BUILDING RECONSTRUCTION USING LIDAR DATA

R. O.C. Tse[*], M. Dakowicz, C.M. Gold, and D.B. Kidner

GIS Research Centre, School of Computing, University of Glamorgan, Pontypridd, CF37 1DL, Wales, UK.
rtse@glam.ac.uk, mdakowic@glam.ac.uk, cmgold@glam.ac.uk, and dbkidner@glam.ac.uk

KEY WORDS: Three-dimensional, Modelling, Euler Operators, Voronoi Diagram

ABSTRACT:

Within the GIS world, LIDAR becomes an important and convenient data source. Many researchers are developing algorithms to extract a bare-earth model and building boundaries from LIDAR data. This paper presents two different methods for building reconstruction using LIDAR data. The first is a traditional method using filtered LIDAR data and combining cadastral building boundaries data (for example, Ordnance Survey Landline data). The second uses the Voronoi Diagram to trace building outlines. To extrude buildings, we use Computer Aided Design (CAD)-type Euler Operators to create a TIN model and then we use the operators to modify the TIN, e.g. extrude buildings, interactive editing or further spatial analysis.

1 INTRODUCTION

LIDAR (Light Detection and Ranging) data is widely used to construct 3D terrain models to provide realistic impressions of the urban environment and models of the buildings. The latest airborne laser scanning technology allows the capture of very dense 3D point clouds from the terrain and surface features. The most common method is to remove all the buildings, trees and terrain objects and generate a bare-earth model. Then building boundaries are extracted from the LIDAR data points. The buildings are reconstructed using CAD software and pasted on top of the bare-earth model.

Section 2 is divided in two parts. The first describes using Ordnance Survey Landline data for building reconstruction. The second part shows an alternative way, the mosaic Voronoi diagram, to trace building outlines from the raw LIDAR data.

Section 3 gives details on building extrusion by using Euler Operators. The use of Euler Operators allows interactive editing of a 3D terrain model (surface) while preserving the topological connectivity (Tse and Gold, 2001).

The last section shows our future plans for building reconstruction using LIDAR data and possibility of using the mosaic Voronoi Diagram for building reconstruction.

2 METHODOLOGIES FOR BUILDING RECONSTRUCTION USING LIDAR DATA

In this section, we show two methods of reconstructing buildings using LIDAR data. These are:

- Landline Tracing
- Voronoi City Modelling

2.1 Landline Tracing

In this section we reconstruct buildings using Ordnance Survey (OS) Landline, original and filtered LIDAR data points. A filtering algorithm is used to remove all objects on the terrain surface, for example, buildings, trees and cars. Morphologic filtering is one of the most common algorithms used to create a bare-earth model, for example slope based filtering (Vosselman, 2003) and modified slope based filtering (Roggero, 2002). We use one of the filtering algorithms to generate a bare-earth model.

Several steps are used to reconstruct buildings using Landline Tracing. They are:

- Create a TIN model with the filtered LIDAR data
- Add Landline data to the terrain surface using a line tracing algorithm
- Use a point-in-polygon algorithm to find out the LIDAR data points for each building and calculate the average heights (z-value) of each building
- Extrude the building using CAD-type Euler Operators. The details will be shown in the next section

There are several steps in the line-tracing algorithm:

1. Insert two points on the terrain surface (Points A and B in figures 1)
2. Check if any triangle edge connects these two points
   - Stop if this is true, or
   - Insert a point half way between these two points if no edge connects them.
3. Repeat step 2 for each half of the line recursively until points A and B are connected by triangle edges.

We add points on the surface and estimate the height of the points by using the surface interpolation method of Dakowicz and Gold (2002).

We use a point-in-polygon algorithm to find out the LIDAR data points which are inside the building boundary and find out the average height of each building.

The steps of the point-in-polygon algorithm are:

[*]Corresponding author.

