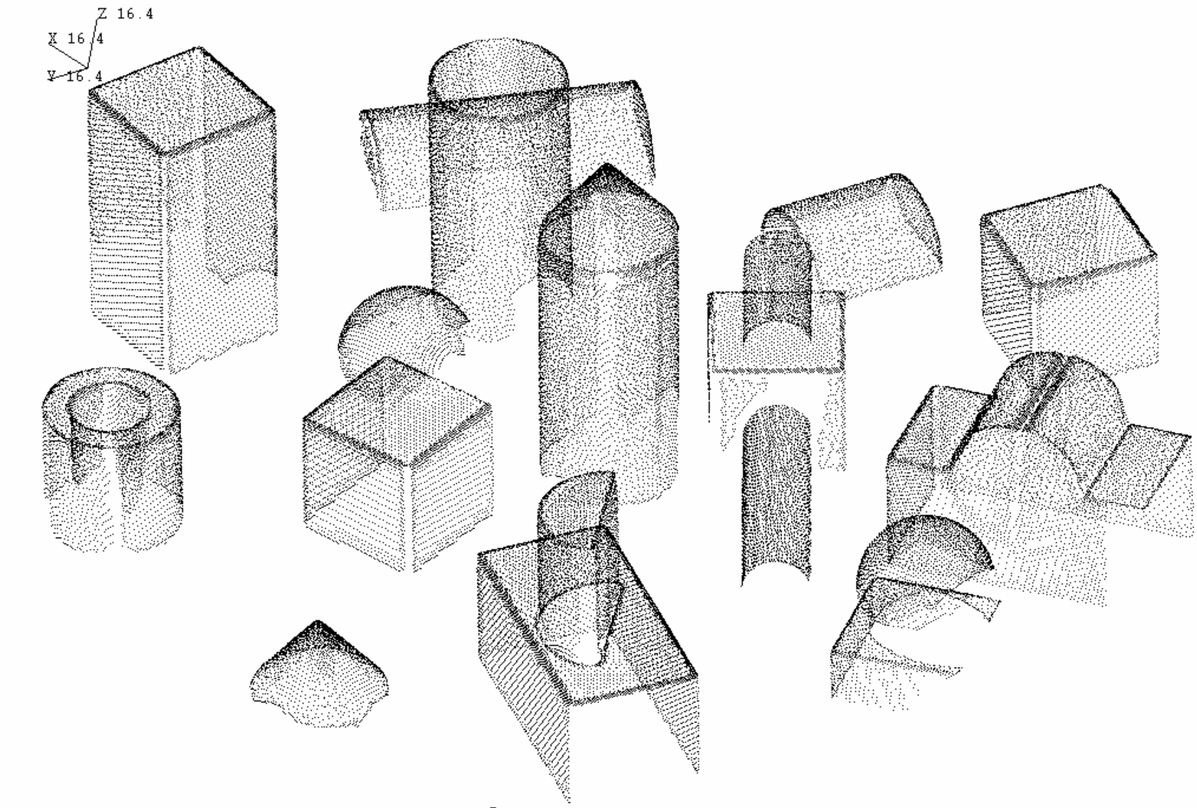
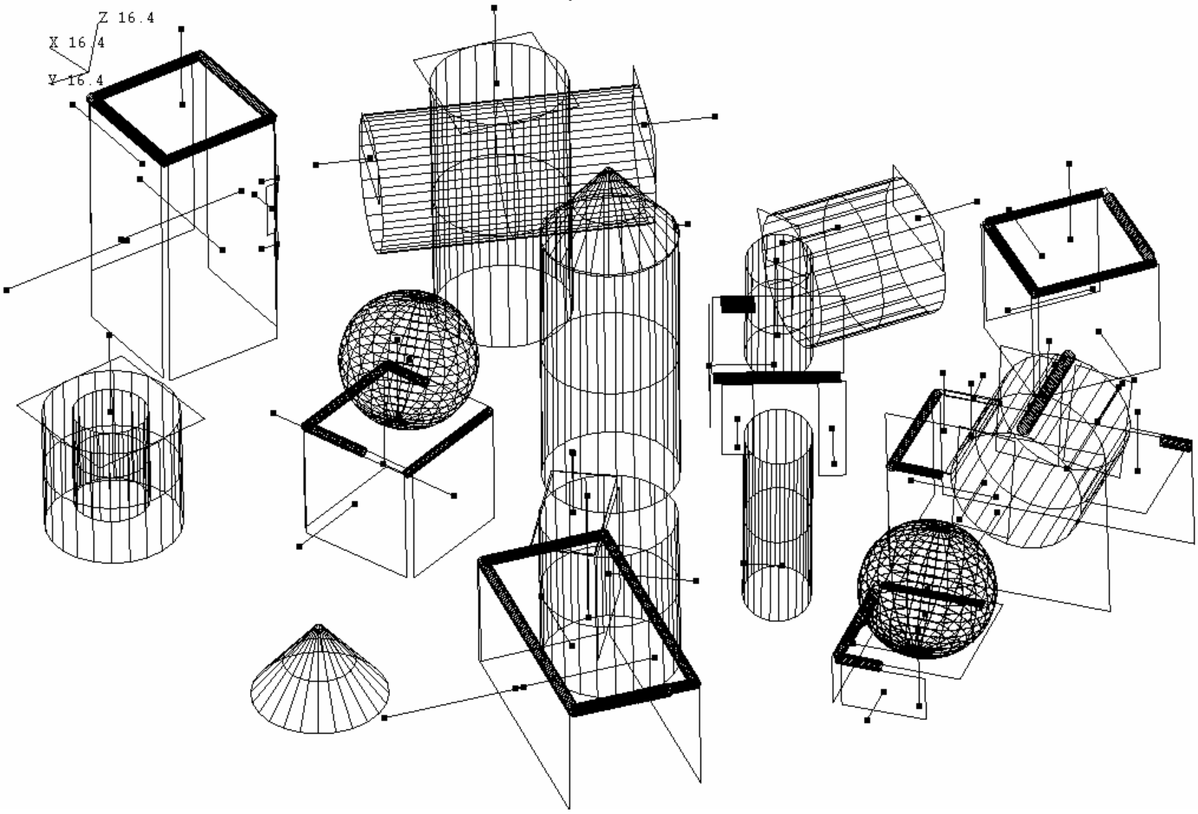
8th Int. Symp. on Measurement and Quality Control in Production (ISMQC)
Erlangen, Germany, Oct. 12–15, 2004
Session: B2 Coordinate Measuring Machines 2

