DETERMINATION OF TERRAIN FEATURES IN A TERRAIN MODEL FROM LASER RADAR DATA

Fredrik Lantz, Erland Jungert, Mats Sjövall
Swedish Defence Research Agency (FOI)
Box 1165, S-581 11 Linköping, Sweden
tel +46 13 37800, fax +46 13 378050
{jungert, flantz}@foi.se

COMMISSION III, WG 3

KEY WORDS: Digital terrain model, laser radar data, terrain features, symbolic structures, matching

ABSTRACT:

Determination of terrain data features in a high resolution terrain data model are required in a large variety of applications where visualization is an important aspect. This is a complex problem because a terrain model is generally a mapping of the reality into a continuous 3D surface model of arbitrary shape, requiring an extremely large amount of data especially when a high resolution is required. Data from such a terrain data model need to be efficiently stored, maintained and visualized. Furthermore reduction of the stored data without loss of relevant information is required. In this work a terrain model that allows the determination of various terrain features through symbolic filters is proposed. The filters are compositions of symbolic surface elements. It will be demonstrated how these filters can be used to obtain objects and object features for visualization and other purposes.

1 INTRODUCTION

Development of digital terrain models has for a long time been subject to extensive research efforts, see e.g. (Maguire 1991). The driving force behind these efforts is a large number of applications, e.g. visualization of the terrain in three dimensions, determination of the line of sight or the areas covered by specific sensors. Two main data structures for representation of digital terrain models can be found. The first one is the gridded model, which is favorable because it is regular but may require the storage of too many dense data point if a high resolution is required. At the other extreme we have the TIN model (triangulated irregular network) for which a fairly large number of approaches has been proposed. The advantage of this structure is that it requires a smaller number of data points than the grid structure given the same resolution. The irregular structure requires more complex algorithms both in generation and analysis and it may occasionally result in inefficient storage structures as well. When the purpose of the terrain model is not only to allow visualization or data analysis but also to query data represented at a very high resolution neither of these two structures is useful. The grid structure, because it will require too many data points and the TIN model because of the complexity of the irregular structure. Besides, neither of these two structures are well suited to applications where matching is an important and necessary ingredient. Consequently, another approach must be taken when such operations are required.

The long term goal in this work is to design a query language based on a type of query-by-example technique that should be able to work on three dimensional data ranging from a very high resolution (≤ 0.5 m) and up to a resolution of, say, 30 m. The upper limit is here determined from the observation that objects with a size of a couple of hundred meters in extension correspond to the maximum size of the requested objects in our applications. The data used here for creation of the terrain model is registered by a scanning airborne laser-radar called TopEye (property of TopEye AB, Sweden). The terrain model can be described as a hybrid between a grid structure and a triangle structure and can simply be described as a regular grid structure augmented with a number of significant irregular data points and lines. The atomic parts of this structure are called a tile. Each such elementary structure can be triangulated in isolation. Another advantage is that these tiles can be stored in a database in accordance to a relatively simple structure (Lantz 2000;Elmqvist 2001b) from which data can be efficiently accessed. A further advantage is that the data structure will allow the generation of a high-resolution terrain model with less data points than an ordinary grid structure for the same resolution; this has been demonstrated in (Lantz 2000). The main advantage of this structure is that it can be transformed into a symbolic structure, which efficiently will allow various types of matching processes. Operations that can be supported by such a structure are e.g. detection of changes over time (change detection) in the terrain and detection of specific terrain objects/features defined by the users as well as a combination of the two. This work is primarily concerned with the development of a *symbolic representation* of the terrain model which will be demonstrated for an application where the purpose is to determine the position/existence of user defined objects; this will be illustrated with a number of examples. Apart from being an appropriate structure for querying, the structure can also be used as an index for a height model and should, as such, be seen as a complement to a height model and not as a replacement. The symbolic structure can also be used for determination of drivable terrain which is outside the scope of this work.

