

Scanline forced Delaunay TENs for surface representation

ir. Edward Verbree and prof.dr.ir. Peter van Oosterom
Department of Geodesy – Section GIS-technology
Delft University of Technology
the Netherlands
e.verbree@geo.tudelft.nl

Commission III, Working Group 3

KEY WORDS: Delaunay Tetrahedron Networks (TEN), TIN, Surface reconstruction

ABSTRACT

The general idea that a Delaunay TIN (DT) is more appropriate than non-Delaunay TINs, due to ‘better’ shaped triangles, might be true for many applications, but not for height dependent analytical queries. This is because the distribution of the triangle tessellation is defined in the two-dimensional XY-plane, by ignoring the Z-value in the Delaunay empty circum circle criterion. Alternatively, Data Dependent Triangulations (DDT) aim to identify which triangulation of a given function $z=f(x,y)$ over a given set of points will optimize some quality, i.e. the minimal spatial area of the surface or the volume below the resulting surface. This might be a good approach, but still there is no certainty the TIN represents the actual surface. Besides that, a 2D-TIN (Delaunay or not) is only capable to solve 2D (or 2.5D) data distributions. The reconstruction of the surface given by a set of surface points alone is therefore not unambiguous.

This paper describes a surface reconstruction method based on the scanlines, the lines-of-sight or measurements between the observer (or the measurement platform) and the target (the measured point). As the scanlines do not belong to the surface, we have to use a ‘real’ 3D triangulation construction method, resulting in a Tetrahedronized Irregular Network. This TEN is capable to store all kinds of surface-features (as the target-points) and the scanlines as well. The scanlines are forced to split by adding Steiner points until they are part of the Delaunay TEN. This procedure gives us the additional information needed to use the TEN to reconstruct the surface. The method is demonstrated by the non-trivial case of a set of measured points in a regular square distribution showing the improved surface reconstruction technique.

1 INTRODUCTION

Triangulated Irregular Networks are commonly used for Digital Terrain Modeling. These models are needed for several applications, in which the height value is often an important factor in the calculations and analyses. Visibility computations and line-of-sight communication problems require an appropriate representation of the ‘real’ terrain-surface.

The advantage of a TIN above other tessellations, like regular square grids (RSG), is the possibility to incorporate the exact position of the measured heightpoints, contourlines and breaklines into the model. In that, the actual measurements are well represented. But, one thing we often forget is the fact that the distribution of the triangle-tessellation is defined in the two-dimensional XY-plane, by ignoring the Z-value in the Delaunay empty circum circle criterion (Lawson, 1977).

The general idea that a Delaunay-TIN is more appropriate than non-Delaunay TINs because of well-formed triangles might be true for many applications, but not for height dependent analytical queries. An alternative is to use a data dependent triangulation for surface representation (Dyn, 1990). This might be a better direction to look for, but still there is no certainty that the TIN represents the actual surface. Besides that, a 2D-TIN (Delaunay or not) is only capable to solve 2D or 2.5D data distributions.

We can however use an additional observation: the scanline. One property of an observation is the line-of-sight; no surface is allowed between the observation-point and the target-point (the actual point on the surface). Hence, we consider a method for triangulations forced by scanlines, which means that not only the measured heightpoints are included into the TIN, but the measurements as well.

As the scanlines do not contribute to the representation of the surface, they cannot be treated as the surface features types like breaklines and contourlines. They could even be perpendicular, which makes it impossible to incorporate them into a 2D TIN. So we have to use a ‘real’ 3D triangulation construction method, resulting in a TEN (Tetrahedronized Irregular Network). A TEN is also known as a 3D-TIN, indicating the 3D conditions taken into account at the construction, i.e. in case of a Delaunay TEN the empty circum sphere criterion in contradiction to the empty circle criterion in the 2D or 2.5D case. This TEN is capable of storing all kinds of surface-features, including the scanlines. The final scanline forced TEN gives us the information needed to construct the triangles, which will define the surface we are looking at and the body (volume) of that object as a set of tetrahedrons.

In section 2, we will recall some basics of Delaunay Triangulations (DT) and Data Dependent Triangulations (DDT). Section 3 presents the ideas and some examples for scanline constrained TENs in more detail. Section 4 contains some

concluding remarks, where section 5 gives some directions for further research.