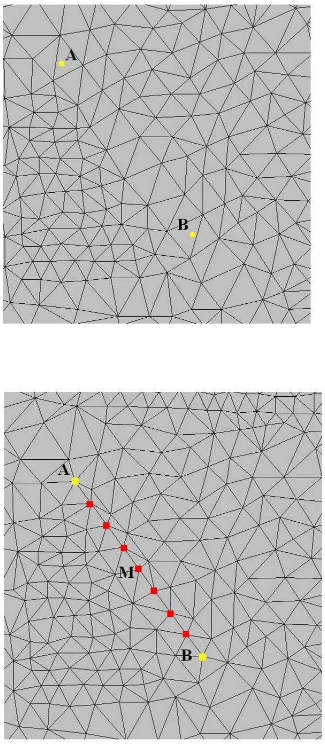

Figure 1: Connects points A and B by adding few (square shape) points in between

1. Delete all the points inside the building boundary
2. Order the vertices of the building boundary in anti-clockwise order
3. Load LIDAR data point from a file
4. Use walk to locate the triangle which contains the LIDAR data point
5. Check if the order of the vertices are anticlockwise
   - Store the point in a list for further calculation if it is true (point A in figure 2), or
   - Do nothing and go to another point if it is not anti-clockwise (point B in figure 3)
6. Calculate the average height in each list for the building height
7. Repeat from step 3 until all the points are searched

Now we have added the building boundary to the terrain surface. Section 3 will describe how to use the Euler Operators to extrude from the ground to its height. Figure 4 shows the building boundaries (top) and the extruded buildings (bottom) of the University of Glamorgan campus.

2.2 Voronoi City Modelling

Though many researchers are interested in automatic filtering and building extraction algorithms, there is still room for improvement. With the use of Delaunay Triangulation and its dual property, the Voronoi Diagram, we suggest an alternative way to solve the problem. We have found that the use of the Voronoi Diagram to create mosaic images is really suited to our use. Therefore we use the z-value from the LIDAR points for comparison. Several steps are used to reconstruct buildings using Voronoi City Modelling. They are:

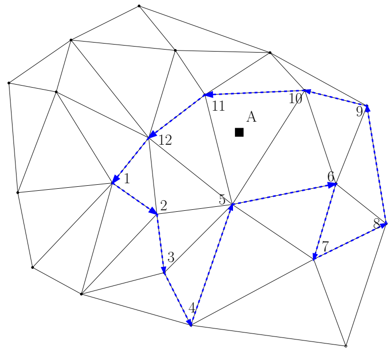

Figure 2: The order of the vertices is in an anticlockwise order 5-10-11

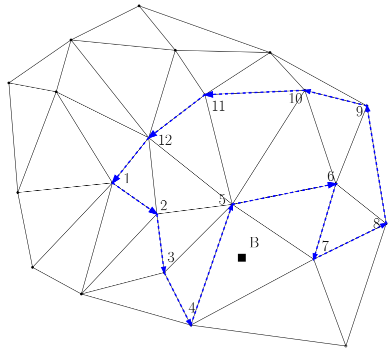

Figure 3: The order of the vertices is in a clockwise order 4-5-7

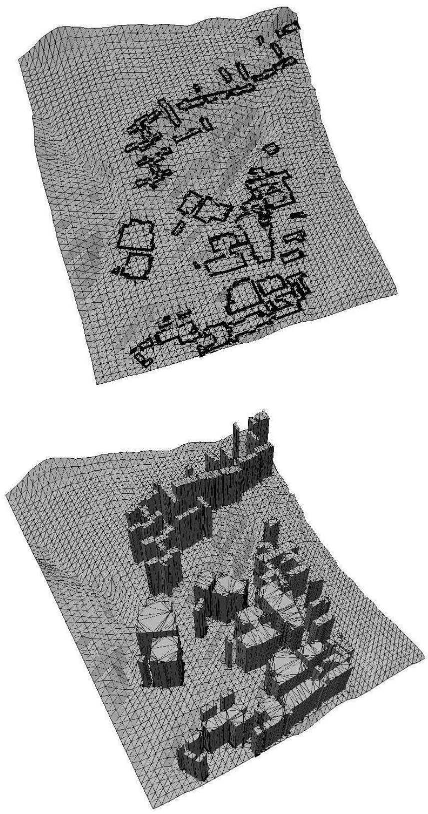

Figure 4: Building boundaries (top) and the extruded buildings (bottom) of the University of Glamorgan campus

