

An Original Way of Building a TIN with Complex Urban Structures

Rebecca, O.C., Tse ^{a,*}, Christopher Gold ^b and Dave Kidner ^a

^a School of Computing, University of Glamorgan, Mid Glamorgan, Wales, CF37 1DL, UK –

rtse@glam.ac.uk*, dbkidner@glam.ac.uk

^b Department of Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong SAR, China -

lscgold@polyu.edu.hk

Commission VI, WG VI/4

KEY WORDS: TIN, CAD, Euler Operators, Quad-Edge data structure

Abstract:

Geographical Information System (GIS) software often models a Digital Terrain Model (DTM) by using triangulations (TINs). Computer Aided Design (CAD) is then used to model complex city structures and superimpose them on the TINs. Since the terrain surface is not connected to the built-up structures, no further spatial analysis (for example, neighbourhood analysis, flow modelling) is possible. Bridges and tunnels are parts of complex urban structures which may be difficult or impossible to be modeled in those TINs.

An original way of building the TIN model may help to solve this problem. The use of the “boundary-representation” structure from CAD systems, the triangulation structure of TINs and the Quad-Edge structure from Computational Geometry are combined to alleviate the problem. The Quad-Edge structure forms the base of the model and CAD-type Euler Operators are the intermediate level operators for building the TIN model. To extend the TIN model with bridges, holes and buildings, additional Euler Operators are used. This original way of building the terrain model not only permits the extrusion of buildings and the formation of bridges and tunnels, but also the development of topological queries and analysis, for example, runoff modelling. It is a simple and reliable method and is illustrated with examples of interactive modelling of part of the Hong Kong city.

1. INTRODUCTION

The TIN has been a well-known model for more than 20 years in the GIS world. Most of the current GIS software visualizes and models the terrain surface by using TINs. However the traditional TIN is a 2.5 dimension (2.5D) model which cannot handle non-planar networks and complex structures, for example bridges, tunnels and buildings, from the real world. Existing GIS systems make use of advanced technology from CAD, like 3D graphics and objects created by 3D Studio MAX (CAD software). They model the terrain surface by using triangulation, and then 3D objects from CAD systems are created and superimposed on top. Database matching is used to locate those 3D objects, but no spatial analysis can be implemented. Will it be possible to make use of the advanced technology from CAD to build a connected TIN model for further spatial analysis?

This paper aims at alleviating this problem by extending the traditional TIN model. First, a terrain surface model should be recognised as a more than a two-dimensional entity – it is an air (or water-) earth interface, the boundary between the “Polyhedral Earth” and the exterior (Tse & Gold, 2002). An underneath surface is put on every TIN to give a polyhedral model. We are interested in the simplicity of the Quad-Edge data structure (Gold, 1998, Guibas & Stolfi, 1985) and the well-developed tools for solid model representation from CAD systems (Baumgart, 1975, Mantyla, 1981).

Since the properties of the triangulation look similar to the boundary-representation (b-rep) structure from CAD, the CAD-type b-rep structure Euler Operators are used to create and

extend the TIN model with holes. The Quad-Edge structure forms the base operators of the model. Euler Operators acts as middle level operators to implement the basic triangulation functions. Finally the TIN model is extended by using additional Euler Operators while the topological connectivity is preserved.

The work has been carrying out with five different stages (Tse & Gold, 2002):

1. Definitions of three levels of operators to achieve our desired system;
2. The implementation of CAD-type Euler Operators using the Quad-Edge data structure;
3. The implementation of basic TIN functions using the Euler Operators
4. The extension of the TIN model with additional Euler Operators
5. The implementation of the extended TIN model with real data (Hong Kong map data)

2. EXISTING GIS SOFTWARE

Most of the existing GIS software create and display the terrain surface by using triangulation and the 3D objects are put on top of the existing triangulation. Fig 1 shows an example from ArcView ESRI® of creating and visualising the DTM and fig 2 shows some 3D objects (buildings) pasted on top of the DTM, which looks like playing “LEGO™”. Further spatial analysis can be performed because of a lack of topological connectivity. Since the DTM and the buildings are not in the same layer, it

may be difficult or impossible to model bridges and tunnels on the terrain surface.

