Using Recover-and-Select Paradigm for Simultaneous Recovery of Planes, Second Order Surfaces and Spheres from Triangulated Data *

Bojan Kverh and Aleš Leonardis
Faculty of Computer and Information Science
University of Ljubljana
Tržaška 25, Ljubljana, SI-1001, Slovenia
e-mail: {bojank,alesl}@fri.uni-lj.si

Abstract

In this paper we present an approach to segmentation of triangulated data by recovering different geometrical models. We show that the Recover-and-Select paradigm can be generalized to various kinds of data, where neighbourhood relation between points can be defined in a proper way. We also show that the Recover-and-Select paradigm is able to simultaneously recover different kinds of models from the data, and it is able to select the models that best fit certain subsets of data points.

1 Introduction

One of the important tasks in computer vision is to obtain a precise and compact 3D model of an object from data captured with a certain sensor. This task consists of several phases including capturing data, segmentation of the data to surfaces, detecting relationship between surfaces and creating the final model. In this paper we deal with the segmentation of the data and extraction of geometrical properties of the surfaces. These properties are expressed as the parametric geometric model that describes a single surface. So called Recover-and-Select paradigm is able to recover different kinds of parametric geometric models from the data and to segment the data simultaneously [3].

The main object of this paradigm is a description. The description consists of a set of points (so called region) and the parametric model defined on them. The standard Recover-and-Select paradigm uses only one type of models [1, 2, 3, 6, 9]. A more general

*This work was supported by the Ministry of Science and Technology of Republic of Slovenia (Project J2-6187), European Union Copernicus Program (Grant 1068 RECCAD), and by U.S.–Slovene Joint Board (Project #95-158).

approach can extend the paradigm to use several types of models [5]. The paradigm has been concisely defined also in an object-oriented framework [2].

The model tries to approximate the points as accurately as possible. The criterion which shows how accurate the model is, is the average of distances of all the region points from the model. This criterion is called the model error. An ideal model would have the error equal to 0 but in general it is not possible to obtain such a model. The paradigm consists of three stages:

1. Placing seed descriptions,
2. Description growing,
3. Description selection.

While stage 1 is performed only at the begining, stages 2 and 3 are repeated several times until all remaining descriptions are fully grown. Stage 3 selects the most significant models with the Minimum Description Length (MDL) criterion and rejects the others in order to decrease the total number of models and thus redundancy, since regions of descriptions start to overlap each other. As the final result, the set of descriptions is obtained that tries to approximate the image data. In this paper we extend the paradigm to triangulated data and to simultaneous use of different types of models (namely planes, second order surfaces and spheres).

The rest of this paper is organized as follows: the input data and the neighbourhood relation are discussed in section 2, and the Recover-and-Select paradigm on triangulated data is described in section 3. We discuss the differences from the standard Recover-and-Select paradigm when using many different types of models in section 4. Some experimental results are given in section 5, which is followed by conclusion.

2 Input data

The Recover-and-Select paradigm has usually been used in conjunction with range images on a regular grid, where the whole image can be written as

$$z = f(x, y) \tag{1}$$

If the input image has width w and height h, then $1 \leq x \leq w$ and $1 \leq y \leq h$. The definition of neighbourhood relation between image points in this type of range images is straightforward — the points (x_i, y_i, z_i) and (x_j, y_j, z_j) are neighbouring points if and only if Eq. (2) is satisfied (so called 8-connectivity),

$$\max(|x_i - x_j|, |y_i - y_j|) = 1. \tag{2}$$

2.1 Triangulated data

Triangulated data is normally used to reduce the amount of the original input data. It consists of the (x, y, z) coordinates of each point and a triangle definition. The triangle definition consists of triples of integers denoting indices of points which form triangles. These triangles can be obtained either directly from the sensor scanlines or by applying alpha shapes [8] on a cloud of points. For our experiments the former method has been used. Probably the most important advantage of triangulated data over range images on a regular grid is that the data of the whole object can be represented with the triangulated data. On the other hand, only data from a single view can be represented by range images on a regular grid. As triangulated data still contains a huge amount of surface points, it is not suitable to be directly used for tasks like object recognition, reverse engineering, etc. Triangulated data is suitable for visualization with the appropriate tools (e.g. Geomview) and for modeling free-form surfaces.

