Robust Segmentation of Primitives from Range Data in the Presence of Geometric Degeneracy

David Marshall, Gabor Lukacs, and Ralph Martin

Abstract—This paper addresses a common problem in the segmentation of range images. We would like to identify and fit surfaces of known type wherever these are a good fit. This paper presents methods for the least-squares fitting of spheres, cylinders, cones, and tori to 3D point data, and their application within a segmentation framework. Least-squares fitting of surfaces other than planes, even of simple geometric type, has been rarely studied. Our main application areas of this research are reverse engineering of solid models from depth-maps and automated 3D inspection where reliable extraction of these surfaces is essential. Our fitting method has the particular advantage of being robust in the presence of geometric degeneracy, i.e., as the principal curvatures of the surfaces being fitted decrease (or become more equal), the results returned naturally become closer and closer to those surfaces of "simpler type," i.e., planes, cylinders, cones, or spheres, which best describe the data. Many other methods diverge because, in such cases, various parameters or their combination become infinite.

Index Terms—Nonlinear least squares, geometric distance, cylinder, cone, sphere, torus, surface fitting, segmentation.

1 Introduction

THIS paper considers the problems of least-squares fitting of spheres, cylinders, cones, and tori to 3D point data and the application of solutions to these problems in a segmentation framework for extracting such primitives reliably from range data. The motivation for this problem lies in reverse engineering of geometric shape [26], [32], [33], although we emphasize that the problem of extracting such geometrical features is common to many other 3D vision tasks, such as object recognition and localization, automated inspection, and robotics.

In our 3D vision system, a laser scanner or similar is used to capture 3D point data sampled from the surface of an object. From this, we would like to construct a CAD model—a boundary representation solid model of the object's shape. In particular, we would like to identify and fit simple surfaces of known type to portions of the boundary wherever these are in good agreement with the point data. The problem can be decomposed into two logical steps: *segmentation*, where the data points are grouped into sets each belonging to a different surface, and *fitting*, where the best surface of an appropriate type is fitted to each set. The new results in this paper mainly concern the latter problem, but also show their application to the former.

While plane fitting is well-understood, least-squares fitting of other surfaces, even of simple geometric type, has been relatively much less studied. We review previous approaches to the fitting of spheres, cylinders, cones, and tori, and then present new results on fitting these surfaces. Our method has the advantage of being robust in the sense that as the principal curvatures of the surfaces being fitted decrease (or become more equal), the results which are returned naturally become closer and closer to the surfaces of "simpler type" i.e., planes, cylinders, or cones (or spheres) which best describe the data. In the limit, as the curvature k drops below machine precision, such simpler surfaces are *automatically* returned *exactly* (Fig. 1). In comparison, many other methods diverge in such cases as various parameters or their combination become infinite.

- *D. Marshall and R. Martin are with the Department of Computer Science, Cardiff University, PO Box 916, Cardiff, UK, CF24 3XF. E-mail: {dave, ralph}@cs.cf.ac.uk.*
- *G. Lukacs is with the Computer and Automation Research Institute, Hungarian Academy of Sciences, H-1518 Budapest, POB 63, Hungary. E-mail: lukacs@sztaki.hu.*

Manuscript received 15 Mar. 2000; revised 28 Sept. 2000; accepted 2 Nov. 2000.
Recommended for acceptance by Y.-F. Wang.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 111715.

2 Segmentation

Segmentation is the problem of grouping the points in the original data set into subsets each of which logically belongs to a single primitive surface. Various approaches exist for segmenting simple surfaces from 3D data [1], [3], [4], [6], [9], [12], [18]. Most commonly, segmentation is treated as a local-to-global aggregation problem with similarity constraints employed to control the process. Often several stages are required, ranging from the estimation of local surface properties, such as curvature, to more complex feature clustering such as symmetry seeking. Typically, small initial seed regions are chosen at random positions. These are then grown and homogeneous regions are merged. However, such approaches tend to isolate the segmentation stage from the representation stage, with the result that the data partitioning may not agree well with the given primitive types. In addition, the sensitivity of these methods to noise in the data (especially outliers) may also lead to misclassification and, hence, poor results [16]. An efficient and reliable segmentation process thus depends on employing geometric knowledge of the primitive types: first, to guide the detection and grouping processes and, second, to assure the coherence and consistency of models throughout the whole segmentation process [1].

One approach used for segmentation in our reverse engineering project is the *recover and select* paradigm of Jaklic et al. [13], [15], [16], [17]. An iterative process recovers and selects specific instances of the required geometric primitives (planes, spheres, cylinders, cones, and tori). The basic approach partitions the data according to primitives by choosing the models such that the description is *best* in terms of global shape and error of fit. Initially, seed regions are placed at arbitrary locations in the data and models of each type of primitive are approximated (Fig. 2). Grossly mismatching models may be rejected at this stage.

An iterative grow and select phase is then operated. All valid models are grown for an equal number of steps (Fig. 2c); note that the models are allowed to overlap. The resulting models are then evaluated and some are selected for further growing. Optimal models from the overlapping sets are selected on the basis of the following criteria:

Area—the number of surface points contained in the model.

Error of Fit—the maximum or average distance between data points and the model.

Parameters of Model—the number of parameters used to describe the model.

Surface Type—the class of surface of the model.

When different models have similar goodness of fit, some types of model may be preferred to others and *Surface Type* is used to override natural ordering suggested by number of parameters in the model. This is typically done in terms of increasing surface type complexity (i.e., plane, sphere, cylinder, cone, torus), where simpler surfaces are chosen first (using similar ideas to those of Besl [3], Besl and Jain [4]). However, for some reverse engineering and other tasks, it may be necessary to impose a special order, for example, where cylindricity may be the most important criterion. The selection process employs a weighted sum of the above criteria as a cost-benefit measure to choose the optimal model or models.

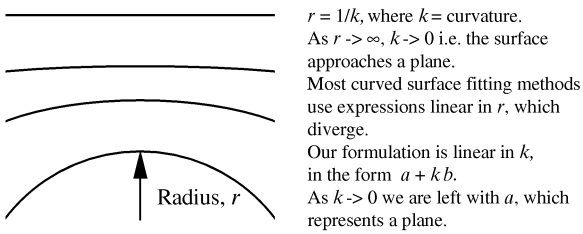

Fig. 1. Robust fitting method as principal curvatures decrease.

Models that have a poor error of fit (even if not overlapping other models) are also rejected at the selection stage.

Now, we discuss novel fitting methods for spheres, cylinders, cones, and tori, and then present results of using these fitting methods in this segmentation framework.

3 Surface Fit and General Nonlinear Least-Squares

We initially outline various basic concepts needed for least-squares fitting of simple geometric surfaces and then review previous approaches to this problem.

Let us assume that each of the 3D input data points $\mathbf{p}_i$ for $i = 1, \ldots, m$ lies close to the same member of a family of surfaces which can be parametrized by $\mathbf{s} \in G \subseteq \mathbb{R}^s$, where G is an open set. Let $d(\mathbf{s}, \mathbf{p}_i)$ be a function which is defined as the distance of the point $\mathbf{p}_i \in \mathbb{R}^3$ from that surface in the family identified by $\mathbf{s}$. Throughout, d will be called the "true" distance function of the surface (fitting methods generally rely on approximations to this distance as will be explained later).

A surface, which goes through all the points, can be viewed as that member of the family which corresponds to the solution of the simultaneous system of m equations:

$$d(\mathbf{s}, \mathbf{p}_i) = 0, \qquad \text{for } i = 1, \ldots, m. \tag{1}$$

Since the number of points m is usually much greater than the number of degrees of freedom s, this system of equations is overdetermined and, in general, cannot be solved. However, it is possible to solve it in the least-squares sense, i.e., to find that surface which is the best fit to the points "on the average," minimizing

$$\sum_{i=1}^{m} d(\mathbf{s}, \mathbf{p}_i)^2. \tag{2}$$

Sometimes, we may have additional nonlinear constraints expressed as

$$H(\mathbf{s}) = \mathbf{0} \in \mathbb{R}^t \tag{3}$$

for $t < s$. For example, the surface form used might describe a general quadric, but we may wish to impose that the surface found is a cylinder, which can by done by imposing constraints on $\mathbf{s}$. Using the principle of Lagrangian multipliers to include these constraints, a nonlinear generalized eigenvalue problem results, which is not easy to solve. A simpler approach is to use (3) to eliminate t unknowns and to reduce the problem to an unconstrained optimization problem in a lower-dimensional space. This is the method we use.

Usually, the family of surfaces is defined as points satisfying an implicit equation:

$$f(\mathbf{s}, \mathbf{x}) = 0, \qquad \text{for } \mathbf{x} \in \mathbb{R}^3, \tag{4}$$

where $\mathbf{s}$ is the family parameter vector. Although if we fix $\mathbf{s}$, f and d have the same roots in space; they may behave quite differently for points which do not lie on the surface. Thus, if instead of (2) one minimizes $\sum f^2$, this may give quite different results. However, this approach can be justified if *both* the function f and the constraint H in (3) are of particularly simple form. If f is linear and H is quadratic in terms of the parameters, then linear generalized eigenvalue techniques work [10]. If f is nonlinear but H is still quadratic, then one can try Taubin's generalized eigenvector fit [25]. Nevertheless, the choice of form for f influences the behavior of the nonlinear fitting algorithm and, consequently, the quality of the solution. For fitting ellipses, Rosin [24] shows that choosing f carelessly can lead to severely biased estimates for $\mathbf{s}$. Below, we choose to use and give particular, "fairly good" f functions (i.e., for which f behaves much like d near the surface) which are highly nonlinear and which have no additional constraints on the parameters $\mathbf{s}$.

