TERRAIN, DINOSAURS AND CADASTRES: OPTIONS FOR THREE-DIMENSIONAL MODELING

REBECCA O.C. TSE AND CHRISTOPHER GOLD

Hong Kong Polytechnic University
Department of Land Surveying and Geo-Informatics
Hong Kong

ABSTRACT

Building a 3D terrain model and cadastre is a necessary development in both GIS and cadastral systems. Existing 2D information could be extended to 3D, while trying to keep the spatial/ topological relationships. Computer-Aided Design (CAD) on the other hand is often fully 3D, which may be excessive for our "extended 2D" mapping needs. We have used the "boundary representation" (b-rep) design from CAD systems, based on manifold models, to solve those problems.

TIN models of the terrain surface are well known in GIS, but are unable to represent cliffs, caves or holes, which are required to represent complex buildings. These simple b-rep structures may be extended using formal Euler Operators to add simple CAD functionality while guaranteeing the connectivity required for a cadastre. The resulting b-rep is a direct extension of current 2D cadastral systems, where ownership is associated with specific surface patches. We believe this provides a simple and reliable extension that is sufficient for many applications.

INTRODUCTION

Definition of cadastre

In the past, a cadastre was stored in 2D, defined by "A Cadastre is normally a parcel-based and up-to-date land information system containing a record of interests in land (i.e. rights, restrictions and responsibilities). It usually includes a geometric description of land parcels linked to other records describing the nature of the interests, and ownership or control of those interests, and often the value of the parcel and its improvements." (FIG, 1991) A new definition of a cadastre was described Kaufmann (1999): "Cadastre 2014 is a methodically arranged public inventory of data concerning all legal land objects in a certain country or district, based on a survey of their boundaries." A more general definition of a land object was introduced, and land can be viewed as a discrete land object with homogeneous conditions inside its outline. This definition matches the current situation, where one land parcel may have one or more different owners. Especially for multi-storey buildings, many people can own parts of the same building and need to have access to some part of its observed exterior "surface". Cadastral systems are described in Williamson (1995) and Henssen (2001).

This raises some difficult questions concerning what a cadastre "is" when extended beyond the two dimensional case. Various progressive models are possible.

1. The cadastre is a set of unconnected polygons in two dimensions, with associated attributes. No attempt is made to specify adjacency.
2. Some form of "topology" is added to Model 1, guaranteeing connectivity and common boundaries and corners.
3. The properties of Model 2 are augmented, by specifying elevation information along the boundaries.
4. The properties of Model 3 are augmented by specifying rights "above" the property (in the air) or "below" (underground).
5. Model 4 is extended by providing a description of the (exterior) surface of the buildings, and assigning rights to portions of that surface in the same way as the previous models.
6. A complete 3D partitioning of the space occupied by the building is given (ignoring the air and ground rights mentioned in Model 4).

Most of these models present some problems. Models 1 to 5 are surface based, and hence the boundaries are observable, but may not catch all ownership situations. Model 6 may not be observable or measurable. Of course, underlying all this is the question whether the cadastre is monumentbased or coordinate-based, as this will affect the viability of several models. Our research is based on Model 5.

Topology vs. database

Topological properties are invariant under continuous distortion. "Neighbourhood" is a topological property because two regions will always be next to each other no matter how a map is distorted. "Enclosure", which relates an interior region to an outer region enclosing it, as well as "Connectivity" which relates a line to a connecting line, are topological properties. These will be the most important topological relationships necessary, and in our case we need to be concerned mostly about the connectivity of a 2D manifold (surface) embedded in 3D space, which separates the inside of the object (the polyhedral earth) from the outside (air or water). This manifold is formed from connected planar elements – usually rectangles or triangles. We will be focusing on triangular elements. The key technologies involved in our work are the traditional TIN model of terrain, and some elements of CAD systems (Zeid, 1991), implemented using data structures developed in Computational Geometry. While individual triangles could be stored in a relational database, as with conventional Shape Files, they would still need modification tools, and basic traversal of triangulations would be extremely inefficient. Other operations, that require the "intersection" of different "topological spaces" (e.g. pipelines with land ownership parcels), would still need to be calculated as required.

