Hierarchical segmentation for unstructured and unfiltered range images

C. S. R. Aguiar, S. Druon, A. Crosnier
LIRMM, UMR 5506 Université Montpellier II - CNRS
161 rue Ada, 34000 Montpellier - France
aguiar,druon,crosnier@lirmm.fr

Abstract

We present a method for the segmentation of unstructured and unfiltered 3D data.

The core of this approach is based on the construction of a local neighborhood structure and its recursive subdivision. 3D points will be organized into groups according to their spatial proximity, but also to their similarity in the attribute space. Our method is robust to noise, missing data, and local anomalies thanks to the organization of the points into a Minimal Spanning Tree in attribute space.

We assume that the 3D image is composed of regions homogeneous according to some criterion (color, curvature, etc.), but no assumption about noise, nor spatial repartition/shape of the regions or points is made. Thus, this approach can be applied to a wide variety of segmentation problems, unlike most existing specialized methods. We demonstrate the performance of our algorithm with experimental results on real range images.

1 Problem statement

Segmentation is the process of grouping parts of data into segments that are homogeneous according to some criteria[1]. It is usually an intermediate phase, in which objective is mostly a substantial reduction in data volume and use segmented regions in higher-level processing, such as image recognition, reconstruction, and modelling. When dealing with three-dimensional (3D) images, segments correspond to compact surfaces or volumes. Recognizing parts on assembly lines, reconstructing a CAD model from an unstructured input data, recognizing physical anomalies from medical 3D images and 3D scene modelling are some applications where segmentation plays a fundamental role.

In this paper, we are interested in segmenting 3D images taken from the real world. These images are composed of free-form objects, from unknown statistic population (usually non-Gaussian [2]), variable points density (scattered) and significative regions in multiple scales, for example, an image composed by a set of objects with different sizes. These images are mostly represented by an unorganized sampled point cloud. No information about their structure, nor the topology of the objects presented are supplied by the acquisition system or by the application.

The segmentation algorithm proposed in this paper takes into account this lack of information about the image characteristics: no prior assumption is made about the data.

In our segmentation algorithm, we group points with local similarities, using a hierarchical approach. 3D image segmentation using local similarity has been considered before [3][4]. The main difference between the present algorithm and its predecessors is that our segmentation algorithm can treat 3D images from different nature, obtained with sensors based on Laser, fringe projectors, CT-scan, MR-scan, SEMs, among others.

The method relies on the recursive cut of a neighborhood graph. The connexity of this graph embodies the spatial relationships between the points, while the weight of each edge represents the local variation of a user defined cutting criterion. From this graph is extracted a Minimum Spanning Tree (MST), which is the support of the segmentation. The use of a MST guaranties the method to converge towards well-conditioned regions in terms of connexity and variations of the segmentation criterion.

This algorithm is not dedicated to one specific cutting criterion. Any locally defined attribute (color, density, curvature, etc.) can be used.

The paper will be organized as follow: in section 2 we give a brief overview of segmentation methods applied to 3D images. In section 3 we describe the proposed algorithm. Our experimental results using real range images are presented and discussed in section 4. Finally, some conclusions and the future works are exposed in section 5.

2 Related work - Segmentation methods

In this paper we address the problem of grouping points into homogeneous regions, or region finding approach.

Region finding methods are categorized by the way in which points are gathered into regions. Mobile centers, also called region-growing and bottom-up approaches, initially segment the image into unit cells (particular points in the image). This step is the most critical, because cells placed on a noise or boundary points lead to erroneous results. Then, neighbor points are merged based on a similarity function. As points are added to the region, the seed points move toward the center of the region. Most of methods falling in this category propose algorithms to robustly place these seed points.

Density-based, or top-down methods, are the second category of region finding methods. In this approach, the original pointset is recursively subdivided into smaller regions until each region reaches constant density according to some region similarity function. Among density-based algorithms, we can cite DBscan [5] and hierarchical segmentation [6].

The similarity function can be either data-based or model-based, depending on the image nature and the application. Model-based similarity functions [7] fit points to some known model. Consequently, a prior knowledge about the image nature is required. Data-based functions [4], on the other hand, are based only on local information, and they are especially efficient when dealing with sampled data.