2.2 Neighbourhood relation in triangulated data

The neighbourhood relation for triangulated data is defined as follows: points with indices i and j (p_i and p_j respectively) are neighbouring if and only if there is a triangle in the data with integers i, j and k (the order of the integers is not important), where k is an arbitrary point index (see Figure 1). As the neighbourhood relation is symmetric, it holds that if p_i is the neighbour of p_j then also p_j is the neighbour of p_i.

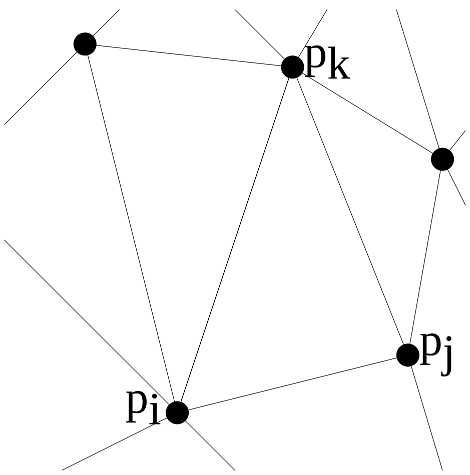

Figure 1: If two points are part of the same triangle, then they are neighbours

Neighbourhood of a single point is defined as the set of all point neighbours. Neighbourhood of a region is a set of all points that are not in the region and are neighbours of at least one point from the region.

3 Recover-and-Select paradigm on triangulated data

As it was stated before, the Recover-and-Select paradigm consists of placing seed descriptions, growing descriptions, and selecting descriptions. The flow-chart diagram depicting the stages is shown in Figure 2. The only difference between using the paradigm on triangulated data and range data on a regular grid is in determination of the neighbourhood as stated in previous section. Thus we have to apply certain changes to the stages that use the neighbourhood relation, that is, placing seed descriptions and growing descriptions. The selection stage does not need to be changed for triangulated data.

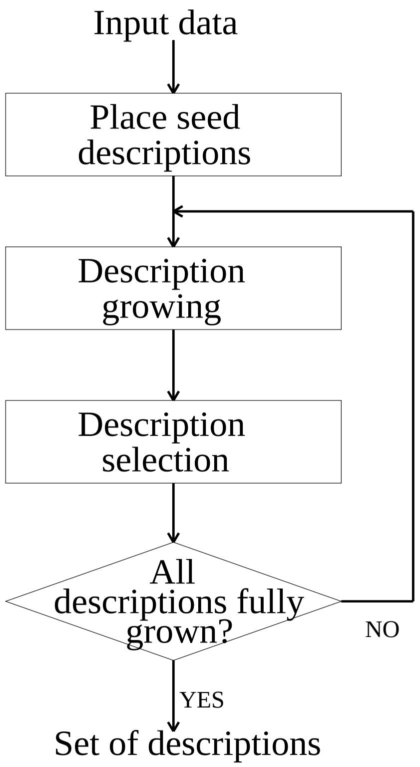

Figure 2: Stages of the Recover-and-Select paradigm

3.1 Placing seed descriptions

This stage is trivial when dealing with range images on a regular grid. Since range data is placed on the grid, it is possible to choose rectangular regions that spread uniformly over the image. Seed size can be defined by its width and height. The model is then fit to the region points, and the description is formed by the model parameters and the corresponding region. The description is not accepted if its model error is too large.

In the case of triangulated data the situation gets slightly more complicated. Since in general there are no two points with the same x and y coordinate, it is not possible to place a rectangular seed region on triangulated data. So our approach is the following:

from a set of data points we choose an arbitrary point and put it into the seed description region. Then we repeatedly merge to this region its neighbourhood until the number of points in the region reaches some predefined threshold, and then try to fit a model to the region. Three steps of merging the neighbourhood to the region are shown in Figure 3. Points in black color belong to the seed description region at a certain moment. When