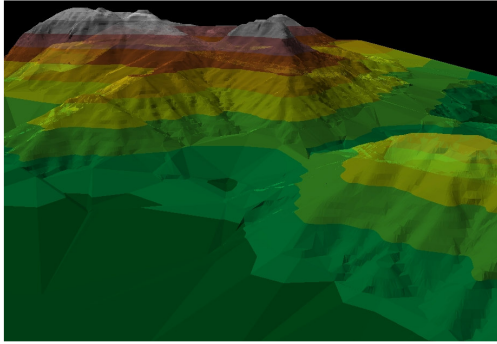


Figure 1 A DTM from ArcView

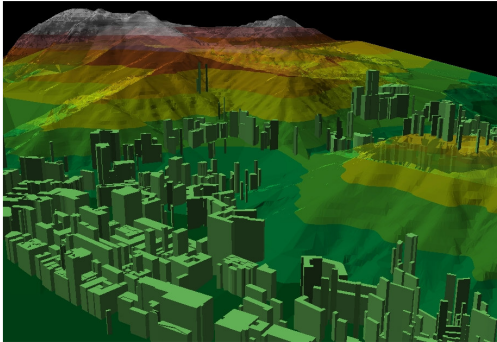


Figure 2 A DTM with buildings

3. DIFFERENT LEVELS OF OPERATORS

3.1. Why these operators?

CAD modelling techniques do not seem attraction at first. The Boolean operators, the union, intersection, and difference of cubes and tubes in Constructive Solid Geometry structure (CSG) are hard to apply to terrain models. The data structure for modelling a non-manifold solid is too complex (Lee, 1999), therefore only manifold-based b-rep CAD models will be considered in enhancing the TIN model. Euler Operators, with the help of the Euler Poincaré formula is a good indicator of the topological validity of the boundary surface (Mantyla, 1981). Euler Operators are used to ensure the integrity of the extended TIN models, since the TIN is a set of triangles represented by a group of boundary edges and vertices.

Implementations of Euler Operators used various kinds of edge-type data structures, for example, half-edge (Weiler, 1985) and winged-edge (Baumgart, 1972) data structures. They seem too complicated for modelling triangulations, and the TIN does not need to represent holes in an individual triangle. The simplicity of the Quad-Edge structure (Guibas & Stolfi, 1985) is used to implement the Euler Operators. Therefore we use the Quad-Edge as the basic structure used to implement a set of Euler Operators in the TIN model. The implementation of triangulation operators using the Euler Operators will be the highest level.

3.2. The Quad-Edge Data Structure

Weiler (Weiler, 1985) gave a topology representation with enough information to recreate nine adjacency relationships without error or ambiguity. Figs 3 and 4 show the only two operators, "Make-Edge" and "Splice", used on the Quad-Edge

structure. Make-Edge creates an individual edge with four connected "Quad" objects and every Quad has three pointers. Splice splits a face into two pieces or merges two faces into one.