4 Approximating the True Distance

As far as possible, singularities of $d(\mathbf{s}, \mathbf{p}_i)$ must be avoided in the range where solutions may lie. These singularities may be places

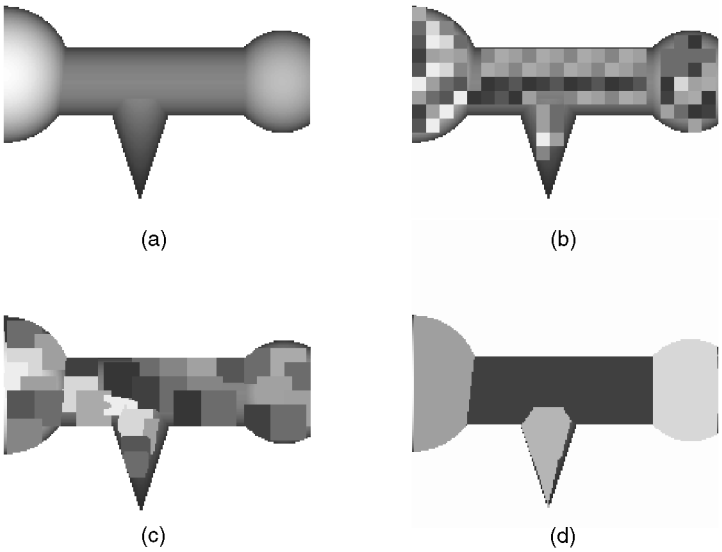

Fig. 2. (a) Initial depth data. (b) Seed placement (c) Intermediate regions. (d) Final segmentation.

where a denominator in $d(\mathbf{s}, \mathbf{p}_i)$ vanishes or where the distance function is not differentiable. Using Euclidean metrics, such singularities arise frequently since the Euclidean distance from a given fixed point is *itself* singular in this sense. Nevertheless, most of these singularities are only *computational*, i.e., inessential discontinuities in the mathematical sense and a limiting value of the distance function can still be found for the critical parameter value. Even so, computation of the distance function (or its derivatives) can be unstable at such points since it may require the subtraction of similar quantities, etc.

Avoiding the effects of singularities can be achieved by means of various techniques. First, choose a suitable parametrization where the critical values do not lie on the border of G. Second, we change the definition of $d(\mathbf{s}, \mathbf{p}_i)$ slightly in order to get rid of singularities. We shall say that this modified definition is *faithful* to the true Euclidean distance function if: 1) the function is zero where the true distance is zero and 2) at these points, the derivatives with respect to the parameters are the same for the true distance and the modified definition.

Faithful distance functions can be obtained if one approximates square roots within $d(\mathbf{s}, \mathbf{p}_i)$ in the following way. Suppose the distance function is of the form:

$$d(\mathbf{s}, \mathbf{p}_i) = \sqrt{g} - h, \tag{5}$$

where both g and h may depend on the parameter vector $\mathbf{s}$ and on the point $\mathbf{p}_i$ in 3D. In order to get rid of the square root, we might try to minimize $\sum (g - h^2)^2$ instead of (2) since $d = 0$ when $g = h^2$. Unfortunately, the effect is now that we are searching for the surface which fits best in terms of the "average" *square* of the distance instead of just the distance. Thus, this transformation amplifies the importance of data points farther from the surface and flattens the goal function in the neighborhood of the solution. Instead, let us use the following approximation:

$$\tilde{d} = \frac{g - h^2}{2h}. \tag{6}$$

and we minimize

$$\sum \tilde{d}^2(\mathbf{s}, \mathbf{p}_i) = \sum \frac{\left(g(\mathbf{s}, \mathbf{p}_i) - h^2(\mathbf{s}, \mathbf{p}_i)\right)^2}{4h^2(\mathbf{s}, \mathbf{p}_i)}. \tag{7}$$

In summary, in our approach, we start with an exact expression for the distance d. This is replaced by a simplification which is easier to compute, but which still has the same zero set and derivatives at the zero set. In contrast, similar work by Taubin [25] starts from a parametrized family of implicit functions $f = 0$. He notes that while f itself is not a good approximation to d, $f/|\nabla f|$ is much better, i.e., he replaces the original implicit function with a new one whose value is a better approximation to d. Although Taubin does not state so explicitly, it is clear that it is better because the derivatives with respect to spatial parameters of this function are the same as those of the distance function. In practice, Taubin's approach can be used only if f is linear with respect to the parameters and the system to be solved then includes a quadratic constraint. Another difference is that our approach is better behaved with respect to singularities.

5 Fitting Spheres, Cylinders, Cones, and Tori

The linear least-squares fitting of second order curves and surfaces has been recently considered by several authors [10], [11], [22], [23]. However, specific linear methods still do not exist for right cylinders and cones—the reason is that the equations expressing the conditions for a quadric to be a right cylinder or a cone are not quadratic. If general linear methods are used for algebraic second order surfaces, the solutions found are usually not right cylinders or cones, and may even be very different from the optimum surfaces of such type. In this sense, algebraic techniques, which use the value of the implicit quadratic form as the "distance" from the surface, approximate the true geometric distance in an unfaithful way.

The situation is much simpler for spheres since straightforward algebraic methods work in this case: under a suitable normalization, the minimized algebraic distance will reflect the geometric distance as well. For example, the method in [22] minimizes

$$\sum_i \left(A(x_i^2 + y_i^2 + z_i^2) + Dx_i + Ey_i + Fz_i + G\right)^2 \tag{8}$$

under the condition

$$D^2 + E^2 + F^2 - 4AG = 1 \tag{9}$$

which is basically equivalent to our minimization in (7). (Note that the simple constraint $A = 1$ may give quite unfaithful results, as shown in [22].) Nevertheless, we give our nonlinear method for spheres in the next section as an illustration of our method, as it has certain advantages.

Nonlinear methods, which take into account the true geometric distance, match the requirements of our segmentation method well. Those points belonging to the same surface are selected by means of the segmentation technique which provides an initial approximation for the parameters of each surface. Starting from these, at the expense of some computing time, one can obtain a more accurate fit. Our nonlinear methods also work well in other applications where an initial approximate fit for the surface is known.

Earlier nonlinear estimation approaches usually worked with cylinders and spheres. As a rule, the equations contain positional parameters of centers or axes and, so, they become ill-conditioned in limiting situations (see e.g., [5]), which is unacceptable if automatic segmentation is the objective. Our nonlinear methods have been carefully designed to overcome this problem.

Recently, surface fitting methods employing a variety of geometric constraints have been developed. Keren and Gotsman [14] constrain implicit polynomials such that their parametrizations are guaranteed to satisfy certain topological properties. Werghi et al. [28], [29], [30], [31], [32] give a framework within which a variety of geometric constraints (e.g., enforcing parallelism or perpendicularity between surfaces) can be imposed. By applying constraints as part of the fitting process, and solving a constrained set of equations, improved extraction of primitives belonging to an object is obtained. This approach is very suitable for reverse engineering and inspection applications. However, some of the parametrizations they use resort to standard representations of geometric primitives (e.g., a standard quadric surface representation of a cylinder or a cone). This leads to the need to enforce highly nonlinear constraints. If our representation of geometric primitives were employed in such a framework, several constraints would be linearized and, thus, more easily and reliably applied. For instance, many geometric constraints are concerned with the alignment of surface axes. In our representation, these axes are explicitly available even for tori and are easily constrained. Curvature is also explicitly represented in our formulation and is available for use in other common geometric constraints. The application of our approach to constraint-based fitting is currently under further investigation.

5.1 A Simple Formula for Vector Products

The following formula is used frequently in the sections below. Let $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{d}$ be arbitrary vectors in 3D. Then,

$$\langle \mathbf{a} \times \mathbf{b}, \mathbf{c} \times \mathbf{d} \rangle = \langle \mathbf{a}, \mathbf{c} \rangle \langle \mathbf{b}, \mathbf{d} \rangle - \langle \mathbf{a}, \mathbf{d} \rangle \langle \mathbf{b}, \mathbf{c} \rangle. \tag{10}$$

5.2 Sphere Fitting

For nonlinear least-squares fit, the parametrization of the sphere will be the following: Suppose that the *closest* point of the sphere (*not* its center) to the origin is $\varrho\mathbf{n}$, where $|\mathbf{n}| = 1$ and the radius of the sphere is $1/k$. Then, if $\mathbf{p}$ is an arbitrary point in space, the distance of this point from the surface of the sphere is

$$\begin{aligned} d(\mathbf{s},\mathbf{p}) &= \left|\mathbf{p} - \left(\varrho + \frac{1}{k}\right)\mathbf{n}\right| - \frac{1}{k} \\ &= \sqrt{\langle \mathbf{p} - \left(\varrho + \frac{1}{k}\right)\mathbf{n}, \mathbf{p} - \left(\varrho + \frac{1}{k}\right)\mathbf{n}\rangle} - \frac{1}{k}. \end{aligned} \tag{11}$$

Since this function is of the form (5), we can apply (6) to give

$$\tilde{d}(\mathbf{s},\mathbf{p}) = \frac{k}{2}\left(|\mathbf{p}|^2 - 2\varrho\langle \mathbf{p},\mathbf{n}\rangle + \varrho^2\right) + \varrho - \langle \mathbf{p},\mathbf{n}\rangle, \tag{12}$$

or, if one introduces the notation

$$\hat{\mathbf{p}} = \mathbf{p} - \varrho\mathbf{n}, \tag{13}$$

one obtains

$$\tilde{d}(\mathbf{s},\mathbf{p}) = \frac{k}{2}|\hat{\mathbf{p}}|^2 - \langle \hat{\mathbf{p}},\mathbf{n}\rangle. \tag{14}$$

Here, $\hat{\mathbf{p}}$ is the expression of $\mathbf{p}$ with respect to an origin at $\varrho\mathbf{n}$. Now, let us parametrize $\mathbf{n}$ using polar coordinates:

$$\mathbf{n} = (\cos\varphi\sin\vartheta, \sin\varphi\sin\vartheta, \cos\vartheta), \tag{15}$$

where ϑ is the angle between $\mathbf{n}$ and the z axis and φ is the angle between the projection of $\mathbf{n}$ onto the plane $z = 0$ and the x axis. Differentiating $\mathbf{n}$ with respect to φ and ϑ, one obtains two partial derivative vectors which are orthogonal to each other and to $\mathbf{n}$ (superscripts denote derivatives); these will be used later:

$$\mathbf{n}^{\varphi} = (-\sin\varphi\sin\vartheta, \cos\varphi\sin\vartheta, 0), \tag{16}$$

$$\mathbf{n}^{\vartheta} = (\cos\varphi\cos\vartheta, \sin\varphi\cos\vartheta, -\sin\vartheta). \tag{17}$$

Thus, $\mathbf{n}$ and, hence, $\tilde{d}$ can be parametrized without constraints by $\mathbf{s} = (\varrho, \varphi, \vartheta, k)$.

