared-distance error terms for all data points to obtain a local quadratic model of the objective function.

(3) Solve a linear system of equations to optimize the local quadratic model to obtain an updated spline surface.

(4) Repeat steps 2 and 3 until convergence, e.g., until a pre-specified error threshold is satisfied or the incremental change of the control points falls below a preset threshold.

We explain below briefly steps 2 and 3.

Step 2. An error term is associated with each data point $\mathbf{x}_i$. The error term to be used in the next iteration is found as follows. Compute a (nonnegative) second order approximant of the squared distance function from $\mathbf{x}_i$ to the current instance of the fitting surface Φ; see equation (4). Let $\mathbf{s}_i = \mathbf{s}_c(u_i, v_i)$ be the closest point on the fitting surface to $\mathbf{x}_i$. Then the error term for $\mathbf{x}_i$ is

$$E_{i,c} = \sum_{j=1}^{3} \alpha_{i,j}[\mathbf{n}_{i,j} \cdot (\mathbf{x}_i - \mathbf{s}_c(u_i, v_i))]^2. \tag{9}$$

When we update the surface Φ to $\mathbf{s}_+(u, v)$ with the new control points, the surface point $\mathbf{s}_+(u_i, v_i)$ given by the same parameters (u_i, v_i) is, in general, no longer the foot point of $\mathbf{x}_i$; moreover, the normal vectors $\mathbf{n}_{i,j}$ and curvature radii $\rho_{i,j}$ will have changed there. However, when the model surface is updated by a small change of the control points, we still use equation (9) to estimate the new fitting error at $\mathbf{x}_i$, by

$$E_{i,+}(\mathsf{D}) = \sum_{j=1}^{3} \alpha_{i,j}[\mathbf{n}_{i,j} \cdot (\mathbf{x}_i - \mathbf{s}_+(u_i, v_i))]^2, \tag{10}$$

where the variables are the control points $\mathsf{D} := (\mathbf{d}_1, \ldots, \mathbf{d}_m)$ of the B-spline fitting surface, in the expressions of the updated surface points $\mathbf{s}_+(u_i, v_i)$; we use $E_{i,+}(\mathsf{D})$ to emphasize the dependence of the error term on the control points D. It has been shown in [25] that this simplification yields a quasi-Newton method for optimization, which is not of a standard type (such as BFGS [11]), but provides a very good trade-off between computational simplicity and fast convergence.

Step 3. We use a B-spline surface, whose representation of the form $\mathbf{s}(u, v) = \sum_k B_k(u, v)\mathbf{d}_k$ is linear in the control points. Substituting this form for $\mathbf{s}_+(u_i, v_i)$ in the objective function yields

$$F(\mathsf{D}) = \sum_{i=1}^{N} E_{i,+}(\mathsf{D}) + F_s(\mathsf{D})$$

$$= F_s(\mathsf{D}) + \sum_{i=1}^{N}\sum_{j=1}^{3} \alpha_{i,j}[\mathbf{n}_{i,j} \cdot (\mathbf{x}_i - \sum_k B_k(u_i, v_i)\mathbf{d}_k)]^2. \tag{11}$$

Here, F_s is a smoothing term, assumed to be quadratic in D. The part coming

from the sum of squared distances is also quadratic in the unknown control points D. Therefore the minimization of F requires only the solution of a linear system.

2.4 TDM and PDM

When setting $\alpha_{i,1} = \alpha_{i,2} = 0$ and $\alpha_{i,3} = 1$ in Eqn. (4) (or Eqn. (10)), we obtain the *tangent distance minimization* or *TDM*, since $F_{d,i}(\mathbf{x}) = [\mathbf{n}_i(\mathbf{x} - \mathbf{s}_i)]^2$ which measures the squared distance from $\mathbf{x}$ to the tangent plane at $\mathbf{s}_i$.

When $\alpha_{i,1} = \alpha_{i,2} = \alpha_{i,3} = 1$ in Eqn. (4) (or Eqn. (10)), we obtain $F_{d,i}(\mathbf{x}) = \|\mathbf{x} - \mathbf{s}_i\|^2$, which measures the distance between the two points $\mathbf{x}$ and $\mathbf{s}_i$; hence, the resulting minimization scheme is called the *point distance minimization* or *PDM*. A detailed discussion of these two minimization methods and their comparison with SDM can be found in [25].

SDM is simplified to the *TDM* method if we approximate the function d^2 at a point $\mathbf{x}_i$ by the squared distance to the tangent plane at the foot point $\mathbf{y}_i$. For registration, a method similar to TDM has been first proposed by Chen and Medioni [5] and is known to be superior to the standard ICP [3]. For B-spline curve fitting, the TDM method has been described by Blake and Isard [4]. It is known [19,25] that TDM corresponds to a Gauss-Newton iteration. Thus, it exhibits quadratic convergence for a zero residual problem and a good initial position. However, this is not a practical assumption and thus one should enhance its stability by applying step-size control, e.g., using the Armijo rule or the Levenberg-Marquardt (L-M) regularization [11].

The standard ICP algorithm [3] approximates d^2 at $\mathbf{x}_i$ by the squared distance to the foot point $\mathbf{s}_i$, i.e., the error term for $\mathbf{x}_i$ is defined in PDM as $\|\mathbf{x}_i - \mathbf{s}_+(u_i, v_i)\|^2$ (ref. Eqn. (10)). This *PDM* method is frequently used for freeform surface fitting [23], exhibits only linear convergence and is prone to be trapped in a poor local minimizer; see, e.g., [25].

3 Combination of Surface Fitting and Registration

Before entering the general discussion, let us explain the main idea with the following example: We want to fit a surface of revolution to a set of data points.