However, the whole TIN model of our land and buildings may be too large to hold in memory, and it is undesirable to separate the cadastral database from the surface model. This problem is fundamental to all GIS methods, and we do not claim to have eliminated it. However, our concerns are similar to those of database experts: how to guarantee the integrity of the database – or, in our case, the topological structure. It is not enough to have a valid data model, or topological structure – we also need to be sure that our "transactions" are complete. For this we have turned to the concept of "Euler Operators" as used in b-rep (surface, or manifold) models used in CAD systems. We will not be concerned with CSG models, where independent solids are combined to give the desired result. Nor will we be concerned with "non-manifold" CAD systems (Lee, 1999), although these could possibly provide the same level of "transaction validation" in the future. As a final note, our Euler Operators are implemented using the Quad-Edge (Guibas and Stolfi 1985) structure, which is both simple and elegant, and recent work (T. Merrett, personal communication) suggests that "long transactions" could be designed to handle a variety of queries on the

Quad-Edges structure in a relational database, in a relatively efficient manner.

Boundary representation vs. non-manifold

Manifold models are very similar to the 3D graphics modelling systems used to create dinosaurs, etc., for computer games. In two dimensions these would be equivalent to "shape" file models, where each object is discrete and has a well defined inside and outside. Here the connected topology is simply the sequence of points forming the boundary. In 3D this would be equivalent to the surface mesh – probably a triangulation – that separates inside from outside. There is no explicit relationship between individual objects – in the dual sense, there are only two nodes, with two colours (solid/air). 2D connected topology, as in a choropleth map, makes no such assumption (hence the famous four-colour map theorem). In this case the boundaries are one-dimensional edges. 3D connected non-manifold topology is similar: the boundaries are formed from triangulated surfaces, separating pairs of solids with differing "colours". Thus "inside" and "outside" are only relevant to a single volume element, and a dual tetrahedral network defines the spatial adjacency relationships. We will restrict ourselves to a single volume element – the polyhedral earth, which can be modelled in the same way as the imaginary dinosaurs.

As stated by Mantyla(1981, 1988), "In a boundary representation model (b-rep), an object is represented indirectly by a description of its boundary. The boundary is divided into a finite set of faces, which in turn are represented by their bounding edges and vertices." According to his definition, b-reps are best suited for objects bounded by a compact (i.e. bounded and closed) manifold. Therefore if the main concern is building a non-manifold object, it will be better to use other models.

In 1988 Weiler proposed non-manifold operators to manipulate topological data in non-manifold models. Moreover, they were not based on the basic Euler-Poincare formula, although some of the topological relationships still can be generalized by a new formula by Masuda et al. (1990). However, the formula is more complicated and more operators are needed. It may be valuable to extend the b-rep with Euler Operators into a non-manifold model in the future.

A further consideration with using non-manifold models for cadastral purposes is that, since there is no defined "inside" and "outside", there is no guarantee that any desired feature (e.g. an interior boundary) is "observable". It is not obvious that a non-observable cadastral boundary is a useful concept. The 2D equivalent is Model 4 above, where rights are assigned "above" and "below" the observable surface, so perhaps interior properties should be defined as rights associated with an observable exterior surface. Thus a manifold-based cadastral model may still be valid.

Current research

In our current work we are using the CAD-type b-rep structure and Euler Operators to create a connected TIN model with holes (bridges or tunnels). Starting with the well-known TIN model, we have built a set of CAD-type Euler Operators, including the ability to form holes. Because the Quad-Edges data structure, derived from research in Computational Geometry, greatly simplifies the implementation of these Operators, they might be appropriate for extending the 2D cadastral map. Firstly we show that the Euler Operators may be constructed in a simple and elegant fashion from the basic Quad-Edges operators of "Make-Edge" and "Splice" of Guibas and Stolfi (1985). Secondly we show that all basic TIN modification operations may be performed with Euler Operators. Thirdly, we show that additional Euler Operators may be used to modify the surface form in various ways, including the insertion and deletion of holes that give the basic form of bridges and tunnels.

IMPLEMENTATION OF QUAD-EDGES DATA STRUCTURE IN EULER OPERATORS

We will use the Quad-Edges data structure to implement our Euler Operators because it allows navigation from edge to edge. "Make-Edge" and "Splice" are two simple operations in using Quad-Edges structure. Points and edges are the two main objects inside Quad-Edges which are formed from four connected "Quads". (Thus an edge's "opposite" is found by two calls to "Rot".) You can create an edge with two points and connect two edges together by using "Splice". Every Quad has three properties.

