BUILDING RECONSTRUCTION FROM AERIAL IMAGES AND CREATION OF 3D TOPOLOGIC DATA STRUCTURE

S. Zlatanova**, M. Pilouk*, K. Tempfli**

* ESRI, USA
*email: mpilouk@esri.com
** International Institute for Aerospace Survey and Earth Sciences (ITC),
Hengelosestraat 99, 7514 AE Enschede, the Netherlands
**email: nedkova@itc.nl, tempfli@itc.nl

Published in: Proceedings of IAPR / TC-7 Workshop, 2-3 September 1996, Graz, Austria

Abstract

The interest in 3D GIS increases due to developments in technology and necessity of users to cope with more complex tasks. Questions related to data acquisition, data storage, data analysis and data visualization are topics for investigations. A key issue on the way of building a 3D model, is an adequate data structure capable to respond to various user queries. The paper elaborates on problems related to 3D model construction for buildings from aerial images, using manual stereo digitizing. The approach relies on a data model based on the 3D Formal Data Structure, which supports 3D topology. The emphasis of the presented work is on an automatic data structuring. The outcomes from the implementation work show that topology facilitates the process of both data acquisition and data visualization.

1 Introduction

1.1 3D-GIS

A demand for 3D geo-referenced information and systems supplying 3D analysis and visualization appears in various areas of human life. The need for 3D spatial information is quite urgent in urban areas. A typical example is a municipal information system where the tendency to increase the range and complexity of municipalities' tasks leads to the necessity of means for three dimensional spatial analysis.

Existing 2D GIS and 3D CAD systems cannot respond completely to the new requirements. 3D CAD products, used widely in architectural planning, for training purposes, etc., offer tools for 3D visualization and links to DBMS for semantic analysis but they do not support 3D spatial analysis. 2D GISs maintain only 2D topology which offers limited capabilities of spatial analysis. Vendors have worked out various combinations of tools coming from these systems in order to improve the capabilities of the systems. Although there are some good tools, there is still not a commercial system providing both a 3D description of spatial relationships and appropriate visualization.

The demand for spatial information and unsatisfactory means offered by commercial products, have led to studying 3D GIS capable to cope with geometry, topology and semantics. Several groups work on this issue (see [3],[8],[9],[13]). The results from theoretical and implementation work on a new 3D data structure carried out at ITC, are very positive (see [1], [2], [10]). Since the developed 3D data models and data structures are in a conceptual stage, their applicability for various problems in urban areas, still needs exploration. The present paper is devoted to one of the aspect of 3D GIS, i.e., data collection from aerial images and creation of a 3D data structure.

1.2 3D Formal Vector Data Structure (3DFDS)

A 3D data model, based on the Molenaar's Formal Data Structure (FDS) (see [1], [8]) is chosen and studied due to its suitability for the purpose of 3D GIS. The 3D-FDS is a comprehensive vector model, comprising geometric and semantic information and maintaining 3D topology. The data about all the properties is stored and handled together in one data base. The vector method for describing 3D geometry is consider to be more appropriate for urban areas than the raster presentation (see [1], [7], [12]). Regular shapes of the buildings, easy way for texturing and visualization, less data for storage, etc. are some of characteristics which favour the vector method of description. Topology provides a better description of spatial relationships in terms of adjacency. The model can be implemented applying an object-oriented approach, which facilitates the visualization process (see [4]).

