

SEGMENTATION OF TIN-STRUCTURED SURFACE MODELS

Ben Gorte

Technical University Delft, Photogrammetry and Remote Sensing Section

b.gorte@geo.tudelft.nl

Working Group IV/6

KEY WORDS: Laser scanning, 3-D reconstruction, segmentation, grouping, TIN

ABSTRACT

Segmentation is an important step during 3-D building reconstruction from laser altimetry data. The objective is to group laser points into segments that correspond to planar surfaces, such as facets of building roofs or the (flat) terrain between buildings. A segmentation method is presented that was inspired by a raster-based algorithm in literature, but works on original (triangulated) laser points. It iteratively merges triangles and already formed segments into larger segments. The algorithm is controlled by a single parameter controlling the maximum dissimilarity for adjacent segments such that merging them is still allowed. The resulting TIN segmentation method is compared with 3-D Hough transform.

1 INTRODUCTION

Laser altimetry is continuously gaining popularity. In the Netherlands, for example, a DTM of the entire country, called AHN (Actueel Hoogtebestand Nederland) has been produced by laser altimetry. Due to advancements in technology, the point density of laser altimetry data increases.

Besides converting laser data into 2.5-dimensional DSMs (Digital Surface Models) and DTMs (Digital Terrain Models) it is of particular interest to derive 3-D object models from high-density laser data. This involves *segmentation* or *grouping*: identifying groups of points that correspond to objects in the terrain, rather than treating points as independent (x,y,z) measurements. Also the location of the objects in 3-D has to be established.

Object modeling is best applied to man-made objects, such as buildings. In that case, the segmentation step in the object modeling process attempts to find planar regions (faces) in the point data set, corresponding to roof facets of buildings. Note that faces may be tilted in 3-D space.

Segmentation, for example by region growing or split-and-merge, is known from image analysis literature. These algorithms attempt to form segments of adjacent pixels that are similar, for example in having similar gray value or texture. When applied to the detection of planar regions in laser altimetry data (or other kinds of surface data), the goal is to create segments with constant slope, rather than with constant elevation. Therefore, when using 'image processing' segmentation approaches in this case, not the surface data themselves would have to be segmented, but data sets derived by gradient filtering.

1.1 Laser data segmentation

In addition to image analysis segmentation methods, algorithms have been developed particularly for the purpose of segmenting laser data. [Geibel and Stilla, 2000] implemented four methods from literature and applied these to the same laser data set.

Moreover, the article of [Geibel and Stilla, 2000] presents a new method termed *FOM procedure*, which is shown to perform better than the others. The algorithms have in common that they work in the raster domain. The original (x,y,z) coordinate measurements first have to be transformed into a raster, where a x and y coordinates defines a (row, column) position of a pixel, in which the corresponding z coordinate is stored as the pixel value.

The FOM procedure is based on iterative growing of similar areas from single pixels into large segments. In each step of the iteration, the procedure finds those two adjacent segments that are most similar to each other, and merges them. The process stops when no pair of sufficiently similar adjacent segments can be found anymore. Similarity is expressed in a distance measure $D(P,Q)$ between adjacent segments P and Q (small distance means large similarity). Its definition is based on the Euclidean distance $D(p, Q)$ in 3-D between a pixel p and a plane fitted through all pixels of segment Q . The directed distance $D(P \rightarrow Q)$ of P to Q is now the maximum of $D(p,Q)$ over all pixels p in P . The distance $D(P,Q)$ between P and Q , finally, is the minimum of $D(P \rightarrow Q)$ and $D(Q \rightarrow P)$.

1.2 Grouping the original laser points

The TIN segmentation algorithm presented here is greatly inspired by the FOM procedure. However, in the footsteps of [Vosselman, 2000], who applies, for example, filtering on the original laser points, TIN segmentation directly uses triangulated laser points, instead of a rasterized version. Thereby, the data do not have to be discretised.

[Maas and Vosselman, 1999] describe laser altimetry segmentation, also based on the original points. They use 3-dimensional Hough transform.

