Robust Methods for Geometric Primitive Recovery and Estimation from Range Images

Irina Lavva¹ Eyal Hameiri¹ Ilan Shimshoni²

1:Dept. of Computer Science The Technion - Israel Inst. of Technology Haifa, Israel. 2:Dept. of Management Information Systems University of Haifa Haifa, Israel.

E-mail: lavva@cs.technion.ac.il, ishimshoni@mis.haifa.ac.il

Abstract-We present a method for the recovery of partially occluded 3D geometric primitives from range images which might also include non-primitive objects. The method uses a technique for estimating the principal curvatures and Darboux frame from range images. After estimating the principal curvatures and the Darboux frames from the entire scene, a search for the known patterns of these features in geometric primitives, is performed. If a specific pattern is identified then the presence of the corresponding primitive is confirmed using these local features. The features are also used to recover the primitive's characteristics. The suggested application is very efficient since it combines the segmentation, classification and fitting processes, which are part of any recovery process, in a single process, which advances monotonously through the recovery procedure. We view the problem as a robust statistics problem and we therefore use techniques from that field. A mean shift based algorithm is used for robust estimation of shape parameters, such as recognizing which types of shapes in the scene exist and after that full recovery of planes, spheres and cylinders. A RANSAC based algorithm is used for robust model estimation for the more complex primitives, such as cones and tori. As a result of these algorithms a set of proposed primitives is found. This set contains superfluous models which can not be detected at this stage. To deal with this problem a minimum description length method has been developed which selects a subset of models which best describes the scene. The method has been tested on series of real complex cluttered scenes, yielding accurate and robust recoveries of primitives.

keywords: 3D object recognition, Range Data, Mean Shift, RANSAC, pbM Estimator, Principal Curvatures, Darboux Frame

I. INTRODUCTION AND RELATED WORK

Three major processes are usually involved in tasks of primitive recovery from range images of a complex scene: segmentation, classification and fitting. During the segmentation process an attempt to associate between a set of scene points and a common primitive or a patch of a primitive's surface is made. The classification process attempts to classify the primitive as a specific type of primitive. To complete the recovery, a fitting process attempts to fit each set of points within its type of primitive, as classified by the previous process, a unique primitive in a formulated representation. These three processes interact among themselves back and forth, iteratively. Recovery algorithms in 3D scenes differ by the methods they use for segmentation and fitting, and by the set of objects used for the classification. Our suggested recovery scheme is much more efficient since all three processes advance simultaneously and monotonously within a single process.

Two main approaches appear in the literature in regard to the segmentation of range images, the edge-based and face-based approach. The edge-based approach attempts to bound patches of surfaces, by revealing edge curves. It assumes that edge curves (moderate edges as well) are the borders between different types of surfaces which therefore have to be segmented separately. Such an approach is used in [29], [23], [18]. On the other hand, the face-based approach attempts to group scene points which have special common features that indicate that the points belong to a common surface or object [2], [3], [12], [35]. A survey comparing many of these methods is presented in [21]. The segmentation component of our combined scheme is more related to the face-based approach but while all previous face-based methods start with seed regions that grow in an iterative procedure of hypotheses and validations, our approach segments the

scene points simultaneously. We do not even require that the patches of a single object be connected to each other.

Fitting primitives to segmented areas has nearly always been based on least-squares fitting. Least-squares fitting is usually tailored to the specific type of surface being fit. Examples for that can be found in [22], [25], [34], [27]. The least-squares fitting procedure is performed iteratively and is strongly tied with the segmentation. Each proposed segmentation is validated using a least-squares fitting. When validating an object, methods have to be developed to determine the most probable model [9]. If the validation fails, the segmentation is modified accordingly and another validation is performed. Therefore, whatever type of fitting is made, bi-quadratic, B-spline or other leastsquares fitting, these methods impose an inconvenient procedure. Least-squares fittings are also very sensitive to outliers and to noise in the range data. Our scheme for primitive recovery has an advantage in regard to these two difficulties described above, since it only uses least squares fitting as a last step after the points belonging to the primitive have been determined by the algorithm.

Our approach for primitive recovery is based on estimating the principal curvatures and the Darboux frame (an orthogonal triplet of vectors consisting of the surface's normal and the two principal directions at a surface's point) for each range point. We use the method developed in [19] to estimate these values from range data. Curvatures were also used in [32] for range data segmentation. Since the characteristics of the principal curvatures of geometric primitives are known for all types of primitives and are distinguishable among the different types, primary segmentation and classification can be made by the principal curvature values alone. But furthermore, from the values of the principal curvatures, associated with the segmented scene points, primary assumptions regarding the fitting is already being made as well. At this stage we use the local data stored in the Darboux frame of each scene point to refine the segmentation and classification and at the same time complete the fitting.

We consider this problem as a robust statistics task. In this framework, the points are divided into inliers which fit the model and outliers which do not. In our case the model is a geometric primitive and the outliers are all the points which do not belong to the primitive being recognized.

In our algorithm we use two robust statistics tools. The first one is mean shift [8], [13], [36]. This tool enables us to recover the feature values, which are the modes of their distributions, without modeling this distribution

(it does not need to be a Gaussian distribution) and it also deals very well with outliers in contrast with leastsquares methods. A statistical analysis of a mean shift based method for plane recovery was presented in [28]. This method is used to recover the simpler primitives. This method is fast and simple but it does not scale up for more complex primitives.

For the more complex primitives the local features extracted for a single point are not sufficient to estimate all the parameters of the primitive. Thus, a more elaborate algorithm from the RANSAC [17] family of algorithms has been developed. This algorithm is based on the modified pbM method [5], [26], [6]. This method robustly estimates all the parameters of the primitive in one step. A RANSAC based algorithm which uses only points and normals to the surface for the recovery of simple primitives using Gauss maps was presented in [4].

The only problem with this stage is that quite a few superfluous models are recovered by it. Therefore in the final stage a method for removing these models is described. The method is based on the concept that the goal is to find the minimal number of models which best describe the point set. The final result of the algorithm is a set of primitives and for each primitive a set of points associated with it.

Once an approximate model of the primitive has been recovered the outlier points are eliminated. Only then can a standard least squares optimization procedure be applied to the inliers to generate a more accurate model.

We tested our method on real scenes which contain different types of primitives, in a variety of sizes and orientations, while some of them were partially occluded. The application proved to be very robust and accurate in its recoveries, even when clutter was added to the scene. Almost no false recovery occurred and as long as enough sampled scene points belong to a primitive, it was recovered quite accurately.

The paper continues as follows. In Section II we review several robust statistics methods that will be used in our recovery algorithm. Section III gives a short description of the geometric and differential properties of the geometric primitives recovered by our system. A general overview of our recovery system is given in Section IV and the details of the mean shift based algorithm and the RANSAC based algorithm are given in Sections V and VI respectively. The final classification stage which removes the superfluous models is described in Section VII. Representative examples of our tests are presented in Section VIII. We end with Section IX which summarizes our conclusions from this work.

II. ROBUST METHODS

We regard the problem of recovering geometric primitives from a scene consisting of several partially occluded primitives and non-primitive objects as a task which requires tools from the field of robust statistics. In this framework a set of points is given to the program. Each point is either an inlier or an outlier. The inliers can be described as points belonging to a model corrupted by noise and the outliers can not be described this way. The task is to separate the inliers from the outliers and to estimate the model.

In our case the models are simply the geometric primitives described by a small number of parameters, the inliers are range points measured on the surface of the geometric primitive and the outliers are the rest of the points. In a complex scene the role of inlier and outlier changes from primitive to primitive. The final goal is to recover the model parameters for each primitive and find the set of points belonging to it.

For this reason we turn to methods from this field. The three methods that will now be reviewed were developed for computer vision applications but they are general methods and can be used for solving problems in many different fields.

A. Mean Shift

Mean shift[8], [13], [36] is a technique for finding modes of empirical distributions defined by a set of npoints $\mathbf{x}_i, 1 \leq i \leq n$ in \mathbb{R}^d . This technique has several applications. One of the major applications is to estimate robustly parameters from the points. The assumption is that the inliers are noisy measurements (or a result of a computation from noisy measurements) of a parameter $\theta \in \mathbb{R}^d$ and the outliers are not. Thus, looking at the set of points as originating from an empirical distribution, this distribution will have a mode close to θ .

