RECOGNISING STRUCTURE IN LASER SCANNER POINT CLOUDS[1]

G. Vosselman [a], B.G.H. Gorte [b], G. Sithole [b], T. Rabbani [b]

[a] International Institute of Geo-Information Science and Earth Observation (ITC)
P.O. Box 6, 7500 AA Enschede, The Netherlands
vosselman@itc.nl

[b] Delft University of Technology, Faculty of Aerospace Engineering
Department of Earth Observation and Space Systems, P.O. Box 5058, 2600 GB Delft, The Netherlands
{b.g.h.gorte, g.sithole, t.rabbani}@lr.tudelft.nl

KEY WORDS: Laser scanning, LIDAR, Surface, Extraction, Point Cloud, Segmentation

ABSTRACT:

Both airborne and terrestrial laser scanners are used to capture large point clouds of the objects under study. Although for some applications, direct measurements in the point clouds may already suffice, most applications require an automatic processing of the point clouds to extract information on the shape of the recorded objects. This processing often involves the recognition of specific geometric shapes or more general smooth surfaces. This paper reviews several techniques that can be used to recognise such structures in point clouds. Applications in industry, urban planning, water management and forestry document the usefulness of these techniques.

1. INTRODUCTION

The recognition of object surfaces in point clouds often is the first step to extract information from point clouds. Applications like the extraction of the bare Earth surface from airborne laser data, reverse engineering of industrial sites, or the production of 3D city models depend on the success of this first step. Methods for the extraction of surfaces can roughly be divided into two categories: those that segment a point cloud based on criteria like proximity of points and/or similarity of locally estimated surface normals and those that directly estimate surface parameters by clustering and locating maxima in a parameter space. The latter type of methods is more robust, but can only be used for shapes like planes and cylinders that can be described with a few parameters.

This paper gives an overview over different techniques for the extraction of surfaces from point clouds. Section 2 describes various approaches for the segmentation of point clouds into smooth surfaces. In section 3 some variation of these methods are described that result in the extraction of planar surfaces. Section 4 describes clustering methods that can be used for the recognition of specific (parameterised) shapes in point clouds. Section 5 shows that these kind of surface extraction methods can be used in a wide range of applications.

2. EXTRACTION OF SMOOTH SURFACES

Smooth surfaces are often extracted by grouping nearby points that share some property, like the direction of a locally estimated surface normal. In this way a point cloud is segmented into multiple groups of points that represent surfaces. The recognition of surfaces can therefore be considered a point cloud segmentation problem. This problem bears large similarities to the image segmentation problem. Because the point clouds used in early research were captured in a regular grid and stored in raster images, this problem is often also referred to as range image segmentation, which makes the similarity to (grey value) image segmentation even stronger. All point cloud segmentation approaches discussed in this section have their equivalent in the domain of image processing.

2.1 Scan line segmentation

Scan line segmentation first splits each scan line (or row of a range image) into pieces and than merges these scan line segments with segments of adjacent scan lines based on some similarity criterion. Thus it is comparable in strategy to the split-and-merge methods in image segmentation. Jiang and Bunke (1994) describe a scan line segmentation method for polyhedral objects, which assumes that points on a scan line that belong to a planar surface form a straight 3D line segment. Scan lines are recursively divided into straight line segments until the perpendicular distance of points to their corresponding line segment is below some threshold. The merging of the scan line segments is performed in a region growing fashion. A seed region is defined as a triple of line segments on three adjacent scan lines that satisfy conditions with respect to a minimum length, a minimum overlap, and a maximum distance between the neighbouring points on two adjacent scan lines. A surface description is estimated using the points of the seed region. Scan line segments of further adjacent scan lines are merged with the region if the perpendicular distances between the two end points of the additional segment and the estimated surface are below some

[1] This paper was written while the first author was with the Delft University of Technology.

threshold. Finally a minimum region size is required for a successful detection of a surface.