“**Automatic Feature Identification in 3-D Measuring Data**”

S.J. Ahn, I. Effenberger and L. Bolboaca
Fraunhofer IPA, Stuttgart, Germany





8th Int. Symp. on Measurement and Quality Control in Production, Erlangen, Germany, Oct. 12–15, 2004

Automatic Feature Identification in 3-D Measuring Data

S.J. Ahn, I. Effenberger, L.I. Bolboaca

Fraunhofer Institute for Manufacturing Engineering and Automation (IPA), Stuttgart, Germany

Abstract:
The automatic feature identification in 3-D measuring data is of great interest in many application fields e.g. metrology, computer vision or reverse engineering. In this paper we present a software tool for the fully automatic object detection and parameter estimation in unordered incomplete and even noisy point clouds with a large number of points. The software consists of three interactive modules each for model selection, point segmentation and model fitting, in which our newly developed algorithms for orthogonal distance fitting (ODF) play an important role. The ODF algorithms estimate the model parameters by minimizing the square sum of the shortest distances between the model feature and the measurement points. The local quadric surface fitted through ODF to a randomly touched small initial patch of the point cloud provides the necessary initial information for the overall procedures of model selection, point segmentation and model fitting. The performance of the presented software tool will be demonstrated on different point clouds.

1. Introduction

A fully automatic and generally applicable solution to *Parametric model recovery* (PMR) might be realized only through a very sophisticated hardware and software technique analyzing all the available information on the objects, such as point cloud, object database, object surface color and texture. In this paper we strive to increase the accuracy and the automation degree of PMR by exploiting the immediately available information on the objects namely the 3-D point cloud.

If we restrict our interest field to the industrial environment, we find out that a large portion of industrial objects including manufacturing facilities and work pieces can be modeled as *exact features*, i.e. planes, spheres, cylinders or cones [6]. Thus, even when we limit the range of our interest model features to geometric primitives, there is still a large demand on the fully automatic identification of these features in fields like reverse engineering, robotics or digitizing plants.