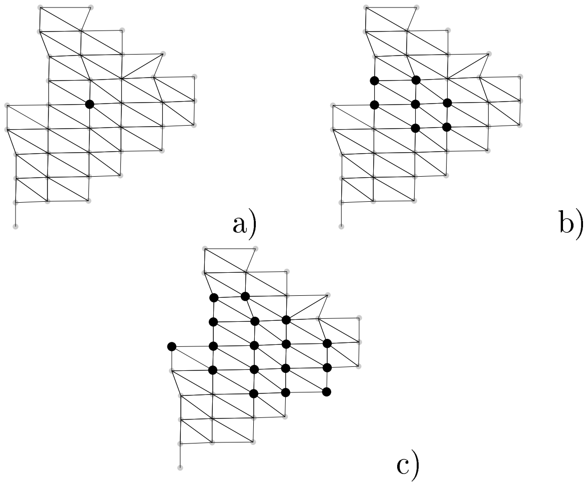

Figure 3: Steps of joining points to the seed region

placing the next seed description, we do not take into consideration the points that were already used in this process in order to prevent overlap between seed description regions.

Our approach is formally outlined in the following algorithm, which also takes care of special cases:

Let W be the set of all data points
repeat
    Choose one point p from W and let $R = \{p\}$
    Let R_n be the neighbourhood region of R
    while number of points in R is less than seed_size **and**
    number of points in R_n is greater than zero **do begin**
        Let $R = R \bigcup R_n$
        Let R_n be the neighbourhood region of R
    end while
    if number of points in R is greater or equal to seed_size **then begin**
        Fit the model to region R and put both into description D
        if the error of D is larger than some threshold **then**
            Destroy D
        end if
    end if
    Let $W = W \setminus R$
until $W = \emptyset$

When a model is fit to the description region, the description is destroyed if its error is too large. The constant seed_size determines the minimal number of points for a single seed description. The fact is that the seed regions are not always created with the same number of points. This could cause some problems at the selection stage although we didn't notice any in our experiments. Probably a better way to place seed descriptions is to restrict the regions to have the same number of points.

3.2 Description growing

One iteration of description growth is formed by the following steps:

> Let R_n be the neighbourhood of R
> Exclude from R_n all the points not compatible with description model
> Merge all the points from R_n to description region
> Fit new description model to description region
> **if** description model error is too large **then**
>   Retain the old description
> **end if**

The description stops growing when its error is too large or there are no points in its neighbourhood that are close enough to the description model. Computation of the neighbourhood of the region was described in the previous section. A more detailed description of description growing based on range images can be found in [2, 3]. An arbitrary point is compatible with a description model if and only if the distance of this point from the description model is less than some threshold. The distance measures for the models used in our work are presented in the next section.

Before each description selection stage, several description growing iterations are performed. The number of iterations is doubled after each description selection stage. The reason for this is that through the process of growing the description regions are getting larger, so it takes more growing iterations for one description to become redundant after n-th selection stage than after $(n-1)$-st one.

3.3 Description selection

Description growing requires a lot of computation time, especially when the number of descriptions is high. Since many of the descriptions become redundant through the description growing process i.e., some regions are almost completely overlapped by regions of other descriptions, the number of descriptions is decreased from time to time by description selection based on the MDL principle [3, 4, 7]. We define an objective function [3] $F(\mathbf{d})$ where $\mathbf{d} = [d_1, d_2, \ldots, d_n]$ and n is the number of descriptions that enter the selection stage. The value of d_i is 1 if the i-th description is selected and 0 otherwise. From the optimal solution vector $\mathbf{m}$, which satisfies Equation (3), the selection process can determine which descriptions have to be rejected;

$$F(\mathbf{m}) = \max_{\mathbf{d} \in \{0,1\}^n} F(\mathbf{d}). \tag{3}$$

To obtain the solution vector $\mathbf{m}$ we use a greedy algorithm. More about description selection can be found in [2, 3, 1].

4 Recover-and-Select paradigm with several types of models

In previous section we explained a standard Recover-and-Select paradigm on triangulated data, which uses only one type of models. In general, it is often not sufficient to use only one type of models. If we deal with data that does not fit well to our type of models, it will require a large number of descriptions to provide reasonable results. Because of that, we generalize the paradigm to employ different types of models.

