TERRAIN MODELLING AND ANALYSIS USING LASER SCANNER DATA

M. Elmqvist, E. Jungert, F. Lantz, Å. Persson, U.Söderman
FOI (Swedish Defence Research Institute)
P.O. Box 1165, SE-581 11 Linköping, Sweden
{magel, jungert, flantz, asaper, ulfso}@foi.se

Commission III, Working Group 3

KEY WORDS: airborne laser scanning, digital terrain model, ground surface modelling, single tree detection, laser data classification, terrain database, qualitative methods, resolution pyramid.

ABSTRACT

Very detailed high-resolution (3D) digital terrain models can be obtained using airborne laser scanner data. However, laser scanning usually entails huge data sets even for moderate areas, making data management and analysis both complex and time consuming. For this reason, automatic terrain modelling and efficient storage structures supporting data access are needed. In this paper a number of methods supporting automatic construction of 3D digital terrain models, especially ground surface modelling and detection and measurement of individual trees will be discussed. Furthermore automatic and/or interactive terrain feature analysis will be discussed. A special data representation structure for the terrain model allowing efficient data storage and data access will be presented. Beside this, it is possible to create a symbolic information structure from the terrain model that can be used in queries for determination of different terrain features, such as ditches or ridges etc., but also for detection of changes in the terrain.

1 INTRODUCTION

Very detailed high-resolution (3D) digital terrain models can be obtained using airborne laser scanner data. There are many applications requiring such models, both civilian and military. Visual simulation and other types of 3D-visualisations are perhaps the most prominent ones due to the growth of easy accessible powerful 3D-computer graphics hardware. However, there are many other important applications, e.g. urban planning, command and control, mission planning and preparation and various terrain analysis problems.

To support these applications development of new methods and algorithms for automatic terrain modelling, terrain feature analysis and databases are needed. Since data acquisition using airborne laser scanners usually entail huge data sets even for moderate areas it is important that computational efficiency, efficient storage and data access are considered.

This paper consists of two main parts. In the first part we will discuss methods supporting automatic construction of 3D digital terrain models, especially ground surface modelling and detection and measurement of individual trees. In the second part we will discuss terrain feature analysis for high-resolution digital terrain models. The various methods have been developed in order to support a number of specific applications where laser-radar data primarily is used

For modelling of the ground surface a method based on active contours have been developed (Elmqvist, 2001). Metaphorically, a 2.5D contour surface that acts like a sticky rubber cloth or a rubber band net is being pulled upwards from underneath the data set. The net is attracted by the data points and sticks to the points that (are assumed to) represent the true ground. The data points not representing the true ground are not reached due to the elasticity of the net. The resulting contour forms a continuous model of the ground surface. Like many other methods for ground surface modelling (Kraus and Pfeifer 1998, Axelsson, 1999, Pfeifer et al, 1999) the implementation is based on a hierarchic and iterative processing scheme. The active contour method will be presented in section 2.

Given the ground surface model, classification of ground points can be done using the distance between the raw-data points and the estimated ground surface. The remaining raw-data points can be further classified with respect to vegetation and non-vegetation using a recently developed segmentation and classification method. For vegetation data, individual trees can then be identified and tree attributes estimated using a novel method (Persson, 2001). The result can be used for construction of high-resolution 3D synthetic natural environments suitable for 3D-visualisation. Another application not covered here is forest inventory (Hyppä et al, 2000, Hyppä et al, 2001). Classification and tree identification will be discussed in section 2.

For automatic and/or interactive terrain analysis a special terrain model allowing efficient data storage and data access is necessary. For this reason the surface model, in terms of a dense regular grid, is subject to a data reduction process combined with a suitable pattern matching technique (Lantz, 2000). This results in a model with a much sparser grid combined with a set of significant irregular data points. The sparse structure corresponds to a terrain model that almost preserves the resolution without any considerable decrease in accuracy. A data reduction in the order of 90 % has been observed depending on the actual terrain. That is, a flat terrain gives a higher reduction rate than a mountainous area. Of importance is that this data model, with its irregular data points, can be stored in a database using a regular database model. Beside this, it is possible to create a symbolic information structure from the terrain model that can be used in queries for determination of different terrain features, such as ditches or ridges etc., but also for detection of changes in the terrain. The data reduction process, storage structure and terrain analysis is discussed in section 3.

For this work we have used data from the TopEye airborne laser scanner system. This system is mainly operated carried by a helicopter. It contains a vertical scanning direct detection laser radar operating at a wavelength of 1.06um. The pulse rate is between 2 and 7 kHz and the emitted energy is about 0.1 mJ per pulse. The operational altitude is approximately 60-900m. The TopEye system is able to produce point position, intensity of reflection as well as multiple return or double echo data. The laser data used in our work was acquired at missions in 1998, 1999 and 2000. We required dense data sets and hence the mission were flown at slow speed, i.e. 10-25 m/s, and at rather low altitudes, 120-375m. Some areas were also flown in two directions perpendicular to each other. The resulting data sets have a density that varies between 2 - 16 points per square meter.

