Automatically Generating Large Urban Environments based on the Footprint Data of Buildings

R. G. Laycock
School of Information Systems
University of East Anglia
Norwich, NR4 7TJ
robert.laycock@uea.ac.uk

A. M. Day
School of Information Systems
University of East Anglia
Norwich, NR4 7TJ
amd@sys.uea.ac.uk

ABSTRACT

This paper focuses on the generation of three dimensional models of large urban/suburban environments. Previous work on the reconstruction of particular environments is based on multiple overlapping aerial or street level images. Unfortunately these approaches do not extend well to large environments. The main reasons for this are that they require expensive high-resolution aerial images and a labour intensive modelling or data capture procedure. Consequently methods have been developed to generate large urban environments based on environmental data such as elevation data or building footprints. This permits the model to be based on actual data for the area being modelled and at a cost far less than that of aerial images. By reducing the data given to the model generation procedure various parameters are undetermined. These include roof style and textured appearance. This paper focuses on the use of building footprint information to construct a three dimensional model. It uses LIDAR data to give the buildings a height value and assigns them a roof using new techniques for roof modelling.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modelling

General Terms

Algorithms

Keywords

Roof Modelling, Straight Skeleton, Virtual Environments

1. INTRODUCTION

The ability to model large urban environments in a computer has many important applications in industrial, recreational and educational fields. A variety of projects have been undertaken around the globe to develop applications to model dense urban environments. These range from those that attempt to model a particular town or city, [11], to those that create purely artificial environments, [1]. Many approaches to the problem of modelling a particular urban environment are infeasible when the area to be modelled increases in size. There are two possible reasons for this; the cost for data capture and the labour intensive modelling procedure involved. For instance modelling a large area using aerial image to model techniques would incur a substantial cost.

This paper will discuss an approach that uses building footprint information to generate three-dimensional environments. The first stage involves the generation of a set of building footprints from a collection of building line segments. The second stage involves the extrusion of the footprints to visualise the building's walls. The final stage illustrates the use of assigning different roof styles to the buildings in the scene. Thus enabling a more varied skyline to be obtained than is possible by simply capping buildings with flat roofs.

SM'03, June 16–20, 2003, Seattle, Washington, USA.

2. PREVIOUS WORK

One approach to generate a large environment is to model some of the landmark buildings and surround them with randomly positioned vernacular buildings. To improve on this Flack et al, [8], use the road network obtained by digitising the road layout from a raster map. The layout allows their buildings to be positioned in an appropriate location. Recently two new methods for generating urban environments have been developed. These are briefly summarised below. In 2001 Parish documented an approach to generate large urban environments using L-Systems [1]. The approach begins by using environmental data to construct a possible road network. The remaining space is then partitioned into building plots and various combinations of buildings are fitted into these "allotments". This approach has the advantage of requiring relatively cheap data but at the expense of reduced accuracy in the modelling of an actual physical location.

In 2002 Yap et al documented an approach to model Manhattan [2]. The approach uses a street map of Manhattan, which contains polygons representing city blocks, as a base for the model. As a preliminary step to their approach they create street sections and street junctions. Their method proceeds by first splitting the map into a set of neighbourhoods. Each neighbourhood contains some information such as zoning classification, age of buildings, dominant architectural style etc. These parameters are propagated through their system enabling building geometry to be created and textures to be retrieved from a database. This approach uses data, which contains more information concerning the location to be modelled. Therefore the models can be generated to reflect the real world environment more accurately than those documented in [1,8]. However in order for the model to represent the real location statistics have to be obtained for the area. In this paper height information, architectural style and other statistics are recorded from the area manually. This increases the data capture time. An additional increase in modelling time is incurred with their approach, as any roofs, other than flat roofs, are modelled manually.

An approach to construct buildings including generating roof models based on the building footprints in the United Kingdom is presented in this paper. Thus providing an environment automatically, with an appearance close to the physical world, to be rendered in real time.