2 DELAUNAY TRIANGULATIONS AND DATA DEPENDENT TRIANGULATIONS

Most commercial GISs are capable to model surfaces by TINs. These surfaces provide insight, reveal trends and solve problems. Simple geometry features like masspoints and in addition breaklines en contourlines are the input for the TIN construction. The Z-value (or an attribute representing a heightvalue) of these features is stored as the Z-value of the nodes of the TIN.

A Delaunay TIN fulfils the 'empty circum circle criterion' (Lawson, 1977). If the point distribution is regular, for example if all points are on a regular square or rectangle (figure 1a), this criterion has two equal choices.

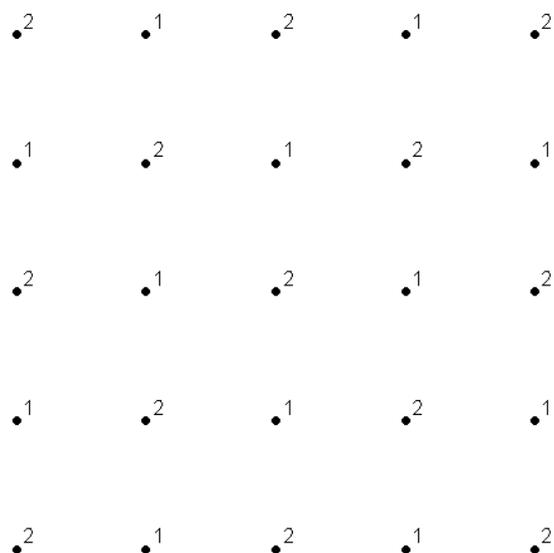


fig 1a: regular datapoint distribution, labels indicate heightvalues

In figure 1b the diagonal of all the triangles is directed northwest to southeast. This could be, under the same Delaunay criterion as four points are located on the common circum circle of a grid cell, for all squares from north-east to south-west, or even randomly chosen.

One has to realize that the given height values (or the Z-value of the nodes) do have consequences for derivatives like slope and aspect, visualization (like hill-shading) and volume statistics (like viewsheds and cut and fill calculations). However they are not used to control the 2D Delaunay triangulation process.

An other artifact is the possible occurrence of so-called 'flat' triangles. Especially TINs based on input of contourlines are known for this problem. Introducing derived drainage channels and ridges by examining the skeleton or medial axes can solve this problem, see for example (Tinghua, 2001).

One can argue that the 2D-Delaunay TIN is just one of the possibilities to triangulate a set of points (nodes) and lines (edges). In fact any triangulation could be a candidate for a 2.5D

terrain surface representation. Extensive research on Data Dependent Triangulations (DDT) proves this observation. The idea is either to maximize or to minimize some cost-function that expresses certain local, regional or global properties of the resulting surface (Dyn, 1990; Bern 1992; Yu, 2001; Lenk, 2001).

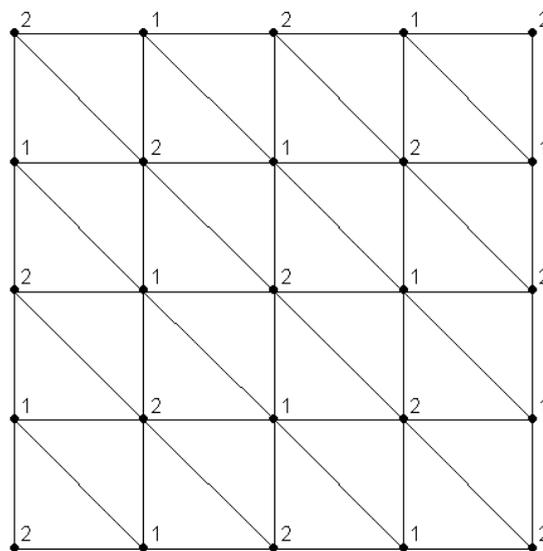


fig 1b: a possible Delaunay TIN of regular datapoint distribution

A few possible options are:

- total surface area minimum
- total content below surface (volume) minimal
- total content above surface minimal
- angles within 3D triangle minimum
- angles / bends between two triangles minimal
- no points in 3D sphere of any triangle

The results of DT can be ambiguous. This is illustrated by the following simple example: determine the surface-area and volume for the surface given by: $p_1=(0,0,0)$; $p_2=(0,1,0)$; $p_3=(1,1,10)$; $p_4=(1,0,0)$.