2 MODELLING AND CLASSIFICATION

2.1 Ground Surface Modelling

For the modelling of the ground surface a new surface estimation method based on active contours has been developed (Elmqvist, 2001). This method is based on the theory on active shape models (Cohen and Cohen 1993, Kass et al, 1998) which has its roots in the area of image processing where it is mainly used for detection of contours in images. Shape models are also referred to as snakes, especially when referring to two-dimensional contours. In such a case the snake is a continuous spline, open with loose ends or closed in a loop. The method for ground surface modelling described here uses a three dimensional active contour in terms of a continuous open surface.

In general, the shape of an active contour is the solution that minimizes an energy function. The function includes internal energy and a potential field. The internal energy is described using physical characteristics associated with the contour, usually material properties like elasticity and rigidity. The potential field is given by the image data, in this case height data. Since an active contour may stick to a local minimum the solution is not always the global minimum.

Metaphorically speaking, the contour used in this case acts like a sticky rubber cloth or a rubber band net that is being pulled upwards from underneath. The net is attracted by the height data points and sticks to points that (are assumed to) represent the true ground. The elasticity forces in the rubber band stops the net from reaching points not representing the true ground. The solution is a net that forms a continuous model of the ground surface. By adjusting various parameters it is possible to achieve different behaviours of the net. For example, if it is preferred that rocks in the terrain are part of the ground surface then the net should be more elastic and sense a greater attraction from the measured points. One example of using this method is illustrated in Figure 1 and Figure 2

The ground surface estimation method based on active contours has been implemented and tested in an experimental set-up. For simplicity and speed of computation the implementation only works on rectangular grid data. It is, however, straightforward to modify the implementation such that it uses the original point cloud and creates a surface in terms of a TIN.

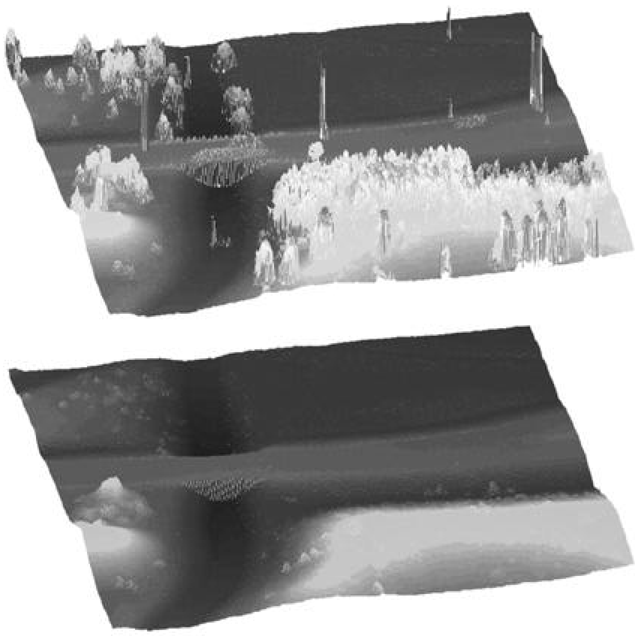

Figure 1 A test area including a road, street lamps, an underpass and a small vegetation area with small pine trees. The post spacing of the grid is 0.33m Top: raw laser data. Bottom: the estimated ground surface

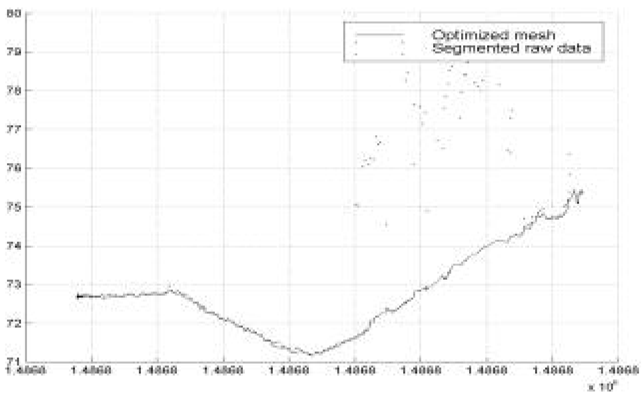

Figure 2 Estimated ground surface for a single laser radar swath. From left: the road, a ditch and a slope with trees.

2.1.1 Implementation

In the experimental implementation the raw data is first resampled in a rectangular grid. The resampling is performed in the easiest way possible. In each mesh the lowest point is selected.

The next step is the optimization of the active contour surface. This process is divided in two phases in which the net is iteratively moved and stretched towards a final solution. In this way a better approximation of the ground surface is achieved. The movement is controlled by a number of "forces" acting on the nodes of the net. In the first phase three different forces are used: elasticity, attraction and gravitation. When the net reaches the convergence criterion of the first phase the second phase starts. In the second phase gravitation is dropped and only elasticity and attraction are used. The iterations continue until the net converges at a final solution. In both phases all the forces are restricted to the z-axes component of a true three-dimensional force vector.

