REAL TIME SYSTEMS FOR URBAN MODELLING

C. SWIFT [1,2], K. LEINEMANN[1], G. SCHAEFER[2]

[1]*Institut für Angewandte Informatik, Forschungszentrum Karlsruhe, Germany*
[2]*School of Computing and Mathematics, The Nottingham Trent University, U.K.*

Abstract. This paper discusses how real time systems for urban modelling are becoming a realistic possibility. Portable computers bring increased flexibility for the urban modeller but at the same time challenges for the system designer. Portable computing also allows for real time on-site measurement and modelling. The work on real time modelling at the Forschungszentrum Karlsruhe is introduced and a flexible method for describing the geometry of roofs from angles and offsets is proposed. Furthermore, an overview of a prototype system developed is given.

Keywords: urban modelling, architecture, real time, portable computing, object orientated model combined systems, on site, total station

1. INTRODUCTION

The surveying of urban areas to create 2D maps for a variety of uses is now an established practice. However, using this survey data to create realistic 3D models is a technology that is still developing. This process is generally described as "urban modelling" and covers a variety of methods that produce models ranging from single buildings to fully modelled cities. The range of uses varies greatly; models may be required for statistical analysis, as production prototypes for models, to aid in planning decisions or for visual simulations such as virtual tourism.

Despite the large amount of current research in the area of urban modelling, the general opinion of the commercial value of urban modelling varies greatly. The current main attraction of the technology seems to be the visual effect, rather than as a way as presenting data for analysis and decision-making [1].

Despite this lack of a clear role for urban modelling the champions of the technology have identified many potential application areas. A selection of the most common is below:-

- Geometrical models for pollution mapping, wave propagation modelling.
- Urban planning models
- Base models for wooden or plastic architectural models by CNC machines.
- Modelling of cultural heritage monuments.
- Virtual models for tourism.

One of the problems with successfully developing urban models is that the diversity of the above areas creates a wide range of requirements in accuracy, data, speed, and model realism [2]. This results in the dilemma for designers of creating expensive specific systems tailored to individual tasks or using the best compromise of current techniques.
The result of this is that few automated or semi-automated techniques exist. Automated and semi-automated techniques improve the speed of data capture, thus making the process cost effective. If cost effectiveness is achieved then the range of application areas is potentially huge. The large number of academic and research institutions that have projects in urban modelling reflects this potential.

2. PORTABLE SYSTEMS

Portable computers offer urban modellers the tantalising possibility of both measuring and modelling on site. Work done in this area suffers however from the comparatively decreased power of notebook and handheld computers over desktop machines. With the gain in increased portability come various interface problems for software designers. Generally portable computers differ from desktop machines in the following:-

- Portable computers have physically smaller displays that currently have a lower resolution than desktop displays.
- Due to size and power consumption the processing power, memory as well as graphics speed and storage lags behind desktop computer capabilities let alone those of a workstation.
- Input devices differ greatly from touch pads to touch sensitive screens.

 ISBN 3-936150-25-7

- Connectivity with current trends, such as a wi-fi or bluetooth connections are as likely if not more so than a traditional RS232 port.

The most significant from the above differences for the user interface designer are the screen limitations and reduced graphics power. 3D modelling packages normally adopt a tri-view approach to modelling. The user interface has four 3D viewers showing an overhead view, two differing side views and a perspective view. For desktop machine with a 21-inch monitor capable of a native resolution of 1600 x 1200 or more, this presents no problem. However for a small device with a screen maybe a quarter the size and resolution this is impractical. Not only because of the lack of clarity created but also because when working in field conditions the views are too small to be manipulated accurately. A solution to this is to use a single 3D viewer that can switch between the four different viewpoints required. As well as being clearer this also requires less graphics power to run.

The small display also creates two further problems. Displaying the hierarchy and attributes of the project being worked on and giving the user access to the functions that are required to work on the project. The normal 3D CAD system solution is to divide the objects into layers to separate large 3D data sets into manageable chunks and then use toolbars and floating tool palettes to give users access to the functions of the program. The problem with this system is that urban models are large typically extending over kilometres and as such consist of a large number of objects that must be subdivided [1]. Using layers this creates either a very large list of layers that is difficult to navigate or a few layers with lots of objects that are hard to find and edit. Furthermore, one problem with toolbars and floating tool palettes is that whilst they are a good way of showing often used functions they also take up valuable screen space, which is important on a portable device.