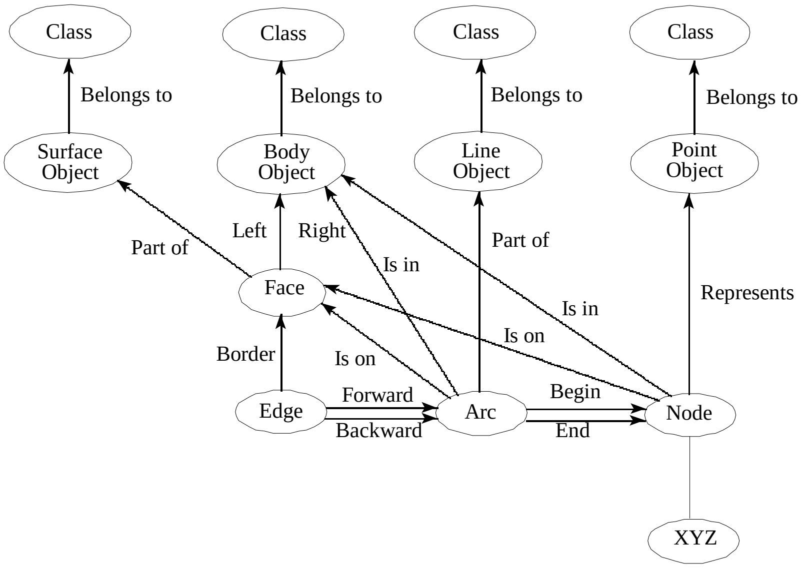

Figure 1: 3D Formal Data Structure

The information about objects in 3D FDS (see *Figure 1*) is structured into three major levels, i.e geometric, object and thematic levels. The lowest geometric level aims to represent the information about position, shape, size and relationships among the objects (topology). The introduced three primitives, i.e. *node*, *arc* and *face* are the structuring components utilized to compose objects. The type of the objects is defined on the second level. Four types of objects are supported by the 3D FDS model: *point*, *line*, *surface* and *body*. The last level constitutes thematic classes. The links like *is on*, *is in*, *part of*, etc., represent both links among the primitives forming the objects, and relationships among the features.

The primitives and links, described above, together with set of conventions, represents the basic idea of 3D FDS. The conventions impose strict rules for structuring the model. Detailed information about the 3D FDS and further extensions can be found in [8], [10], [14]. The data model serves various data queries and can be utilized for a large variety of applications (see [1]).

2 Technique for 3D Digitizing and 3D Reconstruction

Data collection is one of the most expensive processes in terms of time and manpower and, bearing in mind the large amount of data in 3D, a special attention should be paid to problems related to data acquisition and constructing the 3D model, e.g. methods for building the 3D model, combining data from various sources, techniques for speeding up the way of data acquisition, rules and algorithms for ensuring consistency of data, algorithms for automatic building of 3D topology, etc. The method presented in the paper considers the case of using photogrammetric equipment (analytical or digital) for data collection and standard, commercial software. The emphasis, however, is not on manual data acquisition but on the algorithms for building of 3D topology which are also relevant for semi-automatic feature extraction.

2.1 3D Digitizing

The process of feature extraction from aerial images comprises both natural and man-made objects. For model construction, however, it is more useful to distinguish surface objects and solid objects. Surface objects (parcels, roads, canal, etc.) can be described using only one z-coordinate for a given x,y-point. Solid objects (houses, buildings, bridges, etc.) with their vertical extension, introduce multiple z-coordinates. Natural objects (rivers, hills, lakes, etc.) and various man-made objects (roads, rails, canals) which form the terrain can be consider surface objects and therefore their extraction should be related to reconstructing the ground surface. In the following we shall concentrate on buildings, a subset of solid objects.

The data for constructing solid objects differs from data collected to produce 2D maps or DTM. Full 3D information for all the elements of the objects, e.g. roof elements, walls, footprints are necessary (see [6], [14]). The description of relationships in 3D FDS, the process of visualization, rendering and texturing, require every face to be reconstructed. This involves considerably more efforts during the process of data collection. The presented reconstruction is based on the assumption that every solid object can be described by a decomposition of its boundary, i.e, into its primitives *faces*, *arcs* and *nodes* (see *Figure 2*). Looking at the partitioning, assuming that the walls are vertical and always reach either the ground or another surface of the constructed model, it can be seen that the wall *faces* can be constructed using information from the roofs and terrain. Thus it is not necessary to digitize the footprints of the buildings. Complex roofs can be reconstructed by digitizing only edges of roofs like *arcs* (not closed polygons) and using them to define the corresponding *faces.*