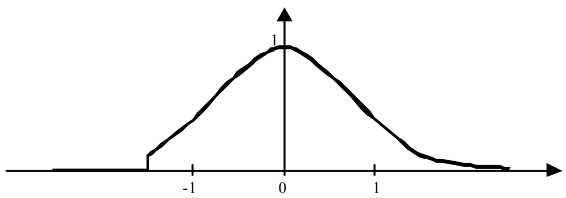

Figure 3 The attraction force as a function of attraction distance. The tail on the left is cut of at a maximum range value to prevent the net to be attracted to points too far away.

The forces are determined for each node as follows:

- Elasticity: an elasticity function is applied to all the connections between the node and its neighbours and the sum is computed. As elasticity function the arctan function is use, hence providing a strongly non-linear force.
- Attraction: an attraction function, see Figure 3**,** is applied to the distance between the node and its corresponding grid point. The force is given a sign such that it always tries to move the node towards the grid point.
- Gravitation: a negative gravitation force.

The start position for the net is set to an elevation below all points in the grid, e.g. one meter below the lowest point. The sign of the final combined force determines if the net should move up or down and the attraction force controls the length of the step. A strong attraction force means a small step; this is to prevent the net to jump past the grid point. In figure 1 one example of using this method is illustrated. Note how the properties of the contour allow the surface to stretch in a steep slope on the sides of the road underpass.

2.2 Classification of Laser Data

After the ground points are classified, the remaining raw-data points can be further classified as vegetation or buildings. Using the maximum height value in each cell, all pixels having an elevation of more than 2 meters above the ground surface are classified. The method is based on texture measures of local differences in height to distinguish artificial surfaces from the natural shape of natural objects (Maas, 1999, Hug 1997). While artificial objects such as buildings consist of continuous, compact surfaces that are bounded by discontinuous edges, natural objects such as vegetation have large vertical variations throughout the objects since the beam can penetrate the canopy of trees.

The measurements used in this method are the second derivative and the maximum slope of each pixel and its eight neighboring pixels. In vegetation, where the height between neighboring pixels varies, the second derivative and slope are larger than within buildings where the change in height of a flat or tiled roof is small. However, the second derivative and the slope are large at edges of buildings and where antennas, chimneys, etc exist. To reduce this noise, the texture measures are median filtered. Based on the two texture measures, each pixel with a height above 2 meters of the ground surface is classified as vegetation and non-vegetation using a maximum likelihood classifier.

Since the texture measures are median filtered, most buildings are correctly classified. Instead some edges of trees are misclassified. To improve the classification result, the smaller areas classified as buildings are checked to see if the area is correctly classified as a smaller building or a part of a tree that is misclassified. The mean value of the second derivative without the median filter and the number of double echoes are calculated for the values inside the boundaries of the areas. Since buildings consist of planar segments, the second derivative is close to zero within the borders of the roofs of buildings. Only at the edges large values occur. In addition, double echoes occur mainly at edges of buildings and in general not within the compact surfaces of roofs. Thus, the mean value of the second derivative and the number of double echoes using only the values inside the borders are small for buildings compared to vegetation. The two mean values are thresholded, and if any of these values are above the threshold, the area is classified as vegetation. Figure 4 shows the classification result over an area of 130x200m.

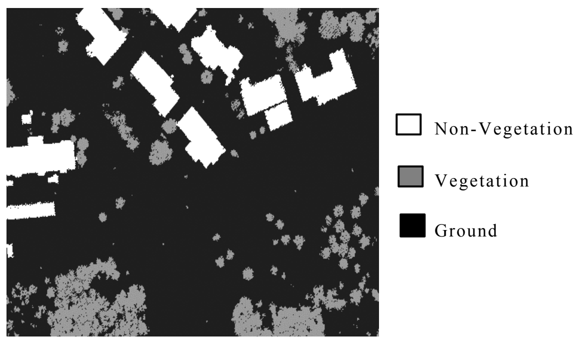

Figure 4. Laser data classification over an area of 130x200m

2.3 Detecting and Estimating attributes of Individual Trees

Using the areas classified as vegetation, individual trees are identified where the position, tree height, and crown diameter of the identified trees are estimated. The method to identify individual trees is based on three steps: 1) create a model of the canopy of trees, 2) smooth the image with different scales, and 3) select the appropriate scale in different parts of the image. The laser beam's ability to penetrate the canopy of trees may result in large variations in height within single trees making it difficult to separate tree crowns from each other. Thus, first the pulses that have penetrated the canopy are removed to create a model of the outer part of the crowns. To remove the penetrations, the same active contour surface that is used to estimate the ground level is applied from above so that the surface follows the outer part of the crowns, see Figure 5

