Generalization of integrated terrain elevation and 2D object models

J.E. Stoter[1], F. Penninga[2] and P.J.M. van Oosterom[2]

[1] International Institute for Geo-Information Science and Earth Observation (ITC), Department of Geo-Information Processing, Hengelosestraat 99, Enschede, the Netherlands
[2] Delft University of Technology, OTB, section GIS technology, Jaffalaan 9, 2628 BX Delft, the Netherlands

Abstract

A lot of attention has been paid to generalization (filtering) of Digital Elevation Models (DEMs) and the same is true for generalization of 2D object models (e.g. topographic or land use data). In addition there is a tendency to integrate DEMs with classified real-world objects or features, the result is sometimes called a Digital Terrain Model (DTM). However, there has not been much research on the generalization of these integrated elevation and object models. This paper describes a four step procedure. The first two steps have been implemented and tested with real world data (laser elevation point clouds and cadastral parcels). These tests have yielded promising results as will be shown in this paper.

1 Introduction

There is a close relationship between Digital Elevation Models (DEMs, 2.5D), based on for example raw laser-altimetry point data, and the topographic objects or features embedded in the terrain. Feature extraction techniques aim to obtain the 2D geometry and heights for certain types of topographic objects such as buildings. There are methods for object recognition in TINs (Triangular Irregular Networks) based on point clouds in which the selection of an object (e.g. building roofs, flat terrain between buildings) corresponds to planar surfaces (Gorte, 2002). This technique can be used for 3D building reconstruction from laser altimetry. However, this will not be the topic of this paper.

On the other hand, 2D objects, from another independent source, such as a cadastral or topographic map, can explicitly be incorporated as part of the TIN structure, which is representing a height surface (Lenk, 2001; Stoter and Gorte, 2003). In this case the TIN structure is based on both 2D objects and point heights. The data structure of the planar partition of 2D objects is within the TIN. Within this data structure, the 2D objects are identifiable in the TIN and are obtainable from the TIN, as a selection of triangles which yield 2.5D surfaces of individual 2D objects. This is the topic of this paper. For the study described in this paper, the following two data sets have been used (see figure 1).

Terrain height points

For the terrain elevation model we use a data set representing the DEM (Digital Elevation Model) of the Netherlands, i.e. AHN (Actueel Hoogtebestand Nederland) (Van Heerd, 2000). The AHN is a data set of point heights obtained with laser altimetry with a density of at least one point per 16 square meters and in forests a density of at least one point per 36 square meters. The point heights are resampled in a regular tessellation at a resolution of 5 meters. Due to availability issues this regular data set is used, whereas the TIN approach is designed with irregular data in mind. The AHN contains only earth surface points: information such as houses, cars and vegetation has been filtered out of the AHN. The heights in the AHN have a systematic error of on average 5 cm and 15 cm RMSE.

Parcel boundaries

The used parcels are from the cadastral database of the Netherlands. In the cadastral database parcel boundaries are organised in a structure of geometrical primitives (boundaries or edges described by their polylines) and parcels are topologically stored via references to boundaries (Lemmen et al., 1998). The typical geometric accuracy is about 10 cm.

For this research four different types of TINs (Triangular Irregular Networks) were generated, all representing surface height models based on point heights obtained from laser altimetry, and the last three also including 2D parcels: unconstrained Delaunay TIN, constrained Delaunay TIN, conforming Delaunay TIN and refined constrained TIN. In section 2 the definition and creation of these TINs is described, together with their results when applied to the test data set. The TINs are stored in the Oracle DBMS, and from this information, some spatial analyses, queries, and visualisation were performed in the context of the DBMS (section 3).