The standard solution to this problem uses estimated surface normals at the data points and line geometry to compute the rotational axis [20]. The axis is then kept fixed and an appropriate generatrix is computed to obtain the

final surface. This method works very well for finding a good initial guess of the axis, but has the disadvantage that the error in the axis estimate, which arises from the normal estimates, cannot be further reduced in the subsequent computation of the generatrix. We present here the following idea.

After an initial guess of the axis A has been found, use a coordinate system in which A is a coordinate axis, say the x_3-axis. A surface of revolution with this axis takes a very simple form. Then, we use SDM optimization to simultaneously update the fitting surface (i.e., control points of its generatrix) and move the set of data points (as a single rigid body system) until the fitting error is minimized. This is carried out by combining the techniques in Section 2.2. and Section 2.3. Moving the data point cloud is a registration process and equivalent to changing the axis. However, we register the data point cloud to a changing surface rather than a fixed one.

More specifically, let us explain the registration of a point cloud to a changing surface $\mathbf{s}$. In each iteration, for each data point $\mathbf{x}_i^0$, we compute its closest point $\mathbf{s}_{i,c} \equiv \mathbf{s}_c(u_i, v_i)$ on the current instance of the fitting surface $\Phi : \mathbf{s}(u, v)$, and set up the SD error term,

$$E_{i,c} = \sum_{j=1}^{3} \alpha_{i,j} [\mathbf{n}_{i,j} \cdot (\mathbf{x}_i - \mathbf{s}_{i,c})]^2.$$

The error after a small displacement of the data point set and a change of the surface is estimated as follows. We use a linearization for the displacement of the data point set, i.e., $\mathbf{x}_i$ will be approximated by $\mathbf{x}' = \mathbf{x}_i^0 + \overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i^0$; the change of the control points updates the model surface to $\mathbf{s}_+(u, v)$; therefore, the surface points $\mathbf{s}_{i,c}$ will be replaced by $\mathbf{s}_{i,+} \equiv \mathbf{s}_+(u_i, v_i)$. Then, the new error term, as an approximation to the squared distance from $\mathbf{x}_i$ and Φ, is

$$E_{i,+} = \sum_{j=1}^{3} \alpha_{i,j} [\mathbf{n}_{i,j} \cdot (\mathbf{x}_i^0 + \overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i^0 - \mathbf{s}_{i,+})]^2, \tag{12}$$

where the variables are the motion parameters $(\mathbf{c}, \overline{\mathbf{c}})$ and the control points D. A sum of these error terms, together with a quadratic fairness term F_s,

$$F = \sum_{i=1}^{N} \sum_{j=1}^{3} \alpha_{i,j} [\mathbf{n}_{i,j} \cdot (\mathbf{x}_i^0 + \overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i^0 - \mathbf{s}_{i,+})]^2 + F_s(\mathsf{D}), \tag{13}$$

gives rise to the minimization of a quadratic function in each iteration. Since the surface points $\mathbf{s}_{i,+}$ depend linearly on the unknown shape parameters, i.e., control points, F is quadratic in those unknowns as well as $(\mathbf{c}, \overline{\mathbf{c}})$, assuming that F_s is also a quadratic function of D. Of course, the data point cloud is updated with an appropriate helical motion as for pure registration. We note that this method is applicable even when shape parameters are not linear variables, such as the weights in a rational B-spline surface – one just needs

9

to apply a linearization (see Section 4.2), like using $(\mathbf{c}, \overline{\mathbf{c}})$ to approximate a rigid motion.

Due to misalignment, multiple review registration for a 3D object usually introduces errors not present in the measurement data. These errors would affect the subsequent surface fitting errors, thus the precision of the final reconstructed CAD model of the 3D object.

Within the present setting, we may use an initially registered point cloud for the initial steps in the constrained fitting procedure. In later steps, however, we can allow *a different motion for each of the K scans* that have been used. In this way we hope to remove inaccuracies resulting from the initial registration (see [9,22]). Given K point clouds $X_k = \{\mathbf{x}_{k,i}^0, i = 1, \ldots, N_k\}, k = 1, \ldots, K$, which represent the individual scans (views), our SDM objective function is reformulated as

$$ F = \sum_{k=1}^{K} w_k \sum_{i=1}^{N_k} \sum_{j=1}^{3} \alpha_{k,i,j} [\mathbf{n}_{k,i,j} \cdot (\mathbf{x}_{k,i}^0 + \overline{\mathbf{c}}_k + \mathbf{c}_k \times \mathbf{x}_{k,i}^0 - \mathbf{s}_{k,i,+})]^2 + F_s(\mathsf{D}), \quad (14) $$

where w_k is a weight for the points in each scan (in practice, we can choose $w_k = 1/N_k$); $\alpha_{k,i,j}$ and $\mathbf{n}_{k,i,j}$ have the same meaning as in (13); $(\mathbf{c}_k, \overline{\mathbf{c}}_k)$ is the velocity field of X_k. We will show some examples in Section 4.3.2.

The new version of the SDM method — a combination of fitting and registration — is as simple as the previously discussed cases of pure surface fitting or registration. Its performance from the viewpoint of optimization is comparable to that of pure fitting [25]. As in [25], TDM with step size control also has good performance.