If the diagonal is chosen as edge (p_1,p_3) , the surface-area equals to:

$$\text{area}(p_1, p_2, p_3) + \text{area}(p_3, p_4, p_1) = \frac{1}{2} * 1 * \sqrt{101} + \frac{1}{2} * 1 * \sqrt{101} = \sqrt{101} \sim 10.05$$

The volume (regarding to 0-level) equals to:

$$\text{volume}(p_1, p_2, p_3) + \text{volume}(p_3, p_4, p_1) = \frac{1}{2} * \frac{1}{2} * 10 + \frac{1}{2} * \frac{1}{2} * 10 = 5$$

If the diagonal is chosen as edge (p_2,p_4) , the surface-area equals to:

$$\text{area}(p_1, p_2, p_4) + \text{area}(p_2, p_3, p_4) = \frac{1}{2} * 1 * 1 + \frac{1}{2} * \sqrt{2} * \sqrt{101 - \frac{1}{2}} = \frac{1}{2} (1 + \sqrt{201}) \sim 7.59$$

The volume (regarding to 0-level) equals to:

$$\text{volume}(p_1, p_2, p_4) + \text{volume}(p_2, p_3, p_4) = \frac{1}{2} * \frac{1}{2} * 0 + \frac{1}{2} * \frac{1}{2} * 10 = 2.5$$

If we minimize the surface-area we will chose for diagonal-edge (p_2,p_4) , but if we maximize the volume below the surface, edge (p_1,p_3) was preferred. As said, the 2D Delaunay TIN has no preferred edge.

In this case it could be argued that the 2D Delaunay criterion could not make the choice and that the better solution is also according to the criterion. However, it is possible to construct other situation in which the 2D Delaunay solution results in a less good 2.5D TIN than a certain alternative; e.g. assume the points $p_1 = (0,1,0)$, $p_2 = (1,0,0)$, $p_3 = (3,1,0)$ and $p_4 = (1,2,10)$. The surface-area on the Delaunay based TIN equal to: $\frac{1}{2} * (\text{sqrt}(204) + \text{sqrt}(516))$ which is clearly more than the non-Delaunay alternative: $1.5 * (1 + \text{sqrt}(101))$

These local optimizations could disregard regional phenomena, like ridges and faults, and as long a projection is made to the XY-plane, no overhanging cliffs or other disturbances are possible. And these disturbances could be the surface-phenomena we are looking for, like (near) vertical walls of buildings or viaducts. We will therefor forget for the moment the data independent (like 2D Delaunay TINs) and data dependent TIN and concentrate on Delaunay 3D-TINs or TENS for surface reconstruction.

3. SCANLINE FORCED TRIANGULATION

Height datapoints are more and more collected by laserscanning from a platform at an airplane or helicopter, but scanning from a ground platform is also possible. The result of this process is a point cloud of target-points. To calculate these target-points, the position of the observer (i.e. the laser itself) has to be known and thus the scanline between the observer point and the target point. In our approach we will use these scanlines for the reconstruction of the surface at an increasing complexity demonstrated by a 1.5D, 2D and a 2.5D scenario.

3.1 1.5D Case

In exploring this problem, first a step back was taken as we consider the 1.5 scenario, where 1D-points are taken, and together with the height value and the scanline a surface is reconstructed. This could be quite trivial to do, because we can order the datapoints on their X-value. But as, later on, in two dimensions this is not straightforward, we have to use an algorithm, which will not take this ordering as a precondition.

The aim of the algorithm is to find the object and boundary defined by a set of heightpoints and scanlines. The volume is bounded by the points left and right of the dataset, both given the heightvalue of zero. The scanlines are shortened to a given value above the most extreme heightvalue, and for these examples dropped as perpendiculars.

We will give the algorithm in pseudo-code:

```

Algorithm 'Scanline_TIN'
step 1: Construct TIN
input target-points
create 'scanline forced' Delaunay TIN

step 2: Transform Edges
for each scanline
  get target-point-A

```

```

create list of connected scanlines
for each connected scanline
  get target-point-B
  create edge(target-point-A,
              target-point-B)
end for
end for

```

Step 1: Construct TIN

First a regular Delaunay TIN is created by the set of target-points. and observer-points. Then, an iterative process is started. Each scanline not being an TIN-Edge is forced to sub-divide into parts. The newly formed nodes are included as points into the TIN. This procedure ends when all scanline-parts are represented by an TIN-Edge in the Delaunay TIN (fig 2a).

