SEGMENTATION OF LIDAR DATA USING THE TENSOR VOTING FRAMEWORK

Hanns-F. Schuster
Institute of Photogrammetry, University of Bonn
Nussallee 15, D-53115 Bonn, Germany
Schuster@ipb.uni-bonn.de

KEY WORDS: LIDAR, Segmentation, Algorithm, Automation, Modelling, Point Cloud

ABSTRACT

We present an investigation on the use of Tensor Voting for categorizing LIDAR data into outliers, line elements (e.g. high-voltage power lines), surface patches (e.g. roofs) and volumetric elements (e.g. vegetation).
The Reconstruction of man-made objects is a main task of photogrammetry. With the increasing quality and availability of LIDAR sensors, range data is becoming more and more important. With LIDAR sensors it is possible to quickly aquire huge amounts of data. But in contrast to classical systems, where the measurement points are chosen by an operator, the data points do not explicitly correspond to meaningful points of the object, i.e. edges, corners, junctions. To extract these features it is necessary to segment the data into homogeneous regions wich can be processed afterwards.
Our approach consists of a two step segmentation. The first one uses the Tensor Voting algorithm. It encodes every data point as a particle which sends out a vector field. This can be used to categorize the pointness, edgeness and surfaceness of the data points. After the categorization of the given LIDAR data points also the regions between the data points are rated. Meaningful regions like edges and junctions, given by the inherent structure of the data, are extracted.
In a second step the so labeled points are merged due to a similarity constraint. This similarity constraint is based on a minimum description length principle, encoding and comparing different geometrical models.
The output of this segmentation consists of non overlapping geometric objects in three dimensional space.
The aproach is evaluated with some examples of Lidar data.

1 INTRODUCTION

With the increasing quality and availability and falling costs of LIDAR-data there is a growing need for automatic detection and reconstruction of the objects contained in the data. A human can easily read the content of a point cloud because our brain is highly trained in such context-based segmentation tasks, but for automatic reconstruction we need to have the location of meaningful features like corners, edges or junctions.
The Problem with LIDAR-data is, that the measured points do not have any context information and the grid in which they are measured is not oriented on these features. Normally the wanted features are only indirectly observable e.g. by segmenting two planes and intersecting them.
In this paper we show the extraction of features like curves, surfaces and junctions from a point cloud. therefore we present a two-step procedure that uses the tensor voting framework as a first step to categorize the input points into three types of appearance. In a second step we use a segmentation to merge the categorized points into curves and surfaces.
The tensor voting framework (Tang et al., 2000) can not only be used for handling 2D or 3D (Tang and Medioni, 1999) data but also to process motion fields (Nicolescu and Medioni, 2003) or stereo data (Lee and Medioni, 1998). In most cases the input data is of small scale (Tang and Medioni, 1998) in contrast to LIDAR-data and the output is only used for visualisation in pixel or voxel representation (G. Guy, 1997).

In section two we will have a look on the tensor voting framework. In section three we show how the output of the tensor voting can be segmented. The results of the approach are presented in section four. In section five a conclusion is presented followed by an outlook.

2 TENSOR VOTING

The goal of the tensor voting is to extract the structure inherently given in the point cloud.
The results of the tensor voting process are three coutinuous vector fields, represented by discrete grid points. The scalar part of these fields represent the likelihood of the location in space to be a point, part of a curve, a surface. The vector part represents the orientation of the occurence.
These three fields can be searched through to find maxima which represent the most likely location of a wanted feature.

2.1 TensorVoting in physical analogy

To explain the concept of Tensor Voting with an analogy to physics, we can compare the Tensorfield with a physical field of force, e.g. a magnetic field. We can imagine that the object which is represented in the point cloud has a magnetic field. It propagates its field into the space around the object.

If we put iron particles into this field, these are affected by the field so that they act as little magnetic dipoles which align their field along the field lines of the object. If we add enough particles we can infer the form of the field of the object and thus the form of the object by interpolating the little parts of the field send out by the particles.

In the case of the tensor voting we walk this path backwards: First we have the particles in space which are our

LIDAR-data points. We know that they lie besides the outliers on the surface of the wanted object. By assigning a little standard field to every particle, the points will propagate it into a certain neighborhood around them and influence the neighbored points. In this way they adapt the form of the field of the respresented object. Afterwards we can interpolate the field in the space between the particles and extract meaningful points like edges and corner points by searching for maxima.