Many methods were proposed in the literature to the specific problem of 3D range images segmentation. In [7], a good overview of a model-based planar range image segmentation algorithms is presented, and experimental results are provided to compare these methods. In [8], 3D images from general scenes are segmented into elementary surfaces, planes, conics, splines and 3D histogram for non-parametric free form objects. In [9], is suggested a method to segment terrain-like and cylindrical volume images.

While the previous methods are more or less application-dependant, we address the problem of segmentation for unstructured and unfiltered 3D data. Thus, the algorithms based on prior knowledge of models cannot be used. Furthermore, because any local criterion can be chosen for segmentation (color, local curvature, etc.), no assumption should be made concerning noise or spatial repartition/shape of the segments.

3 A description of the algorith

3.1 Terminology and algorithm overview

We consider a 3D image defined as a set of sampled points $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. Each element of this set $\mathbf{x}_i = (\mathbf{p}_i, a_i)$ is composed of the point coordinates $\mathbf{p}_i \in \Re^3$ and its attribute value $a_i \in \Re$. Brightness, distance in color space, texture or curvature are examples of such attributes.

From this set of sampled points, we aim at building spatially compact subsets characterized by the continuity of the attribute values over the subset. The overview of our segmentation algorithm is showed in fig. 1, and it proceeds as follows:

1. *Neighborhood Graph Building*: The spatial neighborhood information of each point is gathered in an undirected graph structure. This graph is referred as the Neighborhood Graph. The nodes of this graph are the points of the dataset.

2. *Edge Weighting*: To each edge of the neighborhood graph is assigned a weight which is the distance according to the attribute values.

3. *MST Extraction*: From the weighted neighborhood graph is extracted a Minimum Spanning Tree. The edges of this spanning tree link points which are neighbors in both euclidian and attribute spaces.

4. *Recursive cutting*: The final step of our method is the segmentation itself. A hierarchical cutting algorithm recursively splits the regions until they reach homogeneity in attribute space.

3.2 Neighborhood Graph Building

Neighborhood around a point is a local surface descriptor. In our algorithm, neighborhood is used to obtain points connectivity, and to guarantee that segmented regions are spatially related.

For each point, $\mathbf{p}_i \in \Re^3$, we form a neighborhood $Nbhd(\mathbf{p}_i)$ of all points around $\mathbf{p}_i$ inside a sphere centered in $\mathbf{p}_i$ with radius R. The radius R is chosen to be proportional to the points density, σ.

Points density is an unknown variable which must be identified in our approach. This is done by picking randomly some sample points, and for each sample find its closest neighbor and compute their distance. The density is estimated as being the average distance between closest points. Finally, the sphere radius is defined as $R = N * \sigma$, where N is an input parameter.

The neighboring information is then used to construct an undirected graph G, called *Neighborhood Graph*. Nodes in the graph represent the sampled points and every edge $E_{i,j}$ links the point $\mathbf{p}_i$ and its neighbor $\mathbf{p}_j \in Nbhd(\mathbf{p}_i)$.

3.3 Edge Weighting

We apply an edge weighting to the neighborhood graph in order to represent points relations in the chosen attribute space.

The edge weighting consists in assign to each connected pair points, represented by the edge $E_{i,j}$ in the Neighborhood Graph G, the cost $w_{i,j} = |a_i - a_j|$, which is the points distance in the attribute space. We obtain then not only connectivity information from the edges of the neighborhood graph, but also attribute variation through surface/volume, explicited by the edges weight.

3.4 MST Extraction

The next step of our algorithm is to extract a Minimum Spanning Tree in attribute space from the weighted neighborhood graph.

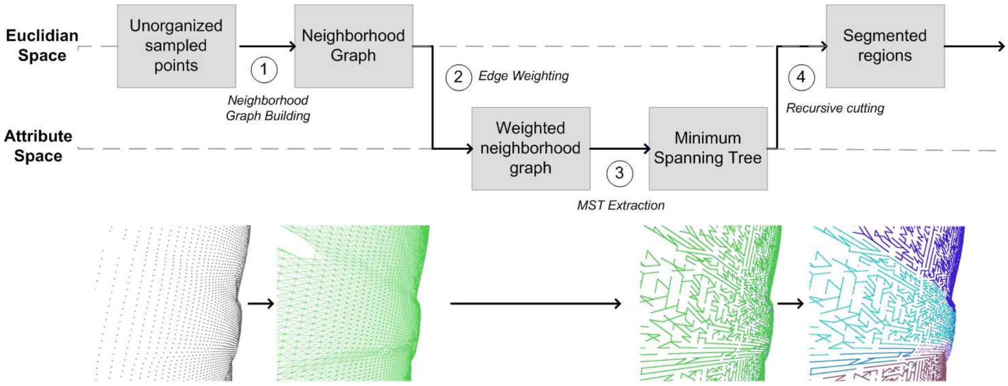

Figure 1: Segmentation algorithm overview.