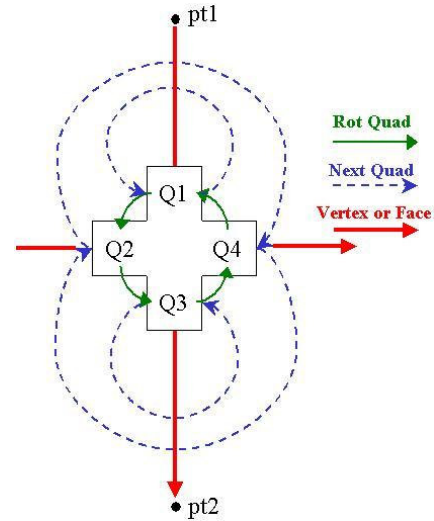


Figure 3 Make-Edge

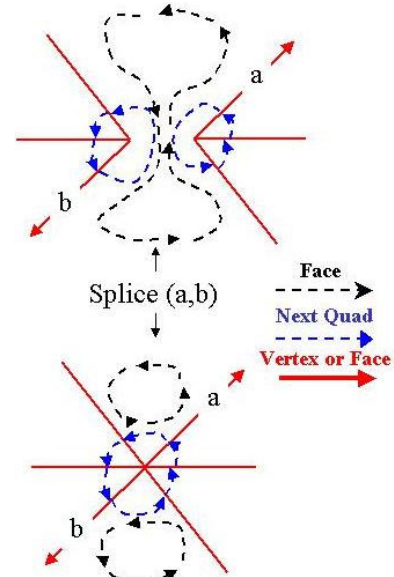


Figure 4 Splice

3.3. The Implementation of Euler Operators Using Quad-Edge Structure

According to Braid (Braid et al., 1978) five spanning Euler Operators suffice to create a valid b-rep model. The six elements of every b-rep solid model are: vertices, edges, faces, loops (or rings), holes and bodies (or shells), however loops (a hole in an individual face) will not be considered in the TIN model. Therefore only five elements will be used in the extended TIN model. Thus four spanning Euler Operators suffice for modelling the extended triangulation.

Similarly three Euler Operators are needed to create a simple TIN model without any holes or bridges because only four elements are including in the model. They are: vertices, edges, faces and bodies. Figs 5 to 7 show the three operators for creating the TIN model. In the figures below, the dash lines represent the connectivity between edges in a face or a vertex. Every edge is represented by one of the four Quads which is

named in the following figures. For example, in fig 9 the name “N1” represents the whole edge with four invisible “Quad” objects.

3.3.1 MEVVFS $\Leftrightarrow$ KEVVFS: Make Edge Vertex Vertex Face Shell (MEVVFS) in fig 5 is an initial operator to create an object (body or shell) with an edge and two vertices (points “pt1” and “pt2”), and its inverse operator Kill Edge Vertex Vertex Face Shell (KEVVFS) kills the only edge on an object.

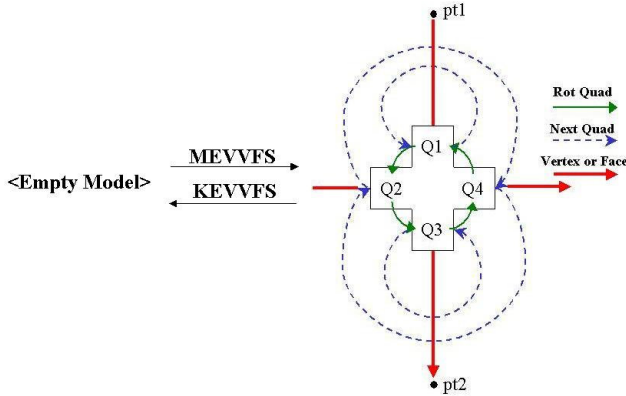


Figure 5 MEVVFS $\Leftrightarrow$ KEVVFS

3.3.2 MEF $\Leftrightarrow$ KEF: Make Edge Face (MEF) splits a face by creating an edge and a face, and its inverse Kill Edge Face (KEF) merges faces by killing a face and an edge. Edge “e” is created to split the face into two pieces and two faces merge by killing this edge.

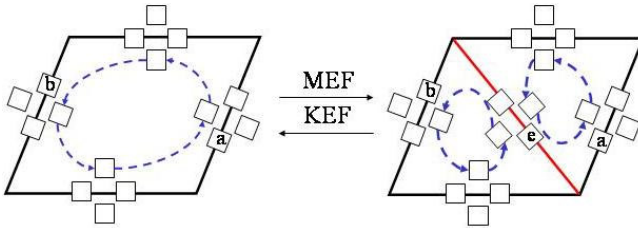


Figure 6 MEF $\Leftrightarrow$ KEF

3.3.3 SEMV $\Leftrightarrow$ JEKV: Split Edge Make Vertex (SEMV) in fig 7 splits an edge by adding a vertex and an edge (edge “a”), and its inverse Join Edge Kill Vertex (JEKV) joins two edges by killing a vertex and an edge. SEMV is acting as an operator to reshape the polygon into a triangle. Edge “a” and point “pt” are added to split edge “e” into two pieces, and the polygon is reshaped into triangle.