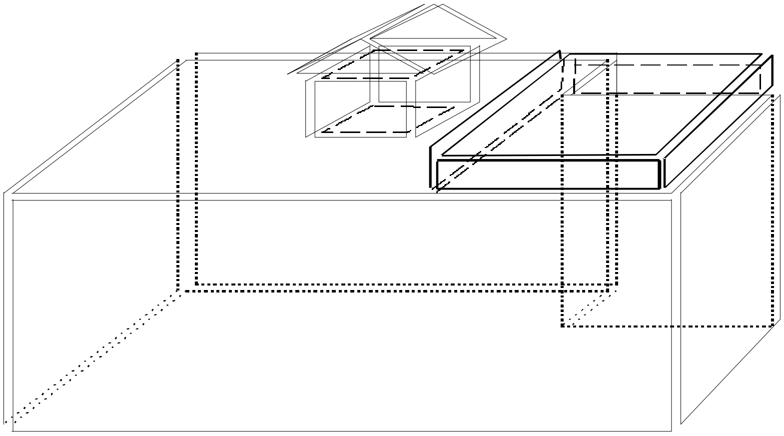

Figure 2: Decomposition of a solid object.

A coding system, aiming to identify the various features for data collection of simple buildings (vertical or plane walls, no overhanging roofs, no facade details), must be developed. Special attention must be paid to the categorization of the roofs (see[15]). The identified four groups of different shapes are in accordance with the subsequent reconstruction.

1. The first group contains all measured polygons that are roof outlines and are projected onto the ground in order to obtain footprints and thus to create the walls. These polygons constitute *faces* of the class roofs. This means that they are used to "close" the building, i.e. they are on the boundary between the "inside" and the "outside" space. The code number is B01 (see *Figure 3a*).

2. The second group comprises the polygons, which are utilized for both purposes: a) projection onto the ground and creation of the walls, and b) construction of the *faces* of hip

and gable roofs. The code number is B02 (see *Figure 3b*).

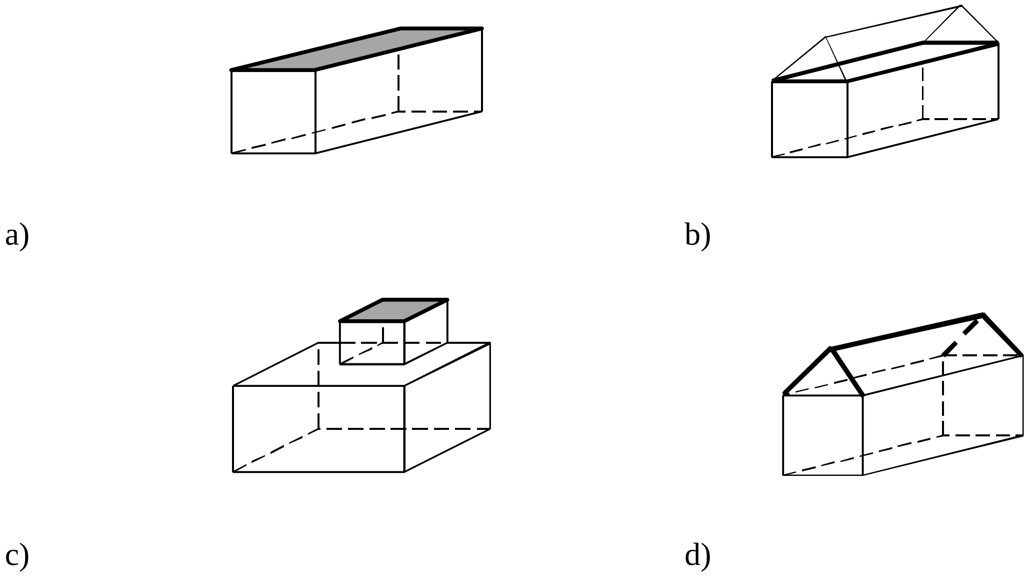

Figure 3: Groups of roof shapes: a) outlines of flat roofs, b) outlines of hip or gable roofs, c) outlines of flat roofs over another surface, d) ridges of roofs

3. The third group contains all roof polygons which are used for creating the walls of bodies which do not lie on the ground. These polygons form *faces* of flat roofs. The projection of such a polygon onto a surface of the building, e.g. another flat roof, is used to compute the bottom outlines of the body. The code number is B03 (see *Figure 3c*).

