SEGMENTATION OF POINT CLOUDS USING SMOOTHNESS CONSTRAINT

T. Rabbani[a], F. A. van den Heuvel[b], G. Vosselman[c]

[a*] Section of Optical and Laser Remote Sensing, TU Delft, The Netherlands - t.rabbani@lr.tudelft.nl

[b] CycloMedia Technology B.V., 4180 BB Waardenburg, The Netherlands - FvandenHeuvel@cyclomedia.nl

[c] Department of Earth Observation Science, ITC ,Hengelosestraat 99, Enschede, the Netherlands - vosselman@itc.nl

Working Group V/I,III

KEY WORDS: Laser scanning, Point cloud segmentation, Reverse Engineering, Industrial Reconstruction

ABSTRACT

For automatic processing of point clouds their segmentation is one of the most important processes. The methods based on curvature and other higher level derivatives often lead to over segmentation, which later needs a lot of manual editing. We present a method for segmentation of point clouds using *smoothness constraint*, which finds smoothly connected areas in point clouds. It uses only local surface normals and point connectivity which can be enforced using either k-nearest or fixed distance neighbours. The presented method requires a small number of intuitive parameters, which provide a tradeoff between under- and over-segmentation. The application of the presented algorithm on industrial point clouds shows its effectiveness compared to curvature based approaches.

1 INTRODUCTION

1.1 Problem statement

Segmentation is the process of labeling each measurement in a point cloud, so that the points belonging to the same surface or region are given the same label. For the problem of industrial reconstruction the point cloud is usually acquired using a laser scanner. Unlike structured light based instruments which provide $2\frac{1}{2}$D data, most of the laser scanners provide an unstructured point cloud. Even in the case of $2\frac{1}{2}$D range images, once two or more such images have been registered, the resulting data loses its $2\frac{1}{2}$D character and has to be represented as an unstructured 3D point cloud. Based on these observations the presented method would work only with unstructured point clouds, and other data representations can be easily converted to this format if required.

1.2 Previous work

Different approaches for segmentation suggested in literature differ mainly in the method or criterion used to measure the similarity between a given set of points and hence for making the grouping decisions. Once such a similarity measure has been defined, segments can be obtained by grouping together the points whose similarity measure is within given thresholds and which are spatially connected. Most of the segmentation methods presented in the literature are for depth-maps as due to their $2\frac{1}{2}$D nature operations from traditional image processing can be directly applied.

Three main varieties of range segmentation algorithms are discussed below:

1.2.1 Edge-based segmentation Edge based segmentation algorithms have two main stages: edge detection which outlines the borders of different regions, followed by the grouping of the points inside the boundaries giving the final segments. Edges in a given depth map are defined by the points where changes in the local surface properties exceed a given threshold. The local surface properties mostly used are surface normals, gradients, principal curvatures, or higher order derivatives. Some of the typical variations on the edge-based segmentation techniques are reported by Bhanu et al. (1986); Sappa and Devy (2001); Wani and Arabnia (2003).

1.2.2 Surface-based segmentation The surface based segmentation methods use local surface properties as a similarity measure and merge together the points which are spatially close and have similar surface properties. These methods are relatively less sensitive to the noise in the data, and usually perform better when compared to edge based methods.

For surface based segmentation methods two approaches are possible: bottom-up and top-down. Bottom up approaches start from some seed-pixels and grow the segments from there based on the given similarity criterion. The selection of the seed points is important because the final segmentation results are dependent on it. Top-down methods start by assigning all the pixels to one group and fitting a single surface to it. Then as long as a chosen figure of merit for fitting is higher than a threshold they keep on subdividing the region Parvin and Medioni (1986); Xiang and Wang (2004). Most of the reported methods for range segmentation use a bottom-up strategy.

1.2.3 Scanline-based segmentation The third category of range segmentation methods is based on scan-line grouping. In the case of range images each row can be considered a scan-line, which can be treated independently of other scan-lines in the first stage. A scan-line grouping based segmentation method is presented by Jiang et al.