2.2 Tensor encoding

The above mentioned field is a tensor field, that means each point in space has an associated tensor. In our case this tensor is a second order symmetrical tensor. If we formulate the tensor like in (1) it encodes a 3D-ellipsoid rotated in space. (2) is an equivalent writing for this. In fig. (1) we can see the geometrical meaning of (1). The normalized vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$ are the main axis of the ellipsoid. They build a local right-handed coordinate system.

$$T = \begin{pmatrix} \boldsymbol{e}_1, & \boldsymbol{e}_2, & \boldsymbol{e}_3 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} \boldsymbol{e}_1^T \\ \boldsymbol{e}_2^T \\ \boldsymbol{e}_3^T \end{pmatrix} \tag{1}$$

$$T = \lambda_1 \boldsymbol{e}_1 \boldsymbol{e}_1^T + \lambda_2 \boldsymbol{e}_2 \boldsymbol{e}_2^T + \lambda_3 \boldsymbol{e}_3 \boldsymbol{e}_3^T \tag{2}$$

This Ellipsoid has the dimensions λ_1, λ_2 and λ_3 in the main axis directions. We define that $\lambda_1 > \lambda_2 > \lambda_3$ i.e. we claim that the ellipsoid is always oriented in the direction of $\boldsymbol{e}_1$. With this definition we can rewrite (2) into (3) by saying that λ_3 is the basic part in all three directions where the differences of $\lambda_2 - \lambda_3$ and $\lambda_1 - \lambda_2$ are added in the directions $\boldsymbol{e}_2$,$\boldsymbol{e}_1$ and $\boldsymbol{e}_1$. The geometrical interpretation is shown in fig. (2). The Ellipsoid is decomposed into the λ_3-part, which constructs a 3D sphere, encodes the likelihood of this location to be a point also called point-ness. The $(\lambda_2 - \lambda_3)$-part which defines a 2D disk in the $\boldsymbol{e}_2$-$\boldsymbol{e}_3$-plane here called the surface-ness of the location and the $(\lambda_1 - \lambda_2)$-part which defines a one-dimensional stick-portion and which is the curve-ness.

$$\begin{aligned} T &= (\lambda_1 - \lambda_2)\boldsymbol{e}_1\boldsymbol{e}_1^T + (\lambda_2 - \lambda_3)(\boldsymbol{e}_1\boldsymbol{e}_1^T + \boldsymbol{e}_2\boldsymbol{e}_2^T) \\ &\quad + \lambda_3(\boldsymbol{e}_1\boldsymbol{e}_1^T + \boldsymbol{e}_2\boldsymbol{e}_2^T + \boldsymbol{e}_3\boldsymbol{e}_3^T) \end{aligned} \tag{3}$$

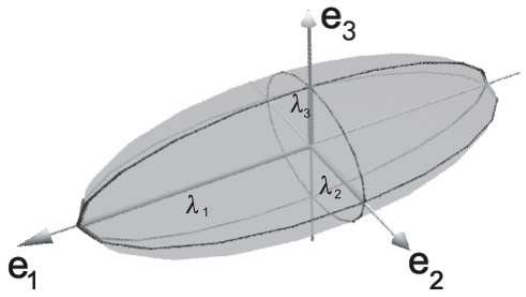

Figure 1: An ellipsoid with its local coordinatesystem and the dimenions $\lambda_1, \lambda_2, \lambda_3$

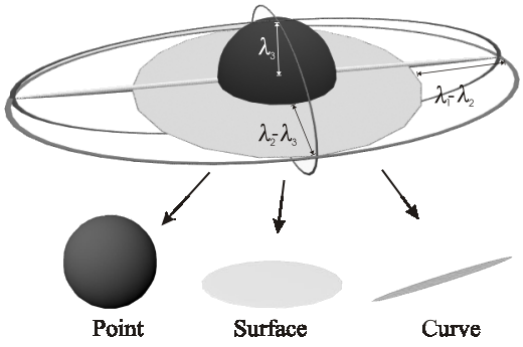

Figure 2: The decomposition of a Tensor

2.3 Voting as communication

Every initial location sends out a tensor field and propagates it into the space in a certain neighborhood. Every other location in this neighborhood is then influenced by this field. To calculate the total influence on a certain location we simply have to summarize the tensor fields of all neighbors in a given radius. therefore we have to look how the tensor field propagates in space.