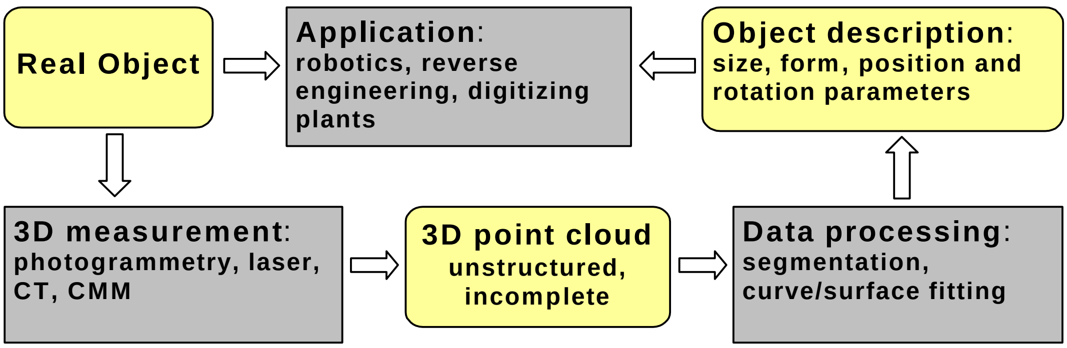


Figure 1: Parametric Model Recovery of real objects via 3-D point cloud

Under the circumstances mentioned above, we have developed a software tool for the fully automatic extraction of exact features from point cloud, based on our previous work on a semi-automatic solution [2]. The functionality of the software tool is analogous to that of the human intelligence searching for the objects of exact feature in a dark room environment. In this paper we describe in detail the algorithmic techniques implemented in our software tool. Beside the experimental result given in this paper, the performance of the software tool on a variety of point clouds generated by different 3-D measuring techniques is demonstrated.

2. Parametric Model Recovery

Real Object, Point Cloud and Model Feature

Before the algorithmic details of the software tool are described we like to survey the conditions under which the parametric model recovery (PMR) from point cloud takes place (Fig. 1). To bear in mind is:

- The point cloud is subject to measurement errors;
- The model feature represents only roughly the true surface of the real object;
- The model parameters are to be estimated from the point cloud, of which results are subject to the applied estimation method.

This means all the three passages between the three parties (real object, point cloud, and model feature) cause inevitably errors in PMR comprised of 3-D measurement, model selection and parameter estimation. In order to layout the feasible technical solution to this inconvenient situation, we review the very fact of PMR.

Model Features for Real Object

For representing an object surface we can consider two ways namely a facet model or an analytic model. The facet model consists of a set of polygons. Although the facet model is suitable for object visualization and lithography, it does not provide applications with the information on shape, size, position, and orientation of the object. The analytic model describes an object surface through mathematical formulas with an appropriate set of model parameters, which is adopted by our software tool.

There are three description forms of the analytic model (curve/surface), i.e., explicit, implicit and parametric form [3], [4], [8]. In general, diverse applications handling dimensional models or objects use the implicit or the parametric form. In addition, many applications, e.g. the bin-picking and the obstacle-avoidance task in robotics, involve describing the real objects in terms of shape, size, position, and orientation. Thus, we group the model parameters of a curve/surface into form $\mathbf{a}_g$, position $\mathbf{a}_p$, and rotation parameters $\mathbf{a}_r$ as follows:

$$\begin{cases} f(\mathbf{a}_\mathrm{g}, \mathbf{x}) = 0 & \text{: implicit feature} \\ \mathbf{x}(\mathbf{a}_\mathrm{g}, \mathbf{u}) & \text{: parametric feature} \end{cases} \tag{1}$$

$$\mathbf{X} = \mathbf{R}^{-1}_{w,j,k}\mathbf{x} + \mathbf{X}_\mathrm{o} \quad \text{or} \quad \mathbf{x} = \mathbf{R}_{w,j,k}(\mathbf{X} - \mathbf{X}_\mathrm{o})\,, \tag{2}$$

$$\mathbf{a}^\mathrm{T} \equiv (\mathbf{a}^\mathrm{T}_\mathrm{g}, \mathbf{a}^\mathrm{T}_\mathrm{p}, \mathbf{a}^\mathrm{T}_\mathrm{r}) = (a_1, \ldots, a_l, X_\mathrm{o}, Y_\mathrm{o}, Z_\mathrm{o}, w, j, k)\,. \tag{3}$$

The form parameters represent the shape and size of the canonical model feature (1) defined in model coordinate frame xyz and are invariant to the rigid body motion (2) of the model feature in reference coordinate frame XYZ. Our software tool extracts the model features of exact feature from the given point cloud and estimates their model parameters in terms of form, position, and rotation parameters (3).