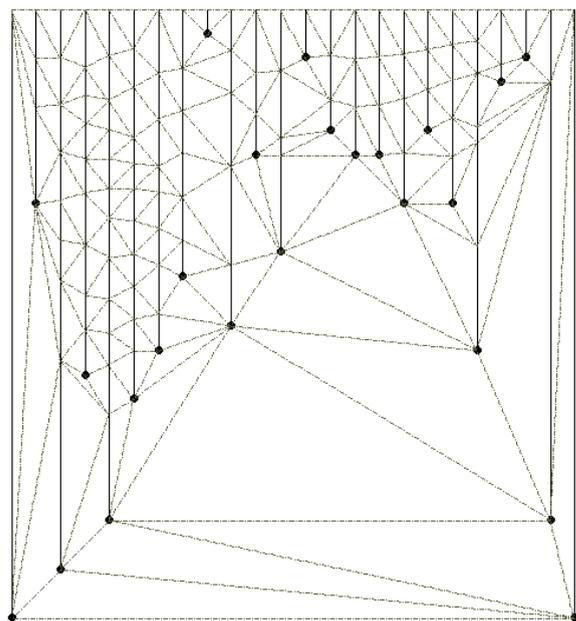


fig 2a: 'scanline forced' Delaunay TIN

This is an alternative to normal constrained triangulations, but a defendable approach, because the added nodes (Steiner points) are used to reconstruct the surface (see next step). The addition of Steiner points to a Delaunay Triangulation is a powerful concept in computational geometry which allows quite theoretical investigations. It forms the basis for many provable optimal triangulation algorithms for various quality criteria (Fleischmann, 1999).

Step 2: Transform Edges

The actual interior and boundary of the object given by the target-points and scanlines is found by the procedure in which for each scanline the connected scanlines are determined by examine the TIN-Edges. All TIN-Edges with one node known as a target-point and one node known as an added Steiner point or an other target-point are selected. If a TIN-Edge links two target-points this TIN-Edge is stored, else the TIN-Edge is dropped and replaced

by a new TIN-Edge. This new TIN-Edge is created by the original target-point and the target point at the end of the scanline belonging to the Steiner point (fig 2b).

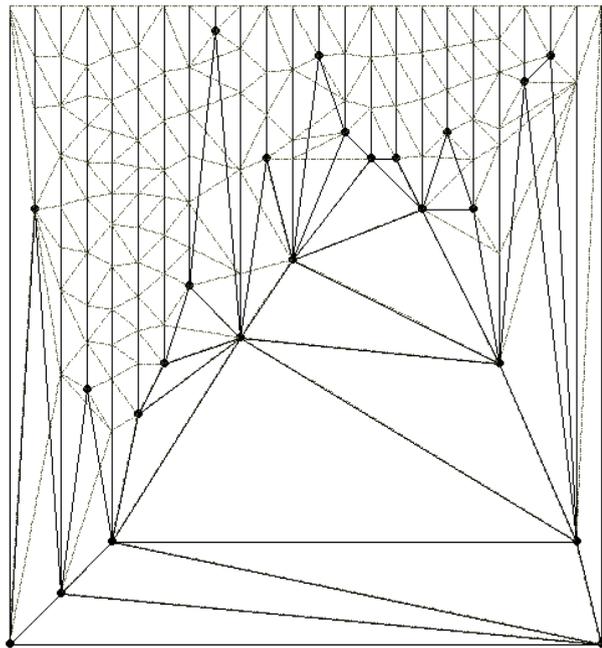


fig 2b: reconstructed solid and surface of 1.5D dataset

The 1.5D-case is quite trivial, but it explains the use of the subdivided scanlines. The dataset is 1.5D as the Z-value is a property (measurement) of the XY-values. The TIN is Delaunay for that part where the newly formed TIN-Edges do not intersect with the original TIN; the other part is non-Delaunay.

This algorithm is implemented as a prototype, written in the object-based Language Avenue of ESRI ArcView3.2, extended by 3D-Analyst (ESRI, 2001).

3.2 The 2D Case

One can argue that the applied algorithm for the 1.5D (X, Z) example given in the previous section is a little ‘over the edge’, as we can simple sort the datapoints on their X-value. To show the possible use of the ‘scanline forced’ triangulation for 2.5D and 3D applications, the same algorithm is applied for the reconstruction of a 2D-polygon with concavities.