- N – link to next Quad ("Next") anticlockwise around a face or vertex
- R – link to next ¼ Edge ("Rot") anticlockwise around the four Quads
- V – link to vertex (or face)

Algorithm of make edge (Fig. 1)
Make edge for points 1 and 2

- 4 Quads will be created
- 4 Quads will link four parts in anti-clockwise order
- Q1 and Q3 point to themselves in anti-clockwise order
- Q2 and Q4 point each other in anti-clockwise order
- Pt1 and Pt2 are created
- 2 V-links point to them

Algorithm of splice (Fig. 2)
Splice(A, B : Quad) → A B : input Quad-Edges
- Get neighbour edges : Alpha & Beta (Guibas & Stolfi, 1985)
- Reconnect the four pointers

We use the following operators in our current research and implement of Quad-Edges. They are MBT ⇔ KBT, MEF ⇔ KEF and SEMV ⇔ JEKV. We have tested the applicability of other Euler Operators in using Quad-Edges as well, but we will show these three operators only, which are sufficient for creating TIN models. We implement Euler Operators using Quad-Edges. Our set of Euler Operators is based on the simplifying assumption that we are always working with a triangulation (or, more specifically, that no faces have holes).

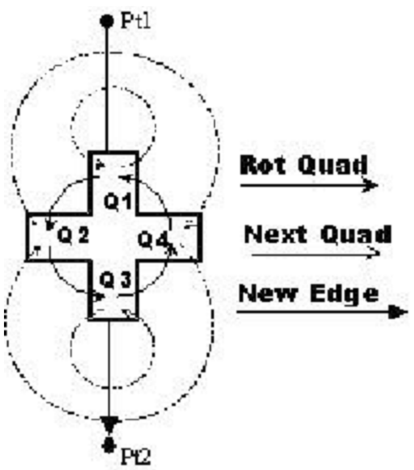

Figure 1: Make edge.

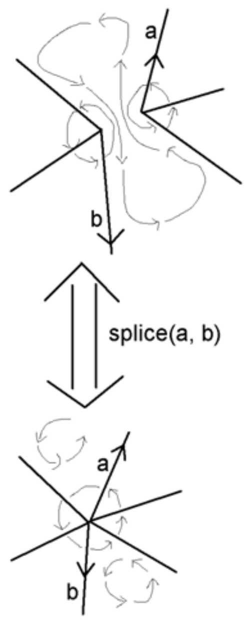

Figure 2: Splice procedure.

MBT ⇔ KBT
In our model we limit ourselves to complete faces with no "dangling edges", so we begin from nothing, and create a big triangle using "Make Body Triangle" (MBT). When we need to kill the big triangle and go back to nothing, we can use "Kill Body Triangle". Fig. 3 shows MBT ⇔ KBT. In MBT, three points are needed as input. If only one big triangle exists, we can use KBT to go back to nothing.

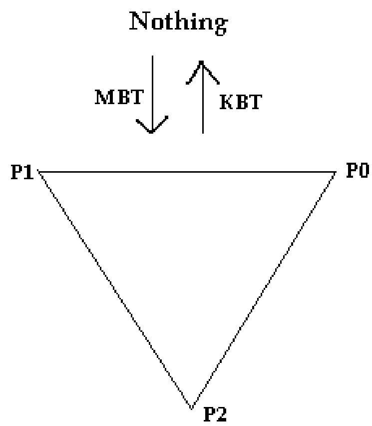

Figure 3: MBT ⇔ KBT.

Algorithm of MBT and KBT
MBT with three point (p0, p1 p2)

- Make a big triangle with three vertices p0, p1 and p2
- Three edges are created, plus two faces and three vertices

KBT

- Kill the big triangle and the edges, faces and vertices

MEF ⇔ KEF
"Make Edge Face" (MEF) and "Kill Edge Face" (KEF). They are used for creating an edge and face and vice versa. In MEF we need to give two quads as parameters to make a new face. In KEF we need to give an edge as a parameter for removing the edge and the related face will be destroyed as this edge is removed. Figs. 4 and 5 show the edges connection in MEF and KEF.

