THE STIN METHOD: 3D-SURFACE RECONSTRUCTION BY OBSERVATION LINES AND DELAUNAY TENS

Edward Verbree, Peter van Oosterom

Delft University of Technology, section GIS-technology
{E.Verbree, P.J.M.vanOosterom}@citg.tudelft.nl

Commission III, WG 3

KEY WORDS: Conforming Delaunay Tetrahedronized Networks, Surface Reconstruction, Data Dependent Triangulations

ABSTRACT:

A proper representation of the surface of the earth and the man-made object build upon it is needed as a data source for environmental modelling and planning. One way to represent the terrain given by a set of surface points is to construct a Delaunay Triangular Irregular Network (DTIN). This DTIN is believed to give the 'best' triangular tessellation as the Delaunay empty circle criterion opts for well-formed 'fat' triangles and the resulting triangulation maximizes the smallest angle within each triangle. This idea is true for many computational geometry applications, but it is not valid for visual and analytical geo-computational queries dependent on the height of the surface. This limitation is given by the fact that the distribution of the triangular mesh is defined in the two-dimensional XY-plane and the Z-value of the surface points is not taken into account by the Delaunay empty circle criterion at all. Alternatively, Data Dependent Triangulations (DDTINs) aim to identify which triangulation 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. The Z-value of the surface points is now taken into account, but still no certainty can be given that the derived TIN represents the actual surface. Hence, the reconstruction of the surface given by only the set of surface points is not unambiguous.

This paper describes a surface reconstruction method based on the Delaunay Tetrahedronised Irregular Network (DTEN), which tessellates the 3D-space with non-overlapping, adjacent, tetrahedrons. The DTEN is constructed by the Delaunay criterion, resulting in a tessellation where the circumscribing sphere of each tetrahedron is empty. The approach presented in this paper is new in that not only the surface points are included into the DTEN, but also the observation lines, i.e. the lines-of-sight between the observer (i.e. an airborne or tripoded laser altimeter) and the targets (the measured points). These observation lines add the information needed to extract the Surface TIN (STIN) from this DTEN. The observation lines can also be artificial or simulated for this purpose. The STIN approach presented in this paper is a full 3D-implementation and refinement of the research presented in (Verbree, 2001).

1. INTRODUCTION

1.1 Limitations on Delaunay Triangulations

TINs are commonly used for surface representation. Given target points on the surface a Triangulated Irregular Network is created. The Z-value of the point features is stored as the Z-value of the nodes of the computed TIN. A Delaunay TIN fulfils the 'empty circle criterion'. This criterion opts for the triangulation with 'fat' triangles, such that the triangulation maximizes the smallest angle within each triangle (Figure 1.1 and 1.2).

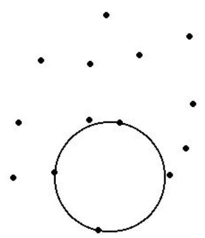

Figure 1.1
Empty circle criterion

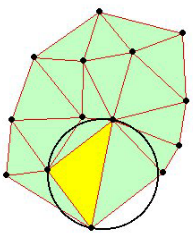

Figure 1.2
Resulting Delaunay TIN

We have to realise however that the 'empty circle criterion' does not take the Z-value of the features into account at all. This is clearly seen if the point distribution is square, as in the following example (Figure 1.3 and 1.4). In these figures 25 target points are given, with an alternating Z-value of 1 or 2.

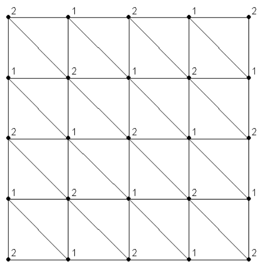

Figure 1.3
One possible Delaunay TIN

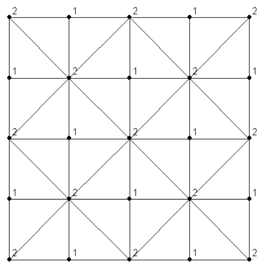

Figure 1.4
Another Delaunay TIN

In Figure 1.3 the diagonal of all triangles is directed north-west to south-east. The four points on a square are on the same circle, so it is with the Delaunay criterion in mind, free to choose a direction for the diagonals, as there is no optimum for the min-max angle criterion. This could be north-east to south-west for all diagonals, more or less data dependent distributed as in Figure 1.4 or even complete randomly distributed. Which one to choose?