The partial derivatives of the approximate distance function (12) are the following:

$$\frac{\partial\tilde{d}}{\partial\varrho} = k(\varrho - \langle \mathbf{p},\mathbf{n}\rangle) + 1, \tag{18}$$

$$\frac{\partial\tilde{d}}{\partial\varphi} = (-k\varrho - 1)\langle \mathbf{p},\mathbf{n}^{\varphi}\rangle, \tag{19}$$

$$\frac{\partial\tilde{d}}{\partial\vartheta} = (-k\varrho - 1)\langle \mathbf{p},\mathbf{n}^{\vartheta}\rangle, \tag{20}$$

$$\frac{\partial\tilde{d}}{\partial k} = \frac{1}{2}\left(|\mathbf{p}|^2 - 2\varrho\langle \mathbf{p},\mathbf{n}\rangle + \varrho^2\right). \tag{21}$$

Note that unlike (8), (12) is nonlinear, but behaves well as the curvature of the sphere decreases, as when $k \to 0$, all the terms are bounded, and (12) reduces to the expression that would be used for least-squares plane fitting. In contrast, observe that some of the terms will tend to infinity both in the objective function given in (8) and in the constraint given in (9).

5.3 Right Circular Cylinder Fitting

The parametrization used for the cylinder is similar to that for the sphere. The closest point of the cylinder to the origin is $\varrho\mathbf{n}$, where

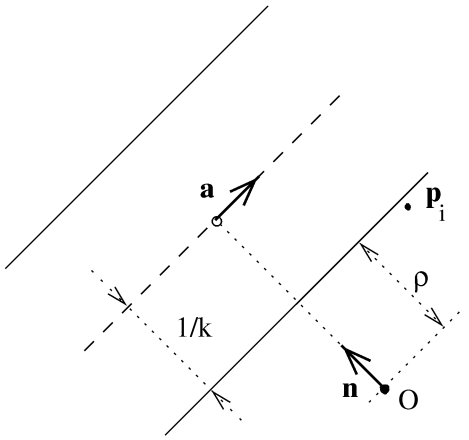

Fig. 3. Parametrization of the cylinder.

$|\mathbf{n}| = 1$. Assume that the direction of the axis of the cylinder is $\mathbf{a}$ with $|\mathbf{a}| = 1$ and the radius of the cylinder is $1/k$.

Note that $\langle \mathbf{n},\mathbf{a}\rangle = 0$. Now, let us suppose that $\mathbf{p}$ is an arbitrary point in space whose distance from the surface of the cylinder is to be found. This is done by finding its distance from the symmetry axis and from that subtracting the radius of the cylinder (see Fig. 3).

$$\begin{aligned} d(\mathbf{s},\mathbf{p}) &= \left|(\mathbf{p} - \left(\varrho + \frac{1}{k}\right)\mathbf{n}) \times \mathbf{a}\right| - \frac{1}{k} \\ &= \sqrt{|\mathbf{p} - \left(\varrho + \frac{1}{k}\right)\mathbf{n}|^2 - \langle \mathbf{p} - \left(\varrho + \frac{1}{k}\right)\rangle\mathbf{n},\mathbf{a}^2} - \frac{1}{k}. \end{aligned} \tag{22}$$

Since this function is of the form in (5), we can apply (6) to obtain

$$\begin{aligned} \tilde{d}(\mathbf{s},\mathbf{p}) &= \frac{k}{2}\left(|\mathbf{p}|^2 - 2\varrho\langle \mathbf{p},\mathbf{n}\rangle - \langle \mathbf{p},\mathbf{a}\rangle^2 + \varrho^2\right) + \varrho - \langle \mathbf{p},\mathbf{n}\rangle \\ &= \frac{k}{2}|\hat{\mathbf{p}} \times \mathbf{a}|^2 - \langle \hat{\mathbf{p}},\mathbf{n}\rangle, \end{aligned} \tag{23}$$

where $\hat{\mathbf{p}} = \mathbf{p} - \varrho\mathbf{n}$ as in (13). Using appropriate parametrizations for $\mathbf{n}$ and $\mathbf{a}$, we would like to minimize

$$\sum_i \tilde{d}^2(\mathbf{s},\mathbf{p}_i). \tag{24}$$

Let us make some observations about the right-hand side of (23). First, it is linear in the curvature k if all other parameters are fixed, which results in a *separable* nonlinear least-squares problem [7]. Such problems can be easier to solve than the fully nonlinear case. Clearly, an initial estimate for k is not needed if we have estimates for the other parameters, as an initial value for k can be found by solving a linear least-squares problem in which all other parameters are fixed. Note that (23) behaves well as k gets smaller (ϱ is bounded within sensible limits by the geometric configuration of the scanner); compare (22) which subtracts two large quantities as k becomes small. In the limit as $k \to 0$, we get $\tilde{d} = \varrho - \langle \mathbf{p},\mathbf{n}\rangle$, and as before, the problem reduces to linear least-squares fitting of a plane.

Again, we would like to parametrize $\mathbf{n}$ and $\mathbf{a}$ to satisfy the constraints

$$|\mathbf{n}| = |\mathbf{a}| = 1, \quad \langle \mathbf{n},\mathbf{a}\rangle = 0; \tag{25}$$

so, again, we use polar coordinates. The parametrization for $\mathbf{n}$ was introduced in (15). Equations (16) and (17) are the partial derivatives of $\mathbf{n}$. Thus, if we put

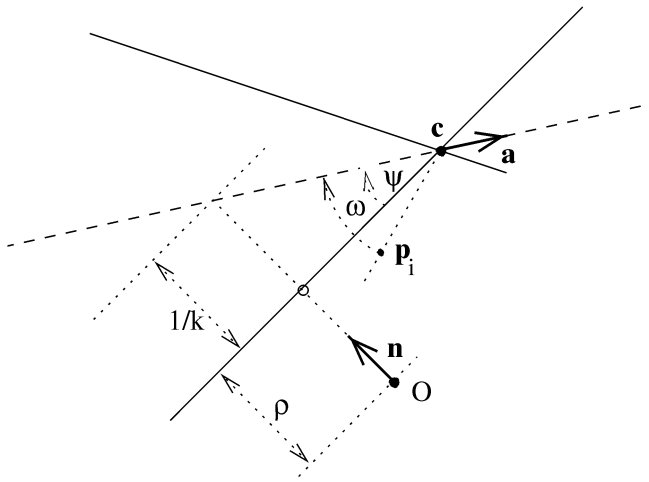

Fig. 4. Parametrization of the cone.

$$\overline{\mathbf{n}^{\varphi}} = (-\sin\varphi, \cos\varphi, 0) = \frac{\mathbf{n}^{\varphi}}{\sin\vartheta}, \tag{26}$$

then $\mathbf{n}^{\varphi}$, $\overline{\mathbf{n}^{\varphi}}$, and $\mathbf{n}$ form an orthonormal basis. Thus, we can parametrize $\mathbf{a}$ as follows:

$$\mathbf{a} = \mathbf{n}^{\vartheta}\cos\alpha + \overline{\mathbf{n}^{\varphi}}\sin\alpha, \tag{27}$$

where α is the angle between $\mathbf{a}$ and $\mathbf{n}^{\vartheta}$. Thus, $\mathbf{n}$ and $\mathbf{a}$ are parametrized through φ, ϑ, and α by means of (15), (17), (26), and (27).

A nonlinear distance function for right circular cylinders, which is faithful up to the first derivative, is given by (23). It is parametrized in terms of ϱ, φ, ϑ, α, and k using (15), (27), (17), and (26). If all other parameters are fixed, this function is linear in terms of the curvature k of the cylinder. The partial derivatives of $\tilde{d}$ (relative to ϱ, φ, ϑ, α, and k) may be determined accordingly (as in Section 5.2). Details may be found in [19].