As we have seen above, the tensor field represents three vectorfields with different meanings. Thus these vector fields behave differently while propagating, we have to handle each part by itself and assemble them again afterwards (4). The reason why we formulate the three fields in a single tensor is the comunication between these fields. This implicit communication is shown in fig. (3) where two votes of two different voting sites accumulate at the recieving site to a surface portion.

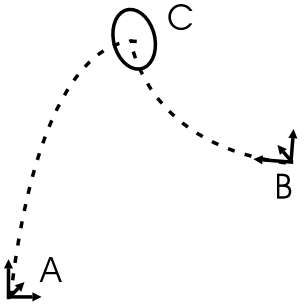

Figure 3: Communication between the curve field

$$T_1^A = T_0^A + \sum_i \left(T_{Ball}^{B_i} + T_{Plate}^{B_i} + T_{Stick}^{B_i} \right) \tag{4}$$

The design of the voting fields is derived by considering the analogy with the flow of force in particle physics (). The three voting fields are shown in fig. (4). The length and orientation of the sticks depicted in fig. (4) is the strength and orientation of the field sent out by the voting location (which is located in the center) and received at the site position relative to the voting location in a local coordinate system.

$$V(d, \rho) = e^{-\frac{ad^2 + b\rho^2}{\sigma^2}} \tag{5}$$

The curve voting field has its main direction along the e_1-axis and is rotational symmetric around this axis. The strength of the vote decays with increasing distance and curvature like which can be written as (5), where d is the distance, ρ is the curvature and σ is a scale factor that has to be chosen in context of the input data. The constants a and b have to be chosen to define the field. The resulting orientation of the voting field is chosen in a way that the total curvature along the path from the voting site to the recieving site is minimized.
The surface and point voting fields we can derive from the curve voting field by rotating the field first around the e_3-axis (surface) and then around the e_2-axis (point).

All three fields have in common not to propagate infinitly but decay with distance that they dont have any influence after a given radius.

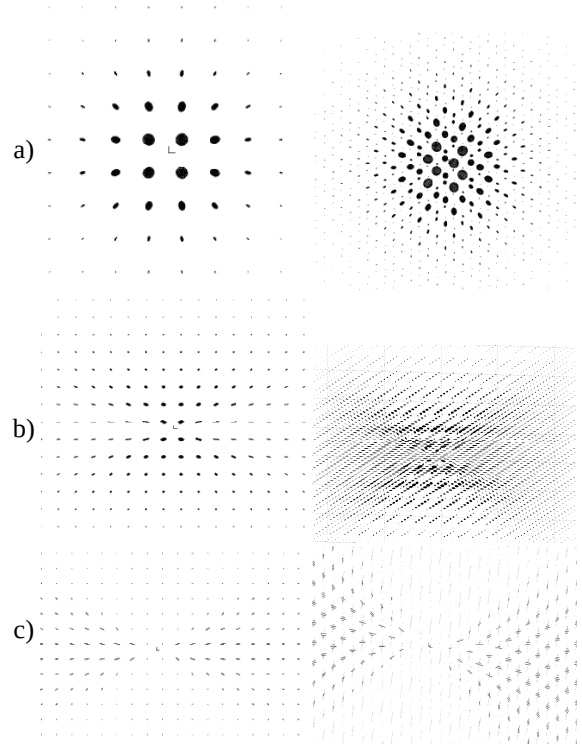

Figure 4: The point (a), surface (b) and curve (c) voting fields. The voting site is centered. On the left the view is along the e_2-axis, on the right the fields are rotated.

2.4 Sparse and dense voting process

To apply the tensor voting to LIDAR-data, we have to encode the data points as described in 2.2. The values for the roation and the dimension of the initial ellipsoid has to be chosen, e.g. scaled and oriented by the confidence ellipse of the measured data, but even other geometrical information can be encoded this way. With no additional knowledge, we can assign a simple ballfield without orientation. In a first step, the voting is carried out among the group of the initial data points, i.e. each of the points is influenced by its neighbor points. In this step the data points dont lie close to each other and the number of point is small in contrast to the number of points necessary to describe the entire object in total. Because of that the first step is called sparse voting.
To obtain an approximation of the continous tensor field we have to interpolate the field between the given initial locations. To do this, we sample the space of the point cloud in a grid. At every grid point we calculate the tensor field value by letting the neighbor points of the initial point cloud vote. With this procedure we obtain the discrete tensor field.