Mean shift is a hill climbing technique to the nearest stationary point of the density, i.e., a point in which the density gradient vanishes. The initial position of the kernel, the starting point of the procedure is chosen as one of the data points \mathbf{x}_i . The points of convergence of the iterative procedure which represent the clusters are the modes (local maxima) of the density.

The classic method for performing this task is the Hough Transform. There, the parameter space is tesselated into cells, the frequency of parameter values in each cell is computed. The cell with the largest number of votes is chosen. There are several major drawbacks of this method compared to mean shift. First, the accuracy of the estimated parameter is limited by the size of the

3

cell and all that can be said is that the parameter lies somewhere within the cell. Second, the tessellation may cause aliasing. For example when the mode lies close to the boundary of a cell the votes for the mode will be distributed between several cells and the mode might not be detected. We conclude therefore that mean shift is the preferred method for robust parameter estimation.

B. RANSAC

RANdom SAmple Consensus (RANSAC) [17], [20] is a robust method for model estimation. This is a general method which we will now illustrate with a simple example. Consider a set of points in the plane which are divided into points lying close to a line (the inliers) and many other points at random positions. The task of the RANSAC algorithm is to estimate the parameters of the line (the model) and separate the inliers from the outliers.

At each iteration of the procedure a pair of points is randomly selected and the equation of the line going through them is computed. Then the quality of the line is estimated by the number of points lying within distance *dist* (the scale parameter) from the line. This procedure is repeated a number of times and the line for which the largest number of points "voted" for is selected. In the last step of the algorithm a more accurate estimate of the line is computed using all the points which are suspected to be on the line.

The number of iterations, N, that ensures with probability, p, that at least one of the random samples of s points (two in this example) is free of outliers is

$$N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)} \tag{1}$$

where ϵ is the probability that a selected data point is an outlier. Thus, the fraction of outliers has a major effect on the runtime of the algorithm.

In a variant of the RANSAC method known as PROgressive SAmple Consensus (PROSAC) [10] the algorithm exploits the linear ordering defined on the set of correspondences by a similarity function used in establishing tentative correspondences and uses it to perform a non-uniform sampling in the RANSAC process speeding it up. The method of maximum likelihood estimation by sampling consensus (MLESAC) [31] is an improvement of RANSAC. Instead of counting the number of matches agreeing with the model, evaluates the likelihood of the hypothesis representing the error distribution as a mixture model of a Gaussian distribution for inliers and a uniform distribution for outliers.

4

In our algorithm we make use the LO-RANSAC method [11]. During the iterations of the algorithm, whenever a new best result is found an inner iteration loop is performed. In these iterations more than two points are selected in order to yield a more accurate estimate for the line. The points selected for generating the line are chosen only from the suspected inliers. This modification speeds up the algorithm and yields more accurate results.

This general method has been applied to many other types of models. This is done by setting the minimal number of points needed to estimate the model and by defining a distance function from a point to a suggested model.

C. Projection Pursuit Based M-Estimator(pbM)

The pbM [5] and its variant the modified pbM [26], [6] are algorithms from the RANSAC family. A similar method was suggested independently in [33]. We will give here a short review of the modified pbM method.

Consider the problem of estimating a hyperplane $[\theta, \alpha]$ in \mathbb{R}^d such that a point x lying on the hyperplane satisfies

$$\mathbf{x}^T \boldsymbol{\theta} - \boldsymbol{\alpha} = 0,$$

where θ is the normal and α is the distance of the hyperplane from the origin. The RANSAC algorithm can be easily applied to solve this problem. However, in order for this algorithm to work well the scale parameter *dist* (points which lie at distance less than *dist* from the hyperplane are considered inliers) has to be supplied by the user. This causes many practical problems because the optimal value of *dist* changes considerably from run to run of the algorithm.

The modified pbM algorithm therefore uses a different function to compute a score for the proposed model. At each iteration of the RANSAC process a model is computed from s = d points. Disregarding α the empirical distribution f_{θ} of $\{\mathbf{x}_i^T\theta\}, 1 \leq i \leq n$ is computed using kernel smoothing. For a correct θ the mode of this distribution should be close to the correct α which is more accurate than the value of α computed only from the *s* points. Moreover, the closer θ is to the correct normal the higher $f(\alpha)$ will be.

Thus, for a given θ

$$\alpha_{\theta} = \arg\max_{\alpha} f_{\theta}(\alpha)$$

and the score of θ is

$$\max_{\alpha} f_{\theta}(\alpha) = f_{\theta}(\alpha_{\theta}).$$

Practically, α_{θ} is found by a uniform scan of the data and a more dense scan around the maximal value found in the uniform scan. A bandwidth parameter *h* has to be given in order to perform the kernel smoothing. In the pbM algorithm the value of *h* is computed from the data set $\{\mathbf{x}_i^T \theta\}, 1 \leq i \leq n$ using a robust technique. This value changes for each value of θ and is therefore denoted h_{θ} .

In the final step of the algorithm the inliers are separated from the outliers by studying f_{θ} and finding two local minima one on each side of α_{θ} . All points with values between these two minima are considered inliers. This adaptive method is opposed to the standard RANSAC technique which decides that all points whose distance from the model is less than the user supplied *dist* are considered inliers.

III. CURVATURES OF GEOMETRIC PRIMITIVES

In a previous work we developed two algorithms to estimate the local differential properties of a point from range measurements of the points and its neighbors [19]. These are modifications of two previously published algorithms, originally suggested by Taubin [30] and Chen and Schmitt [7].

Using this method (or other methods) at each point the following parameters are estimated.

- P: the 3D point.
- N: the normal to the surface at the point.
- κ_1 and κ_2 : the maximal and minimal principal curvatures.
- T_1 and T_2 : their corresponding principal directions.

The orthogonal vector triplet N, T_1 , and T_2 are known as the Darboux frame.

Usually the estimate of the principal curvatures and directions will be less accurate than the point and the normal because they involve computing second order derivatives whose accuracy is reduced due to noise. In this paper we will be using these local features to recognize and estimate the parameters of several geometric primitives.

We will now describe a method to display the principal curvatures from an entire object or scene, and review the characteristics of principal curvatures and directions for several geometric primitives.

Let us assume that we gather the principal curvatures from a finite number of surface points. We can describe all these values within a two dimensional and discrete histogram in which one axis represents the minimal principal curvature (in this work we use the vertical axis) and the other axis represents the maximal principal curvature. The histogram value at a specific location reflects the amount of minimal vs. maximal principal curvature values found in the data. The histogram can also be displayed as a gray-level image. Thus, for every object or surface an image of its principal curvatures can be generated. This image is invariant to rigid transformations, as long as the same portion of the surface is analyzed, but it can be altered due to partial occlusion of the object.

A. Principal Curvature Histograms of Primitives

Exploring the curvature histogram for scene objects can help to perform an initial segmentation for objects types. We will now present a short review of the principal curvature histograms of several geometric primitives.

Planes: In the trivial case of planes, all directional curvatures equal zero, thus for each point on a plane $\kappa_1 = \kappa_2 = 0$. The principal directions in planes are therefore undefined. All planes have a common gray level image of their principal curvatures - one bright point at the center of the image (Figure 1(b)).

Spheres: For a sphere, the absolute value of each principle curvature is the inverse of its radius. The sign of the curvatures is negative due to our definition of the orientation of the normals. The principal directions are undefined. The curvatures histogram is one bright point on the negative side of the main diagonal at a distance of the inverse of the radius of the sphere from both negative axes (Figure 1(b)).

Cylinders: The cylinder's bases are considered as planes. As for the rest of the cylinder's points, they all share the same values for the minimal and maximal principal curvatures. The maximal principal curvature,



Fig. 1. (a) Cylinder - principal curvatures and Darboux frame; (b) Principal curvature typical histogram: 1 - plane, 2 - sphere, 3 - cylinder.

 $\kappa_1 = 0$, and its related tangent direction, T_1 is aligned

with the main axis of the cylinder. The minimal principal curvature, $\kappa_2 = -1/R$, where R is the cylinder's radius. The direction related to the minimal principal curvature, T_2 , is the tangent to the circular normal section at P. The principal curvatures histogram is one bright point on the negative vertical axis at a distance of 1/R from the origin (Figure 1).