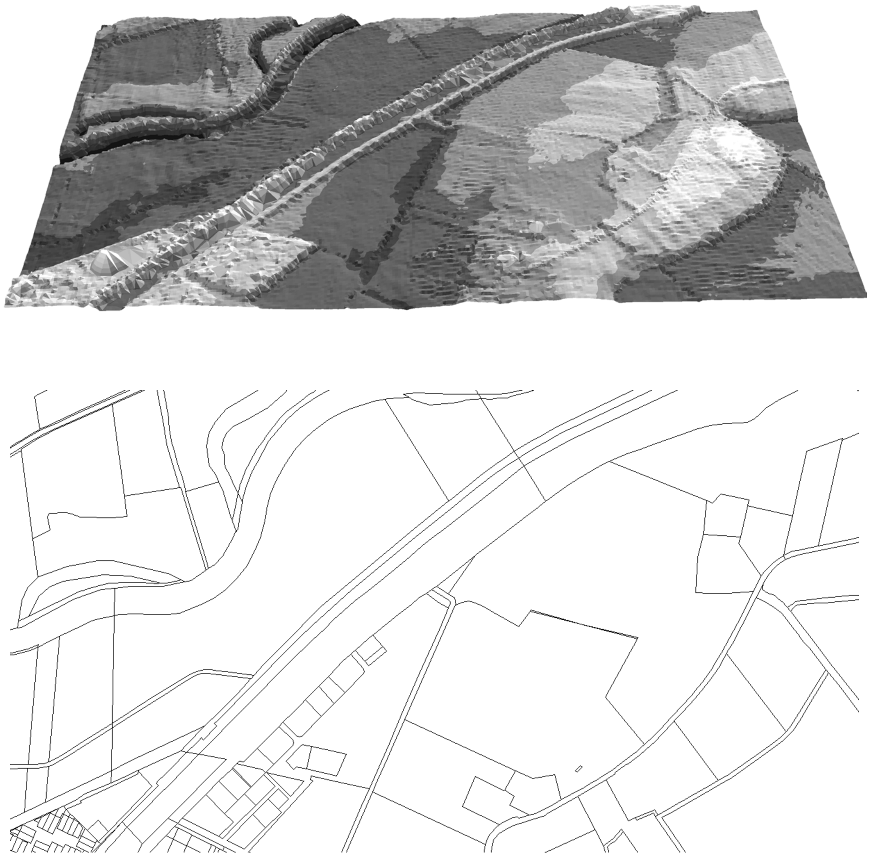

Fig. 1. Data sets used in this research; top: elevation (dark = low, light = high), bottom: cadastral parcels

One of the disadvantages of using a dense laser altimetry data set is the resulting data volume and with that the poor performance of the queries. However, due to the 'sampling' nature of data obtained with laser altimetry not all points are needed to generate an accurate elevation model (within epsilon tolerance in the same order of magnitude as the original height model and cadastral data). Therefore we examined how the number of TIN nodes (and thereby related edges and triangles) can be reduced by removing nodes that are not significant for the TIN, but at the same time maintaining the constraints of the parcel boundaries. Section 4 describes a method, which can be used to generate an effective TIN, including a representation of 2D objects, in which only the relevant points are used. Part of this method has been implemented, specifically the first 'generalization' step: the filtering of non-significant elevation points. The results of this prototype implementation are presented in section 5. This paper ends with conclusions.

2 Integrated TINs of point heights and parcels

To explore the possibilities of including a data set in a 2D planar partition in a TIN structure, four different types of TINs, all representing height models and the last three also including 2D objects, were generated: unconstrained Delaunay TIN (section 2.1), constrained Delaunay TIN (section 2.2), conforming Delaunay TIN (section 2.3) and refined constrained TIN (section 2.4) (also see Shewchuk, 1996).

2.1 Unconstrained TIN

First a TIN was generated using only the point data. The triangulation was performed outside the DBMS since TINs (and triangulation) are not (yet) supported within DBMSs. The ideal case would be just storing the point heights and the parcel boundaries in the DBMS and to generate the TIN of the area of interest on user's request within the DBMS, without explicitly storing the TIN structure in the DBMS. The representation of the implicit TIN could then be obtained via a view. This is more efficient and less prone to decrease in quality because no data transfer (and conversion) is needed from DBMS to TIN software and back. In the future a distributed DBMS structure may be possible within the Geo-Information Infrastructure (GII). An integrated view, based on two different databases (as the different data sets are maintained by different organisations in different databases) may be feasible from the technical perspective. In our research we stored copies of all data sets in one single DBMS.

In our test case, a TIN has been generated with Delaunay triangulation (Worboys, 1995). The Delaunay triangulation results in triangles, which fulfil the 'empty circle criterion', which means that the circumcircle around every triangle contains no vertices of the triangulation other than the three vertices that define the triangle. In general this results in good and numerically stable polygons. It should be noted that the Delaunay TINs are computed in 2D and may therefore be suboptimal for true elevation data. The z-value of points is not taken into account in the triangulation process, but added afterwards. This is perhaps a bit strange if one realises that the TIN is computed for an elevation model in which the z-value is very important; see (Verbree, 2003) for better TIN construction for terrain elevation models.