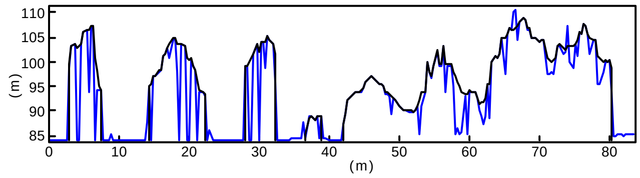

Figure 5 Removing the penetrations in the tree crowns

The process to detect single trees is based on smoothing the image and the location of the trees is estimated by identifying local height maxima. To remove height variations caused by branches within individual tree crowns so each tree has a single height maximum, a certain scale of smoothing should be used depending on the size of the trees. Three different scales are used to smooth the image. The location of the trees is estimated by searching for local height maxima in the smoothed images. Seeds are placed out in every pixel classified as vegetation and let to climb in the direction having the largest slope. When a seed reaches a position where all neighboring pixels have lower values, a local maximum is found. The crown coverage is estimated by grouping those pixels that climbs to the same maximum. The smoothing of the coarsest scale is chosen so that in general no tree has more than one maximum. The finest scale is chosen so that most trees are detected with the effect that some of the larger trees have more than one maximum.

Finally, the segmented areas of trees from the coarser scale are compared with the corresponding area from the finer scale. For cases when the finer scale have detected more than one maximum, the problem is to determine if additional maxima at the finer scale should be judged as separate trees or belong to the treetop detected at the coarser scale. Selection of the appropriate scale in different parts of the image is based on fitting a parabolic surface to the elevation data. Figure 5 shows the crown coverage (a) and the estimated positions of trees marked on the elevation data (b) when a combination of the scales is used.

The height and crown diameter of the detected trees are estimated. For each segment, the maximum height value above the ground surface is chosen as the measure of the tree height. The area of the segments is used to calculate the crown diameter as if the tree crown has the shape of a circle.

A validation of the method has been performed in cooperation with the Swedish University of Agricultural Sciences (SLU). The method was applied to data from at a test site located in southern Sweden where field measurements have been performed. The result shows that most large trees are detected. Most of the undetected trees are hidden trees with a small stem diameter that cannot be seen from above. The mean value of the difference between the estimated position of the detected trees and the field measurements is 0.51 m and the standard deviation is 0.46 m. The height and crown diameter of the detected trees were estimated with a standard error of 0.63 m and 0.61 m, respectively. These results were obtained using an elevation image having a pixel size of 0.33 x 0.33 m.

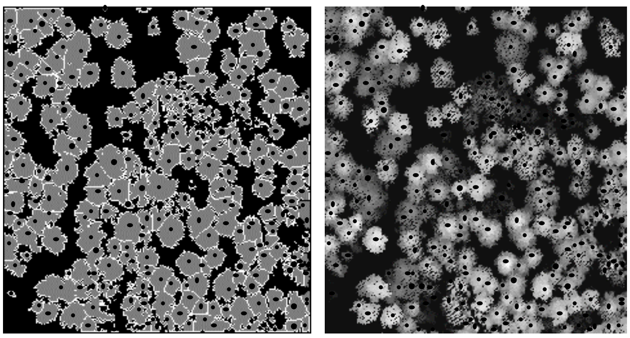

Figure 6. Estimated tree crown coverages and tree positions

2.4 Example of a high-resolution digital terrain model

The methods discussed above have been used for the construction of high-resolution models suitable for real-time 3D visualisation. An example of such a model is shown in Figure 7. This model cover 1 km x 1 km and the ground surface model used has a post spacing of 0.25 m. There are approximately 20 000 trees in the model each having the correct size and position. There are more than 50 buildings which all are reconstructed using a new method which still is under development. In Figure 8 the reconstruction of the seven buildings shown in the upper right corner of Figure 4 is illustrated.

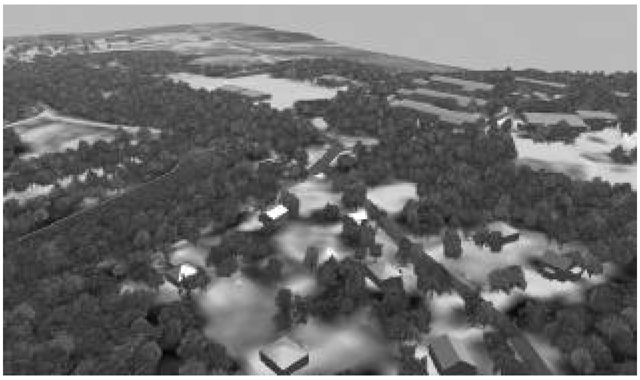

Figure 7. A high-resolution digital terrain model including trees and buildings. Besides the textures all parts of the model are derived from laser scanner data.