1.2 Limitation on Data Dependent Triangulations

The height values of the target points (or the Z-values of the nodes in the DTIN) do have consequences for derivatives like slope and aspect, visualization (hill-shading) and volume statistics (view sheds, and cut and fill calculations). One can argue that the 2D-Delaunay TIN (the triangulation of a 'flat' surface) is just one of the possibilities to triangulate a set of points and lines. In fact, any triangulation can be a candidate for a 2.5D terrain surface representation.

Another approach is to take the Z-value of the target points into account in the triangulation process. Extensive research on Data Dependent Triangulations (DDTINs) proves this observation (Alboul, 2000). The idea behind this concept is to maximize or to minimize some cost functions that express certain local, regional or global properties of the resulting surface (Dyn, 1990; Lenk, 2000). Possible options for this cost functions are: minimize the surface area, minimize the volume, minimize the maximum angle of the surface triangles, etc. But it has to be stated that pure DDTINs can lead to a large amounts of sliver triangles, which gives an artificial result. An example taken from (Alboul, 2000) is shown in Figure 1.5. Furthermore the local and global criteria could disregard certain phenomena, like ridges and faults.

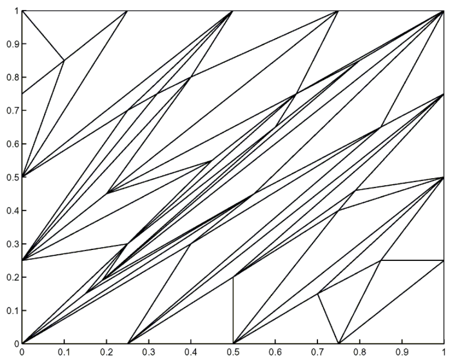

Figure 1.5
DDT of minimising the absolute mean curvature

1.3 The STIN-method; in search for improvement

Both Delaunay and Data Dependent Triangulations are 2.5D surface reconstruction techniques given a discrete 2.5D data set. This limitation is suitable for most terrain applications, but no overhanging cliffs or other disturbances are possible. Reconstructing the surface of caves, buildings or other full 3D-phenomena are only possible for parts of the data set, which first has to be projected to a suitable XY-plane. Therefore a full 3D representation, like the Delaunay Tetrahedronized Irregular Network (DTEN), could be considered. Within this DTEN many, many surfaces through the data points are embedded. A little trick is needed to select the 'best' surface.

One way to retrieve this surface is examining the data acquisition process. The surface point is determined by the position of the observer and the direction and distance of the measurement or observation. So, for each data site the position of the observer (observation point) is known. In case of airborne laser altimetry each target point has one corresponding observation point. Terrestrial laser scanning will result in a large set of target points measured from one or more observation positions. And although unlikely the possibility exists a surface point is measured from two or more observations points. But each pair of target and observation points are connected by one unique observation line.

The line of sight (observation line) between target and observer should be free of obstacles or penetrated by a laser beam, otherwise no measurement can be made. The STIN method takes these observation lines into account in the surface reconstructing process. The observation lines are split with Steiner points until each part of the observation line recurs as an edge in the DTEN. This so-called conforming DTEN (Shewshuk) gives in conjunction with the Steiner points enough information to reconstruct the surface.

In the next chapter the 2.5D surface reconstructing process based on these ideas are described by many figures and examples. Chapter 3 gives applications for the STIN algorithm for 3D surface reconstructions. Chapter 4 will end up with conclusions and recommendations for further research.