4. The fourth group includes all other measured edges of roofs which are needed to delineate roof *faces*. They are not used for computation of footprints or walls. The code number is B04 (see *Figure 3d*).

This coding system was used in an implementation test employing the Traster T10 and the software DEMETER (see [15]). Manual digitizing, using such a minimum set of building features is a quite easy and fast way of gathering data from images. Each group has to be digitized in separate layer. The polygons with code numbers 1, 2 and 3 must be closed, i.e. a snap mode has to be used, because they form the outlines of the buildings. The order of digitizing does not matter (clockwise or anti clockwise). The group of elements with code number 4 consists of open lines, i.e. edges from the roofs.

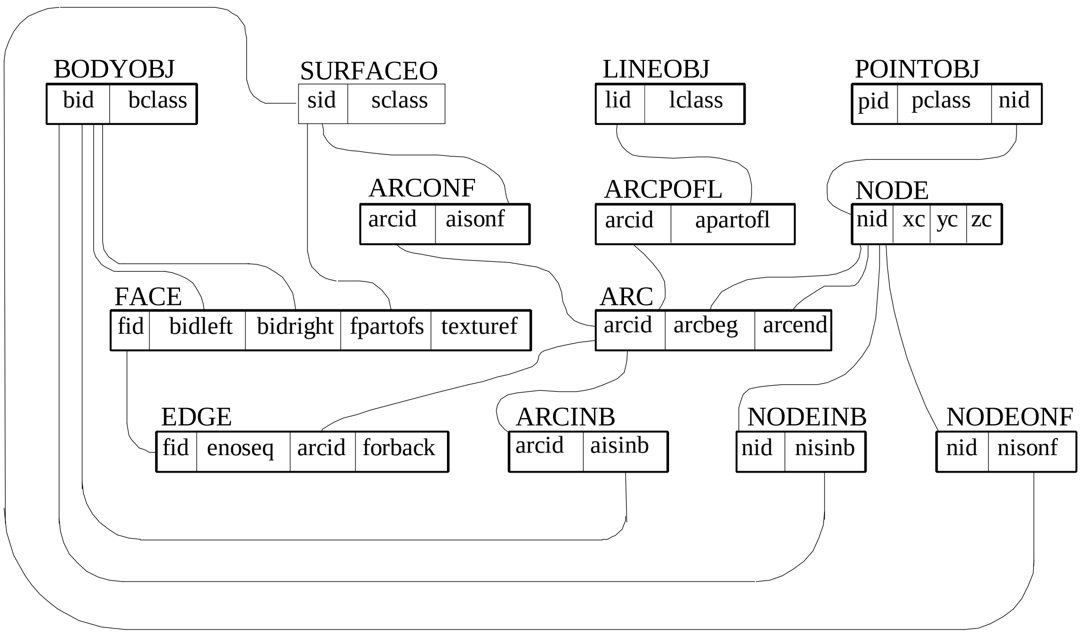

Figure 4: Relational data structure of 3D FDS

2.2 3D Reconstruction

The second part of the 3D model construction is based on software developed at ITC. Its goal is not only the automatic construction of a 3D model, but also the description of the relationships among the elements. For the purpose, the 3D FDS is translated to a relational data base structure (see *Figure 4*). An advantage of 3D FDS is its ability to be implemented into different data base structures. A relational structure was chosen to demonstrate spatial analysis with dBASE IV. Some results of investigations with object-oriented data bases can be found in [10].

