# SKELETONIZATION AND SEGMENTATION OF POINT CLOUDS USING OCTREES AND GRAPH THEORY

A.Bucksch [a] , H. Appel van Wageningen [a]
[a] Delft Institute of Earth Observation and Space Systems (DEOS),
Faculty of Aerospace Engineering,
Delft University of Technology
{a.bucksch h.appelvanwageningen}@tudelft.nl

**Commission V Symposium**

**KEY WORDS:** terrestrial laser data, data processing, skeletonization, skeleton extraction, graph theory, graph replacement, graph collapsing, octree, terrestrial laser scanning

**ABSTRACT:**
In this part of literature we will present a new way of extracting a skeleton out of a point cloud. Our approach is based on looking at the attributes of data points in small parts of the space. The small space parts are derived by an octree subdivision. Out of these small parts we generate a graph which describes the structure. By treating this graph with replacement rules, we generate the skeleton out of it. During this procedure we generate the segmentation information too.
At the end we will give a short overview of implementation issues and make a quick look into the future of this method.

## 1.INTRODUCTION

One of the biggest problems in investigating point clouds is to describe the topology of the represented object. In our case the point cloud is derived from a terrestrial laser scanner. Our approach is to segment the point cloud into several components and investigate the parts by means of a skeletonization procedure. This produces one connected object – a tree graph. Although our current research is focused on segmenting natural trees, we try to handle the problem of generating a valid skeleton in a general way.

## 2.RELATED WORK

Several approaches have been done and discussed to produce a skeleton. Cornea et al. came to the conclusion that the potential field is the best solution, after discussing several approaches. But this is done under the assumption that the point cloud describes a closed hull of an object.

Another successful way is explained by Gorte et. al.. In this approach the method of K. Palágyi et al. is used to treat the point cloud. Morphological operators are used after filtering the tree in the voxel domain. This prefiltering only takes the number of points per voxel into account. But this algorithm has a lot of parameters like resolution of the voxel domain, type and size of the structuring element. Furthermore we experienced an increasing tendency of producing cycles in the resulting graph with increasing resolution of the voxel domain.

In 2006 Gorte introduced another way to extract the skeleton out of a point cloud. This method is called Dijkstra skeletonization. Again he makes use of the voxel domain. But in this algorithm a multiresolution approach is used. Then it searches for the shortest path through the object. The results of this algorithm are promising, under the assumption of a closed hull, which is filled by morphological closing.

## 3.ALGORITHM OVERVIEW

The algorithm consists of three major steps. After those steps we will have gathered the information about the skeleton and about a segmentation of the natural tree.

The three major steps are:

1.Generation of an octree
2.Extraction of a graph from the octree cells
3.Removal of cycles in this graph

Step one is a space division, which cuts a three dimensional box in eight subparts. Those boxes we call an octree cell or just a cell. The six sides of the cells are investigated for points within a certain threshold to one of the sides. Such a side is named a touched side.

Each cell is represented by a (3D-) star graph containing the midpoint of the cell and the midpoint of every touched side as vertices. Midpoints get a label M and the points of the touched sides get the label T. This graph will contain cycles, but will have the attribute of a continuous M-T labelled structure through the whole graph. The cycles will be removed by merging the M-vertices of the graph. At the end of this procedure we will end up with the segmentation and the skeleton of the object.

Step one and two will be described in chapter four, while the removal of the cycles of step three is fully described in chapter five. A short conclusion and summary are made in part six of the publication, before chapter seven gives some practical advice in the implementation hints.
Chapter eight will show examples of the results and chapter nine tries to look into the near future of this method.

## 4.OCTREE GENERATION AND GRAPH EXTRACTION

### 4.1Octree generation

In principle an octree is a recursive subdivision of a cell into eight smaller cells. These eight cells have exactly the same size.

As stated in chapter three, we analyse for every cell the sides which are touched. Out of this analysing process we generate rules to break the octree subdivision.