The main problem when dealing with different types of models is that the descriptions of one type of models may grow "faster" than the others. If the Recover-and-Select paradigm would perform the selection among all the descriptions before they are fully grown, there is a chance that all descriptions of one type of models would be rejected since they have not grown "fast" enough. This can be avoided by performing the standard Recover-and-Select paradigm to the descriptions of the same type of models separately. When all descriptions are fully grown, we perform selection over the whole set of resulting descriptions. This makes a difference only at the selection stage. Instead of performing the selection on all descriptions together at the same time, we perform selection m times (m is the number of different types of models used), each time taking into account only descriptions of the same type[1] (see Figure 4). Since the selection stage is also very time consuming ($O(n^2)$), where n is the number of descriptions, we can also save some computation time. Suppose that selection among n descriptions takes $T_1(n) = cn^2$ units of time, where c is some constant. Time needed to perform m selections, each time selecting among approximately $\frac{n}{m}$ descriptions, is calculated in Eq. (4). In the latter case, we have to take into account also the last selection that finally selects descriptions of all types. But at this point, the number of descriptions is usually low, so the last selection is not very time consuming.

$$T_2(n) \approx m \cdot c(\frac{n}{m})^2 = c\frac{n^2}{m} \tag{4}$$

It is clear that $T_2(n) < T_1(n)$.

As it can be seen from Figure 4, if there are m different types of models, the time complexity of the Recover-and-Select paradigm need not depend on m. The standard Recover-and-Select paradigm for each type of models can be processed in parallel. With

[1] A description type is denoted only by the type of its model.

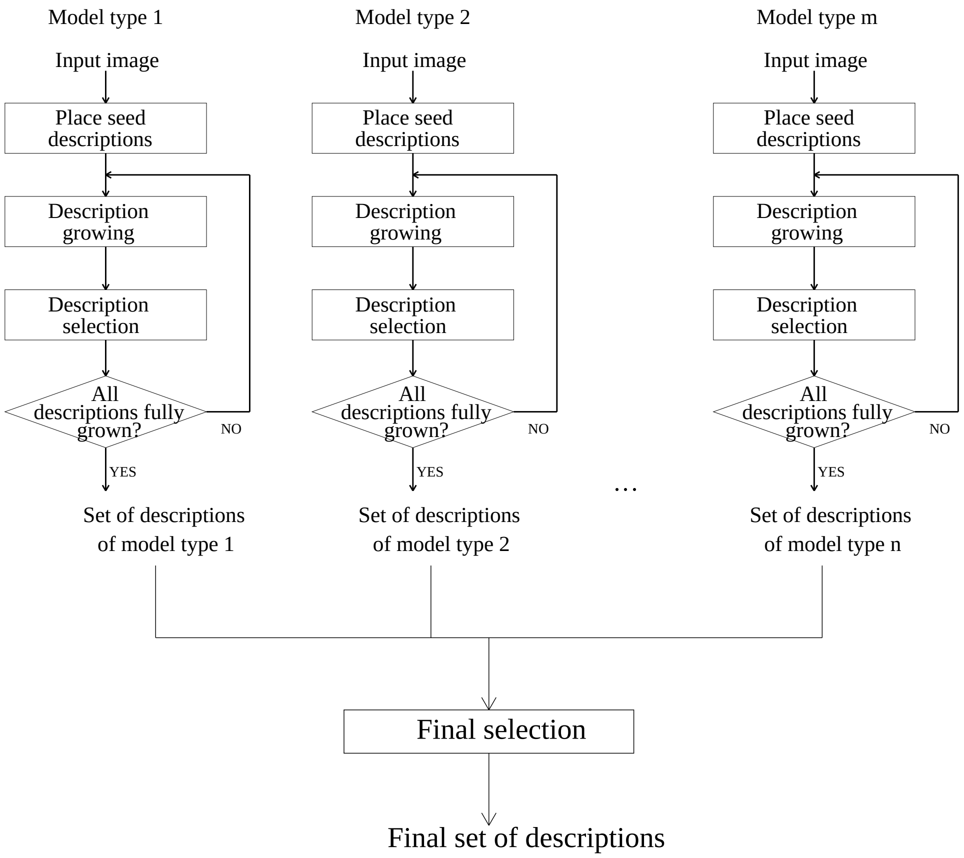

Figure 4: Recover-and-Select paradigm with several types of models

an appropriate computer system using many different types of models should be almost as fast as when using only one type of models.