2. THE STIN METHOD FOR 2.5D SURFACES

2.1 Input target points and observation lines – Figure 2.1

The STIN-method for 2.5D surfaces is illustrated by a basic example where the observation lines of the 2.5D target points are dropped perpendicular from a certain height, see figure 2.1. The algorithms consist of several steps, described in the following sections. These steps are:

Step 1: Input target points and observation lines
Step 2: Construct Conforming DTEN
Step 3: Transform TIN_Edges to Volume_Edges
Step 4: Find STIN_Edges on Surface
Step 5: Create STIN_Faces

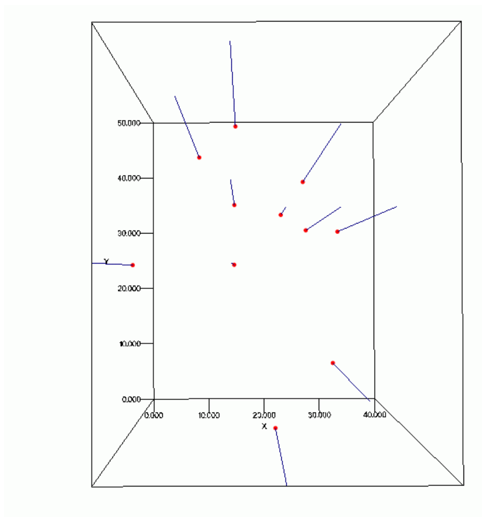

Figure 2.1
Target points and their observation lines – 2.5D case

2.2 Construct Conforming DTEN – Figure 2.2

The target points and their observation lines are included into a conforming Delaunay TEN. This 3D-network should result in a set of non-overlapping adjacent tetrahedrons, which should adhere to the following rules:

1) For each of the tetrahedrons in a Delaunay TEN the circumsphere should not contain any other point of the data set.

2) All observation lines are identified as edges in the Delaunay TEN.

The Delaunay TEN is calculated based on the incremental algorithm (Watson, 1981) and (Bowyer, 1981). This algorithm adds one point at a time to an (initial) valid Delaunay Triangulation. This algorithm is also known as the cavity algorithm, since its deletes all tetrahedrons that are not longer empty after the intersection of the new point. This cavity is tetrahedronized again by connecting the newly inserted point to all vertices on the cavity boundary. This procedure is available as an independent TEN-constructing program, as used in (Kraak, 1992).

To remain the observation lines within the TEN two possibilities are well-known, the constrained TEN and the conforming TEN. Within the constrained TEN the empty circumsphere requirement is loosed to allow an incorporation of the lines within the TEN as an entity. Conforming TENs on the other hand allow the insertion of so-called Steiner points. These extra points are iteratively added on the midpoints of the observation lines until each (part of) the observation line can be identified by an edge of the Delaunay TEN. Here is chosen for the Conforming TEN procedure, because of its minimalist approach (Calvalcanti, 1999), but also because of the use of these Steiner points in reconstructing the surface.

The 2.5D Volume beneath the target points and thus the Surface are found by the procedure, given by the code in the next step. Within this step a little trick is applied to derive the volume edges.