5.4 Right Circular Cone Fitting

The parametrization used for the cone is similar to that for the cylinder. Let $\varrho\mathbf{n}$ with $|\mathbf{n}| = 1$ be that point on the cone surface for which a line in the direction of the surface normal passes through the origin. (Hence, $\mathbf{n}$ is normal to the cone.) Let the nonzero principal curvature of the cone at the point $\varrho\mathbf{n}$ be k. Denote the unit direction of the axis of the cone by $\mathbf{a}$. $\mathbf{n}$ is parametrized by φ and ϑ as in (15). Since $\mathbf{n}$ and $\mathbf{a}$ are not now perpendicular, $\mathbf{a}$ can be parametrized by two polar coordinate angles, σ and τ

$$\mathbf{a} = (\cos\sigma\sin\tau, \sin\sigma\sin\tau, \cos\tau), \tag{28}$$

where τ is the angle between $\mathbf{a}$ and the z axis and σ is the angle between the projection of $\mathbf{a}$ onto the plane $z = 0$ and the x axis. The six parameters $(\varrho, \varphi, \vartheta, k, \sigma, \tau)$ characterize the right circular cone surface.

In order to understand how this works, let the half angle of the cone be ψ (see Fig. 4) and the position of the apex of the cone be $\mathbf{c}$; we shall express ψ and $\mathbf{c}$ using the above parameters later. Let the angle between the axis of the cone and $\mathbf{p} - \mathbf{c}$ be ω. Using these, the distance of $\mathbf{p}$, an arbitrary point, from the cone surface is given by (note that $\mathbf{p}$ may not lie in the plane of $\mathbf{n}$ and $\mathbf{a}$)

$$\begin{aligned} d(\mathbf{s}, \mathbf{p}) &= |\mathbf{p} - \mathbf{c}|\sin(\omega - \psi) \\ &= |\mathbf{p} - \mathbf{c}|\sin\omega\cos\psi - |\mathbf{p} - \mathbf{c}|\cos\omega\sin\psi. \end{aligned} \tag{29}$$

Without loss of generality, we can suppose that both ψ and ω are acute angles. Since the direction of the axis, $\mathbf{a}$, is a unit vector, we have:

$$d(\mathbf{s}, \mathbf{p}) = |(\mathbf{p} - \mathbf{c}) \times \mathbf{a}|\cos\psi - |\langle \mathbf{p} - \mathbf{c}, \mathbf{a}\rangle|\sin\psi. \tag{30}$$

Moreover, since the angle between $\mathbf{n}$ and $\mathbf{a}$ is the complementary angle to ψ, we have:

$$\cos\psi = |\mathbf{n} \times \mathbf{a}| \qquad \sin\psi = |\langle \mathbf{n}, \mathbf{a}\rangle|. \tag{31}$$

Thus, from (30) one obtains:

$$\begin{aligned} d(\mathbf{s}, \mathbf{p}) =& |(\mathbf{p} - \mathbf{c}) \times \mathbf{a}||\mathbf{n} \times \mathbf{a}| - |\langle \mathbf{p} - \mathbf{c}, \mathbf{a}\rangle\langle \mathbf{n}, \mathbf{a}\rangle| \\ =& |\mathbf{n} \times \mathbf{a}|\sqrt{|\mathbf{p} - \mathbf{c}|^2 - \langle \mathbf{p} - \mathbf{c}, \mathbf{a}\rangle^2} - |\langle \mathbf{n}, \mathbf{a}\rangle\langle \mathbf{p} - \mathbf{c}, \mathbf{a}\rangle|. \end{aligned} \tag{32}$$

Again, this function is of the form given in (5) and using (6) gives

$$\begin{aligned} \tilde{d}(\mathbf{s}, \mathbf{p}) &= \frac{|\mathbf{p} - \mathbf{c}|^2\cos^2\psi - \langle \mathbf{p} - \mathbf{c}, \mathbf{a}\rangle^2}{2\langle \mathbf{p} - \mathbf{c}, \mathbf{a}\rangle\sin\psi} \\ &= \frac{|\mathbf{p} - \mathbf{c}|^2|\mathbf{n} \times \mathbf{a}|^2 - \langle \mathbf{p} - \mathbf{c}, \mathbf{a}\rangle^2}{2\langle \mathbf{p} - \mathbf{c}, \mathbf{a}\rangle\langle \mathbf{n}, \mathbf{a}\rangle}. \end{aligned} \tag{33}$$

Now, we express the position of the apex $\mathbf{c}$ in terms of the normal vector $\mathbf{n}(\varphi, \vartheta)$, the distance ϱ, the curvature k, and the axis of the cone $\mathbf{a}(\sigma, \tau)$:

$$\mathbf{c} = \left(\varrho + \frac{1}{k}\right)\mathbf{n} + \gamma\mathbf{a}. \tag{34}$$

Now, $\langle \mathbf{c}, \mathbf{n}\rangle = \varrho$, so $\gamma = -1/(k\langle \mathbf{n}, \mathbf{a}\rangle)$ and, thus,

$$\mathbf{c} = \left(\varrho + \frac{1}{k}\right)\mathbf{n} - \frac{\mathbf{a}}{k\langle \mathbf{n}, \mathbf{a}\rangle}. \tag{35}$$

Substituting this into (33) gives (again, $\hat{\mathbf{p}} = \mathbf{p} - \varrho\mathbf{n}$)

$$\begin{aligned} \tilde{d} = \Big(& |\mathbf{n} \times \mathbf{a}|^2\big(|\hat{\mathbf{p}} - \mathbf{n}/k|^2 - \langle \hat{\mathbf{p}} - \mathbf{n}/k, \mathbf{a}\rangle^2\big) \\ & - \big(\langle \hat{\mathbf{p}} - \mathbf{n}/k, \mathbf{a}\rangle\langle \mathbf{n}, \mathbf{a}\rangle + 1/k\big)^2\Big) \\ & /2\big(\langle \hat{\mathbf{p}} - \mathbf{n}/k, \mathbf{a}\rangle\langle \mathbf{n}, \mathbf{a}\rangle + 1/k\big), \end{aligned} \tag{36}$$

When using Pythagoras' theorem, it is easy to see that the coefficient of $1/k^2$ in the numerator is zero. Multiplying both the numerator and the denominator by k, we arrive at

$$\tilde{d}(\mathbf{s}, \mathbf{p}) = \frac{\frac{k}{2}\left(|\mathbf{n} \times \mathbf{a}|^2|\hat{\mathbf{p}}|^2 - \langle \hat{\mathbf{p}}, \mathbf{a}\rangle^2\right) - \langle \hat{\mathbf{p}}, \mathbf{n}\rangle|\mathbf{n} \times \mathbf{a}|^2}{k\langle \hat{\mathbf{p}}, \mathbf{a}\rangle\langle \mathbf{n}, \mathbf{a}\rangle + |\mathbf{n} \times \mathbf{a}|^2}. \tag{37}$$

Thus, this function $\tilde{d}$ depends on six unconstrained parameters: ϱ, φ, ϑ, σ, τ, and k. The partial derivatives of $\tilde{d}$ with respect to these parameters may be determined accordingly (as in Section 5.2). Details may be found in [19].

5.5 Torus Fitting

Our approach for tori is again similar. A torus can be parametrized using seven unconstrained parameters. A torus can be obtained by sweeping a circular disc around an axis in the plane of the circle. The radius of the disc is the minor radius of the torus and the distance of the center of the disc from the axis is the major radius of the torus. Tori whose major radius is *smaller* then the minor one can also be considered. In this case, the resulting surface is self-intersecting and it is necessary to distinguish the different parts. The smaller arcs sweep a "lemon-torus" (i.e., the inner part of the torus surface), while the larger arcs sweep an "apple-torus" (i.e., the outer part of the torus surface); we will also refer to a non-self-intersecting torus as "apple-shaped." In special cases, the torus may degenerate to a sphere, as the major radius vanishes, or into a cone, as the minor radius tends to infinity. Our equations, given below, appropriately reduce to those for sphere or cone fitting in such cases. (If the major radius tends to infinity, the torus becomes

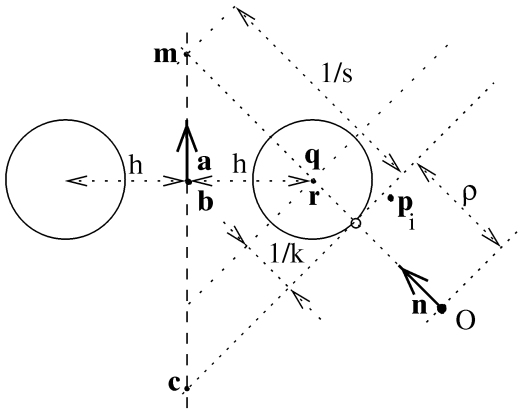

Fig. 5. Parametrization of the torus.

a cylinder. This case will be singular, but mathematically close to the cylinder fit.)

The parametrization used for the torus is the following: The point on the torus where a line through the surface normal passes through the origin is $\varrho\mathbf{n}$, where $|\mathbf{n}| = 1$. The principal curvature of the torus corresponding to the minor radius at the point $\varrho\mathbf{n}$ is k (i.e., the radius of the disk is $1/k$). The other principal curvature is s and the corresponding center of curvature lies on the axis of the torus. Let the unit direction vector of the torus axis be $\mathbf{a}$. (Fig. 5.) We parametrize $\mathbf{n}$ by φ and ϑ as in (15) and the unit vector $\mathbf{a}$ as in (28). The unconstrained parameters $(\varrho, \varphi, \vartheta, k, s, \sigma, \tau)$ entirely characterize the torus surface.

