4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage
VAST (2003)
D. Arnold, A. Chalmers, F. Niccolucci (Editors)

Modelling the Walled City of Nicosia

M. Dikaiakou*, A. Efthymiou and Y. Chrysanthou

Department of Computer Science, University of Cyprus

Abstract

This paper presents our initial results in producing a 3D model of the Chrysaliniotisa Quarter in Nicosia using the GIS data of the region, and an analysis of the structure of the area's buildings. We tried to create a partly-automatic system, which was aimed at producing a realistic model of the geometry and architectural style of the district, rather than an exact reconstruction of every detail. The residential buildings of the particular area follow some well defined architectural styles, which allows us to follow an automatic building generation, based on the 2D digital data and a library of predefined 3D building blocks. Starting from the GIS file, the data is sorted, examined and processed to detect the houses' features and style as accurately as possible. The 3D model is then constructed by stitching together the appropriate blocks from the component library. Besides the automatic-generation method, we have used ImageModeler from RealViz to create accurate 3D representations of landmarks and exceptional buildings as well as for the building blocks.

Keywords:

Typology, component based modelling, automatic reconstruction, urban environment

1. Introduction

Three-dimensional city models provide a valuable tool for numerous applications and they have radically increased in popularity in the past decade. The demand for greater detail and realism has also risen through the years, rendering the creation of such models very challenging. A city is a very intricate synthesis of streets, buildings, architectural styles, eras; all of these aspects need to be taken into account and correctly reproduced. An urban representation is comprised of thousands of polygons and many different textures, which must be rendered in real time to give a realistic effect.

Many modern cities, such as Bonn, London, New York, have been reproduced in 3D with the use of automatic or semi-automatic methods. We aim to design a partly-automatic method to model part of the old city of Nicosia, using the ground plan of the area, and an official classification of the buildings according to their shape, size and position. Furthermore, we will not be using simple textures for our models; we have developed various "building-components" - doors, windows, balconies, arches, kiosks - and we intend to create 3D buildings by combining these components. A rule set is applied to determine which components should be selected for each house, and where they should be positioned. In the area we consider there are certain buildings, for example a couple of churches, which are not possible to reconstruct with the rule set. These are created and inserted separately into the 3D world.

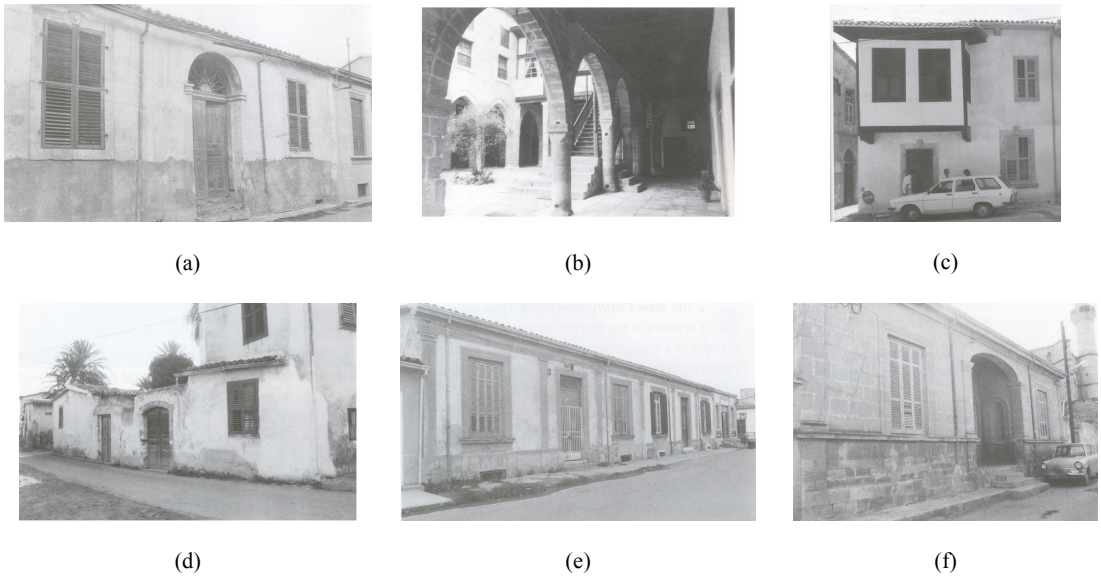
2. Previous Work

There are many approaches to urban environment 3D modelling with a more or less realistic effect.