The following rules are applied:

1. If the cell is touched on two sides
2. If the cell is touched on three sides
3. If the cell is touched on four sides, and the midpoint of the touched sides are lying in one plane[1]
4. If the minimum cell size is reached

Further we have to check if the generated cell contains a very fine structure which results in a under segmentation of the space. For this purpose we formulate an under segmentation criteria.

These criteria checks if there are new or missing touched sides in the next subdivision level. An under segmentation is detected,

1. If the next subdivision level has touched sides, which are not shared with the touched sides of the original cell.
2. If not all sides of the next subdivision level, which are lying inside the original cube are touched.

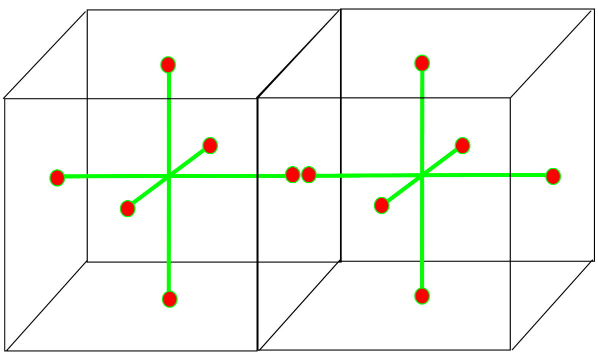

Figure 1. Two octree cells and their corresponding graph. In this case all cell ides are touched. The red sphere denotes a T-vertex., while the M-vertex can be found at the crossing in the middle

### 4.2Graph Extraction

Each of the touched sides is from now on represented through an edge from the midpoint of the side T to the midpoint of the cell M. Every edge get a length as a weight assigned. And every vertex gets a label M or T. This graph we name the representing star graph, which introduces the duality between our graph and the octree subdivision. This duality is shown in figure 1.

[1] This corresponds to the in 4.2 introduced star graph

In the next step all graphs of neighbouring cells are connected to each other through their shared cell sides. This denotes that we are using an 6-neighbour[2]hood as a standard here. The use of a 26- neighbourhood[3] is only indicated if there is no other possible connection between two cells. In our cases this was only rarely the case.

## 5.CYCLE REMOVAL RULES

The extracted graph from chapter 4 has two behaviours, which should be treated.

1. the graph does not preserve the propagated M-T structure from chapter 3. Through out the connection process T-T connections are generated
2. The graph contains different types of cycles, which where generated during the extraction and connection process.

The T-T-structures mentioned are simply eliminated by merging the two connected T-vertices to one. figure 2 is an explanation of this procedure. This is done by averaging the two T-vertices. From now on these T-vertices represent the adjacency information between two elements.

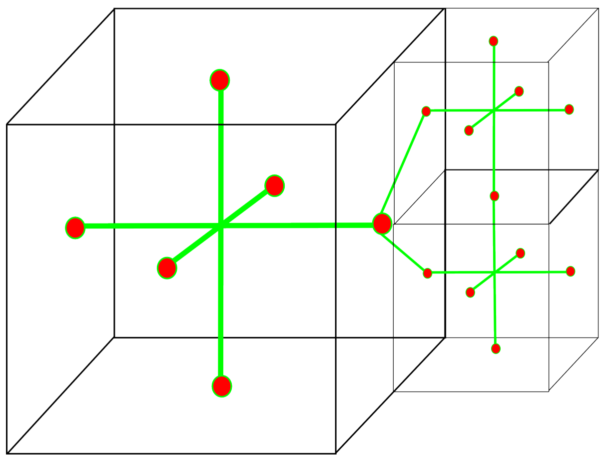

Figure 2. The red spheres indicate the T-vertices which are merged over different levels of the octree subdivision

For the second case we are aware that this is a O(m+n) problem, where m is the number of edges and n denotes the number of vertices.