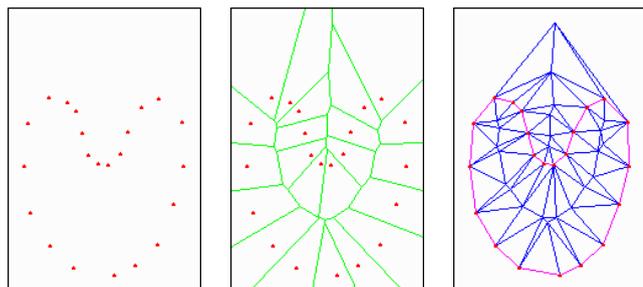


fig 3a: A point set, its Voronoi Diagram and its crust

This method has the same goal as the crust and skeleton method, see fig 3a (Zhang, 2001), where the existents of concavities is not derived by the Voronoi Diagram, but by the given set of scanlines.

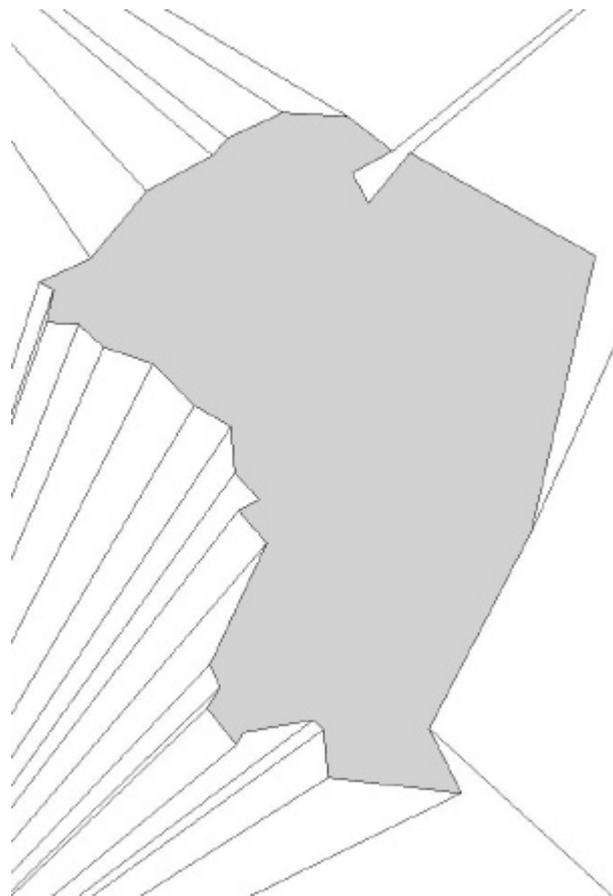


fig 3b: vertices of polygon observed from four corners

For this example a given polygon (the boundary of Annapolis’ country Anne Arunde) is observed from four corners, see fig. 3b. We apply the same algorithm *Scanline_TIN* as in section 3.1. Only the scanlines (connections of target-points and observation-points) are known and used as input. Fig. 3c gives the ‘scanline forced’ Delaunay TIN. Now we perform Step 2 of the algorithm to reconstruct the boundary and interior by a TIN. Again, the resulting TIN is not Delaunay, as some newly formed Edges will cross the original ones.

3.3 The 2.5D Case

The 2.5D Case is not as trivial as the 1.5D or the 2D examples given in the previous paragraphs. Storing and manipulating the surface-points and scanlines by well-known 2D Delaunay TIN constructors could easily solve these cases.

In this 2.5D scenario, the heightvalue of the surface-points could be considered as one possible attribute value of the planimetric co-ordinates (X,Y). As the scanlines do not belong to the surface, we have to use a ‘real’ 3D triangulation construction method, resulting in a Tetrahedronized Irregular Network.