3-D Point Cloud

The optical 3-D measuring devices available on the market can generate millions of dense 3-D points in a few seconds [10]. However, the point cloud is usually not dense enough to cover the details of the object surface. This means we should hold back from under-sampling the point cloud. And, it is generally assumed that the point cloud is not ordered. Furthermore, because of the limited accessibility of the measuring devices to the object surface, the point cloud covers only partially the object surface. On the other hand, with the point cloud generated by CT technology enjoying a full accessibility to the object surface, the segmentation of the closely neighboring object surfaces is a challenging task. Our software tool can handle the unordered incomplete and complex point cloud with a large number of data points.

A measurement point is the probable observation of an unknown *nearest* point on the object surface to the measurement point [1]. The distance between the measurement point and the unknown object point is the true measurement error. In practice, because the true object surface is unknown, it is substituted by the associating model feature [6] and the true measurement error is substituted by the minimum distance (geometric distance, Euclidean distance) between the model feature and the measurement point. This error definition solely outlines the algorithmic functionalities which should be implemented in the software tool for a reliable and accurate PMR from point cloud. The minimum distance should be used not only as the decision measure between the inliers and the outliers of the model feature (segmentation), but also as the error measure to be minimized by the estimation of model parameters (model fitting) [1], [6]. Although the calculation and minimization of the minimum distances are computationally expensive, they are of vital importance to a reliable and accurate PMR from point cloud.

Orthogonal Distance Fitting

We briefly describe the *orthogonal distance fitting* (ODF) that estimates the model parameters by minimizing the square sum of the error distances between the model feature and the given points. Interested readers are referred to [3] for the complete description of the ODF algorithms that estimate the model parameters in terms of form, position, and rotation parameter (1)–(3).

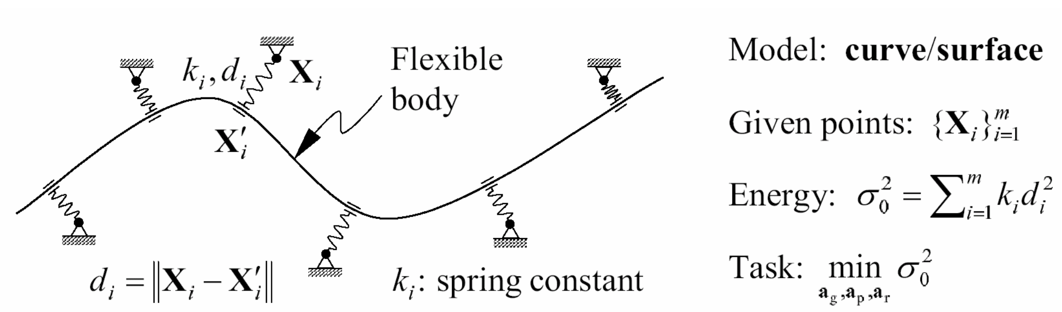


Fig. 2: Interpretation of the orthogonal distance fitting as an energy minimization problem

The ODF task can be interpreted as an energy minimization problem illustrated in Fig. 2, with which the energy (cost) function is defined as:

$$s_0^2 \equiv (\mathbf{X}-\mathbf{X}')^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\mathbf{P}(\mathbf{X}-\mathbf{X}') \quad \text{or} \quad s_0^2 \equiv \mathbf{d}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{d} \quad , \tag{4}$$