Several variations can be made to the above principle. Sithole and Vosselman (2003) describe a scan line segmentation method that groups points on scan lines based on proximity in 3D. These groups do not need to correspond to a sequence of points in the scan line. In this way, points on either side of an outlier to a surface are still grouped together. The scan line segmentation is repeated for scan lines with different orientations. Artificial scan lines are created by splitting the data set into thin parallel slices of a user specified orientation. Several scan lines sets (3 or 4) with different orientations are segmented (Figure 1). Because all points are present in each scan line set, many points will be part of multiple scan line segments with different orientations. This property is used to merge the scan line segments to regions: scan line segments of different orientations are merged if they share one or more points.

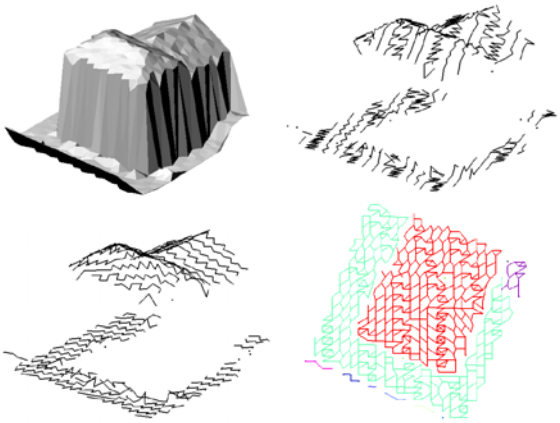

Figure 1: Segmentation of a scene with a building part. Shaded view (top left), segmented scan lines with two different orientations (top right and bottom left) and the result of merging the scan line segments (bottom right).

2.2 Surface growing

Surface growing in point clouds is the equivalent of region growing in images. To apply surface growing, one needs a method to identify seed surfaces and criteria for extending these surfaces to adjacent points. Several variants of this method are described by Hoover et al. (1996).

A brute-force method for seed selection is to fit many planes and analyse residuals. For each point a plane is fit to the points within some distance that point. The points in the plane with the lowest square sum of residuals compose a seed surface if the square sum is below some threshold. This method assumes that there is a part in the dataset where all points within some distance belong to the same surface. Outliers to that surface would lead to a high residual square sum and thus to a failure to detect a seed for that surface. It depends on the application domain whether this smoothness assumption holds. Robust least squares adjustment of planes or Hough transform-like detection of planes (Section 3.1) are more robust methods that would also detect seed surfaces in the presence of outliers.

The growing of surfaces can be based on one or more of the following criteria:

- *Proximity of points*. Only points that are near one of the surface points can be added to the surface. For 2.5 D datasets this proximity can be implemented by checking whether a candidate point is connected to a point of the surface by a (short) edge of a Delaunay triangulation. This condition may, however, be too strict if some outlier points are present. In that case, also other points within some distance of the surface points need to be considered. Otherwise the TIN edge condition may lead to fragmented surfaces.
- *Locally planar*. For this criterion a plane equation is determined for each surface point by fitting a plane through all surface points within some radius around that point. A candidate point is only accepted if the orthogonal distance of the candidate point to the plane associated with the nearest surface point is below some threshold. This threshold and the radius of the neighbourhood used in the estimation of the plane equation determine the smoothness of the resulting surface.
- *Smooth normal vector field*. Another criterion to enforce a smooth surface is to estimate a local surface normal for each point in the point cloud and only accept a candidate point if the angle between its surface normal and the normal at the nearest point of the surface to be grown is below some threshold.

2.3 Connected components in voxel space

It is not uncommon to perform certain steps in processing airborne laser point clouds in the 2-dimensional grid (image) domain. Quite naturally, a point cloud represents a 2.5D surface. When converted to a 2D grid, grid positions are defined by the (x,y) coordinates of the points, and their z-coordinates determine the pixel values. To the resulting regular-grid DSM, operations can be applied that are known from image processing to perform certain analysis functions. Examples are thresholding to distinguish between terrain and e.g. buildings in flat areas, mathematical morphology to filter out vegetation and buildings also in hilly terrain, textural feature extraction to distinguish between trees and buildings, (Oude Elberink and Maas 2000), and region growing to identify planar surfaces (Geibel and Stilla, 2000).