But in our case the complexity is reduced to a few combinations, which we have to detect. In our graph construction procedure only cycles with four T- and four M-vertices or two M- and two T-vertices can occur. But this does not increase the dimension of the problem. During the following cycle removal the extra case of three M-vertices and three T-vertices can arise and will be treated. All cycle removal rules are resulting in a line segment which has the form M-T-M. Larger cycles are only existent as a special case of a skeleton (e.g. a ring), or related to a not sufficient space division resolution of the octree.

[2] A cell is in the neighbourhood of another cell, if it shares a face with another cell.

[3] A cell is in the neighbourhood of another cell, if it shares a vertex, edge or face with another cell.

**5.1 2M / 2T**

The simplest case is the situation with two M- and two T-vertices. This can be recognized in figure 3. Under this circumstance simply both M-vertices are combined by averaging their position. The value is weighted with the amount of edges incident to the vertex. This weighting is applied to every merged pair of M-vertices in the following rules, without to be mentioned explicitly.

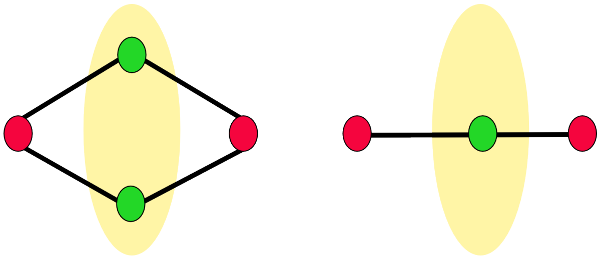

Figure 3. Merging of the 2M / 2T case. The start situation is on the left and the result can be found on the right. Red circles are representing a T-vertex, green circles show a M-vertex. The yellow background denotes the merging operation.

**5.2 3M / 3T**

The following case is the case of three M- and T-vertices. We will merge the two M-vertices which are connected through the shortest incident edge in the cycle. The result is a T-vertex, which is connected to a sub graph (the cycle) described in 5.1. Those T-vertices which are linked to only one other vertex are called T-streaks. In all the following rules T-Streaks will be immediately removed. figure 4 shows the result of this rule.

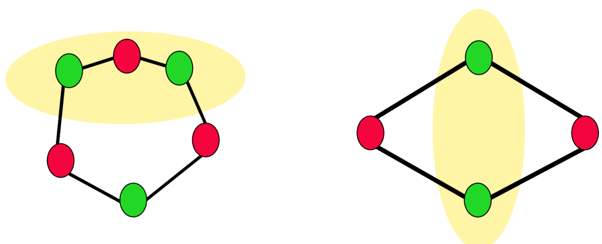

Figure 4. Replacement of the 3M / 3T case. The result on the right side shows the previous 2M / 2T case, without the removed T-streak

**5.3 4M / 4T**

We divide the case of 4M- and 4T-vertices in 4 different sub rules. Those cases are depending on the number of M-vertices with more than 2 Connections in the cycle. We call them xM-vertex.

**5.3.1** 1 xM-vertices

For only one xM connection we will merge all three M-vertices to one xM-vertex. The result is a 2M/2T case with two T-streaks, which will be removed, as shown in figure 5.

**5.3.2** 2 xM-vertices

Two possible combinations have to be solved here.

1. The two xM-vertices are connected through a T M-T subgraph
2. The two xM-vertices are only connected through one T-vertex.

For the first scenario the two M Vertices are collapsed to one xM-vertex. The result of this operation is the desired M-T structure. This can be seen in figure 5 on the right side. For the second circumstance in figure 6 the M-vertices are unioned with their closest xM vertex.

**5.3.3** 3 xM-vertices

Like described in chapter 5.2 we merge the M-vertex with the xM-vertex, which shares the shorter edge with the M-vertex.. This results again in a T-Streak and the case described in 5.2.

**5.3.4** 4 xM-vertices

Again we merge the shortest edge between two xM-vertices. Further more we merge the two other xM-vertices, too.
This outcome of this merging is the previous described M-T structure.