Much research is based on photogrammetric methods, which make use of many photos of the area [6].

A different class of methods is comprised by those that apply sets of rules on various data sets to produce a

* Marianna was a Masters student at University College London visiting University of Cyprus for the duration of the project



(a) (b) (c)

(d) (e) (f)

Figure 1: ***In the top row we see images of Original Courtyard-Houses. In (a) we see a typical façade, in (b) we have a view to the courtyard with the arches and in (c) we see a kiosk which was sometimes used instead of a balcony, especially during the Ottoman period. In (d) we see 2 Minimal Courtyard-Houses, typically having a more modest door and one window to the road. In (e) and (f) we have three Planned Serial-Houses and one New Courtyard-House respectively.***

city model [3, 14]. Our approach also falls in this category, so we focused our research in this field. The following papers offered great insight to the subject:

Extracting geometric information from architectural drawings by Kernighan and Van Wyk [8]: The program extracted the coordinates and composition of walls from the digital floor plan, and discarded all entities except LINES and POLYLINES it found. Then, it attempted to replace "double" or "multiple" walls with single logical walls. In general, it can be used to remove redundancy in any kind of ground plans, so it was kept under consideration for our own project.

A different Manhattan project by Yap et al. [15]: This project uses a number of statistical parameters, for example population and building density, along with digital maps, to yield a 3D model of Manhattan. The system entails three phases, where the input is manipulated, in order to decide upon the division of the city into neighbourhoods, and of each neighbourhood into blocks and lots.

Procedural modelling of cities by Parish and Muller [10]: The authors have created patterns of different ground plans, and are able to apply them individually or overlaid, to create a city's street network. Then, the area is divided into allotments, where buildings will be placed.

Some other very useful sources of information are the survey on 3D urban models of Narushide [11] and the web page of www.casa.ucl.ac.uk/3dcities [19]. The latter has several publications on urban modelling there, and we found many appealing points and methods among them.

3. Typology of Nicosia houses

Before proceeding to a discussion of the project's methodology, we think it is useful to summarize the structural styles of the modelled area's houses. The rule set we defined was based on the architectural types identified by Danilo, in his book "Typology of Nicosia Houses" [5]. He identifies four main categories (see Figures 1 and 5):

- Original Courtyard-House Type: it has a large inner courtyard, and the building or buildings of the property are organized around it. The door opens to a portico, which leads directly to the courtyard. Rather large atriums are common among these houses.

- Minimal Courtyard-House Type: this is quite similar to the above category, but it is much more modest in size. The door might open to a portico or directly to the courtyard. The majority of these houses only have one floor, or a very small second floor.

- Planned Serial-House Type: these houses first appeared around the late 19th century; a line of houses with the same ground plan was arranged over some area of the city. In this type, the courtyard is moved to the back of the house, and it becomes smaller; it is no longer the centre of the house. In almost all cases, no building stands inside the yard or adjoined to the back wall.

- New Courtyard-House Type: this is the most recently developed house; it was very popular around the 1930s. Essentially, it is a combination of the two previous types. It uses a Serial Type element in the part of the plot facing the street, therefore the door leads into the house, which is larger in size as in the third type. However, a wing of the house continues along the side of the courtyard, as for the Courtyard Types.

Furthermore, the architects of NMP (Nicosia Master Plan) gave us detailed descriptions of the positioning of doors, windows, balconies and kiosks on a typical Nicosia house façade. These features were, in fact, positioned on axes, that had set distances between them. This was indispensable later on, when we were deciding where to place such components on our models. An example of such a façade is evident in Figure 2.