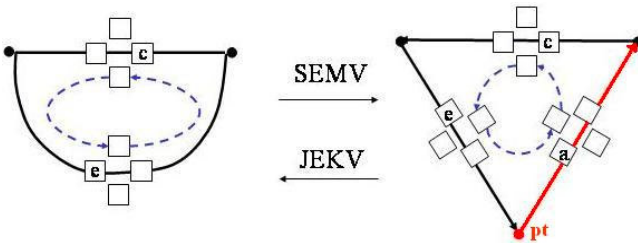


Figure 7 Split Edge Make Vertex

3.4. The Implementation of Triangulation Operators Using The Euler Operators

Big Triangle, Insert Point and Swap are the three operators used to create a triangulation, as show in figs 8 to 10. All of the triangulation operators have their inverses.

3.4.1 Big Triangle: Big Triangle is an initial operator for creating the first triangle and it should be big enough to contain all the data points. To create the first triangle in the triangulation, three Euler Operators are needed: “MEVVFS”, “MEF” and “SEMV”. “MEVVFS” creates the first edge “e1” with two points “pt1” and “pt2”. “MEF” creates a face by adding edge “e3”. “SEMV” splits edge “e3” into two pieces and reshapes the polygon into a triangle. To kill the big triangle, “JEKV” joins edge “e2” and “e3”. “KEF” kills edge “e2” and a face. “KEVVFS” kills the last edge “e1”, returning to an empty space.

3.4.2 Insert Point: Suppose the triangle which contains the new inserted point is located in fig 9. A new edge “N4” is created by using “MEF” and reshaped to point “pt” by using “SEMV”. “MEF” is used again to split the face with an edge “N6”. Three new edges and a point are created inside the triangle. Delete Point is the inverse operator. “KEF” and “JEKV” are used to kill the point inside the triangle.

3.4.3 Swap: Swap is used to swap a common edge between two triangles. The “In-circle” test is used to test the triangle in a Delaunay Triangulation and the swap operator is used to change the failed edges. The empty circumcircle criterion is described in (Guibas & Stolfi, 1985). Input edge “e”, the common edge between two triangles, is swapped in fig 10. “KEF” kills edge “e” and “MEF” creates edge “e” again.