Various algorithms were developed to process the digitized data. The challenge was to reconstruct houses using a minimum number of measurements. Efficient processing is attained by a) multiple use of most of the data sets coming from the measurements and b) utilization of the knowledge about the relationships created at the initial stages of reconstruction.

2.2.1 Processing the polygons necessary for the computation of the footprints.

The data set with code number B01 and B02 takes part in computing footprints, creating the walls and flat (B01) or complex (B02) roofs of the buildings. An initial check, important for creating 3D topology, i.e. eliminating multiple points with equal coordinates, is provided. The work of a case study showed that points with completely equal coordinates are quite common, originating from a longer holding of the pedal during digitizing. The polygons must be reordered into anti-clockwise direction in order to facilitate later rendering process.

Files corresponding to the tables ARC and NODE (see *Figure 4*), consisting of the information necessary for *nodes* (identifier, x,y,z -coordinates) and *arcs* (identifier, begin node, end node), are created at this stage. Since the roof outlines with code number B01 must form *faces*, an additional description about the *arcs* assembling them is stored in files PLG (identifier, number *arcs*, *arc1*, *arc2*,..) .

2.2.2 Creating a DTM

Surface reconstruction is obtained by creating a Triangular Irregular Network (TIN) from the measured elevation data. The surface and the roof outlines (codes B01 and B02) are used for the computation of footprints. The footprints are obtained by applying linear interpolation of z-coordinates on the surface for each x,y-point of roof outlines. The followed re-triangulation of the TIN, using the footprints as “constrains”, is the final step in processing the elevation data. The last procedure is compulsory for fulfilling some of the conventions of the 3D FDS (see [8]) and avoiding undesirable “gaps” in the 3D model. Files with description of *nodes* and *arcs* are created for both data sets B01 and B02. The relationships among the *arcs* of the *faces* (identifier, *arc1*, *arc2*, *arc3*) is in a temporary file which is used in the next steps of building the 3D topology.

2.2.3 Projecting the outlines of the roofs (code B03) onto another surfaces for computing the bottom outlines.

The data set digitized under code B03 has the specific function to construct walls which do not rest on the ground but on the surface, part of the building construction. The initial idea was to measure the height of the walls during digitizing (see *Figure 9*). The case study, however, showed that this is likely to lead to inconsistency in the model (see *Figure 10*). The bottom part of the body which must lie exactly on the surface, “sinks” below it. This result is caused by incomplete 3D topology. According to the conventions in 3D FDS, two *faces* must not touch. A “hole” has to be created in one of the *faces* (see [8]). A vertical projection from the points of this group onto a particular surface (e.g. roof) seems to be the better solution. These data have to be processed similarly to the data from the first and second group using another surface instead of the DTM. The resulting files from this step must be NODE, ARC and PLG files. The PLG file must contain all the *arcs* which constitute *faces* where a “hole” is created, taking into account the *arcs* of the “hole”.

a) b)

Figure 5: Complex roofs: a) combination of roof outlines code B02 and B03 b) couple roof outlines of code B03 one over other.

2.2.4 Creating the complex roofs.

The elements from the fourth group are used for constructing the *faces* of non-flat roofs. The handling of this data set still waits for its complete solutions. Part of the *faces* can be assembled merging data with code number B02 (files ARC and NODE). The missing *faces* can be obtained by further polygonization.