(1996) for the extraction of planar segments from the range image. It uses the fact that a scan line on any 3D plane makes a 3D line. It detects the line segments in the first stage, followed by the grouping of the adjacent lines with similar properties to form planar segments. Some typical variations on this method are presented by Natonek (1998) and Khalifa et al. (2003). Sithole and Vosselman (2003) have used profiles in different directions for the segmentation of air-borne laser scanner data. These profiles are generated by collecting points within a cylindrical volume around a given direction.

A comparison of methods for finding planar segments in range images was done by Hoover et al. (1996). Based on his comparison framework, two methods for segmentation of range images into curved regions were compared by Powell et al. (1998). The first segmentation method of Besl and Jain (1988) (hereafter named BJ method) has two stages. The first stage of coarse segmentation is based on estimating the mean and Gaussian curvature for each point and using their signs for classification into 8 different surface types. This rough segmentation is refined by the second step of region growing, which is based on fitting bivariate polynomial surfaces. The comparison found the BJ method to result in severe over-segmentation even on very simple scenes with low noise. The major reason for this failure was the error in the estimation of principal curvatures from the noisy range data. The BJ segmentation method has 38 different parameters, 10 of which were iteratively optimized to get the best possible results. The speed of this method was very slow, and even on range images which do not have a high cost for searching the neighborhood points, it took more than 6 hours.

The second segmentation method used for the comparison was by Jiang et al. (1996) (hereafter JBM method). This method also consists of two stages. In the first stage the scan lines of the range image are segmented into a set of curves by using a splitting method. In the second stage these edges are grouped together to make surfaces. This method has 10 parameters and the comparison found it to perform much better than the BJ method, both in terms of time (30 seconds compared to 6 hours of the BJ method) and the quality of results.

Looking at the comparison, the JBM method would be a good choice for the segmentation of industrial scenes, but there are some serious limitations. First of all, the method is based on the grouping of scan lines which do not exist for unstructured point clouds. For airborne laser scanner data similar scan lines created by collecting and joining points in a tubular volume have been used by Sithole and Vosselman (2003) for segmentation, but there the data is $2\frac{1}{2}$D . Extensions of this idea for 3D point clouds would require choosing a few preferred directions for scan-lines, making the results of segmentation orientation dependent.

1.3 Problems with existing methods

We observed the following problems with existing segmentation techniques as applied to our problem of processing industrial point clouds:

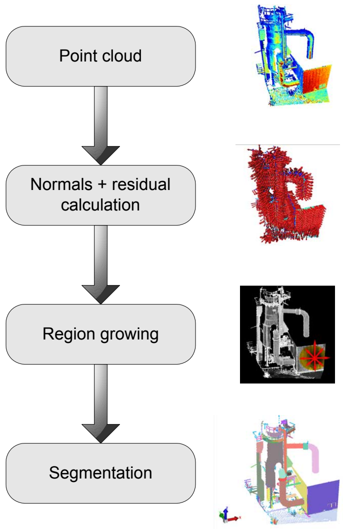


Figure 1: Flowchart of the segmentation algorithm.

1. Many approaches are tailored only for planar surfaces, which are too limiting for industrial scenes.

2. Although principal curvature based approaches can handle curved objects, the unreliable estimation of the curvature from noisy point clouds leads to high rates of over-segmentation. Furthermore, the objects like torus and sphere are always over-segmented because they are not one of the 8 different surface classes identifiable based on the signs of the principal curvatures. The sensitivity of curvature estimates from range data has been analyzed in Trucco and Fisher (1995), where it is suggested that at least for planar segmentation of range data principal curvatures should not be used.

3. Many segmentation methods have a large number of parameters, whose meaning and effects on final segmentation are not always clear. Most of the comparisons used separate iterative optimization methods to find the best set of parameters.

4. Most of the methods are tailored for application to $2\frac{1}{2}$D range images. Sometimes their extension to 3D unstructured point clouds is quite simple like replacement of 8-neighbors with k-nearest neighbors. But in other cases, like defining scan lines for 3D point clouds, there is no straight forward extension, and most approaches introduce new limitations.

