HAL
open science

Extracting cylinders in full 3D data using a random sampling method and the Gaussian image

Thomas Chaperon, Francois Goulette

To cite this version:

Thomas Chaperon, Francois Goulette. Extracting cylinders in full 3D data using a random sampling method and the Gaussian image. Vision Modeling and Visualization Conference 2001 (VMV-01), Nov 2001, Stuttgart, Germany. hal-01259641

HAL Id: hal-01259641

https://hal.science/hal-01259641

Submitted on 21 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Extracting cylinders in full 3D data using a random sampling method and the Gaussian image

Thomas Chaperon[1,2] François Goulette[1]

[1] Centre de Robotique
Ecole Nat. Sup. des Mines de Paris
60 bd Saint-Michel
75272 Paris cedex 06
http://www-caor.ensmp.fr

[2] MENSI
30 rue de la Fontaine du Vaisseau
94120 Fontenay sous Bois
http://www.mensi.com

thomas.chaperon@paris.ensmp.fr
francois.goulette@paris.ensmp.fr

Abstract

This paper presents a new method for extracting cylinders from an unorganized set of 3D points. The originality of this approach is to separate the extraction problem into two distinct steps. The first step consists in extracting a constrained plane in the Gaussian image. This yields a subset of 3D points along with a direction. In the second step, cylinders of known direction are extracted in the corresponding subset of points. Robustness is achieved by the use of a random sampling method in both steps.

Experimental results showing the extraction of pipes in digitized industrial environments are presented.

1 Introduction

The main principle of as-built CAD modeling and reverse engineering is to find a set of geometric primitives properly describing 3D data corresponding to a real scene. The 3D scene is typically provided by a range scanner. Since several views are usually needed, the data often consist of a set of registered and merged range images. In the worst case, the structure of each range image has been lost and the 3D data therefore consist of an unorganized set of 3D points. Obviously, for real data, the scanning process also introduces noise on the points. Moreover, when dealing with large digitized industrial scenes, the density of data points is not uniform [7, 5]. However, the geometric primitives present in the scene are usually surfaces or curves of simple types, such as planes, cylinders, cones, spheres or tori.

Geometric primitive extraction is an important problem of model-based vision. An interesting general framework for geometric primitive extraction was described in [13, 14], generalizing the RANSAC method [2]. In their approach, the extraction of a primitive is performed using random sampling over the set of points. Each sample contains exactly the minimum number of points necessary to define an instance of the primitive — e.g., three points for a plane, four points for a sphere. The extraction is then performed using an general optimization process, i.e. by iteratively considering the primitive defined by a random sample and choosing the primitive maximizing a merit function. This process therefore requires a method for defining the primitive instance from every minimum set of points. It is straightforward for algebraic surfaces of order n given $n-1$ points, since it amounts to the computation of a determinant (see the 'exact fit' in [11]). In 3D, this includes the cases of the plane, the sphere and the quadric — using three, four and nine

points respectively. However, the smallest class of algebraic surfaces including the class of the cylinders is the class of the quadrics. Hence, it is not possible to use the exact fit method of [11] for finding cylinders specifically. Moreover, the attempts of treating the case of the cylinders using elimination theory in [13] proved impractical. The use of surface normals and curvatures was then suggested to yield simpler problems. Similarly, the extraction of cylinders given in [6] first computes the unit normals and then uses the Hough transform on the Gaussian image. As any Hough transform, this method requires a discretization of the parameter space. This is an issue regarding both accuracy and memory requirements. In contrast, we propose to extract cylinders without discretization of the parameter space, using a random sampling algorithm, as for the plane and the sphere in [13].

In [2] and [8], the extraction of cylinders in a range image is performed, but this method is specific to the 2.5D case since ellipses were extracted in cross sections. It is also possible to find a cylinder from a larger set of 3D points by surface fitting [9, 4, 10, 12] . One can fit a cylinder using common nonlinear estimation techniques , e.g. the Levenberg-Marquardt algorithm. Initial estimates are required for this algorithm, and the final results highly depend on the quality of these estimates. More precisely, the initial estimates may be computed from normal or curvature estimates [9] or from an initial quadric fitting [4]. The quality of these estimates depends on the noise on the data points and on the estimation process. Hence noisy input data may yield bad results, even if the cost function is robust. The choice of the cost function — least-squares, M-estimators, etc.—, usually based on an approximation of the Euclidean distance, is also of importance. Indeed, the final parameters may be different depending on the function used.