3 SEGMENTATION

After the Tensorvoting step we have a dense grid of data points, each defining a tensor location in space. To get connected surfaces and lines in 3D space, we have to search for the most likely locations. We will handle this with a feature extraction. Later on we search for equal parts to merge them to geometrical objects, what we will handle in a segmentation task.

3.1 Feature extraction

To find the wanted meaningful features, we decompose the tensor field into its point-, surface- and curve-field to examine them separately. For each field we can extract these features by searching for locations with the most likelihood, i.e. search for maxima in the fields.
In the point field we can simply search for maxima of its strength (λ_3). In the other two fields we have also to look on the orientation. therefore we build the gradient of the strength (6) and project it on the orientation vector (7). In the curve field we have the tangent vector (e_1), in the surface field we take the normal vector to that surface (e_3). Then we can search for zero-crossings of b.

$$\boldsymbol{g} = \begin{bmatrix} \frac{\partial s}{\partial x} \\ \frac{\partial s}{\partial y} \\ \frac{\partial s}{\partial z} \end{bmatrix} \tag{6}$$

$$b = \boldsymbol{n} \cdot \boldsymbol{g} \tag{7}$$

The resulting points are the most likely locations to be part of curves, surfaces or to be single points.

3.2 Segmentation of planes

For Segmentation of planes there exist many different approaches in the literature. These approaches use different types of procedures (Taylor et al., 1989, Boyer et al., 1994) as well as different homogenity criteria (Besl, 1986, Wani and Batchelor, 1994). Every approach has its special application where it performs best (Hoover et al., 1996). We use here a region growing algorithm that uses a minimum-description-length criterion as homogenity criterion. Here we present a quick overview over the algorithm whereas it is review in detail in (Schuster, 2004).

The mathematical model we try to find are planes, that is why we define every location from the output of the feature extraction as input for the segmentation. We encode the locations as little plates of unit size and orientation of

the surface-part of the tensor.
The algorithm compares each location to its direct neighbors and computes a value for each pair by applying the homogenity criterion. These values are ranked then and the couples which fit best together will be merged. For the merged locations the homogenity criterion is again applied in combination with each of the new neighbors. This procedure is repeated until no fitting pairs are found anymore. The metioned homogenity criterion is based on the minimum-description-length (MDL) principle (Rissanen, 1987). The goal of the MDL principle is to select the simplest model that explains the data. This is based on the information theory (Shannon, 1948) where it is possible to express the likelihood of an event with the length necessary to encode its occurance. In this case we have two models to compare: two neighbors fit or dont fit. therefore the plane, its uncertainity of orientation in space and the length of the boundary of the two single regions are compared to the hypotheticaly merged case. The value of saving if coding length in case of a merging is the desired value. The higher is it the earlier the neighbors are merged.

3.3 Segmentation of curves

For the segmenation of 3D curves there are also diverse methods to find in literature (Lindeberg and Li, 1997). In our case we define simple region growing like in the plane segmenter, but so far we dont apply a mathematical model for the curve. As homogenity criterion we define the simple constraint that neighbored locations are merged, if the distance and the difference of the orientation of the curve-part does not exceed a certain value. By applying this criterion we get a collection of points defining a 3D curve. For visualisation we take these points as sampling points for splines.

4 RESULTS

In this section we present three different datasets. The first one is a synthetical one. It containes a dice with a borderlength of five fig. (5). It consists of equally distributet points on the surface of the dice. These points are contamined with gaussian noise. After the first, sparse voting step the points have influenced each other fig. (6). The result of the second voting step is shown in fig. (7), which are the decomposed fields after the maximum search.

The second data set is data of an airborn laser scanner, that contains high voltage power lines fig. (8). In fig. (9) there is an enlargement of the scene where the arcs of the power lines are replaced by a spline which has been created with the segmented points.

The third data set is taken by a terrestrial laser scanner and contains a facade fig. (10). In fig. (11) there are the points replaced by the planes found by the segmenter of 3.2

5 CONCLUSION AND OUTLOOK

We presented a segmentation algorithm which yields good results for the processing of terrestrial or airborne LIDAR-data.