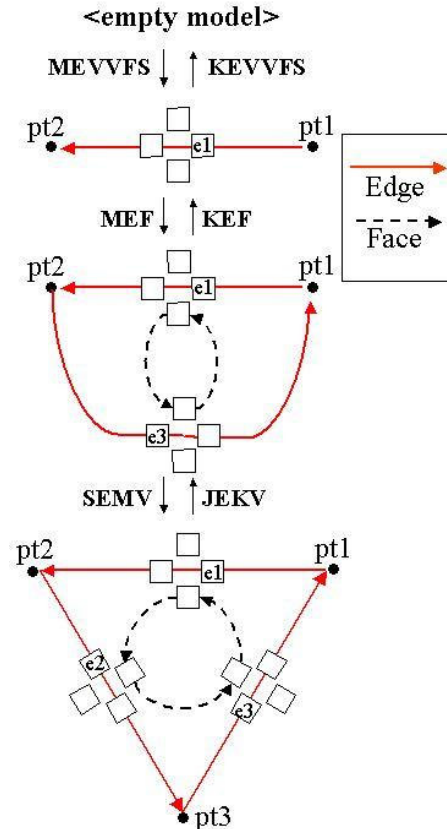


Figure 8 Big Triangle Operator

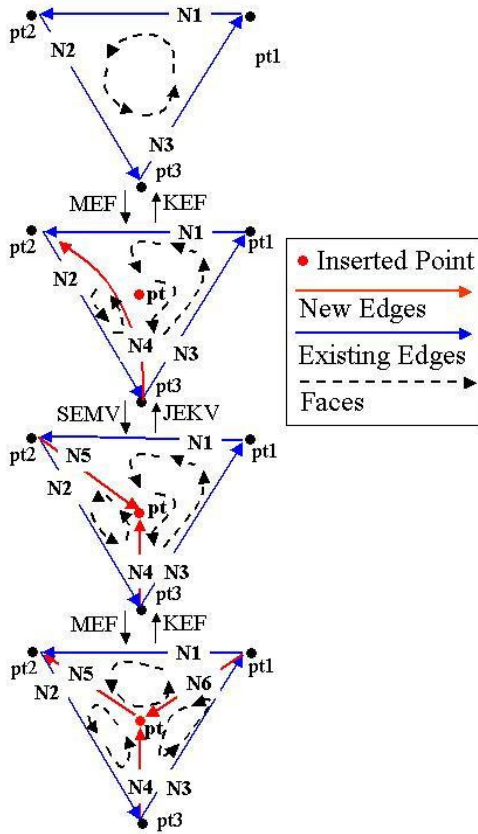


Figure 9 Insert Point Operator

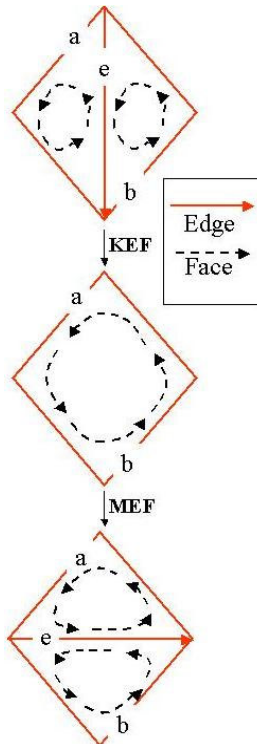


Figure 10 Swap Operator

4. Extending the Terrain Surface

We have re-formulated the traditional incremental algorithm for creating a TIN model. In order to add extra features to our TIN, a topologically connected TIN model is built using Euler

Operators, and then an additional operator is used to modify and extend it. The operator “Make Edge Hole Kill Face” (MEHKF) which has exactly the same program code as “MEF” (Tse & Gold, 2001) is used to create bridges or tunnels. A TIN is built up and two unconnected triangles (Triangles A & B in fig 11) on the TIN surface are selected. Their edges are running in an anti-clockwise order.

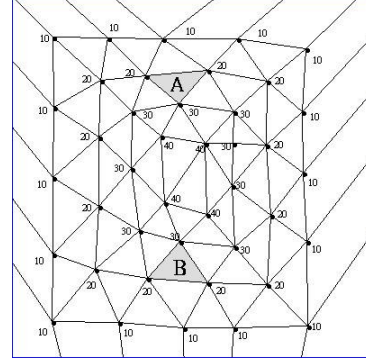


Figure 11 Two unconnected triangles

Fig 12 shows the result of the first step using “MEHKF” to create a hole, a face and an edge between these two unconnected triangles. The two triangles concerned are deleted, an edge is formed between the two triangles, and a new face is formed that loops through each of the deleted triangles and both sides of the new edge (Tse & Gold, 2002). Edge “N1” is the new edge created between two unconnected triangles A and B. The connection of the edges will be $a1 \Rightarrow a2 \Rightarrow a3 \Rightarrow N1 \Rightarrow b3 \Rightarrow b2 \Rightarrow b1 \Rightarrow N1 \Rightarrow a1$ (Face 1). The connectivity of edges in face 1 is in an anti-clockwise order.