These problems do not occur when constructing a primitive from a minimal subset of points. That is why we chose to investigate the construction of cylinders directly from minimal sets of points and normals.

In this paper, we present a new method to extract cylinders in an unorganized set of 3D points. This approach has two main particularities: the first one is to separate the extraction into two distinct steps, the second is to use a random sampling algorithm both on the normal estimates and on the set of points. The first step solves the problem of finding the direction of the cylinder. The second step finds its position and size. This leads to a two-stage method that is robust not only with respect to the noise on the data points but also with respect to the unit normal estimates. This method is applied to finding pipes in parts of large digitized industrial environments, consisting of unorganized sets of 3D points with a non-uniform density.

This paper is structured as follows. Section 2 describes the underlying principle of the approach developed. Section 3 deals with the extraction algorithm itself. The main algorithm is presented in section 3.3, while sections 3.3.2 and 3.3.3 give a more detailed description on each of the two steps. Finally, section 4 presents experimental results, on the extraction of pipes in real industrial scenes.

2 The main idea behind the approach

2.1 Parameters for the right circular cylinder

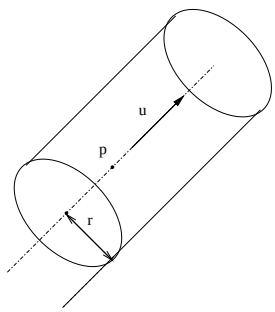

Figure 1: Parameters for a right circular cylinder

The most intuitive way of defining a right circular cylinder is to give a point $\mathbf{p}$ on the axis, a unit vector $\mathbf{u}$ for the direction of the axis, and a radius r (see figure 1). Note that we are not concerned here with the bounds or the height of the cylinder, which we may consider infinite along its axis.

Obviously, there are other possible sets of parameters for defining a cylinder. See for instance [9] for other geometric parameters, and [15] for a set of five free parameters.

2.2 The Gaussian image of a cylinder

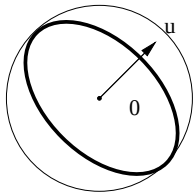

Figure 2: The Gaussian image of a cylinder

The mapping from a point on a surface to the unit normal of the surface at this point is called the Gauss map [3]. The Gaussian image is simply the result of the Gauss map when applied to the whole set of surface points.

Special properties have been shown for ruled surfaces in [3]. In a simpler way, it is straightforward to see that the Gaussian image of a cylinder — even not necessarily circular — is a great circle on the Gaussian sphere (see figure 2). Equivalently, one can say that the Gaussian image of a cylinder is the intersection of the Gaussian sphere with a plane joining the origin. Our method is essentially based on this remark, which is illustrated in figure 2, where the Gaussian image of the cylinder of figure 1 is shown.

In our context, the scene consists of a set of 3D points, not of ideal surfaces. Consequently, the Gaussian image of a scene will denote the set of unit normals estimated on the set of points. We assume that a unit normal is estimated at each point of the scene, and that an index gives access both to the point and to the corresponding normal. Moreover, Gaussian images of scenes containing several surfaces will be estimated (see figure 4). However, the remark made on the Gaussian image of a cylinder still holds for our case: the Gaussian image of a set of points lying on a cylinder is a set of points on the Gaussian sphere that approximately lie on a plane that goes through the origin. The points on the Gaussian sphere belong to the plane up to a noise, which is created during the estimation of the unit normals on the set of 3D points.

Moreover, an important feature of the Gaussian image of a cylinder is that it is symmetrical with respect to the origin. This means that the exact orientation of each unit normal is not needed. This fact is interesting when one estimates unit normals on an unorganized set of points, since the usual estimation procedure — which computes a local inertial plane — does not yield an orientation. Note that this property does no longer hold for Gaussian images of surfaces like the cone.

3 The extraction method

The main extraction algorithm follows the idea of [13], in that the primitives are extracted using random samples of minimal size. As shown in [13], the question of defining a cylinder through five points is not simple, and is not analytic. Intuitively, it is the direction vector that causes most of the trouble.

In contrast, the method we present here uses a much simpler means to estimate the direction parameters. As described in section 2.2, the direction of a cylinder may be estimated directly by a plane on its Gaussian image. Hence, the first step of our method consists in extracting such a plane on the Gaussian image of a whole scene. The second step deals with the estimation of the other parameters of the cylinder, once the direction has been found. Both steps use a random sampling algorithm and minimal subsets of points.