Cones: All points lying on a cone (except for its apex and base) share the same maximal principal curvature - $\kappa_1 = 0$. T_1 always points from P to the apex (Figure 2). T_2 , is the tangent to the elliptical normal section contained in the plane which is tilted by the half opening angle of the cone, α , relative to the plane containing the circular cross section that passes through P. The minimal principal curvature at point P is given by $(-r^{-1} \cdot \cos \alpha)$, where r is the radius at point P.



Fig. 2. Cone: (a) Darboux frame; (b) Typical histogram of principal curvatures.

The histogram of the principal curvatures is theoretically a half infinite straight line along the negative vertical axis which starts at $(0, -R^{-1} \cdot \cos \alpha)$ where Ris the radius of the base. Practically, if we look at a finite number of data points then the histogram is a finite straight line from $(0, -R^{-1} \cdot \cos \alpha)$ to $(0, -\hat{R}^{-1} \cdot \cos \alpha)$ (Figure 2) where \hat{R} is the cone's radius at the closest analyzed point to the cone's apex.

Tori: For all points on a torus, the minimal principal curvature equals $-r^{-1}$ where r is the minor radius of the torus. The minimal principal direction at point P is the torus tangent at P that is also contained in the plane which is perpendicular to the center circle of the torus (Figure 3(a)). The maximal principal curvature at P varies from $(R-r)^{-1}$ to $-(R+r)^{-1}$ where R is the major radius of the torus. The corresponding principal direction is the radial tangent. Analytically, the principal curvatures are:

$$\kappa_1 = -\frac{\cos(\alpha)}{R + r\cos(\alpha)} \qquad \qquad \kappa_2 = -\frac{1}{r}$$



Fig. 3. Torus: (a) Darboux frame; (b) Typical histogram of principal curvatures.

The principal curvature histogram is a horizontal straight line from $(-(R+r)^{-1}, -r^{-1})$ to $((R-r)^{-1}, -r^{-1})$ crossing the negative vertical axis at $(0, -r^{-1})$ (Figure 3(b)).

IV. THE RECOVERY STRATEGY

As described in Section III, every type of primitive has its unique "signature" in the principal curvatures histogram of the entire scene. For most types of primitives there is even a different signature for primitives of different dimensions within the same type. All parts of the geometric primitive have the same signature as the entire primitive. Therefore, it is possible to discover the presence and characteristics of a specific geometric primitive by analyzing the curvatures histogram of the entire scene, even if the object is partially occluded. This stage of the process, in which we examine the principal curvatures histogram, will be termed as the first stage of the recovery process. At the end of the first stage, hypotheses for the presence of specific types of primitives, or even for specific primitives, already emerge.

The second stage of the recovery process involves the Darboux frames which were extracted from the scene. For the second stage we have developed two different types of algorithms for primitive recovery. One for the simpler objects (planes, spheres and cylinders) and the other for the more complex primitives (cones and tori). A method of the second type has also been developed for the simpler types.

The main difference between the two types of primitives which warrants two different types of algorithms is that when given a point and its extracted parameters belonging to a simple primitive, all the parameters of the primitive can be extracted. The extracted values are very noisy but using mean shift the parameters of the primitive can be extracted and the inliers separated from the outliers.

This is not the case for cones and tori. Here more than one point is needed to estimate the shape of the object and therefore the mean shift method can not be applied in a straight forward manner. Instead, we use a modification to the RANSAC/pbM method. There a small set of randomly selected points is used to estimate most of the shape parameters of the object and the value of the last parameter of the shape is recovered using the distance value which yields the maximal density.

In the next two sections we will describe the specific methods for primitive recovery.

V. MEAN SHIFT BASED PRIMITIVE RECOVERY

We begin with the estimations of normals, principal curvatures and principal directions for each point in the scene as explained in [19]. The recovery procedure is performed by a quick search for peaks in n dimensional (nD) histograms of the features and/or of their functions. The first stage involves only the 2D histogram of the principal curvatures and the second stage uses histograms of higher dimensions.

The peak extraction procedure is straight forward. At first the multidimensional histogram is searched for cells with a large number of votes and then mean shift is started from points within those cells. The histogram is only used to speed up the process and is not an essential part of mean shift.

We will now elaborate on the specific methods developed for the simple primitives: the planes, the spheres and the cylinders.

A. Planes

A peak in the 2D curvatures histograms at a point where $\kappa_1 = \kappa_2 = 0$ indicates one or more planar elements. Let $N \cdot P + D = 0$ be the equation of the plane where N is the plane's normal, P is a point on the plane and D a scalar. All points on a specific plane share the same surface normal. Another common feature is $D = -N \cdot P$. Now, if a planar element is indeed present in the analyzed scene there must be a set of scene points which share the same values for N and for $-N \cdot P$. We therefore expect to find a local peak in the histogram of N and $-N \cdot P$. When we locate a local peak we actually get a confirmation for the presence of a planar element and also reveal its parameters according to the specific location of the local peak. Thus by locating one or more local peaks we can separate between different planes and find for each one of them its parameters. Figure 5 demonstrates segmentations initiated from a local peak located at the center of a curvatures histogram of the real scene presented in Figure 4. Note that many points which lie on planes, as well as some points that do not are included within the segmentation at the end of the first stage. However, at the end of the second stage only planar points are segmented as planes. Although it does not affect the recovery, in some planes the segmentation covers less of the original plane's area due to higher level of imaging noise resulting in less accurate estimations of curvatures. Also note that one local peak of the first stage is refined into four distinct local peaks at the second stage, yielding four different planes.



Fig. 4. Real scene of partially occluded primitives and freeform objects: (a) Range image; (b) Local peaks at the end of the first stage illustrated upon the center of the principal curvatures histogram.

B. Spheres

Spherical elements appear in the curvatures histogram as peaks along the main diagonal where $\kappa_1 = \kappa_2 \neq 0$. Each peak represents one or more spherical elements of a specific radius and therefore it is more efficient to analyze each peak separately. A peak at (κ, κ) suggests that one or more spherical element of radius $|\kappa^{-1}|$ exists in the scene. We can recover their characteristics and distinguish between them by recovering their centers. For every point *P* lying on a sphere, the sphere's center



Fig. 6. The common features in the sphere case

is located at a distance of the radius, i.e. $|\kappa^{-1}|$, in the direction opposite to the normal N at P (Figure 6). Thus, we search for peaks in the 3D histogram of $P - |\kappa^{-1}|N$. In addition to getting a conformation to the presence of spherical elements of radius $|\kappa^{-1}|$, we also recover their centers. Note that if the first stage of the recovery, locates a false local peak which was generated from arbitrary scene points that do not lie on a common sphere (or primitive in the general case), then the second stage will not confirm the hypothesis because these points will fail to generate a local peak in the histogram of the second stage. A demonstration of such a process with a real scene is presented in Figure 7 where we can see several steps of the first stage. At each iteration of the mean shift process, more and more points which lie on the sphere are found. As can be noticed, some points that do not lie on the sphere are also included in that segmentation but these points do not pass the criteria of the second stage, and thus only the sphere's points are included in the final segmentation at the end of the second stage.

C. Cylinders

A local peak in the curvatures histogram located at $\kappa_1 = 0, \kappa_2 \neq 0$ is an indication for the existence of one or more cylinders of radius $|\kappa_2^{-1}|$. We deal with each "cylinder peak" separately, but still each peak might indicate several cylinders of the same radius but of different orientations and/or locations within the scene.

We use the direction of the cylinder's main axis and a point on one of the main planes through which the main axis passes as common features. Let P be a 3D scene point contributing to the peak in the histogram. The main axis direction is given by the maximal direction at Pdenoted by T_1 (Figure 8).



Fig. 8. The common features in the cylinder case



Fig. 5. Segmentation results of planes during the first and second stages of the recovery process: (a) and (b) are the segmentations at two different iterations of the first stage which searches for the local peak located at the origin of the curvatures histogram of the real scene presented in Figure 4. (c) (d) (e) and (f) illustrate the segmentations at the end of the second stage after the first stage's peak was refined to four different local peaks of the second stage. Note that after the second stage all points not on the recovered planes have disappeared.