Algorithm of MEF and KEF (Figs. 4 and 5)
MEF with a and b as two input quads:

- Pt3, Pt1 are two vertex points
- e = MakeEdge (Pt3, Pt1)
- Splice a and e edges
- Splice e's Opposite and b edges

KEF with "e" an edge input for removing

- Splice a and e
- *[Disconnect a and e edges]*
- Splice e's Opposite edge and b
- *[Disconnect e's Opposite and b edges]*
- Remove edge "e"

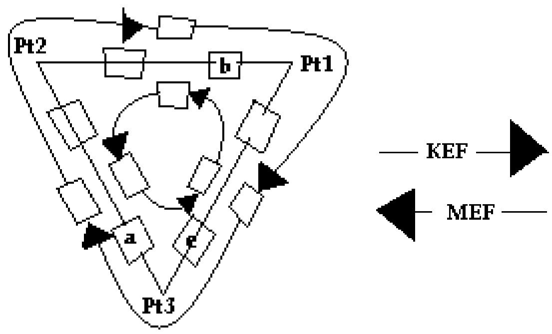

Figure 4: Before KEF.

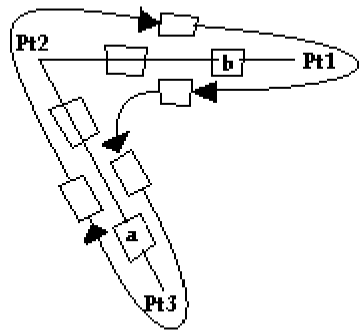

Figure 5: Before MEF.

SEMV ⇔ JEKV

We use "Split Edge Make Vertex" and "Join Edge Kill Vertex" to split one edge into two pieces. This procedure adds or removes a point on a line. It is useful for creating triangles without creating a dangling arc. Fig. 7 shows the result of splitting an edge and making a new vertex (SEMV) and Fig. 6 shows the connection after joining the edges and killing one vertex (JEKV).

Algorithm of SEMV and JEKV

SEMV inputs "e"one edge and "Pt"one point

- P1 and P3 are two vertices of e's edge
- Splice e's Opposite and c *[Disconnect e and c edges]*
- Connect e's Opposite vertex to Pt
- A new edge a will be created by using MEF (e's Opposite, c)

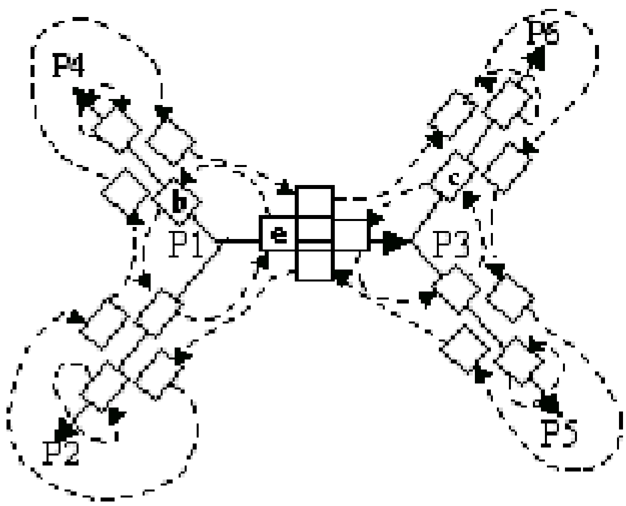

Figure 6: Before SEMV.

JEKV inputs "e" one edge

- Splice e's Opposite and a edges *[Disconnect e's opposite and a edges]*
- Splice a's Opposite and c *[Disconnect a's Opposite and c edges]*
- Connect e's opposite vertex to P3
- Splice e's Opposite and c edges *[connects e's Opposite and c edges]*
- Remove vertex Pt and edge a *[Release one point and one edge]*

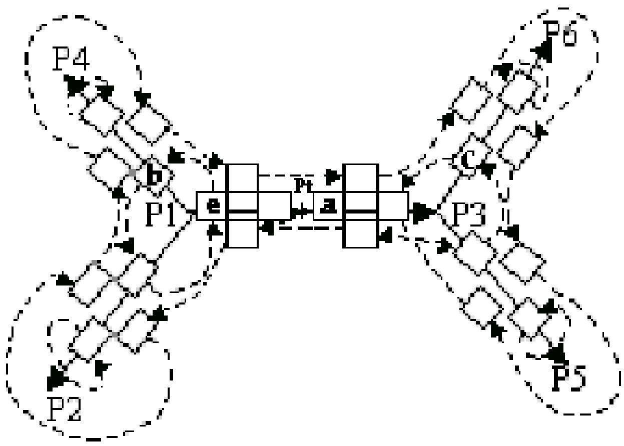

Figure 7: Before JEKV.

