Creation of a 3-D city model of Zurich with CC-Modeler

Author(s):
Grün, Armin; Wang, Xinhua

Publication Date:
1998

Permanent Link:
https://doi.org/10.3929/ethz-a-005746848 →

CREATION OF A 3-D CITY MODEL OF ZURICH WITH CC-MODELER

Armin Gruen   Xinhua Wang
Institute of Geodesy and Photogrammetry
Swiss Federal Institute of Technology (ETH) Zürich
ETH Hönggerberg, CH-8093 Zürich, Switzerland
Fax: 41-1-6331101
E-mail: agruen@geod.ethz.ch   wang@geod.ethz.ch

ABSTRACT

In this paper, we introduce a semi-automated topology generator for 3-D objects, CC-Modeler (CyberCity Modeler). Given the data as point clouds measured on Analytical Plotters or Digital Stations, we present a new method for fitting planar structures to the measured sets of point clouds. While this topology generator has been originally designed to model buildings, it can also be used for other objects, which may be approximated by polyhedron surfaces. We have used it so far for roads, rivers, parking lots, ships, etc. CC-Modeler is a generic topology generator. The problem of fitting planar faces to point clouds is treated as a *Consistent Labeling* problem, which is solved by *probabilistic relaxation*. Once the faces are defined and the related points are determined, we apply a simultaneous least squares adjustment in order to fit the faces jointly to the given measurements in an optimal way. We first present the processing flow of the CC-Modeler. Then, the algorithm of structuring the 3-D point data is outlined. Finally, we show the results of data sets of Zürich that have been produced with CC-Modeler.

1. INTRODUCTION

The generation of 3-D city models is a relevant and challenging task, both from practical and scientific point of views. There are two basic research topics involved in the generation of 3-D city models. One is the data acquisition and the other is data management and manipulation. The former is to generate and model the data set from data sources. The latter is to investigate the data structure, data model, spatial analysis tools, etc. In this paper we will introduce our new method for data acquisition of 3-D city models. The second theme will be investigated in another paper (Wang, Gruen, 1998).

Photogrammetry is an appropriate tool to provide information about man-made objects, vegetation cover and the like. Recently, many approaches for automated and semi-automated extraction of buildings and roads from aerial images have been proposed (Gruen et al., 1997). Due to the complexity of natural scenes and the lack of performance of image understanding algorithms, the fully automated methods cannot guarantee results stable and reliable enough for practical use. Therefore, we are investigating also semi-automated approaches which would give the human operator strong computational support in order to generate 3-D city models from aerial images efficiently. Previously, we developed a method which fits generic building models to measured, unstructured 3-D point clouds, which have been generated by a human operator on an Analytical Plotter or a Digital Station (Dan, 1996, Gruen, 1998). This system, TOBAGO, although well proven in many pilot projects, is restricted to the modeling of buildings. With CC-Modeler (CyberCity Modeler) we present a new method for fitting planar structures to measured sets of point clouds. While this topology generator has been originally designed to model buildings, it can also be used for other objects, which may be approximated by polyhedron surfaces. We have used it so far for roads, rivers, parking lots, ships, etc. CC-Modeler is a generic topology generator and, from a practical point of view, it can be considered a generalization of our previous modeler TOBAGO. However, it follows a totally different algorithmic principle.

In the following section we will introduce the overall data flow scheme of CC-Modeler. In section 3 the key reconstruction algorithm is explained and in section 4 some data sets of Zürich which have been generated with CC-Modeler are presented.

2. GENERAL DATA FLOW

To generate 3-D descriptions of man-made objects from aerial photographs involves two major components: photogrammetric measurements and automated structuring. In CC-Modeler, the feature identification and measurement is implemented in manual mode, on an Analytical Plotter or a Digital Station. During the data acquisition, 3-D points belonging to a single object should be coded into two different types according to their functionality and structure: boundary points and interior points (see Figure 4). Boundary points must be measured in a particular order, either clockwise or counter-clockwise. Interior points can be measured in an arbitrary sequence. Since the human operator is responsible for the interpretation and measurement, it is possible to acquire any level of object detail for buildings, roads, waterways and other objects. With this technique, hundreds of objects can be measured in one day.

CC-Modeler is an automatic topology generator for 3-D objects. The main components of the system are shown in Figure 1. The first obligatory step is preprocessing, which includes the checking of the measurement order of the boundary points (BP), detection of redundant points, and determination of the possible groups of faces, based on sets of adjacent (BP) point pairs (compare section 3).