4 Applications

In this section we present three applications of our combined framework of registration and fitting to: 1) surface reconstruction from archeological pottery; 2) shell shape model verification; and 3) constrained CAD model reconstruction in reverse engineering. In all these applications we need to fit a surface of a special type to a given set of 3D scanned data points. For the convergence analysis we use the average error defined by the following *root mean squared error* :

$$ Ave\_Error = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{s}_{i,c}\|^2} \quad (15) $$

The data set is first normalized by uniform scaling to fit in the unit box $[0,1]^3$, to make the optimization parameters independent of model dimensions.
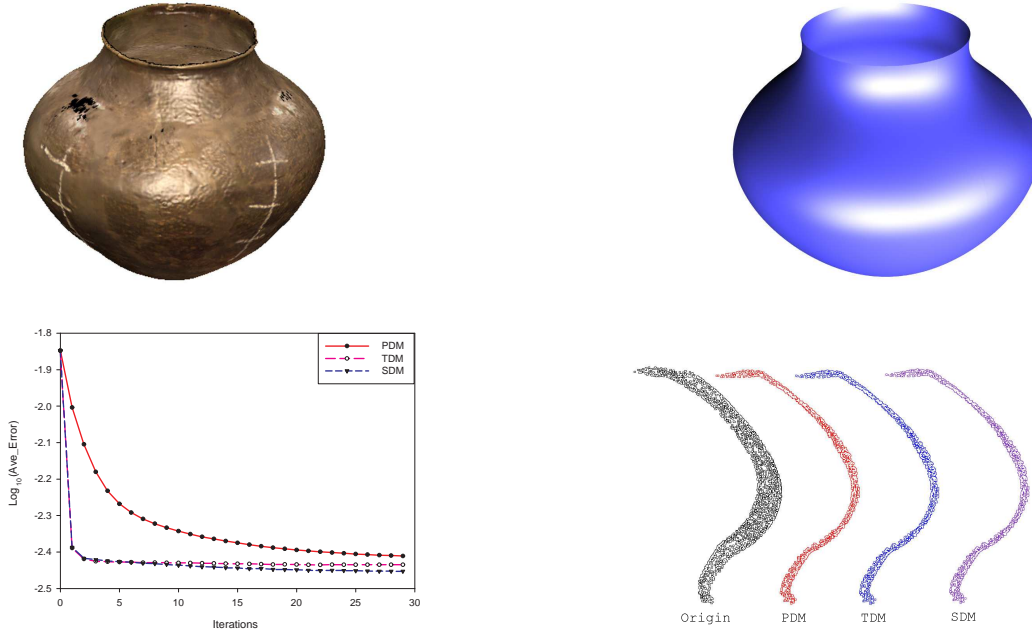
10

Fig. 1. Approximation of an archeological finding by a rotational surface: (top left) original model; (top right) surface reconstructed with SDM method; (bottom left) average errors versus the number of iterations. Note that SDM and TDM are nearly the same, and both better than PDM; (bottom right) data points rotated into a profile plane. From left to right: initial guess and results after optimization with PDM, TDM and SDM. (# of points: 2,260)

4.1 Surfaces of revolution

In order to fit a surface of revolution to a set of data points, we use the well-known line geometric method to compute an initial guess [20]. The axis is then used as x_3-axis of the coordinate system. For the generatrix we take a B-spline curve, $(r(u), z(u)) = \sum_k (r_k, z_k) B_k(u)$ with control points $\mathbf{p}_k = (r_k, z_k)$. Then the surface is

$$\mathbf{x}(u, v) = \sum_k (r_k \cos v, r_k \sin v, z_k) B_k(u) + (0, 0, pv).$$

Here p is the pitch of the helical surface; we have $p = 0$ for a surface of revolution. The essential parameters $\mathbf{p}_k = (r_k, z_k)$ and p appear linearly, therefore the SDM method from Section 3 can be applied. For pure surface fitting, according to Section 2.3, the parameters (u_i, v_i) of the closest point $\mathbf{s}_{i,c} = \mathbf{s}_c(u_i, v_i)$ to $\mathbf{x}_i$ are kept unchanged when we move to $\mathbf{s}_{i,+} = \mathbf{s}_+(u_i, v_i)$.

The registration part does not require the full motion group. We exclude translations parallel to the x_3–axis by setting $\overline{\mathbf{c}} = (\overline{c}_1, \overline{c}_2, 0)$. Moreover, rotations about the x_3–axis are excluded by setting $\mathbf{c} = (c_1, c_2, 0)$. This is done for both surfaces of revolution and helical surfaces. In the latter case, eventually, neces-
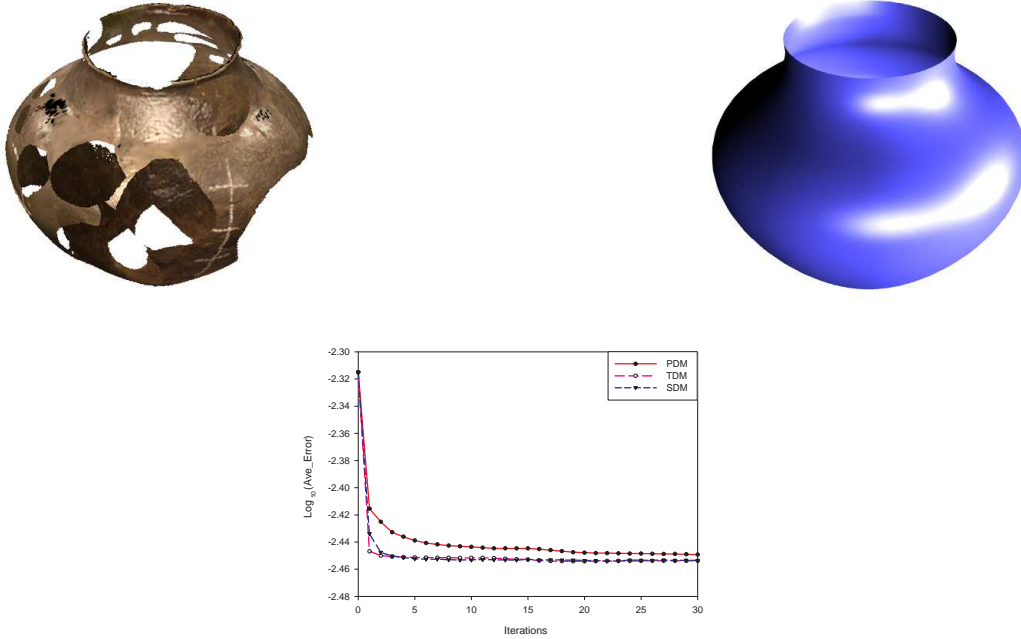
11