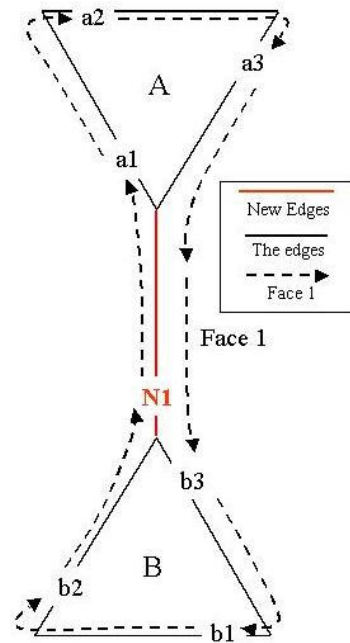


Figure 12 Make Hole Edge Kill Face Operator

The “MEF” operator is used to split the face inside the tunnel shown in fig 13. A new edge “N2” is created to split the face into two pieces and one new face is made, and the connectivity of the edges are $a1 \Rightarrow N2 \Rightarrow b2 \Rightarrow N1 \Rightarrow a1$ (Face 2). Face 1 is split into two pieces using “MEF” again in fig 14. One face “Face 3” and edge “N3” are created. There are three rectangular faces inside the hole: face 1, 2 and 3, therefore three more “MEF” operators are used to split them into triangles. We have

shown in our model that the connectivity of the edges is preserved when you modify the model. You can still “walk through the hole”.

The whole procedure of creating the hole satisfies the Euler Poincaré formula and the topological validity is preserved. The elementary Quad-Edge based Euler Operators are shown to be able to generate and modify the traditional TIN structure, permitting basic CAD operations (Tse & Gold, 2002).

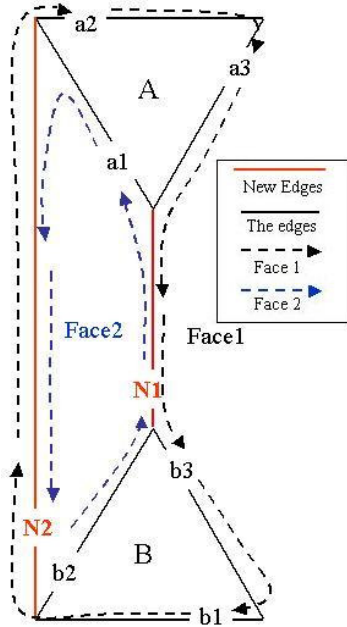


Figure 13 Make Edge Face splits the face

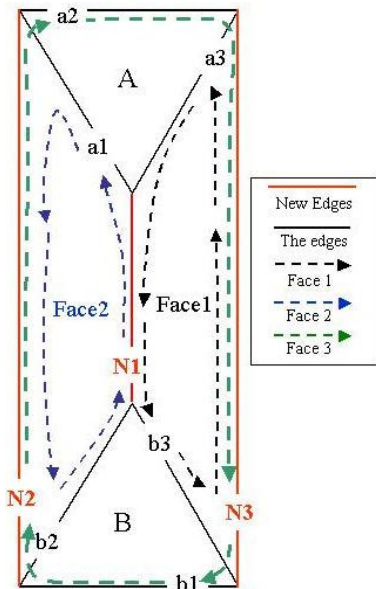


Figure 14 Second MEF splits the face

Besides creating holes and bridges, buildings are created with connected topology on the terrain model. Fig 15 show two simple buildings extruded from the terrain surface. In figs 12 to 14, we describe the procedures for creating a triangular tunnel; the swap operator is used to enlarge the hole in fig 16. Swap could be used to enlarge the bridges. Two bridges are created to connect two complex buildings and two hills in figs 17 and 18 separately.