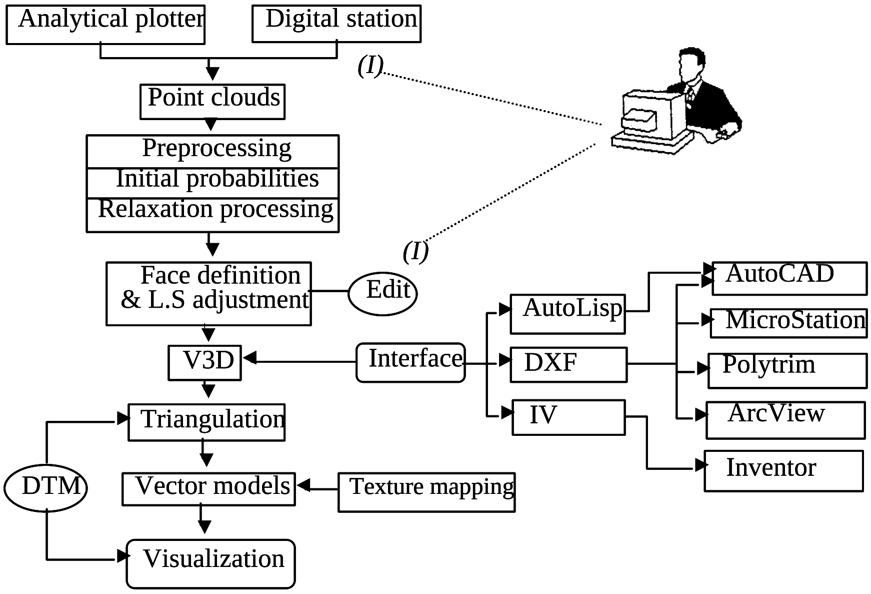


Figure 1: The data flow of CC-Modeler
(I) ---- Interactive functions

The next step is to build the face model of the 3-D object, i.e. to determine how many faces the 3-D object has, which points define an exact face and the spatial relations of the faces. This is implemented through a *Consistent Labeling* algorithm by probabilistic relaxation operations, in which two procedures are involved, the initial probability determination and the relaxation processing. The detailed algorithm will be presented in the following section. The result *of Consistent Labeling* is the face definition for every face. Then, least squares adjustment is performed for all faces simultaneously, fitting the individual faces in an optimal way to the measured points and considering the fact that individual points are usually members of more than one face. This adjustment is amended by observation equations that model orthogonality constraints of pairs of straight lines between boundary points. Finally, a vector description of 3-D objects is obtained, which is represented in a self-developed data structure (V3D). For the purpose of visualization, the CC-Modeler can also triangulate the faces to a TIN structure. Although the procedure is automated, human intervention and interaction with the automatic procedures is also available, as outlined in Figure 1.

V3D is a self-developed vector data structure, which builds the facet model of objects (Figure 2). The basic geometric element is the point. Points are used to express faces or line segments. An object consists of faces, and a polyline consists of line segments. Once some attributes are attached, an object or polyline becomes an entity class. Polyline representation is typically used to describe 1-D objects, like property boundaries, sidewalk borders, line features on roads, etc. This allows also to integrate one-dimensional features from scanned map data. In addition, CC-Modeler has the ability to map images onto a 3-D object or DTM in order to create a more natural scene. The texture, taken from the original images, is attached to an exact face as a special attribute. The DTM is considered a particular entity class in V3D. With the help of interfaces, one can conveniently translate the V3D data structure into different data files, such as DXF, IV, and AutoLisp, which are readable by AutoCAD, MicroStation, Polytrim, ArcView and Inventor.

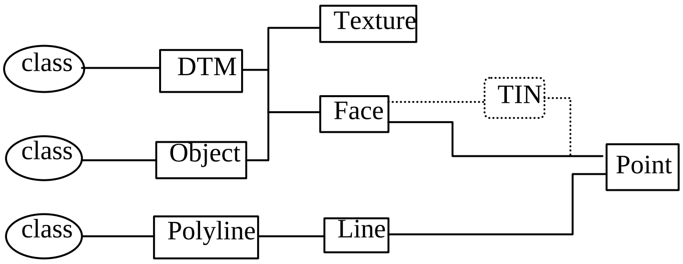
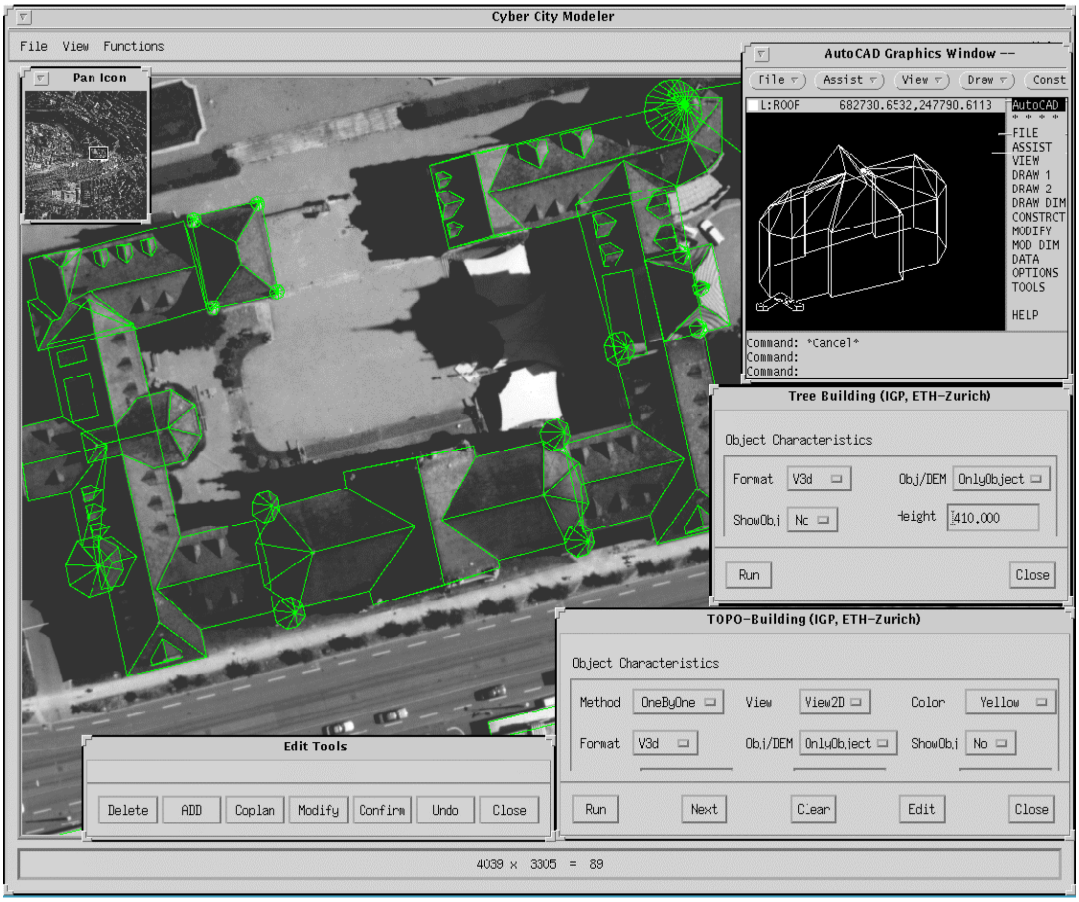