Fig. 7. Segmentation snapshots of the first and second stages of the algorithm related to a single local peak of the curvatures histogram of the scene in Figure 4. The peak has the (κ, κ) form implying a sphere. (a) (b) (c) (d) and (e) illustrate the segmentation during the first stage of the algorithm; (f) The segmentation results at the end of the second stage. Note that in the second stage all points which do not belong to the sphere have disappeared from the segmentation.

A point on the cylinder's main axis can be found at a distance of the radius $|\kappa_2^{-1}|$ from *P* in the direction opposite to the normal, *N*. Now we can find the intersection points of the axis with the main planes (at least one exists). As a result a local peak in the 4D histogram of the two features will be found confirming the existence of the hypothesized cylinder as well as

distinguishing between them and revealing the equations of their main axis. A complete process of recovering a single cylinder in a real complex scene is demonstrated in Figure 9. Along the first stage's illustrations we can see the accumulated segmentation of the smaller cylinder (there are two different cylinders in the scene). The segmentation includes some points that do not lie on



Fig. 9. Segmentation along the first and second stages which is related to a single local peak of the curvatures histogram of the scene in Figure 4. The peak has the $(0, \kappa)$ form implying a cylinder.(a) (b) (c) (d) and (e) illustrate the segmentation along the first stage; (f) Segmentation at the end of the second stage. Note that in the second stage all points not on the smaller cylinder have disappeared. Also note that less scene points which lie on the smaller cylinder were segmented in the second stage in comparison to the end of the first stage. Some points failed to pass the criteria of the second stage although enough of them were left to enable the cylinder's recovery.

the cylinder as well. At the end of the second stage only points of the smaller cylinder are left although less of them are included when compared to the last iteration of the first stage. This phenomenon is caused by inaccuracies in the curvatures and Darboux frame estimations due to noise and its extent also depends on the adjustments of the mean shift parameters. This phenomenon however, does not affect the recovery since enough points are still included in the final segmentation.

VI. RANSAC STRATEGY FOR SHAPE RECOGNITION

Now we present a different strategy for extracting shape parameters from raw data designed for more complex primitives. It uses an adaptation we performed to the modified pbM method to deal with geometric objects. Algorithms of this type can be developed for all parametric primitives. To demonstrate this general technique we will present algorithms for cylinders (for which an algorithm of the previous type has already been presented), cones and tori.

At the first stage after mean shift on the curvatures has been performed, the scene is divided into groups of points by the curvature values of the modes. The goal of this operation is to place in a single group most of the points belonging to a primitive and to remove most of the outliers. For example, we look for cones and cylinders only in modes where κ_1 has a value close to zero. For the cylinders the value of κ_2 is a constant and estimates 1/R, where R is the radius of the cylinder. For a cone, its points are divided arbitrarily to several modes. Therefore, all the modes are merged and models are searched using all the points belonging to these modes. For tori detection we explore modes that are not plane and sphere modes. In this case κ_2 is a constant and estimates 1/r, where r is the minor radius of the torus.

In the second step a variant of the modified pbM technique is applied. Two (for cylinders and cones) or three (for tori) points are chosen at random. These points are used to estimate a shape whose characteristic is that all points on the primitive are equidistant from it. For example in the case of the cylinder, the central axis is estimated. The change with respect to the modified pbM is that the density is not measured as the distance from a hyperplane but as the geometric distance from the point to the shape estimated from the 2-3 points. In order to eliminate most incorrect models its correctness is tested with respect to the local differential features of the points used to create it. Only for models that pass this initial test, we compute their score using all the points.

As in modified pbM, the score of the model is the density value of the mode. The distance for which the mode was found is used in conjunction with the central axis to estimate the parameters of the primitive. Consider for example the case of the cylinder. There the major axis combined with the distance from it to the range points which is the radius of the cylinder yields the complete set of parameters. The reason for estimating the radius using the pbM and not from the pair of points is because this estimate is more accurate as it is generated from all the points suspected to belong to the cylinder and not only from the two points used to generate the central axis. This principle is used for all models to yield a more accurate estimate of one of the parameters.

For a given estimate of the axis, the distribution of the distance of the points to the axis is estimated. In addition, the normal of the surface at the point and the vector from the point on the shape to the closest point to it on the axis should be parallel. Therefore, the distribution of the angle between these two vectors is also estimated. Thus, a point is considered an inlier when its distance to the axis lies between the two local minima around the mode found by the pbM procedure and when the angle is less than the first local minimum which is greater than the mode of the second distribution.

This process is performed N times, where N is computed using Eq. 1 in order to guarantee with high probability that an axis will be found. Whenever an axis is generated whose score is higher than found before it is chosen as the new best axis. Then a LO-RANSAC step is performed. In it another set of RANSAC iterations is performed. This time more than the minimal number of points are used but they are chosen only from the points characterized as inliers by the axis found. This way the probability of choosing an inlier is increased and the chance of choosing a set of inliers is non-negligible. The advantage of this second step is that the axis recovered will be more accurate than the axis recovered from only two points.

When the whole RANSAC process has been completed, all the points considered inliers by the pbM procedure are given to the Nelder-Mead simplex optimization procedure [24] to yield the least squares estimate of the shape. The starting position of the optimization process is the shape recovered by the RANSAC process. At this stage we are sure that no outliers are left in our point set and the most accurate estimate of the shape can be recovered. This final optimization step is performed for all types of primitives.

The models recovered by the process may contain spurious models, i.e, models recovered several times or incorrect models whose points actually belong to other models. The reasons why this happens and the solution for this problem are described in Section VII. This solution is general for all shape types which are recognized in the scene.

This general technique has been applied to all three

types of primitives. We will now elaborate on the specific details for each primitive type concentrating on how their central axes are computed and how the distance value at which the mode has been found is used.

A. Cylinder

In Section V-C we described a mean shift based method for cylinder recovery. Here we will show the RANSAC based method.

The geometric definition of a cylinder is a set of points of radius R from a central axis. Using this definition the axis is estimated in the following manner. Select at random two points P^1 and P^2 . For each point compute

$$Q^{i} = P^{i} - N^{i} / |\kappa_{2}^{i}| \approx P^{i} - N^{i} R.$$
 (2)

The points Q^i lie close to the central axis of the cylinder. Thus, the axis $\widehat{cN} = Q^i - Q^j$ can be estimated. The initial test that is applied to the two points is that this direction is orthogonal to T_2 and to N^i .

For each point in the point set the distance to the model is computed. The mode of the distribution of distances will usually be found close to distance R. The angle between the normal N^i estimated at point P^i and the vector $\overrightarrow{P^i Q^i}$ connecting P^i to the closest point $\widetilde{Q^i}$ to it on the axis should be close to zero. Thus, as mentioned above both the distance R from the axis and the angle between the normal N^i and vector $\overrightarrow{P^i Q^i}$ are used to detect inliers.

In the inner RANSAC loop more than two points are randomly chosen from the set of suspected inliers. In this case linear regression [15] is used to estimate the line closest to the set of points Q^i .

B. Cone

The algorithm for the cone is a modification of the algorithm for the cylinder (as illustrated in Figure 10(a)). Here we have to estimate the central axis, the apex and α , the half opening angle of the cone. As before two points P^1 and P^2 are randomly selected. Points on the central axis

$$Q^i = P^i - N^i / |\kappa_2^i| \tag{3}$$

are computed. Here the distance to the axis changes from point to point as κ_2 is not constant. We define the central axis

$$\widehat{cN} = (Q^j - Q^i)\operatorname{sign}(|\kappa_2^i| - |\kappa_2^j|)$$
(4)

such that the direction from the apex to the points on the central axis is positive. The angle α is defined as

$$\alpha = \angle (N^i, \hat{cN}) - \pi/2, \tag{5}$$

$$apex_{init} = Q^i - \widehat{cN} \cdot / (|\kappa_2^i| \cdot \sin(\alpha)).$$
 (6)

The initial test that is applied to the two points is that the direction of the central axis is orthogonal to T_2 . The position of the apex is usually quite far from the correct position due to noise. When all the other parameters of the cone are correct, the cone we get is at a certain constant distance from the correct cone. Computing the distribution of the distances of the points from the estimated cone yields a mode at a certain distance *dist*. This distance is used to modify the position of the apex of the cone

$$apex = apex_{init} - \widehat{cN} \cdot dist / \sin(\alpha).$$
 (7)

Now the mode of the distance is at zero. As in the case of the cylinder, the normal N^i to the point should be parallel to the vector connecting the point P^i to the estimated cone.

