Surface Model Generation from Range Images of Industrial Environments

Angel D. Sappa
Computer Vison Center
Edifici O, Campus UAB
08193 Bellaterra - Barcelona, Spain
angel.sappa@cvc.uab.es

Abstract

This paper presents an hybrid segmentation technique that combines both the speed of an edge based approach with the robustness of a surface based approach. It consists of three stages. In the first stage a scan line approximation process extracts the edges contained into the given range image. These edges are later on used to define the positions of seed points. Through the second stage a two steps region growing technique is applied. First a 2D growing process enlarges the original seed points generating bigger regions. Next, each region is fitted to a plane and a cylinder. The one that best fit the given points is selected to represent that region and used during the 3D growing stage. The 3D growing stage is carried out taking into account the approximation error from candidate points to be added to the fitted surface. In this way, each surface is grown until no points can be added according to a user defined threshold. Finally, in the third stage, a post-processing algorithm merges neighbour regions that belong to the same surface. Experimental results by using industrial environments are presented.

1. Introduction

In spite of the great improvements carried out during the last years, 3D modelling is still an open field with a lot of work to be done in different domains. One of these domains is the automatic segmentation of range images that has attracted the attention of the computer vision community. The latter is due to the fact that it is more direct to obtain segmented representations from range images than from intensity images.

However, despite the effort, the extraction of CAD models, geometrical primitives in general, is still a handmade work in most of the commercial packages existing in the market. It means that a scene containing more than eleven million of points, scanned in about five minutes [1], will require several hours to be represented as a CAD model. This problem becomes a big bottleneck when industrial environments have to be processed. There, hundred of range images are required—an average of six hundred for a refinery or a petrochemical facility. In this way, the range image segmentation, which is an attractive point in the research field, becomes a so hard and boring work in modelling tasks. It takes a lot of time, which not only means money but also means that works with hard time constraints cannot be tackled.

Several approaches have been proposed in the literature for range image segmentation. They can be classified as: *1)* edge-based, *2)* region-based and *3)* hybrid techniques.

Edge based approaches (e.g., [2], [3]) consist in considering that every surface can be identified and then fitted by extracting its boundary. It is, first extracts the points lying on edges and then links them to obtain the closed contour that defines each surface. Generally, these techniques are very fast, but they have the problem that cannot guarantee closed boundaries. In addition, some times there are not clear boundaries defining a surface (regions connected by means of smooth surfaces).

Region based approaches consist in grouping points into connected regions according to some similarity criteria. The drawbacks of these techniques are to define the similarity criteria to be used and to define the position and number of initial seeds to start with the clustering process—region growing process (e.g., [4], [5]).

Finally, hybrid techniques are developed seized on the advantages of the previous approaches (e.g., [6], [7]). They combine the efficiency of edge based techniques with the robustness of surface based techniques. The information provided by an edge detection stage is then used to estimate the amount and positions of seeds, starting point of a clustering stage.

This paper presents a new hybrid segmentation technique that consists of a fast edge extraction technique followed by a region growing strategy (surface based segmentation). The obtained result is improved by a post-

The author has been supported by *The Ramón y Cajal Program*.

Proceedings of the 2nd Intl Symp 3D Data Processing, Visualization, and Transmission (3DPVT 2004)

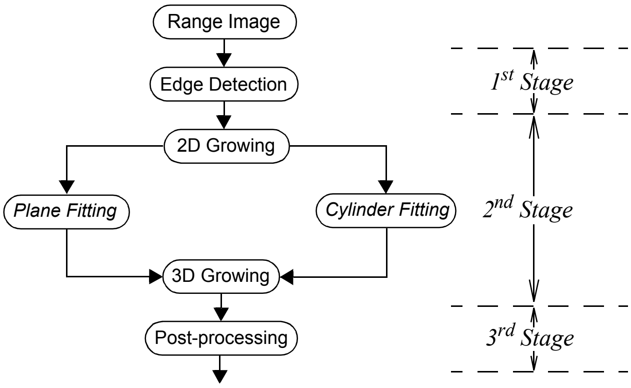

Figure 1. Flow charts of the proposed segmentation technique.

processing stage connecting neighbour regions that define the same surface. The proposed technique is described in section 2. Section 3 presents experimental results and finally, conclusions and further improvements are given in section 4.