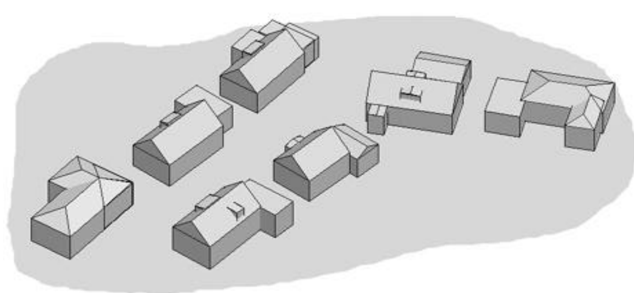

Figure 8. Reconstructed buildings from the area shown in Figure 4

3 TERRAIN ANALYSIS

3.1 Qualitative representation of the terrain in symbolic categories

To represent the terrain qualitatively the surface is partitioned into quadratic *tiles,* with 2 m sides (Lantz, 2000), that will be used as a smallest, atomic modelling element of the terrain. The purpose is to classify the tiles qualitatively into what here is called *categories*. The categories are described in terms of symbolic strings (Chang, 1996, Jungert, 2001). One of the characteristics of this qualitative modelling is that the distinctions made between different modelling elements should be *relevant*, i.e. all distinctive structures should be included without any unnecessary details. Another factor of importance, when distinguishing the characteristics of the categories depends on the uncertainty of the data. It is not appropriate to model distinctions between categories that are too small in relation to the sensor uncertainties. It is also desirable decompose the representation into subgroups that can be accessed independently. Given that a symbolic representation suppresses the unimportant details, this suppression provides data reduction in the sense that the description of the tiles will be more compact. It also reduces complexity as the number of allowed surface forms is reduced. Another advantage is that it enhances the stability of the form interpretation over time, as the distinctions made should be less sensitive to sensor uncertainties.

Which are the relevant distinctions when to query terrain objects, to perform change detection or to visualize? These operations may vary depending on their representation. Therefore, we have chosen to exclude the absolute height of the surfaces, and the magnitude of the inclination of the surface, from our qualitative representation. This does not mean that this information is disregarded; it only means that it is not interpreted qualitatively.

What is an adequate degree of modelling accuracy and what details should be suppressed? This depends on the error tolerance of the application, and on the relation between the size of the modelling elements and the resolution of the original surface. The latter determines the possible change within the tiles and thus how much structure the elements must be able to represent. A simple representation would be to approximate every square with a single plane with some inclination. This is clearly very restrictive, but even with this approximation some properties can be determined, i.e. the plane is totally determined by its normal vector which, for instance, can be split into projections along the z-axis and the xy-plane and qualitatively interpreted.

A slightly more complex approach, which allows considerably more information to be modelled, is to allow the tiles to be to be approximated by two planes. A set of restrictions, when combining the two planes, has been introduced to keep the number of categories at a manageable level. The first restriction is to allow just two types of planes, *flat* or *inclined* (although they will be to inclined in different directions). As have been mentioned, the sensor inaccuracies make it unwise to model too small height differences. Here we have chosen to ignore differences below a given threshold and tiles with less height are thus considered flat. To make qualitative distinctions between different magnitudes of inclinations is difficult. For instance, determining if a vehicle can pass a tile with a certain inclination may be difficult. E.g., a "large inclination" depends on the capacity of that vehicle. Thus, we are left with the distinctions between tiles that are flat and those that have an inclination. Another restriction on the combination of the two planes is that just categories where the edge formed by the intersection of the planes parallel to the xy-plane will be considered. These forms are by no means obvious, but after some considerations the forms described in figure 1 becomes appropriate. Other allowed forms combine planes where one of the planes is flat and cases where the inclinations of the two planes are in opposite directions. Apart from these forms and the flat category, categories with a single inclined plane will also be allowed.

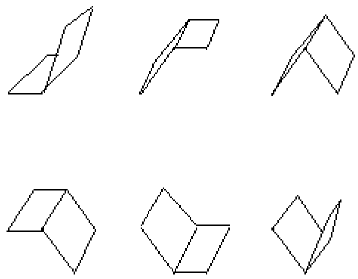

Figure 9. Basic category forms.

After this level of reduction have been reached we still have to qualitatively interpret the different ways to divide the tiles into two planes to decompose a full category description into sub indicators, that can be described in terms of symbolic strings. Of concern is to let all tiles with certain *distinct* divisions between separate planes belong to different categories. In order to determine which divisions that should be considered the tiles are split into subparts, i.e. the *corners*, the *edges* and the *interior,* as seen in Figure 10. Every division that has a start or end-point within different subparts (not considering the interior) are defined as a distinct division. Consequently, all tiles with a start point or an end point within different tile parts belong to different categories. The motivation for this division is its generality and its independence of the maximum resolution of the tile.