Some related work concerned with techniques for querying terrain data for various features and objects can be found in the literature. Although at this time no symbolic query structure has been found. Despite this, some of the given approaches are quite interesting. In (Bradly 1999) Bradley presents a high-resolution terrain database for a gridded representation covering parts of Mars and where data should be stored in a relational database system. Queries can be written in SQL. The main purpose is to explore the planet and for that reason a fairly large number of terrain features have been determined. Clematis et al. (Clematis 1998) describes a technique for parallel fuzzy queries applied to a spatial data model. A 3D-query space for GIS is described by Fritsch and Schmidt (Fritsch 1994). Their system includes digital terrain data described in terms of a conceptual model called a NIAM-diagram and by also using the entity-relationship model these diagrams can be used to describe occurring object relations in a powerful way. Queries can be formulated in standard SQL. The database is not limited to just terrain data, other types of geo-information is integrated as well. The terrain database includes gridded data and a type of hybrid TIN-structure.

2 THE PROBLEM DOMAIN

The main purpose of this work has been to use laser radar data in developing a technique for symbolic description of the terrain and to create a structure with the following characteristics:

- *The symbolic terrain elements used for description of the terrain should correspond to a classification of a tile.*

This problem corresponds to identifying the members of a class of surface structures, where each member belongs to a particular category. These members can be interpreted in a symbolic way and they will subsequently be called symbolic tiles. Furthermore

- *The symbolic tiles should also be used for identification of different terrain objects or features. This process is carried out by defining a set of symbolic filters. Each such filter should correspond to a composition of a set of connected symbolic tiles. To determine the requested objects a suitable and efficient search strategy must be provided as well.*

Examples of objects that should be possible to determine by means of the filters are, e.g. ditches and fields for landing of helicopters. However, the latter type requires also information about various types of land cover classes such as lakes or forests since a helicopter cannot land on areas covered by such object classes. The terrain must also be available in different resolution levels, which however, will not be discussed further in this paper. As already mentioned, the eventual query language must allow queries where the filters, which may be represented in multiple resolutions, will be used for determination of more or less arbitrary three dimensional terrain objects. The method should thus be applicable to the search for objects of all sizes.

3 SPATIAL CATEGORIES

3.1 The tiles

In order to determine the spatial categories of the terrain, the original sensor data is pre-processed to eliminate non-terrain data, e.g. trees and buildings (Elmqvist 2001a). The pre-processing step produces a surface defined on a rectangular grid with 0.5 m between grid points. Formally, given a compact and connected domain $\Omega \subseteq \Re^2$, a finite number of points $P = \{(x_i, y_i)\} \subseteq \Omega$ and a function $f: P \to \Re$, $\psi(P,f) = \{(x_i, y_i, f(x_i, y_i)) | (x_i, y_i) \in P\}$ is called the sampled surface corresponding to *f*. The purpose of this section can loosely be stated as to find a more suitable surface representation, that can be substituted for $\psi(P, f)$ in subsequent spatial reasoning tasks, e.g. filtering for detection and visualization of certain terrain objects.

The first step in creating such a surface representation is to partition the domain into sub regions and to define a class of sub functions with non-zero only support on those sub regions. Let $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ be a regular partition of Ω into square regions with sides 2 m and let $F_\Gamma = \{f_1, \ldots, f_n\}$ be any set of functions $f_j : \Re^2 \to \Re$ where

- f_j is continuous
- $\int_{\Re^2} f_j(x, y) dx dy = 0$
- $f_j(x,y)=0 \ \forall (x, y) \notin \gamma_j$

A pair (γ_j, f_j) is called a tile and the set of all tiles over Ω is *T*. A surface model consisting of tiles will be determined by substituting every (sampled) sub surface $\psi_j = \{(x, y, f(x, y)) | (x, y) \in \gamma_j \cap P\}$, *j=1,...,n* by a particular tile, chosen from a carefully selected set. The definition of that set will be presented in the following sections.