2. Hybrid Segmentation Technique

The proposed algorithm consists of three stages. In the first stage, the edges contained in the given range image are extracted by means of a scan line approximation algorithm. Taking in mind that region growing segmentation algorithms demand high CPU time, in the second stage a two steps region growing algorithm has been implemented. Firstly a 2D growing is applied—considering only the seed positions a dilation process is applied in a 2D space until the region being grown reach a boundary. Then the points defining each region are fitted to a plane and a cylinder. Afterward, one of them is selected to represent the points contained into that region (the one that best fit these region's points). This surface is next used by a 3D growing stage. Finally, in the third stage, oversegmentation surfaces are joined improving the quality of the obtained results. Fig. 1 illustrates the different stages of the algorithm; they are further described below.

2.1. Edge detection

Edge based representations are compact ways to describe the geometry of the objects present in a given range image. Edges are defined by those points where a surface or orientation discontinuity appears. This section summarizes the implemented edge based segmentation technique. A complete description of this technique can be found in [2].

Several approaches have been proposed in the literature in order to extract the edges from a given range image (see details in [3]). In the current work a scan-line processing algorithm has been implemented. At this stage, every row and column (hereinafter called scan lines) is approximated by a set of oriented quadratic functions. Points in a scan-line are studied as in a 2D space, where x represents their column (or row) position and y represents their depth. In the current implementation, differently than previous works, only rows and columns but not diagonals are considered as scan-lines. The algorithm consists of two steps. First, the jump edge points—discontinuity in the surface—are detected by using a threshold adapted according to the resolution of the scanned points; these points are used to cut the original scan-line into a set of sections (set of consecutive points) and to define the starting and ending points of each one of them. Second, a recursive algorithm approximates each section (set of points) by quadratic functions oriented along edge y:

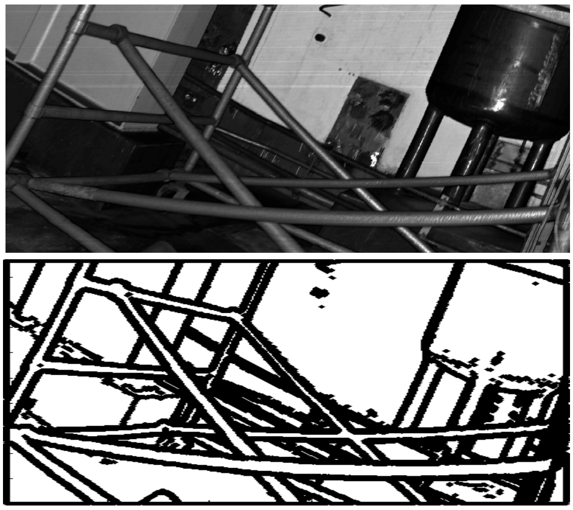

Figure 2. (*top*) Original range image represented by its intensity image. (*bottom*) Segmentation result.

$$y = ax^2 + bx + c \quad (1)$$

quadratic functions have been selected due to they allow to generate a more generic edge based representation than if only straight line segments were considered; moreover for range images acquired from industrial environments, second-order surfaces are numerous and quadratic approximations of the edges allow to reduce the number of points in that representation.

The approximation algorithm is applied to every section separately. It has been implemented by means of the classical recursive splitting algorithm [8]. Thus the parameters of equation (1) are obtained analytically by using three points (starting with the first, middle and last points of the considered section). The approximation error between the obtained quadratic function and every point is computed. If this error is greater than a given threshold ξ, the set of points is split into two set of points at that position where the biggest error appears. Next, that

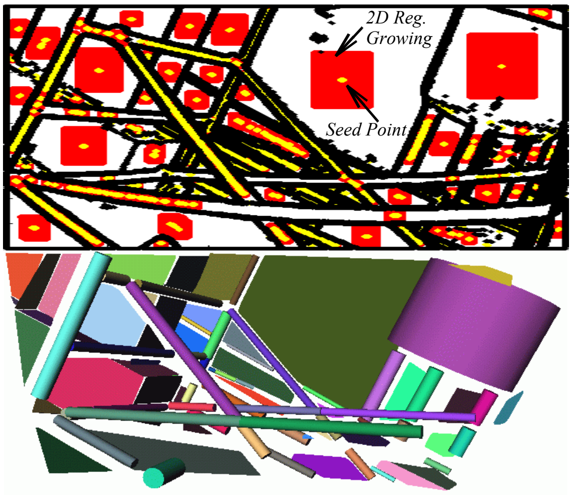

Figure 3. (*top*) Illustration of the seed points and 2D region growing step. (*bottom*) Segmentation result after the 3D growing process.

position is considered as last point, the middle point is computed and the parameters of (1) are computed again. This splitting algorithm is applied recursively while the approximation error is greater than ξ.