Point clouds obtained by terrestrial laser scanning, however, are truly 3D, especially when recordings from several positions are combined. There is not a single surface that can be modelled by z=f(x,y) as there is in the 2.5D case, and converting such point clouds to a 2D grid would cause a great loss of information.

Recently we adopted the alternative approach of converting a 3D point cloud into the 3-dimensional grid domain. The cells in a 3D grid are small cubes called voxels (volume elements, as opposed to pixels or picture elements in the 2D case). The size of the grid cells determines the resolution of the 3D grid.

Usually, the vast majority of voxels (grid positions) will contain no laser points and get value 0, whereas the others are assigned the value 1, thus creating a binary grid with object and background voxels. A slightly more advanced scheme is to count the number of points that falls into each grid cell and to assign this number as a voxel value.

Similar operations as in 2D image processing can be applied to the 3D voxel spaces. An important class of 2D image processing operators is formed by neighbourhood operators, including filters (convolution, rank order) and morphologic dilation, erosion, opening and closing. In the 3D case, 3D neighbourhoods have to be taken into account, which means that filter kernels and structuring elements become 3 dimensional as well. Roughly speaking, mathematical morphology makes most sense for binary voxel spaces, whereas filters are more useful for 'grey scale' cases, such as density images. The former can be used to shrink and enlarge objects, suppress 'binary' noise, remove small objects, fill holes and gaps in/between larger objects, etc. etc. Application of convolutions and other filters also has a lot of potential, for example for the detection of 3-dimensional linear structures and boundaries between objects.

At a slightly higher level, 2D operations like connected component labelling, distance transform and skeletonisation can be defined, implemented and fruitfully applied in 3D (Palagyi and Kuba 1999, Gorte and Pfeifer 2004).

In all cases, the benefit of voxel spaces, compared to the original point cloud, lies in the implicit notion of adjacency in the former. Note that each voxel has 26 neighbours. Regarding voxels as cubes that fill the space, 6 of the neighbours share a face, 12 share an edge and 8 share a corner with the voxel under consideration.

3. ITERATIVE EXTRACTION OF PLANAR SURFACES

For many applications, the objects under study are known to be polyhedral. In that case, the segmentation algorithms should determine planar surfaces. This can be considered a specific case of smooth surface extraction. Several small variations can be made to the methods described in the previous section which enable the extraction of planar surfaces.

3.1 Plane growing

Both the scan line segmentation (Section 2.1) and the surface growing (Section 2.2) algorithm contain a merging step. This step can easily be adapted to ensure the extraction of planar surfaces.

The scan line method as described by Jiang and Bunke (1994) in fact was originally designed for the extraction of planar surfaces. In the process of grouping the scan line segments (that are linear in 3D space) they demand that the resulting surface is planar. A new scan line segment is only added if the end points of that segment are within some distance of the plane. The plane equation is updated after adding a new scan line segment.

The surface growing algorithm is transformed into a plane growing algorithm if the criterion to enforce local planarity is modified to a global planarity. This is achieved by using all points of the surface in the estimation of the plane equation.

3.2 Merging TIN meshes

In most algorithms, the surfaces are defined as a group of points. Gorte (2002), however, describes a variation in which the triangular meshes of a TIN are the units that compose a surface. Planar surfaces are extracted by merging two surfaces if their plane equations are similar. At the start of the merging process, a planar surface is created for each TIN mesh. Similarity measures are computed for each pair of neighbouring surfaces. Those two surfaces that are most similar are merged and the plane equation is updated. This process continues until there are no more similar adjacent surfaces.

4. DIRECT EXTRACTION OF PARAMETERISED SHAPES

Many man-made objects can be described by shapes like planes, cylinders and spheres. These shapes can be described by only a few parameters. This property allows the extraction of such shapes with robust non-iterative methods that detect clusters in a parameter space.

4.1 Planes

A plane is the most frequent surface shape in man-made objects. In the ideal case of a noiseless point cloud of a plane, all locally estimated surface normals should point in the same direction. However, if the data is noisy or if there is a certain amount of surface roughness (e.g. roof tiles on a roof face), the surface normals may not be of use. The next two paragraphs describes the extraction of planes without and with the usage of surface normals respectively.