In the inner RANSAC iterations, when more than two points are selected at random, the central axis is computed in the same way as the central axis of the cylinder. The other parameters are simply computed by averaging their values over all the points.



Fig. 10. Illustration of cone and torus algorithms

C. Torus

The algorithm for the torus also works on a central axis (as illustrated in Figure 10(b)). Instead of looking for the distance from the line, we look for the distance from the axle circle. Each point on the torus is at distance r from its closest point on the circle. Each point on the central axle is estimated by

$$Q^i = P^i - N^i / |\kappa_2^i| \approx P^i - N^i r.$$
(8)

In order to estimate the equation of the central axle three points are used. The radius of this circle is the major radius of torus. As in previous algorithms we will look for the mode of the distance from the computed circle. The position of this mode is our estimate of the minor radius of the torus. As in other shapes, the normal N^i should be parallel to the vector connecting the point P^i to the central axle. This characteristic and that T_1 should be orthogonal to the circle plane normal are checked initially on the three points to eliminate most incorrect models.

A torus is a very general object and its algorithm can be used to recover cylinders and spheres also. A cylinder is a torus with $R_{torus} = \infty$ and $r_{torus} = R_{cylinder}$ and a sphere is a torus with $R_{torus} = 0$ and $r_{torus} = R_{sphere}$.

To illustrate the RANSAC based algorithm consider the following simple scene of a torus and a plane. In Figure 11 we show four examples of models suggested by the RANSAC process starting from a model with a low score and ending with the best model found for this example. Looking at the distance density plots (Figure 11(c)), we can see that the quality of the model improves with the height of the mode and the narrowing of its spread. This is well correlated with the number of points (in blue) which agree with the model. Figure 11(b) shows the values for which the distribution was evaluated by the algorithm. Combining the central axle with the distance which achieved the maximal density yields a complete model for the torus. The distribution of the distances from this model is shown in Figure 11(c). Finally, Figure 11(d) shows the distribution of the angle between the normal and the vector from the point to the central axle. Only points selected (in blue) by both Figure 11(c) and (d) are considered inliers by the algorithm (Figure 11(a)).

VII. THE FINAL CLASSIFICATION STAGE

After the recognition process is over we are faced with two false-classification cases: the same model can be detected several times and groups of points may be classified as belonging to several models.

The reason for the first problem is that after the scene is separated into mode groups recovered by the mean shift algorithm from their curvature values we start looking for models in each group separately. It is therefore possible that points belonging to one object will be present in different modes. Therefore, several very similar models will be recovered from different mean shift modes.

The second type of problem is demonstrated in Figure 12. The two red tori were falsely detected. This is a result of the way the models are detected. While detecting models by random sampling in the RANSAC



Fig. 11. Demonstration of the RANSAC method



Fig. 12. Scene classification result with several incorrect models

recognition stage, each found model is verified with scene points. Therefore the same group of points can

be recognized as different models. One of the reasons why this happens is because regions of those models have Darboux frame vectors with similar directions.

Figure 13 shows the detection results of a correct and incorrect tori. As can be seen from the density functions, it is impossible to distinguish at this stage between correct and incorrect models. Both models yield similar density functions.

In order to eliminate false models we developed a minimum description length type algorithm to choose the best subset of models from the set of all proposed models U. First we evaluate the grade function of fitting each scene point to each model based on combining distance, normal and tangency vector direction differences between the scene point and their closest model points. Each of those measures has a different distribution,

therefore these distances have to be normalized when they are combined. Usually distances are divided by the standard deviation of their distributions. However, a single outlier point can change this value significantly. Therefore, we chose to use the median value as a robust normalization factor, because all the data we work with contains many outlier points, and the median's breaking point is 50% outliers.

First, for each point P^i we compute the distance between it to the closest point S_j^i on model j. Define the distances between scene points and their appropriate models points by:

$$dist_{j}^{i} = ||S_{j}^{i} - P^{i}||_{2}.$$
(9)



(b) incorrectly detected torus

Fig. 13. Robust detection of torus points

Define the differences between scene points normal directions N^i and their appropriate models points normal directions N^i_i by $angleN^i_i$:

$$angleN_i^i = \angle (N_i^i, N^i). \tag{10}$$

Define the differences in tangent directions between scene points T_1^i or T_2^i and their appropriate models points directions T_{1j}^i or T_{2j}^i as:

$$angleT_{1j}^i = \angle(T_{1j}^i, T_1^i) \quad or \quad angleT_{2j}^i = \angle(T_{2j}^i, T_2^i).$$
(11)

In our implementation we compare the T_1 directions for cone models, and T_2 directions for tori models.

Second, we compute the distribution histograms of distance, normal and tangency vector direction differences between scene points and their appropriate models points that were detected as inliers in the previous stage of the algorithm and evaluate their median values.

$$\Delta dist = \operatorname{med}_{\substack{\forall j \in U : i \text{ is inlier of } j \\ \forall j \in U : i \text{ is inlier of } j \\ \forall j \in U : i \text{ is inlier of } j \\ \Delta angle T = \operatorname{med}_{\substack{\forall j \in U : i \text{ is inlier of } j \\ \forall j \in U : i \text{ is inlier of } j \\ \forall j \in U : i \text{ is inlier of } j \\ \end{bmatrix}} (12)$$

Third, we define s_j^i as the score of point i belonging to a recognized model j

$$s_j^i = \frac{1}{3} \left(\frac{|dist_j^i|}{\Delta dist} + \frac{angleN_j^i}{\Delta angleN} + \frac{angleT_j^i}{\Delta angleT} \right).$$
(13)

The best score of point i is:

$$\sigma_U^i = \min_{j \in U}(s_j^i) \tag{14}$$

and the best fitted model for point i is:

$$U^{i} = \underset{j \in U}{\operatorname{arg\,min}}(s_{j}^{i}). \tag{15}$$

To evaluate the score of the scene we use the sum of the score function over the entire scene:

$$\sigma_U = \sum_{i=1}^n \sigma_U^i. \tag{16}$$

This score will now be used to find an optimal subset of models from U. This is done by removing one model at a time. We evaluate the classification grade for the scene without one of the models. If the grade does not change significantly - it indicates that all points of the removed model can be easily associated with other models and therefore we can assume that this model was falsely detected. When the grade changes significantly this hints that for the points belonging to this model there is no other candidate model to take its place. This means that the model was correctly detected. Define J as a group of

$$\Delta \sigma_j = \frac{\sigma_{U \setminus \{j\}} - \sigma_U}{|J|}.$$
(17)

The weakest model will receive the grade of:

$$\Delta \sigma_{candidate} = \min_{j \in U} \Delta \sigma_j, \tag{18}$$

and its respective model is removed.

We stop the algorithm when the minimal grade al teration is larger than the maximal value of the grade computed in the initial classification guess

$$\Delta \sigma_{MAX} = \max_{\substack{\forall j \in U : i \text{ is inlier of } j}} (s_j^i).$$
(19)

It is evident that if the model score changed by more than this score their points were incorrectly classified by their new chosen models.

The complexity of this final classification stage is:

$$O(|U|^2 n), \tag{20}$$

where n is the number of points in the scene, and the |U| is the number of detected models.



(a) classification grades for all models in the scene

to be removed models

Fig. 14. Demonstration of the final classification algorithm.

14

We will now illustrate the algorithm using the same example used before. The results are shown in Figure 14. The algorithm ran four iterations. Each row shows one iteration and the columns demonstrate what happens in each iteration. Between iterations the points are moved to the next closest model according to its grade value. Each model is shown in the same color in all the figures, and in the score graphs.



Fig. 15. Final classification algorithm results; (a) The correct models are detected; (b) For each model the points belonging to it are detected.