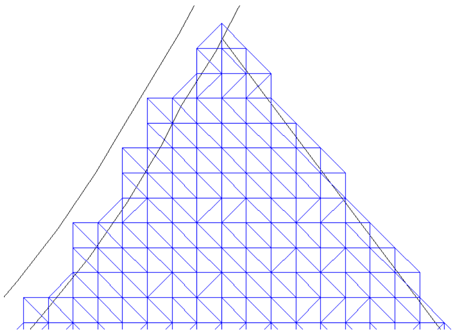

Fig. 2. A parcel surface (detail) based on an unconstrained TIN

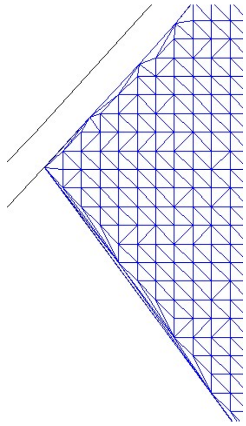

Fig. 3. A parcel surface (detail) based on a constrained TIN

2.2 Constrained TIN

The selection of triangles from the unconstrained TIN (partly) overlapping one parcel surface represents an area larger than the parcel itself since triangles cross parcel boundaries (figure 2). Therefore, to improve the selection of a parcel surface, a constrained TIN was generated. In order to obtain a more precise parcel surface, a constrained TIN was generated, using the parcel boundaries as constraints. We assigned z-coordinates to the nodes of parcel boundaries by projecting them in the unconstrained TIN. In contrast with the unconstrained TIN, each triangle in the constrained TIN (figure 3) belongs to one parcel only and therefore the selection of triangles exactly equals the area of a parcel. However, as can also be seen in figure 3, keeping the parcel boundaries (edges) undivided leads to elongated triangles near the location of parcel boundaries. This has two important drawbacks. First, the very flat elongated triangles may be numerically unstable (not robust, as small changes in the coordinates may cause errors) and the visualisation is unpleasant. Second, and maybe even more important, a long original parcel boundary will remain a straight line in 3D even when the terrain is hilly, because there are no intermediate points on the parcel boundaries by which it is not possible to represent height variance across the parcel boundaries.

2.3 Conforming TIN

Keeping the original edges in the constrained TIN undivided in the triangulation process leads to elongated triangles if parcel boundaries are much longer than the average distance between DEM points (5 meters) which is the case in using parcel boundaries with the AHN data set. An alternative to the constrained TIN may be the conforming TIN. The computation starts with a constrained TIN, but every constrained edge which has a triangle to the left or right not satisfying the empty circle condition is recursively subdivided by adding so-called Steiner points (and locally recomputing the TIN with the two new constrained edges). The recursion stops when all triangles, also the ones with (parts of) the constrained edges, satisfy the empty circumcircle criterion (the Delaunay property). The conforming TIN has both the Delaunay property and the advantage that all constrained edges are present, possibly subdivided in parts, in the resulting TIN. Figure 4 shows a conforming TIN, covering several parcels (different shades of grey). To improve visualisation the height has been exaggerated (10 times).

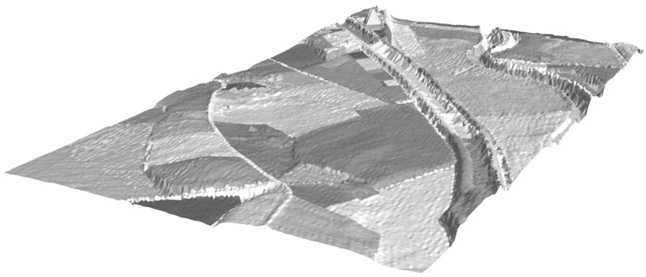

Fig. 4. Conforming TIN in which point heights and 2D planar partition of parcels (each with own shade of grey) are integrated.

2.4 Refined constrained TIN

However, also a (normal) conforming TIN has its drawbacks compared to a constrained TIN. In case of two very close 'near parallel' constrained edges, a large number of very small triangles are generated while these constrained edges are split in many very small edges (see figure 5). Something similar, can also happen when AHN points are very close to the constrained edges. These small triangles have no use, as they do not reflect any height differences (at least the height differences cannot be derived from the AHN points) and they also do not reflect additional object information.

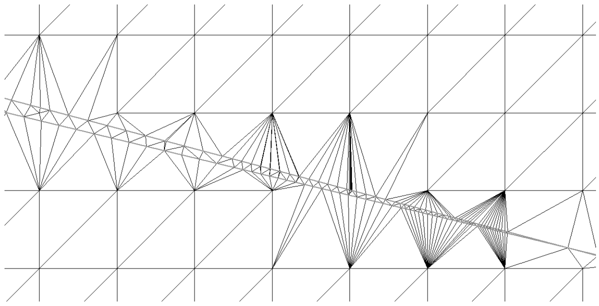

Fig. 5. Conforming TIN results in very small triangles in case of two very close near parallel constrained edges or in case AHN points are very close to constrained edges, while no extra information is added

A solution for this is splitting the constrained edges, before inserting them, into parts not larger than two or three times the average distance between neighbour AHN points and then computing the (normal) constrained TIN. Figure 6 shows the refined constrained TIN for one parcel. The edges of the parcel boundaries were split into parts of at most 10 meters. These edges were then used as constraints in the triangulation, which resulted in a refined constrained TIN. This improves the shape of triangles considerably (too flat and too small triangles are avoided). Moreover, since points are added on the parcel boundaries for which the height has been deduced based on the unconstrained TIN, it is possible to represent more variation in height across a parcel boundary. Also the problem of many, very small triangles (in case of close 'near parallel' constraints and input points close to constraints) in the conformal TIN is avoided.