The Minimum Spanning Tree is a subgraph of a graph. The MST connects all points (nodes) from the graph, forming a tree. All edges presented in the MST is extracted from the original graph.

In a MST, there are no cycles between any two points. This defines a tree, and among all possible spanning trees, the MST is the one (or ones) where the sum of all costs is minimal. All the algorithms encountered to construct MST are based on the following two properties:

- *Identifying edges that must be in a MST:* given a graph G (fig. 2, top left), a **cut** in the graph is the partition of the nodes into two disjoint sets. A **crossing edge** is the one that connects a node in one set with a node in the other (fig. 2, top right). Given any cut in a graph, every minimal crossing edge belongs to some MST of the graph [10].

- *Identifying edges that must not be in a MST:* given a graph G (fig. 2 top left), if we add a new edge $E_{i,j}$, the new MST is the one constructed by adding $E_{i,j}$ to the original MST and deleting a maximal edge on the resulting cycle (fig. 2 bottom left).

In our segmentation algorithm, a MST is extracted from the Neighborhood Graph using the classic Prim's algorithm [11]. From the MST algorithm construction, it is always preferred paths in gradient direction and, if any edge $E_{i,j}$ is removed from the MST, two new independent MSTs are formed. The first property ensures that paths passing through discontinuity regions in attribute space are avoided and noise points and its neighbors are leaf nodes in the MST. From the second property, if nodes of the MST represent points of a given region, removing any edge of the MST create two independent MSTs and, consequently, the original pointset is divided into two subsets.

Noise is amplified when weighting the edges of the Neighborhood Graph in attribute space, due to the differential nature of the edge cost. Because of that, one can expect a large cost between noise regions and the rest of the samples. In MST algorithm they're naturally placed in leaf nodes. This behaviour is explored in segmentation procedure to filter region segmentation.

3.5 Recursive cutting

The final step of our algorithm recursively breaks the regions until they reach homogeneity according to the chosen attribute. We used the hierarchical clustering, an unsupervised hierarchical segmentation technique applied specially to segment sampled data from an unknown distribution [6] .

In Hierarchical clustering, the number of partitions obtained at the end of segmentation is unknown and it depends mostly on the input data, the noise level and the similarity function. This behaviour makes it the most appropriate technique to segment 3D images, when they are treated just as a sampled points cloud.

At the initialization, all points belong to the same region and the MST explicits the relation between points in attribute space. The principle of the segmentation procedure is to recursively remove edges from the MST. Every time an edge is removed, two new disjoints MSTs are created, and the original region is splitted in two new regions. The attribute variation through each new region is smaller than the variation through the original one. This process is terminated when homogeneity in attribute space is achieved.

The segmentation algorithm recursively proceeds as follow: it traverses a given MST (representing a region) and it takes the link that has the biggest cost, $E_{i,j}^{max}$. Then, a noise test is performed on the link nodes of $E_{i,j}^{max}$ by seeing if one of these points $\mathbf{p}_i$ or $\mathbf{p}_j$ are leaf nodes of the MST. If one of the nodes is a leaf node, smoothing is applied to the link cost $w_{i,j}^{max}$ and the segmentation procedure is restarted. Otherwise, the similarity function is applied to determine if the region must be splitted in two or not. In our algorithm, the similarity function is simply a comparison between the