A nonlinear distance function for tori, which is faithful up to the first derivative, is now given by:

$$\tilde{d}(\mathbf{s}, \mathbf{p}) = \tilde{d}_0(\varrho, \varphi, \vartheta, k, \mathbf{p}) - \delta_\epsilon(\varrho, \varphi, \vartheta, k, s, \sigma, \tau, \mathbf{p}), \tag{38}$$

where $\tilde{d}_0$ is the approximate distance function for the sphere (12):

$$\begin{aligned}\tilde{d}_0 &= \frac{k}{2}\Big(|\mathbf{p}|^2 - 2\varrho\langle\mathbf{p}, \mathbf{n}\rangle + \varrho^2\Big) + \varrho - \langle\mathbf{p}, \mathbf{n}\rangle \\ &= \frac{k}{2}|\hat{\mathbf{p}}|^2 - \langle\hat{\mathbf{p}}, \mathbf{n}\rangle,\end{aligned} \tag{39}$$

while

$$\begin{aligned}\delta_\epsilon = \left(\frac{k}{s} - 1\right)\Bigg[&\epsilon \cdot \operatorname{sign}\left(\frac{k^2}{s} - k\right)|(\hat{\mathbf{p}} - \mathbf{n}/s) \times \mathbf{a}||\mathbf{n} \times \mathbf{a}| \\ &+ \langle(\hat{\mathbf{p}} - \mathbf{n}/s) \times \mathbf{a}, \mathbf{n} \times \mathbf{a}\rangle\Bigg],\end{aligned} \tag{40}$$

where $\epsilon = +1$ for an apple torus surface and $\epsilon = -1$ for a lemon torus surface.

The partial derivatives of $\tilde{d}$ with respect to these parameters may be determined accordingly (as in Section 5.2). Details may be found in [19].

It is worth emphasising that this formulation of the torus (38) behaves well in limiting situations. If $k = s$, then, from (40), we simply get the distance expression for a sphere given in (14). If $k \to 0$ and s is bounded from below then either for $\epsilon = -1$ or for $\epsilon = +1$, we get the distance expression for a cone given in (32). If $s \to 0$, then (38) degenerates to the distance function of a cylinder given in (23) with axis direction $\mathbf{n} \times \mathbf{a}$. (See [19], [20].)

6 Initial Estimates

To find the solution of any of the above nonlinear least-squares problems, an iterative technique is used; we use the Levenberg-Marquardt method [7]. Any such algorithm requires some good initial estimate of the solution which is then refined. A bad initial estimate will lead to problems of local minima or long convergence times. Here, we give one reliable method of finding such initial estimates.

For all surfaces, we need to apply Step 1 below. For surfaces other than spheres, we also apply Steps 2 and 3. Step 2 is explained in greater detail below:

1. We pick a surface point at which we have a normal vector estimate and place the origin at this point, the "base point." Thus, we have an estimate for $\mathbf{n}$ and, hence, φ, ϑ, the initial estimate for ϱ is 0. (Note that the solution surface need not pass through the base point since ϱ can change.)

   For sphere fitting, these values alone are sufficient and an estimate for k can now be found by solving a linear least-squares problem as mentioned earlier.
2. For other surfaces, we need to find an estimate of the rotational axis. The method in [19], [20] only uses estimates of the surface normal vector and computes the axis from a number of four-tuples of normals. An alternative approach which finds the best (least-squares) rotational axis has recently been suggested by Pottmann and Randrup [21].
3. For cylinder fitting, we adjust the normal to be perpendicular to the axis. α is then found as the angle between $\mathbf{a}$ and $\overline{\mathbf{n}^\vartheta}$ and we compute the distance of the base point from the axis, which is $1/k$ signed with the direction of $\mathbf{n}$.

   In the case of cone fitting, after estimating the rotational axis $\mathbf{a}$, we compute the distance of the base point from the axis *along* the estimated normal line in order to obtain $1/k$.

In the torus case, for better conditioning, one has to pick a base point at which the normal subtends as large an angle as possible with the estimated axis. We get $1/s$ as the distance along the normal between the base point and the estimated axis. For simplicity, we can put $k = 0$ (i.e., start with a cone) and try both cases $\epsilon = +1$ and $\epsilon = -1$. More robustly, one can opt to estimate principal curvatures of the surface at the base point. As s is one of the principal curvatures, we can compute the other, k, even if we only estimate the Gaussian curvature. We can then determine ϵ by noting on which sheet of the torus the base point lies. Thus, $\epsilon = +1$ if $|k| > |s|$ or $ks < 0$ and $\epsilon = -1$ otherwise (the decision should be clear if we have a well-placed base point chosen as described above). Here, we assume that the point set being fitted does not contain points belonging to both the apple and lemon sheets of the same torus simultaneously. This is very unlikely to happen in practice, but if it is considered to be a possibility, before fitting we should separate the points into two sets, one for each sheet, using curvature estimates and the given criterion.

These methods of computing initial estimates have been carefully chosen to provide adequate starting values for the nonlinear optimization, given the accuracy of the commercial scanner we use.

6.1 Estimating the Axis of Rotation

Suppose we have a number of points and corresponding surface normal vectors on a surface of revolution. We would like to compute the axis of rotation of the surface. This task can be expressed in the following geometric form: Given m straight lines in 3D, compute the straight line intersecting all of them (if such a line exists).

Let us denote by $\mathbf{p}_i$ a fixed point on the ith straight line and by $\mathbf{n}_i$ the direction of that straight line $i = 0, 1, \ldots, m-1$. Now, let us draw a straight line through two generic points of the 0th and 1st straight lines, respectively. Assume that these points have the parameters t_0 and t_1. A necessary condition that a straight line passing through the point $\mathbf{p}_i$ in the direction $\mathbf{n}_i$ ($i = 2, 3, \ldots, m-1$) intersects the previous line is that the vectors $\mathbf{p}_0 + t_0\mathbf{n}_0 - \mathbf{p}_i$, $\mathbf{p}_1 + t_1\mathbf{n}_1 - \mathbf{p}_0 - t_0\mathbf{n}_0$, and $\mathbf{n}_i$ are coplanar. This means that the following vector triple product must vanish:

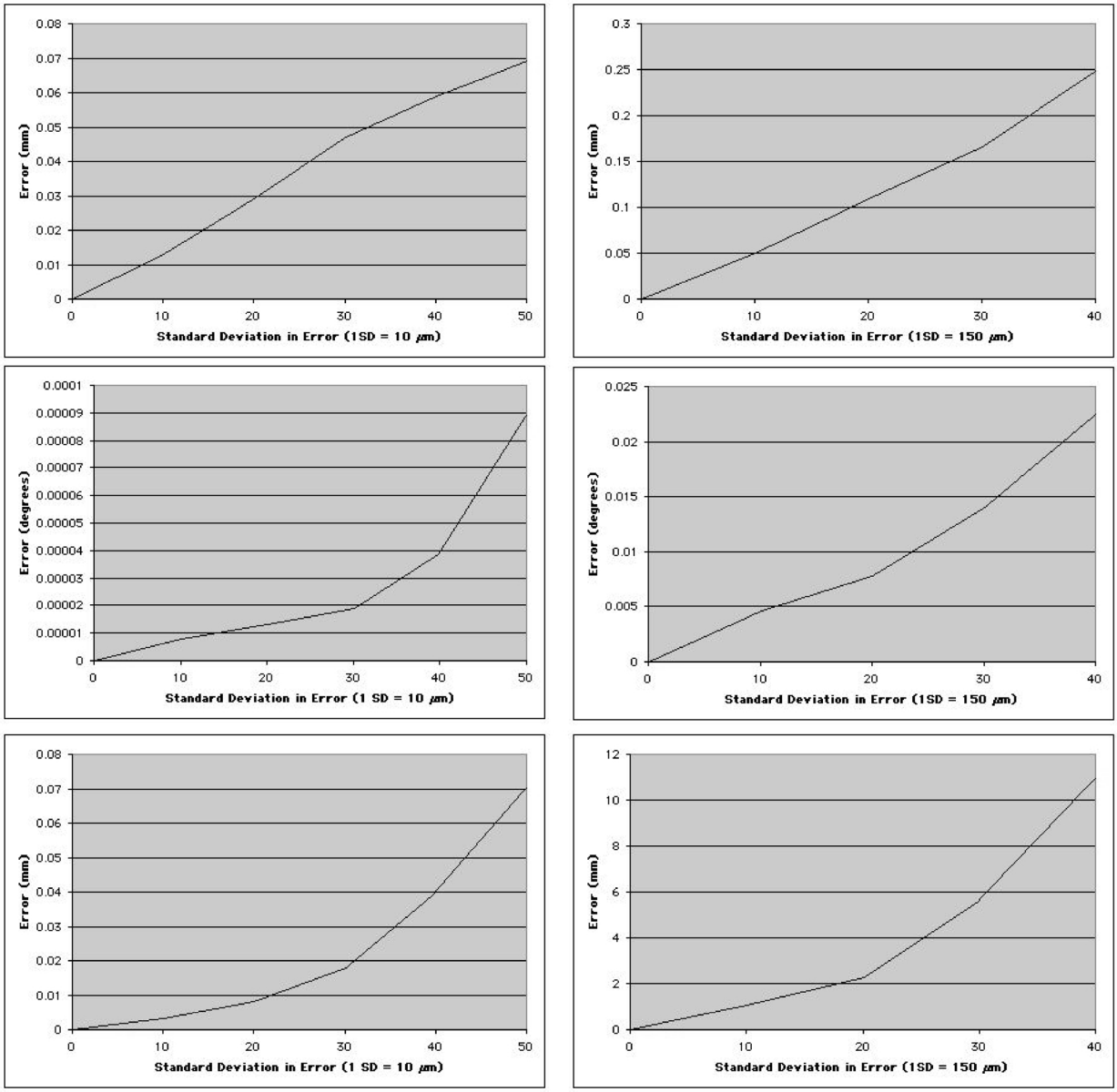

Fig. 6. Simulation 1 (left column), Simulation 2 (right column), figures per row: RMS error of plane fit, angular error in plane normal, error in plane distance at the edge of the plane.