Hough transforms were first developed to find lines and other parameterized shapes in point sets in 2-D images [Duda and Hart, 1972]. For each point the sets of possible parameter values are accumulated in a discretized parameter space. Consequently, parameter sets that obtain high accumulated values represent shapes passing through many of the input points. [Maas en Vosselman, 1999] apply the same principle to

3-D point sets, such as laser data sets. For each point p , the parameters a , b and c of planes $z=ax+by+c$ containing p are accumulated in a discretized (a,b,c) parameter space. Combinations of a , b and c that occur many times represent planes that pass through many points. [Goos, 2001] uses a (ρ,θ,λ) parameterization for a plane:

$$\rho = x \cos(\theta) \cos(\lambda) + y \cos(\theta) \sin(\lambda) + z \sin(\theta)$$

where ρ is the distance from the plane to the origin; θ and λ are the angles describing slope steepness and direction. The advantage of a (ρ,θ,λ) over an (a,b,c) parameter space is that the former is finite.

A drawback of the above-described Hough transforms is that they do not take adjacency or closeness (or any spatial relationship other than coplanarity) of the original points into account. A plane can be constructed through points that are very far away from each other and, therefore, belong to completely different objects.

This paper presents TIN segmentation as a method to segment point clouds, which does not suffer from this drawback. A comparison between the Hough transform segmentation and the algorithm proposed here will be given at the end of the paper.

2 TIN SEGMENTATION

The goal of TIN segmentation is to identify planar regions in a Digital Surface Model (DSM) that is represented in a TIN (triangular irregular network) data structure. The algorithm creates segments that are supposed to coincide with these regions. Each segment consists of a set of adjacent triangles in the original TIN, and each triangle is going to belong to one segment (complete partitioning). The program yields a list of segments with certain attributes including parameters (a,b,c) of the plane-equation in 3-D, as well as a segment id. It assigns to each triangle the id of the segment it belongs to.

Boundaries between segments are composed of arcs in the original TIN. Of course, not every arc in the TIN is going to belong to a segment boundary. Likewise, the breakpoints in the boundaries are original data points, but many data points are going to be inside a segment and not on a boundary.

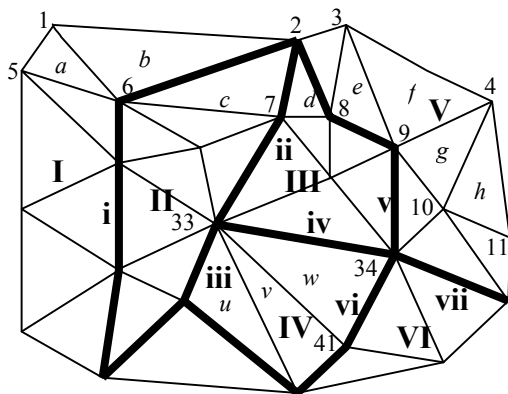


Figure 1: Segmented TIN (intermediate result).
1,2,3 : points; a, b, c : triangles; I, II, III: segments;
i, ii, iii: boundaries.

2.1 Data structure

The data structure that governs the algorithm can be described as a relational schema with four tables, for points (PNT), triangles (TRI), segments (SEG) and boundaries (BND). The point table has an ID-column and three columns x , y and z for the coordinates of each point. Each triangle in the triangle table contains the ID's of three points and a segment ID, denoting the segment that the triangle belongs to. The point table and the triangle table (with empty segment-ID's) are the input for the TIN segmentation algorithm. After the program finishes, the segment-ID column of the triangle table is filled, as are the segment and boundary tables. The segment table stores for each segment the number of participating points and triangles, the area of the segment and the parameters (a,b,c) of the plane through all points according to a least squares fit. The boundary table stores segment topology: the ID of the two segments to either side of the boundary. Also the distance between the two segments is stored, as well as the number of points on the boundary and its length.

The complete data structure consists of 4 relations (tables), for points (PNT), triangles (TRI), segments (SEG) and boundaries (BND). An example of these tables, corresponding to the situation in Figure 1., is shown in Figure 2.

2.2 Algorithm

While the program executes it iteratively merges triangles into segments, and segments into larger segments. During the initialization phase one segment is created for each triangle. The three vertices of the triangle uniquely define the three parameters of the plane containing the segment. Subsequently, all pairs of adjacent segments are analyzed. Between the two segments of each pair similarity (to be defined) is calculated. The most similar pair is selected and the two segments are merged into a single one. The new segment contains more than three points (the number can get arbitrarily large), which probably are not exactly in one plane. The parameters of the plane equation are obtained by a least square fit. The process is repeated until no pair of segments with sufficient (as specified by the user) similarity can be found anymore.