Figure 2: The data structure of V3D



L:ROOF 682730.6532,247790.6113

Command: *Cancel*
Command:
Command:

Format V3d
Obj/DEM OnlyObject
ShowObj No
Height 410.000

Method OneByOne
View View2D
Color Yellow
Format V3d
Obj/DEM OnlyObject
ShowObj No

4039 x 3305 = 89

Figure 3: An overview of the CC-Modeler userinterface

CC-Modeler has been successfully implemented on workstations (Sun SPARC) under X-Windows and OSF/Motif. Figure 3 is an overview of the CC-Modeler userinterface. One can optionally set CC-Modeler to work in the automatic or the interactive mode. When working in the interactive mode, one can edit or modify 3-D objects with the help of 2-D or 3-D views, if any mistakes occur due to inadequate point measurements. With the 2-D view, CC-Modeler will project the vector data of 3-D objects onto the original image (see Figure 3) to help check the results. In the 3-D mode, a 3-D vector model of the object will be shown to monitor the procedure.

3. STRUCTURING THE 3-D POINT DATA

It is assumed that a 3-D point cloud for each object of interest has been generated, e.g. by photogrammetric stereo model measurement. For building roofs the sequence of the points should be partly in an ordered fashion such that the boundary points of an object (P_1, ..., P_{10} in Figure 4) are to be measured either clockwise or counter-clockwise and labeled (BP). All other points (called "interior points”) can be measured in an arbitrary sequence and are labeled (IP).

The points of an object are expressed as the nodes of a graph, and each line is expressed as an edge with two nodes. The topology structure of this object is shown in Figure 4 together with the related graph. Obviously, every sub-circuit (such as P_1 P_2 P_{13} P_{12}) in the related graph is a basic face of the object, and every two adjacent boundary points (BP) can be always considered as a basic edge of a sub-circuit (such as P_1 P_2). Thus, our problem is to investigate how to construct every sub-circuit based on two adjacent points (BP) as its basic edge.

From a geometric point of view, every two adjacent points (BP) together with an interior point construct a face (such as P_1 P_2 with P_{13} in Figure 4). The combination of the adjacent points (BP) with different interior points will generate different faces. Vice versa, every interior point (IP) belongs to more than one face.

In theory, there are various labeling methods available, but only one solution is desired, which meets the inherent topological constraints of the object. From a geometrical point of view, the inherent topological constraints can be summarized as:

(1) A 3-D object is a closed multiple-plane object
(2) Planes are not supposed to pierce each other
(3) Every two adjacent boundary points are always part of a face.

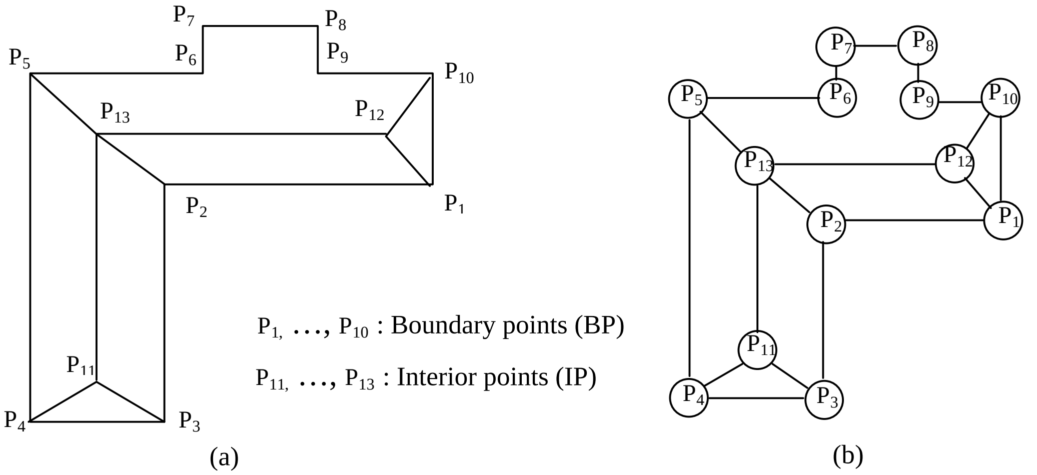

P7
P8
P5
P6
P9
P10
P13
P12
P2
P1
P11
P4
P3
P1, ..., P10 : Boundary points (BP)
P11, ..., P13 : Interior points (IP)
(a)
P7
P8
P5
P6
P9
P10
P13
P12
P2
P1
P11
P4
P3
(b)

Figure 4: (a) Point definition in CC-Modeler
(b) Related graph

Therefore, our problem is to determine for instance which interior points (P_{13}, P_{12} in Figure 4) lie on a face determined by an adjacent point pair (P_1 P_2), and what link order those points take to construct the face ($P_1 \rightarrow P_2 \rightarrow P_{13} \rightarrow P_{12}$).