IMPLEMENTATION OF EULER OPERATORS IN THE TIN MODEL

In the TIN model we have three main operators which are: creation of a first triangle; insert a point; and swap an edge. We use Euler Operators to create the TIN model. We use operators MBT, KBT, MEF KEF, SEMV, and JEKV to build a TIN model. We will show the operators to use for a TIN model and how to work with holes.

First triangle

We start the TIN model from a big triangle with three point using MBT to create the first triangle. We can reduce the first triangle to nothing by using KBT, but first we need to make sure that only the first triangle is left.

Insert point

Insert point is the other procedure that will use in the TIN model. We need to go through the TIN model and look for the location of the inserted point. In the TIN model the whole surface is formed by triangles, therefore we will insert a new point in an existing triangle. We can use MEF, SEMV and MEF to insert the new point in the existing triangle. Then we could use KEF, JEKV and KEF to delete the point. Figure 8 shows the procedure of inserting a new point.

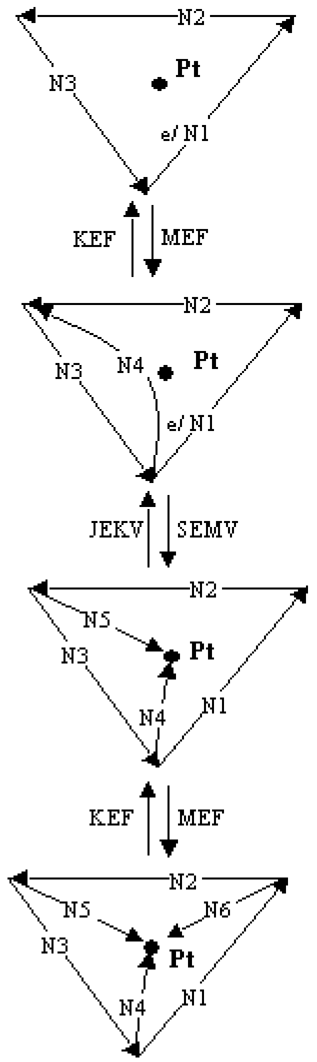

Figure 8: Insert Point.

Algorithm to insert point

Insert Point input a new point "pt"

N1, N2, N3, N4, N5, N6 are edges to be used or created

e is the nearest edge for the point location.

- Use a "Walk" function to get the new point location
- *[It locates the point in which triangle and return the nearest edge in the triangle]*
- N1 is the return edge "e"
- N2 is the Next edge of "N1"
- N3 is the Next edge of "N2"
- N4 is a new edge created by MEF
- N5 is a new edge created by SEMV of N4 edge and the new point "Pt"

- N6 is a new edge created by MEF
  [After insert a new point, two faces will be created, as one face already belongs to the existing triangle]

Swap

Swap is a procedure for swapping two edges inside the TIN model. We use the “in-circle” test to test the triangle if required, and use the swap operator to change edges. Fig. 9 shows the steps of using swap with Euler Operators. We would use KEF and MEF to swap the edge between two triangles. In swap, we need to input the edge to be changed.

Algorithm to swap edge

Swap input: an edge “e” which needs to be changed

- Kill edge e using KEF
- Make a new edge using MEF

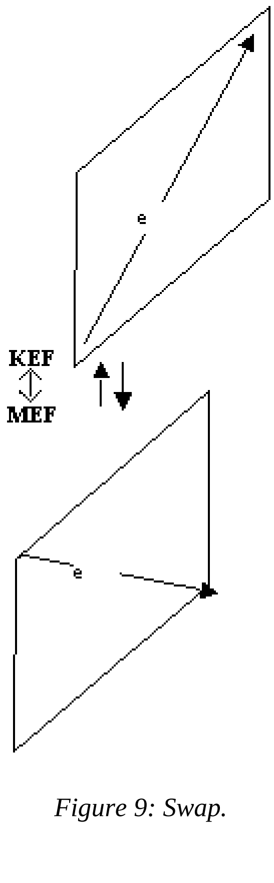

Figure 9: Swap.

ADDITIONAL EULER OPERATORS FOR CREATING HOLES

We can use existing Euler Operators and modify the surface by the insertion and deletion of holes. In our research we make a hole in our TIN model using Euler Operators, as this validates the correctness of the topological structure, and the connectivity from outside to inside. We can use MEHKF and MEF to create a hole in the TIN and keep the connectivity between outside and inside the hole. MEHKF is "Make Edge Hole Kill Face" this is the same procedure as MEF, but one face will be killed and one hole will be created. (MEF operates on two edges of the same face loop; MEHKF operates on separate faces.)