4.1.1 3D Hough transform

The 3D Hough transform is an extension of the well-known (2D) Hough transform used for the recognition of lines in imagery (Hough, 1962). Every non-vertical plane can be described by the equation

$$Z = s_x X + s_y Y + d \qquad (1)$$

in which s_x and s_y represent the slope of the plane along the X- and Y-axis respectively and d is the height of the plane at the origin (0, 0). These three plane parameters define the parameter space. Every point (s_x, s_y, d) in this parameter space corresponds to a plane in the object space. Because of the duality of these two spaces, every point (X, Y, Z), according to Equation 1, also defines a plane in the parameter space (Maas and Vosselman 1999, Vosselman and Dijkman, 2001).

The detection of planar surfaces in a point cloud can be performed by mapping all these object points to planes in the parameter space. The parameters of the plane equation in the point cloud defined by the position in the parameter space where most planes intersect. The number of planes that intersect at a point in the parameter space actually equals the number of points in the object space that are located on the plane represented by that point in the parameter space.

For implementation in a computer algorithm, the parameter space, or Hough space, needs to be discreet. The counters of all bins of this 3D parameter space initially are set to zero. Each point of the point cloud is mapped to a plane in the parameter space. For each plane that intersects with a bin, the counter of this bin is increased by one. Thus, after all planes have been mapped to the parameter space, a counter represents the number of planes that intersected the bin. The coordinates of the bin with the highest counter define the parameters of the object plane with the largest amount of points of the point cloud.

These points, however, do not necessarily belong to the one and the same object surface. They may belong to multiple coplanar object surfaces and some points may be even part of object surfaces that only intersect the determined plane. To extract surfaces that correspond to planar object faces, one therefore needs group the points based on proximity. Only groups that exceed some minimum size should be accepted as possible planar object surface.

For the optimal bin size of the parameter space, a balance needs to be determined between the accuracy of the determined parameters on the one hand and the reliability of the maximum detection on the other hand. The smaller the bin size, the more accurate the determination of the plane parameters will be. However, with very small bin sizes, all planes in the parameter size will not intersect with the same bin due to noise in the point cloud. Therefore, the maximum in the Hough space will become less distinct and may not be detectable any longer.

4.1.2 3D Hough transform using normal vectors

If normal vectors can be computed accurately, they can be used to speed up the Hough transformation and to increase the reliability. The position of a point in object space, together with the normal vector, completely defines a plane in object space. Therefore, the parameters of this plane can directly be mapped to a single point in the parameter space. In this way, there is no need to calculate the intersection of a plane in the parameter space with the bins of that space. Only the counter of a single bin needs to be incremented. The availability of the normal vector information will also reduce the risk of detecting spurious object planes.

To reduce the dimension of the parameter space and thereby memory requirements, it is also possible to split the plane detection in to steps: the detection of the plane normal and the detection of the distance of the plane to the origin. In the first step the normal vectors are mapped onto a Gaussian (half) sphere. Because all normal vectors belonging to the same plane should point into the same direction, they should all be mapped to the same position on the Gaussian sphere. This sphere is used as a two-dimensional parameter space. The maximum on the Gaussian sphere defines the most likely direction of the normal vector. This normal vector defines the slopes s_x and s_y of Equation 1. This equation can be used to calculate the remaining parameter d for all points with a normal vector similar to the maximum on the Gaussian sphere. These values d can be mapped to a one-dimensional parameter space. The maximum in this space determines the most likely height of a plane above the origin.

4.2 Cylinders

Cylinders are often encountered in industrial scenes. A cylinder is described by five parameters. Although one could define a five-dimensional parameter space, the number of bins in such a space make the detection of cylinders very time and memory consuming and unreliable. To reduce the dimension of the parameter space, the cylinder detection can also be split into two parts: the detection of the cylinder axis direction (2 parameters) and the detection of a circle in a plane (3 parameters). For this procedure the availability of normal vectors is required.