All 3-D points, expressed as a set *H*, can be decomposed into two subsets, i.e. the adjacent point (BP) pair set *C* and the interior points (IP) set *A*. According to graph theory, if a set of nodes of a graph *G* can be divided into two non-empty subsets *A* and *C*, such that an edge of *G* connects with a node of *A* and a node of *C*, this graph is called a *Bipartite Graph*. Our problem of generating the topology of a 3-D object is equivalent to the determination of the spatial relation between the elements in the set *H*, particularly that between the set *A* and the set *C*. Therefore, we define our problem as a *Bipartite Graph Matching* (Wilson, 1979) or a one-to-multiple *Consistent Labeling* problem (Haralick, Shapiro, 1979), that is , to assign every interior point A_i to the right faces, based on the inherent geometric constraints mentioned above.

Figure 5 shows the basic principle of *Consistent Labeling,* in which C_j *(j=1,…, m)* are BP pairs and A_i *(i=1,…, n)* are the interior points. Each link between A_i and C_j is the result of labeling. For more detail about *Consistent Labeling* we refer to Haralick, Shapiro, 1979.

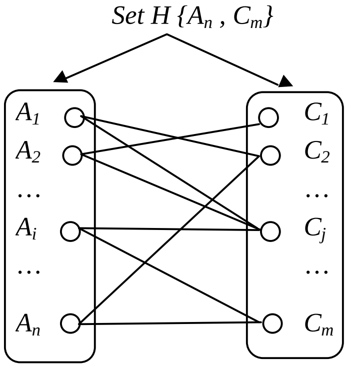


Figure 5: The principle of *Consistent Labeling*

For the solution of *Consistent Labeling* many methods have been proposed (Rosenfeld et al., 1976, Li, Wang, 1996) for different applications. With CC-Modeler, we developed a labeling algorithm based on *probabilistic relaxation*. The result of labeling A_i Î C_j is valued by probability, i.e. the probability of labeling each interior point as belonging to a particular face. The labeling probability is modified according to the observations and the geometric constraints existing between A_i and C_j. For more detail about probabilistic relaxation, we refer to Rosenfeld et al., 1976.

The classical probabilistic relaxation principle can be expressed as: An objective set *A {A_1, A_2, …, A_n}* will be labeled into *m* classes, and the class set is expressed as *C {C_1, C_2, …, C_m}*. The labeling procedure is interrelated. For example, labeling A_i Î C_j will affect the result of labeling A_h Î C_k. The interrelation is defined as a cooperative coefficient *l (i, j; (h, k))*. Assuming that p_{ij} is the probability of A_i Î C_j, and $\overline{A_i}$ is the adjacent area of A_i, then the standard relaxation approach is expressed as:

$$p_{ij}^{n+1} = p_{ij}^{n} \frac{(1+q_{ij}^{n})}{norm_i^{n+1}} \qquad (1a)$$

$$norm_i^{n+1} = \sum_{j=1}^{m} p_{ij}^{n}(1+q_{ij}^{n}) \qquad (1b)$$

$$q_{ij}^{n} = \sum_{h \in \overline{A_i}} \sum_{k=1}^{m} l\,(i,j;(h,k))p_{hk}^{n} \qquad (1c)$$

The traditional relaxation algorithm must be modified to work in our case, because our labeling procedure is not a one-to-one correspondence, but that of one-to-multiple matching. This means that an element in the objective set *A* will be labeled into more than one class sets.

The formula (1) is a recursive equation. p_{ij}^{n} is the probability of labeling *i* to *j* in the *n*th recursion, and $0 \le p_{ij}^{n} \le 1$ is always valid. The variable q_{ij}^{n} can be considered as the added magnitude of the probability labeling *i* to *j* in the next recursion step, and $q_{ij}^{n} \ge 0$. In fact, two types of possibilities always exist, i.e. labeling A_i Î C_j and A_i Ï C_j. Assume that $\overline{p}_{ij}$ is the probability of labeling A_i Ï C_j, and $\overline{q}_{ij}$ is the added magnitude of $\overline{p}_{ij}$, such that:

$$\overline{p}_{ij}^{n} = 1 - p_{ij}^{n} \qquad (2a)$$

$$\overline{q}_{ij}^{n} = \sum_{h \in \overline{A}_i} \sum_{k=1}^{m} l\left(i, j; (h,k)\right) \overline{p}_{hk}^{n} \qquad (2b)$$