The result of this recursive algorithm is a set of quadratic curves approximating the considered scan line's section. Once a section is approximated, the recursive algorithm is carried out over the next section of the given scan-line. From each quadratic curve, the first and last points—points used to compute the parameters of equation (1)—are selected and their positions in a two dimensional binary map ***R****(r,c)* are labelled. Labelled points in that binary map define the edge points (Fig. 2 (*bottom*)).

Once a given scan-line has been approximated, the algorithm starts again over the next scan-line, thus the entire scan lines—rows and columns—are processed and the edge based representation is computed. Fig. 2(*bottom*) shows the edges extracted after segmenting the range image corresponding to the image showed in Fig. 2(*top*).

2.2. 2D and 3D region growing

At this section a two steps region growing algorithm is implemented. First, seeds points are placed along the edge representation. The position of these points is defined by the intersection between middle points obtained from rows and columns processing (points placed between two edge points when a row or column is considered). The first step consists in dilating those seed points (2D region growing by using as structuring element a *3x3* square element), represented in the array ***R***, until some edge point is reached. In the second step a 3D growing is applied. It consists in approximating the points defining each region by a plane and a cylinder. A plane is

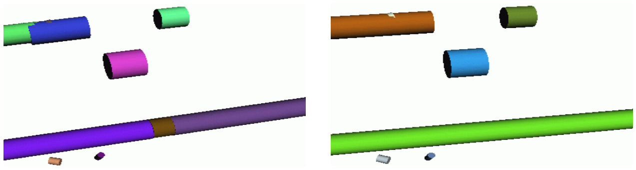

Figure 4. (*left*) Enlargement of the result obtained after the 3D growing stage. (*bottom*) Result obtained after the post-processing stage.

defined by $N \cdot X = d$ where $(\cdot)$ is the dot product, N is the normal vector and d represents the distance to the reference frame. A cylinder is defined by (P_1, P_2, r), where (P_1, P_2) represents the points that define the axis position and r represent its radius. Next, the surface that best fit that cloud of points is selected to represent that region and the 3D region growing process starts.

This second region growing process is carried out over each surface, but now considering the distance between the point which will be included or not into the considered region and the fitted surface—approximation error ∂. In case that this error is below than a given value, the point is included into the surface, next its neighbours points (which have not been already tested) are considered by the region growing process. On the contrary, if ∂ is higher than the given threshold, the point is not included into the region. The 3D growing process ended after all the points in a region have been considered by the growing process (no new points are added). The points contained into each region are refitted again in order to update the surface parameters. Then, the 3D growing stage starts again, but now considering these new surface parameters. At the end of this second iteration, the surface parameters are updated again and the resulting fitted surface is considered the final result. Fig. 3(*top*) shows an illustration of the 2D region growing step, while Fig. 3(*bottom*) presents the segmentation result obtained after the 3D region growing step.

2.3. Post-processing stage

As it is showed in Fig. 4(*left*), sometimes the algorithm generates an oversegmentation, it means that a single surface is represented by a set of regions. In that illustration, consecutive cylinders approximating the same pipeline are showed. They were obtained due to more than one seed have been placed in that region. Therefore, a post-processing stage has been implemented in order to merge neighbour regions belonging to the same surface.

First, neighbour regions are detected. It is computed by representing the points in the 2D array ***R*** and by checking the labels of the region boundary's points. Next, only neighbours regions defined by the same kind of surface (both of them are planes or both cylinders) will be considered as candidate to be merged. Two different cases have

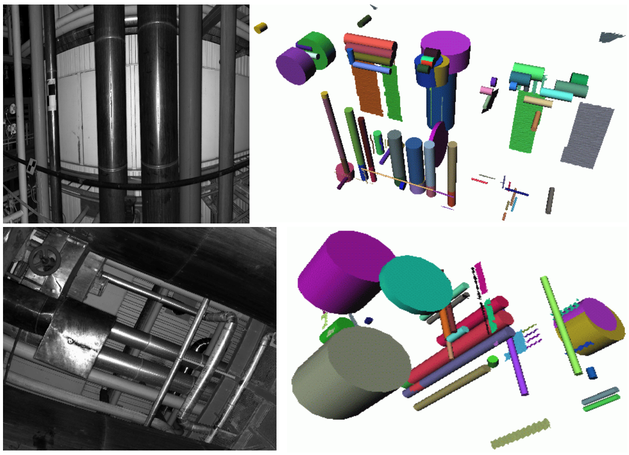

Figure 5. (*top-left*) Range image defined by *624x688* points. (*top-right*) Segmentation result. (*bottom-left*) Range image defined by *638x318* points. (*bottom-right*) Segmentation result.

to be considered, one if both surfaces are planes and second if both are cylinders.