Fig. 2. Approximation of a broken archeological finding by a rotational surface: (top left) original model; (top right) surface reconstructed with SDM method; (bottom) average errors versus the number of iterations. SDM and TDM are nearly the same, and both better than PDM. (# of points: 2,863)

sary translations in axis direction or rotations about the axis can be handled via a translation of the profile curve parallel to the axis.

Figure 1 shows an example of 2,260 data points of a scanned pot. The profile curve is a cubic B-spline curve with 7 control points and uniform knots. The significant improvement of the rotation axis is illustrated by the data points rotated into the profile plane: after optimization this cloud is much thinner than the one obtained in the initial fit via line geometry. The results of using SDM, TDM, and PDM are shown. Here and for the examples below, the stability of SDM is enhanced by applying a regularization similar to the L-M method, a trust-region based regularization conventionally applied to the Gauss-Newton method. It is observed that for this example SDM and TDM have nearly the same convergence behavior, while PDM is much slower.

4.2 Spiral surfaces

A spiral surface is generated by a curve undergoing a spiral motion (one-parameter subgroup of similarities in $\mathbb{R}^3$), which is the composition of a rotation about a spiral axis A and an exponential scaling from the so-called spiral center $C \in A$. Placing C at the origin and setting A to be the x_3-axis, a spiral

surface with a B-spline profile can be represented as

$$\mathbf{x}(u, v) = e^{pv} \sum_k [r_k \cos v, r_k \sin v, z_k] B_k(u). \tag{16}$$

Spiral surfaces are frequently taken as models for shapes of shells [12]. We would like to test the precision of this mathematical model using our new optimization method. Very recently, we devised a method which is in some sense analogous to the line geometric approach and can estimate the spiral axis and center from a set of data points and estimated normals, under the assumption that the data points are close to some spiral surfaces [8]. Using this method to provide an initial fit, we now present an improved spiral fitting algorithm based on SDM.
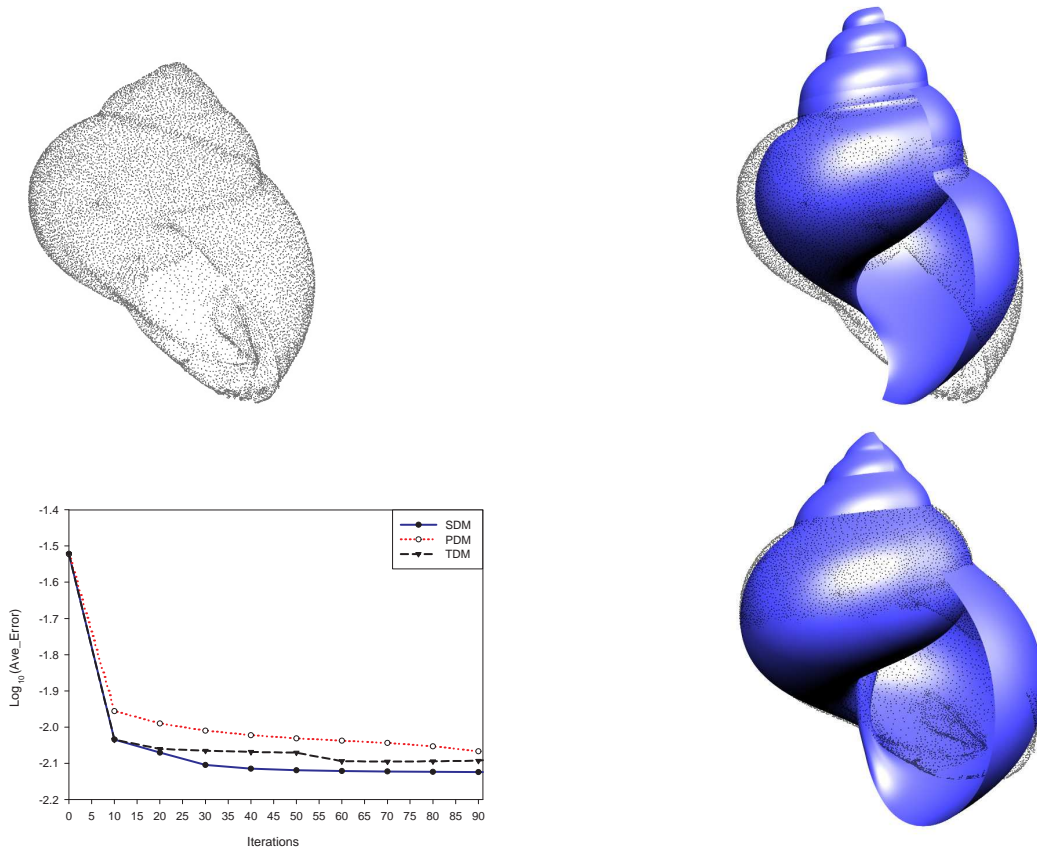
Fig. 3. Shell *helix pomata*: (top left) data points of a shell; (top right) initial fit; (bottom left) average error v.s. the number of iterations. Note that SDM performs better than PDM and TDM; (bottom right) fit computed by SDM. (# of points: 13666)