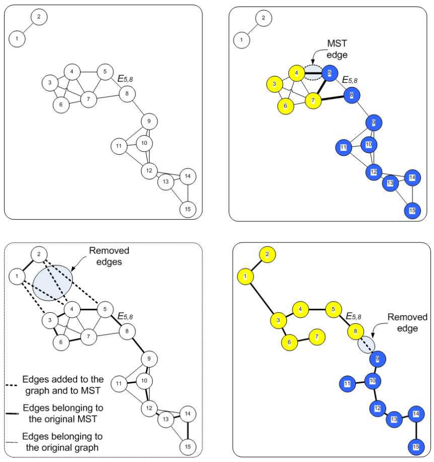

Figure 2: Minimum Spanning Tree properties.

biggest cost, $w_{i,j}^{max}$, and a cutting criterion. The cutting criterion is a threshold value and the region is splitted in two if the biggest cost is above the threshold. Region partition is done by removing the edge $E_{i,j}^{max}$ of the original MST. Segmentation procedure is repeated until all edges linking the points of a given region (MST) have their cost bellow the threshold value.

The drawback of this technique is that the cutting decision depends upon only one link between two points. It makes the original algorithm sensitive to noise and to data distribution, and it could cause surface/volume over-segmentation. Robustness is achieved in our algorithm by applying the cutting procedure to the MST.

The cutting criterion is a simple threshold that depends on the attribute space and the level of detail one wants to extract from the image. For small threshold values, large surfaces tend to be over-segmented and smaller surfaces are correctly extracted. The opposite is observed for big values. Tuning is necessary to choose the best threshold value but it depends only on the attribute space, and it doesn't depend on the image nature. The cutting criterion can be also a value associated with some statistical property. This makes the process automatic, but is out of the scope of this paper, once we want to work with points from unknown population.

4 Algorithm evaluation

In this section we present the algorithm setup, the nature and characteristics of the images treated and some results.

4.1 Algorithm setup

Three input parameters are taken in our algorithm: the radius R used to construct the Neighborhood graph, the attribute space we want to segment the image and the threshold used as cutting criterion. The experiments presented here will illustrate how these parameters influence the image segmentation.

Attribute space is chosen depending on the information of the input data one wants to segment. Normal vectors, curvature, distance between points, are examples of attributes associated to a point that characterize locally a surface. In our experiments, two differential descriptors which provide curvature information can be used as attribute: sphere fitting and the angle between normal vectors. They're preferred because their variation on homogeneous surfaces is small and they require low memory space, which is an important issue when dealing with large data sets.

When the sphere fitting is chosen as attribute, we assign to each point $\mathbf{p}_i$ the radius estimated by the sphere fitting to the neighborhood $Nbhd(\mathbf{p}_i)$. When angle between normal vectors is chosen, we assign to each point $\mathbf{p}_i$ the biggest angle between the normal vector of $\mathbf{p}_i$ and its neighbors.

A fundamental geometric component used to estimate those curvature attributes is the oriented tangent plane associated with a point, $Tp(\mathbf{p}_i)$, composed of its center point, $\mathbf{o}_i$ together with its unit normal point $\hat{\mathbf{n}}_i$. In our algorithm, the normal vectors are estimated by plane fitting using the technique described in [12]. From the neighborhood graph, we take only a fixed and small $k-nearest$ points around $\mathbf{p}_i$ to estimate the normal vector, in order to augment the robustness of the plane tangent estimation [13].

4.2 Results and Discussion

Most of our image data sets were composed by range images. Different attributes like color, texture, and intensity are supplied by distinct acquisition systems. Range images are generally contaminated by heavy noise, different resolution and missing data. Besides that, we don't use any pre-processing algorithm to smooth the surface or to eliminate noise.

Range images representing complete and partial object were explored. We tested the capability of the algorithm to segment the image into meaningful surfaces, the influence of the input parameters in the final segmented images and we analysed if equivalent surfaces in different images were treated similarly by our algorithm and if these surfaces had compatible attribute descriptors.

To validate our algorithm and its applicability to a large variety of 3D images, we took performance measurement over a large number of range images, as suggested in [7]. Over 30 range images of our database were used, and the most significative results are showed here.