3.2 Overview of the process

The next step is to define a number of categories of tiles, i.e. the spatial categories. A spatial category is a set of tiles with similar features, e.g. with similar inclination or with edges at similar locations. The definition of the categories is carried out in three steps. Firstly, a set of representative tiles $REP \subseteq T$ is determined. Secondly, a suitable distance metric *dist:* $T \times T \to \Re_+$ for comparing the similarity of the tiles is needed. Finally, a category [r], $r \in REP$ can be defined, using the metric *dist*, as the set of all tiles that are more similar to *r* than to any other $k \in REP$. In practise, the categories are never calculated, only their representatives. After definition of the categories, every representative (and thus every category) is given a partially symbolic interpretation by a unique string, called a symbolic tile. Formally, this is done by a Symbolic Interpretation mapping *SI:* $REP \to M_\Gamma \times A \times B \times A \times A \times \Re_+$, where *A={0,...,8}*, *B={0,1,2}* and $M_\Gamma = \{(u_j,v_j)\}$, is the set of centre points of γ_j, *j=1,...,n*. A sub surface ψ_j is compared with every representative on $\gamma_j \cap P$ by first subtracting the mean of *f* over γ_j and then using a discrete counterpart of the above distance metric. The best match r_k is selected and ψ_j is substituted by the string *s* such that $SI(r_k)=s$.

3.3 Defining the representatives

The complete, formal definition of the representatives is beyond the scope of this work. Instead, an informal but more easily accessible presentation will be given. Three different representations are necessary to achieve a sufficiently accurate surface model. The types respectively describe sub-surfaces using:

1. Two linear functions with non-zero support on separate parts of the square, but where their composition is continuous.
2. One single linear function.
3. A function with a single extreme point in a specified location.

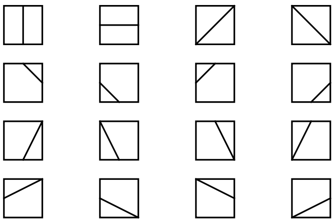


Figure 1. The 16 allowed partitions of a square region.

To describe the first type, consider the 16 partitions of any square γ_j into two parts shown in figure 1. Every representative of this type is completely determined by one of these defining partitions, together with a certain combination of linear functions g_j and h_j. In this work, there are *nine* such combinations for each partition. The functions g_j and h_j are chosen such that their composition is continuous and the sub surface associated with a part has a direction of the inclination (in the xy-plane) among those shown in figure 2. Only three directions are allowed for a partition, zero inclination and the two opposite inclinations that are perpendicular to the direction of the partition edge. The allowed directions for two partitions are shown in figure 3. The representatives can be quite similar, only differing in the location of the edge that partitions the square. In fact, all representatives of this type can be seen as variations of the basic category forms, shown in figure 4, rotated and translated to “fit” the partition. The second type of representative can be seen as a sub type of the first. This can be seen by considering the case where the two functions have exactly the same partial derivatives, e.g. as for the top left tile in figure 3. Note that some representatives are given by more than one partition, e.g. the flat square is given by every partition.

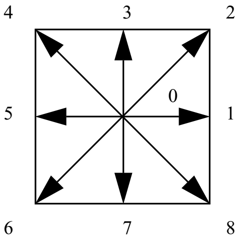


Figure 2. The allowed values for the partial derivatives for a sub function, displayed as vectors seen from the centre of the square. Also shown are the names of the points defining an inclination and a partition (at the head of the arrow).

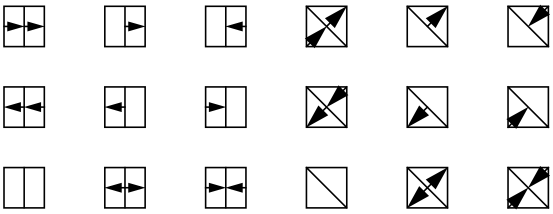


Figure 3. The allowed representatives for two partitions. An arrow indicates the inclination direction for that part. No arrow means the part is flat.

A common terrain feature that is poorly approximated by the above defined representatives is extreme points at other locations than the corners. Therefore ten additional representatives, five minima and five maxima, are allowed with extreme points at the middle of the four borders of the square and at the middle point of the entire square. All together this makes 115 different categories; 96 of type 1, 9 of type 2 and 10 of type 3.

3.4 The categories

A general distance metric for comparing similarity between tiles $(\gamma_j, k_j), (\gamma_l, k_l) \in T$ is:

$$dist((\gamma_j, k_i), (\gamma_l, k_l)) = \|(k_j / \|k_j\|) - (k_l / \|k_l\|)\| \quad .$$

This metric will determine different categories depending on which norm that is used. For the examples in the following sections, the norm used is the integral over absolute value.