where the vectors $\mathbf{X}^{\mathrm{T}}=(\mathbf{X}^{\mathrm{T}}_1, \ldots, \mathbf{X}^{\mathrm{T}}_m)$ and $\mathbf{X}'^{\mathrm{T}}=(\mathbf{X}'^{\mathrm{T}}_1, \ldots, \mathbf{X}'^{\mathrm{T}}_m)$ are the coordinate row vectors of the m given points and of the m corresponding points on the model feature, respectively. $\mathbf{d}^{\mathrm{T}}=(d_1, \ldots, d_m)$ is the distance row vector with $d_i = ||\mathbf{X}^{\mathrm{T}}_i - \mathbf{X}'^{\mathrm{T}}_i||$. The diagonal elements of the weighting matrix $\mathbf{P}^{\mathrm{T}}\mathbf{P}$ correspond to the spring constants $\{k_i\}^m_{i=1}$ in Fig. 2. To minimize the cost functions (4) the ODF algorithm minimizes not only the square sum but also every single distance $\{d_i\}^m_{i=1}$ between the model feature and the given points. Because the minimum distances $\{d_i\}^m_{i=1}$ are nonlinear to the model parameters, the ODF task is inherently a *nonlinear minimization problem* that must be solved through iteration. The computing cost and the memory space usage of the ODF algorithms in [3] are proportional to the number of data points, thus the algorithms are suitable for processing a massive point cloud. By investigating the resulting cost (4) and the parameter covariance matrix, we can test the overall performance and reliability of the model selection and model fitting.

3. Fully Automatic Feature Extraction

For a given set of points the feature extraction procedure consists of two substantial sessions of segmentation and model fitting, respectively (Fig. 1). At this point we confront a chicken-and-egg dilemma. Namely, without the geometric information on the model feature we cannot decide between the inliers and the outliers of the model feature (segmentation), and reversely, without the inliers we cannot get the geometric information on the model feature (model fitting). To resolve this information deadlock, either of the two sides should provide the seed information triggering the other side.

Model Selection and Initial Model Parameters

We obtain the geometric seed information on the model feature from a small patch of point cloud, which is comparable with touching an object then guessing its geometry in a dark room environment:

1. Cut (touch) a small initial patch from the point cloud.
2. Fit a plane to the patch through the moment method (non-iterative linear ODF) [7].
3. Fit a quadric surface to the patch through ODF starting from the plane parameters.
4. Get the orthogonal footing point on the surface from the mass center of the patch.
5. Calculate the surface normal, principal curvatures and axes at the footing point [5].
6. Choose the model type for the patch by analyzing the signed curvature radii (Fig. 3).
7. Derive the initial values for size, center, and orientation of the chosen model feature from the surface normal, curvature radii and principal axes at the footing point.
8. Fit the initial model feature to the initial small patch through ODF starting from the model parameters derived in the last step.

The classification of the local surface types (Fig. 3a [5]) according to the local curvatures (including the mean and Gaussian curvatures) of a curved surface fitted to a small point patch is known in literature [9]. Instead of the curvatures, our software tool employs the curvature radii which correspond to the feature radius (Fig. 3b). During the above procedure,