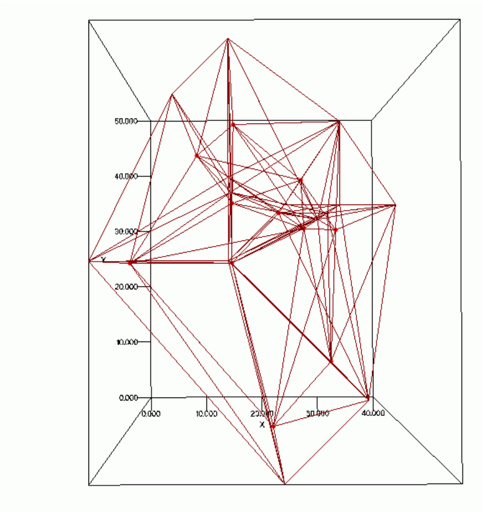

Figure 2.2
Conforming Delaunay TEN

2.3 Transform TIN_Edges to Volume_Edges – Figure 2.3

All TEN_Faces are examined. If a TEN_Face has one Steiner point and two target points the algorithm will replace the Steiner point by the target point at the end of the observation line. The TEN_Edges of this TEN_Face are stored as Volume_Edges. Also the TEN_Edges of the TEN_Faces with three target points are stored as Volume_Edges and the remaining TEN_Faces are discarded. These set of Volume_Edges represents the Body of the object defined by the 3D-convex hull of the target points. The complete set of Volume_Edges can be linked together to a TEN, which is partly Delaunay. The 'top' of the Body, as visible from the observation points, defines the Surface TIN we are looking for.

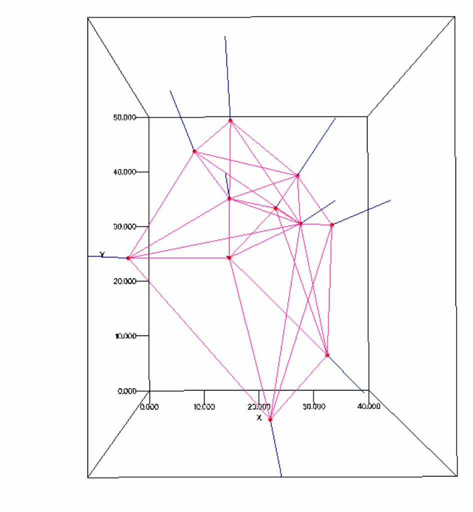

Figure 2.3
Volume_Edges of Conforming DTEN

2.4 Find STIN_Edges on Surface – Figure 2.4

In this step a hidden edge removal algorithm is applied on the Volume_Edges to retrieve the STIN_Edges. The algorithm applied projects the Volume_Edges to 2D and tests each one with the intersecting Volume_Edges. The intersection point is calculated in 2D, and the algorithm continues with calculation of the Z-value of the Volume_Edges at the intersection point. The Volume_Edge with the lowest Z-value is the furthest away from the observer and therefore not at the surface. This one is removed from the Volume_Edges. All remaining edges are declared to be STIN_Edges.

A problem arises in that some removed Volume_Edges are to be considered as STIN_Edges to obtain a complete and valid STIN_Surface. The removed Volume_Edges that have no 2D-intersection with another removed Volume_Edge are therefore promoted to STIN_Edges.

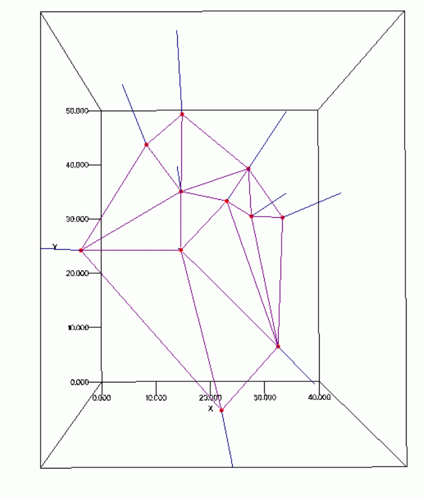

Figure 2.4
TEN_Edges on Surface

2.5 Create STIN_Faces – Figure 2.5

Finally the STIN_Faces (Surface triangles) are constructed. The STIN_Edges on the surface gives a complete and non-overlapping triangulated partitioning of the surface. The internal numbering of the nodes of the STIN_Edges gives enough information to construct the STIN_Faces.