We note that $\mathbf{x}(u, v)$ (16) is no longer linear in the unknowns p (spiral parameter) and $\mathbf{p}_k = (r_k, z_k)$. Hence, we use the following first order Taylor approximant at the current values p_c and $\mathbf{p}_{k,c}$ of the unknowns,

13

$$\mathbf{x}(u, v) = e^{p_c v} \sum_k [r_k \cos v, r_k \sin v, z_k] B_k(u) +$$

$$+ v(p - p_c) e^{p_c v} \sum_k [r_{k,c} \cos v, r_{k,c} \sin v, z_{k,c}] B_k(u).$$

Keeping the parameter values (u_i, v_i) of the foot points $\mathbf{s}_{i,c}$, we arrive at new points $\mathbf{s}_{i,+}$, which depend linearly on the unknown parameters. Therefore, SDM from Section 3 works again.

Figure 3 shows that an initial fit can be improved significantly by our optimization algorithm, although for this example the residual fitting error is still rather large. For this example, we see that PDM and TDM do not converge as fast as SDM. Since the optimized fit is not ideal, we are inclined to conclude that the spiral surface is not an accurate model for this type of shell.

4.3 Constrained 3D shape reconstruction

4.3.1 Constrained fitting to a single set of data points

In the applications described above, it is an advantage to represent a fitting surface in an *adapted coordinate system* Σ; for example, we put the rotational axis into a coordinate axis or the spiral center into the origin. Choosing an adapted coordinate system may also be possible for a typical engineering object: important elements such as rotational axes, planar faces, etc. can be brought into a special position with respect to Σ. Using an adapted coordinate system, we can set up a parametric model (see Figures 4,8,10). Varying the parameters gives a family of models all of which satisfy the constraints. Thus, our viewpoint leads to an unconstrained optimization problem for the parameters of the model.

Identifying such a coordinate system Σ for building the parametric model is made feasible either with some prior knowledge about the model or by user interaction. Within the general SDM procedure described in Section 3, we now adapt the model shape parameters (if necessary, using linearization as in Section 4.2) and update the position of the data set using a rigid motion with respect to the model shape.

We use an example to illustrate these steps. Figure 4 shows 33,981 measured data points of a machine part, and the parametric CAD model of the part and three side views, with constraints and model parameters indicated. Figure 5 shows the initial fit, final fit and error curves. The constraints of the model are preserved strictly and only the values of the model parameters are updated in each iteration. Here and later for the examples in Section 4.3, only the TDM method is used, since many faces of the CAD model are planar, i.e., having

zero curvature, thus making SDM reduce to TDM at these planar locations. The variations of all the model parameters are shown in Figure 6.

For evaluation, using the same data and CAD model in Figure 4, we compare our combined approach with an approach that optimizes position and surface shape alternatively [9]; (the latter is therefore called the *alternating method*). TDM is used in both position and shape updates in the alternating method. One position update and its successive surface shape update are counted as one iteration of the alternating method. The two error curves are shown in Figure 5 (bottom right), from which it is clear that the combined approach is more efficient. A theoretical explanation is that in the alternating method the iterate follows a zig-zag path in the subspace of motion parameters and the subspace of shape parameters, thus making the alternating method have only linear convergence. On the other hand, the combined method using TDM behaves similarly to the Gauss-Newton method, and therefore can have near quadratic convergence for small-residual problems.
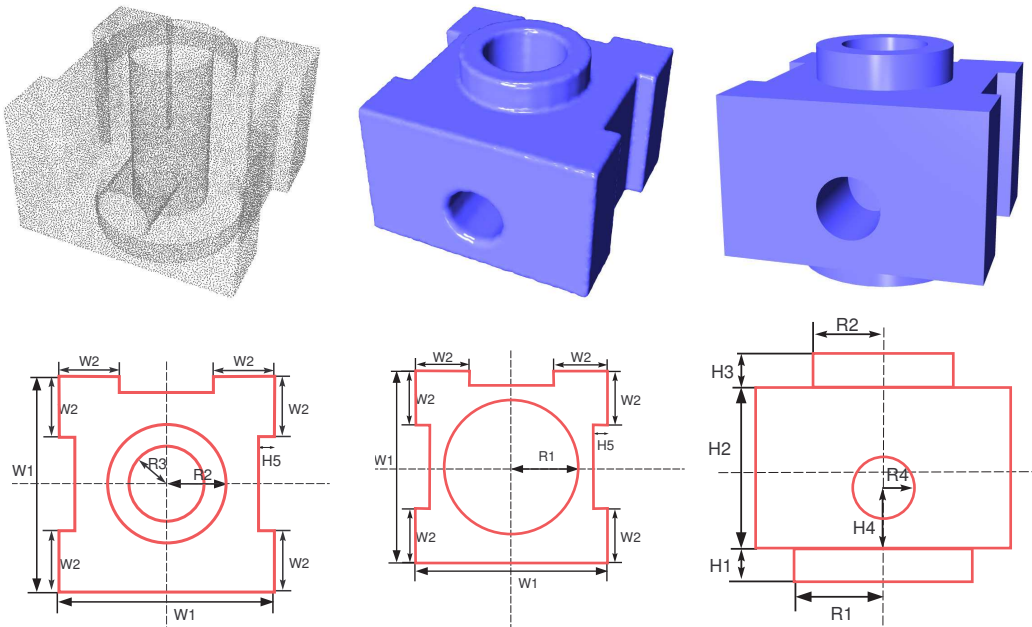
Fig. 4. (top left): A data set of 33,981 points from a machine part; (top middle): its triangulated surface; (top right): a parametric model; (bottom left): top view; (bottom middle): bottom view; (bottom right): front view.