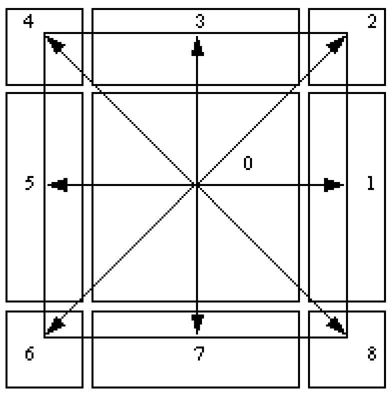

Figure 10. The sub-parts of a tile, their integer encoding and the allowed inclination directions.

As a final restriction to our representation the number of *inclination directions* allowed in a category should be restricted as well. The inclination direction is the projection of the maximal, positive tangent vector of the plane to the xy-plane. All allowed inclination directions can be seen in Figure 10. In some cases, the inclination direction is totally determined by the division of the two planes, while others are ambiguous. In all cases, but for categories determined by a single plane, there are some constraints. However, only one inclination direction for each category will be allowed. The inclination direction closest to the average inclination direction of all possible alternatives will be chosen as a representative direction. Finally, there is another type of category allowed that is not a combination of two planes. These are categories with extreme points. The reason for allowing them is that they are quite common terrain features and they will be difficult to approximate by two planes. In this case, consider two categories as different if they have extreme point in different subparts including the interior part as well. Categories with extreme points in the corners will not be accepted, as that shape is similar to the category with an edge between the borders and with the corner as a common point.

The number of categories can be calculated as follows: there are 16 ways of dividing a tile into distinct planes. Each of these has 6 possible combinations of inclined or flat planes, see Figure 9, which gives 16*6=96 categories. Adding to this is the categories with no feature, which are 9, and those with extreme points, which are 10; giving a total of 115 different categories.

3.2 Category determination and data reduction

The determination of category membership and which grid points to store for visualization and quantitative analysis can be made in many ways. This will not be discussed in this paper as it is subject to ongoing research. However, a method for categorization that is somewhat slow and primarily intended for validation, creates first a canonical representation for each category. For the categories with edges starting and ending in corner points, there is only one choice, but for all others there are more than one candidate. The number of candidates depends on the number of grid points in each category. A natural choice in this case is to use the point in the middle of each sub-part. When a canonical representation has been selected for a tile, we must transform it into a representation that allows comparison with other tiles. Note that all that is required is to find a value for a category in each grid point of the original surface. A suitable representation is then to form a sub grid (matrix) for each category. Apart from the flat category, the categories are independent of both absolute height and relative height. We can consequently choose any values, as long as they form a sampling of the plane that the grid point belongs to. The points in the canonical representations have been given canonical values. Thus the maximal value of every category is set to 1 and the minimal to 0. Then we can transform the height values in each tile to the same value range, using the minimal height and the height difference in every tile to compare with each category. Using some distance measures, for instance the L^1 norm, the comparison can be carried out simply. The result of this process can be seen as the actual *definition* of what it means to a tile to belong to a certain category. The method to determine which points to keep for visualization can also vary. Here we have chosen to keep any point that is a part of the feature, along with the corner points. Thus, there is an a priori, lower bound on the reduction determined by the four corner points, which are absolutely necessary in order to visualize a tile.

3.3 The terrain database structure

The terrain structure with its different categories is basically a grid structure completed with some irregular points. However, the main purpose of this structure is to develop a query structure that can be used for determination of different terrain features and objects represented in 3D the structure must allow objects of different size to be found. That is, small objects like ditches and large objects like canyons must be possible to find. To accomplish this, the terrain structure must be represented in different scales and the method chosen here can be seen as a generalization in 3D of the resolution pyramid [4]. The chosen grid sizes are, beside the original 2m grid, 4, 8 and 16 meters.

The terrain database must efficiently mirror the multi resolution pyramid structure permitting a formal symbolic description of the surface model at all the different resolution levels. The primary purpose of the database structure is to allow access of the symbolic categories, for operations like change detection; given two versions of the same area registered at different times, and for determination of terrain features of special interest. The latter should be determined by means of filters describing the objects; an example of such an object is a ditch. Efficient triangulation of the terrain for visualization purposes should also be possible. Consequently, the terrain data set must include information corresponding to all resolution levels with all the categories as well as their coordinate points and their elevation values. This data set can logically be described in the following terms:

x_g, y_g {the coordinates of the lower left corner of a grid area}
z_g {the elevation of the lower left corner of a grid area}
c_2, c_4, c_8, c_{16} {the category indicators of the different resolution levels}
x_{2e1}, y_{2e1}, z_{2e1} … x_{2ei}, y_{2ei}, z_{2ei} {The supplementary data points of the grid areas at the 2m resolution level including their altitude levels.}
x_{4e1}, y_{4e1}, z_{4e1} … x_{4ej}, y_{4ej}, z_{4ej} {The supplementary data points of the grid area at the 4m resolution level including their altitude values.}
x_{8e1}, y_{8e1}, z_{8e1} … x_{8ek}, y_{8ek}, z_{8ek} {The supplementary data points of the grid area at the 8m resolution level including their altitude values.}
x_{16e1}, y_{16e1}, z_{16e1} … x_{16el}, y_{16el}, z_{16el} {The supplementary data points of the grid area at the 16m resolution level including their altitude values.}

The category indicators of the different resolution levels can be split up with respect to the sub-indicators describing a category, i.e. *inclination, feature* and *orientation*. The complete category value of a grid area is a compound of these sub-indicators.