3. BUILDING FOOTPRINT GENERATION

In this section it will be explained how a collection of line segments used for visualising building footprints can be transformed into a data structure encompassing the building footprints topology. The motivation for undertaking this transformation is to enable the building footprints to be extruded. Furthermore once a simple closed polygon has been achieved roof procedures can be applied to cap the polygons. By storing the connectivity of the line segments back face culling can be applied to increase the frame rate when rendering the scene. In the United Kingdom Ordnance Survey produce the LandLine.Plus product [6]. This product available in the National Transfer Format, NTF or Drawing Interchange File Format, DXF covers the UK on a tile by tile basis. LandLine.Plus data contains a collection of line segments, points and text representing many features such as building outlines, road centrelines and roof indicators. The main use of this product is for visualisation. Consequently there is no topological information concerning the data. The requirement is a method which will take the collection of line segments and roof indicators and construct a set of building footprints. Each building footprint will be represented by a set of integers indexing a set of vertices in an anticlockwise order. In this way no redundant vertex information has to be stored.

The roof indicators in the Landline.Plus product are points which indicate the presence of an object with a roof. It is suggested in the product's file format that the roof indicators can be assumed to be surrounded by building features. Consequently a building footprint can be found by taking a roof indicator and finding the surrounding line segments. This is achieved by first constructing a horizontal ray from the roof indicator. The first line segment, L1, that the horizontal ray intersects is on the surrounding building footprint. The remaining line segments can be found by performing a boundary walk. Unfortunately the line segments are stored with the only clue of connectivity being that they have collocated end points. Consequently to perform a boundary walk the algorithm first has to determine a set of candidate line segments which are connected to L1. From this connected set of candidates it chooses the line, L2, that makes the least interior angle. This procedure repeats until the walk returns to the beginning line segment. This procedure is time consuming in two stages. The first occurs when determining the nearest line segment, L1, from the roof indicator along a horizontal ray and the second is where the algorithm constructs the set of connected line segments to L1. To improve on these problems a spatial partitioning algorithm is required to remove some of the line segments from the list of potential candidates. The approach used for achieving this spatial partition is to group all the connected line segments together. This aggregation of line segments into connected sets can be performed using the Merge Find Set Data Structure [3]. For each connected set the exterior boundary can be obtained by performing a boundary walk always choosing the next line which makes the largest exterior angle. The exterior boundary provides a polygon in which a point in polygon test can be undertaken. This allows roof indicators to be used with a connected set of line segments only if it is within the corresponding exterior boundary.

To avoid computing a point in polygon test for all roof indicators and connected sets an axis aligned bounding box test is performed. The algorithm proceeds as follows. It iterates through all the sets of connected line segments. For each connected set of line segments it tests each roof indicator. First the roof indicator is tested to determine if it is inside the bounding box of the connected set of line segments. This bounding box is computed when the exterior boundary of the connected set of line segments is computed. It is defined by two extremal x coordinates and two extremal y coordinates of the exterior boundary. If the roof indicator passes the bounding box test it is then used in the point in polygon test [5]. If it passes this test the roof indicator is checked to determine that it is not inside any existing building footprints. This stage is required to ensure that duplicate building footprints are not constructed, due to the possibility of many roof indicators being located in one building footprint. If the roof indicator passes these tests it is then used to initialise a boundary walk. By performing these tests the roof indicator will only need to determine the first line segment the horizontal ray intersects with from the set of connected line segments. Once the roof indicator has been used to select a building footprint from the connected set of line segments it is removed from the list of roof indicators. Thereby ensuring that the roof indicator is not used to partition any other connected set of line segments. The following outlines the Building Footprint Generation procedure.

Procedure: Building Footprint Generation

Input: LandLine.Plus data in NTF format.

Step 1: Read in the data and collect all building footprint line segments and roof indicator points.

Step 2: Use the Merge Find Set data structure to group connected line segments

Step 3: For each group of connected line segments determine the exterior boundary.

Step 4: Perform validation to introduce further vertices at junction points.

Step 5: Partition the connected set of line segments into building footprints.

Step 6: Process the connected set of line segments to obtain a list of vertices and a list of integers indexing the vertex list.