1. Sample the raw LIDAR data to give a low density of data points.
2. Create two Voronoi Diagrams, from the original and sampled data points respectively.
3. Assign the z-value of the sample Voronoi centres to the original data points at the same location.
4. Pick a sampled Voronoi cell.
5. Compare the picked sampled Voronoi cell with its neighbour cells.
6. Move the cell and see whether it increases the height differences when compared with its neighbour.
   - Move it and re-assign the z-value of the newly moved sampled Voronoi cells to the original data points at the same location if it does.
   - Do nothing and go to another cell if it does not increase the differences.
7. Repeat steps 4 to 6 until every cell is processed.
8. Display the sampled Voronoi cells in a 3D view.
9. Iterate the whole process from steps 1 to 7 until it shows the best building shapes in the 3D view.
10. Extrude all the Voronoi cells using Euler Operators.
11. Merge the Voronoi Cells with similar heights.
12. Show all the building blocks on the terrain.

Figures 5 and 6 show the Voronoi Cells of an original (top) and after few iterations (bottom) in a 2D view, and figure 6 shows the 3D view where the building outlines have more or less formed.

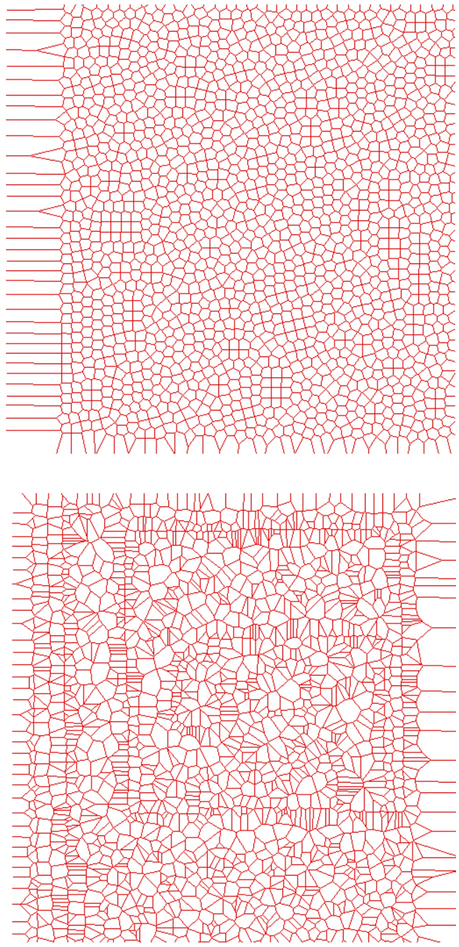

Figure 5: The original (top) and iterated Voronoi Cells in 2D view

3 BUILDING EXTRUSION USING EULER OPERATORS

Constructive Solid Geometry (CSG) and VRML are the two common methods for modelling and rendering buildings on the terrain. Brenner (1999) and Suveg and Vosselman (2004) used CSG to generate a complex building with the Boolean operations of union, intersection and differences. Rottensteiner and Briese (2003) used VRML to display the generated buildings. However topological relationships are not kept during the construction. All the buildings are superimposed on the terrain surface without actually connecting to it. If the topological connectivity is preserved, more kinds of spatial analyses can be performed.

We started by creating a 2.5D TIN model and used Euler Operators to extend the model. Using Euler Operators to extrude a building includes four steps (Tse and Gold, 2001).