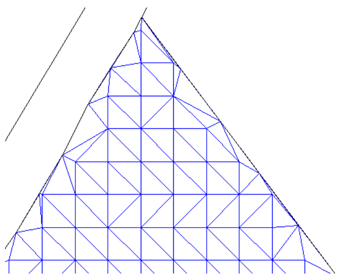

Fig. 6. A parcel surface based on a refined constrained TIN

3 Analysing and querying parcel surfaces from the DBMS

The actual extraction of a parcel surface is performed within the Oracle DBMS. A kind of topologically structured model is used in which the triangles are not explicitly stored, but are represented via the references to their nodes. During the analysis of parcel surfaces, all triangles that are covered by one parcel are selected by means of a spatial query. To select these triangles, first the realisation of the geometries of triangles needs to be performed. To illustrate the query to extract a parcel surface from the DBMS, the refined constrained TIN has been used. To speed up the query first a function-based index was built on the TIN table (R-tree index):

```
create table TIN_vertex (id number(10), location sdo_geometry,
   z number(10));
create table TIN_r (id number(10), v1 number(10), v2 number(10),
   v3 number(10));

insert into user_sdo_geom_metadata values
   ('TIN_R', 'return_geom(id)', mdsys.sdo_dim_array(
      mdsys.sdo_dim_element('X', 0, 254330, .001),
      mdsys.sdo_dim_element('Y', 0, 503929, .001)), NULL);
create index tin_idx on tin_r(RETURN_geom(ID))
indextype is mdsys.spatial_index;
```

The spatial query to find all points or triangles that are located within one parcel can be performed with a spatial function (sdo_geom.relate in the Oracle implementation). The query to select triangles that are within a specific query parcel (number 4589, municipality GBG00, section D) using the spatial function, is:

```
select id, return_geom(id) shape
from tin_r, parcels par
where parcel=' 4589' and municip='GBG00' and section=' D'
   and sdo_geom.relate(par.geom, COVEREDBY+INSIDE',
return_geom(tin_r.id),1)='TRUE';
```

For the unconstrained TIN we used the option 'anyinteract', since otherwise we miss the triangles that cross parcel boundaries.

3D area of parcel surface

The cadastral map is a 2D map containing projection of parcels. Consequently the cadastral map does not contain the true area of surface parcels. In mountainous countries the true area of parcels may be needed, since tax rates are based on the area of parcels. The integrated TIN based on height data and parcels can also be used for obtaining the true area of a parcel. The area of a parcel in 3D space can be computed by summing up the true area in 3D space of all triangles covering one parcel. DBMSs do not support 3D data types and consequently they also do not contain functions to calculate the area in 3D. Arens (2003) describes a research in which a 3D primitive, together with 3D functions, has been implemented as an extension of the Oracle geometrical model. The implementation is based on a proposal of (Stoter and Van Oosterom, 2002). To be able to compute the area of all triangles covering one parcel in 3D, the function 'area3D' that was implemented as part of the research of (Arens, 2003) was used. The 3D area calculation can therefore be performed inside the DBMS. First, we calculated the 2D area of the original parcel polygon. The query parcel is the parcel with a small 'hill' on it (see figure 4):

```
select sdo_geom.sdo_area(geom, 0.1) from parcels
where parcel=' 4589' and municip='GBG00' and section=' D';
```

The area in 2D is 6,737 square meters. The 3D area of the same parcel, which resulted in 6,781 square meters, is performed with the following query:

```
select sum(area3d(return_geom(ID)) from tin_r, parcels par
where parcel=' 4589' and municip='GBG00' and section=' D' and
  sdo_geom.relate(par.geom, 'COVEREDBY+INSIDE',
return_geom(id),0.1)='TRUE';
```

As can be seen from these results, the difference between the projected area and the real area in 3D of this parcel is 44 square meters. Other queries can be performed as well, e.g. fine steepest triangle, find all triangles pointing to the south, or find the highest (lowest) point in this parcel:

```
select max(z), min(z) from tin_vertex, parcels par
where par.parcel=' 4589' and par.municip='GBG00' and par.section='
D'
  and sdo_geom.relate(par.geom, 'COVEREDBY+INSIDE',location,0.1)
  ='TRUE';

   MAX(Z)    MIN(Z)
   ------  ---------
   14.24     10.027
```

4 Generalization of the integrated height and parcel TIN

Both the conforming TIN and the refined constrained TIN (with constraints based on subdivided parcel boundaries in order to avoid long straight lines) look promising: the triangles are well shaped (not too flat and in case of the conforming TIN, the Delaunay criterion is fulfilled) and points are added on parcel boundaries in order to represent more height variance on them. However, after some analyses we suspected that far too many points are used in order to represent the surface TIN with the same horizontal and vertical accuracy as the input data sets (AHN points and cadastral map). Note that this is already the case in the input data of AHN, but has become somewhat worse in the conforming TIN and refined constrained TIN. A problem of having huge data sets, is the resulting data volume and with that poor performance of queries and analyses. Therefore filtering of the data set aiming at data reduction (generalization) is needed.

This section describes two methods to improve the initial integrated height and object model: a detailed-to-coarse approach (section 4.1) and a coarse-to-detailed approach (section 4.2). In section 4.3 a more advanced generalization method of the integrated model is discussed (that is, more than based on height only).

4.1 Detailed-to-coarse approach

The first method starts with the complete integrated model. From this model a number of non-relevant point heights are removed while maintaining the significant points, e.g. removing the points where the normal vectors of the incident triangles have a small maximum angle. After removing such a point, the triangulation is locally corrected and it is explicitly checked if the height difference at the location of the removed point in the new TIN is within this tolerance. If so, the point was indeed not significant for the TIN and can be removed. In this process the parcel boundaries are still needed as constraints, since the aim is to be able to select a parcel surface from the TIN.