Step 4 requires additional processing of the data to introduce vertices at junctions which are not stored in the LandLine data. Some intersections between the end points of line segments and the interior of other line segments are not stored. In order for step 5 to be able to partition the connected set of line segments into building footprints these junctions are required. Extra vertices are added by checking the intersection of each line in the connected set with every other line in the connected set. If a line is found with an end point, EP, on another line segment, L3, in the connected set then L3 is partitioned into two line segments at the coordinate of the end point, EP.

Once the building footprints have been constructed and stored they are ready for processing. This will involve extruding the buildings to construct the walls and capping the polygons with various roof styles.

4. WALL GENERATION

The building footprints can be extruded to an approximately accurate height using Light Detection and Ranging, LIDAR, data. The LIDAR data has been obtained from the Environment Agency, [12], in ArcView Ascii Grid Format. The Environment Agency has implemented a filter to perform the removal of surface features from the LIDAR data to achieve the terrain height map. Consequently by subtracting the filtered data from the unfiltered data the surface feature's height can be obtained.

The building footprints generated in the previous stage are held in anticlockwise order enabling point in polygon tests. This will allow the assignment of points in the two dimensional grid of height data to building footprints. Thus permitting the building footprints to be raised to the height as indicated by the LIDAR data automatically. To reduce the number of point in polygon tests points in the two dimensional grid need only be used if they fall in the bounding box of the building footprint. This means that a loop will be constructed to iterate only over those points in the axis aligned bounding box of the building footprint. If more than one point exists in the footprint the maximum height value will be taken.

5. ROOF MODELLING

Once a set of building footprints has been obtained and extruded to a particular height, each set of walls then requires to be capped with a roof. There are many ways of achieving this. For instance Baillard et al in [13] document the use of multiple overlapping aerial images to obtain a three dimensional point cloud. Planes are subsequently fitted to the point cloud to obtain a polyhedral model of the roof. This approach has the advantage of being capable of modelling varied roof structures but with the disadvantage of required multiple overlapping images. Alternatively a high resolution digital elevation model could be used as documented by Vosselman in [14]. This approach fits roof planes to the three dimensional point cloud. Both of these approaches have the drawback of requiring expensive data sources. To overcome this a single orthorectified aerial image could be used.

Brenner, [7], assigns complete roof models to building footprints extracted from a single aerial image. In this paper the straight skeleton is adopted to obtain a roof model for any given simple polygon. The straight skeleton is unique for any given polygon, consequently to increase the number of roof possibilities for each footprint the generalisation of the straight skeleton was implemented. The technique involved is to first extend the walls vertically up and fold them along the walls to give roof planes. These are used in intersection tests to obtain a set of possible roof vertices. These vertices are then connected in such a way as to create a valid roof. Although this approach does provide varied roofs, rather than just the unique straight skeleton, the style of the roof is always the hip roof style. An urban environment contains many more roof styles than just the hip roof style. Brenner discusses the ability to alter the roof models appearance by changing the gradient of some of the roof planes. The remaining sub sections will discuss methods for obtaining roof models of various roof styles. The roof models generated can then be assigned directly to the walls of the building footprints for visualisation purposes.

5.1 Straight Skeleton - Hip Roof Modelling

Before outlining the modifications made to the straight skeleton the straight skeleton itself will be defined. The straight skeleton can be formed by shrinking a polygon boundary and reducing the edge lengths. The vertices of the polygon boundary will move along the bisectors coming from each vertex. As the polygon shrinks in size two events can occur. These are called the edge and split events and are defined by Aicholzer [9] as follows. An edge event occurs "when an edge shrinks to zero, making its neighbouring edges adjacent". The split event occurs when "a reflex vertex runs to this edge and splits it, thus splitting the whole polygon".

The left of figure 1 illustrates the straight skeleton being applied to a simple concave polygon. It illustrates the two types of event which may occur. To avoid the iterative shrinking polygon procedure the implementation as documented in [4] has been used. This allows the straight skeleton to be computed in O(nm + n log n) time, where n denotes the total number of polygon vertices and m the number of reflex vertices.