4.3.2 Constrained fitting to multiple views

In this section we provide an experimental validation of our conjecture that relaxing the initial registration by allowing different motions for the individual scans can reduce the overall error. As seen from Figures 8 and 10, the
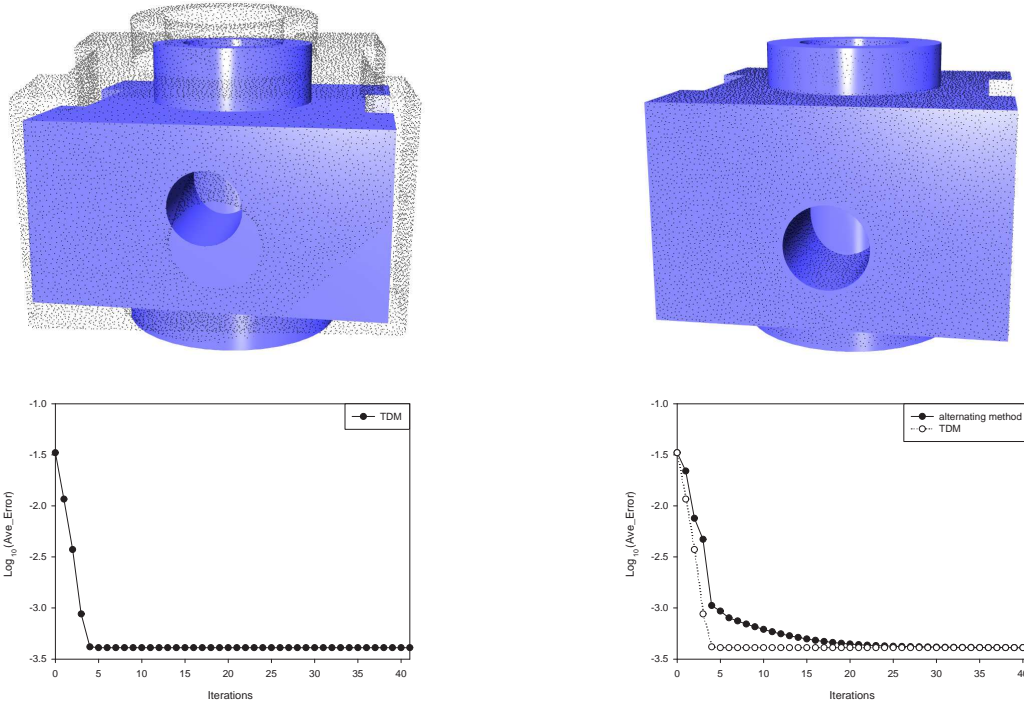
15

Fig. 5. (top left): Initial fit; (top right): optimized fit after 6 iterations of TDM; (bottom left): average error of our method v.s. the number of iterations; (bottom right): comparison with the alternating method.
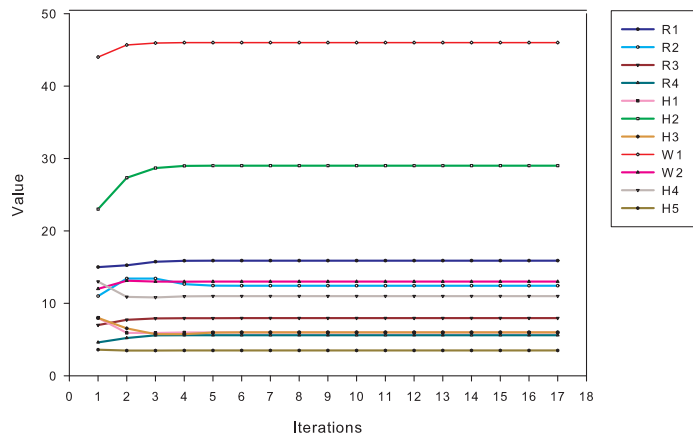

Fig. 6. Variations of the parameters for the machine part model in Figure 4.