5. There are some model-based approaches to segmentation which segment and recognize surface types at the same time Marshall et al. (2001). These approaches do not fit in our pipeline of separate segmentation and object recognition through Hough transform and thus cannot be applied to our problem.

1.4 Objectives and motivation

Noting these weaknesses we decided to develop a simple segmentation strategy that follows the following guidelines:

1. We will assume a raw unstructured 3D point cloud as the input to the algorithm. Although the assumptions about structure of data (range image, TIN etc) can make the job of neighbor search faster, they at the same time make the algorithm less general purpose.

2. We will use only surface normals as they can be reliably estimated even in the presence of noise (provided the neighborhood is sufficiently big to average out effects of noise).

3. The algorithm should allow the user to choose between the degree of over and under-segmentation by changing a few parameters. In our modeling pipeline the segmentation is followed by the stage of object recognition, which processes each segment separately. That stage can detect multiple objects in one segment (under segmentation), but if an object is split into multiple segments (over-segmentation), its detection and correction would be more difficult.

4. The algorithm should have a low time and space complexity. Furthermore, we aim to use a minimum number of parameters having a physically intuitive meaning.

2 SEGMENTATION ALGORITHM

As stated earlier the basic purpose of the presented segmentation algorithm is to subdivide the input point cloud into meaningful subsets, while avoiding both under and over-segmentation with a preference for under-segmentation.

The segmentation method has two steps, Normal estimation and region growing. The details of these steps are given in algorithm 1 and further explained below. See also Figure 1.

2.1 Normal estimation

The normal for each point is estimated by fitting a plane to some neighboring points (Figure 2(c)). This neighborhood can be specified in two different methods.

K nearest neighbors (KNN) In this method for a given point we select the k points from the point cloud having the minimum distance. The distance metric used can be Euclidean, Manhattan, or any other distance metric obeying the triangle inequality.

As the number of points k is fixed, the method adapts the area of interest (AOI) according to the point density. Assuming that the point density is an indicator of the measurement noise (which is usually the case as for a given laser scanner the density falls down inversely with the distance and the angle of incidence), this results in overall better estimation of the normals as a bigger AOI is used in the areas of lower point density (Figure 2(a) and 2(b)). Moreover, this method always uses the given number of points and avoids degenerate cases (e.g. a point having no neighbors).

Algorithm 1 Segment a given point cloud using smoothness constraint

Inputs: Point cloud = $\{\mathbf{P}\}$, point normals $\{\mathbf{N}\}$, residuals $\{\mathbf{r}\}$, neighbor finding function $\mathbf{\Omega}(.)$, residual threshold r_{th}, angle threshold θ_{th}
Initialize Region List $\{\mathbf{R}\} \leftarrow \Phi$, Available points list $\{\mathbf{A}\} \leftarrow \{1 \cdots P_{count}\}$
while $\{\mathbf{A}\}$ is not empty **do**
  Current region $\{\mathbf{R}_c\} \leftarrow \emptyset$, Current seeds$\{\mathbf{S}_c\} \leftarrow \Phi$
  Point with minimum residual in $\{\mathbf{A}\} \rightarrow P_{min}$
  $P_{min} \overset{\text{insert}}{\rightarrow} \{\mathbf{S}_c\}\&\{\mathbf{R}_c\}$
  $P_{min} \overset{\text{remove}}{\rightarrow} \{\mathbf{A}\}$
  for $i = 0$ to $\mathbf{size}(\{\mathbf{S}_c\})$ **do**
    Find nearest neighbors of current seed point $\{\mathbf{B}_c\} \leftarrow \mathbf{\Omega}(\mathbf{S_c\{i\}})$
    for $j = 0$ to $\mathbf{size}(\{\mathbf{B}_c\})$ **do**
      Current neighbor point $P_j \leftarrow \mathbf{B}_c\{j\}$
      if $\{\mathbf{A}\}$ contains P_j and $\cos^{-1}(|\langle \mathbf{N}\{\mathbf{S}_c\{i\}\}, \mathbf{N}\{P_j\}\rangle|) < \theta_{th}$ **then**
        $P_j \overset{insert}{\rightarrow} \{\mathbf{R}_c\}$
        $P_j \overset{remove}{\rightarrow} \{\mathbf{A}\}$
        if $\mathbf{r}\{P_j\} < r_{th}$ **then**
          $P_j \overset{insert}{\rightarrow} \{\mathbf{S}_c\}$
        end if
      end if
    end for
  end for
  Add current region to global segment list $\{\mathbf{R}_c\} \overset{insert}{\rightarrow} \{\mathbf{R}\}$
end while
Sort $\{\mathbf{R}_c\}$ according to the size of the region.
Return $\{\mathbf{R}_c\}$