In the first step, the normal vectors again plotted on the Gaussian sphere. Because all normal vectors on the surface of a cylinder point to the cylinder axis, the Gaussian sphere will show maxima on a big circle. The normal of that circle is the direction of the cylinder axis. Figure 2 shows an industrial scene with a few cylinders. The Gaussian half sphere of this scene shows maxima on several big circles that correspond to the different axis directions of the cylinders. By extracting big circles with high counters along a larger part of the circle, hypothesis for cylinder axis directions are generated.

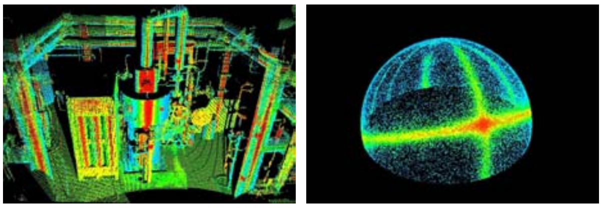

Figure 2: Industrial scene with points colour coded by their surface normal direction (left). Gaussian half sphere with circles corresponding to the dominant cylinder axis directions (right).

All points that belong to a selected big circle are now projected onto a plane perpendicular to the hypothesised cylinder axis. In this plane, the points of a cylinder should be located on a circle. The detection of a circle in a two-dimensional space is a well-known variant to the Hough transformation for line detection (Kimme et al. 1975). The three-dimensional parameter space consists of the two coordinates of the circle centre and the circle radius. Each point is mapped to a cone in the parameter space. I.e. for each radius, each point is mapped to a circle in the parameter space. The circle centre is known to lie on this circle. If usage is made of the normal vector information, the number of bins that need to be incremented can again be reduced. In this case each point can be mapped to two lines in the parameter space (assuming that the sign of the normal vector is unknown). I.e. for each radius and a known normal vector, there are only two possible locations of the circle centre.

4.3 Spheres

A sphere can be detected in a four dimensional parameter space consisting of the three coordinates of the sphere centre and the radius. For each radius, each point in the point cloud defines a sphere in the parameter space on which the sphere centre should be located. Making use of the normal vector information, each point can be mapped to two lines in this four dimensional space. I.e. for each radius and a known normal vector, there are two possible location for the sphere centre.

Alternatively, one can first locate the sphere centre and then determine the radius. Each point and a normal vector define a line along which the sphere centre should be located. In this case the parameter space consist of the coordinates of the circle centre and each point with a normal vector is mapped to a line in the parameter space. Ideally, the lines belonging to the points of the same sphere should intersect in the sphere centre. Once, the sphere centre is located, the determination of the radius is left as a one-dimensional clustering problem, like the determination of the distance of a plane to the origin as discussed in paragraph 4.1.

5. EXAMPLES

In various research projects at the Delft University of Technology, we have been using and developing several of the above point cloud segmentation techniques. Results are presented here where segmentations have been used to model industrial installations, city landscapes, digital elevation models and trees.

5.1 Industrial installations

Three-dimensional CAD models of industrial installations are required for revamping, maintenance information systems, access planning and safety analysis. Industrial installation in general contain a large percentage of relatively simple shapes. Recorded with terrestrial laser scanners the high point density point clouds accurately describe these shapes. Point clouds of such scenes can therefore be processed well with clustering methods as described in section 4.

However, scenes with a large number of different objects may result in parameter spaces that are difficult to interpret. Therefore, large datasets (20 million points) have been segmented using the surface growing technique as described in paragraph 2.2. Using a very local analysis of the surface smoothness, this segmentation will group all points on a series of connected cylinders to one segment. Such segments usually only contain a few cylinders and planes. The point cloud of each segment is then further analysed with the methods for direct recognition of planes and cylinders.

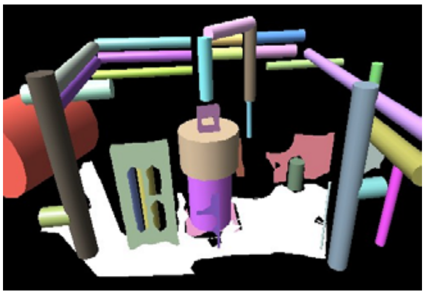

Figure 3: Cylinders and planes in an industrial scene.

Figure 3 shows the result of the automatic modelling of the point cloud shown in Figure 2. A large percentage of the cylinders is recognised automatically.