3.1 Defining a plane through the origin by two points

Let us first consider the problem of defining an unconstrained plane from a minimal subset of 3D points. In this case, a minimal subset is a set of three non-collinear points. Indeed, any set of three non-collinear points $\mathbf{x_1}$, $\mathbf{x_2}$, $\mathbf{x_3}$ in $\mathbb{R}^3$ defines a unique plane, whose parameters may be given as follows: a point on the plane, say $\mathbf{x_1}$, and a unit normal vector $\mathbf{n}$:

$$\mathbf{n} = \frac{(\mathbf{x_2} - \mathbf{x_1}) \times (\mathbf{x_3} - \mathbf{x_1})}{\|(\mathbf{x_2} - \mathbf{x_1}) \times (\mathbf{x_3} - \mathbf{x_1})\|}$$

where $\mathbf{a} \times \mathbf{b}$ denotes the cross-product between $\mathbf{a}$ and $\mathbf{b}$ in $\mathbb{R}^3$.

In our case, we are more specifically interested in extracting a plane joining the origin, since the Gaussian image of a cylinder is a great circle. The plane is then constrained, but can nonetheless be defined in the same way as in the unconstrained case: one merely needs to replace the point $\mathbf{x_1}$ by the origin $\mathbf{0}$. A set of two other points that are not collinear with the origin are then enough to define such a plane.

For our problem, introducing this constrained plane is interesting for two reasons. First, one always gets a plane that goes through the origin, thus avoiding cases where the plane might be nearly tangent to the Gaussian sphere. Secondly, reducing the minimal subset from three to two points speeds up the extraction since less random samples are needed (see [13] for a detailed discussion on the number of random samples).

Note that since the Gauss map is not bijective, extracting a plane on the Gaussian image will not be enough to extract a cylinder directly. Indeed, the plane on the Gaussian image may correspond to points on several parallel cylinders, or on other primitives such as planes that are parallel to the cylinder.

3.2 Defining a cylinder of known direction from three points

We assume that the unit direction vector $\mathbf{u}$ of the axis of the cylinder is known here. Let us define an orthonormal coordinate frame $(\mathbf{0}; \mathbf{u}, \mathbf{v}, \mathbf{w})$ from the known direction $\mathbf{u}$. Any point in the plane $\mathbf{0} + \mathbf{u}^{\perp}$ — i.e., the plane perpendicular to $\mathbf{u}$ passing through the origin $\mathbf{0}$ — may be expressed as a linear combination of vectors $\mathbf{v}$ and $\mathbf{w}$.

Given that $\mathbf{u}$ is known, the only parameters that remain to be defined are a point $\mathbf{p}$ on the axis and the radius r. It is clear that these parameters are the ones of the circle obtained by a perpendicular cross section of the cylinder. Finding these parameters thus amounts to finding the parameters of a circle in 2D.

Furthermore, any set of three non-collinear points in 2D define a unique circle, whose parameters — namely the 2D center C and the radius R —, may be derived analytically (see appendix A). Then, once the parameters of the circular section have been found, the parameters of the cylinder are deduced: $r = R$, and $\mathbf{p} = C_v\mathbf{v} + C_w\mathbf{w}$ if C_v and C_w are the coordinates of the 2D center expressed in the 2D coordinate frame $(\mathbf{0}; \mathbf{v}, \mathbf{w})$.

Hence, any set of three 3D points whose projections onto the plane $\mathbf{0} + \mathbf{u}^{\perp}$ are not collinear (and a fortiori not equal) provide a value for the remaining parameters of the cylinder.

3.3 The main algorithm

Our main algorithm uses the previous ideas in the following way. First, the unit normal at each point is estimated (off-line), classically by estimating the inertial plane on a neighborhood of k nearest neighbors. Note that the quality of the normal estimates is sensitive to this size. If the size is small, the result might be noisy because of the noise on the data points. If the size is large, normals estimated on neighbor points may be very similar (smoothing effect). For our application, we chose to avoid the smoothing effect arising with large sizes, by considering 10 nearest neighbors.

3.3.1 Overall description

We proceed as follows:

1. Extraction of a constrained plane in the Gaussian image. A subset of points is obtained.
2. Extraction of cylinders on the extracted subset of points.

The extraction of the plane in the Gaussian image and the extraction of cylinders in the actual scene follow the same general principle. Random samples of minimum size are iteratively chosen among the set of points and a primitive through these points is constructed. Then, for each relevant sample, a merit function corresponding to the current primitive is evaluated. In our implementation, the merit function is the number of inliers, i.e. the number of points that are located at a distance smaller than a threshold from the primitive. This corresponds to a binary merit function in the description given in [13].