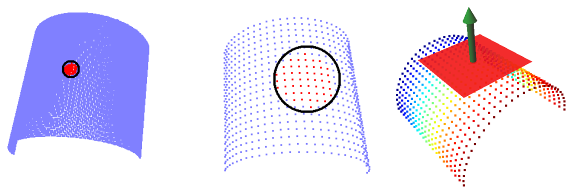
Search for KNN can be optimized using different space partitioning strategies like k-d trees Arya et al. (1998); Goodman and O'Rourke (1997).

Fixed distance neighbors (FDN) This method uses a given fixed AOI, and for each query point, selects all the points within this area. The distance metric used is usually Euclidean but can be changed similar to KNN. For FDN search the number of points changes according to the density of the point cloud. As the number of points is directly proportional to the density of the points in the neighborhood, this method does not have the adaptive behavior of KNN.

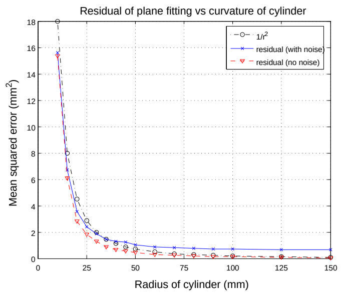
Compared to KNN, here the number of points is less in the areas of low density (high noise) and as a result the estimation of the normals is on the whole contains more noise.

This method is more suitable if the density of the points does not change a lot throughout the data. Similar to KNN there are optimized methods for doing FDN searching Goodman and O'Rourke (1997); Willard (1985)

One of the above described methods for neighborhood search are also used during the stage of region growing (Section 2.2).



(a) (b) (c)



(d)

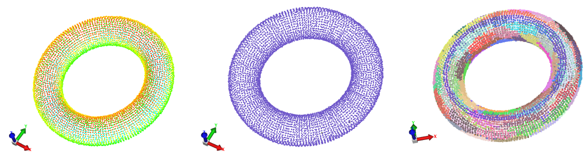
Figure 2: (a-b)Adaptive change in selection area for k-neighbors for different point densities (a) hight density, 50 KNN (b) low density, 50 KNN (c) Normal estimation by fitting a plane to the points in the neighborhood (d)Residual of the plane fitting gives an approximation to the local surface curvature

2.1.1 Plane fitting To fit any surface to a set of given points, in a least squares sense, we want to find that set of parameters that minimizes the sum of squares of the orthogonal distances of the points from the estimated surface. In general this is a nonlinear least squares problem, but as shown below, in case of planes this can be reduced to an eigenvalue problem.

The plane can be parameterized with its normal $\mathbf{n} = \begin{pmatrix} n_x & n_y & n_z \end{pmatrix}$ and its distance from the origin ρ. This is also called Hesse normal form of the plane. The distance of any given point $\mathbf{p} = \begin{pmatrix} p_x & p_y & p_z \end{pmatrix}$ from the plane is given by $\mathbf{n} \cdot \mathbf{p} - \rho$ provided $\mathbf{n} \cdot \mathbf{n} = 1$. This is a constrained problem and can be solved using Lagrange multipliers. The solution results in an eigen-value problem for more discussion see Hoppe et al. (1992).