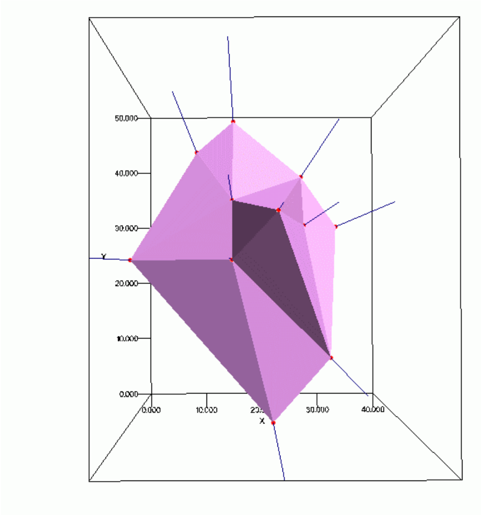

Figure 2.5
TEN_Faces on Surface

3. THE STIN METHOD FOR 3D SURFACES

3.1 Extending the STIN method to 3D – figure 3.1

The STIN method is described and explained with an example data set in 2.5D. The only step in the STIN method, which uses this property, is the hidden edge algorithm to find the STIN_Edges on the surface (Section 2.4). That fast algorithm could be applied because the observer was thought to be far above the scene and thus could the observation lines, like airborne laser scanning, be considered perpendicular.

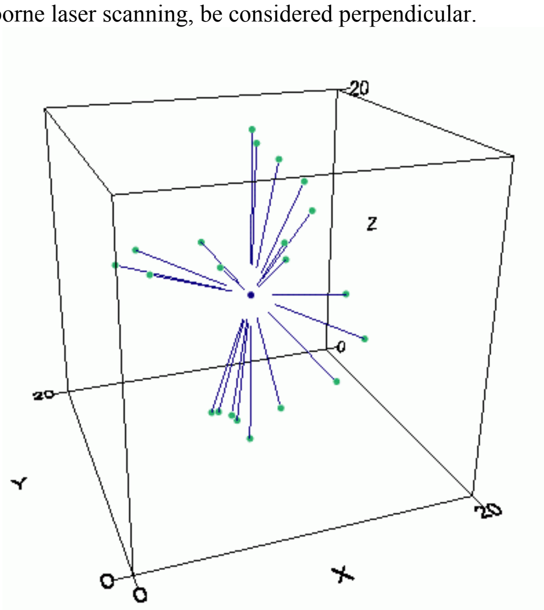

Figure 3.1
One observation point, many target points - 3D case

If the observer is more or less within the scene, like in the case of terrestrial laser scanning, another hidden edge algorithm should be applied. This modification is necessary, because the target points are now really distributed in 3D. But that is the only modification. The remaining algorithm is not affected. This is demonstrated by the following example, where one observer from within the object scans several target points around it in 3D space.

3.2 Construct Conforming DTEN – Figure 3.2

The observation lines has to be cut off at a certain distance from the observer, for sake of unwanted site-effects (unlimited addition of Steiner points) in the calculation of the conforming DTEN.

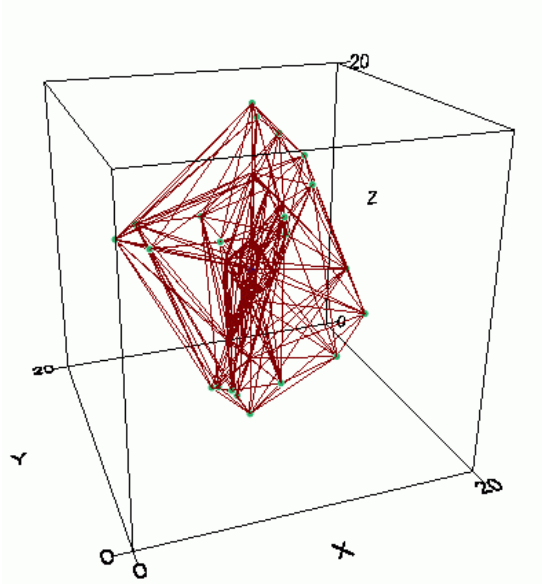

Figure 3.2
Conforming Delaunay TEN

3.3 Transform TIN_Edges to Volume_Edges – Figure 3.3

The same algorithm as in Section 2.3 is applied.