combination of fitting with multiple-view registration leads to higher accuracy than first performing the registration on the point cloud data and then fitting a model to the registered data. We consider it an important feature of our algorithm that multiple-view registration can so easily be combined with fitting.
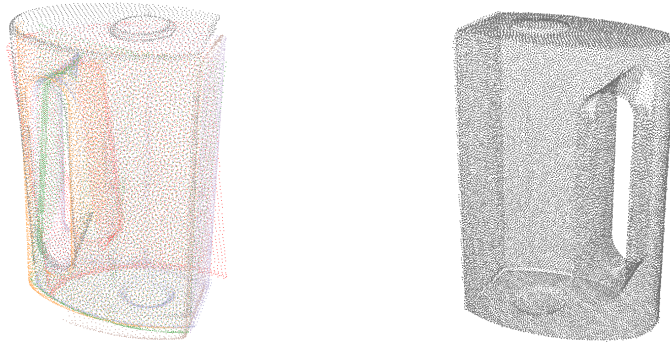
Fig. 7. left: A data set of 331,150 points from a 3D model using 7 scans; right: the registered point set
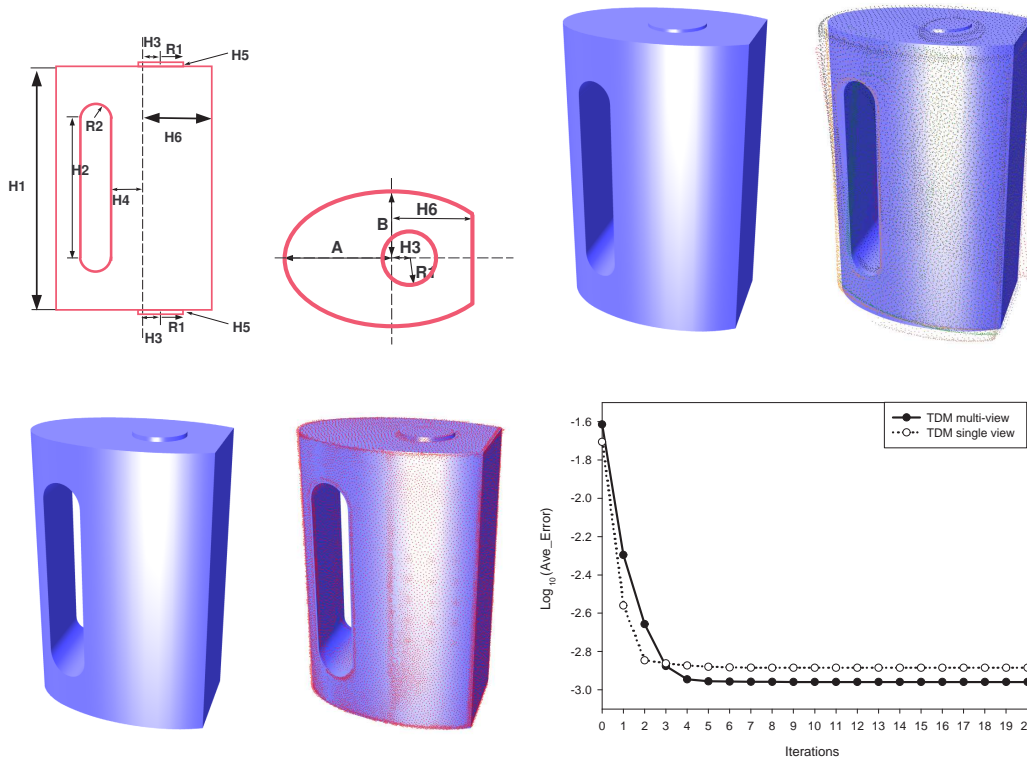

Fig. 8. (top left): front and bottom views of a parametric model; (top right): initial model and initial fit; (bottom left): optimized fit after 8 iterations of TDM; (bottom right): average error v.s. the number of iterations, and comparison with the single view case. We see that combining fitting and multiple-view registration lead to higher accuracy.

4.4   Remarks on the implementation

In this section we provide a few details on the actual implementation.

Closest points. Our algorithm requires to find for every data point $\mathbf{x}$ its closest point (foot-point) $\mathbf{s}$ on the initial surface. If there are too many points, this
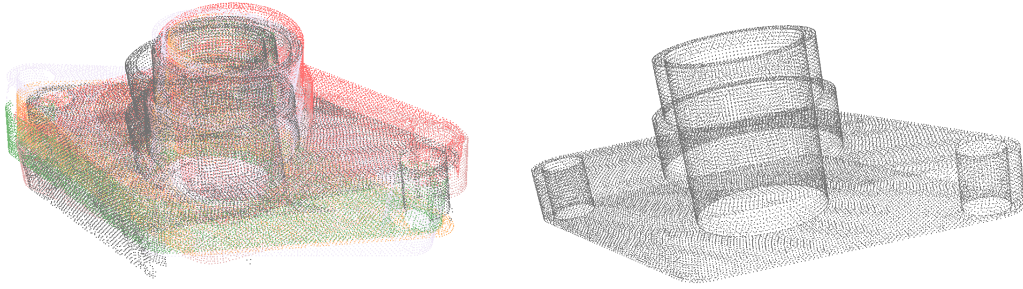
Fig. 9. left: A data set of 107670 points from a CAD model based on 7 scans; right: the registered point set
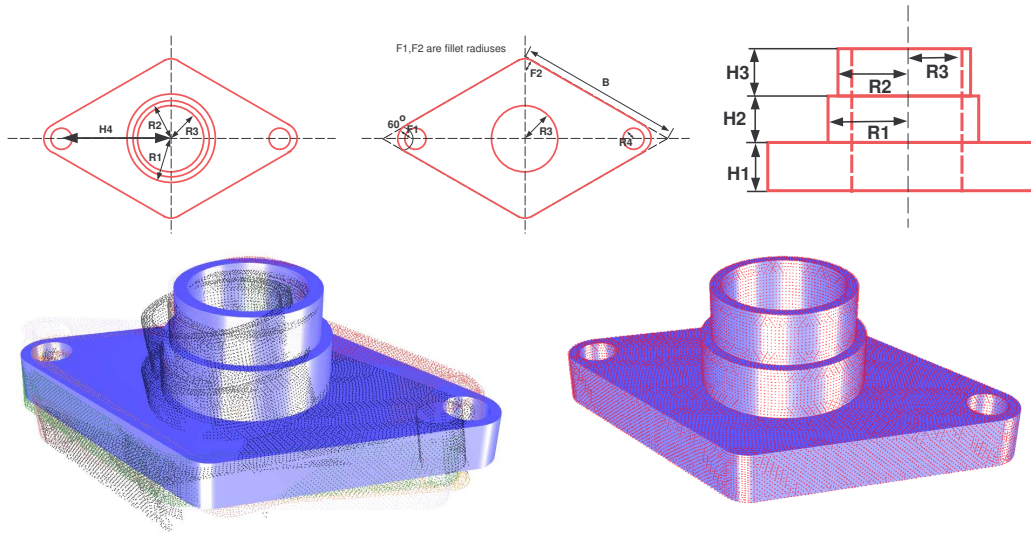

Fig. 10. Combining fitting and multiple-view registration. (top left): front view of the parametric model; (top right): side view; (middle left): initial fit; (middle right): optimized fit after 12 iterations of multiple-views TDM; (bottom) average error v.s. the number of iterations, and comparison with the single view case.

step will be very time consuming. To speed up this step, we sample a number of points on the surface firstly, then construct a kd-tree structure for finding the nearest point in the set of sample points. Viewing the nearest sample point as an initial point, we then apply a Newton iteration to compute a more

precise foot-point on the surface. Of course, if the surface is very simple, like a plane or cylinder, we can find the closest point directly.