The algorithm needs to measure similarity between adjacent sub-segments. Clearly, similarity has to encompass coplanarity of sub-segments. Moreover, it must be ordered, to be able to select a maximum and compare with a user-specified threshold. Two similarity measures were considered: angle and distance. The angle between two segments is defined as the angle at which the two planes (whose parameters are stored with each segment) intersect. The distance between two segments is defined similarly to the definition in the FOM procedure. The distance from a point p to a segment Q is the (vertical or orthogonal) distance from p to the plane defined by the parameters in Q . The (directed) distance from a segment P to a segment Q is the maximum of the distances from each point in P to Q . Finally, the distance between two segments P and Q is defined as the minimum of the distances from P to Q and from Q to P . Of course, the relationship between *distance* and *similarity* is an inverse one: a small distance means that similarity is high. So far, the algorithm was applied to laser data with high point density (8 points per m^2) and standard deviation of 20-25 cm. In such a situation the orientations of triangles in one planar region may vary considerably, making 'angle' not a suitable similarity measure. Good results were obtained, however, using 'distance', where the threshold value

can be easily related to the standard deviation in the point measurements.

PNT				TRI				
ID	x	y	z	ID	p1	p2	p3	seg
1				a	1	6	5	I
2				b	1	2	6	I
3				c	2	7	6	II
4				d	2	8	7	II
5				e	3	9	8	V
6				f	3	4	9	V
7				g	4	10	9	V
...
33				u				IV→III
34				v	33	34	41	IV→III
41				w				IV→III
...

SEG						
ID	#npnt	#tri	area	a	b	c
I	9	7				
II	9	8				
III	7→10	6→9	!	!	!	!
IV	5	3				
V	9	8				
VI	5	3				
...

BND					
ID	seg1	seg2	#pnt	len	dist
I	I	II	5		!
ii	II	III	3→4	!	!
iii	II	IV	2		
iv	III	IV	2		
v	III	V	4		!
vi	IV→III	VI	3		!
vii	V	VI	2		
...

Figure 2: Data structure and content corresponding to Fig. 1, and updates when removing boundary **iv** between segments **III** and **IV**. Attributes with ! are recalculated.

2.3 Post processing

On the basis of attributes in the segment-table, selections can be made as to keep only the relevant ones. Selection can be based on segment size (using number of point, number of triangles or area), or on segment slope (derived for the plane parameters).

A segmented TIN (either containing the complete set of segments or a selection) can be easily transformed into a VRML model, as is shown in the next section. In the Figures shown there, the laser points were projected into the planes they belong to. Note that points on segment boundaries will be projected into two different planes, and the intersecting line of these planes usually does not pass through the boundary points.

3 EXAMPLES

Three examples will be shown. The first two contain buildings from the village of Wijhe in the East of the Netherlands. They were recorded with the helicopter-mounted FLI-MAP system of Fugro-Inpark. It measures 12,000 points per second. From an altitude of around 60m, a point density of 5 to 10 points per square meter can be reached.

The first example (Figures 3, 4 and 5) is a relatively simple house with a gable roof, an extension (also with a gable roof) and a dormer. Note that the house has roof facets on the short sides of the main wing. The Figures show that all these facets are detected. After selecting segments with slopes less than 150% only the roof and the ground segments remain.