Thus, the modified relaxation algorithm is expressed as:

$$p_{ij}^{n+1} = p_{ij}^{n} \frac{(1+q_{ij}^{n})}{norm_i^{n+1}} \qquad (2c)$$

$$norm_i^{n+1} = p_{ij}^{n}(1+q_{ij}^{n}) + \overline{p}_{ij}^{n}(1+\overline{q}_{ij}^{n}) \qquad (2d)$$

p_{ij}^{n} expresses the probability that event A_i Î C_j is 1, i.e. the probability that the objective element A_i belongs to the class C_j. q_{ij}^{n} takes the same form as formula (1c). $\overline{A_i}$, the adjacent area of A_i , can spread over the full set A. $l(i, j; (h, k))$ is the cooperative coefficient. Generally, in the event of A_i Î C_j and A_h Î C_k being fully cooperative, $l(i, j; (h, k)) = 1$. If the event A_i Î C_j is not related to the event A_h Î C_k , $l(i, j; (h, k)) = 0$. For our problem, the following function is employed to compute $l(i, j; (h, k))$:

$$l\left(i, j; (h,k)\right) = \begin{cases} \cos 2a & for(j = k) \\ 0 & for(j \neq k) \end{cases} \qquad (3)$$

α is the internal angle between the normal vectors constructed by the faces A_i Î C_j and A_h Î C_k, which is obtained by computing the scalar product of both normal vectors. Notice that to compute $l(i, j; (h, k))$ one should follow a basic criterion, i.e. the result of labeling A_i (A_h) to C_j (C_k) should lead to a graph in which no intersection between two circuits exists.

It should be noted that the determination of the initial probability for p_{ij}^{0} is very important. Good initial probabilities cannot only accelerate the iterative procedure, but also improve the reliability of labeling results. The initial probability of A_i Î C_j can be determined according to the spatial distance from A_i to C_j. The following formula is employed:

$$p_{ij}^{0} = b_1 + b_2 \frac{Max(d) - d(i \Rightarrow j)}{Max(d) - Min(d)} \qquad (4)$$

Max(d) is the longest distance between element C_j and every point in the adjacent area of A_i and *Min(d)* is the shortest one. *d(iÞj)* is the distance from A_i to C_j. $\beta_1 = 0.1$ and $\beta_2 = 0.8$ are constants which are determined empirically.

The procedure of formulas (1) and (2) is iterative. Finally, the labeling results of every point A_i are determined according to the probability p_{ij} . Thus, all elements (i.e. points) of each face are obtained. To construct the final face, the elements in each face are ordered by a spatial search procedure. Figure 6 shows the procedure, in which it is assumed that P_1P_2 is the basic link of a face, P_i (i = 3, 4, ..., 7) are the labeled elements for the face. The first approach is to calculate the center of the point group, P_c. If P_cP_2 is considered as the base, all internal angles that the vectors P_cP_i (i = 3, 4, ..., 7) generate in relation to the base P_cP_2 can be obtained (see Figure 6(b)). Obviously, ordering the point group is equivalent to ordering these internal angles, which is a simple procedure. Thus, the final ordering is shown in Figure 6(c).

We have obtained the face definition with a (BP) pair as its basic edge. In some particular situations, a roof unit may have some faces that are constructed by (IP) points. Therefore, CC-Modeler links all interior points of all faces to generate a subgraph, and then checks it. If a sub-loop exists, a new face is defined.

In the follow-up step all planar faces are simultaneously fit to their related 3-D point observations by a joint least squares adjustment. These observation equations are amended by observation equations which model the orthogonality constraints of pairs of straight lines to ensure that measurement errors do not lead to a violation of building construction rules. The complete adjustment is performed iteratively in the following manner:

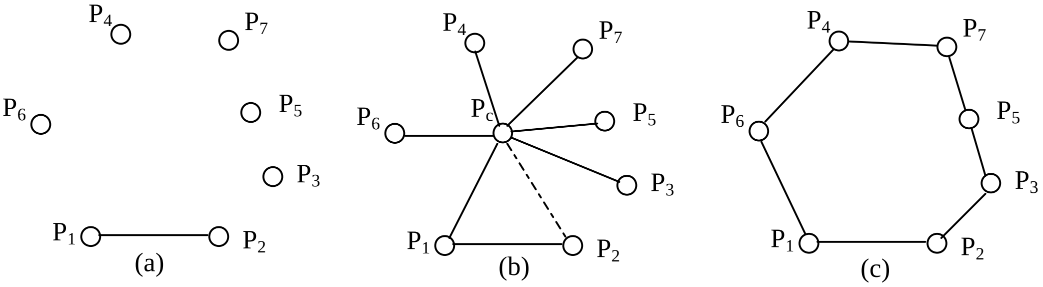


Figure 6: Principle of ordering the elements of a face
(a) Original points (b) Centering of points (c) Ordering result

(1) Individual plane adjustment

Assuming that the face F_j is composed of k points (including interior points and boundary points), the adjusted observations of these points should fit an exact planar face function. Each point i gives an observation equation of the form:

$$v_i = A_j x_i + B_j y_i + C_j z_i + D_j \quad ; p_i \qquad (5)$$

p_i ...... weight for plane condition

A_j , B_j , C_j , D_j ...... estimated parameters.

(2) Projection of measured points onto all planes under consideration of orthogonality constraints

In a second step we formulate for all faces F_j and all points (i) the following observation equations:

$$v_i = A_j dx_i + B_j dy_i + C_j dz_i + (A_j x_i + B_j y_i + C_j z_i + D_j) \quad ; p_i \quad i \in k \qquad (6)$$

The unknown parameters dx_i ,dy_i ,dz_i represent changes to the original point locations such that the adjusted point location is optimal with respect to the fitted planes.