A related method, for the extraction of planes with a width, has been investigated in [1].

3.3.2 Details on the constrained plane extraction in the Gaussian image

The main parameters for this routine are: the number of iterations, $K_{\text{N,max}}$, the width for defining the inliers, σ_N , and the minimal number of inliers for a plane to be accepted as a candidate, $N_{\text{N,min}}$ (subscript N stands for 'Normals'). In this manner, the number of iterations denotes the number of planes that are likely to be the optimal plane, rather than the whole set of tested planes. The whole step may be outlined as follows:

Step 1

While ($k < K_{\mathrm{N,max}}$) do
- Choose 2 points at random in the Gaussian image
- If the 2 points and $\mathbf{0}$ are not collinear, compute the plane through these 3 points
- Estimate the number of inliers for this plane
- If this number is optimal, then update the optimal plane

Mark each point whose corresponding unit normal lies on the extracted plane

Return the set of inliers and the direction $\mathbf{u}$ (as the normal vector to the plane)

3.3.3 Details on the constrained cylinder extraction

The main parameters for this routine are: the number of iterations,$K_{P,\mathrm{max}}$, the width for defining the inliers σ_P, the minimal number of inliers $N_{\mathrm{P,min}}$ for a cylinder to be accepted as a candidate, the size of the neighborhood (subscript P stands for 'Points').

With no control on the quality or relevance of the cylinders found, one may get some cylinders that do not exist in the scene. This is all the more true when primitives of different types are present in the scene. In order to tackle this problem, an extra criterion has been added for a cylinder to be considered as 'optimal'. This criterion is based on the heuristics that whenever the cylinder found is wrong — in the sense that there should be no cylinder at this location —, the distribution of the points on the cylinder should be very sparse. The criterion currently used consists in examining the angular distribution of the points projected onto a perpendicular cross-section. The whole range (2π) is discretized into `NbAngularCells` angular cells. A boolean sequence is then created, assigning a value — true or false — to each cell according to whether the number of points that fall in is larger than a given threshold. Finally, the longest run of consecutive 'true's in the sequence is measured. This represents a measure of the angular range of the inliers for the cylinder considered. If this range is smaller than a threshold `AngularThreshold` (expressed as a number of angular cells), the cylinder, being likely to be a false fit, may not be considered as 'optimal' in the algorithm.

At the end, if an optimal cylinder has been found, the set of true inliers is computed. More precisely, a few points of other primitives might actually lie on the unbounded cylinder. In order to eliminate these sparse points and compute the real bounds of the cylinder, the distribution of the inliers along the axis is examined. In a most similar way as for the angular criterion, we compute a boolean sequence of cells along the axis, and search the longest run of consecutive cells for which the number of points is larger than a threshold `DensityThreshold`. This yields a filtering method, which might be compared to region growing procedures.

Step 2

While ($k < K_{\mathrm{N,max}}$) do
- Choose 3 points at random (in a neighborhood)
- If the set is not degenerate, compute the cylinder whose direction is $\mathbf{u}$ (computed in step 1) through these 3 points
- Estimate the number of inliers for this cylinder
- If this number is optimal and the angular criterion is valid, then update the optimal cylinder

Apply the filtering along the axis to get the actual bounds

Mark the corresponding inliers and remove them from the set of points for further processing

For extracting several cylinders, this method is simply iterated, the number of iterations being specified by the user.

4 Results

Results are shown on three example scenes (figures 3, 6 and 8). These scenes are small parts of complete environments, digitized by the SOISIC range scanner, of company MENSI. They contain mostly cylinders (scene 2), but also a small amount of other surfaces (scene 3) and small unidentified parts (scene 1). The number of points in the scenes is approximately 5000 for scene 1, 25000 points in scene 2, 6500 points for scene 3.

The main features of these scenes used are:

1. a non-uniform density of points (depending on the distance of the object to the scanner, occlusion)
2. a non-uniform noise (typically varying with the distance to the scanner, with the angle of scanning, etc.).

However, by selecting constant widths σ_P and σ_N over the whole scene, we implicitly assume that the noise is approximately uniform on the selected zone. This can be justified in the present experiments, since small parts of the initial scene were selected.