The straight skeleton partitions the polygon into roof planes which can be raised to form the roof model [9]. The result is shown in the right half of figure 1. In figure 1 edge events produce vertices in white squares and a split event produced the vertex in the white circle.The roof planes are labelled from one to six. Roof plane two has its supporting edge labelled E.

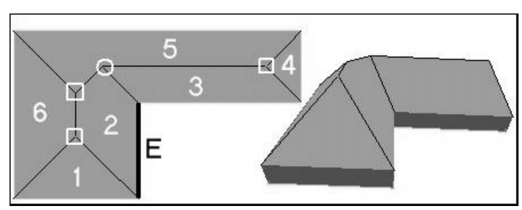

Figure 1: Left: A simple polygon and its straight skeleton. Right: The Hip Roof style generated from the straight skeleton, [9]

The hip roof style forms the basis for the other roof styles such as Gable, Mansard and Dutch Hip. Modifications to the straight skeleton permit these different architectural styles to be visualised for any given simple polygon. Figure 2 illustrates the results of performing these modifications.

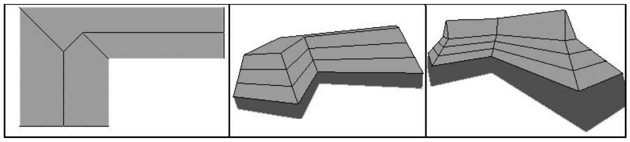

Figure 2: Left: Top View for the Gable Roof, Centre: Mansard Roof, Right: Dutch Hip Roof.

5.1.1 The Gable Roof Style

The gable roof type differs from the hip type by having some roof planes, which have a slope of 90 degrees to the horizontal rather than 45 degrees. The left diagram in figure 2 illustrates the alteration from the standard straight skeleton shown in figure 1. To achieve the above result the following adjustment to the output of the straight skeleton was undertaken.

Step1: Identify a vertex that was created by the intersection of two bisectors. These bisectors emanate from corners of the original polygon.

Step2: Move the vertex from its original intersection position to the midpoint of the line which is incident to both the bisectors that created the intersection point.

5.1.2 The Mansard Roof Style

The illustration in the centre of figure 2 displays the mansard roof style. The first step in creating this roof style for any simple polygon is to determine the point where the nearest split or edge event occurs. Once this distance is determined a new instance of the polygon can be created and shrunk to 80% of this distance to prevent any of the two previously mentioned events occurring. By joining lines between the corners of the old polygon and the newly shrunk polygon the slope of the mansard roof can be created. The top can then be capped with a flat polygon by performing a triangulation of the hole.

This approach is adequate for polygons where an event does not occur too near to the original polygon boundary. If an event does occur near the polygon boundary then the mansard roof style will have a sharp gradient. The section on combining roof styles describes an alternative procedure to be undertaken should this scenario occur.

5.1.3 The Gambrel Roof Style

The gambrel roof can be constructed using a modified version of the mansard roof type procedure. The modifications include altering the proportions between height of roof and distance to shrink the polygon. The mansard roof in figure 2 shows lines on its roof planes that can be altered in height and separation distance. This gives the functionality to vary the roof style.The introduction of further lines would increase the resolution of the mesh enabling curved roof planes to be modelled, perhaps based on Bezier curves.

5.1.4 The Dutch Hip Roof Style

The dutch hip roof can be created by performing one level of the mansard roof type followed by the gable roof type. It is shown in on the right of figure 2. By adjusting the height of the apex of the gable roof, to give the gable's roof planes' a negative gradient, a well roof can be created.

5.1.5 Combining Roof Styles

Up until this point methods have been described to modify the straight skeleton to allow different roof styles to be generated. However, many buildings combine these styles. The occurrence of a split or edge event in the straight skeleton permits the mansard roof style to be used in conjunction with the hip roof. The following summarises the process to be undertaken. Figure 3 illustrates the process.

Procedure: Roof Style Combination

1) Compute the straight skeleton

2) Form a set of roof planes for the polygon.

3) For each roof plane extend and move the supporting edge, E, towards the interior of the polygon.

4) Test for intersection between the moved supporting edge and its corresponding roof plane.