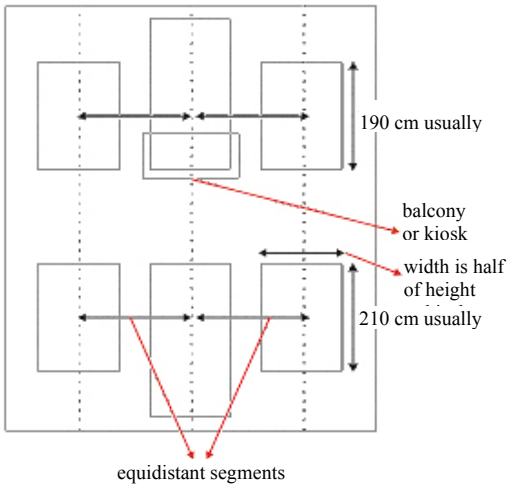


Figure 2: ***Depiction of axes of a façade.***

It is important to note that what is described here is the traditional architecture which prevailed until early 20th Century. Houses built after that kept little, if any, of the above characteristics. The Chrysaliniotisa quarter is the only sizeable region in the Government controlled part of Nicosia that has preserved that character with almost no modern buildings in it.

4. Method

The project consists of two distinct 3D representation techniques; the individually modelled components and the rule-based automatic generation. Our first step was to use ImageModeler to produce our component library, as well as models of unique buildings.

The GIS data was the input to our algorithms; we parsed it and applied heuristics to detect inner and outer walls, roads, and even had a first attempt at identifying the architectural style of each house. The final step was producing the 3D output, which demanded great care while positioning the 3D components, and creating the roofs. We created a graphics panel to review the output of our algorithms. These steps are described in detail further on.

4.1. Construction of the 3D Component Library and the Special Buildings

To add to the realism of the urban scene, we decided not to use simple textures on the buildings, but to create 3D prototypes of windows, doors, balconies, kiosks, arcs and roofs, and use them to assemble appropriate facades and atrium surroundings. We used ImageModeler for this task; we went around the old city of Nicosia and took pictures of characteristic examples of doors, windows etc. We needed at least three different views of each component and at least four for each unique building, as input for ImageModeler. Later on, when the 3D model of the house is created, the components that are used to assemble its façade - door, windows, balcony - and atrium - arcs- are selected according to the house's style. After the components are extracted in VRML format (see Figure 3 for example of doors from the library), they are used as the most detailed part of an LOD node.

Figure 3: ***3D models of typical doors for "Original Courtyard Houses", from our component library.***

Besides the components needed for each house, we wanted to create 3D models for the area's unique buildings that could not be reproduced by our rule set. This

task was much more problematic than creating the individual components because of the difficulty in acquiring suitable pictures, and the complexity of the architecture of such buildings. The narrow streets did not allow for panoramic shots of whole buildings, and even if we could find an open space, capturing the roof was almost impossible without hiring a crane. Our most successful attempt was the Crysaliniotisa church, where we were assisted by the area's residents, and we were able to enter their homes and capture images of the church from second floor balconies facing it. It took a lot of photographs and careful modelling to get decent textures and geometry for the roof. The result can be seen in Figure 4.

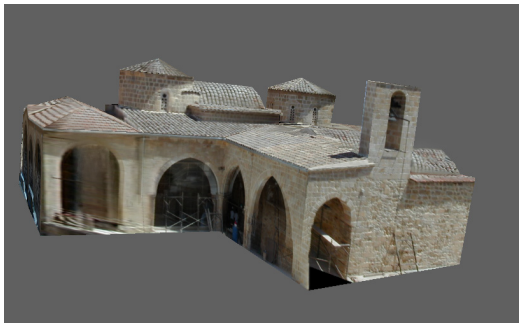


Figure 4: ***A 3D model of the Chrysaliniotissa church, built with ImageModeler.***

4.2. Reading and Parsing the GIS Input Files

The digital data we received was organized in seven layers; each layer contained a different set of information. There was one layer for the administration quarters, one for the administration blocks, one for parcels (lots), one for buildings, and three with various topological data. The file format was the standard MapInfo .mif-.mid. Unfortunately, the digital maps were not as thorough as the printed maps we had reviewed thus far, so we had to make adjustments to our algorithms to allow for road and building style detection.