The filtering (aiming at data reduction) is based on filtering the TIN structure and not the point heights themselves. The filtering can use the characteristics of the height surface. On location with little variance in height, points can be removed while on the location with higher variances points are maintained to define the variance in height accurately. Important advantages of data reduction in a TIN structure, compared to gridded structures, are that they can be used on irregularly distributed points and that locations with high height variance will remain as such in the new data set. The prototype implementation is based on this method and more details can be found in section 5. The result of the generalization is shown in figure 7. This may be considered the first step of a larger generalization process (see section 4.3).

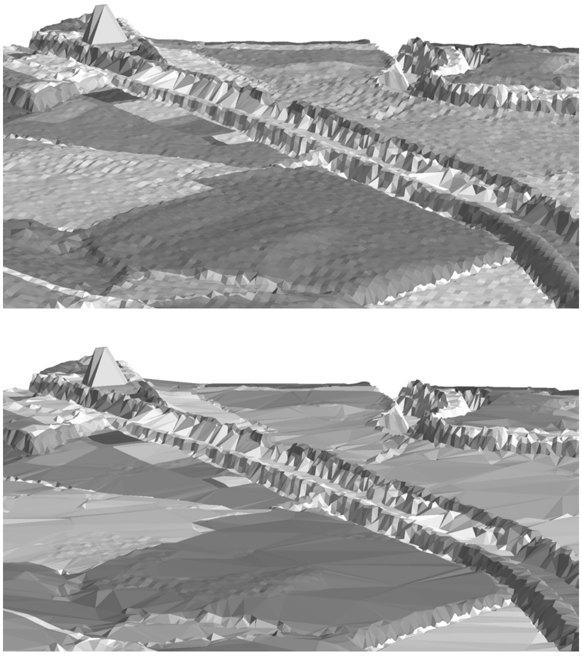

Fig. 7. Conforming TIN in which point heights and 2D planar partition of parcels are integrated, before (top) and after (bottom) filtering

4.2 Coarse-to-detailed approach

The procedure described above starts with all available details and then tries to remove some of the less relevant details, which is not always easy. An alternative method would be starting with a very low detail model and then adding points where the errors are the largest. The initial model could be just the constraints (with estimated z-values at every point of the parcel boundary) inserted in a (conforming/constrained) TIN. In the next step the AHN height point with the largest distance to this surface is located. If this point is within `eps_vert` distance from the surface (epsilon tolerance in the vertical direction), then the model already satisfies the accuracy requirements. If this point is not within the tolerance, then it is added to the TIN (and the TIN is re-triangulated under the TIN conditions). This procedure is repeated until all AHN height points are within the tolerance distance. This procedure is a kind of 2.5D counterpart of the well-known Douglas-Peucker (Douglas and Peucker, 1973) line generalization.

4. 3 Integrated height and object generalization

Until now, only the height was taken into consideration during the generalization process, both in the detailed-to-coarse and coarse-to-detailed approach. However, as the model is supposed to be an integrated model of height and objects, also the objects should participate in the generalization. Therefore the integrated height and object model could be further generalized by taking into account both the elevation aspect and the 2D objects at the same time. It is already possible to separately generalize the terrain model (Bottelier, 2000; Brugelmann, 2000; Jacobsen and Lohmann, 2003; Passini and Betzner, 2002) and 2D objects (Douglas and Peucker, 1973; Van Oosterom, 1995). However, the integrated generalization of the height and object model makes this model also well suitable for truly integrated generalization of both the terrain elevation model and the object model at the same time. This is a novel approach. Starting with the detailed-to-coarse approach one could identify the following steps:

Step 0: Integrate raw elevation model (AHN) and objects (parcel boundaries) in a (conforming or refined constrained) TIN; see section 2.

Step 1: Improve the efficiency of the TIN created in step 0 by removing AHN points under the conditions of the TIN until this is now longer possible given the maximum tolerance value `eps_vert_1` (as described in section 4.1). Note that this tolerance could be adjusted for different circumstances, but the initial value should be the same size as the accuracy of the input data.

Step 2: Now also start generalization of the object boundaries, for example with the Douglas-Peucker line generalization algorithm, by removing those boundary points which do not contribute significantly to the shape of the boundary. This can be done in 2D (standard Douglas-Peucker), but it is better to apply this algorithm in 3D. Keep on removing points until this is impossible within the given tolerance `eps_hor_1`. After this line generalization of the constraints, re-triangulate the TIN according to the rules as in step 1 (of a conforming or refined constrained TIN).

Step 3: Finally, for multi-resolution purposes, also start aggregating the objects, for example in our case: parcels to sections (and the next aggregation level would be sections to municipalities, followed by municipalities to provinces, etc.). In fact this is removing some of the constrained edges (original parcel boundaries) from the input of the integrated model. Repeat step 1 and 2 with other values for the epsilon tolerances at every aggregation level: `eps_vert_2`, `eps_hor_2` (at the section level), `eps_vert_3`, `eps_hor_3` (at the municipality level), ...