For m faces, these observation equations are set up simultaneously.

The geometric orthogonality constraints of pairs of straight lines are involved as additional observation equations. A tolerance parameter (usually ±6°) for the deviations from orthogonality is selected. This procedure is performed in 2-D in the x-y plane.

Assuming that the angle between the straight lines l_1 $((x_{j-1}, y_{j-1}), (x_j , y_j))$ and l_2 $((x_j , y_j), (x_{j+1}, y_{j+1}))$, is less than the tolerance allows (see Figure 7), the following equation is formulated:

$$v_j = (\Delta x_{j+1,j} - \Delta x_{j,j-1}) dx_j + (\Delta y_{j+1,j} - \Delta y_{j,j-1}) dy_j + \Delta y_{j+1,j} \Delta y_{j,j-1} + \Delta x_{j+1,j} \Delta x_{j,j-1} \quad ; p_j \qquad (7)$$

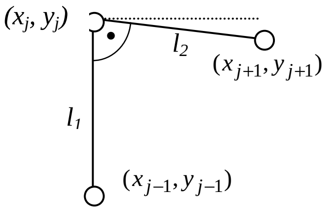


Figure 7: Geometric orthogonality in 2-D (x, y)

Equations (6) and (7) are solved simultaneously. Step (1) and (2) are performed in an iterative manner. After the solution of (6), (7) the system (5) is solved again with improved point coordinates and a new solution of (6), (7) is computed.

In a follow-up step, every face can be triangulated for the purpose of visualization. Here an algorithm similar to Delauney triangulation is employed.

The photogrammetric measurement principle allows for a free choice of the object resolution, accuracy and fidelity. The CC-Modeler generates a planar world, in which curved surfaces can be approximated by a set of planar patches. Special objects can be generated and inserted. We have demonstrated this with trees (compare Figure 8), waterways and some houses with curved-shaped roofs. CC-Modeler has been successfully applied to several data sets. The results are overall positive.

4. RESULTS OF SEVERAL PROJECTS AND VISUALIZATION

CC-Modeler has been tested in several pilot projects, from which the statistics of two Zürich data sets are presented in Table 1. "Structured automatically" refers to the number of roof units that CC-Modeler builds successfully with full automatic processing, and "structured interactively" refers to the number of roof units that needed to be manually modified in some faces. Obviously, the success rate of CC-Modeler's automatic processing is better than 95%, and almost all roof units can be constructed by using the convenient editing tools. The main factor determining the performance is the degree of familiarity of the human operator with the concept of automated reconstruction. Investigating the procedure of CC-Modeler processing, we find that the reason of CC-Modeler failing to process one object is that the measured point cloud was incorrectly coded, which results in the fact that the labeling results do not meet the topological constraint conditions mentioned above. In addition, CC-Modeler may result in some inadequate face definitions for some objects (about 4% in Tab.1), which need interactive modification with the help of editing tools. The main reasons for that are measurement errors and ambiguities in topological relations (under certain conditions there may exist more than one valid solution).