$$[(\mathbf{p}_0 + t_0\mathbf{n}_0 - \mathbf{p}_i), (\mathbf{p}_1 + t_1\mathbf{n}_1 - \mathbf{p}_0 - t_0\mathbf{n}_0), \mathbf{n}_i] = 0. \quad (41)$$

On expansion, this simplifies to:

$$\begin{aligned}&[\mathbf{n}_0\mathbf{n}_1\mathbf{n}_i] \cdot t_0t_1 + [(\mathbf{p}_i - \mathbf{p}_1)\mathbf{n}_0\mathbf{n}_i] \cdot t_0 + \\ &[(\mathbf{p}_0 - \mathbf{p}_i)\mathbf{n}_1\mathbf{n}_i] \cdot t_1 + [(\mathbf{p}_0 - \mathbf{p}_i)(\mathbf{p}_1 - \mathbf{p}_0)\mathbf{n}_i] = 0.\end{aligned} \quad (42)$$

Now, taking $\mathbf{p}_i = \mathbf{p}_2, \mathbf{p}_3$ and $\mathbf{n}_i = \mathbf{n}_2, \mathbf{n}_3$, respectively, in turn, we get an equation system for t_0 and t_1 of the form

$$a_{01}t_0t_1 + a_0t_0 + a_1t_1 + a = 0, \quad (43)$$

$$b_{01}t_0t_1 + b_0t_0 + b_1t_1 + b = 0. \quad (44)$$

From here, we can eliminate t_0t_1 and express t_1 or t_0 as a linear function of the other. After substituting this expression for t_1 or t_0 in one of the above equations, we obtain a second degree equation which we can solve.

This leads to two solutions for the rotational axis. Various methods using further $\mathbf{p}_i$ can be used to choose between them or to find an overall best axis for the whole set of $\mathbf{p}_i$. In this application, perhaps it is simpler just to try to fit the rotational surface using both axes and select the result with the smaller least-squares residual.

7 Results

The fitting routines described in this paper were tested using the segmentation approach outlined earlier. We have tested our approach on artificial simulated data, real data obtained from a laser scanner, and also real range data made available on the Internet by Edinburgh University: ftp://ftp.dai.ed.ac.uk/pub/vision/range-images/.

7.1 Segmentation on Simulated Data

7.1.1 Studying the Effects of Noise on the Fitting Routines

In order to test the effects of noise on our fitting techniques, we tested each surface fitting method in turn on known surface types with added Gaussian noise of varying standard deviation. This

TABLE 1
Error in Least-Squares Fit (Simulation 1)

Error Measure	SD = 10	SD = 30	SD = 50
Sphere: Fit	0.013	0.035	0.083
Sphere: Curvature	5×10^{-6}	2×10^{-5}	2×10^{-4}
Sphere: Position	0.005	0.021	0.084
Cylinder: Fit	0.014	0.041	0.088
Cylinder: Curvature	2×10^{-6}	2×10^{-5}	2×10^{-4}
Cylinder: Angular Error	0.005	0.025	0.086
Cone: Fit	0.012	0.049	0.089
Cone: Curvature	8×10^{-5}	2×10^{-4}	8×10^{-4}
Cone: Angular Error	0.006	0.022	0.063
Torus: Fit	0.003	0.018	0.182
Torus: Curvature s	0.021	0.024	0.028
Torus: Curvature k	0.010	0.012	0.018
Torus: Axis	0.002	0.08	0.23

TABLE 2
Error in Least-Squares Fit (Simulation 2)

Error Measure	SD = 10	SD = 30	SD = 50
Sphere: Fit	0.06	0.173	0.31
Sphere: Curvature	0.068	0.41	0.83
Sphere: Position	0.04	0.37	0.98
Cylinder: Fit	0.07	0.155	0.30
Cylinder: Curvature	0.089	0.46	0.88
Cylinder: Angular Error	0.054	0.136	0.63
Cone: Fit	0.02	0.05	0.33
Cone: Curvature	0.01	0.43	0.91
Cone: Angular Error	0.02	0.12	0.89
Torus: Fit	0.189	0.83	2.01
Torus: Curvature s	0.05	0.2	0.86
Torus: Curvature k	0.04	0.14	0.36
Torus: Axis	0.2	1.05	3.7

was done in order to assess how well our methods might perform when used with scanners of varying accuracy.

Two simulations were used which reflect the characteristics of sources of the real data used later:

Simulation 1—this simulation reflects the accuracy of a commercial Replica 500/25H scanner (available from 3D Scanners). The scanner operates at a range of about 100 mm with a quoted accuracy of 10μm.

Simulation 2—this simulates the accuracy of Edinburgh University's research scanner. This scanner has a quoted accuracy of 150μm and operates at a range of about 700 mm.

In each simulation Gaussian noise was added, assuming a single standard deviation error of the quoted accuracy for 0-50 standard deviations in steps of 10 s.d. Depth maps were simulated at the quoted ranges.

In these experiments, surface patches were fitted to regions of depth data of known surface type. The regions were chosen to encompass many points of a fixed surface type. No segmentation was performed.

Typical simulated objects were about 500 mm in size. The radii of the test objects varied from about 10 mm to 200 mm. All results quote distance errors in mm, angular errors measured in degrees, and curvature errors in mm^{-1}.

In Simulation 1, with high accuracy data, our fitting methods are seen to perform extremely well. The errors of fit are in keeping with results obtained for fitting a plane to data of similar characteristics (Fig. 6 and Table 1).

In all cases, the estimates of curvature and position (center of sphere, axis of cylinder, cone, and torus) are good even at levels of noise well above the characteristics of the scanner.

In Simulation 2, which provides a greater test in terms of robustness to noise, our fitting methods again perform well (Fig. 6 and Table 2). The errors in overall fit, estimation of curvature, and localization are again of the same order and comparable to those for plane fitting.

It should be noted that good results were obtained even in cases where the surface fitted was only part of the complete surface (see Fig. 7). If multiple surfaces are present, then a starting region which includes points from more than one surface will clearly give poor fitting results. However, our segmentation approach is designed to overcome this problem, as shown in the next section.

7.1.2 Artificial Test Objects

In this section, we study the behavior of the segmentation *and* fitting methods *together* on some artificial test objects. We used simulated depth data obtained from our object models produced using the ACIS solid modeling package. In the case of simulated 3D point data which was accurate to five significant digits, in all cases the models recovered fitted the data to an accuracy of at least four significant digits and segmentations consistent with the underlying geometric primitives were obtained. An example of the final segmentation obtained for the test object composed of a cylinder, cone and two spheres is shown in Fig. 2.

7.2 Segmentation of Real Depth Data

We tested our approach on real 3D data from two sources: a commercial Replica scanner and data produced by Edinburgh University's Machine Vision Research Unit. The latter came from a high quality research device. The objects chosen included a variety of surfaces combined in ways typical of real, albeit simple, components for which we might want to perform reverse engineering.

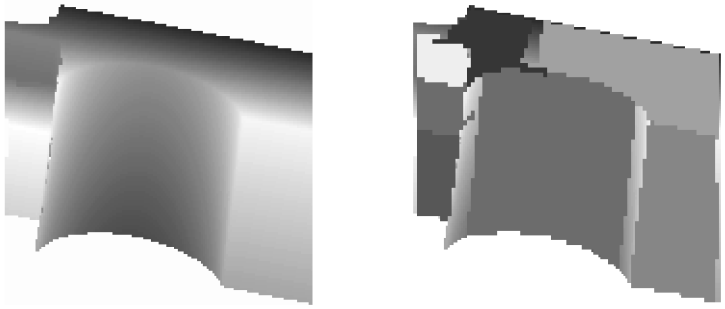

Fig. 7. Heriot-Watt part depth-map, final segmentation.

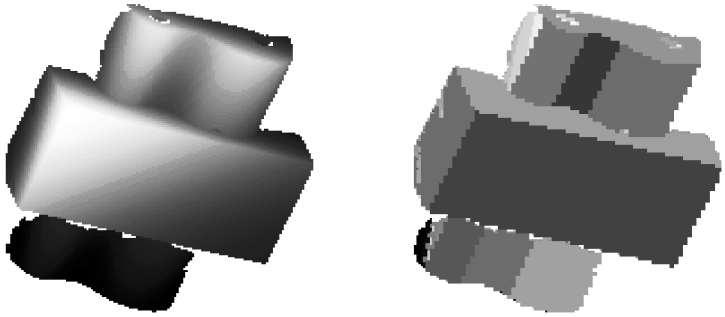

Fig. 8. RECCAD test object depth-map, final segmentation.

7.2.1 Objects Scanned by a Replica Scanner

Our fitting methods also work well in practice with real scanner data, which is somewhat less accurate than the simulated data. The segmentation process for a depth-map of the well-known "Heriot-Watt" test part [8], [27] captured using a Replica device is illustrated in Fig. 7. The segmentation results here are not as clean as for the simulated parts, but are adequate for input to the further model building processes of our reverse engineering system [26], [32], [33]. Only the uppermost surface has not been segmented cleanly—it has been segmented as two planes instead of one due to noise in the data. Most other surfaces have been segmented well and can clearly be seen in Fig. 7 to correspond to surfaces in the original depth. The main cylindrical surface has not been segmented to the observed boundaries of the underlying surface in the depth map as specular reflections often result in poor depth data in such regions. The quoted accuracy of the Replica device is of the order of $10\mu\mathrm{m}$ although, in practice, the performance achieved was rather worse than this—errors of the order of $100\mu\mathrm{m}$ occurred, together with many outlying data points caused by spurious reflections of laser scanning light off internal surfaces and object edges, accounting for the segmentation results obtained.