4.1 Types of models: planes, 2^{nd} order surfaces and spheres

In our work we use planes, 2^{nd} order surfaces, and spheres as description models. Points (x, y, z) that satisfy Eq. (5), form a plane specified by a set of 4 parameters ($\{a, b, c, d\}$).

$$ax + by + cz + d = 0 \tag{5}$$

2^{nd} order surfaces and spheres have the following sets of parameters ($\{a, b, c, d, e, f\}$ and $\{a, b, c, r\}$, respectively) and consist of points that satisfy Eq. (6) and Eq. (7), respectively.

$$ax^2 + bxy + cy^2 + dx + ey + f - z = 0 \tag{6}$$

$$(x - a)^2 + (y - b)^2 + (z - c)^2 - r^2 = 0 \tag{7}$$

Table 1 summerizes the parameters and the distance measure for each used model.

Model	Model parameters	Distance of (x, y, z) from model
Plane	a, b, c, d	$\lvert ax + by + cz + d \rvert$
2^{nd} order surface	a, b, c, d, e, f	$\lvert z - ax^2 - bxy - cy^2 - dx - ey - f \rvert$
Sphere	a, b, c, r	$\left\lvert \sqrt{(a - x)^2 + (b - y)^2 + (c - z)^2} - r \right\rvert$

Table 1: Distance measures for a point (x, y, z) from a particular model.

5 Experimental results

In this section we present some of the results we achieved during the experiments. We extracted planes, 2^{nd} order surfaces, and spheres from the triangulated data of the object shown in Fig. 5. This object is composed of one sphere, box, and two cylinders.

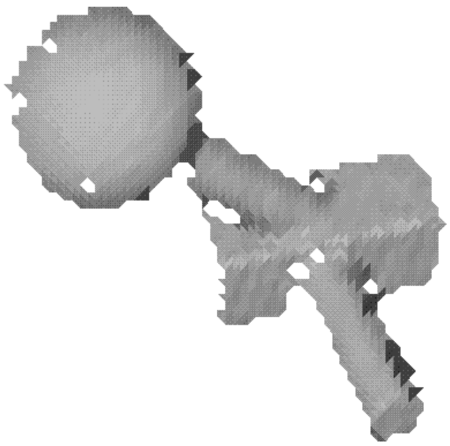

Figure 5: Triangulated data for experiments.

Seed descriptions on the triangulated data from Fig. 5 with plane, 2^{nd} order surface, and sphere models are shown in Fig. 6a, 6b and 6c, respectively.

After the seed descriptions had been placed over the data, the standard Recover-and-Select algorithm was run separately for each description type. The results of performing