5 Prototype

The first steps (step 0 and step 1) have been implemented in a prototype. During the development of the filtering algorithm inspiration was drawn from Cellular Automata and more specific from the Game of Life (Wojtowicz, 2004). Cellular automata is the generic name of a set of mathematical point operators, which change repeatedly the state of a collection of cells in a chaotic order. For example, in the 'Game of Life' the starting point is a grid with cells that can be either 'on' or 'off'. A small set of rules defines the state of a cell in the next generation, based on the state of the direct neighbouring cells. John Conway, the mathematician who invented the best-known cellular automation 'Game of Life', defined the following criteria for cellular automata (Wojtowicz, 2004):

- There should be no initial pattern for which there is a simple proof that the population can grow without limit.
- There should be initial patterns that apparently do grow without limit.
- There should be simple initial patterns that grow and change for a considerable period of time before coming to an end in three possible ways:
    - fading away completely (from overcrowding or from becoming too sparse),
    - settling into a stable configuration that remains unchanged thereafter, or
    - entering an oscillating phase in which they repeat an endless cycle of two or more periods.

The beauty of these procedures is its fuzzy-like, unpredictable behavior, which provided useful stable results in modeling amongst others urban development and chemical, physical and biological dynamic processes.

In the prototype the basic idea of cellular automata is adopted as the points in the surface TIN are considered as having a boolean state, namely 'characteristic' or 'non-characteristic', which determines whether the point makes a significant contribution to the shape of the surface model. The analogy with the cellular automata can be carried further in the decision criteria, i.e. two criteria are defined, one that defines whether a point's state changes from 'characteristic' to 'non-characteristic' and one for the reverse operation. The question whether points are characteristic, is in the prototype dependent on the following conditions (note that these conditions are, analogously to cellular automata, based on local characteristics):

- A point is characteristic if the angles of neighbouring triangles of the point are significantly different (Brugelmann, 2000). To detect this, the normal vectors for the neighbouring triangles are determined and compared. If the difference is bigger than a given threshold angle, the point will be defined as characteristic and not be removed from the TIN.
- Local minima and maxima are also characteristic points of a TIN. If two neighbouring triangles are in the same direction in the first condition, the change in angle is less important than in the case the change in angle demarcates a top or a valley. Therefore, a smaller threshold angle is used in case the specific point is a local minimum or maximum. A minimum or maximum is the case when the azimuths of two neighbouring triangles are opposite of each other, which can be determined by calculating the differences in the azimuths. If the difference is bigger than a given threshold value, a smaller threshold angle is used in the first condition. In the used software the azimuth of a triangle is one of the automatically generated TIN attributes and therefore available without additional calculations

Based on these conditions a point's state is determined. If two neighbouring triangles of one point already fulfil one of the criteria, the point's state will be 'characteristic' and the point will therefore be maintained. All other points are marked 'non-characteristic' and therefore removed from the surface TIN. Subsequently after determining these two types of nodes, it is checked whether all non-characteristic nodes are really allowed to be removed (given the height tolerance). This is done by calculating the height difference at the location of the removed point between the original TIN and the new TIN that is generated based on the reduced data set. If this difference is bigger than a threshold value the state changes from 'non-characteristic' to 'characteristic' and the removed point is re-added. This now completes one iteration of data reduction. After this step the data reduction is performed again as the data reduction is an iterative process until a more or less stable data structure is obtained.

The prototype has been implemented in the 3D Analyst extension of ArcView (ESRI) using the macro language of ArcView (Avenue). In ArcView the TIN is recognised as an object and therefore the TIN data structure can be used directly in the reduction algorithm and in addition the results can easily be visualised. Figures 8 and 9 clearly illustrate how the filtering maintains all terrain shapes but reduced the number of points substantially (height is exaggerated ten times). This prototype shows already the possibility of data reduction on a TIN, but should be implemented as part of the database in the future, once a TIN data structure is supported as data type in a geo-DBMS.

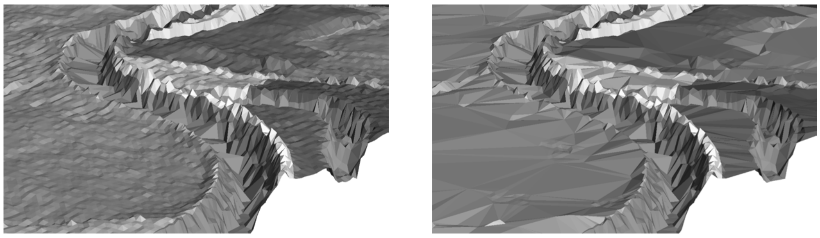

Fig. 8. Detail of filtering results: before (left) and after (right) data reduction