We used three of the seven layers of data available to us, the ones containing the administration blocks, the parcels, and the buildings. These files had to be read consecutively, to allow the program to build a hierarchy pyramid. First, the administration blocks were read, then the parcels were inserted into their respective block, and finally, the buildings were added to their respective parcel. Thus, we would be able to work on smaller parts of the area when testing and generally allow greater flexibility in the program. The parcels and buildings are recorded in the files in no particular order, so we tried to create a form of edge index, which would help us when detecting neighbouring houses and parcels, roads and courtyards. The edges were sorted by their orientation – in rough numbers – which greatly enhances our search functions.

4.3. Identifying Roads and Building Features

As we have mentioned before, each building of the 3D area reconstruction is assembled with the use of the appropriate components. To place these components correctly, we need to establish which sides of the house face the road, which face the inner courtyard, and which are adjacent to other buildings or surrounding walls. We must also know the height of the house – whether it is one or two storeys high. To discover which edges are adjacent, we needed to compare the edges of all parcels in each administration block to each other. Since two adjacent lines will have the same angle, when we compare the edges of two different parcels, we only need to compare those with the same orientation. Once two such edges are found, we check if their endpoints fall within a small distance from the other edge. If they do, the edges are tagged as "neighbouring"; this attribute is used further in the identification process, and during the 3D reconstruction.

When all the parcels of each administration block are examined, the only edges that are not tagged are the ones in the front of the parcel, facing the road. These are set as "facing road". It may seem lengthy to go through the procedure of sorting edges by orientation, and then retrieving and comparing the matching ones, but it has actually proved very efficient. The task of identifying "facing road" edges was attempted without going through the sorting and simply comparing every edge with every other edge in each administration block. It was timed and took 8.2 sec to complete; the approach followed took less than 3 sec, which means we get a significant speed-up, for the portion of Nicosia modelled up until now – the machine used to conduct these tests is mentioned in the results section. For larger input, the benefits will be even greater.

After the parcels, the buildings file is opened and read. Each building is added on the parcel it belongs, and once again, the buildings' edges are sorted by orientation. While each edge is read, it is checked against the corresponding parcel's edges – the ones with the same orientation. If the building-edge is found to fall within a very small distance from a "facing road" parcel-edge, the former is also set as "facing road". If not, the building-edge is checked against the "neighbouring" parcel-edges. If it falls within a small distance from this it is also set as a "neighbouring" edge. These edges do not need rendering in the 3-D reconstruction, since they are hidden by the building they are adjacent to, they can even be removed from the model. The remaining edges are set to be "facing atrium". An exception is made if no part of the building

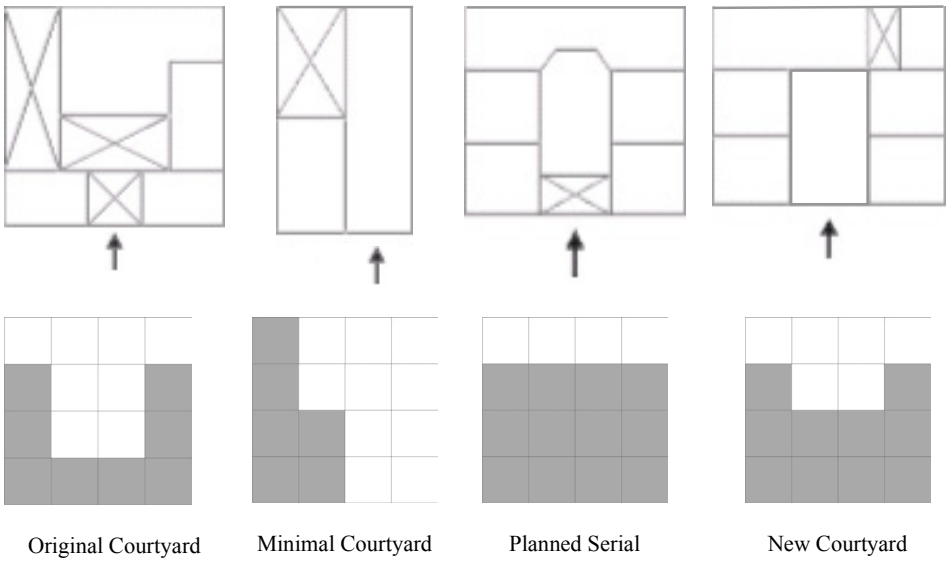


Figure 5: ***Ground plan and master template for typical Nicosia houses. The crosses in the ground plan indicate a covered open space. The arrows point to the main entrance.***

is found to face the road with this process, which is rather rare. In that case, all the surrounding walls are visible from the road, so they are also set to be facing it.