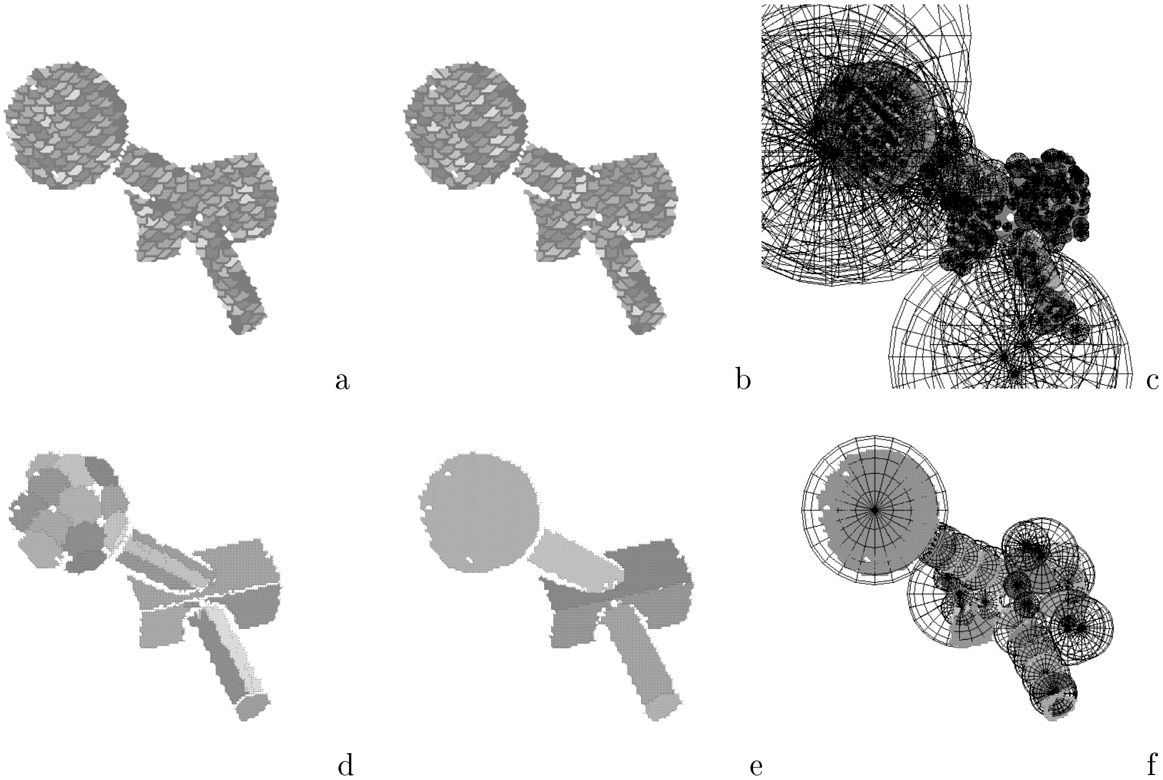

Figure 6: Seed descriptions for each description type are shown in a, b and c. The final results of performing the standard Recover-and-Select paradigm with each description type are shown in d, e and f.

the standard Recover-and-Select paradigm on each description type are shown in Fig. 6d, 6e, and 6f for planes, 2^{nd} order surfaces, and spheres, respectively. The final result (see Fig. 7) is obtained after all remaining descriptions of each individual type are presented to the last selection. The result consists of the following descriptions:

- three plane descriptions (two of them represent the surface of the box, while the third one represents the bottom part of one of the cylinders),
- two 2^{nd} order surface descriptions which represent surfaces of both cylinders,
- one sphere description which represents the sphere in the original data.

In Fig. 7 regions of different descriptions are colored with different gray levels. A sphere model is also drawn together with regions to give the impression of the size of the specific part. The bottom part of Fig. 7 shows the recovered parameters for each description.