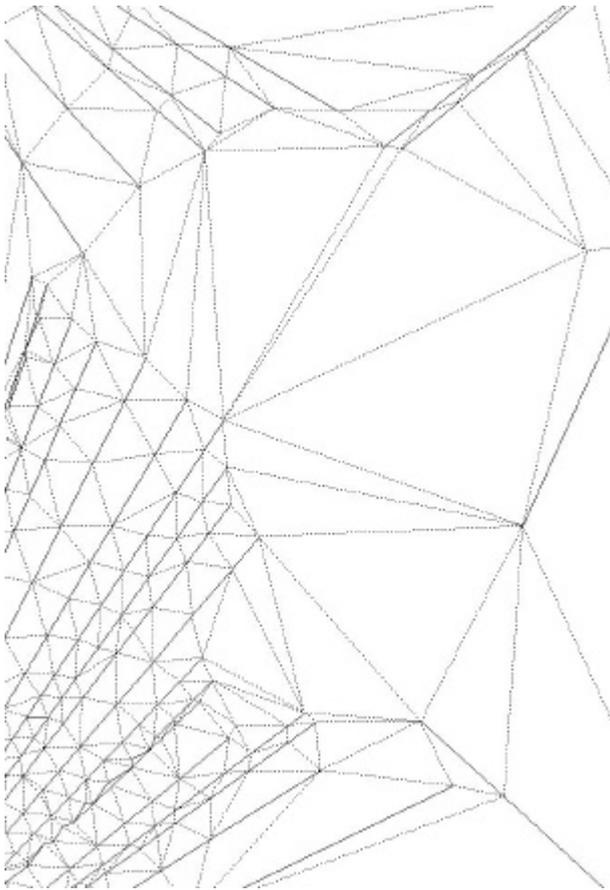


fig 3c: 'scanline forced' Delaunay TIN

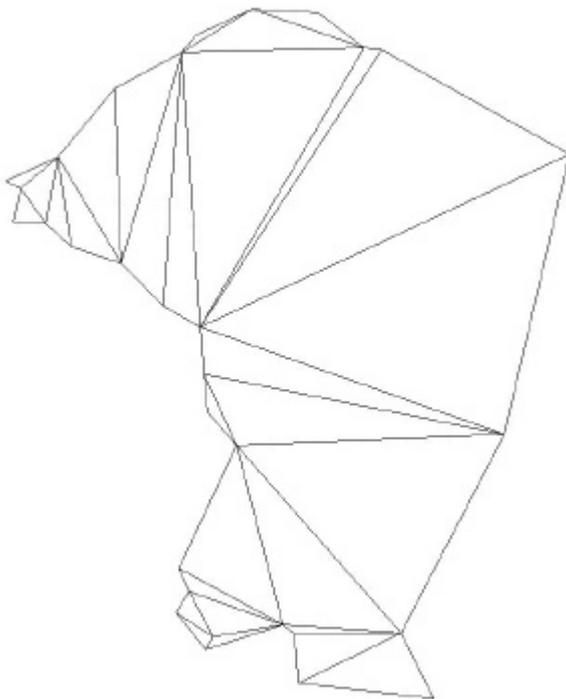


fig 3d: reconstructed boundary and interior of polygon

The triangulation should result in a set of non-overlapping tetrahedrons, which together fill a convex solid. Each network of tetrahedrons should adhere to the following:

- a) Of each tetrahedron its four vertexes should not be located in the same plane;
- b) Each tetrahedron should not contain any other points of the dataset;
- c) A TEN-Face (triangle) is on the boundary of the solid or is exactly shared by two internal tetrahedrons.

To create a set of Delaunay tetrahedrons one condition has to be added:

- d) For each of the tetrahedrons in a Delaunay TEN the circumsphere should not contain any other point of the dataset.

To create a scanline-forced Delaunay TEN the last condition taken into account is:

- e) All scanlines are identified as edges in the Delaunay TEN.

We will apply the same algorithm as given in section 3.1, but all operations are one dimension higher.

Algorithm 'Scanline_TEN'

Step 1: Construct TEN

```
input target-points and scanlines
create 'scanline forced' Delaunay TEN
```

Step 2: Transform Faces

```
for each Face in TEN
  if (two Nodes on Face are target-points)
    then
      if (third Node is on scan-line) then
        get third target-point at end of
          scanline
        construct object-Face
          (first target-point, second target-
            point, third target-point)
      end if
    end if
  end for
```

Step 3: 'Reconstruct Surface

```
for each object-Face-A
  for each other object-Face-B
    if (center object-Face-A below object-
      Face-B)
      then
        remove object-Face-A
      end if
    end for
  end for
```

Step 1: Construct TEN

3D-TIN or TEN construction algorithms are not as common as 2D-TIN constructors, so we can not use a standard GIS-

environment. Recent developments in Computational Geometry Algorithms Library (CGAL, 2001) are promising, but we have used an own 3D implementation of the incremental point algorithm, given in (Lee, 1980).

This same program is used by one of the authors to create animated time series (Kraak, 1992). Here an interface to this program is written in the object-based language Avenue to control the program and visualize the results in ArcView 3.2 (ESRI, 2001).

To be Delaunay, the scanlines are divided until each part is an edge in the TEN, in contradiction to a constrained TEN, where the Delaunay-criterion is loosen to hold the scanline as one part. However, some objects exists that can not be divided into tetrahedrons without adding extra Steiner vertices (Eppstein, 2001).