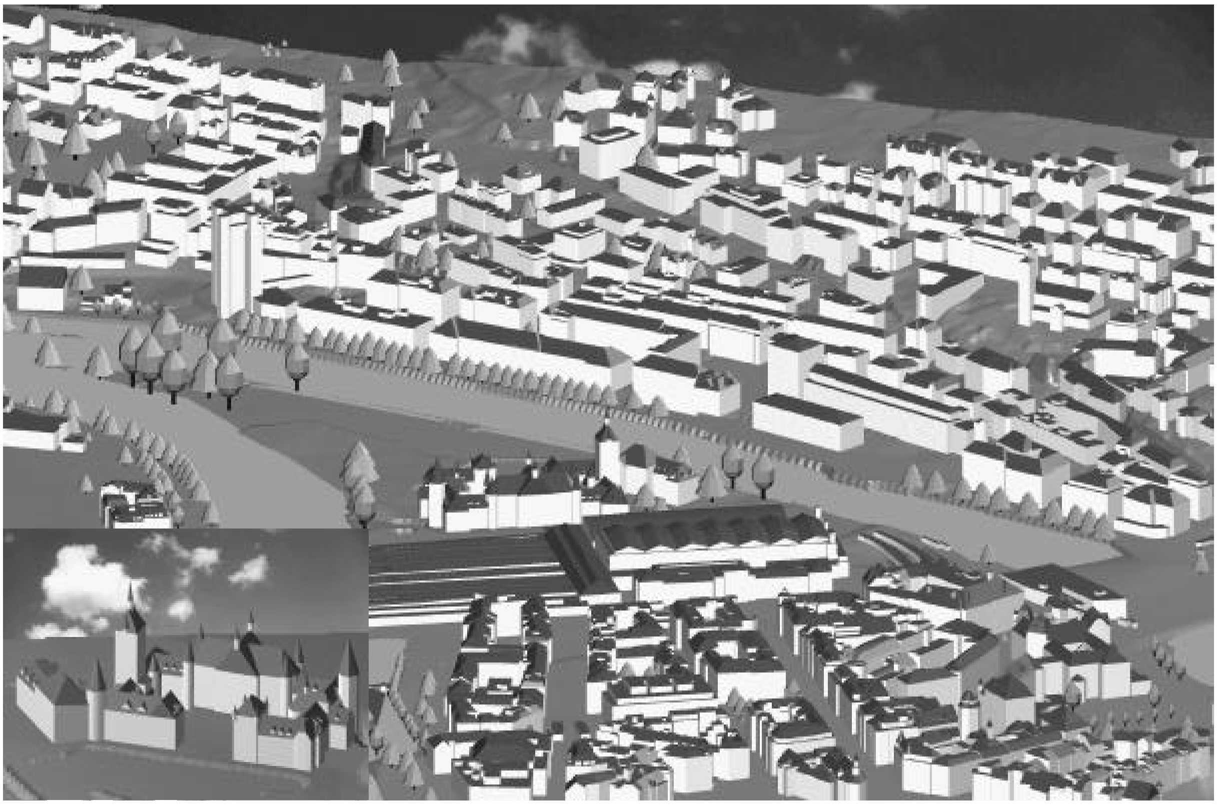


Figure 8: City model “Zürich Hauptbahnhof”

Table 1: CC-Modeler statistics of Zürich projects

Project	Total no. of roof units	Structured automatically	Structured interactively	Failures	CPU time (sec)
Zürich Hauptbahnhof	1733	1656	76	1	547
Zürich Bahnhofstrasse	649	629	20	0	128
Total	2382	2285	96	1	675

With a person well familiar with CC-Modeler, over 500 roof units can be generated per day. The CPU times (4 roof units per second, based on Sun SPARC platforms) indicate that the procedure of structuring can be executed on-line, while the operator is measuring , and in real-time.
For the visualization and animation of the data sets we use various software: AutoCAD, MicroStation, Inventor, and Polytrim. Figure 8 shows a view of the city model " Zürich Hauptbahnhof" created with MicroStation, including buildings, rivers, trees and DTM. For photorealistic rendering we combine the vector data of the buildings and the DTM with image raster data. The raster images are taken from aerial images. Figure 9 shows the result of mapping image data onto the DTM and some roofs, created with Inventor. Figure 10 shows " Zürich Bahnhofstrasse", generated with MicroStation. CC-Modeler can also map digital images, taken with still video cameras, onto 3-D faces, such as the walls of buildings.