5.2 City landscapes

Three-dimensional city models are used for urban planning, telecommunications planning, and analysis of noise and air pollution. A comparative study on different algorithms for the extraction of city models from airborne laser scanning data and/or aerial photographs is currently conducted by EuroSDR. Figure 4 shows the results of modelling a part of the city centre of Helsinki from a point cloud with a point density of a few points per square meter.

A large part of the roof planes was detected using the 3D Hough transform (paragraph 4.1.1). For other parts of the roof landscape the operator interactively decided on the shape of the roof to be fitted to the point cloud of a building.

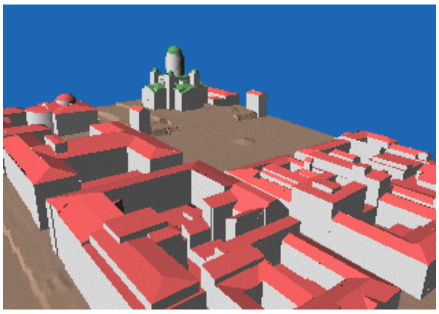

Figure 4: Model of the city centre of Helsinki

The terrain surface was extracted by segmentation of the point cloud into smooth surfaces with the surface growing method (paragraph 2.2). Most terrain points are grouped in one segment. Around the cathedral, the stairs and the plat-form were detected as separate segments. By a few mouse clicks, the operator can specify which segments belong to the terrain. In particular in complex urban landscapes, some interaction is often required to determine the terrain surface.

5.3 Digital elevation models

Digital elevation models are widely used for water management and large infrastructure construction projects. The extraction of digital elevation models from airborne laser scanning data is known as filtering. A large variety of filtering algorithms has been developed in the past years. Sithole and Vosselman (2004) present and experimental comparison of these algorithms. Typical filtering algorithms assume that the point cloud contains one low smooth surface and locally try to determine which points belong to that surface. Most often, they do not segment the point cloud.

Segmentation can, however, also be used for the purpose of the extraction of the terrain surface. Segmenting the point cloud has the advantage that large building structures can be removed completely (something which is difficult for algorithms based on mathematical morphology). Segmentation also offers the possibility to further analyse point clouds and detect specific structures.

The definition of a digital elevation model (DEM) is often dependent on the application. Even within the domain of water management, some tasks require bridges to be removed from the DEM, whereas they should be part of the DEM for other tasks. Bridges, as well as fly-overs and entries of subways or tunnels, are difficult to handle for many filter algorithms. On some sides these objects smoothly connect to the terrain. On other sides, however, they are clearly above or below the surrounding terrain. Often, these objects are partially removed.

Sithole and Vosselman (2003) use the scan line segmentation algorithm with artificial scan lines sets with multiple orientations (paragraph 2.1) to extract objects. Because bridges are connected to the terrain, they are extracted as part of the bare Earth surface. In a second step bridges are extracted from this surface by analysing the height differences at the ends of all scan line segments. Scan lines that cross the bridge will have a segment on the bridge surface that is higher than the two surrounding segments. By connecting those scan line segments that are raised on both sides, objects like bridges and fly-overs are detected. A minimum size condition is applied to avoid the detection of small objects.

Figure 5 (left) shows a scene with a bridge surrounded by dense vegetation and some buildings. The right hand side depicts the extracted bare Earth points and the bridge as a separately recognised object.

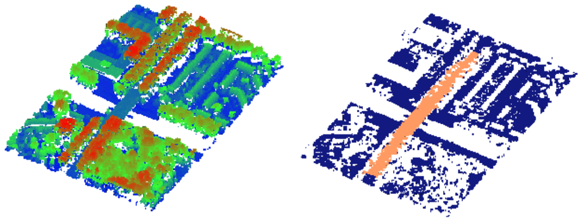

Figure 5: Point cloud with a bridge, dense vegetation and buildings (left). Extracted digital elevation model and bridge (right).