The error of surface fit for segmented surfaces was on the order of 0.03 mm RMS for most good surfaces of large area. Such errors in accuracy agree with our simulated studies (Section 7.1.1) where the standard deviation in error is 10-20 times the quoted accuracy. Estimates of curvature and axis, etc., are in keeping with out simulated results also. Many smaller surfaces and regions around edges were not segmented due to the large number of outliers. For this part (Fig. 7), several planar surfaces and a cylindrical surface were segmented successfully.

The "RECCAD" test object (Fig. 8) presents further challenges to a segmentation method. The curved surfaces present are actually freeform translational surfaces. Our segmentation approach finds three cylindrical surfaces which to a first approximation meet along smooth edges. As can be seen in Fig. 8, good segmentations are obtained of these main cylindrical surfaces and also of several planar surfaces. It should be noted that the segmentation does well to approximate the translational surface with three cylinders. This is the best result that could be obtained without explicitly segmenting translational surfaces (a problem tackled elsewhere in the RECCAD project [2]).

The soap dish object (Fig. 9) only has several single smooth freeform surfaces of a complex type. Here, our approach segments a torus segment for the rim and a sphere for the main part of the dish.

A final test was done on a ventilator object (Fig. 10) from which the major surface of a sphere and several planar surfaces were segmented. (In the last two objects (Figs. 9 and 10), there is an observed "blockiness" in the final segmentations. Because segmentation is time consuming, we did not perform it to the same level of detail as the depth map resolution. For model building for reverse engineering, we do not need accurate boundaries for the surfaces anyway, as they are later computed by intersection of the fitted surfaces.)

7.2.2 Edinburgh Test Objects

Finally, we present results from data from a research scanner based at Edinburgh University. Their scanner has a quoted accuracy of about $150\mu\mathrm{m}$ standard deviation, $600\mu\mathrm{m}$ maximum error. In all cases, we obtain errors of surface fit of about 0.2 mm RMS for the primary segmented surfaces. This indicates that the data obtained from our commercial scanner is more accurate. The slightly less accurate Edinburgh data, nevertheless, provides a good test for our methods. Results on two of the more complex objects from those available in the Edinburgh set objects are shown here.

The optical stand is a fairly complex part comprising several cylinders and planar surfaces. The main surfaces are successfully segmented by our method (Fig. 11).

A light bulb is the second test object. It consists of a sphere smoothly blending into a cylinder. In this final segmentation (Fig. 12), a cylinder, a sphere, and a toroidal blend between them are obtained.

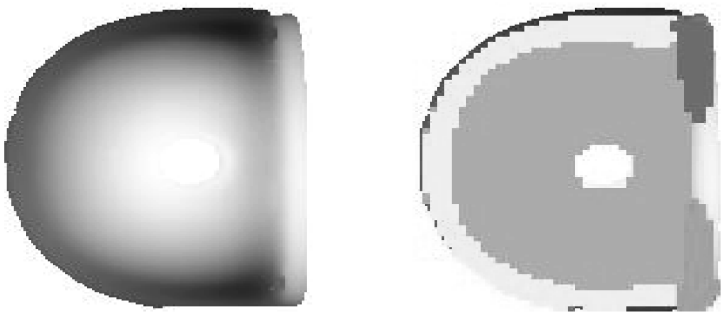

Fig. 9. Soap dish depth-map, final segmentation.

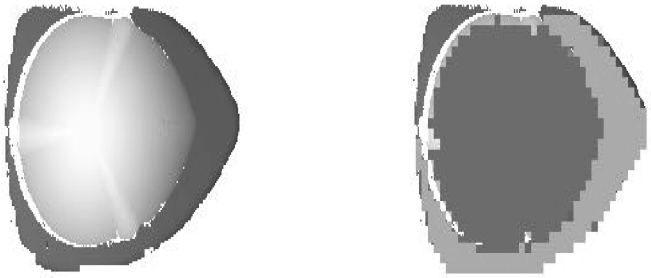

Fig. 10. Ventilator depth-map, final segmentation.

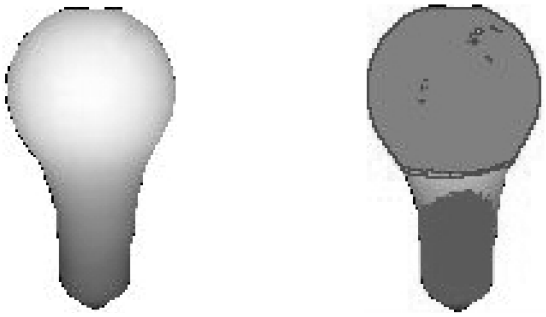

Fig. 12. Light Bulb depth-map, final segmentation.

7.3 Robustness to Degenerate Data

Our final tests on our methods aim to illustrate the inherent robustness of our methods when attempting to fit a cylinder or other surface to nearly planar regions (i.e., regions of very low curvature). In this section, we present results on artificial data and data obtained from our Replica scanner. In all the tests described here, we presented cylindrical patches of data with the radii ranging from 1,000 mm to 5,000 mm (curvature 0.001 and 0.0002, respectively) to our system. Noise was added to the artificial data as described previously, in this case, simulating the Replica scanner characteristics. Our results on real data were obtained by scanning the top surface of the housing of an Epson Stylus Color 740 printer; this surface was estimated by eye to be an approximately cylindrical region of about 1,000 mm radius.

We first analyze our experiments on artificial data. The results are summarized in Table 3 which shows RMS fitting errors in mm. The same amount (300×300 pixels) of simulated depth data were used in all tests, with 1 mm grid sampling in the x and y dimensions. Initial seed regions were 25×25 pixels (except for the 5,000 mm radius case), the regions were grown to a patch size of around 50 pixels and cylinders or planes fitted to obtain the *grow* data presented below. In practice, the only real limit on the fitting is in obtaining good initial estimates for our curved surface patches. An initial region of 25×25 pixels for 1 mm grid spacing gives good estimates for radii of sizes up to about 3,000 mm. In order to obtain good initial seed estimates for 5,000 mm radius, a seed size of 45×45 pixels had to be used. Clearly, the actual initial seed region size chosen should depend on the sampling frequency and spacing of the data and the underlying curvature of the data. Obtaining reliable estimates of curvature for such low curvatures is difficult regardless of the estimation technique utilized.

Our experiments here show that we can obtain practically useful fitting and segmentation starting with reasonable seed sizes. As can be seen in Table 3, our fitting methods for a cylinder are very robust even for large radii (low curvatures). Our initial estimates for small seed regions are good and the errors in fit for initial seed placings are similar to the final fits for the large surface area. Clearly, the fitting of planes to small seed regions yields a low error given such high curvatures and, once the seed is grown to full patch size, the errors are a result of fitting a plane to a nonplanar surface. The errors are what one would expect given the grown patch sizes.

The estimated curvatures were always correct to four significant figures even with large standard deviation in the noise.

One other experiment carried out only requires a brief discussion but further illustrates the robustness of our methods to very low or zero curvatures. If we repeat the above experiment on planar data, we always obtain estimates of curvature that are within a numerical tolerance of zero—cylinder fitting returns a plane in this manner. The planes also have accurate fits and will always be selected by our information theoretic criteria that less parameters are required to represent a plane.

In order to demonstrate that our methods work well on real data with low curvature, we also experimented with scanned data from the gently curved housing of an Epson printer. We attempted to fit a plane and cylinder to the data. For a small (25×25) initial seed region, we obtained an initial fit error of 0.0014 mm for a plane and a cylinder. After growing the region to fit the whole of the area, we obtained errors of 0.538 mm for a plane and 0.145 mm for a cylinder fit. The estimate of curvature for the final cylinder was 0.0011 which implies a radius of about 909 mm. While not having ground truths to compare with these results, our estimates by hand suggest that this curvature is of the right order. However, observations by the human eye also indicate that while the surface has a very low curvature, it is clear that it is not truly cylindrical which accounts for the error in fit for the cylindrical patch. Nonetheless, this experiment demonstrates that our methods are robust with respect to low curvature data and work well on real data.

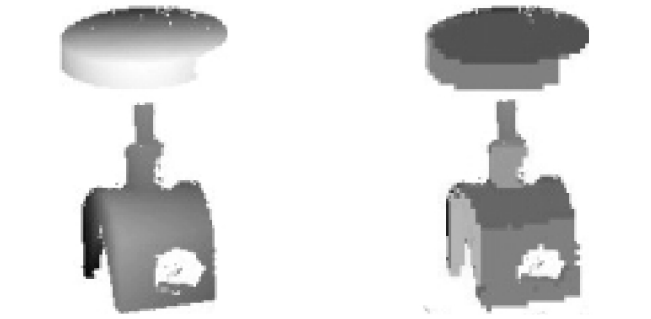

Fig. 11. Optical Stand depth-map, final segmentation.