1. The Basic Quad-Edge data structure (with only two topological operators) (Guibas and Stolfi, 1985).
2. The implementation of Euler Operators using the Quad-Edge data structure.
3. The implementation of triangulation models using Euler Operators.
4. The building extrusion using additional Euler Operators.

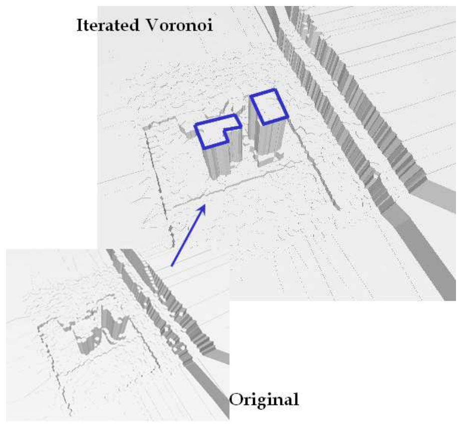

Figure 6: The original and iterated Voronoi Cells in 3D view

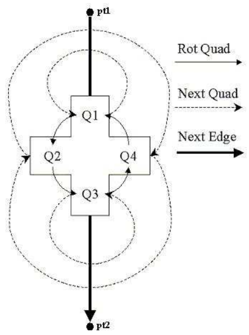

Figure 7: Make Edge creates a new independent edge

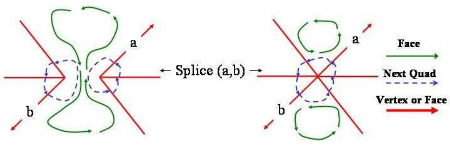

Figure 8: Splice and its own inverse, either splits a face or merges two faces

3.1 The Basic Quad-Edge Data Structure

We used the Quad-Edge data structure as the base of our model. More particularly, the Quad-Edge structure was used to implement a set of Euler Operators that sufficient for the maintenance of surface triangulations. According to Weiler (1988), if a topological representation contains enough information to recreate nine adjacency relationships without error or ambiguity, it can be considered a sufficient adjacency topology representation. These Euler Operators form the basis of the standard (two-dimensional) incremental triangulation algorithm. In addition, Euler Operators can serve to generate holes within our surfaces, thus permitting the modelling of bridges, overpasses etc. that are so conspicuously lacking in the traditional GIS TIN model. The individual Quad-Edge and Euler Operators take only a few lines of code each. "Make-Edge" and "Splices" (figures 7 and 8) are the two simple operations on the Quad-Edge structure, which is formed from four connected "Quads" objects, using the simple implementation of Gold (1998). Every Quad has three points:

- N - link to next Quad ("Next") anticlockwise around a face or a vertex
- R - link to next 1/4 Edge ("Rot") anticlockwise around the four Quads
- V - link to vertex (or face)

3.2 Implementation of Euler Operators Using Quad-Edge Data Structure

Five spanning Euler Operators (plus the Euler-Poincaré formula) suffice to specify the number of elements in any boundary representation model (Braid et al., 1978). In TINs there are no loops (holes in individual faces), so these will not be considered. Thus four spanning Euler Operators suffice for TINs with tunnels or bridges. Figure 9 shows "Make Edge Vertex Vertex Face Shell" (MEVVFS) to create an initial shell, and its inverse "Kill Edge Vertex Vertex Face Shell" (KEVVFS) removes it. "Make Edge Face" (MEF) and its inverse "Kill Edge Face" (KEF) in figure 10 creates and kills an edge and a face. "Split Edge Make Vertex" (SEMV) splits an edge and creates a vertex, and its inverse is "Join Edge Kill Vertex" (JEKV) in figure 11.