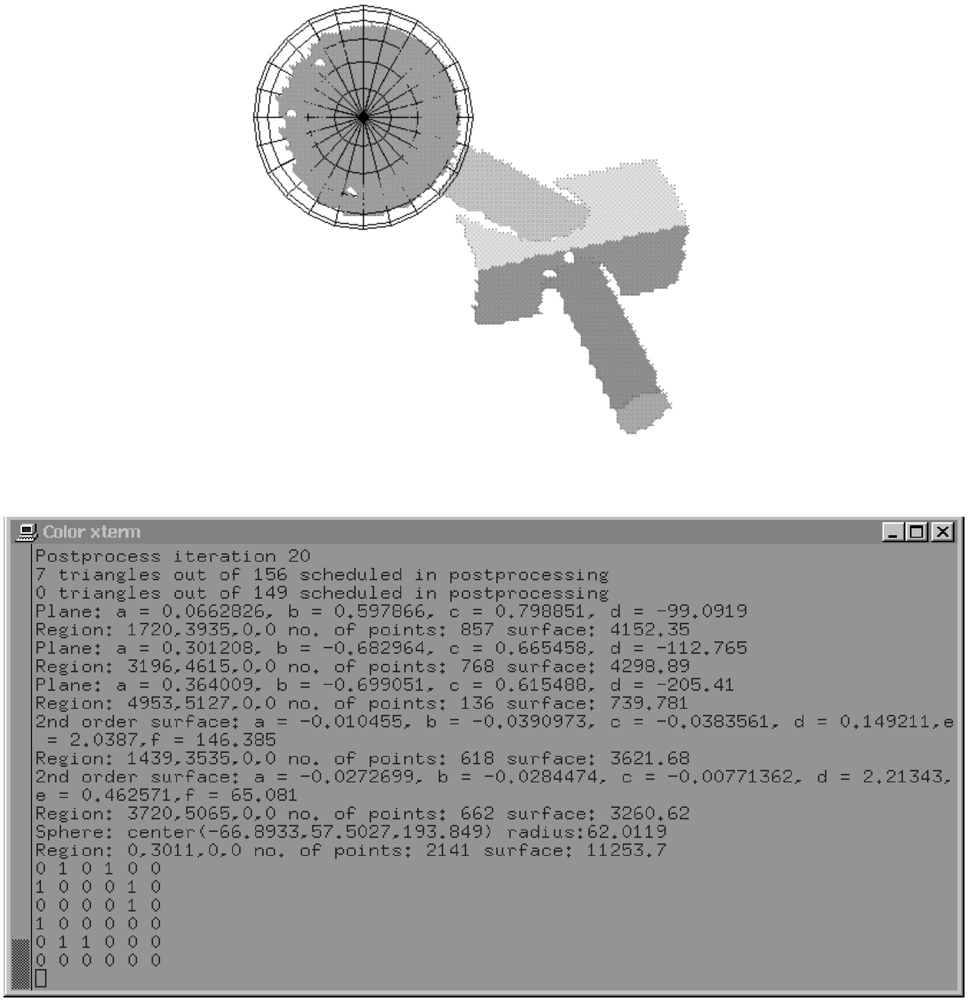

Figure 7: Final results

6 Conclusion

We showed that an automatic selection of the best geometric model of the 3D data is possible by combining the Recover-and-Select paradigm with different types of models. The set of geometric models that we used for our experiments can easily be extended with others like cylinders, cones, and also superquadrics. Further work will be directed towards improving the postprocessing and registration of data from multiple views based on descriptions extracted from all individual views. In this way we expect to be able to obtain description of the whole object and further, to create CAD model of the object.

References

[1] A. Jaklič. “Construction of CAD models from range images”, *PhD dissertation*, Faculty for Computer and Information Science, University of Ljubljana, 1997.

[2] A. Jaklič and A. Leonardis and F. Solina. “Segmentor: An Object-Oriented Framework for Image Segmentation”, In *Proceedings of the 3rd Slovenian-German and the 2nd SDRV Workshop*, pp. 331–340, Ljubljana, Slovenia, 1996.

[3] A. Leonardis and A. Gupta and R. Bajcsy. “Segmentation of Range Images as the Search for Geometric Parametric Models”, *International Journal of Computer Vision*, 14:253–277, 1995.

[4] M. Kovačič and B. Kverh and F. Solina. “Optimal Models for Visual Recognition”, In F. Solina, W. G. Kropatsch, R. Klette, R. Bajcsy, editors, *Advances in Computer Vision*, March, 1997.

[5] B. Kverh and A. Jaklič and A. Leonardis and F. Solina. “Simultaneous Recovery of Surface and Superquadric Models”, In *Proceedings of 20. Workshop of the Austrian Pattern Recognition Group (OAGM/AAPR)*, pp.27–36.

[6] B.Kverh and A.Jaklič and A.Leonardis and F.Solina. “Using Recover-and-Select Paradigm on Triangulated Data”, In *Proceedings of the Sixth Electrotechnical and Computer Science Conference ERK'97*, Vol. B, pp.241–244, Portorož, Slovenia, September 1997.

[7] M. Pilu and R. B. Fisher, “Recognition of Geons by Parametrically Deformable Contour Models”, In R. Cipolla and B. Buxton, editors, *Fourth European Conference on Computer Vision*, volume I of Lecture Notes in Computer Science, pages 71–82, April 1996.

[8] H. Edelsbrunner, E. P. Mücke, “Three-Dimensional Alpha Shapes”, In *ACM Transaction on Graphics*, Vol. 13, No. 1, pp.43–72, January 1994.

[9] A. Leonardis and A. Jaklič and F. Solina. “Superquadrics for Segmenting and Modeling Range Data”, In *IEEE Transaction on PAMI*, Vol. 19, No. 11, pp. 1289–1295, November 1997.