In the first case, two planar neighbour regions are merged together if the following two criteria are satisfied: *i)* the angles between their normal vectors $\{N_i, N_j\}$ is below than ten degrees and *ii)* $|d_i - d_j| < 0{,}1(d_i + d_j)/2$. In the second case, two cylindrical neighbour regions are merged together if the following three criteria are satisfied: *i)* similar orientation (ten degrees between their axes' orientation), *ii)* $|r_i - r_j| < 0{,}1(r_i + r_j)/2$ and *iii)* the distances between the points defining the axis of one of them and the line containing the axis of the other one is below to some given threshold, this comparisons is also considered in the other sense (in the current implementation $(r_i + r_j)/2$ has been defined as a threshold). All the aforementioned threshold values have been obtained experimentally. When two surfaces (planes or cylinders) satisfy their corresponding criteria the points defining them are joined and refitted updating the surface parameters. Fig. 4(*right*) shows the result after the post-processing stage is carried out over the surfaces showed in Fig. 4(*left*).

3. Experimental Results

The proposed technique has been tested with several real range images from industrial facilities (e.g. petrochemical, refinery, nuclear power station). The CPU time to compute the different stages have been measured on a Pentium III, 1GHz processor. Fig. 3(*bottom*) shows the segmentation result obtained after processing the range image presented in Fig. 2. This representation has been generated after 146.6 sec. (11.44 sec. for the edge extraction stage and 135.16 sec. for the region growing and post-processing stages). Fig. 5 shows two illustrations of range image segmentation. The intensity images, from the corresponding range images, are shown at the left; they are defined by *624x688* points (*top*) and *638x318* points (*bottom*) respectively. The CPU times to generate the representations presented at the right were 136.09 sec. and 94.23 sec. for the top and bottom respectively. Notice that the segmentation results are presented from different point of views.

4. Conclusions and Further Improvements

A new approach for segmenting range images of industrial environments has been presented. It merges the efficiency of an edge detection technique with the robustness of the region growing. In order to speed up the process the region growing stage is carried out firstly in a 2D space and then in the 3D space.

At the present only planes and cylinders were considered, a further work will be to include more geometrical primitives. For example, sections of toroids are now being considered to represent the bend of a pipeline. Cone and general quadratic expression will also be considered. An extension to the post-processing stage will be recovering occluded surfaces [9].

The author wants to thank the Vision System Department of Z+F UK Ltd. for all the support received during the development of this work.

5. References

[1] J. Hancock et al., "Active laser radar for high-performance measurements", *IEEE Int. Conference on Robotics and Automation*, Leuven, Belgium, May 1998.

[2] A. Sappa and M. Devy, "Fast range image segmentation by an edge detection strategy", *IEEE 3rd. Int. Conf. on 3-D Digital Imaging and Modeling*, Quebec, City, Canada, May-June 2001.

[3] X. Jiang and H. Bunke, Edge Detection in Range Images Based on Scan Line Approximation, *Computer Vision and Image Understanding*, Vol. 73, No. 2, February 1999.

[4] H. Frigui and R. Krishnapuram, A Robust Competitive Clustering Algorithm with Applications in Computer Vision, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21, no. 5, pp. 450-465, May 1999.

[5] M. A. García and L. Basañez, Fast Extraction of Surface Primitives from Range Images, *13th IAPR Int. Conf. on Pattern Recognition*, Vienna, Austria, August 1996.

[6] O. Pereira Bellon, A. Direne and L. Silva, Edge Detection to Guide Range Image Segmentation by Clustering Techniques, *IEEE Int. Conf. on Image Processing*, Kobe, Japan, October 1999.

[7] K. Koster and M. Spann, MIR: An Approach to Robust Clustering-Application to Range Image Segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 5, pp. 430-444, May 2000.

[8] D. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1972.

[9] F. Stlup, F. Dell'Acqua and R. Fisher, "Reconstruction of surfaces behind occlusions in range images", *IEEE 3rd. Int. Conf. on 3-D Digital Imaging and Modeling*, Quebec, City, Canada, May-June 2001.