The result after this removal procedure is a tree. To generate a Skeleton out of this tree, we give a geometric constraint to all of the M-vertices during the graph extraction. We achieve this by placing the M-vertices in the center of mass of the point cloud in an octree cell. At the end of the removal procedure we average every T-vertex with respect to its directly linked M-vertices.

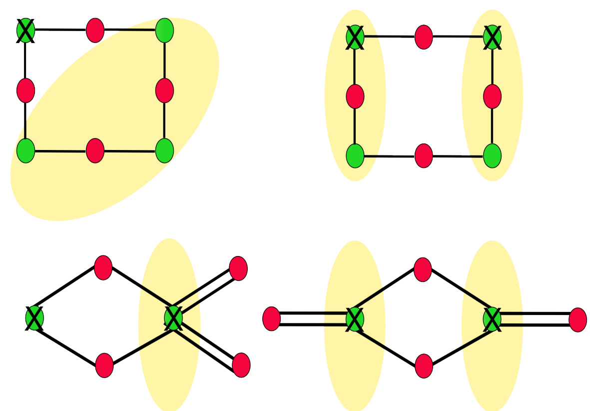

Figure 5. The colouring of the Vertices is the same like in the previous pictures. xM-vertices are additionally marked with an X. On the left the 1xM is shown. The right side gives a picture of the first 2xM situation. The results can be found under each start case. The double linked T-vertices represent the T-streaks, which will be removed. The yellow background denotes the merging operation.

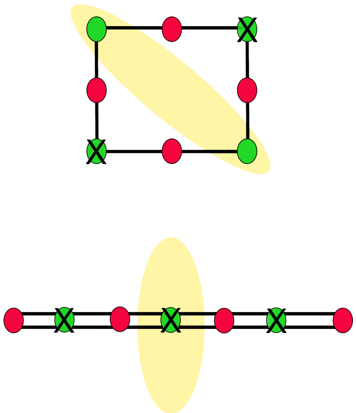

Figure 6. The colouring is according to figure 5. On top the start situation is shown, the result on the bottom. The double inked T-Streaks will be removed. The remaining double connections are merged to a single connection; therefore they do not have any direction.

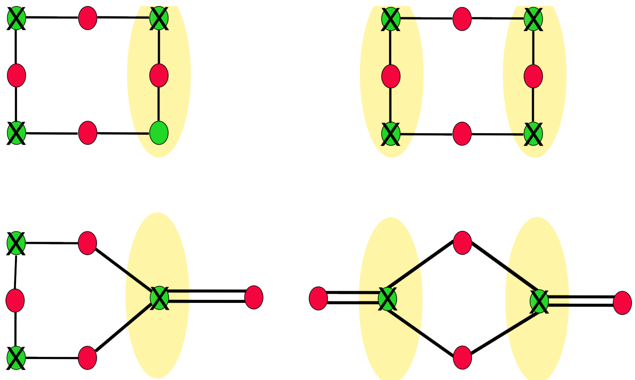

Figure 7. This figure uses again the colour scheme of figure 5. On the left the case with 3 xM-vertices. The right side shows the scenario of 4 xM-vertices.

## 6.RESULTS

In figure 8 a subdivided tree with its octree cells is shown. The overall minimum size of the octree cell was 5 cm. Figure 9 illustrates the corresponding segmentation, while figure 10 give an impression of the resulting skeleton. The chosen minimum cell size for the shortest side of the cell was 10 cm in this case. In figure 9 only small effects of the chosen cell size and the merging procedure are visible as a segmentation error on the stem. Furthermore noise is detected as a part of the branches. The skeleton in figure 10 illustrates all important components of the tree. It reflects some of the segmentation errors, as a missing junction, or wrong segmented noise as a little extra branch.