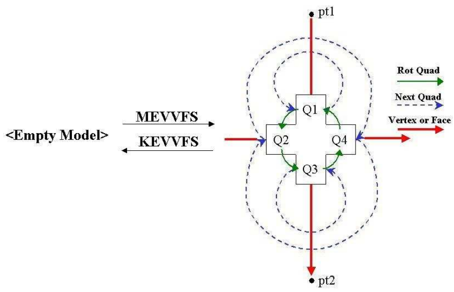

Figure 9: MEVVFS and its inverse KEVVFS

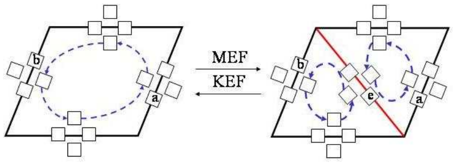

Figure 10: MEF and its inverse KEF

3.3 Implementation of the TIN Model Using Euler Operators

In the TIN model we have three main functions, which are:

- Create a "First Triangle" big enough to contain all the data points and its inverse kill the first triangle (figure 12)
- Insert or its inverse delete a point (figure 13)
- Swap an edge (figure 14)

In creating the First Triangle, three points are needed as input. Three different Euler Operators are used: MEVVFS, MEF and SEMV. 3 points (pt1, pt2, and pt3) are input to create the First Triangle and 3 edges (e1, e2, and e3) are the output. MEVVFS creates the first edge e1. MEF creates a new edge e3. SEMV splits edge e3. To kill the First Triangle, JEKV joins edge e2 and e3. KEF kills edge e3 and a face. KEVVFS kills the last edge e1, giving an empty space.

Figure 13 shows inserting and its inverse, deleting, a new point which makes three edges and two faces. MEF creates an edge N4. SEMV splits an edge N4. In the last step MEF creates a new edge N6. We use KEV and JEKV to delete a point. To delete a point, we use MEF to kill an edge and face. Then we use JEKV to join two edges and kill a vertex. We use KEF to kill edge N4 and its associated face.

Swap is a procedure for swapping two edges inside the TIN model. For the Delaunay Triangulation, we use the "in-circle" test to test the triangle, and use the swap operator to change edges. The Delaunay Triangulation is based on the empty circumcircle criterion (Guibas and Stolfi, 1985). Figure 14 shows the steps for Swap

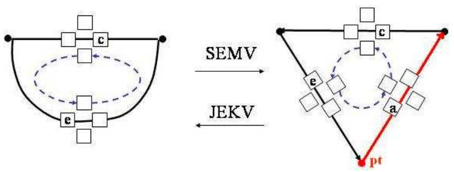

Figure 11: SEMV and its inverse JEKV

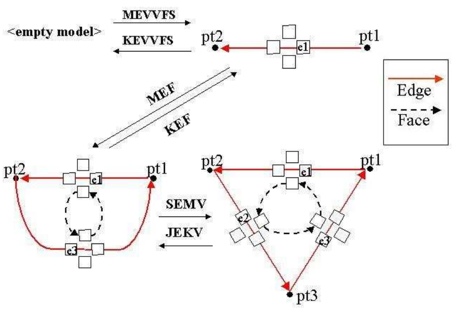

Figure 12: First Triangle should be big enough to contain all data points

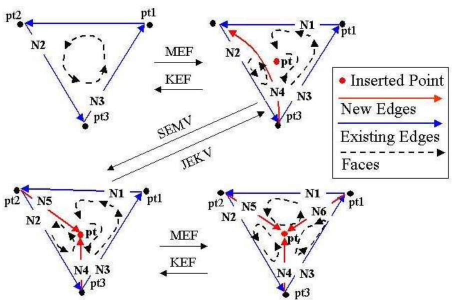

Figure 13: Insert and its inverse delete a point

using Euler Operators. We need to input an edge e to be changed. KEF kills the edge and MEF creates the edge. It swaps the edge between two triangles.

3.4 Building Extrusion Using Euler Operators

We used one more Euler Operator to simplify the procedure. ”Make Zero-Length Edge Vertex” (MZEV) and its inverse ”Kill Zero-Length Edge Vertex” (KZEV) is used to create and kill a zero length edge and a vertex.