Among the building edges that are on the road, only one should have a door and windows. We find the longest edge, and set it as the building's façade; it will be comprised by a door and windows, and possibly a balcony or kiosk, if this is a two-storey house.

The building's height should have been given in the input files, but unfortunately it wasn't the case. The old paper maps done by the British while they were in charge of the island, are on a scale 1:500, and it is clear in them what parts are one- or two-floors high and even what is a portico or atrium (covered open spaces are marked by diagonal lines). However on the new digital maps most of this information is lost. For that reason and until we have better information, we decided to select in random 20% of the houses and set them to have a second floor. In most cases, there were more than one building in each parcel, so we wanted to consider one of them as the main house. If there was only one house facing the road that would be marked as the main house. If not, the house with the largest area would be marked.

Furthermore, we want to find out which of the parcel's edges will constitute the courtyard's surrounding walls. Any parcel-edges that do not coincide with building-edges are indicated as "visible".

4.4. Typology Classification

An algorithm was developed to extract as much information as possible from the outline of a house, in order to classify it successfully to its type. The process would probably be time-consuming, but even a few minutes of processing were deemed acceptable, since this procedure would be taking place before the actual modelling.

The algorithm starts off by considering each parcel separately and simplifying it to a 4-sided shape. This is done by taking the end-points of two consecutive edges, finding the line joining them, and distance of their intersection point from that line. We discard the edges with the smallest distance, and substitute by the connecting line.

We continue to collapse edges, until a 4-sided shape is left. Then we divide the shape into 2 triangles, and perform 2 affine transformations (one for each triangle) to fit corners into a new square's corners. We can apply these transformations with no further calculation to the interior of each triangle, or apply a weighted average, depending on the distance of each interior point from the line separating the two triangles.

This process yields a square, for the parcel, and the buildings inside the parcel take up some of its space. We divide the square into 16 equal squares, and mark each of the 16 with 1, if it is mostly filled with a part of a building, or 0, if it is mostly empty. This way, we produce a template which describes as closely as possible the setting of the building and the courtyard in the parcel. We compare this template with the four master-templates of Figure 5 to find the one that matches the house best. The parcel is assigned the architectural style of the master-template that provides the closest match.

Thus, when constructing the 3D model of the house, we can select the components that correspond to its architectural style.

The classification algorithm is not functioning fully at this stage. It is still work in progress.

4.5. 2D Visualisation

We created two panels to assist us in reviewing the digital data, a graphics panel where the user may view the area map, and a text panel, where the x, y coordinates of any click on the map appear. This feature was incorporated into the project after the initial planning was completed, but, nevertheless, provided more than basic functionality. The graphics panel was designed to allow changes in the colours of different parcels and/or administration blocks. A second panel was also used, where useful textual information regarding the buildings on the map appear. It was considered useful to be able to click on any house on the area, and view its coordinates, centre, or unique building code on the text panel. Any other information that would be needed in the course of the project was also easy to display. An example of the graphics panel is presented in Figure 6; the screenshot shows the result of the segmentation of the area into building blocks, while we were trying to isolate the roads. Even though we had not planned to render 2D graphics, the module proved to be a very useful tool while debugging, and while improving the various identification algorithms.

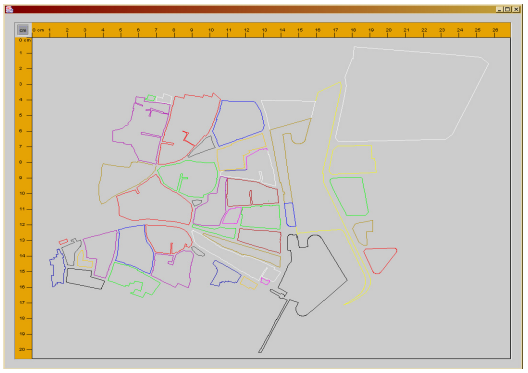


Figure 6: ***Output of segmentation into building blocks on the 2D graphics panel***.