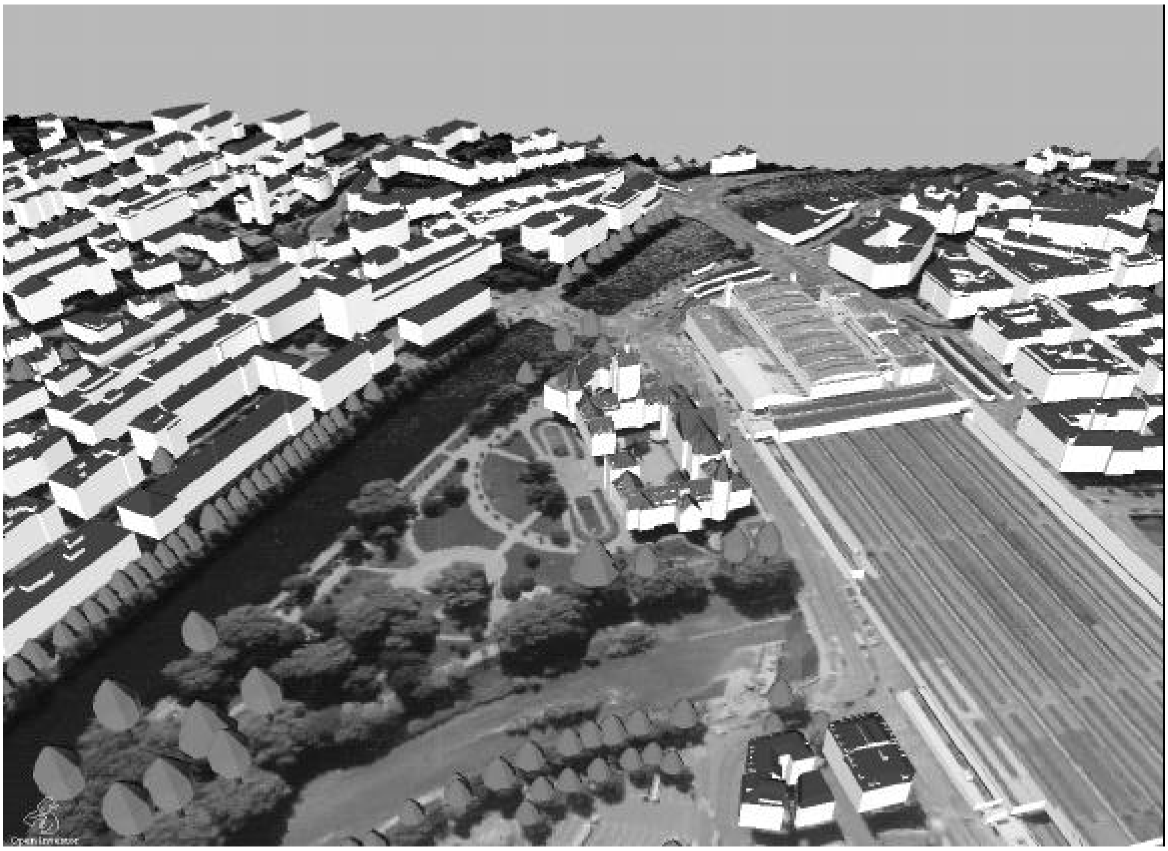


Figure 9 : “Zürich Hauptbahnhof” with mapped texture

5. CONCLUSIONS

CC-Modeler is a powerful data acquisition tool for the generation of 3-D city models. It has proved to be superior to our previously used system TOBAGO. Our experiments show, that it is flexible, robust and accurate. In four pilot projects we have achieved a success rate of better than 95% percent in fully automated structuring. Remaining problems are indicated and can be solved interactively. We have developed our own data structure V3D with interfaces to a variety of CAD and visualization packages. CC-Modeler cannot only reconstruct multiple kinds of 3-D and 2-D objects such as buildings, waterways, roads, trees, DTM, etc., but also map images onto these objects. This can be combined with data from general land use, communication systems, utilities, property and administrative boundaries, etc. to generate a complete 3-D city model.

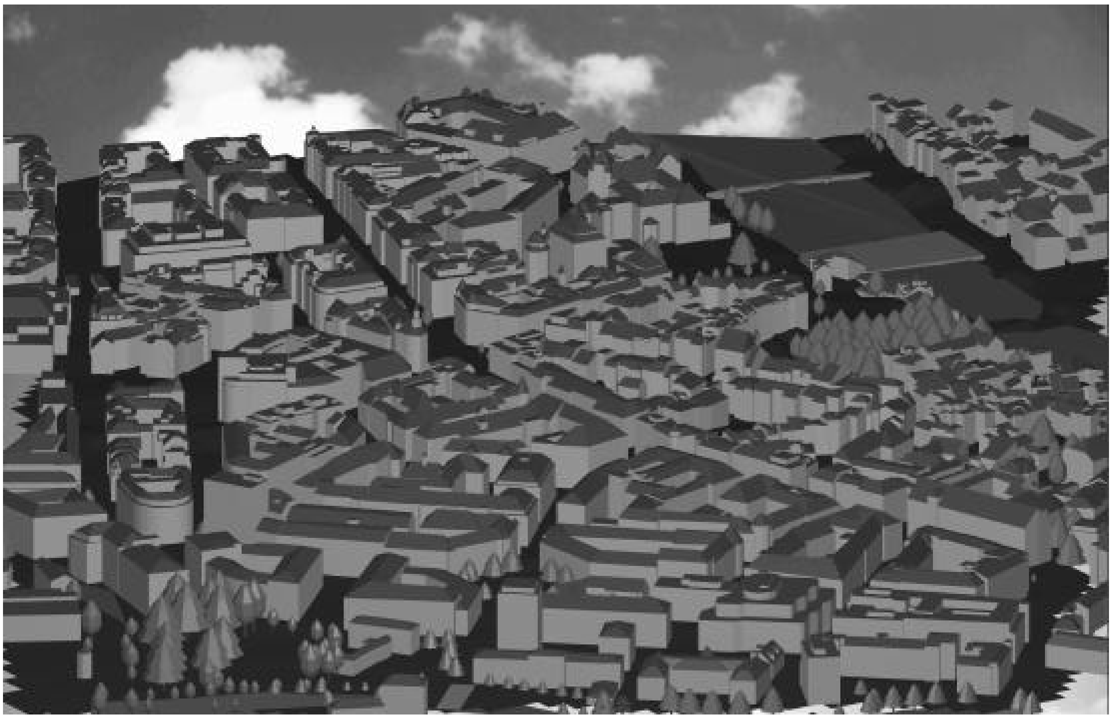


Figure 10: Data set of Zürich Bahnhofstrasse

If required, the data generated may be operated by a data base management system to form a Spatial Information System (SIS) (Wang, Gruen, 1998). This in turn can be integrated into a multimedia environment for the purpose of better user interaction.

REFERENCES

Dan, H., 1996. Rekonstruktion generischer Gebäudemodelle aus Punktwolken und deren Abbildungskorrekturen in Orthobildern. Dissertation Nr. 11759, ETH Zürich, Mitteilungen Nr.60 des Instituts für Geodäsie und Photogrammetrie, 93 pages.

Gruen, A., 1998. TOBAGO - a semi-automated approach for the generation of 3-D building models. ISPRS J. Photogrammetry & Remote Sensing, Vol.53, No.2, pp. 108-118.

Gruen, A., Baltsavias, E., Henricsson, O., (eds.), 1997. Automated extraction of man-made objects from aerial and space images(II). Proceedings of the Monte Verita Workshop, May 1997, Birkhauser Verlag, Basel.

Haralick, R. M., Shapiro, L. G., 1979. The consistent labeling problem: Part I. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.1, No.2, pp. 173-184.

Li, D., Wang, X., 1996. Relational structure description and matching algorithm for 3D objects. International Archives of Photogrammetry and Remote Sensing, Vol. XXXI, Part B3, pp. 442-448.

Rosenfeld, A., Hummel, R. A., Zucker, S. W., 1976. Scene labeling by relaxation operations. IEEE Transactions on System, Man, and Cybernetics, Vol.6, No.6, pp. 420-433.

Wang, X., Gruen, A., 1998, “A 3-D city Model data structure and its implementation in a relational database”, paper submitted to International Conference of Spatial Information Science and Technology, 13-16 December, Wuhan, China.

Wilson, R. J., 1979. Introduction to Graph Theory. Academic Press, New York, pp. 115-118.