5.4 Trees

Conversion and subsequent processing of terrestrial laser points to a 3D voxel space has been applied recently during a cooperation between Delft University of Technology and the Institute for Forest Growth (IWW) in Freiburg. The purpose was 3D model reconstruction of trees, and to estimate parameters that are relevant to estimate the ecological state, but also the economical value of a (production) forest, such as wood volume and length and straightness of the stem and the branches.

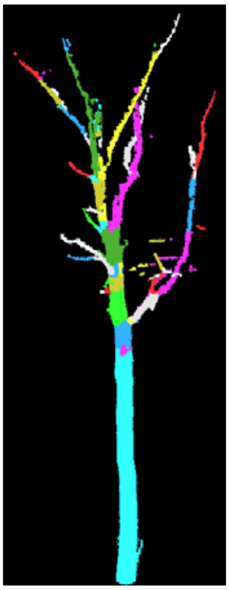

Figure 6: Segmented tree

A crucial phase in the reconstruction process is segmentation of the laser points according to the different branches of the tree. As a result, to each point a label is assigned that is unique for each branch, whereas points are removed (labelled 0) that do not belong to the stem or a significant branch (leafs, twigs, noise).

In terms of voxel spaces, the problem resembles that of recognizing linear structures in 2-dimensional imagery, such as roads in aerial photography. Therefore, we decided to tackle it in a similar manner, i.e. by transferring a variety of image processing operators to the 3D domain, such as erosion, dilation, overlay, skeletonisation, distance transform and connected component labelling. Also Dijkstra's shortest-route algorithm plays an important role during the segmentation (Gorte and Pfeifer, 2004).

The examples of this section demonstrate that segmentation of point clouds is an important step in the modelling of objects. Various segmentation algorithms were discussed and applied. The most suitable segmentation algorithm may depend on the kind of application.

REFERENCES

IAPRS – International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences.

Geibel R. and U. Stilla, 2000. Segmentation of laser-altimeter data for building reconstruction: Comparison of different procedures. *IAPRS*, vol. 33, part B3, Amsterdam, 326-334.

Gorte, B., 2002. Segmentation of TIN-Structured Surface Models. In: *Proceedings Joint International Symposium on Geospatial Theory, Processing and Applications*, on CD-ROM, Ottawa, Canada, 5 p.

Gorte, B.G.H. and N. Pfeifer (2004). Structuring laser-scanned trees using 3D mathematical morphology, *IAPRS*, vol. 35, part B5, pp. 929-933.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A., Fisher, R.B., 1996. An Experimental Comparison of Range Image Segmentation Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (7): 673-689.

Hough, P.V.C., 1962. Method and means for recognizing complex patterns. U. S. Patent 3,069,654.

Jiang, X.Y., Bunke, H., 1994. Fast Segmentation of Range Images into Planar Regions by Scan Line Grouping. *Machine Vision and Applications* 7 (2): 115-122.

Kimme, C., Ballard, D.H., and Sklansky, J., 1975. Finding Circles by an Array of Accumulators. *Communications of the ACM* 18 (2):120-122.

Maas, H.-G., Vosselman, G., 1999. Two Algorithms for Extracting Building Models from Raw Laser Altimetry Data *ISPRS Journal of Photogrammetry and Remote Sensing* 54 (2-3): 153-163.

Oude Elberink, S. and H. Maas, 2000. The use of anisotropic height texture measures for the segmentation of airborne laser scanner data. *IAPRS*, vol. 33, part B3, Amsterdam, pp. 678-684.

Palagyi, K. and A. Kuba, 1999. Directional 3D thinning using 8 subiterations, *LNCS* 1568, Springer Verlag, pp. 325-336.

Sithole, G., Vosselman, G., 2003. Automatic Structure Detection in a Point Cloud of an Urban Landscape. In: *Proceedings 2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas*, URBAN2003, Berlin, Germany, pp. 67-71.

Sithole, G., Vosselman, G., 2004. Experimental Comparison of Filter Algorithms for Bare Earth Extraction from Airborne Laser Scanning Point Clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 59 (1-2): 85-101.

Vosselman, G., Dijkman, S., 2001. 3D Building Model Reconstruction from Point Clouds and Ground Plans. *IAPRS*, vol. 34, part 3/W4, Annapolis, pp. 37-43.