We selected a rectangle with two triangles and a common edge to extrude a building. MEF creates a face and edge N1, which runs from point pt1 to pt4 (in figure 16. Three face loops are inside the selected rectangle.

Repeat MEF three more times and three more faces and edges are created in figure 17. They run in the direction opposite to that of the boundary edges.

MZEV can be replaced by using MEF, SEMV and KEF. MZEV can simplify the complicated procedures. In figure 18 MZEV is used to split point pt1 in two pieces. Point pt5 is created and edge N5 is created. Point pt5 is vertically on the top of point pt1. They have the same x and y coordinates, but different height value. The height of point pt5 is equal to the building height.

Three more MZEV operators are used. If the building has more than four corners, more MZEV operators are used, until the entire buildings' corner points are split. MZEV split pt2, pt3, and pt4 by adding pt6, pt7 and pt8 respectively in figure 19.

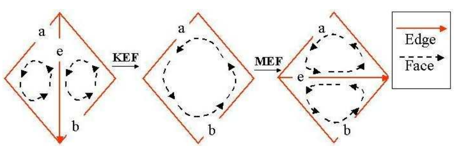

Figure 14: Swap an edge

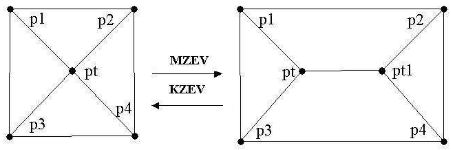

Figure 15: MZEV and its inverse KZEV

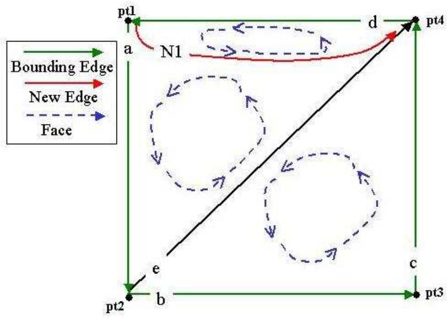

Figure 16: MEF splits the left-hand side triangle face

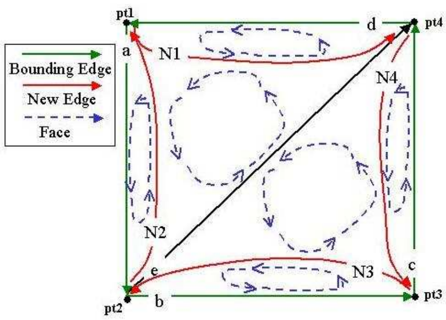

Figure 17: Repeat MEF four times and create four edges and faces

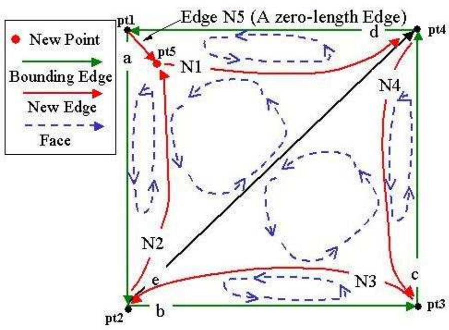

Figure 18: MZEV creates an edge and a vertex (with the building height)

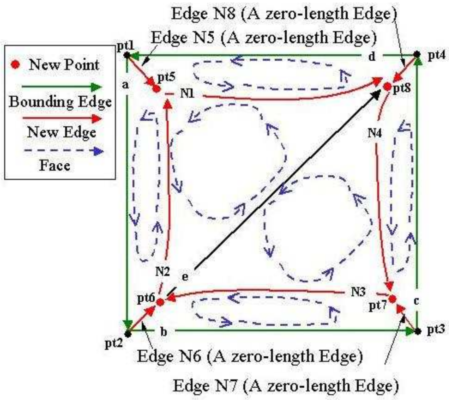

Figure 19: Repeat MZEV four times and create four points to the building height

4 CONCLUSION AND FUTURE WORK

We are exploring the possibility of using laser scanning and Landline data to generate a 3D terrain model automatically as another approach to reconstructing buildings using LIDAR data. It may be a convenient and quick solution. However because of the immature filtering and building extraction algorithm of LIDAR data, it may not be as easy as we think.