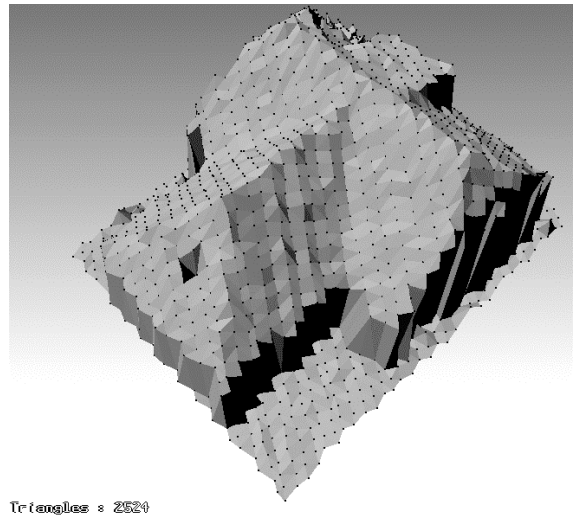


Figure 3: Simple house, triangulated

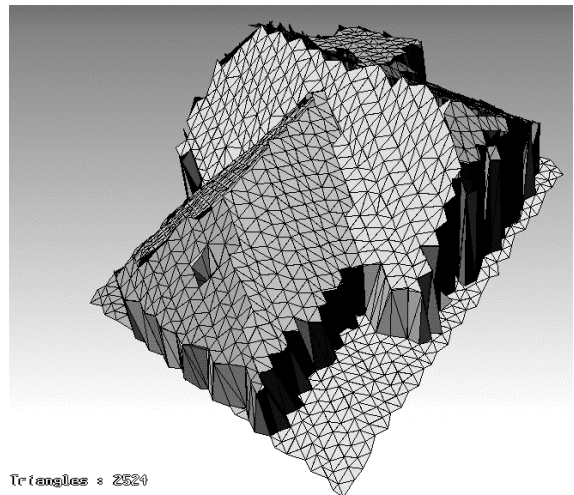


Figure 4: Simple house, segmented

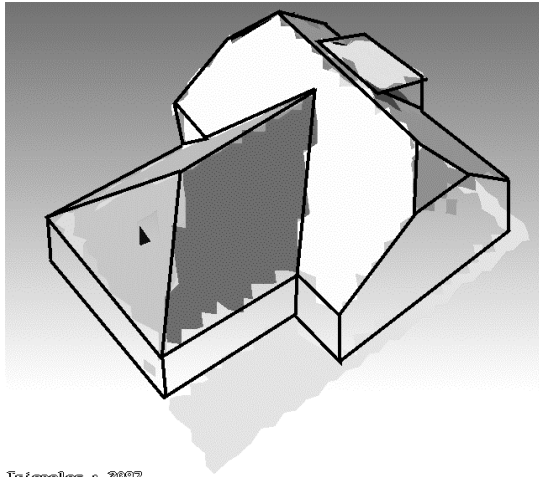


Figure 5: Simple house, selected segments. The lines were added manually for clarification.

The second example concerns a more complex scene with several buildings. Also some vegetation is present, which could be removed after segmentation by selecting segments with a slope less than 150% that contain more that 10 points.