The first column shows the classification grades for all the models. The shape of the markers represents the type of the model: a cone is represented by a triangle, a torus by a circle and a plane by a square. The second column shows the model to be removed, and the third column shows the remaining models after the iteration. In the first iteration model number 3, the small redorange torus was chosen to be removed, and all its associated points happen to move to model number 4, the small orange torus. This is actually an example of two very similar models that were recognized from the same group of points and one of them remained after the classification stage. In the second iteration 9, the blue torus was chosen to be removed, and all its associated points happen to move to model number 2, the red colored cone. In the third iteration model number 8, the cyan torus, was chosen to be removed, and its points moved to model number 7, the closest green torus. In the fourth iteration no model was removed because the $\Delta\sigma$ of all the remaining models was larger then $\Delta \sigma_{MAX}$. Therefore the algorithm terminated. In Figure 15(a) the final result is presented. In this figure the five tori and two cones were detected and their associated points are colored in different colors. It can be seen that all false detected models were removed and all models were detected correctly.

After these stages are completed only the correct models remain. In the final stage we verify that the points chosen by the grade function actually belong to the model. Using the density function of the grade, points between the two local minima are identified as belonging to the model. Figure 16 shows the detection of the cone points. Figure 15(b) shows the final result of whole algorithm. The main difference between the two results is that the cones' plane cross sections (which is not part of the our definition of a cone) are now not associated with the cone models because even though these points are closest to the cone the algorithm detected that they not part of the model.



Fig. 16. Final point association to cone model.

VIII. EXPERIMENTAL RESULTS

We have tested our application on a large number of range images in different scenes of partially occluded geometric primitives and non-primitive objects. Our application succeeded to recover accurately every primitive as long as a minimal number of its points (relatively to other objects or other primitives of its type) were sampled in the range image. All the range images used in our experiments were acquired by the Cyberware Laser Scanner, Model 3030 (1993 model)[14]. The scanning process captures an array of digitized points, with each point represented by X, Y, and Z coordinates. The sampling pitch in the X direction is $500\mu m$, in the Y direction is $350\mu m$, and in the Z direction is $75-300\mu m$.

A. Mean Shift Based Technique

We present here two typical results of running the mean shift based algorithm. The first scene, consists of two identical spherical objects placed on the same horizontal plane, two different cylinders in different orientations and one box which was placed such that three of its facets are visible (Figure 17). The ground truth of the cylinders and spheres radii was obtained by physical measurements. The error of these measurements is $\pm 0.01cm$. Quantitative recovered results are summarized in Table I: for radii of primitives, when it is



Fig. 17. Real scene of partially occluded primitives: (a) Range image; (b) Local peaks at the end of the first stage illustrated upon the center of the principal curvatures histogram.

relevant, a comparison of the recovered to the measured values is presented. The units are in centimeters.

At the end of the first stage, four local peaks were located at (-0.394332, -0.394332) with 6669 scene points, (-0.003446, -0.334160) with 14,843 points, (0.000462, -0.677247) with 4068 points and (-0.002089, -0.003453) with 2484 scene points (Figure 17). The four local peaks reflect the scene as expected. All primitives were detected correctly and there were no false alarms. The box was recovered as 3 planes. Although we did not make any accurate measurements of the features of the scene, all the checks we have performed agreed with the recovery results. For example, the inner product of the normals of the three recovered planes are 0.012438, 0.000241 and 0.000965 - as expected since the planes are orthogonal to each other. The centers of the two identical spherical elements have almost the same value for their vertical coordinate. Again, this was expected since they were placed at the same vertical height. The smaller cylinder's main axis is practically horizontal and indeed this was its orientation as can be seen in the scene's illustration.

The second presented scene contains free-form objects together with several primitives (Figure 4). We placed in this scene two different cylinders (with different radii and orientations), one box, one sphere and two general and non-primitive objects. The configuration of the objects creates partial occlusion of some of the primitives. In Figures 5, 7 and 9 the mean shift process is demonstrated with the second scene.

The results are summarized in Table II. Again all primitives were detected correctly. The inner product between the normals of the 3 orthogonal plane's are 0.00214, 0.00039 and 0.01062. The visible base of one of the cylinders was recovered as an additional plane

TABLE I

 $SCENE \ (Figure \ 4 \ \text{and} \ 18 \ (\text{a})) \ \text{with} \ 49626 \ \text{points} \ \text{of primitives} \ \text{only} \ \text{-} \ \text{recovered results} \ \text{vs.} \ \text{measured features}.$

Cylinde	rs	number of points	radius[cm]	Normal	Center[cm]
_	Mean Shift	16249	2.993	-0.303 0.952 -0.028	3.459 0.000 -3.995
⊖ a ∣	optimization	16259	2.915	-0.230 0.960 -0.023	4.620 -4.246 -3.836
	Measured	-	2.98	_	_
_	Mean Shift	7191	1.477	0.999 -0.010 -0.014	0.000 -0.960 1.277
b	optimization	7098	1.431	0.999 0.000 0.005	1.006 -0.994 1.366
	Measured	_	1.41	_	_
Spheres		number of points	radius[cm]		Center[cm]
-	Mean Shift	6609	2.536		-1.572 -4.556 -2.768
a	optimization	6609	2.573		-1.562 -4.527 -2.806
	Measured	-	2.52		-
_	Mean Shift	8850	2.536		4.195 -4.529 1.387
🛑 в	optimization	8849	2.520		4.202 -4.520 1.402
	Measured	_	2.52		_
Dlanag		number of		Namal	D[am]
rialles		points		INOTITIAL	D[cm]
	Mean Shift	2431		-0.637 0.661 0.396	-2.107
📖 a	Measured	-		-	-
	Mean Shift	2749		-0.018 -0.540 0.841	-4.180
b b	Measured	-		-	-
	Mean Shift	2083		0.770 0.529 0.355	5.067
c c	Mean Shift	-		-	-

TABLE II

A scene with 60330 points (Figures 17 and 18(b)) of primitives and free-form objects - recovered results vs. measured features.

Cylinders		number of points	radius[cm] Normal		Center[cm]	
	Mean Shift	15854	3.070	0.757 -0.650 -0.009	0.000 2.662 -4.205	
	optimization	15671	2.970	0.826 -0.563 -0.017	0.000 2.560 -3.994	
	Measured	_	2.98	-	-	
	Mean Shift	3123	1.443	0.899 -0.320 0.297	0.000 0.992 -1.878	
b 🖯	optimization	5085	1.413	0.944 -0.166 0.285	0.000 1.887 -1.780	
	Measured	_	1.41	-	-	
Sphere		number of points	radius[cm]		Center[cm]	
	Mean Shift	6330	2.575		3.374 -3.251 0.443	
	optimization	6261	2.457		3.357 -3.256 0.560	
	Measured	_	2.52		-	
Diaman		number of		N	D[]	
Planes		points		INOrmai	D[cm]	
	Mean Shift	2099		0.343 0.825 0.447	1.451	
a 🗆 a	Measured	_		-	-	
— ,	Mean Shift	3760		-0.803 0.014 0.596	-2.795	
b 🔤	Measured	-		-	-	
c c	Mean Shift	1511		0.494 -0.564 0.661	-3.594	
	Measured	-		-	-	
d	Mean Shift	530		0.914 -0.291 0.282	4.215	
	Measured	_		-	-	

and indeed its normal is very close to the direction of cylinder's axis.

Finally, in Figure 18 we show results of these two scenes together with two other scenes. For each primitive we show its wireframe model and which points have been selected by the mean shift procedure to be inliers. These points are then given to a least squares optimization procedure which improves the quality of the recovered model. Points which are selected as inliers using the improved model are shown in the second figure. In all scenes the primitives are partially occluded but in Figure 18(d) the large green cylinder is not even contiguous. Even so the algorithm is able to recover it as a single cylinder.



Fig. 18. Mean shift classification results

B. RANSAC based technique

We present here three results of running of the RANSAC based algorithm. Pairs of scenes were merged to generate more difficult inputs for the algorithm by increasing the percentage of outliers. This was done to make the example more challenging for the algorithm. This results of course in an increase in runtime.

The first scene which was used to illustrate the algorithm in Section VII is shown in Figures 19 and 20 consists of five tori and two identical cones, which were all detected by our system. Figure 19(a) shows the distribution of the modes in the (κ_1, κ_2) space. The tables in the figures show the modes which were found by the mean shift algorithm. Each colored point represents points belonging to a mode. For modes with $\kappa_1 \approx 0$ the cone recovery process is run.