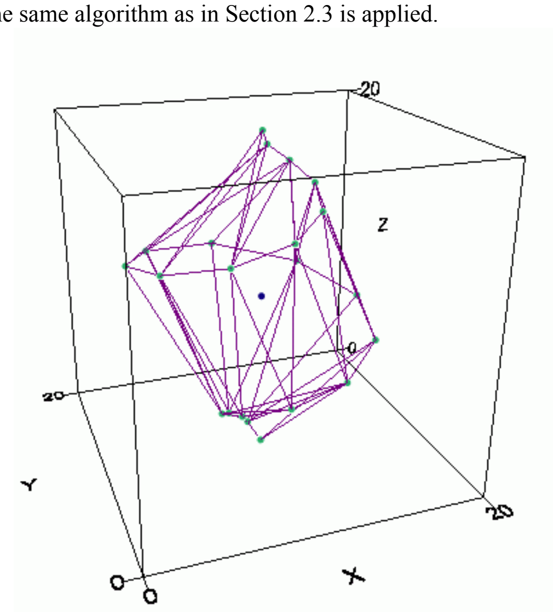

Figure 3.3
Volume Edges of Conforming DTEN

Again all TEN_Faces are examined and transformed if one of the nodes is an added Steiner point. In that case the target point at the end of the observation line replaces this node. All TEN_Edges of the TEN_Faces are now stored as Volume_Edges.

3.4 Find STIN_Edges on Surface – Figure 3.4

To determine which TEN-Edges are on the surface a full 3D hidden edge algorithm had to be applied. This algorithm is a straightforward three dimensional generalization of the 2D method presented in (Aftosmis) and described by (O'Rourke, 1994). Figure 3.4 gives an illustration of this method.

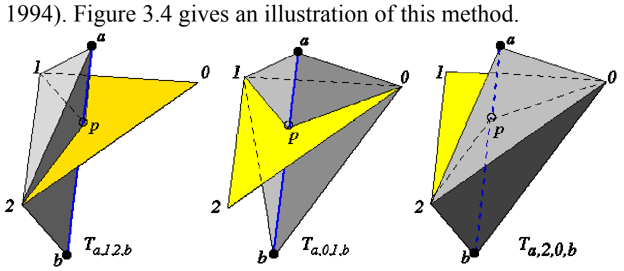

Figure 3.4
Test visibility Edge (a,b) and Edge (1,2) from Observer (0)

Each Volume_Edge is tested against all other Volume_Edges. To determine whether or not Volume_Edge (a,b) is in front of Volume_Edge (1,2) given the observation from point (0) three tetrahedrons T(a,1,2,b), T(a,0,1,b) and T(a,2,0,b) are constructed. The Volume_Edge (a,b) is in front of Volume_Edge(1,2) if the sign of the determinant of these tetrahedrons is the same (all positive or all negative). These Volume_Edges are eliminated and the remaining Volume_Edges are on the surface and declared as STIN_Edges.

Again (as in the 2.5D example) some of the removed Volume_Edges are needed to obtain a complete and valid STIN_Surface. Given the set of visible Volume_Edges the eliminated ones are examined. One case one of these is visible it will be restored and declared as a STIN_Edge.

3.5 Create STIN_Faces – Figure 3.5

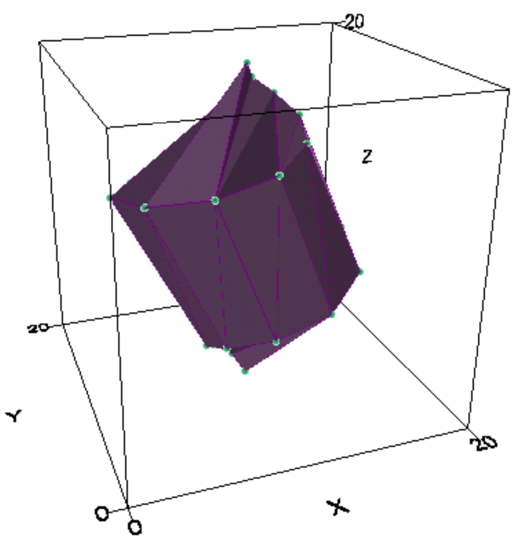

Figure 3.5
TEN_Faces on Surface

The set of STIN_Edges gives a complete and non-overlapping partitioning of the Surface and all STIN_Faces are constructed. However, one extra test is necessary. Although the observation lines enforces the selecting of the STIN_Edges, it is possible that a triple of STIN_Edges constructs a STIN_Face, which is intersected by an observation line. Each constructed STIN_face is tested against intersecting by an observation line and deleted in case of.