Though Voronoi City Modelling at this stage may not be a perfect solution, it can be a new solution for an un-solved problem. However it may need some time until it becomes better. It may involve manual editing to refine the outlines of the buildings and it can be an open question for the future.

When we get the extruded buildings on the terrain, we will try to find out a way to reconstruct the roof. There are different kinds of roof structures in the UK. Some of them can be complicated. Many researchers are still looking for a perfect solution for roof reconstruction. It can be an interesting topic for future development because every building needs a roof.

REFERENCES

Braid, I., Hillyard, R., and Stroud, I., 1978. Stepwise construction of polyhedra in geometric modelling. In K. Brodlie, ed., *Mathematical Methods in Computer Graphics and Design*, pp. 123–141. A Subsidiary of Harcourt Brace Jovanovich, Leicester.

Brenner, C., 1999. Interactive modelling tools for 3D building reconstruction. In D. Fritsch and R. Spiller, eds., *Photogrammetric Week '99'*, pp. 23–34. Wchmann Verlag, Heidelberg.

Dakowicz, M. and Gold, G., 2002. Extracting meaningful slopes from terrain contours. In P. Sloot, C. Tan, J. Dongarra, and A. Hoekstra, eds., *In Proceedings : Computational Science - ICCS 2002, Lecture Notes in Computer Science*, volume 2331, pp. 144–153. Springer-Verlag Berlin Heidelberg, Amsterdam, The Netherlands.

Gold, C. M., 1998. The quad-arc data structure. In T. Poiker and N. Chrisman, eds., *8th International Symposium on Spatial Data Handling*, pp. 713–724. Vancouver, BC, Canada.

Guibas, L. and Stolfi, J., 1985. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4(2):pp. 74–123.

Roggero, M., 2002. Object segmentation with region growing and principal componenet analysis. In *ISPRS commission III Sysmposium "Photogrammetric Computer Vision"*, pp. 298–294. Graz, Austria.

Rottensteiner, F. and Briese, C., 2003. Automatic generation of building models from LIDAR data and the integration of aerial images. In H.-G. Maas, G. Vosselman, and A. Streilein, eds., *Proceedings of the ISPRS working group III/3 workshop '3-D reconstruction from airborne laserscanner and InSAR data'*, volume 34 Session IV. Institute of Photogrammetry and Remote Sensing Dresden University of Technology, Dresden, Germany.

Suveg, I. and Vosselman, G., 2004. Reconstruction of 3D building models from aerial images and maps. *ISPRS Journal of Photogrammetry & Remote Sensing*, 58(3–4):pp. 202–224.

Tse, R. O. C. and Gold, C. M., 2001. Terrain, dinosaurs and cadastres - options for three-dimension modelling. In C. Lemmen and P. van Oosterom, eds., *Proceedings: International Workshop on "3D Cadastres"*, pp. 243–257. Delft, The Netherlands.

Vosselman, G., 2003. 3d reconstruction of roads and trees for city modelling. In H.-G. Maas, G. Vosselman, and A. Streilein, eds., *Proceedings of the ISPRS working group III/3 workshop '3-D reconstruction from airborne laserscanner and InSAR data'*, volume 34, Part 3/W13. Institute of Photogrammetry and Remote Sensing Dresden University of Technology, Dresden, Germany.

Weiler, K. J., 1988. Boundary graph operators for non-manifold geometric modeling topology representations. *Geometric Modeling for CAD Applications*.

5 ACKNOWLEDGEMENT

The research discussed in this paper was supported by the Ordnance Survey, Southampton (Research and Innovation).