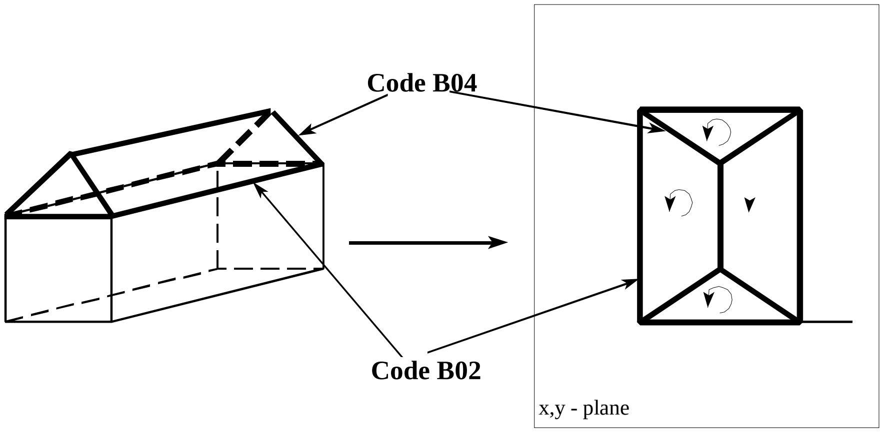

Figure 6: Polygonization of roof *faces*

The major idea of the polygonization software is to project all the data into the x,y-plane (see *Figure 6*) and construct the *faces* which are projected as closed polygons (applying a certain threshold). Since the approach eliminates the z-coordinate, some *faces* cannot be created automatically. The conclusion after the case study was that many roofs had to be processed manually to eliminate two main bottlenecks:

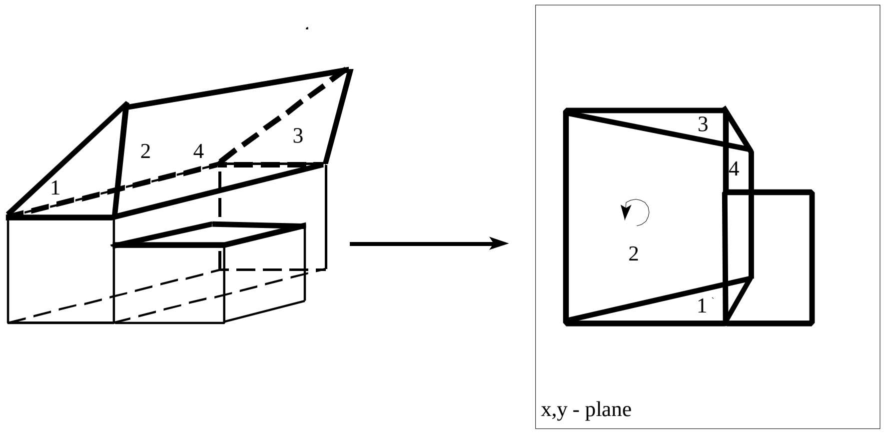

Figure 7: An example of kind of roofs, where the polygonization algorithm cannot be applied.

- Intersection of the polygons after projection, which was not caused by mistake, but due to the different height of the roofs, hanging or inclined roofs (see *Figure 7*). For the time being, the only a possible way to get over this obstacle was to remove some of the *arcs* by a manual deletion and create the corresponding *faces*.

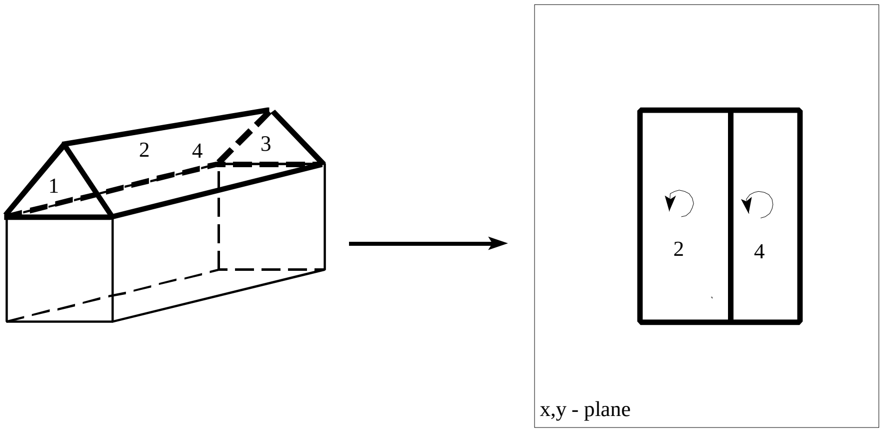

Figure 8: Gable roofs, where faces 1 and 2 cannot be assembled

- Special shape of roofs, in which some edges were projected onto themselves, and where one or two of the faces could not be constructed (see *Figure 8*). One way was some of the edges to be deleted, e.g. a common *face* to be constructed from the roof face 1 and the wall below it.. Another solution was to apply projection onto another plane, e.g. the y,z-plane.