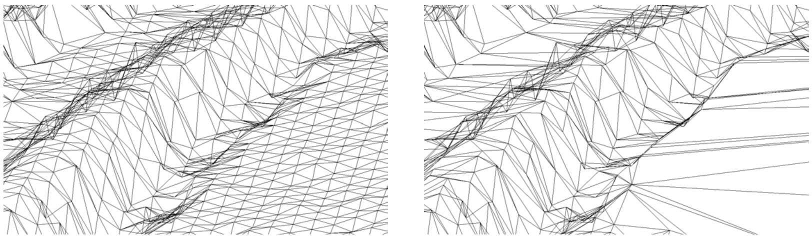

Fig. 9. Detail of filtering results: before (left) and after (right) data reduction

For our initial test, the data reduction is performed on the unconstrained TIN. This means that the 2D objects and the point heights are kept separately during the data reduction process in order to get a first impression of the achievable results. Incorporating the constraints at this stage would have made the data reduction process more complex. Figure 7 already showed the conforming TIN of our test data set, before and after filtering. We did experiments with different parameters. The parameters that showed best results for an initial generalization (filtering), were: minimum angle between two neighbouring triangles to be a characteristic point 4.5 resp. 3 degrees (if a point resp. is or is not a top or a valley), difference in azimuth between two neighbouring triangles to determine if two triangles are opposite of each other: 120 degrees. The maximum allowed difference in height to determine if a removed point should be re-added was 0.25 meters. Apart from the minimum angle, the chosen parameters values are based on previous research (Penninga, 2002).

The data set used in this example covers an area of 1,450 by 800 meters and contains 44,279 AHN points (maximum z-value 14.2 meters, minimum z-value 6.7 meters, mean 9.5 meters). Three iterations steps were used to filter the data set. In the first iterative step, 34,457 AHN points were removed (9,822 were considered to be characteristic). The average height difference between the original and the new TIN was 0.09 meter. 3,243 points were re-added since they exceeded the height difference of 0.25 meter, which resulted in 13,065 points after the first step. After the second iteration step 8,697 points were determined as characteristic, the average height difference between the new and the original point was again 0.09 meter, 3,469 points were re-added and this all resulted in 12,166 points. After the third iteration step, 8,455 points were considered to be characteristic (average height difference 0.09 meter) and 3,529 points were re-added. After this step the data reduction process was stopped. The results of the data reduction process are listed in the following table:

Table 1. Results of iteration steps in the data reduction process

Iteration step	# input points	# rem. points	# char. points	# re-added points	Reduction rate
1	44,279	34,457	9,822	3,243	70 %
2	13,065	4,368	8,697	3,469	73%
3	12,166	3.711	8,455	3,529	73%

After the total data reduction process 11,984 points from the original 44,279 points were maintained. This is a reduction of 73%. As can be seen from figure 10, points were removed from areas with little height variance, while density of point heights in areas with high height variance (e.g. on the dikes) is still high.

Fig. 10. Results of data set after reduction (points not removed are black)

6 Conclusions

Incorporating the planar partition of 2D objects, e.g. the cadastral map, into a height surface makes it possible to extract the 2.5D surfaces of 2D objects and to visualise 2D maps in a 3D environment by using 2.5D representations. As described and discussed in section 2 it is not easy and straightforward to create a good integrated elevation and object model. Several alternatives were investigated, unconstrained Delaunay TINs, constrained TINs, conforming TINs, and finally refined constrained TINs. After some analyses, the most promising solution, the refined constrained TIN, was selected and applied with success to our test case with real world data: AHN height points and parcel boundaries.

The integrated model however, contains too many AHN points, which do not contribute much to the actual terrain description. Therefore we proposed a method to generalize the integrated model. We implemented the first step of this method into a prototype. In this prototype non-characteristic points are removed from the (unconstrained) TIN in an iterative filtering process based on cellular automata. The filtering was conducted directly on the surface instead of on the point attributes. As can be concluded from experiments with the prototype, it is possible to determine important terrain characteristics by using a simple criterion (difference in angle of neighbouring triangles). With this method it is possible to reduce the data set considerably. The test data set contained about 4 times fewer points after generalization, but still within the epsilon tolerance of the same size as the quality of the original input data sets. On the other hand significant information on the height surface is still available in the TIN. The initial filtering yielded therefore a much improved integrated model.

Finally, it was indicated that the integrated height and object model could be further generalized by taking into account both the elevation aspect and the 2D objects at the same time. This is a complete new research topic and no previous results are known. Of course, it is possible to separate generalization of the terrain model and 2D objects. However, the integrated generalization of the height and object model should make this model also well suitable for other resolutions (scales) or even in a multi-resolution context. An initial algorithm has been outlined. Future work with respect to the integrated generalization of the height and object model includes:

- Implement TIN data structure and triangulation within the geo-DBMS, as well as the data reduction methods (removing need to in- and export data between DBMS and TIN applications).