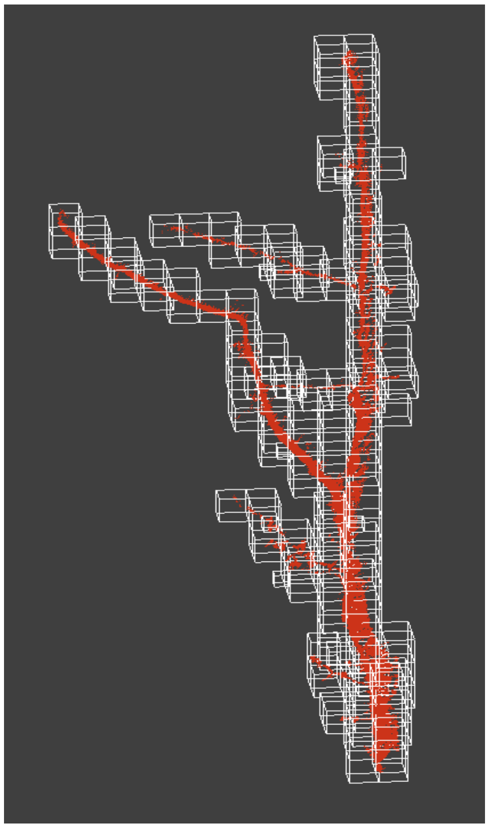

Figure 8. Octree subdivision of a natural tree

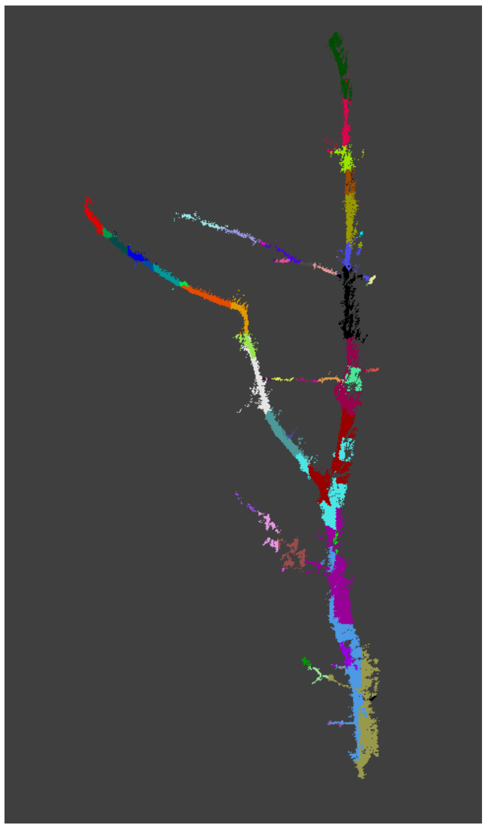

Figure 9. Segmentation of the natural tree by colouring the merged octree cells.