3.6 Examples – Figures 3.6 and 3.7

In this example all steps of the STIN method are made visible. Given are the eight target points on the corners of a cube and six target points slightly pushed inside the cube. The quest is to reconstruct the surface, given an observation point. In figure 3.6 this observer is in the mid of the cube and the STIN Surface is found given the procedure described in the former sections.

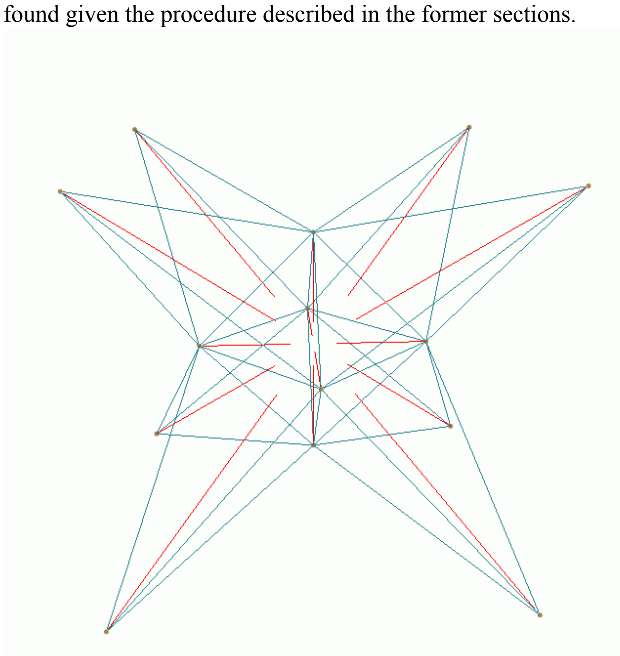

Figure 3.6
STIN Surface of 'Pushed' Cube with observation inside

Figure 3.7 shows the reconstructed STIN Surface of almost the same dataset (one of the pushed target points is eliminated to give sight inside the cube) and the effect of the position of the observer (slightly below the position of the eliminated target point). The observer is now outside the object and a complete different - but still valid - surface is derived.

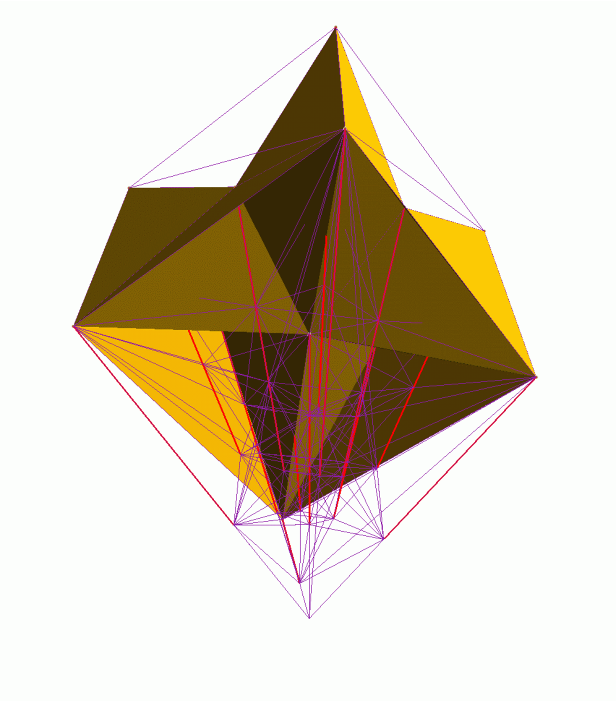

Figure 3.7
STIN Surface of 'Pushed' Cube with observation outside

4. CONCLUSIONS AND RECONMANDATIONS

The standard Delaunay TIN (DTIN) method has to be handled with care for surface reconstruction purposes, as the Z-value of the target points is not considered in the construction. Within the Surface TIN (STIN) method the Z-value of the target points is taken into account along the position of the observer and the observation lines. The surface is created and derived within a Tetrahedronised Irregular Network (TEN) in three dimensions. This method lines up with all kinds of Data Dependent Triangulations (DDTINs). The STIN method is capable to reconstruct surface out of a given point cloud in 3D as long as the location of the observer is known.