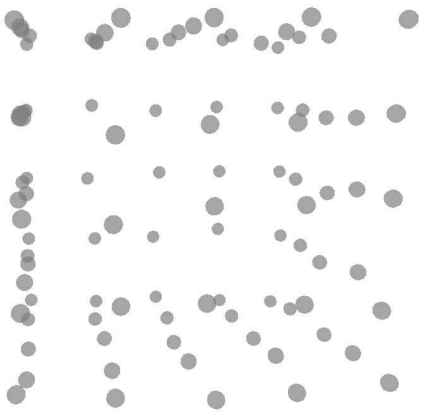

Figure 5: The test dice

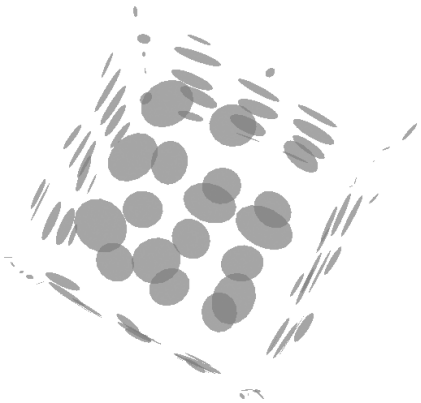

Figure 6: Result after the first voting step

The advantage of the preprocessing with the tensor voting framework is that featured like edges, even if it is only implicitly contained in the data, emerge from the point cloud by continuing the tensor field. By this the decision of fitting data points to one or an other mathematical model in the segmentation step can be avoided, and the segmentation algorithm can be kept simple.
A problem is the value of σ, the range of influence inside the tensor voting. It has to be chosen in the context of the data-characteristics and can destroy the results if it is badly chosen.
Here it is solved in the knowledge that the data have the characteristics of LIDAR-data. So they are euqally distributed in a grid-like structure in x-y-plane and can be triangulated in 2D in the view from the Laser scanner. The value of the dense grid and the σ variable is calculated as mean value over all distances of the triangulated Network over the point cloud. More investigation is necessary at this point.
Some improvement can also be done by implementing different geometrical models in the segmentation procedures. In this context it would be interesting to extract for example general surfaces and chain lines.

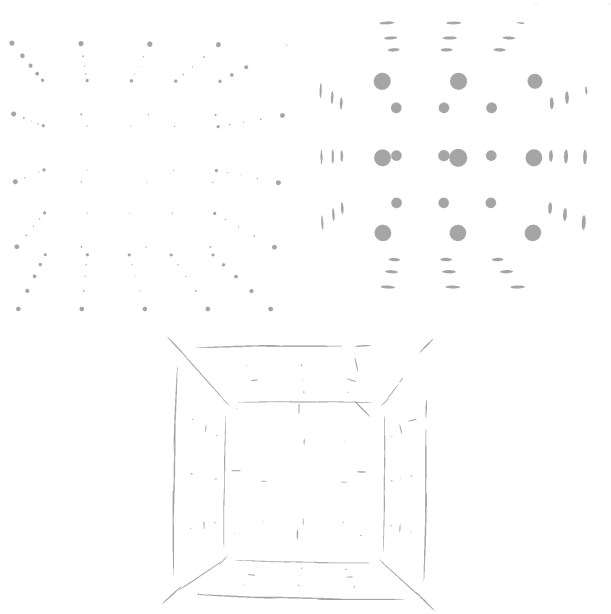

Figure 7: Result after second voting step with decomposition into point, surface and curve field

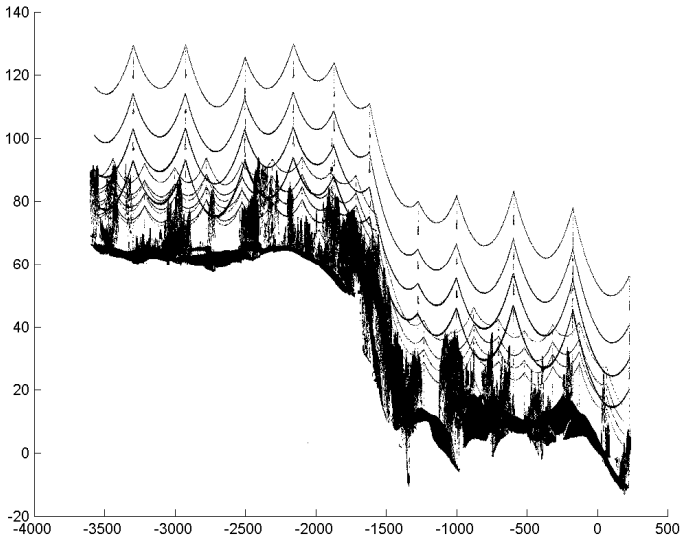

Figure 8: Airborn LIDAR data with high voltage power lines.