TABLE 3
Error in Least Squares Fit for Low Curvature

Radius	**SD = 0**		**SD = 1**		**SD = 10**	
Region Fit	*seed*	*grow*	*seed*	*grow*	*seed*	*grow*
1000 mm:						
Plane	0.020	0.985	0.021	0.991	0.041	1.010
Cylinder	0.000	0.000	0.002	0.003	0.021	0.026
2000 mm :						
Plane	0.01	0.626	0.014	0.627	0.003	0.685
Cylinder	0.000	0.000	0.002	0.002	0.020	0.025
2500 mm :						
Plane	0.01	0.580	0.014	0.582	0.032	0.585
Cylinder	0.00	0.000	0.002	0.002	0.029	0.30
3000 mm:						
Plane	0.01	0.505	0.012	0.507	0.024	0.510
Cylinder	0.00	0.00	0.002	0.002	0.022	0.029
5000 mm						
Plane	0.003	0.280	0.010	0.288	0.021	0.298
Cylinder	0.00	0.00	0.002	0.002	0.024	0.028

7.4 Summary of Results

We have presented a series of practical examples demonstrating our new surface fitting methods and their use for segmentation. The artificial results have shown that our fitting methods work very well, providing primitives which are good fits to a variety of surface types. The tests on artificial data also show that our methods are robust in the presence of noise and that the quality of surface fits produced from real data is consistent with our simulated data test.

The results on real test objects, obtained from two different scanners, show that our methods work successfully in practical environments and are accurate for their intended tasks.

We have also clearly demonstrated that our methods are robust and handle degeneracy in both estimating and fitting surfaces with very low curvature.

8 Conclusions

While our motivation is the reverse engineering of boundary representation solid models from 3D depth-maps of scanned objects, we believe that the fitting methods described in this paper are of interest to the computer vision and CAD communities in general.

In summary, we have described novel methods for the least-squares fitting of spheres, cylinders, cones, and tori to point data. Generalizing these methods to more complex surfaces would be hard, if not impossible. However, many simple engineering objects are bounded by just these surfaces (and blends).

We have outlined how these methods can be used in a segmentation strategy that is capable of extracting these surfaces from 3D data. Initial results show that the accuracy achieved by these methods is good. Our fitting methods have the major advantage of being robust in the sense that as the principal curvatures of the surfaces being fitted decrease (or as they become more equal), the results which are returned naturally become closer and closer to surfaces of "simpler type," i.e., planes, cylinders, or cones (or spheres, in the case of equal curvatures) which best describe the data. Furthermore, our methods inherently avoid all singularities (except in the case of the torus, for which the problem can readily be overcome by choosing the origin appropriately).

Acknowledgments

The authors gratefully acknowledge the support of EC COPERNICUS grant *RECCAD* No. 1,068 which enabled Gabor Lukacs to visit Cardiff University.

References

[1] R. Bajcsy, F. Solina, and A. Gupta, "Segmentation versus Object Representation—Are they Separable?" *Analysis and Interpretation of Range Images,* R. Jain and A.K. Jain, eds., New York: Springer-Verlag, 1990.

[2] P. Benko, G. Kos, and T Varady, "Detecting Translational and Rotational Symmetries in Reverse Engineering," *Proc. Advanced Workshop Confluence of Computer Vision and Computer Graphics,* Aug. 1999.

[3] P.J. Besl, *Surfaces in Range Image Understanding.* New York: Springer-Verlag 1988.

[4] P.J. Besl and R.K. Jain, "Segmentation through Variable-Order Surface Fitting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 10, no. 2, pp. 167-192, Feb. 1988.

[5] R.M. Bolle and D.B. Cooper, "On Optimally Combining Pieces of Information, with Application to Estimating 3D Complex-Object Position from Range Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 5, pp. 619-638, May 1986.

[6] R.M. Bolle and B.C. Vemuri, "On Three-Dimensional Surface Reconstruction Methods," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 1, pp. 1-13, Jan. 1991.

[7] Å. Björk, *Numerical Methods for Least Squares Problems.* Philadelphia: SIAM, 1996.

[8] J. Clark, E. Trucco, and H.F. Cheung, "Using Light Polarization in Laser Scanning," *Image and Vision Computing,* vol. 15, pp. 107-117, 1997.

[9] O.D. Faugeras, M. Hebert, and E. Pauchon, "Segmentation of Range Data into Planar and Quadric Patches," *Proc. Third Computer Vision and Pattern Recognition Conf.,* pp. 8-13, 1983.

[10] A.W. Fitzgibbon, M. Pilu, and R.B. Fisher, "Direct Least-Square Fitting of Ellipses," *Proc. 13th Int'l Conf' Pattern Recognition,* Aug. 1996.

[11] W. Gander, G.H. Golub, and R. Strebel, "Least-Squares Fitting of Circles and Ellipses," *BIT,* vol. 34, pp. 558-578, 1994.

[12] M. Hebert and J. Ponce, "A New Method for Segmenting 3D Scenes into Primitives," *Proc. Sixth Int'l Conf. Pattern Recognition,* pp. 836-838, Oct. 1982.

[13] A. Jaklic, A. Leonardis, and F. Solina, "Segmentor: An Object-Oriented Framework for Image Segmentation," Technical Report LRV-96-2, Computer Vision Laboratory, Faculty of Computer and Information Science, Univ. of Ljubljana, 1996.

[14] D. Keren and C. Gotsman, "Fitting Curves and Surfaces with Constrained Implicit Polynomials," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 1, pp. 31-41, Jan. 1999.

[15] A. Leonardis, A. Gupta, and R. Bajcsy, "Segmentation as the Search for the Best Description of the Image in Terms of Primitives," *Proc. Third Int'l Conf. Computer Vision,* pp 121-125, 1990.

[16] A. Leonardis, "Image Analysis Using Parametric Models: Model-Recovery and Model-Selection Paradigm," PhD dissertation, Faculty of Electrical Eng., and Computer Science, Univ. of Ljubljana, May 1993.

[17] A. Leonardis, A. Gupta, and R. Bajcsy, "Segmentation of Range Images as the Search for Geometric Parametric Models," *Int'l J. Computer Vision,* vol. 14, pp. 253-277, 1995.

[18] P. Liong and J.S. Todhunter, "Representation and Recognition of Surface Shapes in Range Images: A Differential Geometry Approach," *Computer Vision, Graphics, and Image Processing,* vol. 52, no. 1, pp. 78-109, 1990.

[19] G. Lukács, A.D. Marshall, and R.R. Martin, "Geometric Least-Squares Fitting of Spheres, Cylinders, Cones, and Tori," RECCAD, Deliverable Document 2 and 3, COPERNICUS project, no. 1,068 (Budapest) Geometric Modelling Laboratory Studies/1997/5, Computer and Automation Research Institute, Budapest, R.R. Martin and T. Várady, eds., July 1997. http://www.cs.cf.ac.uk/Dave/3DVG/FITTING.

[20] G. Lukács, A.D. Marshall, and R.R. Martin, "Faithful Least-Squares Fitting of Spheres, Cylinders, Cones, and Tori for Reliable Segmentation," *Proc. Fifth European Conf. Computer Vision (ECCV '98),* H. Burkhardt and B. Neumann eds., vol. I, pp. 671-686, 1998.

[21] H. Pottmann and T. Randrup, "Rotational and Helical Surface Approximation for Reverse Engineering," *Computing,* vol. 60, pp. 307-322, 1998.

[22] V. Pratt, "Direct Least-Squares Fitting of Algebraic Surfaces," *Proc. Ann. Conf. Series Computer Graphics,* vol. 21, no. 4, pp. 145-152, July 1987.

[23] P.L. Rosin, "A Note on the Least Squares Fitting of Ellipses," *Pattern Recognition Letters,* vol. 14, pp. 799-808, 1993.

[24] P.L. Rosin, "Analysing Error of Fit Functions for Ellipses," *Pattern Recognition Letters,* vol. 17, pp. 1461-1470, 1996.

[25] G. Taubin, "Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 13, no. 11, pp. 1115-1138, Nov. 1991.

[26] W. Thompson, J. Owen, J. de St. Germain, S. Stark, and T. Henderson, "Feature-Based Reverse Engineering of Mechanical Parts," *IEEE Trans. Robotics and Automation,* vol. 15, no. 1, pp. 1-9, 1999.

[27] E. Trucco, R.B. Fisher, A.W. Fitzgibbon, and D.K. Naidu, "Calibration, Data Consistency, and Model Acquisition with a 3D Laser Striper," *Int'l J. Computer Integrated Manufacturing,* vol. 11, no. 4, pp. 293-320, 1998.

[28] N. Werghi, R.B. Fisher, A. Ashbrook, and C. Robertson, "Improving Model Shape Acquisition by Incorporating Geometric Constraints," *Proc. British Machine Vision Conf. (BMVC '97),* A. Clark, ed., pp. 520-529, 1997.

[29] N. Werghi, R.B. Fisher, C. Robertson, and A. Ashbrook, "Modelling Object Having Quadric Surfaces Incorporating Geometric Constraints," *Proc. Fifth European Conf. Computer Vision (ECCV '98),* H. Burkhardt and B. Neumann, eds., vol II, pp. 185-201, 1998.

[30] N. Werghi, R.B. Fisher, C. Robertson, and A. Ashbrook, "Towards Object Modelling by Incorporating Geometric Constraints," *Proc. IEEE Workshop Model-Based 3D Image Analysis,* E. Cuchet and G. Subsol, eds., pp. 45-53, 1998.

[31] N. Werghi, R.B. Fisher, A. Ashbrook, and C. Robertson, "Faithful Recovering of Quadric Surface from 3D Range Data," *Proc. Second Int'l Conf. 3D Digital Imaging and Modelling,* M. Rioux, P. Boulanger, and D. Laurendeau, eds., pp. 45-53, 1999.

[32] N. Werghi, R.B. Fisher, A. Ashbrook, and C. Robertson, "Object Reconstruction by Incorporating Geometric Constraints in Reverse Engineering," *Computer Aided Design,* vol. 31, pp. 363-399, 1999.

[33] T. Varady, R.R. Martin, and J. Cox, "Reverse Engineering of Geometric Models—An Introduction," *Computer Aided Design,* vol. 29, no. 4, pp. 255-268, 1997.