Given the information above an efficient terrain database structure that corresponds to a single flat file structure can be determined that contains all the coordinate points and their altitude for all resolution levels. Unfortunately, the category indicators cannot be stored in this structure without causing redundancy in the database. The terrain database can thus, due to its simple structure, be accessed e.g. trough a B-tree. However, to allow access of the data points of certain resolution level, the data points of the various resolution levels must be possible to distinguish. This can be accomplished by introducing a variable (*n*) that indicates the resolution level(s) of a certain data point. Since many points belong to more than one level, *n* must have a value that is simple to decode. The solution to this problem is to use a four bit binary structure. Once all the data points of a certain resolution level have been accessed they have to be sorted with respect to their grid area membership. This is a fairly simple operation since it only requires ordering of the points with respect to their grid interval in the x- and y-directions.

The category of a grid area is possible to determine for each resolution level by accessing a simple file where the key elements are the x- and y-coordinates of the lower left corner of the grid area are stored. Considering all these aspects the resulting database structure can thus be described as:

Tdb: x, y, z, n {The terrain database}
C_2DB: x_g, y_g, c_2 {the category database for 2 m grids}
C_4DB: x_g, y_g, c_4 {the category database for 4 m grids}
C_8DB: x_g, y_g, c_8 {the category database for 8 m grids}
C_{16}DB: x_g, y_g, c_{16} {the category database for 16 m grids}

The terrain data is also subject to various research efforts for which a number of applications are in focus. Two main applications can, beside visualization of a triangulated terrain model, be identified. These two applications are *determination of terrain features* and *detection of spatial changes over time*. The principles of these applications will be discussed. Besides this design of a query language [2] concerned with other sensor data types as well and where sensor data fusion will play a fundamental role is also going on. Access of all occurring data points, their elevation and their category types for the 2m grid can be made as follows:

$(x_{lower\text{-}boundary}, y_{lower\text{-}boundary},$
$x_{upper\text{-}boundary}, y_{upper\text{-}boundary}) \Rightarrow$
$\{x_g, y_g\} \Rightarrow C_2DB \Rightarrow \{x_g, y_g, c_2, n\}$
$\{x_g, y_g\} \Rightarrow Tdb \Rightarrow$ {x,y,z-coordinates}
$\Rightarrow$ /triangulate and visualize /

The main operation here is obviously to access a specified area for triangulation and visualization.

Queries for the determination of different object types can basically be described in two steps. In the first step, the grid size most suitable for the requested objects is determined. In the second step object filters that describe these terrain objects or features are matched against the actual sequence of grid area categories of the area of interest (AOI). The most suitable resolution level of the resolution pyramid depends on the size of the objects and may, e.g. for ditches correspond to the 2m grid, whereas for large objects the 16m grid is better. Logically, this matching or filtering process can be described in the following high-level terms:

$(x_{lower\text{-}boundary}, y_{lower\text{-}boundary},$
$x_{upper\text{-}boundary}, y_{upper\text{-}boundary}) \Rightarrow$
$\{x_g, y_g\} \Rightarrow C_jDB \Rightarrow$
$\{x_g, y_g, c_j, n\}$ where $j \in \{2, 4, 8, 16\}$
$Match(\{c_j\}_{feature\text{-}filter}, \{x_g, y_g, c_j\}) \Rightarrow$
$\{x'_g, y'_g\}_{succesful\text{-}match}$

The goal here is to determine the object type filters more or less automatically from a type of formal description, which can be integrated into the query language.

Change detection means that changes of an area made over a period in time should be determined. The principle is to compare the category types of the two versions of the area against each other to determine existing changes, register their positions and eventually determine the types of changes that have occurred. For example, has a wall been built since the last registration. This can be determined by means of the filter technique at the position of change in both the generations of data. This access structure can be described as:

$(x_{lower\text{-}boundary}, y_{lower\text{-}boundary},$
$x_{upper\text{-}boundary}, y_{upper\text{-}boundary}) \Rightarrow$
$\{x_g, y_g\}_{version\text{-}1} \Rightarrow C_jDB \Rightarrow \{x_g, y_g, c_j, n\}_{version\text{-}1}$
where $j \in \{2, 4, 8, 16\}$
$(x_{lower\text{-}boundary}, y_{lower\text{-}boundary},$
$x_{upper\text{-}boundary}, y_{upper\text{-}boundary}) \Rightarrow$
$\{x_g, y_g\}_{version\text{-}2} \Rightarrow C_jDB \Rightarrow \{x_g, y_g, c_j\}_{version\text{-}2}$ where $j \in \{2, 4, 8, 16\}$
$Comp(\{x_g, y_g, c_j\}_{version\text{-}1}, \{x_g, y_g, c_j\}_{version\text{-}2}) \Rightarrow \{x'_g, y'_g\}_{changed}$

This type of matching can be made on all levels of of resolution.