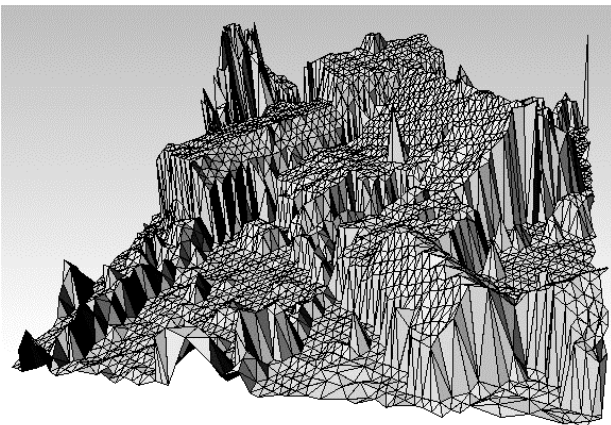


Figure 6: Complex scene, triangulated

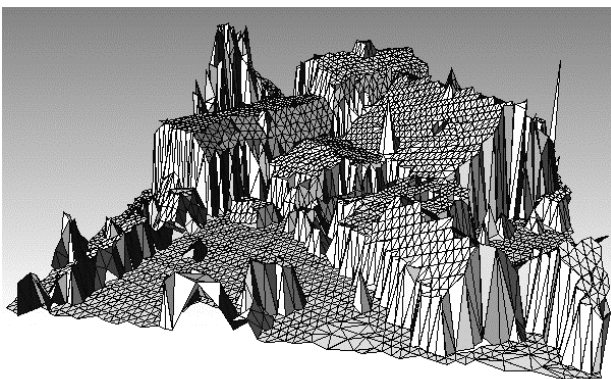


Figure 7: Complex scene, segmented

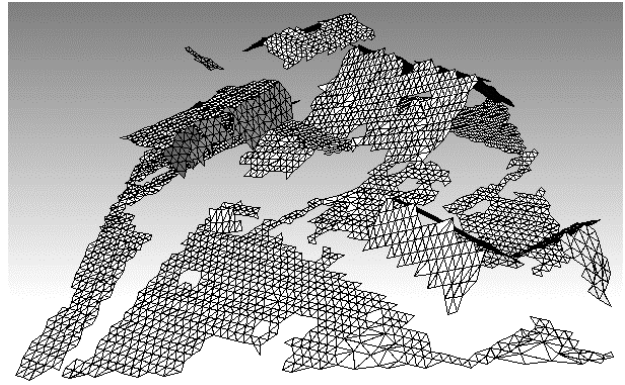
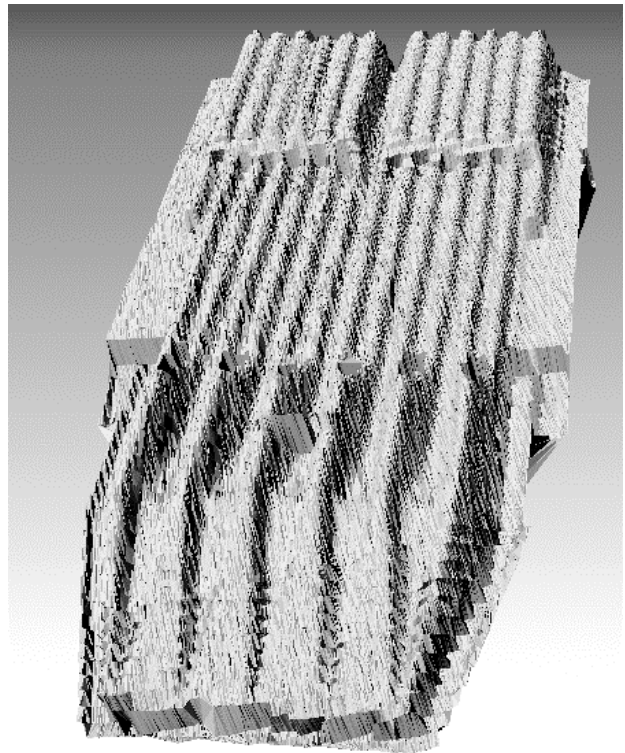


Figure 8: Complex scene, segment selection

The third example is a Toposys laser scan of the roof of the Aula of Delft Technical University. The Toposys scanner is characterized by a very small sampling distance between points within each scan line, and a much larger distance between adjacent scan lines.

The surface of the roof has the form of waves; a profile across the waves would have a sinusoidal shape. These are far from ideal circumstances for TIN segmentation. As a result, many points on segment boundaries are projected to quite different positions into the two segments.