At this stage most of the points on both cones belong to the same cluster. Only in the RANSAC stage they are separated. Once one of the cones is detected, the points belonging to it are removed from the data set and the algorithm is run on the remaining points. This process is repeated until no models are left. For torus detection only modes with $\kappa_1 \neq 0$ are selected. As can be seen in Figure 19(b) most of the points lying on the cones have been removed. For all of the points belonging to the modes shown in this figure the torus recovery RANSAC process is run.

To demonstrate the quality of the result we drew the recovered shapes on the scene. Figure 20(a) shows the best result achieved by the RANSAC process. As can be seen a few points have not been classified as belonging to any object. After the least squares optimization (Figure 20(b)), this problem has been solved and the quality of the models has also improved. The results are especially striking for the small tori for which only a small fraction of the points were recovered by the RANSAC process but after optimization the results have improved considerably. Figure 20(c) shows the grades of all model after applying the final classification stage and Figure 20(d) shows the final result without the false models. In all the experiments the results are close to the ground truth as can be seen in Table III. For example, in this experiment the two small orange and yellow tori are detected with similar radii. The angle of the two cones is also similar. The two green tori were detected with the same radii also. The angle between the directions of central axes of the blue torus which is placed on the red cone and the red cone is 5.34° . The brown and red cones are partially occluded and not even contiguous. Even so the algorithm is able to recover them as a single cone.



(a) modes of (κ_1, κ_2) (b) modes for tori (with $\kappa_1 \neq 0$). Fig. 19. First scene curvature map

The second scene contains six tori, a cone, and a plane. Figure 21 shows the curvature map. The recovery and classification results are shown in Figure 22. As in the previous example the optimization stage improves the

Cones		number of points	Center [cm]	Normal	$\alpha [^{o}]$	
	RANSAC	18663	-4.78 -0.89 15.49	0.868 0.157 -0.472	4.330	
1	optimization	18856	-3.62 -0.87 15.14	0.8620.164 -0.479	4.419	
- 1	measured	_	-	-	4.93	
	RANSAC	10139	63.78 37.28 9.25	0.793 0.601 0.099	1.202	
$ \land 2$	optimization	16011	25.58 7.58 5.79	0.807 0.564 0.176	4.566	
	measured	-	-	-	4.93	
Tori		number of	Center [cm]	Normal	R [cm]	r [cm]
	DANGAC	2202	21 50 2 56 9 01	0.244.0.7(0.0.501	0.214	0.465
	RANSAC	3202	21.59 2.56 8.01	0.244 0.769 0.591	2.314	0.465
4	optimization	6053	21.60 2.52 8.05	0.24/ 0.750 0.613	2.301	0.496
	measured	-	-	_	2.30	0.43
	RANSAC	1086	7.35 -9.32 8.03	0.229 -0.752 -0.618	2.416	0.609
5	optimization	2096	7.34 -9.33 8.01	0.229 -0.742 -0.630	2.381	0.578
	measured	-	-	-	2.30	0.43
	RANSAC	6356	16.34 -7.90 6.93	-0.291 -0.941 -0.174	4.798	1.267
6	optimization	12190	16.35 -8.14 6.50	-0.281 -0.942 -0.185	4.780	1.693
	measured	_	-	-	4.85	1.65
	RANSAC	11942	4.30 5.30 4.15	0.066 -0.688 -0.722	3.732	2.133
7	optimization	15335	4.45 3.32 2.66	-0.091 0.983 0.162	4.328	1.990
	measured	-	-	-	4.85	1.65
	RANSAC	6892	7.50 -4.97 2.15	0.792 0.477 0.381	2.772	1.355
10	optimization	8240	7.18 -5.51 1.44	0.859 0.490 0.148	3.361	1.350
- 10	measured	_	-	-	3.24	1.47

 TABLE III

 First scene with 105934 points (Figures 12-16, 19 and 20) results





(a) best RANSAC results





(b) post-optimization results

(c) final classification grades (d)final classification result

Fig. 20. First scene segmentation and recognition

quality of the results and the final classification stage removes the false models. Figure 22(c) shows the classification grades in the first iteration of final classification, Figure 22(d) shows the grades in the last iteration, and Figure 22(e) shows the grades of the models that were classified as false models during the algorithm. At the end of classification algorithm all initially detected plane models which were actually patches of other tori were classified as false models and the points were transferred to the tori on which they lie. In this experiment the



2.5 × 10⁶

results are also close to the ground truth as can be seen in Table IV. For example in this experiment tori 19 and 18 are placed on plane 13 thus the normal to the plane and the directions of the central axes of the two tori should be similar. The results show that the angles between the three vectors are at most 3.6° . The radii of the same torus pairs are very close: 16 and 17, 18 and 20, and

The third scene contains five tori and one cone. The recovery and classification results are shown in Figure 23. In this experiment the results are also very close to the ground truth as can be seen in Table V. This scene contains two small tori: an orange torus 2

19 and 22, respectively.

Fig. 21. Second scene curvature map k1- k2 modes

Planes		number of		Normal	D [cm]	
	DAMOA C	points		0.0010.0.0020.0.4207	0.44	
	RANSAC	3992		0.0019 0.9029 0.4287	2.44	
13	optimization	4924		0.000 0.905 0.425	2.389	
	measured	-		_	-	
Cones		number of	Center [cm]	Normal	o [0]	
		points		Normai	α[]	
	RANSAC	7439	-6.12 1.84 -3.53	0.974 0.064 0.218	4.218	
14	optimization	8162	-3.63 1.69 -2.87	0.974 0.075 0.215	4.548	
	measured	-	-	-	4.93	
. .		number of		NT 1	R [cm]	г 1
Iori		points	Center [cm]	Normai		r [cm]
	RANSAC	2317	22.84 -5.85 10.11	-0.213 -0.688 -0.694	2.320	0.457
16	optimization	4423	22.84 -5.87 10.05	0.214 0.694 0.687	2.296	0.520
	measured	_	-	-	2.30	0.43
	RANSAC	2982	21.58 2.54 8.16	0.260 0.742 0.618	2.402	0.478
17	optimization	6119	21.60 2.56 8.05	0.248 0.746 0.618	2.306	0.505
	measured	_	-	-	2.30	0.43
	RANSAC	13960	18.91 -3.05 5.30	0.042 0.881 0.471	2.937	0.918
	optimization	17019	18.92 -3.38 5.12	0.041 0.881 0.472	3.119	1.324
	measured	_	-	-	3.24	1.47
	RANSAC	25719	8.01 -3.28 7.16	0.007 0.902 0.432	4.861	0.971
19	optimization	30055	7.90 -3.56 6.97	-0.004 0.889 0.458	4.843	1.363
	measured	-	-	-	4.85	1.65
20	RANSAC	16777	12.90 3.52 8.81	0.063 0.885 0.462	3.040	1.028
	optimization	18691	12.81 3.29 8.64	0.080 0.881 0.467	3.118	1.298
	measured	-	-	-	3.24	1.47
	RANSAC	12554	4.58 3.49 2.81	0.052 -0.980 -0.193	4.802	1.376
22	optimization	13758	4.52 3.31 2.61	0.006 0.984 0.177	4.756	1.685
	measured	—	-	-	4.85	1.65

 TABLE IV

 Second scene with 137714 points (Figures 21 and 22) results

and a blue torus 9, their recovered parameters are quite close. The radii recovered for the green torus 10 and the cyan torus 13 are similar also. In this example there are groups of tori which were detected several times by the RANSAC algorithm and at the end of the final classification algorithm from each group only the best fitted one torus remained.