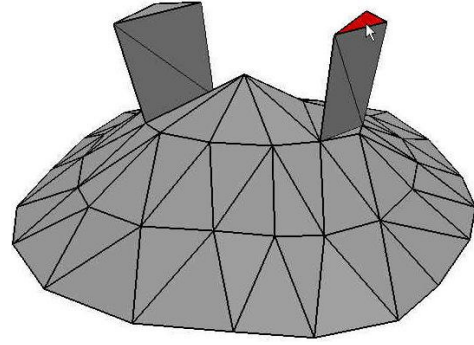


Figure 15 Two buildings on the extended TIN model

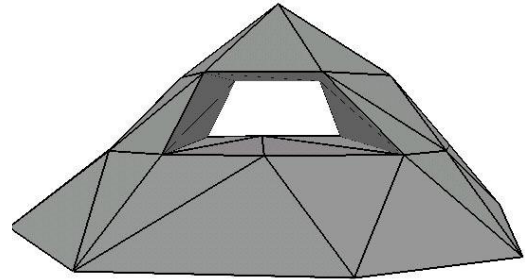


Figure 16 The Extended TIN with a hole

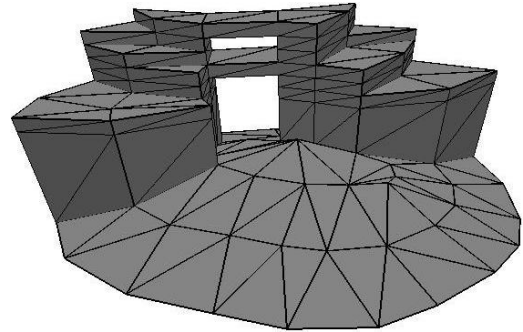


Figure 17 The Extended TIN with bridges and buildings

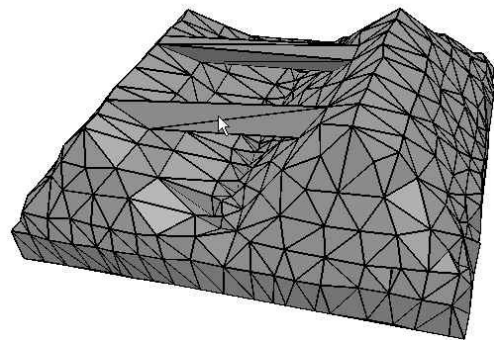


Figure 18 The extended TIN with two bridges

5. Implementation Using Real World Data

Part of Hong Kong is used to illustrate the extended TIN model. A harbour area is chosen (Tsim Sha Tsui) in fig 19, with several buildings, a bridge and a cross-harbour tunnel. Figs 20 and 21 show the result using the map data. The cross-harbour tunnel in fig 22 connects between two sides of the harbour area. The colourless area is formed by, the “underneath” triangles facing towards to the user.