Current research is undertaken to extent the method to:

- Handle large data sets. The STIN method is now available in a prototype environment written in the scripting language Avenue of ArcView 3.2a (ESRI) and own TEN-constructing software. It is possible to create surfaces to 1000 points in reasonable time. The use of a more robust and scalable environment as the Computational Geometry Algorithm Library (CGAL) should be considered.
- Handle contouring datasets. The obtained surfaces of this kind of dataset are notorious for their problems with flat triangles and missing ridges and vaults when triangulated with a Delaunay triangulation. The idea is to retrieve Surface TIN based on locale constructed TENs instead of one TEN for the entire dataset.
- Handle more observation points for the 3D-surface reconstruction procedure. This will introduce some complexity in the algorithm as observation lines will cross and possible intersect each other. But the potential to reconstruct 3D-surface measured from two or more tripoded laser altimeters is promising.

Furthermore research is undertaken to:

- Give a formal proof of the correctness of the reconstructed surfaces (no holes or overlapping parts).
- Compare the 2.5D STIN surfaces into detail with results from Data Dependent Triangulations.
- Combine 2.5D surfaces with full 3D objects for terrain modelling applications. The 3D objects will be retrieved by the full 3D-STIN method, while the surface is reconstructed by the 2.5D-STIN method, where after these models have to 'glued' together to one 2.5D/3D datamodel.

ACKNOWLEDGEMENTS

The authors of this paper would like to express their appreciations to the valuable remarks of the reviewers.

REFERENCES

Aftosmis, M.J., Intersection of Generally Positioned Polygons in R3, http://people.nas.nasa.gov/ ~aftosmis/cart3d/bool_intersection.html

Alboul, Lyuba, G. Kloosterman, C. Traas, R. van Damme, Best data-dependent triangulations, Journal of Computational and Applied Mathematics 119 (2000), pp. 1-12

Bowyer, A., Computing Dirichlet tessellations, Computer Journal, 24(2):162–166, 1981.

Calvalcanti, P.R. and Ulisses T. Mello, Three-dimensional Constrained Delaunay Triangulation: a Minimalist Approach, Proceedings of the 8th International Meshing Roundtable, Lake Tahoe, California, October 1999, p.119-129

CGAL, www.cgal.org

Dyn, N., D. Levin and S. Rippa. Data dependent triangulations for piecewise linear interpolation. IMA J. Numer. Anal. 10 (1990), pp. 137-154.

ESRI, www.esri.com

Kraak, M.J., and Verbree, E., Tetrahedrons and Animated Maps in 2D and 3D Space, in Proceedings 5th International Symposium on Spatial Data Handling, 1992, pp. 63-71.

Lenk, U., Optimisation Criteria for Degenerated Delaunay Triangulations, proceedings GIScience 2000, Savannah, Georgia, USA, October 28-21, 2000.

O'Rourke, J., Computational Geometry in C. Cambridge University Press, New York, 1994.

Shewchuk, http://www-2.cs.cmu.edu/~quake .defs.html#cdt

Verbree, E. and P.J.M. van Oosterom, 2001, Scanline Forced Delaunay TENs for surface representation, ISPRS Workshop on Land Surface Mapping and Characterization using Laser Altimetry, Annapolis, Maryland, USA, pp. 45-51.

Watson, D.F., Computing the n-dimensional Delaunay tesselation with application to Voronoi polytopes. The Computer Journal, 24(2):167–172, 1981.