At last, the category determined by $r \in REP$ can be defined as $[r] = \{ k | k \in T \wedge ((\forall (r_l \in REP))(dist(k, r) \leq dist(k, r_l))) \}$. As mentioned above, there is no need to calculate the categories, but only the representatives. The calculation of which category a certain ψ_j belong to can be carried out straightforwardly by computing the distance to every representative tile. One important exception to this rule is the representative with partial derivatives all zero, i.e. the totally flat representative. To compensate for the fact that tiles with uninteresting, small height differences will be given undue consideration, all sub surfaces with norm below some pre-determined threshold will be considered a member of the flat category.

3.5 A symbolic interpretation

The symbolic interpretation SI is a composition of five sub mappings, *Position*: $REP \rightarrow M_\Gamma$, *Inclination:* $REP \rightarrow A$, *Featurestate:* $REP \rightarrow B$, *Featureorientation*: $REP \rightarrow A \times A$ and *Norm:* $REP \rightarrow \Re_+$. Position is defined as $Position(\gamma_i , f_i)$= (u_i,v_i) and $Norm(\gamma_i, f_i) = \|f_i\|$. Featureorientation is an encoding of the partition of the tile in the case the tile is of type 1. A particular partition is identified by the pair (*a,b*) of points on the border of the square that it divides. The points are named as shown in figure 3. For a tile of type 3, Featureorientation map the tile to (*a,a*) if *a* is the location of the extreme point. For type 2, the encoding is irrelevant. Inclination is an encoding of the average inclination direction of the tile. As in the case of Featureorientation, the set *A* is used to identify an inclination direction. Naturally, if two sub regions are inclined in opposite directions the average inclination is zero. Featurestate encode a deviation from Inclination and is used to differentiate between tiles with the same average inclination. The deviation can be positive, 1, negative, 2 or zero, 0. In figure 4 the top row corresponds to featurestate(r) = 1 and the bottom row to 2.

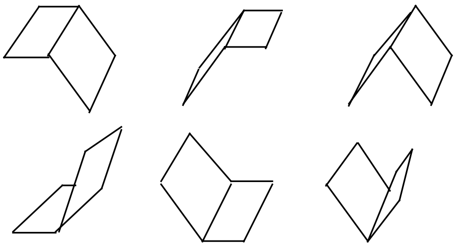


Figure 4. Basic category forms.

4 FILTER STRUCTURES

4.1 Finding segments

The filter matching technique is linear and the filters correspond to a sequence of connected symbolic tiles describing the feature that will be subject to the search. The search is carried out by applying the filters across the given area of interest (AOI), i.e. the given terrain model, to find the requested terrain features. The approach taken here is to use the specific and characteristic cross-sections, which exist for all terrain formations. For instance, a hill has always slopes going either upwards or downwards depending on the position of the viewer. Thus any cross-section of a hill can first be described with an upward directed slope followed by a slope directed downwards. A ditch can be described with an opposite structure. At a conceptual level the cross-section of a terrain formation consist of a start segment, a possible middle segment and an end segment, see figure 5.

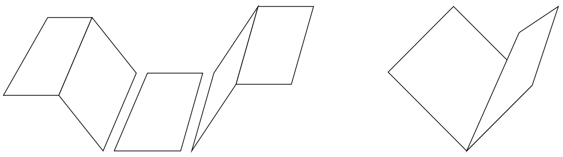
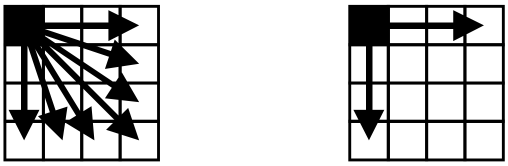


Figure 5. The filter structure to be applied to the cross-section of a ditch

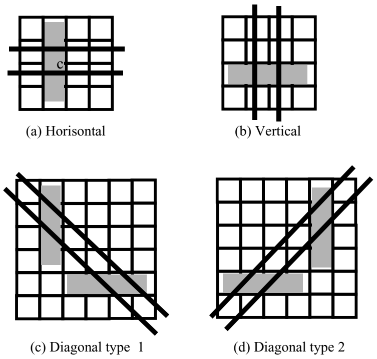
The filter can be of different size depending on the type and size of the requested terrain feature. Consequently, the search strategy must be sufficiently flexible to accommodate these demands. It turns out that a two level search strategy is the most appropriate. The simplest high level search, i.e. tile indexing can be based on row-ordering (Worboys 1995). This type of search algorithm explores the tile structure row by row, from top to bottom, and column by column, from left to right, see figure 6.