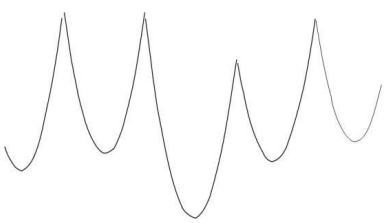

Figure 9: The segmented line points, taken as sample points for splines.

Figure 10: A facade recorded by a terrestrial laser scanner.

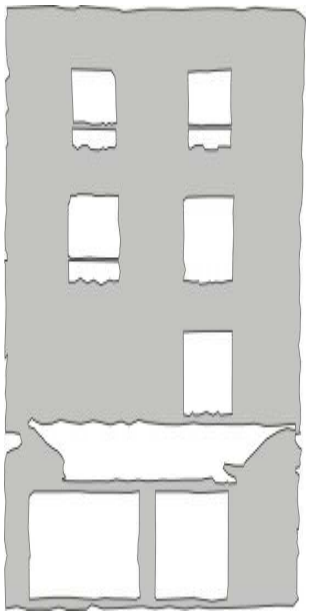

Figure 11: The main plane in the segmented facade.

6 ACKNOWLEDGEMENT

Thanks to GÉOMÈTRE EXPERT FONCIER, B. MOREL †, VONNAS, FRANCE and TOPSCAN GMBH, STEINFURT, GERMANY for kindly providing the datasets.

REFERENCES

Besl, J., 1986. Invariant surface characteristics for 3d object recognition in range images. CVGIP 33, pp. 33–80.

Boyer, K., Mirza, M. and Ganguly, G., 1994. The robust sequentioal estimator: A general approach and its application to surface organisation in range data. IEEE T-PAMI 16, pp. 987.

G. Guy, G. M., 1997. Inference of surfaces, curves and juctions from sparse, noisy 3d data. IEEE T-PAMI 19(11), pp. 1265–1277.

Hoover, Flynn, Bunke and Bowyer, 1996. An experimental comparison of range image segmentation. IEEE T-PAMI 18(7), pp. 673–689.

Lee, M. and Medioni, G., 1998. Inferring segmented surface description from stereo data.

Lindeberg, T. and Li, M.-X., 1997. Segmentation and classification of edges using minimum description length approximation and complementary junction cues. Computer Vision and Image Understanding: CVIU 67(1), pp. 88–98.

Nicolescu, M. and Medioni, G., 2003. Perceptual grouping from motion cues - a 4-d voting approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(4), pp. pp. 492–501.

Rissanen, I., 1987. Minimum description length principle. Encyclopedia of Statistical Sciences 5, pp. 523–527.

Schuster, H.-F., 2004. Segmentation of image and range data using a mdl-criterion. Technical report, IPB Institut for Photogrammetry Bonn.

Shannon, C. E., 1948. A mathematical theory of communication. Bell Syst. Technical Jrnl. 27, pp. 379–423, 623–656.

Tang, C.-K. and Medioni, G. G., 1998. Extremal feature extraction from 3-D vector and noisy scalar fields. In: D. Ebert, H. Hagen and H. Rushmeier (eds), IEEE Visualization '98, pp. 95–102.

Tang, C.-K. and Medioni, G. G., 1999. Robust estimation of curvature information from noisy 3d data for shape description. In: ICCV (1), pp. 426–433.

Tang, C.-K., Lee, M.-S. and Medioni, G., 2000. Tensor Voting. Elsevier.

Taylor, R., Savini, M. and Reeves, A., 1989. Fast segmentation of range imagery into planar regions. CVGIP 45, pp. 42–60.

Wani, M. and Batchelor, B., 1994. Edge-region based segmentation of range images. IEEE T-PAMI 16, pp. 314–319.