The first experiment verifies the repeatability of our segmentation algorithm. Fig. 3 illustrates the original and segmented range images taken from the same object, but from different points of view. Angle between normal vectors was the chosen attribute in segmentation for all images, due to the richness of geometric information of the object. The same input parameters were used in segmentation of all images. Looking at the segmented regions obtained on all three images by our segmentation algorithm and comparing them, we can notice that the algorithm is repeatable, once it generates similar regions, with similar attribute descriptors, for equivalent surfaces. The differences observed in the obtained regions were mainly due to missing data. We can also notice that details in different scales were correctly extracted by our algorithm. Some small and meaningless surfaces are generated especially because of boarder points. These points have generally smaller density compared to the rest of the image points, and consequently, attribute estimation is less accurate over the entire region, and it is not recognized by our algorithm as noise. A simple region merging algorithm could overcome this problem, but it was not implemented in the present approach.

The second experiment verifies the influence of the input parameter values on the final segmented regions. The original and segmented images are showed in fig. 4. The only input parameter that remained unchanged in this experiment was the attribute space, where we chose the angle between the normal vectors space. Fig. 4(a) shows the original image, and fig. 4(b) is the segmented image, when we take the same input parameters as the ones used in the first experiment. These input parameters are used as default when working in angle between normal vectors space, so the segmented image (fig. 4(b)) is taken as reference. The analysis of the influence on the input parameters on the segmentation is based on the number of regions formed, the quantity of points in each region, and in visual inspection.

Figs. 4(c) and (d) show the segmented image when we reduce and increase in 20% the default threshold value used as cutting criterion, respectively . Comparing the segmented images, we can notice how the threshold value interferes the higher-level information provided by the segmented regions. For smaller threshold, image is decomposed into many more segments, where small parts composing the image, like the statue's eyes and mouth, were efficiently extracted. However, some meaningless regions were also presented due to noise, border regions, and holes presented in the image. For bigger threshold, on the other hand, the number of regions decreases, but these regions contains a larger amount of points and they provide a good information about the global topology of the image, once it is only outlined the main parts that compose it.

Fig. 4(e) shows the segmented image when we use a neighborhood radius 40% smaller than the one used as default. In this configuration, noise visibly affects the segmentation, once the image was clearly over-segmented. This result shows that the robustness of our algorithm to overcome noise is dependent on the choice of the neighborhood sphere's radius. This parameter not only influences the MST construction, and consequently, how noise paths are avoided or placed on leaf nodes, but in the present case, it also determines the accuracy on the estimation of the attribute, once the angle between normal vector attribute is estimated around the neighborhood $Nbhd(\mathbf{p}_i)$.

The last experiment outlines the influence of the choice of the attribute space in segmentation. For that, we take the same image and segment it according to different attribute spaces: the angle between the normal vectors and the sphere fitting. The original and segmented images are showed in fig. 5. Both segmentation procedures took the same radius R to construct the Neighborhood graph. Although both attributes provide local information about surface curvature, we can observe that our algorithm segmented the image into different regions. This is due to the attribute's nature and the influence of noise on attribute estimation. The main problem of these attributes is that they're computed around a small region and when derivatives are computed, noise present in the input and in normal estimation are amplified. We can see that sphere fitting estimation is more affected by noise, since segmentation in this attribute space created many small and meaningless regions.

5 Conclusions and future works

We have presented here an algorithm for segmenting 3D images into homogeneous regions, according to some attribute. The results showed that our algorithm is robust to noise, missing data and local anomalies. This was only possible because we organize points into a Minimum Spanning Tree in attribute space, extracted from the weighted Neighborhood Graph. The properties of the MST construc-