The obtained Delaunay TEN holds many possible boundary solutions. We will use the added 'Steiner' vertices at the scanlines to reconstruct the surface as in the 1.5D example.

The volume of the object is controlled by an added ground point at a given height value for each target-point.

To illustrate the algorithm the same datapoints as in fig. 1a. are used (see fig 4a).

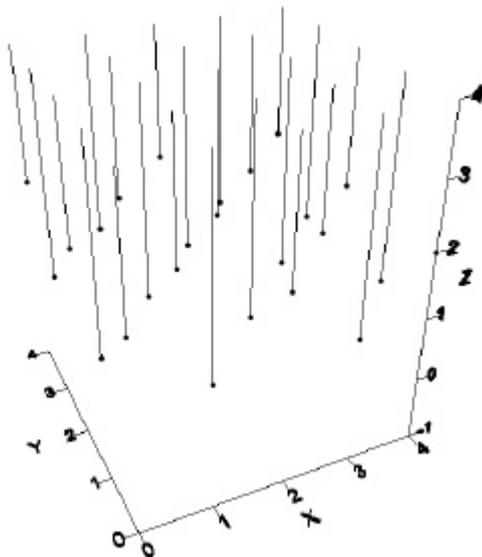


fig 4a: datapoints and scanlines of 2.5D dataset

Step 2: Transform Faces

The actual interior (solid) and boundary (surface) of the object given by the target-points and scanlines is found by the procedure in which for each scanline the connected scanlines are determined by examine the TEN-Faces. All TEN-Faces with two nodes known as a target-points and one node known as an added Steiner point or another target-point are selected. If a TEN-Face links three target-points this TEN-Face is stored, else the TEN-Face is dropped and replaced by a new TEN-Face. This new TEN-Face

is created by the original two target-points and the target point at the end of the scanline belonging to the Steiner point (fig 4b).

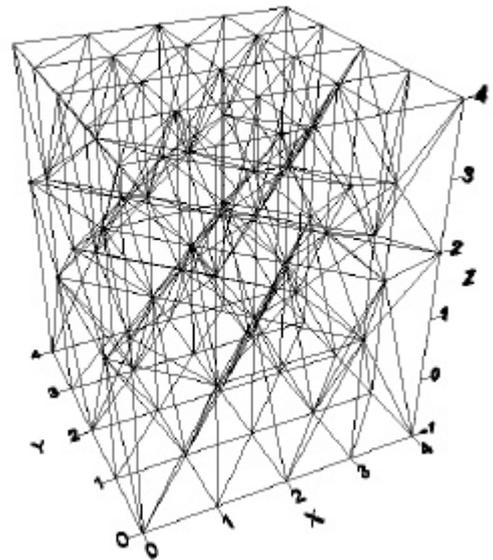


fig 4b: 'scanline forced' Tetrahedron Network (TEN)

Step 3: Reconstruct Surface

Finally a hidden-face removal algorithm finds the surface. Each newly created face is checked whether or not it is above the other faces. If not, this face will be discarded; else it is restored as an surface face.

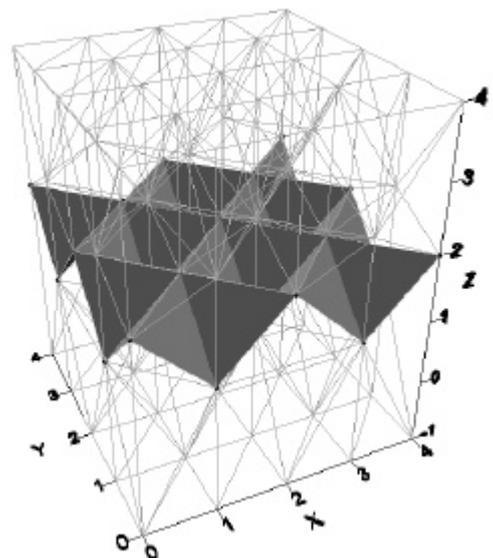


fig 4c: reconstructed surface of 2.5D dataset

If we compare the obtained surface TEN with the 'normal' TIN solution (fig. 4d, fig 1b) we will see the volume below the surface has its maximum obtained.