5. 3D Reconstruction

At this point, all the information needed to create a 3D model building has been assembled. This module uses the data from the input files and the output of the heuristics module. First of all, the ground is created for each parcel, by connecting all of its vertices. The actual buildings of the old city are not of one single colour; therefore we wanted the virtual buildings to have varying colours. We defined five colours that are quite common in the area, and randomly select one for each parcel, as it is created in the VRML file. All the buildings inside the specific parcel will have that colour.

The facades and atriums are handled separately, since they are assembled by individual components. All the other edges in a parcel or building become 3D walls with a slight manipulation of their 2D coordinates. Specifically:

- The x-coordinate remains as is in the transform from 2D to 3D
- The y-coordinate is 0, for the lower part of the wall, and takes the building's height, for the upper part. The height of each house depends on whether it has one or two floors.
- The Z of each 3D vertex is -y. If we use y and not -y, we get a mirror of the area we want. To translate correctly from a 2D Cartesian system to a right-handed 3D world, we need to negate the 2D coordinate.

To create the façade, we considered the length of the edge we would use. The first component to be added would be the door; depending on the space left, we may add one or more windows. Each component has a set size, so we can calculate the distance between them, as well as the "padding" needed to complete the wall. A similar approach is used for the edges on the interior, surrounding the courtyard. The arches have a set size, so we add as many as can fit, and a blank-wall "padding" is added for the rest. To beautify our model and bring it closer to the prototype, we randomly added palm trees to the courtyards. Such plants are very common in the old city, so we found a relatively simple 3D model of a palm tree and scattered it around.

Furthermore, this module creates a VRML file with the selected area in 3D. The building components are separate VRML files – the output of ImageModeler – and they are included in LOD nodes. They are only rendered when the viewer reaches close enough to be able to distinguish the depth and detail of these features, otherwise flat textures are used instead. This way, we don't burden our model with too much unnecessary detail.

5.1. Creation of Roofs

Initially, we designed a simple algorithm to produce the roof of each house using a heuristic approach, but it failed in many situations. Then we came across the work on "straight skeleton" computation [2, 7] and we are using it in the current model, which is much more stable and reliable. The straight skeleton was used earlier for the same purpose by [Laycock03], although we were not aware of it until after they published their paper recently.

The straight skeleton can be defined as a propagation of wavefronts. A wavefront starts on each edge and propagates moving away from the edge in a direction normal to it. The straight skeleton, is then defined as the "paths" followed by the vertices of the wavefronts (Figure 7)

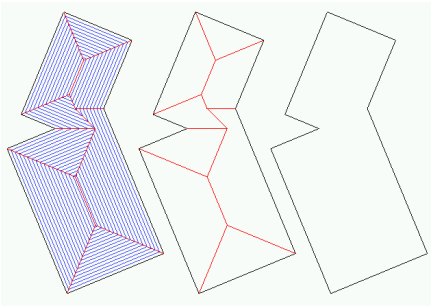


Figure 7: ***Example of "straight skeleton" computation.***

The algorithm for the construction of the topology of the straight skeleton, but not for the 3-D representation, was given to us by Petr Felkel and Stepan Obdrzalek in C++; the rest ofour program was written Java. We compiled the algorithm of the straight skeleton into a windows .dll file and used JNI to incorporate it into our program. The input that we used for the algorithm was the vertices of each building; the output was the topology of their straight skeleton. Then we constructed the polygonal roof from the topology of the skeleton using our own algorithm, so that all faces would have the same slope and the right height based on thelocal architecture.

6. Results

The algorithms' performance was tested thoroughly, and successive minor alterations took place before they reached an acceptable performance level. Most procedures included thresholds, which needed to be tested many times on different test data-sets, to find the best combinations. In its current state, the program is almost always correct when detecting the various building features. One of its most important tasks is to distinguish the edges of each house that are visible from the outside, and the ones that are facing the inner courtyard. It does so successfully in the vast majority of cases; it faces problems only when the building under inspection has no edges close to the road, and its entrance is not placed on the "front". Even so, it manages to identify that such a building is in fact visible from the road all around its parcel. The courtyard-edges are a little easier to identify, and there are no cases of false positives or negatives, with this data set.