The parameters described in section 3 have been set to the following values throughout the experiments: $K_{\mathrm{N,max}}$=500, σ_N=0.1 (which corresponds roughly to a 6 degree angle), $N_{\mathrm{N,min}}$=300 for the first step; $K_{P,}$max=500, σ_P=5.0 mm, $N_{\mathrm{P,min}}$=100 for the second step; also `NbAngularCells`=50, `AngularThreshold`=10 (which corresponds to a $2\pi/5$ angular threshold), `DensityThreshold`=10. The size of the neighborhood (i.e. the number of nearest neighbors) in step 2 has been set to 200.

Figures 5, 7 and 9 show that the cylinders are effectively extracted from the point clouds, even when other primitives or outliers are present, which shows the inherent robustness of the approach. The number of random samples in each step, namely $K_{\mathrm{N,max}}$ and $K_{\mathrm{P,max}}$, has been intentionally set to a high value in order to minimize the effects of individual random samples. Results have shown to be sensitive to the angular threshold, which was predictable, since the angular criterion is directly involved in the choice of the optimal cylinder.

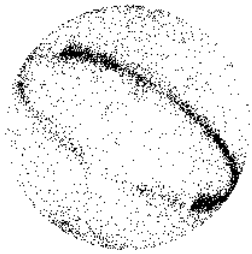

Figure 4: Scene 1. Gaussian image

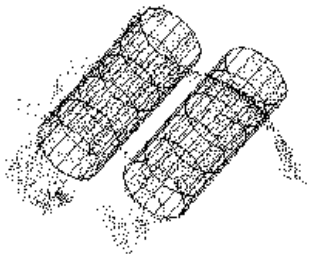

Figure 5: Scene 1. Results

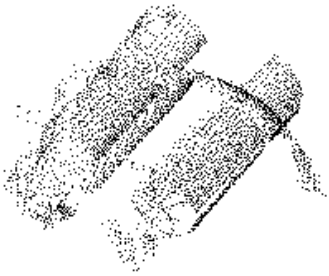

Figure 3: Scene 1. Input points

5 Conclusion and future work

In this paper a new method for extracting cylinders in full 3D data has been presented. Its originality is to separate the problem into two steps: an extraction step on the Gaussian image to compute the direction of a possible cylinder, and an extraction step on the set of points to compute its size and location. Each step uses a random sampling algorithm on minimal subsets and is therefore inherently robust.

Further work will be done on the empirical comparison of the robustness of this approach with other cylinder-specific extraction or fitting methods.

Another subject currently under investigation is how cylinders may be extracted using a random sampling algorithm from the mere set of 3D points.

Finally, future work will also include the study of similar approaches for the extraction of other types of primitives, such as the cone or the torus — corresponding to pipe bents and reducers in industrial scenes. The final purpose is to integrate these approaches into a global extraction algorithm, where primitives of different types would be simultaneously extracted.

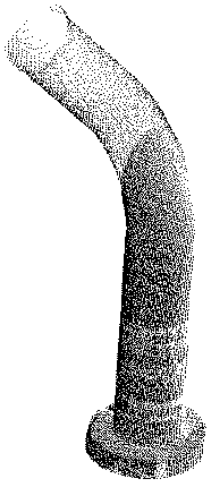

Figure 6: Scene 2. Input points

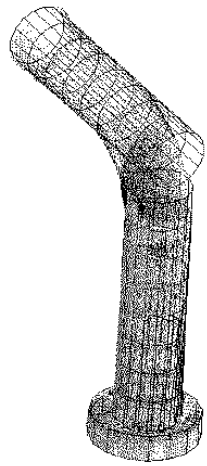

Figure 7: Scene 2. Results

6 Acknowledgments

The scenes were acquired by the SOISIC range scanner of the MENSI company.

A Parameters of the circle from three points in 2D

Any set of three non-collinear points (x_i, y_i) $(1 \leq i \leq 3)$ in $\mathbb{R}^2$ define a unique circle, whose parameters may be derived as follows. There holds:

$$\begin{vmatrix} 1 & x_1 & y_1 & {x_1}^2 + {y_1}^2 \\ 1 & x_2 & y_2 & {x_2}^2 + {y_2}^2 \\ 1 & x_3 & y_3 & {x_3}^2 + {y_3}^2 \\ 1 & x & y & x^2 + y^2 \end{vmatrix} = 0$$