Exhaustive search
(a)

Simplified search
(b)

Figure 6. All possible search directions of a 4 x 4 filter (a) and the orthogonal search directions of a simplified search (b).



(a) Horisontal

(b) Vertical

(c) Diagonal type 1

(d) Diagonal type 2

Figure 7. Approximate orientation of long thin objects (bold lines at the edge of the object). Filters applied in two directions (grey boxes) will find objects in all orientations.

By applying the simplified search in horizontal and vertical directions the two filters will find the objects corresponding to all possible orientations. Intuitively it seems like two more filters may be necessary to find the diagonal cases but as can be seen in figure 7, this is not the case. Some minor problems may arise when the simplified search is used. The filters cannot be given a width equal to the maximum width of the terrain formation if the terrain structure has an orientation that is diagonal. The width of the filter must in such a case be somewhat larger than the actual terrain formation. An approximation that is certain to find the formations in these cases is the sum of the width of the horizontal and the vertical filters, i.e. a choice that resembles the triangle inequality. Another obvious consideration in the search is that the filters should not be applied such that any part goes outside the edge of the AOI.

As a result of the first matching step a large number of segments will be found. To find out which of these segments that are part of the feature that is searched for adjacent segments must first be connected. To complete this process, two different approaches have been developed.

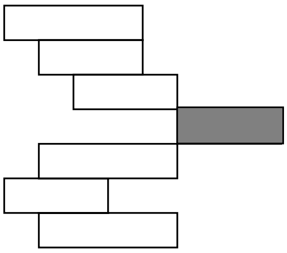


Figure 8. Illustration of the vertical segment search.

4.2 Connecting segments

To connect adjacent segments into objects a search is applied from top to bottom and from left to right. Segments found in step 1 (section 4.1) are connected only if the segments horizontally or vertically overlap each other as can be seen in figure 8 where the white segments will be connected into two different object. The position of the shaded segment prohibits further connections.

4.3 Edges and inclination connections

The algorithm described above is not concerned with the underlying symbolic structure; it operates only on the tiles filtered out in stage 1. A more powerful technique is to take the details of the shape of the filtered tiles into account. To do this we first have to find adjacent tiles which are chosen with respect to some given constraints in the search process. To find tiles that we consider to belong to the same feature, we must also determine whether different existing edges and inclinations proceed into the neighboring tiles. In the case of a tile with both an inclination and an edge the edge takes precedence over the inclination. In each step of the search process a check is made to make sure that the current edge matches the previous edge. To expect an exact match of the edges is in most cases unrealistic so a tolerance of +/- 45 degrees will be allowed. If the check is successful the search direction will be strictly determined by the direction of the edge. Only one search direction will be chosen unless the edge divides the tile asymmetrically, in which case there are two search directions. The chosen directions are depending on the angle of the edge. A case where the tile contains an edge is illustrated in figure 9 by a category with an edge running from point "5" to point "8" as specified in figure 1. Some special cases exist here which are discussed further in (Sjövall 2002).

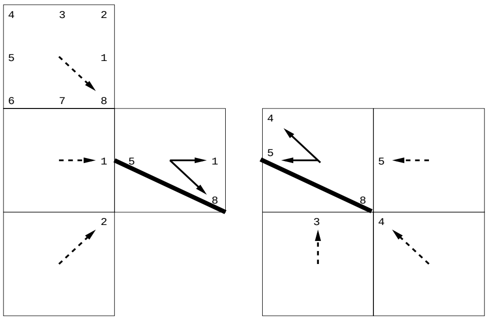


Figure 9. Illustration of the search method in the edge connection algorithm.