2.1.2 Residual as approximate curvature The residual in the plane fitting can arise either from noise or from nonconformity of the neighborhood of a point to the planar model. The second case hints that the residual can be used to find areas of high curvature. Of course we do not get the principal curvatures and their direction from this approximation, but still the edges and the areas of high surface normal variation can be detected based on high residual values of plane fitting.

To check the relationship between curvature and residual of plane fitting we generated data consisting of cylinders of different radii. The normals for these cylinders were estimated by fitting planes to 40 k-nearest neighbors, and then plotted against $\frac{1}{r^2}$ (Figure 2(d)). There we see that for the case of no noise the residuals are quite similar to $\frac{1}{r^2}$ except a difference of scale. In the presence of noise the trend remains the same but in addition to a scale factor there is also a shift related to the amount of noise. This supports our idea of using residuals of plane fitting as indicator of areas of high curvature. The regions of high curvature are detected by introduction of r_{th} in Algorithm 1.



(a) (b) (c)

Figure 3: Comparison of segmentation for a toroidal surface (a) point cloud (b) segmentation using presented approach (c)curvature based segmentation

In Figure 4(a), 4(e), 4(g) we show the residual of plane fitting as color. As expected the areas on edges and points of high curvature have higher residuals.

2.2 Region growing

The next step in the segmentation process is region growing. This stage uses the point normals and their residuals, in accordance with user specified parameters to group points belonging to the smooth surfaces. This grouping tries to avoid over-segmentation at the cost of under-segmentation.

This stage is based on the enforcement of these two constraints.

Local connectivity The points in a segment should be locally connected. This constraint would be enforced by using only the neighboring points (through KNN or FDN) during region growing.

Surface smoothness The points in a segment should locally make a smooth surface, whose normals do not vary “too much” from each other. This constraint would be enforced by having a threshold (θ_{th}) on the angles between the current seed point and the points added to the region. Additionally, a threshold on residual values r_{th} makes sure that smooth areas are broken on the edges.

The process of region growing proceeds in the following steps.

1. Specify a residual threshold r_{th}. Alternatively, calculate this threshold automatically using a specified percentile of the sorted residuals (95+% can be a representative number).

2. Define a smoothness threshold in terms of the angle between the normals of the current seed and its neighbors. If the smoothness angle threshold is expressed in radians it can be enforced through dot product as

follows $\|\mathbf{n_p} \cdot \mathbf{n_s}\| > cos(\theta_{th})$. As the direction of normal vector has a 180^o ambiguity we have to take the absolute value of the dot product.

3. If all the points have been already segmented go to step 7. Otherwise select the point with the minimum residual as the current seed.
4. Select the neighboring points of the current seed. Use KNN or FDN with the specified parameters for this purpose. The points that satisfy condition 2 add them to current region. The points whose residuals are less than r_{th} add them to the list of potential seed points.
5. If the potential seed point list is not empty, set the current seed to the next available seed, and go to step 4.
6. Add the current region to the segmentation and go to step 3.
7. Return the segmentation result

Region growing tries to group points belonging to smooth surface patches together. Although, we want to avoid over-segmentation but still we do not want the whole point cloud coming out as one segment. The inclusion of residual threshold (r_{th}) makes sure, that we can strike a balance between the above mentioned extremes. As $r_{th} \to 0$ we go towards more segments with the extreme case being each point belonging to one segment. Similarly as $r_{th} \to \infty$ we have less segments and the extreme case of the whole point belonging to one segment.