The algorithm was implemented in Matlab. The runtimes of each part of algorithm are summarized in Table VI. Note that most of the runtime was consumed by the standard least squares algorithm. The algorithm's runtime can be improved by converting the code to C



Fig. 22. Second scene segmentation and recognition

Fig. 23. Third scene segmentation and recognition

Cones		number of points	Center [cm]	Normal	α [^o]	
	RANSAC	10975	28.81 -6.40 12.43	-0.805 0.085 -0.587	6.281	
1	optimization measured	11176 -	31.39 -6.69 13.78	-0.823 0.089 -0.561	4.870 4.93	
Tori		number of points	Center [cm]	Normal	R [cm]	r [cm]
	RANSAC	2389	2.04 -4.34 3.05	0.735 -0.566 -0.373	2.372	0.513
	optimization	2352	2.04 -4.32 2.93	0.750 -0.571 -0.334	2.296	0.580
	measured	-	-	-	2.30	0.43
_	RANSAC	5341	21.58 2.34 7.87	-0.260 -0.784 -0.564	2.270	0.611
9	optimization	5728	21.58 2.38 8.02	0.241 0.759 0.604	2.249	0.553
	measured	_	_	-	2.30	0.43
	RANSAC	19958	7.96 -3.50 2.79	0.199 -0.750 -0.631	3.391	1.786
10	optimization	20008	7.74 -3.27 3.03	0.147 -0.774 -0.616	3.145	1.395
	measured	-	-	-	3.24	1.47
	RANSAC	18884	12.85 3.28 9.16	0.095 0.812 0.575	2.867	1.180
13	optimization	18982	12.80 3.14 8.64	0.079 0.882 0.464	3.105	1.308
	measured	-	-	-	3.24	1.47
	RANSAC	13599	4.42 4.03 2.42	0.008 0.958 0.285	5.495	1.288
14	optimization	13735	4.50 3.13 2.66	0.001 0.986 0.169	4.695	1.697
	measured	-	-	-	4.85	1.65

 TABLE V

 Third scene with 98208 points (Figure 23) results

and by running it on only a sub-sample of all the scene points.

TABLE VI Algorithm runtime results in seconds

	Scene1	Scene2	Scene3	Average
number of points	105934	137714	98208	113952
Figures	12-16,	21, 22	23	
	19-20			
Curvature classification	17	23	14	18
RANSAC recovery	83	104	98	95
post-recovery	275	248	209	244
optimization				
final classification	11	23	8	14
Total	386	398	329	371

IX. CONCLUSIONS

We have presented a robust and accurate method for the recovery of 3D geometric primitives from complex cluttered range scenes. The application presents a new recovery approach based on robust statistics methods.

It maintains accurate recoveries even when the range images include partially occluded objects or contain in addition free-form objects. The application was tested on range images of various scenes. All primitives, in all scenes, were recognized and their parameters were recovered very accurately. We presented two types of algorithms: a fast mean shift based algorithm for robust recovery of the geometric primitives which was used for the more simple primitives. For the more complex primitives a more elaborate RANSAC/pbM based algorithm was presented which robustly estimates the model. In the final step of our algorithm a new method to remove false detections has also presented. This method can be extended to deal with more complex parametric surfaces. The recovery application also demonstrates the possibilities in using curvature and Darboux frame estimators within a practical application.

Although our method deals only with geometric primitives its results can be used to recognize more complex objects which can consist of primitive and non-primitive components as shown in [16], [1].

REFERENCES

- F. Arman and J. K. Aggarwal. Model-based object recognition in dense-range images: a review. ACM Comput. Surv., 25(1):5–43, 1993.
- [2] P. J. Besl. Surfaces in range image understanding. Springer-Verlag New York, Inc., New York, NY, USA, 1988.
- [3] P.J. Besl and R.C. Jain. Segmentation Through Variable-Order Surface Fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):167–192, 1988.
- [4] T. Chaperon and F. Goulette. Extracting cylinders in full 3-D data using a random sampling method and the gaussian image. In Vision, Modeling, And Visualization, 2001.
- [5] H. Chen and P. Meer. Robust regression with projection based m-estimators. In *Proceedings of Ninth IEEE International Conference on Computer Vision*, pages II:878–885, October 2003.

- [6] H. Chen and P. Meer. Heteroscedastic projection based Mestimators. In Workshop on Empirical Evaluation Methods in Computer Vision, June 2005.
- [7] X. Chen and F. Schmitt. Intrinsic surface properties from surface triangulation. In *Proc. European Conf. Comp. Vision*, pages 739– 743, 1992.
- [8] Y. Cheng. Mean shift, mode seeking and clustering. *IEEE Trans.* on Patt. Anal. and Mach. Intell., 17(8):790–799, August 1995.
- [9] P. Chiabert and M. Costa. Statistical modelling of nominal and measured mechanical surfaces. J. Comput. Inf. Sci. Eng., 3(1):87– 94, 2003.
- [10] O. Chum and J. Matas. Matching with prosac "progressive sample consensus. In CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1, pages 220–226, Washington, DC, USA, 2005. IEEE Computer Society.
- [11] O. Chum, J. Matas, and J.V. Kittler. Locally optimized RANSAC. In DAGM03, pages 236–243, 2003.
- [12] D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. ACM Trans. Graph., 23(3):905–914, 2004.
- [13] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 24(5):603–619, May 2002.
- [14] Cyberware. Model 3030 color 3d scanhead. http://www.cyberware.com/products/scanners/3030.html.
- [15] N.P. Draper and H. Smith. Applied Regression Analysis. New York : Wiley, second edition, 1981.
- [16] T.J. Fan, G. Medioni, and R. Nevatia. Recognizing 3-d objects using surface descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1140–1157, 1989.
- [17] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, June 1981.
- [18] P.F.U. Gotardo, O.R.P. Bellon, K.L. Boyer, and L. Silva. Range image segmentation into planar and quadric surfaces using an improved robust estimator and generic algorithm. *IEEE Systems, Man, and Cybernetics B*, 34(6):2303–2315, Dec 2004.
- [19] E. Hameiri and I. Shimshoni. Estimating the principal curvatures and the Darboux frame from real 3D range data. *IEEE Systems, Man, and Cybernetics B*, 33(4):626–637, August 2003.
- [20] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.
- [21] A. Hoover, G. Jean-Baptiste, X. Jiang, P.J. Flynn, H. Bunke, D.B. Goldgof, K. Bowyer, D.W. Eggert, A. Fitzgibbon, and R.B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996.
- [22] D. Marshall, G. Lukacs, and R. Martin. Robust segmentation of primitives from range data in the presence of geometric degeneracy. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(3):304–314, March 2001.
- [23] M. J. Milroy, C. Bradley, and G. W. Vickers. Segmentation of a wrap-around model using an active contour. *Computer-aided Design*, 29(4):299–320, 1997.
- [24] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. Numerical Recipes in C. Cambridge University Press, 1988.
- [25] D. Rogers and N. Fog. Constrained B-spline curve and surface fitting. *Computer Aided Design*, 21(10):641–648, 1989.
- [26] S. Rozenfeld and I. Shimshoni. The modified pbM-estimator and a runtime analysis technique for the RANSAC family. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages I:1113–1120, 2005.
- [27] C. M. Shakarji. Least-squares fitting algorithms of the nist algorithm testing system. *Journal of Research of the National Institute of Standards and Technology*, 103(6):663–641, Nov 1998.

- [28] M. Singh, H. Arora, and N. Ahuja. A robust probabilistic estimation framework for parametric image models. In *Proc. European Conf. Comp. Vision*, pages I:508–522, 2004.
- [29] D.R. Smith and T. Kanade. Autonomous scene description with range imagery. *Computer Vision, Graphics, and Image Processing*, 31(3):322–334, September 1985.
- [30] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proc. Int. Conf. Comp. Vision*, pages 902–907, 1995.
- [31] P. H. S. Torr and A. Zisserman. Mlesac: a new robust estimator with application to estimating image geometry. *Comput. Vis. Image Underst.*, 78(1):138–156, 2000.
- [32] E. Trucco and R. B. Fisher. Experiments in curvature-based segmentation of range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):177–182, 1995.
- [33] H. Wang and D. Suter. Mdpe: A very robust estimator for model fitting and range image segmentation. *Int. J. of Comp. Vision*, 59(2):139–166, September 2004.
- [34] M. Weiyin and J. P. Kruth. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Computer-aided Design*, 27(9):663–675, 1995.
- [35] J. Wu and L. Kobbelt. Structure recovery via hybrid variational surface approximation. *Computer Graphics Forum*, 24(3):277– 284, 2005.
- [36] C. Yang, R. Duraiswami, D. DeMenthon, and L. Davis. Meanshift analysis using quasi-newton methods. In *IEEE International Conference on Image Processing (ICIP)*, pages III:447–450, 2003.