Geometrically the relationship between differing sub-divisions in a city/urban area follows a tree shape. This is because they are artificial constructs designed and planned artificially [3]. They can be therefore subdivided into objects such as suburbs, streets, industrial areas, and buildings that can wholly contain other objects. Because of this relationship the whole project can be represented using tree lists. The advantages over layers include:-

- The user can decide which data to show by expanding and collapsing different branches of the tree.
- Single objects can be selectable and also part of higher-level groups.
- By engineering a basic tree object type that other objects are derived from, any object could be part of any branch, with the rules governed by the designer not the system.

Having decided on showing tree lists for the project structure the problem of cluttering the screen with toolbars and palettes is even more pressing. Putting all the functionality of the program in the main menu system presents the user with a bewildering array of options seemingly unconnected to the object or task selected. Another approach is therefore required.

Popup menus can be dependant upon cursor position and selection. This provides the possibility of having the selected object or task dictating which options are available. This can, not only save space but also help lead the user through the program. Using these popup menus to launch dialog boxes for user input and control removes the need for large floating tool palettes similar to those used in image processing packages

3. REAL-TIME SYSTEMS

Traditionally measurement and modelling were separate functions that were conducted by separate programs. The data was measured by the instrument, loaded onto the computer by an interface program and then loaded into a separate modelling package. This method created the limitation that it was not possible to have a real time system. However with the increased take up of portable computers for urban modelling real time systems are becoming a realistic possibility.

A real time system offers several possible advantages over the traditional method: -

- The model can be viewed and created on site and on the journey to and from the site increasing productivity and making recognition of errors easier. Once recognised, errors can be corrected on site. This reduces travel time to make corrections.
- Information can be entered directly onto the finished model during measurement making the need to sketch the survey with pen and paper redundant. (This electronic sketchbook method is introduced in the next section.)
- Traverse calculations, area calculations, and free stations can be included in the software providing quicker traversal, increased flexibility and if correctly implemented reduced complexity for the user.
- Coupling of measurement techniques to modelling allows semi automatic creation of models, including draft creation (sketching) and then measurement of all points within

objects or the use of attributes (angles distances etc) to define objects

Reflectorless and servo motorised total stations are commonly in use today. However the ability to control these instruments by computer is not commonly used. Computer control of these total stations can remove the need for post survey traversal calculations, improve the accuracy of measurement, and speed up the survey process. Because of this, total stations are normally the central instruments in any combined system.

4. OBJECT-ORIENTED MODELLING

As stated previously one of the primary aims of implementing the system in real time is to provide a method of reducing both errors on site and the amount of hand written data created.

To facilitate this a system of object orientated sketching is proposed. The objects that form the building (doors, roofs, ground plans, etc.) are sketched in the 3D viewer by the user before measurement. Rather than calculating the geometry seperately after all sketching has taken place, the sketching, measurement systems and graphics architecture are integrated together to provide real time sketching, measurement and display.

This should be made possible by removing complex surface calculations required and by recalculating only entities affected by each measurement.

The objects to be sketched are implemented as objects within the system. This allows intelligent measuring and definition strategies to be used saving time and simplyfying the measurement process.

5. MODELLING ROOFS

The most documented roof creation algorithm is the straight skeleton algorithm [4]. However this does not allow the creation of standard roof types from measurement offsets or angles.
Despite a lot of research no single algorithm can create all roof types. One of the problems is that roofs are simply so diverse that from one ground plan multiple roof shapes are possible. This diversity increases with the complexity of the underlying ground plan.

Due to this diversity any system wishing to comprehensively model roofs must provide several creation systems. The system should then suggest to the user the correct creation method depending on the type of roof to be modelled and the input parameters available. After creation or update using one of the algorithms the roof geometry can then be passed to the standard roof object and additional attributes set and added as necessary.

6 TRESTLE DEFINITION METHOD

The system for roof modelling developed at the Forschungzentrum Karlsruhe is based on the premise that modelling a roof using its trestles, which define the roofline geometry, is a sensible abstraction since the structure of the roof can be defined by its trestles alone.