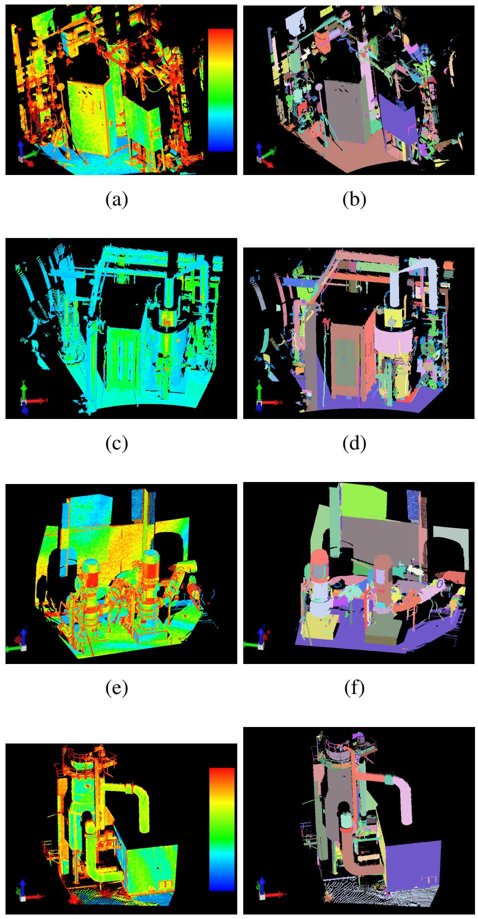
We can differentiate between the following cases which may lead to the start of a new segment during the process of region growing.

Step edge A step edge is defined by two planes which have the same orientation but different offset from the origin. The segmentation algorithm leads to their separation provided the offset between planes is greater than the AOI for the neighborhood search. For KNN this depends both on the value k and point density, while for FDN it is equal to the fixed distance specified by the user.

Intersection edge A intersection edge is defined by the intersection of two surfaces, whose surface normal at the intersection line make an angle greater than the given threshold ($\cos^{-1} \mathbf{n}_1 \cdot \mathbf{n}_2 > \theta_{curv}$). An example of such an edge would be the edge coming from the intersection of the two planar sides of a box.

The surfaces on both side of the edge would be segmented because of the smoothness constraint (θ_{th}). Additionally the points on the edge would be marked unsuitable for inclusion in the next generation seeds, as they will have residuals greater than r_{th}.

The effectiveness of the method to detect smoothly varying surface patches is shown best in Figure 3. While the method of curvature based segmentation leads to high over-segmentation (Figure 3(c)), our method divides the data in only one segment (Figure 3(b).



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4: Results of segmentation (a) residuals of data set 1 (b) segmentation 1 (c) data set 2 (d) segmentation 2 (e) residuals of data set 3 (f) segmentation 3 (g) residuals 4 (h)segmentation 4

3 RESULTS

The presented algorithm was applied to four sets of point clouds acquired from four industrial sites. The results are shown in Figure 4. For these results the θ_{th} was set to 15^o and 30 nearest neighbors were used ($k = 30$); r_{th} was automatically calculated by the 98^{th} percentile of the plane fitting residuals. In the results we see both goals of grouping smooth areas and avoiding over-segmentation have been successfully achieved. There are areas where large under-segmentation occurred, but it can be explained based on the values the parameters θ_{th} and r_{th}.

For example in Figure 4(d) a whole U-section of pipe is segmented as one region, because it is smoothly connected. Similarly in Figure 4(h) the L-junctions of pipes have been grouped as one region rather than being split into two pipes and one curve. As for the presented results the residual threshold r_{th} was calculated using the percentile method, it leads to data dependent values. In Figure 5 we took two segments from the results of Figure 4(f) and segmented them again. As now the data is more limited the threshold r_{th} is lower and more strict. This leads to segmentation having more regions along with some over-segmentation.

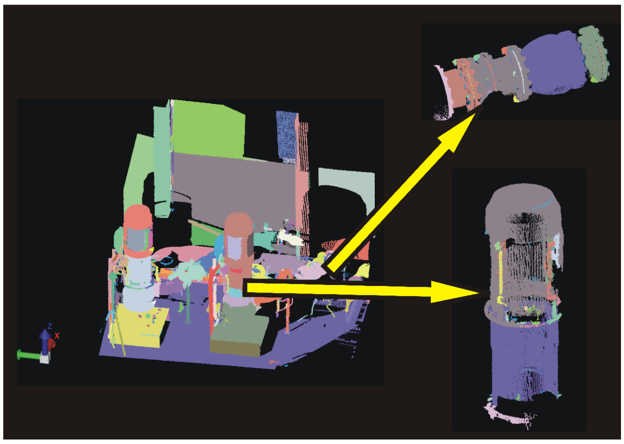


Figure 5: Effects of changing r_{th} on the final segmentation. Two segments are re-segmented but with lower r_{th} resulting in more segments

Thus by choosing a proper value for r_{th} the required balance between under and over-segmentation can be achieved.