5) If E intersects with the edges of the roof plane then partition the roof plane into two pieces.

6) Raise the vertices of the roof planes to form the roof.

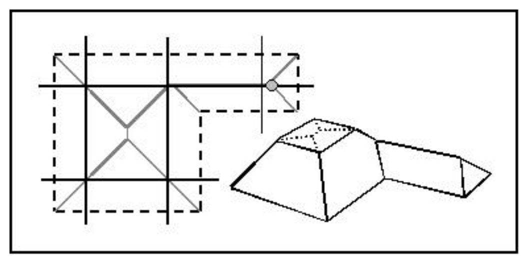

Figure 3: Left: The grey lines indicate the polygon's straight skeleton. Each of the dashed lines indicate a supporting edge, E, to a roof plane. These dashed lines are extended and moved towards the interior of the polygon, represented by solid black lines in the figure. Right: The three dimensional view of the plan shown on the left.

The above process yields exactly the same results as the mansard roof style if the distance the polygon edges are moved is less than the nearest split or edge event. In the case that the polygon edges are moved further than the distance to the nearest split or edge event then it will mean that some of the polygon edges will move outside of their roof planes. In figure 3 the smallest polygon edge, the thin line in figure 3 has moved beyond the vertex marked with a grey circle. Consequently this roof plane will be unaltered. If the line does intersect the roof plane then the roof plane will be partitioned into two by the newly moved polygon edge. This situation occurs in the roof plane with the minimum y coordinate in figure 3. In step 3 of the procedure a distance has to be determined which will be used to shrink the polygon boundary. An appropriate choice is just beyond the distance to the nearest split event. This enables one part of the polygon to have one type of roof style and another part of the polygon to have a different roof style.

Various combinations of the other roof styles can be achieved using this procedure. This is because the mansard and hip roof styles can be altered to reflect the properties of the other roof styles. These alterations were discussed in the previous sections.

5.2 Roof Modelling for Rectilinear Polygons

Using the straight skeleton as the basis for roof modelling has its advantages in terms of the speed of computation and the ability to generate roof types for any simple polygon. However, the roof model created using this technique is not always the most desirable. The left of figure 6 illustrates the problem that can occur and the right illustrates a possible solution. In some instances a single footprint should be partitioned to allow it to have several roof models assigned. Consider the following scenario. Given a polygon footprint a roof is generated by roof draftsmen drawing the plan of the roof over the footprint. This is akin to how the straight skeleton works by constructing a roof model based on the entire polygon footprint. However many vernacular buildings start their life as simple footprints with few vertices and become more complicated through the addition of extensions. In this situation more vertices are added to the footprint and the roof for the extension is added to the existing roof. For this reason a roof modelling procedure should generate a roof with this idea in mind.

The LandLine.Plus data was observed to see the general appearance of the footprints present in the data. It turns out that around 45% of the buildings in a selected village have vertices with interior angles of 90 degrees plus or minus 5 degrees. Therefore for these particular footprints an approach can be used that generates roofs based on the fact that they are close to being rectilinear.

The first step of the procedure is to translate the vertices into new locations to ensure the polygon is rectilinear. To achieve this the algorithm proceeds around the boundary taking three consecutive vertices at a time. If the polygon is to be rectilinear the middle point of the consecutive list of vertices should have an interior angle of 90 degrees. To achieve this the middle point is moved to the nearest location that allows a right angle to be constructed with the other two points in the list. Once a rectilinear polygon has been obtained step 2 is to partition it into rectangles. This is achieved by first extending horizontal and vertical lines from all reflex vertices. These lines are subsequently clipped against the polygon boundary ensuring all lines are interior to the polygon. Performing intersection calculations between each line and the rest of the lines in the set creates extra vertices. The vertices are sorted first by y then by x coordinate. This enables the vertices to be grouped to construct rectangles that form a partition of the original polygon. These rectangles are shown in figure 4 as solid black outlines and are numbered from one to three.