Figure 1 shows in blue the trestles that the system would use to define the roofline points of the example roof.

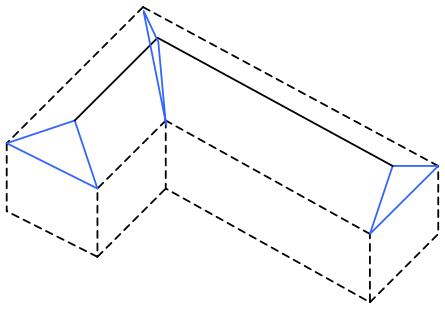

Figure 1 Trestles within a roof structure

6.1 Sketching

Before measurement or definition of the roof the user first sketches the roof object. To sketch a roof the user is required to select the points that form the roof base. The system presents the user with a list of possible roof types based on the number of points selected. This assists the user and ensures the system defines a correct roof type. This is shown in figure 2.
A roof generator object is created that encapsulates the required number of trestle objects to define the roof shape.
Roof types can be dynamically defined at runtime from data files. The definition of the roof types also includes default angles between the trestles and the base of the roof.
This enables the system to render the roof in 3D before a single measurement has been taken or parameter defined. Having sketched a roof it can then be precisely defined by measurement or definition from given parameters whenever this is required.

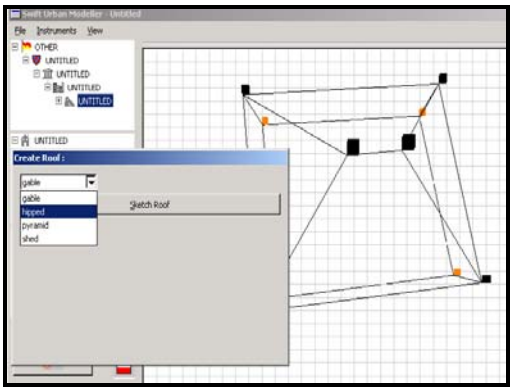

Figure 2 Roof sketching based on selection of roof base line points.

The following vernacular roof types are currently supported:-

- Gable
- Saltbox
- Hipped
- Pyramid
- Shed
- Cross-hipped
- Cross gable
- M-shaped

6.2 Measurement

The sketched roof can be defined by measuring the roofline points using the reflectorless mode on the total station. This requires the existence of a line of sight between the instrument and the points to be measured.
The quickest method is to measure a single point as the roofline point. This however is likely to be only accurate to within a few centimetres as the laser targeting spot diverges over distance. Work has been done on using several measurements to get a more exact definition of a target point [5]. However, this sacrifices speed for enhanced accuracy. The user should therefore pick the measurement type most appropriate to the application area of the urban model being generated.

6.3 Parametric Definition

There are several cases when taking a measurement of the roofline points is not desirable or possible. For example, it may not be possible to establish a direct line of sight between the instrument and the roofline points. The user may also be in possession of architectural plans detailing dimensions and/or angles of the structure. Using these to define the roof geometry may the clearly be preferred. Another possibility would be that the user is defining a large urban area possibly from pre-measured ground plans mainly for visual purposes and therefore millimetre accuracy would not be required.

To cope with these situations the system gives the user the option of defining each of the roofline points based on offset distances from the walls or using the angles made between the trestle and the roof base. Due to the simplicity of the calculations once a parameter is redefined the roof geometry can be updated in real time. Details on roof geometry algorithms and calculations are given in the Appendix.

The system resists using constraints based on the geometry of the roof type to reduce the number of measurements as this would also reduce the flexibility of the system and hence lead to undesired results for irregularly shaped ground plans. Flexibility is retained within the system by allowing the user to define each roofline point from measurement, angles or offsets. Having defined a point using one method it can then be redefined using another. This allows the methods to be mixed within a single roof construct.