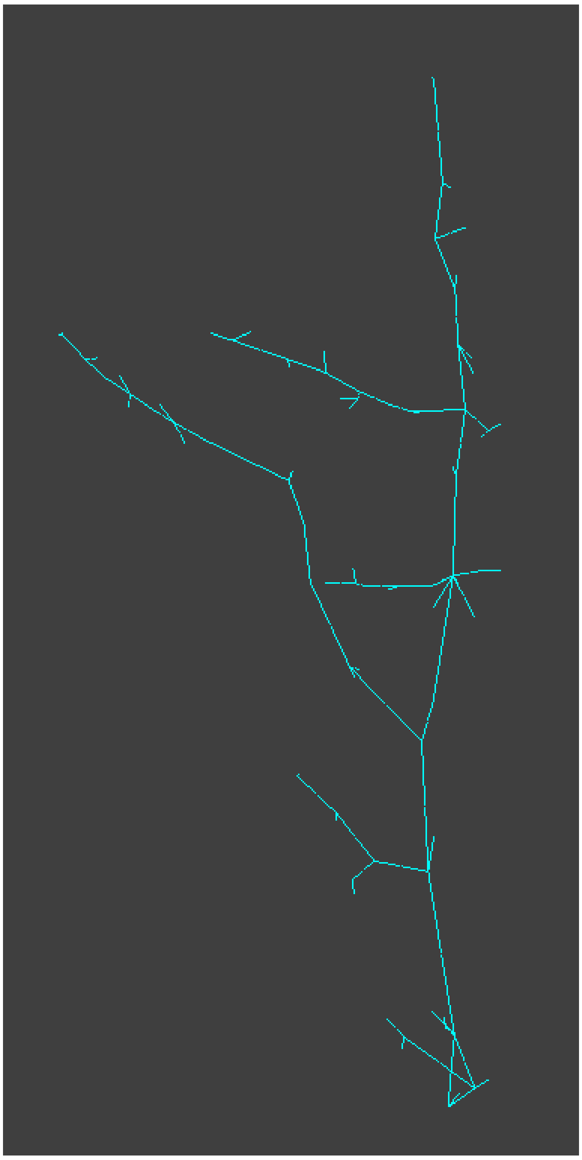

Figure 10. Skeleton extracted from the tree of figure 8 and 9.

Figure 11 shows an unregistered point cloud of a cherry tree, which simulates the influence of gaps in the point cloud on the skeleton. These gaps can happen because of occlusion problems. In this case we simply used the back part of an unregistered tree to visualize this problem. This problem usually occurs more often at inner branches, than at the outer ones like in our illustration. The corresponding octree is illustrated in figure 12. The influence of the gaps can be recognized in figure 13.

Figure 11. Point cloud of a cherry tree

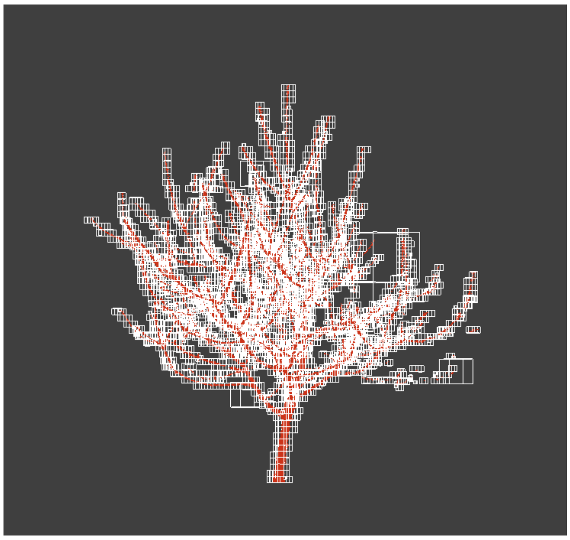

Figure 11. Octree of the cherry tree from figure 11

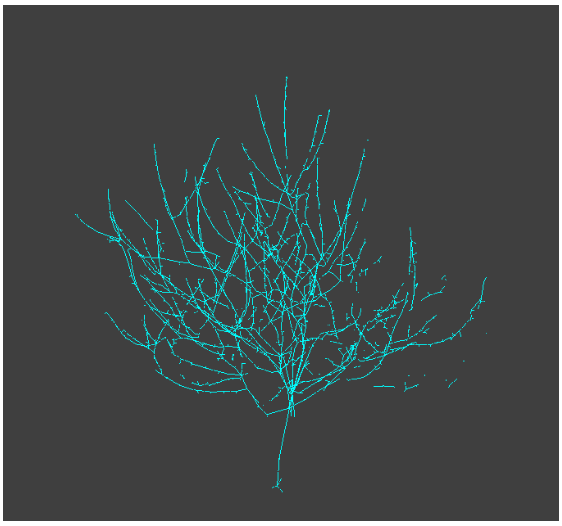

Figure 12. Skeleton of the cherry tree

## 7.CONCLUSION AND SUMMARY OF THE ALGORITHM

In this paper we described a new geometric approach on generating a skeleton out of point clouds by investigating the attributes/behaviour of points in a closed volume (an octree cell). From this attributes we extracted a describing graph, under the assumption of a basic element, which is the star graph from all touched sides to the midpoint of the cell.
By breaking the subdivision procedure of the octree we get the benefits of a multiresolution method.