The aim of the next stage is to group these rectangles to form a set of roof models for the given polygon, as illustrated in the right half of figure 4. To aid in deciding which rectangles belong to which roof model the straight skeleton can be computed. This will create a series of horizontal and vertical lines, shown as dashed grey lines connected to solid grey lines in figure 4. In the roof model obtained from the straight skeleton these lines would form the apexes of the roof. The shortest distance between the line and its supporting edge determines the height of each apex. By taking each line and growing an axis aligned rectangle an overlapping rectangle cover of the polygon is obtained. The horizontal and vertical lines in figure 4 have been grown using the following procedure to obtain two rectangles comprising of dashed grey lines labelled R1 and R2.

Procedure: Rectangle Growing Procedure

1) For each horizontal or vertical line, L, with height, h,

2 a) Form two parallel lines to L, II, which are of distance h away.

2 b) If L is vertical (horizontal) find all polygon edges which are horizontal (vertical) and either intersect the II or are inside the region defined by II but do not intersect line L.

2 c) Partition the set of lines into two sets A and B where the lines in A have the line, L on their left and the lines in B have the line L on their right.

2 d) If L is vertical (horizontal) take the maximum y (x) from set A and minimum y (x) from set B.

By completing the above procedure each line L is enlarged to an axis-aligned rectangle, R. The next stage involves grouping the rectangles computed in step 2 into a set. The rectangle labelled R1 in figure 4 has rectangles one and two in its set. The rectangle labelled R2 has rectangles two and three in its set. This assignment is achieved by testing each rectangle to determine if its centroid is within R1, R2 or both. To ensure one rectangle is uniquely assigned to each set of rectangles the set difference operation is applied. It is applied observing the height of the line L used to create the rectangle R. The idea being to relinquish rectangles to a higher priority set of rectangles. The priority values are equal to the height of the line L. Thereby allowing more dominant roofs to grow. This enables each R to have a unique set of rectangles. In figure 4 the set difference operation will remove rectangle two from R2, as the vertical line of the straight skeleton used to create R1 is further from the supporting edge than the horizontal line used to create R2.

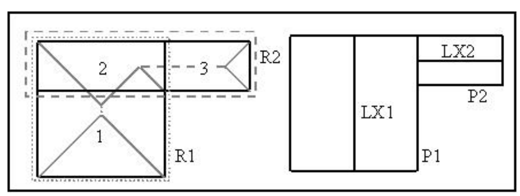

Figure 4: Left: Illustrates the Rectangle Growing Procedure. Right: P1 and P2 illustrate the two roof models which are generated. Raising their centre lines, LX1 and LX2, would form two gable roof models.

For each R a polygon P is created which is formed by the exterior boundary of the set of rectangles. In figure 4 the polygon P1 corresponding to R1 consists of the union of rectangles one and two. The polygon P1 is then partitioned into two by extending the line L, labelled LX1 in figure 4, to create the roof model. This process is undertaken for all R. At this point the roof could be rendered and the result would be a set of roof models for the rectilinear polygon.

However the result is a set of gable roofs each roof per horizontal or vertical line in the straight skeleton (unless the set difference causes the removal of one of the sets of rectangles). In figure 4 two gable roof models have been created. The first has a base equal to the union of rectangles one and two and the second has a base equal to rectangle three. The apex is taken from the vertical and horizontal lines and is raised to a height equal to the distance the line is from the nearest supporting edge. To obtain a realistic roof each of these gable roof models should be merged together. This is achieved by merging a roof with all the other roof models, which have higher priority. Altering the edges that are coincident in two polygons performs the merging. The following two cases are used and are illustrated in figure 5.

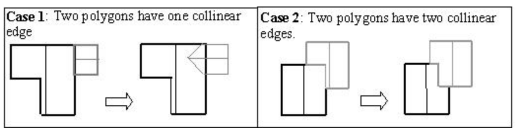

Figure 5: Merge Cases used in the rectilinear roofing algorithm.