Another issue was the use of components in the 3D reconstruction, to add realism to the model. The rules set forth by the NMP are followed quite well, and indeed there seems to be symmetry in all of the facades of the model. However, using these 3D components increases greatly the number of polygons in the scene. As mentioned, we have used LOD nodes for all of the components; only through the careful use of this is it possible to render in real-time. We use the 3D geometry only for very few buildings that appear from close up.

7. Conclusions

Throughout the project, we were given the opportunity to use and test two approaches to 3D urban modelling. Although the project is far from finished, it is still very much work in progress, we have reached certain conclusions as to their value and their ability to provide a realistic and accurate result for this task.

With the use of ImageModeler, someone proficient with the program can produce very good 3D models. But it requires a considerable effort and furthermore, the software calls for many high-quality images, which need to be such that convey information for the whole object. When, the object is a building, located in a heavily built up area, such as the core of a town, it is very difficult to obtain pictures covering all its sides and roof. The output of the program is indeed realistic, if one devotes enough time on it, but the models might be very detailed. This means that in some cases they can be comprised by a large number of individually textured polygons, so possibly not too many of such objects should exist in the same scene.

Automatic methods have been used in urban modeling for years. We believe that our system was quite successful in its model generation, and would perform even better, had we received the data we wanted. Our digital maps did not include many of the features present on the printed ones. The latter specified which part of the house was the atrium, which parts were one or two storey high, and distinguished a kiosk from a balcony. Furthermore, the roads were clearly outlined, thus eliminating the need for our intervention in detecting them. These features would be very useful input for our system, and we are hoping they will be included in the next edition of the maps.

The system was tested on an Intel PentiumIV, 1.6 GHz PC, with 256 MB DDR RAM, and proved to be rather quick, since it requires roughly 15 seconds to run the algorithms on the whole Crysaliniotisa area and output the VRML file.

8. Future Work

The model may be extended in many ways. First of all, there are tasks, such as the road recording and the building type identification, which are left unfinished. They would make a great addition to the project. Then, one could create an application used in city develop-

ment, where the user may remove or add buildings to the 2D map, and will be able to view the new 3D scene, to examine the potentials of such a change. Another supplement that would enhance the project is using the elements in the four remaining data layers; they contain smaller details of the map, such as fences, tanks etc. We could also add a better variety of vegetation in the courtyards, and make a much more realistic model.

Acknowledgements

This research is supported in part by the EU funded project CREATE (IST-2001-34231). We wish to thank Nicosia Master Plan for their advice and support and Petr Felkel and Stepan Obdrzalek for sharing the straight skeleton code with us.

References

1. O. Aichholzer, F. Aurenhammer, D. Alberts, B. Gärtner, "A novel type of skeleton for polygons", J. Universal Computer Science, 1(12):752-761, 1995
2. O. Aichholzer, F. Aurenhammer, "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, 1996
3. S. Browne, P. Flack, J. Willmott, A.M. Day and D.B. Arnold. "Scene Assembly for large Scale Reconstructions", Proceedings of VAST 2001, Athens November 2001
4. M. C. Daconta, A. Saganich, E. Monk, "Java 2 and Javascript for C and C++ Programmers", Wiley, USA, 1999
5. D. Danilo. Typology of Houses of Nicosia. United Nations Development Program, Nicosia 1997. ISBN 9963-8272-0-9
6. P.E. Debevec, C.J. Taylor and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proceedings of ACM SIGGRAPH 96*, ACM Press
7. P. Felkel, S. Obdrzalek, "Straight Skeleton Implementation", 14th Spring Conference on Computer Graphics, April 23 -25, Budmerice, Slovakia, 1998.
8. Kernighan and Van Wyk. "Extracting Geometric Information from Architectural Drawings". Proc. Workshop on Applied Computational Geometry, May 1996, pp. 82-87.
9. R. G. Laycock, A. M. Day. Automatically Generating Large Urban Environments based on the Footprint Data of Buildings. *The 8th Solid Modeling Symposium, Seattle, USA*, 2003
10. Parish. Yoav I.H. and Muller Pascal. "Procedural Modelling of Cities", in Proc of ACM SIGGRAPH 2001, pages 301-308.
11. Narushige Shiode. 3D urban models: recent developments in the digital modelling of urban environments in three dimensions. GeoJournal 52, 2001.
12. R. Pressman. Software Engineering: A Practitioner's Approach. 4th ed. McGraw Hill. New York: 1997.
13. M. Slater, A. Steed , Y. Chrysanthou. Computer Graphics and Virtual Environments. Addison Wesley. London: 2002
14. P. Wonka, M. Wimmer, F. Sillion, William Ribarsky. Instant Architecture. *Siggraph 2003.*
15. Yap, C.K., Biermann, H., Hertzman, A., Li, C., Meyer, J., Pao, H.K., Paxia, T. "A Different Manhattan Project: Automatic Statistical Model
16. A Guide to 3D world creation http://philliphansel.com/index.html
17. Designing Roofs of Buildings http://www.sable.mcgill.ca/~dbelan2/roofs/roofs.html
18. Centre for Advanced Spatial Analysis http://www.casa.ucl.ac.uk
19. Tutorial of VRML97 http://www.sdsc.edu/~nadeau/Courses/Siggraph98vrml/vrml97/slides/mt0000.htm
21. The VRML 2.0 Sourcebook http://www.wiley.com/legacy/compbooks/vrml2sbk/cover/cover.htm