5 FILTER SPECIFICATION

The filters are specified in a plain text file that is parsed by a Java-program and applied to the symbolic tile data file. Each segment will first be searched for horizontally then all categories in the filter-specification are rotated 90 degrees and the search continues vertically. All filter lengths and widths are specified in terms of multiples of symbolic tiles. A filter segment as defined in section 4 starts with a *beginsegment* statement and ends with an *endsegment*. Everything between these statements represents the string that is to be searched for. An optional parameter for the segment is invert that will invert the whole segment, i.e. a positive state will become a negative state etc. This can, for instance, be used to quickly transform a ditch-filter into a ridge-filter and vice-versa. The fact that all segments need to be applied both vertically and horizontally requires special attention. Thus the statements *horizontal* and *vertical* specifies the actual search directions.

A segment may contain several sub segments surrounded by *beginsubsegment* and *endsubsegment*. The *beginsubsegment* statement takes three parameters. The first two are mandatory: *minimum length* and *maximum length*, which is the number of consecutive symbolic tiles that should used in this sub-segment. The third is the optional keyword *exclude* which means that this sub segment should not be part of the final segment although it still has to match. The sub segments may also contain the statement *maxerrors* that has a parameter that is the allowed maximum number of not matched tiles. If this is not specified, no errors are allowed. The sub segments contain a set of allowed symbolic tiles which are specified with the statements *begincategory* and *endcategory*. A symbolic tile is specified by the statements *inclination*, *state* and *orientation*. The keyword not is used to tell which values that are not be permitted. The keyword any will allow all possible values.

After the segment-specifications, the post-processing algorithms can be specified. The statement *connect* will connect tiles using the algorithm specified as the first parameter. Allowed algorithms are *edge* and *segment*. Parameters two to five is minimum/maximum length and width of the final object. These parameters are required but by using a value of -1 the checks can be disabled. If the segment algorithm is used a maximum error can be set as an additional parameter. The edge algorithm is implemented as a recursive algorithm which could lead to problems if the recursion depth is too great. Although this should not happen unless a filter is specified for very large objects, in which case the resolution of the tiles should be chosen differently. The segment algorithm first searches from top to bottom followed by a search from left to right. This means that the algorithm only handles objects that are laid out in a straight line. Objects that changes direction and are curved will not be found. The statement *h-and-v* uses the logical-AND algorithm to combine horizontal and vertical segments. This statement does not require any parameters.The statement *findrectangle* will find rectangles of a specified size. The first parameter is the *width*, the second the *height* and the third set the allowed relative error. An example of a filter with just a single sub segment is

```
beginsegment            /*Start the segment*/
  invert                /*Invert the segment, i.e.
                           the real inclinations to be
                           searched for is 2, 3 and 4.*/
  beginsubsegment 1 3
                        /*Start a subsegment with a
                          minimum length of 1 and a
                          maximum length of 3.*/
    begincategory       /*Start category*/
         inclinations 6,7 and 8
                        /* Allow inclina-
                          tions 6,7 and 8*/
      state any         /*Allow any state*/
      orientation not 14 15 16
                        /*Allow all orientations
                          except 14, 15 and 16.*/
    endcategory
  endsubsegment
endsegment
connect edge 10 -1 -1 -1
                        /*Connect categories using
                          the edge-algorithm and a
                          minimum length of 10*/
```

The complete segment will have a possible length of 1 to 3 categories (the minimum and maximum length of the subsegment.

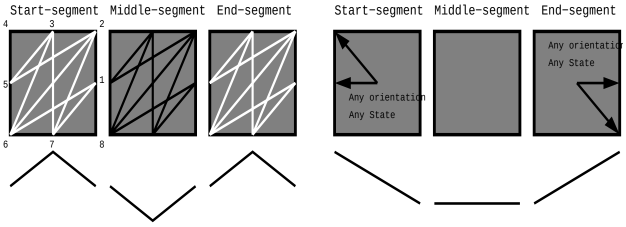


Figure 10. Categories allowed in the ditch-segment.

6 SOME ILLUSTRATIONS

A number of filter types for different features have been developed and tested. Among them can filters for determination of ditches, ridges, hill tops, ponds and flat (sub)- areas including roads be mentioned. We will here give some illustrations of the application of these filters including some of the short-comings.