The first case occurs when two polygons are separated by a single edge. In this case an extra vertex is added at the midpoint of the edge. The vertex is then moved in an outward direction away from the polygon interior. If it collides with a roof plane then the point is fixed at this intersection point otherwise the original polygon without the new vertex is taken. The second case occurs when two polygons are separated by two connected edges. If this situation arises the polygon vertices incident to the two connected edges are moved outwards away from the polygon interior in a specified direction. The paths of the vertices attempt to collide with a roof plane. The directions of the paths are equal to the direction of the edge, which is incident to the vertex being moved and is not equal to one of the connected edges. Figure 6 illustrates the final result for a rectilinear polygon.

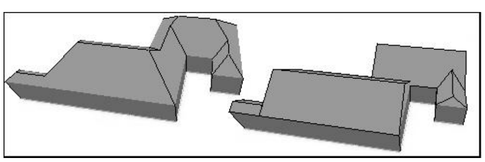

Figure 6: Left: Straight Skeleton used to create a roof model, Right: Rectilinear polygon partitioned into four roof models

6. CONCLUSION

This paper has documented an approach to modelling large environments based on building footprint information. The first stage involves the aggregation of building features into a data structure containing the topology of the building footprints. It achieves this using the Merge Find Set data structure to collect connected line segments. It then partitions the set of connected line segments into building footprints. The second stage explains how the buildings are given an approximately accurate height. The height is calculated from LIDAR data automatically. The final stage is to use one of the roofing methods described. Having a variety of roof models available for any given polygon permits a varied skyline to be obtained for the environment.

Once the base geometry for the urban environment has been obtained the realism can be improved using textures. There are variety of ways this could be accomplished including using a library of textures or procedural methods. The method chosen will need to be stored efficiently, rendered quickly and permit a variety of building appearances to be visualised. To increase the detail of the model the geometry of the walls should be altered to reflect the information depicted in the textures. For instance windows sills and door frames should be extruded from the walls of the buildings. The texture and fine geometry generation stage is the topic for future work.

7. ACKNOWLEDGEMENTS

We wish to thank EPSRC for funding the Rapid World Modelling Project[10].

8. REFERENCES

[1] Y. Parish, P. Mller, Procedural Modelling of Cities, Conference Proceedings Siggraph 2001, Los Angeles, pp 301-308, 2001

[2] C. K. Yap, H. Bierman, A. Hertzman, C. Li, J. Meyer, H. K. Pao, T. Paxia, A Different Manhattan Project: Automatic Statistical Model Generation, Proc. 14th Ann. Symp., Electronic Imaging 2002, San Jose, California, Jan 2002

[3] M. A. Weis, Algorithms, Data Structures, and Problem Solving with C++, Addison-Wesley, 1995, ISBN 0-8053-1666-3

[4] P. Felkel, S. Obdrmálek, Straight Skeleton Implementation, 14th Spring Conference on Computer Graphics, pp 210-218, Budmerice, Slovakia, April 1998.

[5] J. O'Rourke, Computational Geometry in C, 2nd Edition, Cambridge University Press, New York, 1998, ISBN 0-521-64976-5

[6] Ordnance Survey Web Site, http://www.ordsvy.gov.uk/

[7] C. Brenner, Towards fully automatic generation of city models, ISPRS, Vol XXXIII, Amsterdam, 2000

[8] P. A. Flack, J. Willmott, S. P. Browne, A. M. Day, D. B. Arnold: Scene Assembly For Large Scale Reconstructions. Proceedings of VAST 2001, Athens

[9] O.Aicholzer, F. Aurenhammer, D. Alberts, B. Gartner. A novel type of skeleton for polygons. Journal of Universal Computer Science, 1995.

[10] Rapid World Modelling Project, http://www.sys.uea.ac.uk/ research/rwm/rwm1.htm

[11] Urban Simulation Team, http://www.ust.ucla.edu/ustweb/ust.html.

[12] Environment Agency, http://www.environment-agency.gov.uk

[13] C. Baillard, A. Zisserman, Automatic Reconstruction of Piecewise Planar Models from Multiple Views, Proc. IEEE Computer Vision and Pattern Recognition, pg 559-565, 1999.

[14] George Vosselman, Building Reconstruction using planar faces in very high density height data, ISPRS Conference on Automatic Extraction of GIS objects from digital imagery, Sept 1999