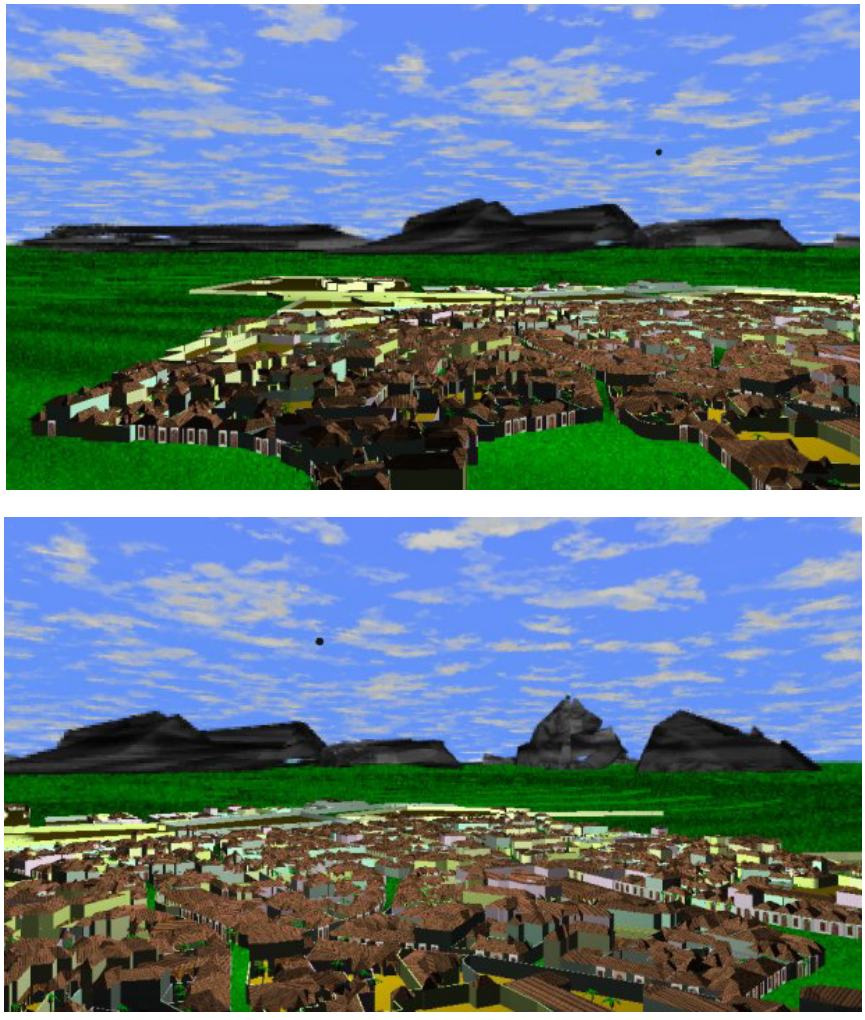


Figure 8: ***Screenshots of our model in its current form.***