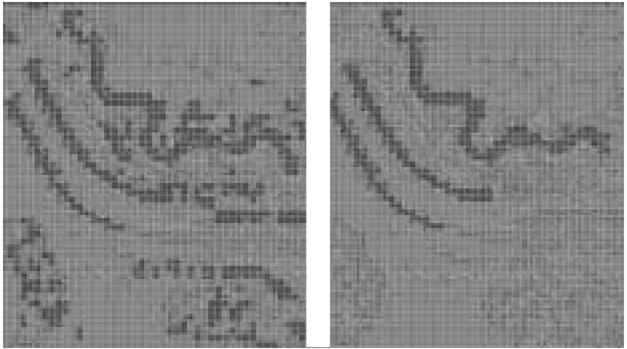


Figure 11. The result of the application of the ditch filter to the test area(left).The final result of the ditch filter as completed with the edge-connect algorithm (right).

6.1 Ditches

This filter finds narrow ditches and is also very tolerant to changes in direction of the ditch. The specification contains three different segment-types:

- The first is a single flat tile with a negative edge (i.e. all orientations except extreme points), see figure 5 at right.
- The second is a segment with a down-slope, flat and up-slope structure. This part finds segments that are part of a ditch that has a northeast-southwest orientation. Figure 10 shows all possible tiles for each sub segment and since the north-south (*3 to 7*) edge as well as the inclinations *1* and *5* are part of the start and end sub-segment, segments with a north-south direction will also be found, see figure 5 at left.
- The third segment is similar to the second, only the difference in the orientation, which is northwest-southeast.

Figure 12. The small stream in real-life.

These three segment types are sufficient to find ditches oriented in all directions since the filters are also rotated 90 degrees. The minimum width of a ditch-segment is one tile and the maximum width is three tiles. Figure 11 shows to the left the result of the segment filter where a lot of false hit segments occur as well.

The segments are then connected using the connect-edge algorithm with a minimum allowed length of 10 (i.e. 10x2 meters). This will remove most of the smaller object hits that may not correspond to ditches. The results can be seen at right in figure 11 where three ditches can be seen. Two of these are in reality ditches along a road and the third a small stream, see figure 12.

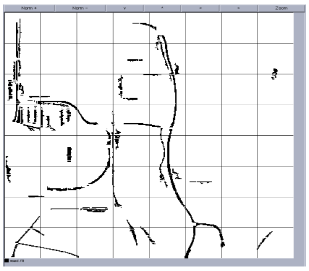


Figure 13. The result of the road-filter

6.2 Roads

The principle of the road filter is to find segments that start and end with a tile that is not flat and with several flat tiles in between. The enclosing categories should not be part of the segment; this is done by applying the *exclude* parameter in the *beginsubsegment* statement to the non-flat categories. The only segments that will remain are those including just flat tiles. These segments become connected by using the connect-segment algorithm. A minimum length of 10 is used with an allowed error frequency of 20%. The filter is applied to the complete data set and the result is shown in figure 13. Several shorter and isolated road segments can be seen in several places and is most likely flat areas where houses were removed by the ground-segmentation. However, this should not be considered a serious problem since it is possible to mark these areas in the removal process. Furthermore, the connect segment algorithm does not handle changes in the direction of the elevation well, thus some parts of the roads are missing here and there.

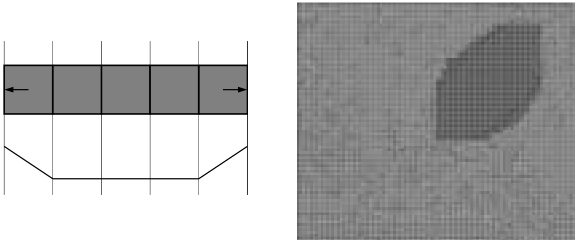


Figure 14. The blob segment and its cross-section at left.The resulting pond using the blob-filter to the right.

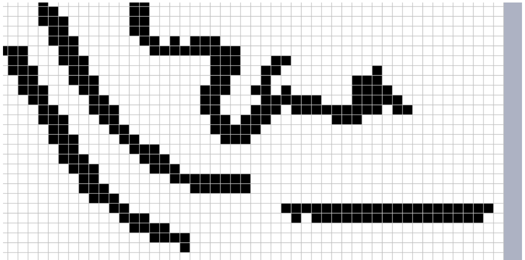


Figure 15. Using the minimum threshold for the ditch-filter. Compared to figure 11, one more terrain object appear at lower right.