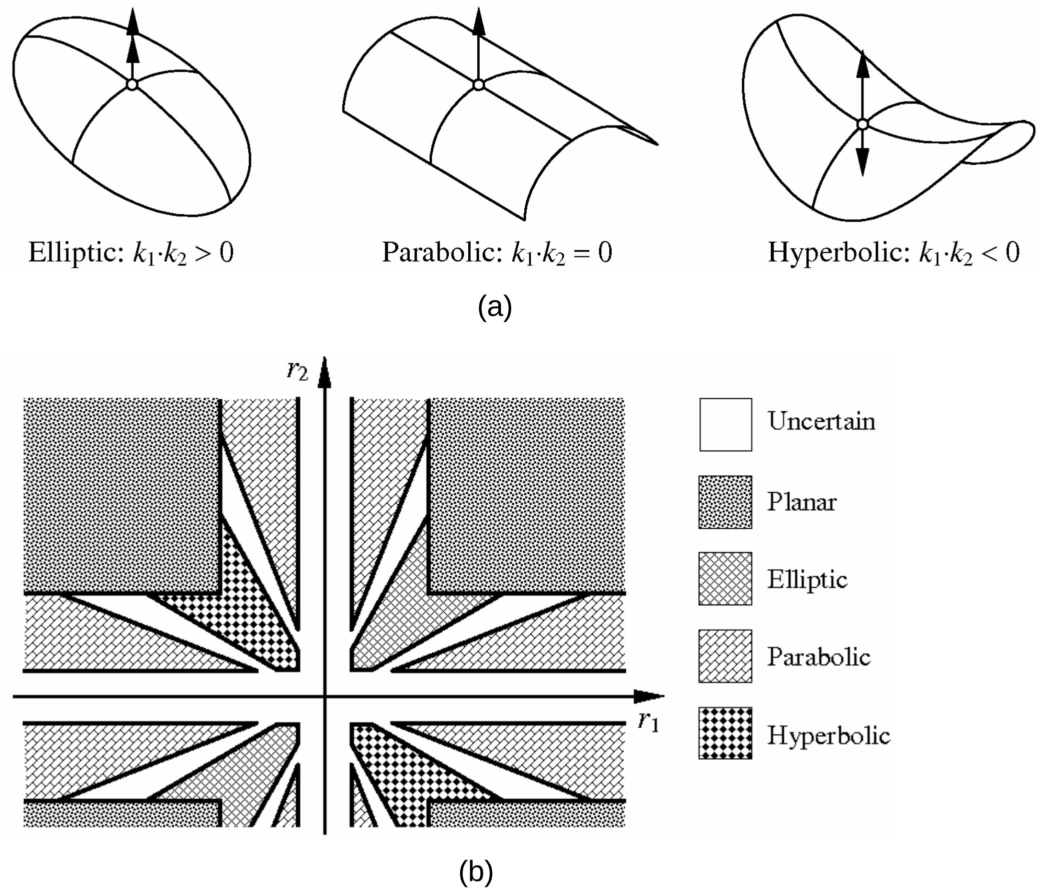


Fig. 3: Classification of local surface types according to the two principal curvatures k_1 and k_2. (a) Flat for plane, elliptic for sphere/torus, parabolic for cylinder/cone, and hyperbolic for torus [5]; (b) Curvature radius map for local surface types ($r_1 = 1/k_1$, $r_2 = 1/k_2$)

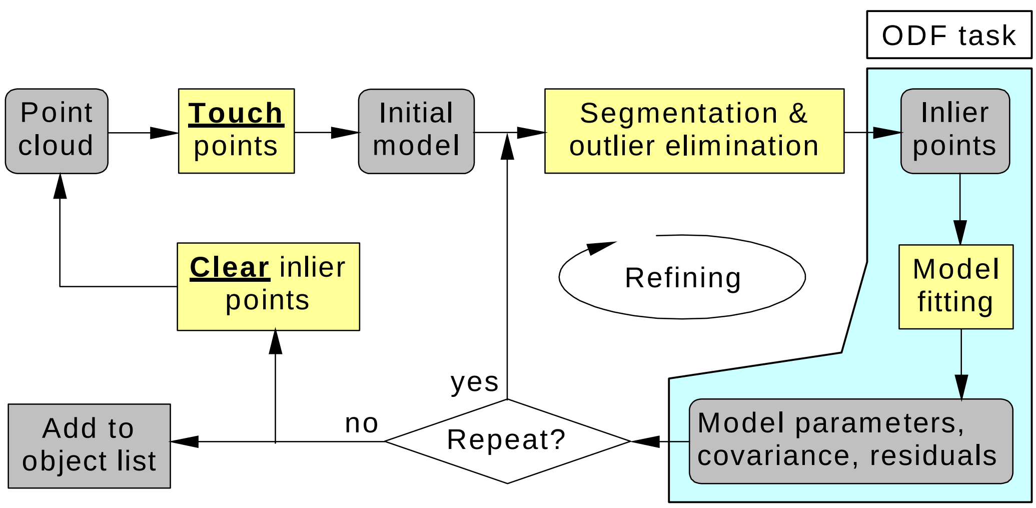


Figure 4: “Touch & Clear” in point cloud for automatic feature extraction

the ODF algorithm plays an important role in determining the parameters of the intermediate quadric surface and of the initial model feature from the small initial patch. Other fitting algorithms than the ODF algorithm, which minimize some error measures other than the minimum distance, are prone to fail to fit a model feature to a small point patch.