Step size control. In each step we minimize the objective function by solving a linear system. We use Levenberg-Marquart regularization in order to avoid instabilities and too large steps [11].

5   Conclusions

We have shown that 3D shape fitting in the presence of constraints may be simplified and made more efficient by combining registration and surface approximation. To achieve a good convergence behavior, we have implemented this idea within the framework of SDM. We have also compared SDM with TDM and PDM, and found that SDM and TDM are more efficient than PDM, and with proper step size control TDM is as good as SDM in many cases. Moreover, we have shown that relaxing the initial registration in the final phase of our algorithm is easily formulated within our framework and improves the fitting accuracy.

Among the topics for future research we would like to mention a thorough investigation of the combination of registration and implicit surface fitting [16].

References

[1] L. Ambrosio, C. Mantegazza, Curvature and distance function from a manifold. *Journal of Geometric Analysis*, Vol 8, pp. 723–748, 1998.

[2] P. Benkö, G. Kos, T. Varady, L. Andor, R. Martin, Constrained fitting in reverse engineering, *Comp. Aided Geom. Design*, Vol. 19, pp. 173–205, 2002.

[3] P. J. Besl, N. D. McKay, A method for registration of 3D shapes, *IEEE PAMI*, Vol. 14, pp. 239–256, 1992.

[4] A. Blake, M. Isard, *Active Contours*, Springer, 1998.

[5] Y. Chen, G. Medioni, Object modeling by registration of multiple range images, *Image and Vision Computing,* pp. 145–155, 1992.

[6] R. B. Fisher, Applying knowledge to reverse engineering problems, *Comp.-Aided Design*, Vol. 36, pp. 501–510, 2004.

[7] G. Glaeser, H. Stachel, Open Geometry, Springer, 1999.

[8] M. Hofer, B. Odehnal, H. Pottmann, J. Wallner: 3D shape recognition and reconstruction based on line element geometry, accepted by ICCV 2005.

[9] H. Jin et al., Surface reconstruction from misregistered data, Proc. of SPIE, Vol. 2573, pp. 324–328, 1995.

[10] A. Karniel, Y. Belsky, Y. Reich, Decomposing the problem of constrained surface fitting in reverse engineering, *Comp.-Aided Design*, Vol. 37, pp. 399–417, 2005.

[11] C. T. Kelley, *Iterative Methods for Optimization,* SIAM, 1999.

[12] H. Meinhardt, *The Algorithmic Beauty of Sea Shells*, Springer, 1995.

[13] S. Lavallee and R. Szeliski, Recovering the position and orientation of free-form objects fromimage contours using 3D distance maps *PAMI*, Vol 17, pp. 378-390, 1995.

[14] B. Curless and M. Levoy, A volumetric method for building complex models from range images, *Computer Graphics*, Vol 30, pp. 303-312, 1996.

[15] J. Montagnat and H. Delingette, A hybrid framework for surface registration and deformable models, In *Computer Vision and Pattern Recognition, CVPR'97*, San Juan, PuertoRico, June 1997.

[16] S. Osher, R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces,* Springer, New York, 2003.

[17] N. Paragios, M. Rousson, Shape analysis towards model-based segmentation, in: S. Osher and N. Paragios (eds), *Geometric Level Set Methods in Imaging, Vision and Graphics*, Springer, pp. 231–250, 2003.

[18] H. Pottmann, S. Leopoldseder, A concept for parametric surface fitting which avoids the parametrization problem, *Comp. Aided Geom. Design*, Vol. 20, pp. 343–362, 2003.

[19] H. Pottmann, S. Leopoldseder, M. Hofer M, Registration without ICP, *Computer Vision and Image Understanding*, Vol. 95, pp. 54–71, 2004.

[20] H. Pottmann, J. Wallner, *Computational Line Geometry*, Springer, 2001.

[21] D. Terzopoulos and K. Fleisher, Deformable models, *Visual Computer*, Vol. 4, pp. 306-331, 1988.

[22] D. Tubic, P. Hebert and D. Laurendeau, A volumetric approach for interactive 3D modeling, *Computer Vision and Image Understanding*, Vol. 92, pp. 56–77, 2003.

[23] T. Várady, R. Martin, Reverse Engineering, In *Handbook of Computer Aided Geometric Design*, G. Farin et al., eds., Elsevier, pp. 651–681, 2002.

[24] B. Vemuri, Y. Chen, Joint image registration and segmentation, in: S. Osher and N. Paragios (eds), *Geometric Level Set Methods in Imaging, Vision and Graphics*, Springer, pp. 251–269, 2003.

[25] W. Wang, H. Pottmann, Y. Liu, Fitting B-Spline curves to point clouds by squared distance minimization. TR-2004-11, Dept. Comp. Science, Univ. of Hong Kong, 2004.

[26] A. Willis, X. Orriols, D. Cooper, Accurately estimating sherd 3D surface geometry with application to pot reconstruction. CVPR Workshop, 2003.

[27] H. Yang, W. Wang, J. Sun, Control point adjustment for B-spline curve approximation, *Comp.-Aided Design*, Vol. 36, pp. 639–652, 2004.