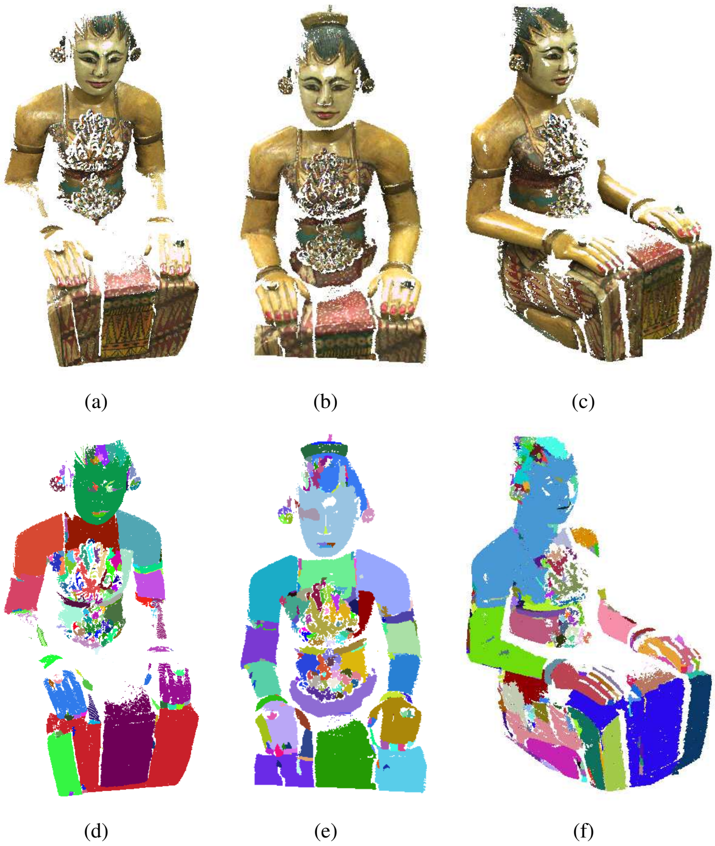

Figure 3: Repeatability and correspondence test. Original range images (a), (b) and their respective segmented images, (c), (d). Attribute considered in segmentation: angle between normal vectors.

tion and the spatial connectivity constraint imposed by the neighborhood graph ensure that discontinuity is avoided, noise are placed in leaf nodes and they can be recognized in the recursive cutting process. The use of MST guarantees that the method converges towards compact and homogeneous well conditioned regions.

For future work, we aim to study the effect of the choice of the radius R used to construct the Neighborhood Graph on the robustness of segmentation algorithm. The same theorical analyses could be performed on the other input parameters, as the cutting threshold. This would make our algorithm even more robust to noise and if these input parameters could be estimated correctly, the segmented algorithm proposed would become automatic. Some good results were obtained when applying the segmentation algorithm to volumetric images, but more experiments are still required.

References

[1] Klaus Koster and Michael Spann. MIR: An approach to robust clustering-application to range image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(5):430–444, 2000.

[2] Kim L. Boyer, Muhammad J. Mirza, and Gopa Ganguly. The robust sequential estimator: A general approach and its application to surface organization in range data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(10):987–1001, 1994.

[3] Gustavo Osorio, Pierre Boulanger, and Flavio Prieto. An experimental comparison of a hierarchical range image segmentation algorithm. In *CRV '05: Proceedings of the The 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, pages 571–578, 2005.

[4] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(2):167–192, 1988.

[5] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In

(a) (b)

(c) (d) (e)

Figure 4: Influence of the input parameters. Original range image(a). Segmented images: (b) using the same input parameters of the repeatability and correspondence test, cutting criterion threshold (c) reduced in 20% and (d) increase in 20%, and (e) 40% reduction in neighorhood radius R.

Evangelos Simoudis, Jiawei Han, and Usama Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, Oregon, 1996. AAAI Press.

[6] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. 0-471-05669-3. wiley-Interscience Publication, 2000.

[7] Adam Hoover, Gillian Jean-Baptiste, Xiaoyi Jiang, Patrick J. Flynn, Horst Bunke, Dmitry B. Goldgof, Kevin Bowyer, David W. Eggert, Andrew Fitzgibbon, and Robert B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(7):673–689, 1996.

[8] Feng Han, Zhuowen Tu, and Song-Chun Zhu. Range image segmentation by an effective jump-diffusion method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1138–1153, 2004.

[9] Kang Li, Xiaodong Wu Danny Z. Chen, and Milan Sonka. Optimal surface segmentation in volumetric images-a graph-theoretic approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(1):119–134, 2006.

[10] Robert Sedgewick. *Algorithms in C++: Part 5 Graph Algorithms*. Addison Wesley, third edition, January 2002.

[11] R. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[12] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.

[13] Ranjith Unnikrishnan, Jean-Francois Lalonde, Nicolas Vandapel, and Martial Hebert. Scale selection for the analysis of point-sampled curves. In *Third International Symposium on 3D Processing, Visualization and Transmission (3DPVT 2006)*, June 2006.

(a) (b) (c)

Figure 5: Influence of the attribute space choice on the segmentation. Original range image (a). Segmented image in (b) angle between normal vector space, and (c) sphere fitting space.