(see the exact fit method [11] referred to in section 1). Expanding this determinant through its bottom row, one gets the equation:

$$-\Delta_1 + x\Delta_x - y\Delta_y + (x^2 + y^2)\Delta_{x^2+y^2} = 0$$

where the Δ's are the corresponding minor determinants. Thus the center of the circle is given by:

$$C = \frac{1}{2\Delta_{x^2+y^2}} \begin{pmatrix} -\Delta_x \\ \Delta_y \end{pmatrix}$$

and the radius:

$$R = \frac{\sqrt{\Delta_x^2 + \Delta_y^2 + 4\Delta_1\Delta_{x^2+y^2}}}{2|\Delta_{x^2+y^2}|}$$

References

[1] M.E. Bock and C. Guerra. A geometric approach to the segmentation of range images. In *3DIM'99, 2nd Int. Conf. on 3-D Digital Imaging and Modeling*, pages 261–269, October 1999.

[2] R.C. Bolles and M.A. Fischler. A ransac-based approach to model fitting and its application to finding cylinders in range data. In *IJCAI, Int. Joint. Conf. on Artificial Intelligence*, pages 637–643, Vancouver, Canada, 1981.

[3] M. P. do Carmo. *Differential geometry of curves and surfaces.* Prentice-Hall, 1976.

[4] R. B. Fisher, A. W. Fitzgibbon, and D. W. Eggert. Extracting surface patches from complete range description. In *3DIM'97, International Conference on Recent Advances in 3-D digital imaging and modeling*, May 1997.

[5] F. Goulette. Automatic CAD modeling of industrial pipes from range images. In *3DIM'97, 2nd Int. Conf. on Recent Advances in 3-D Dig-*

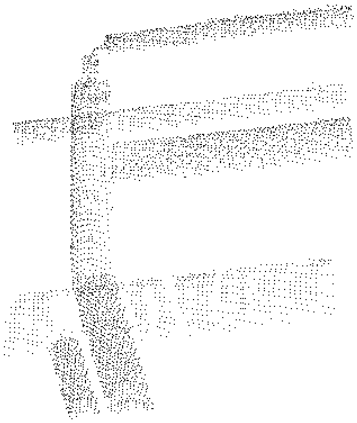

Figure 8: Scene 3. Input points

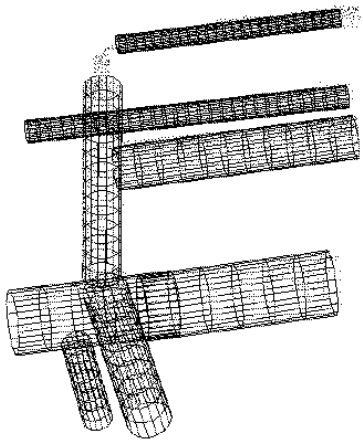

Figure 9: Scene 3. Results

ital Imaging and Modeling, pages 229–233, Ottawa, May 1997.

[6] M. Hebert and J. Ponce. A new method for segmenting 3-d scenes into primitives. In *ICPR, Int. Conf. on Pattern Recognition*, pages 836–838, 1982.

[7] A. E. Johnson and M. Hebert. Recognizing objects by matching oriented points. In *3DIM'97, Ottawa*, Internaional Conference on Recent Advances in 3-D digital imaging and modeling, 1997.

[8] T. Lozano-Perez, W.E.L.G. Grimson, and S.J. White. Finding cylinders in range data. In *Int. Conf. on Robotics and Automation*, pages 202–207, Raleigh, NC(USA), 1987. IEEE.

[9] G. Lukacs, R. Martin, and A.D. Marshall. Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. In *ECCV'98, 5th European Conf. on Computer Vision*, volume 1 of *LNCS 1406*, pages 671 – 686, Freiburg, 1998.

[10] T.S. Newman, P.J. Flynn, and A.K. Jain. Model-based classification of quadric surfaces. *CVGIP: Image Understanding*, 58(2):235–249, September 1993.

[11] V. Pratt. Direct least-squares fitting of algebraic surfaces. *Computer Graphics*, 21(4):145 – 152, July 1987.

[12] C. Robertson, R.B. Fisher, N. Werghi, and A.P. Ashbrook. An improved algorithm to extract surfaces from complete range description. In *World Manufacturing Conference WMC'99 (ISMT'99)*, ICSC Academic Press, Durham, pages 587 – 591, September 1999.

[13] G. Roth and M. D. Levine. Extracting geometric primitives. *CVGIP: Image Understanding*, 58(1):1–22, 1993.

[14] G. Roth and M. D. Levine. geometric primitive extraction using a genetic algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):901 – 905, 1994.

[15] 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*, 13:1115–1138, 1991.