7. REAL TIME SKETCHING SYSTEM

This section gives an overview of a prototype system that was implemented in (Visual) C++ using the Fox Toolkit and OpenGL to provide a GUI interface suitable for mobile computers. Its main aim is to prove some of the concepts outlined in this paper. In its current version it has the following capabilities: -

- Implementation of a real time sketching system allowing the user to sketch building floor plans, extrude the floor plan and then sketch a roof before measurement. Giving full 3D visualisation of the scene.
- Provision of a custom TCR Interface API to allow real-time measurement from a TCR30X series Leica total station.
- GUI allows visualisation of project hierarchy on a small (800 x 600) notebook screen without sacrificing the ability to view the scene as adjustments to the project are made.
- A roof generator allows creation of roofs as discussed in Section 5. These can then be adjusted by angles or by measuring the roofline points using the laser on the TCR.
- A 3D viewer supports plan, side and projection views with zoom in all three and translation in the plan and side views.

A screen shot of the prototype platform can be seen in Figure 3.

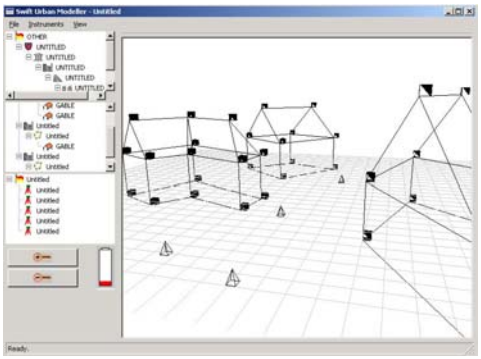

Figure 3 Urban modelling prototype system.

8. CONCLUSIONS

The development of real time portable urban modelling systems is now technically possible. However despite great potential, computer based urban modelling systems are still too much in theory and discussion and as such of no real interest to the software developers [1]. Standardisation and a target market must first be established before the concept meets its potential.

REFERENCES

[1] BOURDAKIS V, 1998, On Developing Standards for the Creation of VR City Models, Laboratory of Enviromental Communication and Audiovisual Documentation, University of Thessaly, Greece.

[2] MUELLER H, Three-Dimensional Virtual Reconstruction of Buildings: Techniques and Applications. Institute for Spatial Information and Surveying Technology i3mainz, University of Applied Sciences at Mainz, Germany.

[3] ALEXANDER C, April 1965, A City is not a Tree parts I, Architectural Forum, Volume 122, No 1, pp 58-62 (Part I),

[4] AICHHOLZER O and AURENHAMMER. F, 1996, Straight skeletons for general polygonal figures in the plane. *Proc. 2nd Annual International Conference Computing and Combinatorics*, pp. 117-126. Lecture Notes in Computer Science 1090, Springer.

[5] SCHERER, 2002, Advantages of the Integration of Image Processing and Direct Coordinate Measurement for Architectural Surveying – Development of the System Total -, XXII International Congress Washington DC. USA April 19-26-2002

APPENDIX

Basic trestle algorithm

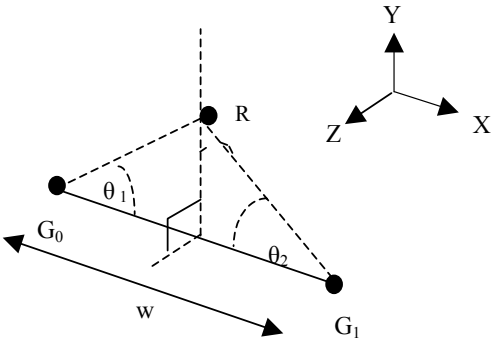

Figure 4 Roof with roofline point over G_0G_1

Figure 4 shows a roofline point R that lies directly above G_0G_1. The point can be defined from the angles θ_1 and θ_2.

First the distance G_0G_1 is calculated by Pythagoras's theorem.

$$w = \sqrt{(G_{1x}\text{-}G_{0x})^2 + (G_{1z}\text{-}G_{0z})^2}$$

Next we build two equations to describe the lines G_0R and G_1R. These are straight lines and therefore in the format $y=mx+c$. Firstly the values of m are calculated as

$$m_1 = \frac{\cos\theta_1}{\sin\theta_1} \quad m_2 = \frac{\cos\theta_2}{\sin\theta_2}$$

The intercept values of these simultaneous equations are C_1 and C_2. C_1 is then equal to 0 and C_2 equal to w. The two equations are solved simultaneously to find the height h of the intersection and the distance along the line G_0G_1 is the solved value of c.