In Fig 10, an existing TIN model, we select two triangles A and B. Their edges are ordered anti-clockwise. We will use them to create a hole and inside the hole the connectivity of edges is running anti-clockwise. This preserves the connectivity of inside and outside edges

In Fig. 11 the two triangles from the TIN model are shown. The order of the three edges inside two triangles are 1 => 2 => 3 => 1 and 4 => 5 => 6=> 4. We will make a new edge between these two triangles and a hole will be created. One face is left inside the hole and one face is killed

Fig. 12 shows the picture after the first step – MEHKF. The connection of the edges will be 1=>P =>5 =>6 =>4=> Q=> 2=> 3=> 1. One face is killed and one hole is created, the edge connectivity is preserved. Fig. 13 shows the result after performing one more edge and face MEF. One new face is made and the connectivity of the edges will be 1 => P =>5 =>R =>1 (New face) and 3 => S => 6 => 4 => Q=> 2=> 3. Figure 14 shows the result of the final MEF. One new edge and face are made. There are three faces inside the hole, but the connectivity is preserved and you can walk from outside to inside the hole.

We have therefore shown that the elementary Quad-Edge based Euler Operators are able to generate and modify the traditional TIN structure, extended as a general b-rep manifold. The assignment of attributes and rights to various portions of this surface provides a visible and easily modifiable 3D cadastral system.

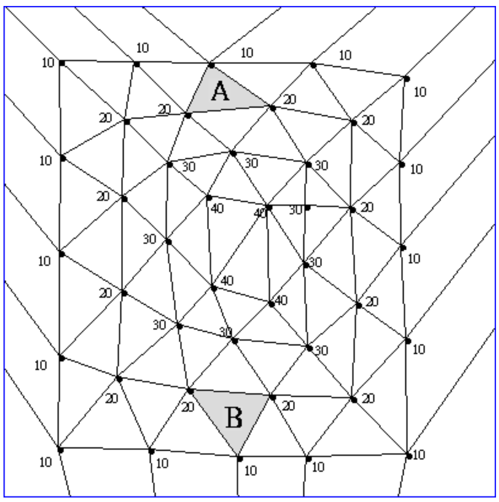

Figure 10: TIN model with two selected triangles.

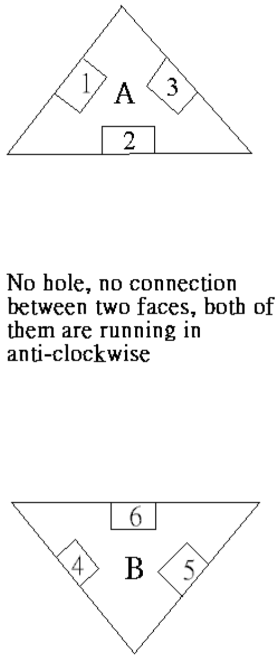

Figure 11: Shows two selected triangles.

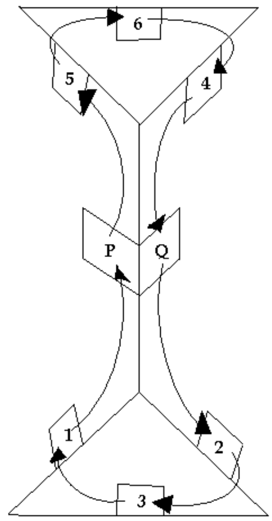

Figure 12: MEHKF.

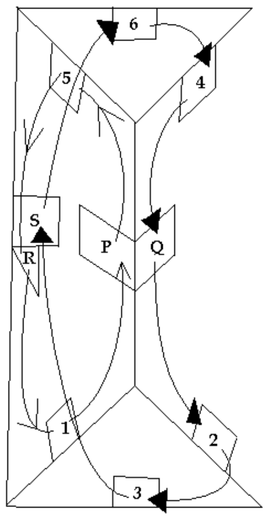

Figure 13: Second step MEF.