4 CONCLUSIONS

In this paper methods supporting automatic construction of 3D digital terrain models - ground surface modelling and detection and measurement of individual trees - and terrain analysis have been dicussed.

A new method for modelling of the ground surface based on active contours has been presented. This method works well and is robust and creates a continuous model of the ground surface. For classification of ground points and also the remaining data points with respect to vegetation and non-vegetation a recently developed segmentation and classification method have been presented. It has been tested on several types of areas and the result is promising. A novel method for identification of individual trees and tree attribute estimation has also been presented. This method has recently been validated using field measurements and the result is very promising.

The terrain database structure described in the context of terrain analysis is basically concerned with three aspects, i.e. (1) triangulation and visualization, (2) determination of object/terrain features and (3) change detection. Of these three the first one is trivial while the two others are more complex and for this reason they are subject to further research. Furthermore, they are also based on a process of matching of elements described in terms of symbolic 3D surface categories. The advantage of this approach is that a very simple operation including very simple comparisons has been achieved. The results of these research activities are promising and especially in querying the outcome are very interesting.

REFERENCES

[Axelsson, 1999] Axelsson, P., 1999, Processing of laser scanner data – algorithms and applications, ISPRS Journal of Photogrammetry & Remote Sensing 54 (2).

[Ballard, 1982] D. H. Ballard and C. M. Brown, 1982, Computer Vision, Prentice Ha1l, pp 106-111.

[Chang, 2000] S.-K. Chang, G. Costagliola, E. Jungert, 2000, Spatial/Temporal Query Processing for Information Fusion Applications, In Advances in Visual Information Systems, R. Laurini (Ed.), Springer Verlag, Berlin, pp 127-139.

[Chang, 1996] S.-K. Chang and E.Jungert, 1996, Symbolic Projection for Image Information Retrieval and Spatial Reasoning, Academic Press, London, U.K.

[Cohen and Cohen, 1993] Cohen L. D. and I. Cohen 1993. Finite Element Methods for Active Contour Models and balloons for 2D and 3D Images. IEEE Transaction on Pattern Analysis and Machine Intelligence, PAMI-15.

[Elmqvist, 2001] Elmqvist, M., 2001, Ground Estimation of Laser Radar Data using Active Shape Models. Presented at the OEEPE workshop on Airborne Laserscanning and Interferometric SAR for Detailed Digital Elevation Models, Stockholm, Sweden.

[Hug, 1997] Hug, C., 1997, Extracting artificial surface objects from airborne laser scanner data, In A. Gruen, E.P. Baltsavias, O. Henricson, eds, Automatic Extraction of Man-Made Objects from Aerial and Space Images (II), Birkhäuser Verlag, Basel,

[Hyyppä et al, 2000] Hyyppä J., U. Pyysalo, H. Hyyppä, and A. Samberg 2000. Elevation accuracy of laser scanning-derived digital terrain and target models in forest environment. In Proceedings of the 4th EARSeL workshop on LIDAR Remote Sensing of Land and Sea, Dresden,Germany.

[Hyyppä et al, 2001] Hyyppä J., O. Kelle, M. Lehikoinen, and M. Inkinen 2001. A Segmentation-Based Method to Retrieve Stem Volume Estimates from 3-D Tree Height Models Produced by Laser Scanners. IEEE Transaction on Geoscience and Remote Sensing, Vol. 39, No. 5, May 2001, pp 969-975.

[Jungert, 2001] E. Jungert and F. Lantz, 2001, Symbolic Tile CategorieResolution, In Multimedia Databases and Image Communication, M. Tucci (Ed.), Springer Verlag, Berlin, pp 140-151.

[Kass et al, 1998] Kass M., A. Witkin, and D. Terzopoulos 1998. Snakes: active contour models. Int. J. of Computer Vision, 1:321-331.

[Kraus and Pfeifer 1998] Kraus, K., and N. Pfeifer, 1998, Determination of terrain models in wooded areas with airborne scanner data, ISPRS Journal of Photogrammetry and Remote Sens, 53.

[Lantz, 2000] F. Lantz and E. Jungert, 2000, Dual Aspects of of a Mult-Resolution Grid-Based Terrain Data Model with Supplementary Irregular Data Points, Proceedings of the 3rd Int. Conf. on Information fusion, Paris, July 0-13.

[Pfeifer et al, 1999] Pfeifer, N., T. Reiter, C. Briese, and W. Rieger, 1999, Interpolation of high quality ground models from laser scanner data in forested areas in Proceeding of the ISPRS III workshop on Mapping surface structure and topography by airborne and spaceborne lasers, La Jolla, CA.