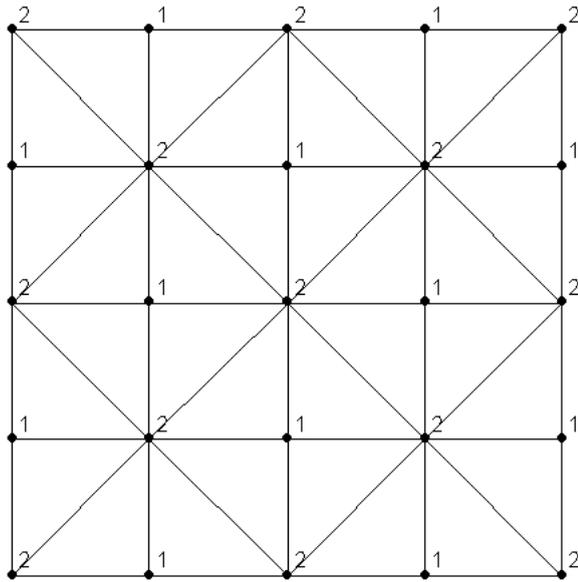


fig 4d: reconstructed TEN-surface

4. CONCLUSIONS

The use of scanline forced TENs gives us the possibility to construct data dependent TINs without the need for predefined and ambiguous data-dependent criteria like surface-area or volume.

The reconstruction of the surface with scanlines is a more complex method than the normal data dependent and data independent algorithms, because of the need to triangulate in three-dimensional space.

The proposed algorithm is capable not only to better reconstruct the surface, but also the volume underneath. This gives us good opportunities for real 3D-calculations (cut-and-fill of caves and other irregular shapes).

5. FURTHER RESEARCH

The surface obtained by the scanline forced Delaunay TENs has to be compared in detail with surfaces obtained by Data Dependent Triangulations for 2.5D datasets with respect to several existing optimization criteria.

The described method for 2.5D surface representation by scanline forced Delaunay TENs has to be tested in further detail for topological correctness (complete and non-overlapping partitioning) of the obtained surface.

As with 2D-TINs the surfaces obtained by the scanline forced Delaunay TENs should be able to store known surface features as breaklines and contourlines.

The (to be derived) angle of the scanline and the obtained surface could give reason to an iterative approach, where only accurate measured datapoints (given by a more or less perpendicular scanline) are taken into account.

The proposed algorithm is able to reconstruct 2D boundaries given a set of surface points and their scanlines (section 3.2). Although directions are given for higher dimensions (compare section 3.1 and section 3.3), this concept has to be proven for 3D-datasets and to be compared to existing methods like the crust and skeleton approach.

REFERENCES

- [Bern, 1992] M. Bern, Herbert Edelsbrunner, David Eppstein, Scott A. Mitchell, Tiow-Seng. *Edge Insertion for Optimal Triangulations*, in: Proc. Latin Amer. Sympos. Theoret. Informatics (LATIN '92).
- [CGAL, 2001] www.cgal.org
- [Dyn, 1990] N. Dyn, D. Levin and S. Rippa. *Data dependent triangulations for piecewise linear interpolation*. IMA J. Numer. Anal. 10 (1990), pp. 137-154.
- [Eppstein, 2001] www.ics.uci.edu/~eppstein/junkyard/untetra/
- [ESRI, 2001] www.esri.com
- [Kraak 1992] Menno-Jan Kraak and Edward Verbree. *Tetrahedrons and animated maps in 2D and 3D space*. In: Proceedings of the 5th Int. Sym. On Spatial Data Handling, Charleston, USA, pp. 63 – 71.
- [Fleischmann, 1999] Peter Fleischmann. *Mesh Generation for Technology CAD in Three Dimensions*. Dissertation Technische Universität Wien, 1999
- [Lawson, 1977]: C.L. Lawson. *Software for C¹ Surface Interpolation*, in: Rive, J. (Ed.) *Mathematical Software III*, pp. 161-194.
- [Lee, 1980] Lee, D.T. and Schachter, B.J. *Two algorithms for constructing a Delaunay Triangulation*. International Journal of Computer and Information Sciences, vol. 9, no. 3, pp. 219-242
- [Lenk, 2000] Ulrich Lenk, *Optimisation Criteria for Degenerated Delaunay Triangulations*, proceedings GIScience 2000, Savannah, Georgia, USA, October 28-21, 2000
- [Yu, 2001]: Xiaohua Yu, Bryan S. Morse, and Thomas W. Sederberg, *Image Reconstruction Using Data-Dependent Triangulation*, IEEE Computer Graphics and Applications, May/June 2001, pp62-68
- [Tinghua 2001]: Tinghua Ai, *Extraction of landform structure lines based on vector band analysis*, to be publicized, 2001
- [Zhang, 2001] www.cim.mcgill.ca/~yzhang/skeleton.htm