Triangles : 80000

Figure 9: Aula roof

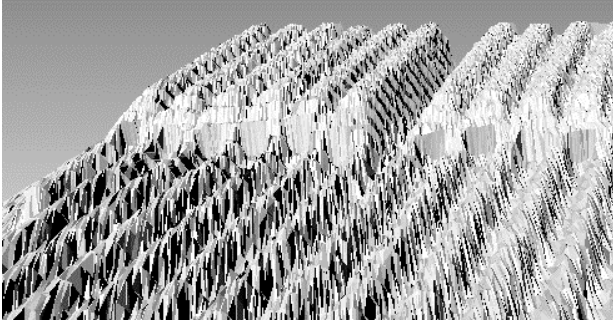


Figure 10: Aula roof detail

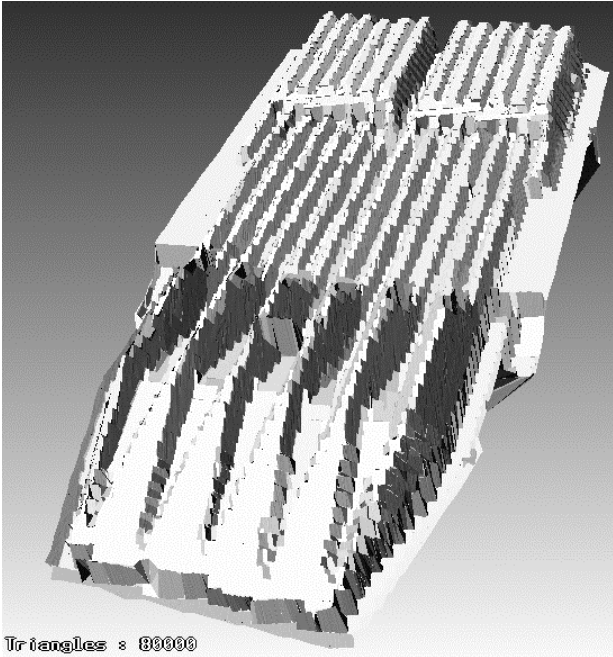


Figure 11: Aula roof segmented

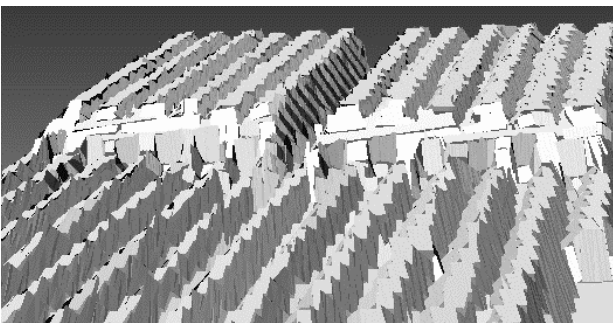


Figure 12: Aula roof segmented detail

4 EVALUATION

TIN segmentation is defined as a recipe and not in terms of the result. It is not guaranteed that the segmentation is optimal in some sense. Perhaps other segmentations are possible satisfying the same maximum-distance requirement and these segmentations may even be 'more optimal', for example have fewer segments. Many segmentation algorithms suffer from order dependency, meaning that the (relative) position of data-items (pixels) in the input data set influences the segmentation

results. For example, quadtree-based segmentation (split-and-merge) is generally not invariant to rotations or shifts of the input data. TIN segmentation does not suffer from order dependency, as long as distances between different pairs of adjacent segments are not (exactly) equal.

Performance

Given the execution time of TIN segmentation for a certain data set, a twice as large data set requires twice as many boundary removals. Note that only one boundary is removed during each pass of the main loop of the algorithm. Each pass requires a single traversal of the boundary information to find the boundary with the smallest distance. The subsequent removal of that boundary requires that distances between the newly formed (combined) segment and all its neighbors are recalculated. Although this is a local operation (not depending on the size of the data set), the boundary table has to be traversed again to find these neighbors. The new distances are stored in the boundary table, which avoids that any distance will be calculated more than once. Despite this optimization, the execution time is still Order N-squared. Unfortunately this matters, given the size of laser altimetry data sets. Segmenting the aula roof (55,133 points and exactly 80,000 triangles into 1789 segments) took already 5 hours and 10 minutes using a C implementation on a 350 MHz Pentium PC under Linux.

5 CONCLUSIONS

Compared to 3-dimensional Hough transform, the most striking advantage of TIN segmentation is its simplicity, also towards the user. The whole process is controlled by only one parameter, the maximum allowable segment distance before merging, which is related to the measurement's standard deviation. At first sight, Hough transform also does not seem very complex, but in its original form it performs badly. This is mostly because Hough transform does not ensure that detected planes are actually connected. Experiments show that TIN segmentation performs at least as well as a heavily modified, rather complex 3-D Hough transform [Goos, 2001], which is controlled by five parameters. Further optimization of the TIN segmentation implementation is required to handle data sets with realistic sizes.

REFERENCES

- [Duda and Hart, 1972] Duda R.O., Hart P.E., Use of the Hough transform to detect lines and curves in pictures. *Comm. of ACM*, 15(1), 1972, pp. 11-15
- [Geibel and Stilla, 2000] Geibel, R. and U. Stilla, Segmentation of laser altimetry data for building reconstruction: different procedures and comparison, *IAPRS*, vol. 33, part B3, pp. 326-334.
- [Goos, 2001] M.I.M. Goos, 2001, Surface identification from irregularly distributed laser altimetry data. M.Sc thesis, Delft University, 69 pp.
- [Maas and Vosselman, 1999] Maas, H.-G. and G. Vosselman, Two algorithms for extracting building models from raw laser altimetry data, *ISPRS J. of Photogrammetry and Remote Sensing*, vol. 54, pp. 153-163.
- [Vosselman, 2000] G. Vosselman, Slope based filtering of laser altimetry data, *IAPRS*, vol. 33, part B3, pp. 935-941.