6.3 Blobs

This filter will find a smaller blob-like object, e.g. a pond. This is done by looking at the cross-section which ideally will look like the illustration in figure 14. Since a long flat segment is bound to have errors in it, a certain error tolerance is used. To make sure the object is "closed" the cross-section has to be found both horizontally and vertically by using the post-processing filter h-and-v. The edge-connection algorithm is then applied to allow a minimum length and width of 15 tiles. The final result can be seen in figure 14.

6.4 Objects with different inclinations

The above described technique of finding terrain objects identified by their cross sections can not distinguish between tiles with different magnitude of inclination. Nevertheless, the symbolic tiles carry all necessary information to do so in the norm of a tile. In the current version, the previously mentioned filters can be added by a minimum threshold value used to set all tiles with a norm below the threshold as flat. The threshold can be changed dynamically as specified by a user request. The consequence of using a different threshold when searching for ditches can be seen in figure 15, where a curb is detected. This represents a significant improvement over the results in. (Jungert 2002) where cross sections with as small height differences as, in this case, 1.5 dm could not be detected.

7 Conclusions and future work

In this work, techniques that can be used for identification of different types of terrain object features have been described. The technique can be split into two different steps where the first step concerns development of a method for the description of the terrain by means of symbolic tiles. The second step includes methods for determination and connection of non-attached segments.

Future work must include improvement of the filters and further types of terrain-objects must be possible to identify as well. By using the formal filter-specification it is possible to more adequately and correctly describe the terrain-objects which will be the foundation for further work on the development of a query-language.

References

D. J. Maguire, M. F. Goodchild and D. W. Rind (Eds.), 1991, Geographical Information Systems, Longman Scientific and Technical, London, vol. 1, pp. 269-297.

F. Lantz, E. Jungert, 2000, Dual aspects of a Multi-Resolution Grid-Based Terrain Data Model with Supplementary Irregular Data Points, In: *Proceedings of the 3rd International Conference on Information Fusion*, Paris, France, July 10-13.

J. Bradley, 1999, An Efficient Modularized Database Structure for a High-resolution Column-gridded Mars Global Terrain Database, *Journal of Software - Practice and Experience*, vol. 29, no 5, pp. 437-456.

A. Clematis, A. Coda, M. Spagnuolo, S. Spinollo, T. Sloan, 1998, Parallelising Fuzzy Queries for Spatial Data Modelling on a Cray T3D, In: *Proc. of the 4th Int. Workshop on Applied Parallel Computing, Large Scale Scientific and Industrial Problems (PARA'98),* pp. 76-81.

D. Fritsch, D. Schmidt, 1994, DTM integration and three-dimensional query spaces in geographic information systems, In: *Proceedings of the SPIE - The International Society for Optical Engineering vol. 2357, pt 1, pp.* 235-242.

M. Elmqvist, 2001a, Ground Estimation of Laser Radar Data using Active Shape Models. In: *OEEPE workshop on Airborne Laser-scanning and Interferometric SAR for Detailed Digital Elevation Models,* Stockholm, Sweden.

M. Worboys, 1995, GIS A Computing Perspective, Taylor and Francis, London.

M. Elmqvist, E. Jungert, F. Lantz, Å. Persson, U. Söderman, 2001b, Terrain Modelling and Analysis Using Laser Scanner Data, In: *Proceedings of the Workshop on Land Surface Mapping and Characterization Using Laser Altimetry*, Annapolis, Maryland, October 22-24.

M. Sjövall, 2002, Object and feature recognition in a Digital Terrain Modell, Master thesis, University of Linköping, Sweden, LITH-IDA-Ex-02/16.

E. Jungert, F. Lantz, M. Sjövall,2002, Determination of Terrain Features in a Terrain Data Model Using Symbolic Filters, In: *Proc of the 8th int conf on Distributed Multimedia Systems*, San Francisco, CA, Sept 26-28, pp. 664-667.