4 CONCLUSIONS

A segmentation algorithm for dividing a given unstructured 3D point cloud into a set of smooth surface patches has been presented. The algorithm uses only surface normals as a measure of local geometry, which are estimated by fitting a plane to the neighborhood of the point. As fitting of higher order surfaces to noisy point clouds is quite error prone, we approximate the local curvature by the residual of plane fitting. The method has two parameters (θ_{th} and r_{th}), which have intuitively clear meaning. Both k nearest neighbor and fixed distance neighbor variations of the algorithm are possible. The results on point clouds acquired from industrial sites were presented that show the effectiveness of the method and its ability to choose between under-segmentation and over-segmentation.

References

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R. and Wu, A. Y., 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of ACM 45(6), pp. 891–923.

Besl, P. J. and Jain, R. C., 1988. Segmentation through variable-order surface fitting. PAMI 10(2), pp. 167–192.

Bhanu, B., Lee, S., Ho, C. C. and Henderson, T., 1986. Range data processing: representation of surfaces by edges. In: Proceedings of the Eighth International Conference on Pattern Recognition, pp. 236–238.

Goodman, J. E. and O'Rourke, J., 1997. Handbook of discrete and computational geometry. CRC Press, Inc., Boca Raton, FL, USA.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K. K., Eggert, D. W., Fitzgibbon, A. W. and Fisher, R. B., 1996. An experimental comparison of range image segmentation algorithms. PAMI 18(7), pp. 673–689.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., 1992. Surface reconstruction from unorganized points. In: Proceedings of ACM SIGGRAPH '92, ACM Press, New York, NY, USA, pp. 71–78.

Jiang, X. Y., Bunke, H. and Meier, U., 1996. Fast range image segmentation using high-level segmentation primitives. In: Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96), IEEE Computer Society, Washington, DC, USA, p. 83.

Khalifa, I., Moussa, M. and Kamel, M., 2003. Range image segmentation using local approximation of scan lines with application to cad model acquisition. Machine Vision Applications 13(5-6), pp. 263–274.

Marshall, A. D., Lukács, G. and Martin, R. R., 2001. Robust segmentation of primitives from range data in the presence of geometric degeneracy. PAMI 23(3), pp. 304–314.

Natonek, E., 1998. Fast range image segmentation for servicing robots. In: The IEEE International Conference on Robotics and. Automation, pp. 406–411.

Parvin, B. and Medioni, G., 1986. Segmentation of range images into planar surfaces by split and merge. In: CVPR '86, pp. 415–417.

Powell, M. W., Bowyer, K. W., Jiang, X. and Bunke, H., 1998. Comparing curved-surface range image segmenters. In: Proceedings of the Sixth ICCV, IEEE Computer Society, Washington, DC, USA, p. 286.

Sappa, A. D. and Devy, M., 2001. Fast range image segmentation by an edge detection strategy. In: Third International Conference on 3-D Digital Imaging and Modeling, pp. 292–299.

Sithole, G. and Vosselman, G., 2003. Automatic structure detection in a point-cloud of an urban landscape. In: 2nd Joint Workshop on Remote Sensing and Data Fusion over Urban Aereas (Urban 2003).

Trucco, E. and Fisher, R. B., 1995. Experiments in curvature-based segmentation of range data. PAMI 17(2), pp. 177–182.

Wani, M. A. and Arabnia, H. R., 2003. Parallel edge-region-based segmentation algorithm targeted at reconfigurable multiring network. Journal of Supercomputing 25(1), pp. 43–62.

Willard, D. E., 1985. New data structures for orthogonal range queries. SIAM Journal of Computing 14(1), pp. 232–253.

Xiang, R. and Wang, R., 2004. Range image segmentation based on split-merge clustering. In: 17th ICPR, pp. 614–617.