From this the x, y, and z coordinates of the roofline point can be calculated relative to G_0.

$$R_x = G_{0x} + \left(\left(\frac{c}{w}\right) \times (G_{1x} - G_{0x})\right)$$

$$R_y = G_{0y} + h$$

$$R_z = G_{0z} + \left(\left(\frac{c}{w}\right) \times (G_{1z} - G_{0z})\right)$$

Offset trestle algorithm

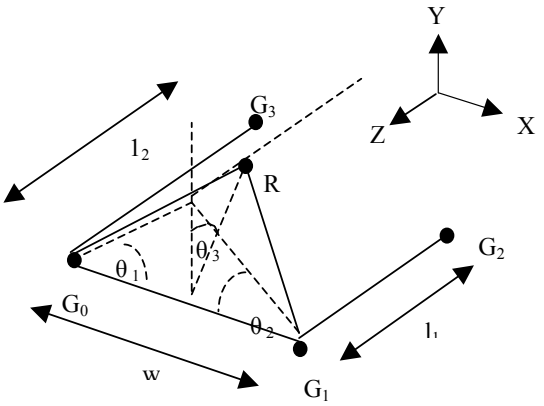

Figure 5 Roof with roofline point translated along ($G_1G_2 + G_0G_3$)

The previous solution is not going to be applicable to many roofline points. This is because the slope of many roofs is along the roofline as well as across it (shown as θ_3 in Figure 5). This occurs mostly at sections where the roof ends. Therefore it is necessary for the definition of this slope within our algorithm.

Steps 1 and 2 from the previous example are utilised and then an additional value l is calculated.

$$l=\sqrt{(G_{2x}-G_{1x})^2+(G_{2z}-G_{1z})^2}+\sqrt{(G_{3x}-G_{0x})^2+(G_{3z}-G_{0z})^2}$$

This is the length of the vectors G_1G_2 and G_0G_3 summed. This is because the roofline is defined as $G_1G_2 + G_0G_3$. Defining the roofline from the sum of two vectors takes into account the shape of the roof.

Then, the roofline point is calculates as

$$R_x = G_{0x} + \left((c/w)\times(G_{1x}-G_{0x})\right) + \left(\left(y/\tan\left(\frac{\theta_3}{l}\right)\right)\times\left((G_{2x}-G_{1x})+(G_{3x}-G_{0x})\right)\right)$$

$$R_y = G_{0y} + h$$

$$R_z = G_{0z} + \left((c/w)\times(G_{1z}-G_{0z})\right) + \left(\left(y/\tan\left(\frac{\theta_3}{l}\right)\right)\times\left((G_{2z}-G_{1z})+(G_{3z}-G_{0z})\right)\right)$$

Definition from Offsets

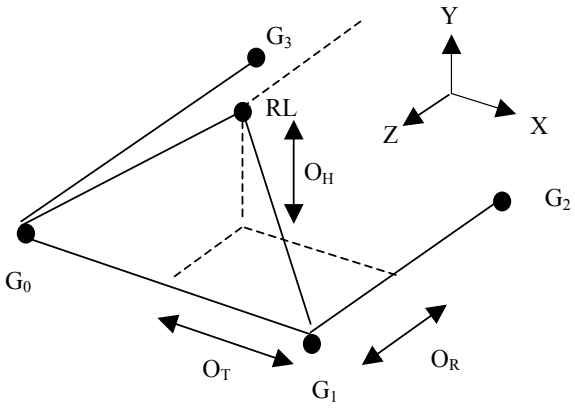

Figure 6 Roof with roofline point determined by offsets

Below is the definition of the roofline point R calculated from the 3 required offsets (see Figure 6). Again as with the offset trestle definition the roofline point is translated along ($G_1G_2 + G_0G_3$).

$$R_x = G_{0x} + \left((o_T/w)\times(G_{1x}-G_{0x})\right) + \left((o_R/l)\times\left((G_{2z}-G_{1z})+(G_{3z}-G_{0z})\right)\right)$$

$$R_y = G_{0y} + o_h$$

$$R_z = G_{0z} + \left((o_T/w)\times(G_{1z}-G_{0z})\right) + \left((o_R/l)\times\left((G_{2z}\ G_{1z})+(G_{3z}-G_{0z})\right)\right)$$

where o_h, o_T, and o_R are the height, trestle and roof point offsets respectively.