Overall Process and Experimental Result

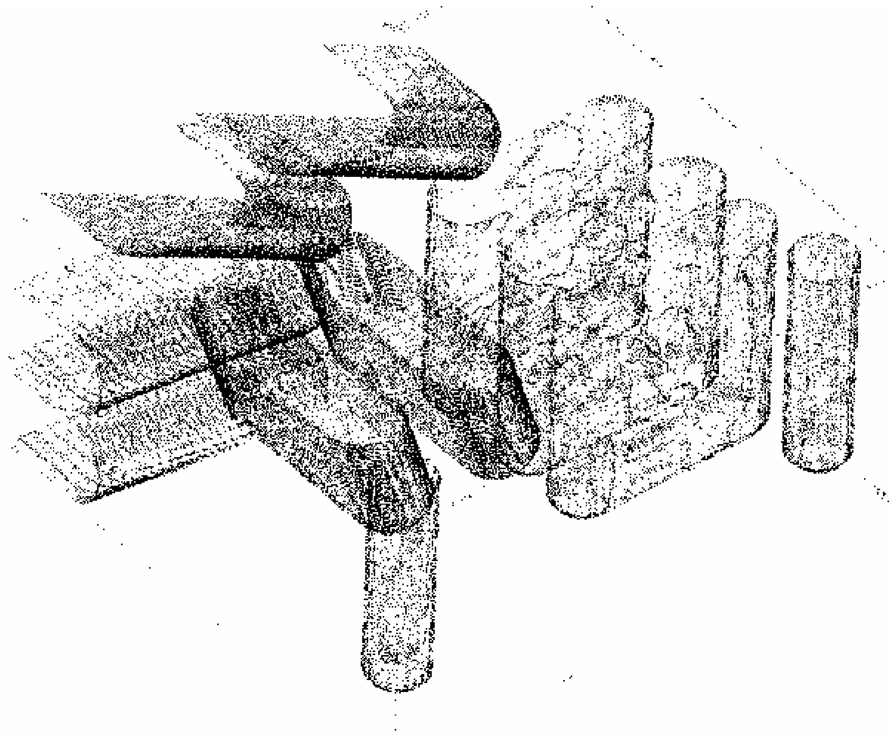
Once the model type and parameters are initialized, the interaction loop between the segmentation and the model fitting can be triggered. As noted in Sect. 2, the minimum distance of a given point to the model feature should be used as the decision measure whether a point is inlier point of the model feature. However, with regard to a specific model feature, the large part of a point cloud is occupied by plain outliers, causing a high computing cost of unnecessarily calculating the minimum distances. Through utilizing the parameter grouping (1)–(3) and the properties of the implicit model description, we can efficiently eliminate the plain outliers from the point cloud. The overall process of the automatic feature extraction can be described below (see Fig. 4):

1. Initialize the model feature (model type, size, position, and orientation).
2. Put a domain box enclosing the interest volume of the model feature in xyz frame.
3. Stamp all the points lying outside the domain box as plain outlier.
4. Except for the inlier candidates lying between the two (inner and outer) iso-features of the model feature, stamp all points as plain outlier.
5. Evaluate the rms distance of the inlier candidates to the model feature.
6. Stamp only the inlier candidates as inlier, of which distances to the model feature are not larger than 2–3 times the rms distance.
7. Update the model parameters through ODF to the inliers.
8. If necessary, repeat from the second step ('Refining' in Fig. 4).
9. Save the model parameters, and, clear the inliers from the point cloud.
10. Repeat from the first step until no more dense point patch can be found (touched).