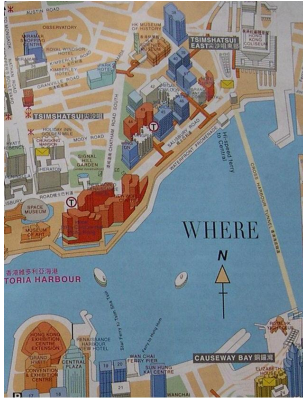


Figure 19 The chosen area in Tsim Sha Tsui

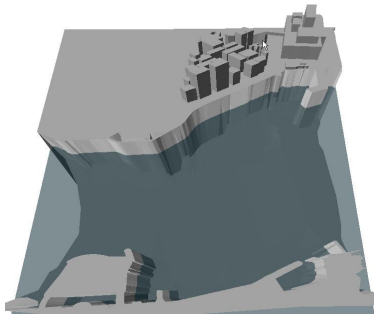


Figure 20 An overview of the extended TIN model

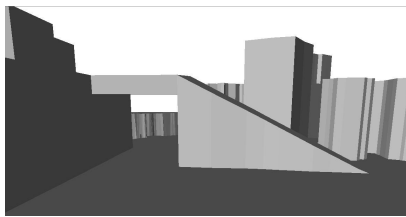


Figure 21 A bridge connects to a building

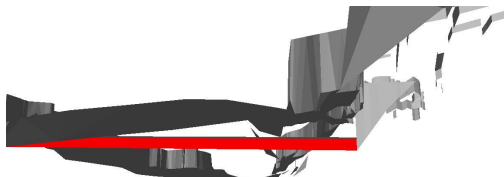


Figure 22 The cross-harbour tunnel

6. Future Developments and Conclusion

The original way of building and extending TIN model without breaking continuity has been shown. With CAD-type Euler Operators, the TIN model is built and extended with buildings, holes and bridges. The extended TIN model is based on the simple Quad-Edge structure, which takes only a few lines of program code to implement. The preserved topological connectivity can serve as a basis for further development, including neighbour analysis, flow modelling and network analysis etc.

This sample using Hong Kong data seems to be a good start. In the future, we hope to include different kinds of data source, for example, laser images (LIDAR) and aerial photos. Existing filtering algorithms may help to categorize the features into different levels of detail.

Reference:

Baumgart, B.G. (1972) Winged Edge Polyhedron Representation (Stanford Artificial Intelligence Report No. CS-320, Stanford University Computer Science Department).

Baumgart, B.G. (1975) A Polyhedron Representation for Computer Vision, American Federation of Information Processing Societies (AFIPS), *The National Computer Conference*, vol. 44, 19-22 May, 1975, AFIPS Press, Anaheim, CA, USA, pp. 589-596.

Braid, I.C., Hillyard, R.C. & Stroud, I.A. (1978) Stepwise construction of polyhedra in geometric modelling, in: K.W. Brodlie (Ed) *Mathematical Methods in Computer Graphics and Design* (Leicester, A Subsidiary of Harcourt Brace Jovanovich).

ESRI (Environmental Systems Research Institute, Redlands, California.) ArcView ® www.esri.com

Gold, C.M. (1998) The Quad-Arc Data Structure, 8th International Symposium on *Spatial Data Handling*, T.K. Poiker & N.R. Chrisman (Eds), Vancouver, BC, Canada, pp. 713-724.

Gold, C.M., Charters, T. & Ramsden, J. (1977) Automated Contour Mapping Using Triangular Element Data, *Computer Graphics*, vol. 11, 170-175.

Guibas, L. & Stolfi, J. (1985) Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams, *ACM Transactions on Graphics*, vol. 4(2), April, pp. 74-123.

Lee, K. (1999) *Principles of CAD/ CAM/ CAE Systems* (Seoul, Addison Wesley Longman, Inc), 582p.

Mantyla, M. (1981) *Methodological Background of the Geometric Workbench* (Finland, Helsinki University of Technology, Laboratory of Information Processing Science), 70p.

Tse, R.O.C. & Gold, C.M. (2001) Terrain, Dinosaurs and Cadastres - Options for Three-Dimension Modelling, *Proceedings: International Workshop on "3D Cadastres"*, C. Lemmen & P. van Oosterom (Eds), Delft, The Netherlands, pp. 243-257.

Tse, R.O.C. & Gold, C.M. (2002) TIN Meets CAD - Extending the TIN Concept in GIS, *Computational Science - ICCS 2002, International Conference, Proceedings of Part III. Lecture Notes in Computer Science*, P.M.A. Slood, C. J. K. Tan, J. Dongarra & A.G. Hoekstra (Eds) vol. 2331, March, Springer, Amsterdam, the Netherlands, pp. 135-143.

van Kreveld, M. (1997) Digital Elevation Models and TIN Algorithms, *Algorithmic Foundations of Geographic Information Systems, Lecture Notes in Computer Science*, M. van Kreveld, J. Nievergelt, T. Roos & P. Widmayer (Eds), Springer-Verlag, Berlin, Germany, pp. 37-78.

Weiler, K. (1985) Edge-Based Data Structures for Solid Modeling in Curved-Surface Environments, *IEEE Computer Graphics & Applications*, vol. 5(1), January, pp. 21-40.