- Implement more of the proposed integrated model, especially the boundary line generalization and the object aggregation.
- Test the proposed integrated generalization with the original AHN point data (that is not-filtered and not resampled to a grid structure).
- Maintain the result of the generalization in a multi-scale data structure, as the costs of the computations are significant. A data structure similar to the BLG-tree (for line generalization based on the Douglas-Peuker algorithm, (Van Oosterom, 1995)) or the GAP-tree (for storing the result of generalization of an area partitioning (Van Oosterom, 1995a)) should be developed for the integrated height and object model.
- As indicated in section 2, the current TIN computation takes place in the 2D plane. It may be better to compute the integrated height and object model in true 3D space, based on tetrahedrons (and then finding the proper surface within this tetrahedron network) (Verbree 2003).
- Test with applications based on integration of height and objects from other domains than the current cadastral parcels; examples of these are topographic, soil type, or land-use data sets. Also these objects can be aggregated to larger objects during the generalizations (and the result can be stored in the earlier mentioned GAP-tree, but this is purely 2D until now).

Acknowledgements

We would like to thank RWS-AGI and the Netherlands' Kadaster for making respectively their AHN and cadastral data available for our research. Further we would like to thank the developers of the Triangle software for making their product available. Finally, we would like to thank our partners in the GDMC (Geo Database Management Center), Oracle and ESRI, again for making their software available. This publication is the result of the research programme 'Sustainable Urban Areas' (SUA) carried out by Delft University of Technology.

References

Arens, C., J.E. Stoter, and P.J.M. van Oosterom, 2003, Modelling 3D spatial objects in a GeoDBMS using a 3D primitive. In Proceedings AGILE 2003, Lyon, France, April 2003.

Bottelier, P., 2000, Fast Reduction of High Density Multibeam Echosounder Data for Near Real-Time Applications. The Hydrographic Journal, (98), 2000.

Brugelmann, R., 2000, Automatic break line detection from airborne laser range data. International Archives of Photogrammetry and Remote Sensing, 33(B3/1):109–116, 2000.

Douglas, D.H. and T. K. Peucker 1973, Algorithms for the reduction of points required to represent a digitized line or its caricature. Canadian Cartographer, 10:112–122, 1973.

Gorte, B., 2002, Segmentation of TIN-structured surface models. In Joint Conference on Geo-spatial theory, Processing and Applications, Ottawa, Canada, July 2002.

Heerd, van. ,R.M., et al., 2002, Productspecificatie AHN 2000. Technical Report MDTGM 2000.13, Rijkswaterstaat, Meetkundige Dienst, 2000 (in Dutch).

Jacobsen, K. and P. Lohmann, 2003, Segmented filtering of laser scanner DSMS. In ISPRS working group III/3 workshop 3D reconstruction from airborne laserscanner and InSAR data, Dresden, Germany, October 2003.

Lemmen, C.H.J., E. Oosterbroek, and P.M.J. van Oosterom, 1998, New spatial data management developments in the Netherlands Cadastre. In Proceedings of the FIG XXI International Congress, commission 3, Land Information Systems, pages 398–409, Brighton UK, July 1998.

Lenk, U., 2001, Strategies for integrating height information and 2D GIS data. In Joint OEEPE/ISPRS workshop: From 2D to 3D, establishment and maintenance of national core spatial databases, Hannover, Germany, October 2001.

Oosterom, van, P.J.M., 1995, The GAP-tree, an approach to 'On-the-Fly' Map Generalization of an Area Partitioning. In Müller, J.P. Lagrange, and R. Weibel, editors, GIS and Generalization, Methodology and Practice, chapter 9, pages 120–132. Taylor & Francis, 1995.

Oosterom, van, P.J.M. and V. Schenkelaars, 1995a, The Development of an Interactive Multi-Scale GIS. International Journal of Geographical Information Systems, 9(5):489–507, 1995.

Oosterom, van, P.J.M., J.E. Stoter, C.W. Quak, and S. Zlatanova, 2002, The balance between geometry and topology. In D. Richardson and P.J.M. van Oosterom, editors, Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling, Ottawa, Canada, July 2002.

Passini, R. and D. Betzner, 2002, Filtering of digital elevation models. In Proceedings FIG, ACSM/ASPRS, Washington D.C. USA, April 2002.

Penninga, F., 2002, Detectie van kenmerkende hoogtepunten in TIN's voor iteratieve datareductie (in Dutch). In Geo-informatiedag Nederland 2002, Ede, the Netherlands, February 2002.

Shewchuk, J.R., 1996, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In First Workshop on Applied Computational Geometry, pages 124–133, Philadelphia, Pennsylvania, USA, May 1996.
Stoter, J.E. and B. Gorte, 2003, Height in the cadastre, integrating point heights and parcel boundaries. In Proceedings FIG Working Week, Paris, France, April 2003.
Stoter, J.E. and P.J.M. van Oosterom, 2002, Incorporating 3D geo-objects into a 2D Geo-DBMS. In Proceedings FIG, ACSM/ASPRS, Washington D.C. USA, April 2002.
Verbree, E. and P.J.M. van Oosterom, 2003, The STIN method: 3D surface reconstruction by observation lines and Delaunay TENs. In ISPR workshop '3-D reconstruction from airborne laserscanner and InSAR data, Dresden, Germany, October 2003.
Wojtowicz, M., 2004, http://psoup.math.wisc.edu/mcell/, 2004.
Worboys, W.F., 1995, GIS, a computing perspective. Taylor and Francis, London, 1995.