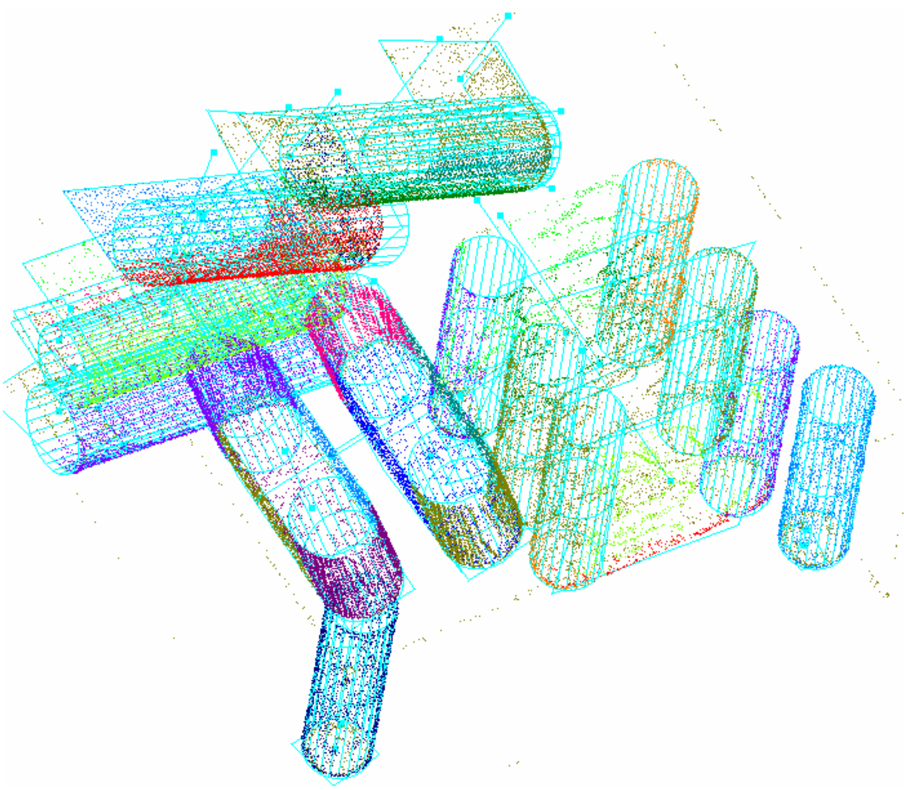
As an experimental example for the automatic feature extraction from point cloud, we applied our software tool to a point cloud generated by X-ray CT (Fig. 5a). All the relevant model features could be extracted fully automatically and correctly (Fig. 5b).

4. Summary

We have developed a software tool for the fully automatic extraction of geometric primitives from unordered incomplete and error-contaminated 3-D point clouds. The necessary information for model selection, segmentation, and model fitting could be obtained from an local quadric surface fitted to a small initial patch of the point cloud. The geometric error measure is of vital importance to both the segmentation and the model fitting, although the required computing cost is relatively high. In order to save the computing cost of the segmentation, we exploited the parameter grouping and the properties of the implicit model description. We demonstrated the outstanding performance of the software tool on a set of real measurement points generated by X-ray CT.



(a)



(b)

Figure 6: Feature extraction from a point cloud. (a) Unordered and incomplete point cloud of X-ray CT technology; (b) Fully automatically extracted exact features

5. References

[1] Adcock, R.J.: Note on the method of least squares. The Analyst **4** (1877) 183–184

[2] Ahn, S.J., Effenberger, I., Roth-Koch, S., Westkämper, E.: Geometric Segmentation and Object Recognition in Unordered and Incomplete Point Cloud. Lect. Notes Comput. Sc. **2781** (2003) 450–457

[3] Ahn, S.J.: Least-Squares Orthogonal Distance Fitting of Curves and Surfaces in Space. Dissertation, University of Stuttgart. Lect. Notes Comput. Sc. **3151** (2004) 125 pages

[4] Bolle, R.M., Vemuri, B.C.: On Three-Dimensional Surface Reconstruction Methods. IEEE Trans. Pattern Analy. and Mach. Intell. **13** (1991) 1–13

[5] do Carmo, M.P.: Differential Geometry of Curves and Surfaces. Prentice-Hall (1976)

[6] ISO 10360-6:2001: Geometrical Product Specification (GPS) - Acceptance test and reverification test for coordinate measuring machines (CMM) - Part 6: Estimation of errors in computing Gaussian associated features. ISO (2001)

[7] Pearson, K.: On Lines and Planes of Closest Fit to Systems of Points in Space. The Philosophical Magazine: Ser. 6 **2** (1901) 559–572

[8] von Seggern, D.H.: CRC standard curves and surfaces. 2nd Ed., CRC Press (1993)

[9] B.C. Vemuri, A. Mitiche, J.K. Aggarwal: Curvature-based representation of objects from range data. Image and Vision Computing **4** (1986) 107–114

[10] www.pobonline.com: 2004 3D Laser Scanner Hardware and Software Survey. Point of Beginning, January (2004) 20–25