The extracted graph will be treated by rules which do not generate cycles bigger then the existing cycles in the graph[4]. The described rules produce a skeleton after the removal of all cycles. In the same time we collected the segmentation information by counting all cell indices, which where merged together during the cycle removal. This information is represented in the xM-vertices. The information of adjacency sections is represented in the remaining T-vertices.

## 8.IMPLEMENTATION HINTS

### 8.1Cycle detection during octree generation

Currently the detection of the cyles is the bottle neck of the algorithm, as it is described here. But the creation of such a cycle can be triggered during the generation of the octree. So the time consuming search for cycles is eliminated.

### 8.2Octree of the bounding box

In our special case we achieved a higher speed in the octree generation with satisfying results, by subdividing the bounding box of the natural tree, instead of subdividing the smallest enclosing cube.

### 8.3Adjacency List

It is indicated to use an adjacency list instead of the adjacency matrix. This is because of the fact, that the amount of needed memory can be very huge.

## 9.FURTHER RESEARCH

We will implement a version of the algorithm, which is able to trigger the generation of cycles during the subdivision procedure. We expect a significant speed increase of the algorithm.

Further we want to split the M-vertices with more than four links to M-vertices with three connections. This will avoid the problem that the algorithm can produce a skeleton that is running out of the surface.

During the removal procedure we collect the information of which original vertices are merged to one, and we keep track of the borders to other segments. We will investigate how much this information can improve the segmentation result. We want to try different methods known from cluster analysis and graph weighting techniques.

We want to investigate if the extracted junction points can be used for inverse kinematics, e.g. for use of animation of virtual characters in games and videos.

In the near future we try to increase the accuracy of measuring objects such as trees by using these skeletons.

We want to assure, that in the future a skeleton will have the properties:

1. Being inside the point cloud.
2. A vertex with more than two links should be representable as an animation point in a kinematical model.
3. The points along one edge of the skeleton should be treatable like a rigid object.
4. That all components of an object are represented in the skeleton.

## ACKNOWLEGEMENTS

This research was sponsored by “Arbeitsgemeinschaft industrieller Forschungsvereinigungen” (AiF).

## REFERENCES

Chuang, J., Tsai, C., Min-Chi, K,. Skeletonization of Three-Dimensional Object Using Generalized Potential Field, *IEEE PAMI*, 22(11):1241-1251, 2000.

Cornea, N., Sliver, D., Min P. 2005:. Curved Skeleton applications, *IEEE Visualization Conference (Vis'05), Oct 2005*, Rutgers University, New Jersy, USA.

Gorte, B., Pfeifer, N., 2004: Structuring laser-scanned trees using 3D mathematical morphology. *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXV, B5, pp. 929-933

Gorte, B., 2006: Skeletonization of Laser-Scanned Trees in the 3D Raster Domain, *3DGeoInfo06, Malaysia*, TU Delft, The Netherlands.

Katz, S., Tal, A., Hierarchical Mesh Decomposition using Fuzzy Clustering and Cuts, *SIGGRAPH 2003, ACM Transactions on Graphics*, Vol. 22 , Issue 3, July 2003, 954-961, Technion – Israel Institute of Technology

Palágyi, K., Sorantin, E., Balogh, E., Kuba, A., Halmai, Cs., Erdôhelyi, B., Hausegger, K., 2001: A sequential 3D thinning algorithm and its medical applications, in Proc. *17th Int. Conf. Information Processing in Medical Imaging*, IPMI Davis, USA, Lecture Notes in Computer Science, Springer, 2001

Wilson, Robin J., 1985. Introduction to Graph theory – third edition, Longman Scientific & Technical, Essex,pp.8-96.

[4] The critical part of the rule design is to verify that the complexity of the problem is not increasing.