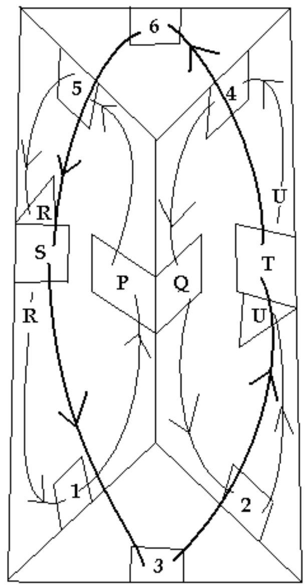

Figure 14: Last step of making hole with MEF.

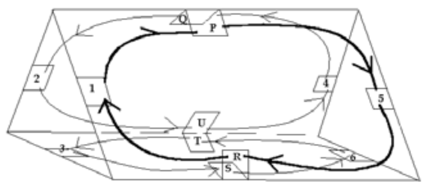

Figure 15: Looking from outside with hole.

CONCLUSIONS

Thus, in summary, we are following a CAD-based b-rep sense of connectedness. The benefits are: a) it is developed from the well-known TIN model; b) the addition of the CAD-type properties of Euler operators, including the guarantee of maintaining manifold connectedness, and the addition of features such as holes and caves; and c) a greatly simplified implementation using Quad-Edges rather than the traditional winged-edge structure. The most difficult aspect of maintaining a cadastre is probably maintaining connectedness between spatial elements. This will be much worse in 3D. We believe that a validated b-rep model is a viable extension of the current 2D cadastral systems.

ACKNOWLEDGEMENTS

The first author would like to acknowledge the financial support of a research studentship from the Hong Kong Research Grants Council.

REFERENCES

FIG, (1991): http://www.fig.net/figtree/thesaurus/chapter-4/cadastre.htm

Guibas, L. and Stolfi, J., (1985):Primitives for the implementation of general subdivisions and the computation of Voronoi diagrams. Transactions on Graphics, v. 4, pp.74-123

Henssen, J., (2001): Basic principles of the main cadastral systems in the world, 2001. http://www.swisstopo.ch/fig-wg71/doc/delft_seminar/paper2.html

Kaufmann, J., (1999): Cadastre 2014: A tool to make land administration more effective? FIG99, Sun City, South Africa.

Lee, K., (1999): Principle of CAD/ CAM/ CAE systems

Mantyla, M., (1981): Methodological background of the geometric workbench

Mantyla, M., (1988): An introduction to solid modelling

Masuda, H., Shimada, K., Numao, M., and Kawabe, S. (1990): A Mathematical Theory and Applications of Non-manifold Geometric Modeling, in Advanced Geometric Modeling for Engineering Applications, North-Holland, Amsterdam

Weiler, K. (1988): Boundary Graph Operators for Nonmanifold Geometric Modeling Topology Representations, in Geometric Modeling for CAD Applications, Elsevier Science, Amsterdam.

Williamson, I.P., (1995): Appropriate cadastral systems, Delft-Seminar, 1995.

Zeid, I., *(1991)*: CAD/ CAM theory and practice, McGraw-Hill, New York, pp368-388

ABOUT THE AUTHORS

Rebecca Tse Oi Chi is currently a MPhil student at the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong, China. Her Supervisor is Prof. Christopher Gold. She obtained her Bachelor of Science in Land Surveying and Geo-Informatics from The Hong Kong Polytechnic University in 1999. Her dissertation developed an algorithm for bus route connectivity using Visual Basic. After her bachelor degree, she worked for one year as a GIS database analyst in a private firm. She started her MPhil research in 2000. Her main research interests lies in developing a topological structure for non-planar networks. It is related to the 3D representation of DTMs with underpasses or bridges for real networks. The Quad-Edge structure will be used to manage the 3D representation. The research will use mainly Delphi for development and OpenGL for visualization

Christopher Gold currently holds the position of Professor in the same department, on leave from Laval University, Quebec. He first worked with TIN models in the 1970s, and has always been interested in spatial data structures. His PhD concerned spatial modelling in Geology, and more recently he has been very interested in Voronoi diagrams as explicit representations of spatial relationships, and have developed a variety of applications.

CONTACT ADDRESS

Name:	Research Student, Rebecca Tse
Name:	Professor Christopher Gold
Institution:	The Hong Kong Polytechnic University
Office Address:	Hung Hom, Kowloon Hong Kong
Telephone:	+852 2766-4304
Telephone:	+852 2766-5955
Fax :	(852)2330-2994
E-mail:	Rebecca.Tse@polyu.edu.hk
E-mail:	christophergold@voronoi.com