Although the duration of the case study was not sufficient to solve these problems, some ideas were generated and their implementation is subject of further work.

2.2.5 Creating the walls

All the data necessary for forming the wall *faces* are already at our disposal at this stage, i.e. files ARC and NODE from data sets B01, B02 and B03, containing *arcs* of roof outlines and footprints or/and bottom outlines. The process of forming wall *faces* includes a) creation of new vertical *arcs* and b)proper orientation of the *faces*, i.e. the normal vector for every face must be directed outside of the building. The software takes care about the further building of the 3D topology. Files corresponding to the tables EDGE and FACE of relational data structure (see *Figure 4*) are created.

2.2.6 Merging different data set, i.e. constrained DTM, all the roofs and walls.

A number of files that must be merged in order to reconstruct the entire model, are obtained after all the steps with the various data sets. The process of merging is not a simple combination of all the data but a construction of all the relationships according to the relation data structure. The software saves the 3D data structure developed for every data set up to this stage (ARC, NODE, EDGE, FACE files), and controls the existence of the new primitives in such a way, that only new elements and relationships are added to the final model.

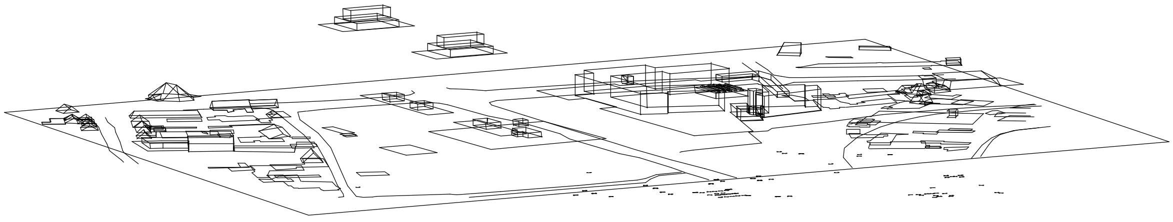

Figure 9: Digitized data set

3 Case study

The case study had to clarify whether the coding system was detailed enough to include all roofs appearing in reality, define the obligatory data control in conformity with requirements of 3D topology, and validate the way of processing of every data set.

The test area around the old ITC building was covered by one stereo model of M 1:2200. The 3D model was constructed using the procedure described above. The programming languages were C++ and Pascal. The hardware platform was PC. Since we have not developed specific software for visualization and texturing yet, AutoCAD and 3D Studio were used for final visualization. DBASE IV was used for storing the metric and semantic information and the description of relationships. The necessary elements of the buildings were digitized on T10 with DEMETER (see *Figure 9*). Although the T10 offers an automatic DTM generation, it was not applied for the ITC model, due to the flat terrain where several spot heights were sufficient. The outlines of the roofs and the other important features were measured in separate layers, according to the coding system. Some additional regulations related to the digitizing procedure are described in [15]. DXF file format was used as an output from DEMETER for further processing.

The information from the DXF file was loaded into AutoCAD and the data set was checked by visual inspection. Some mistakes arising after incorrect introduction of a code number; some complex roofs, e.g. data set with code number B03 might contain multilevel roofs which require their separation into several DXF files (see *Figure 5*), some roof edges which cannot be connected applying thresholds, etc. showed that the pre-processing work was inevitable. When the distribution of the data into different DXF files was completed, the further processing of data and construction of a 3D model could be automated. The data set with code number B03 required a lot of manual editing due to problems discussed in 2.2.4 and therefore was excluded from the model.

The case study established a possible production line for building a 3D model supporting 3D topology. The steps were:

- digitize roof features and surface elements (street edges, stop heights, etc.) using DEMETER and applying the coding system.
- visualization and pre-processing of the DXF file(s) in AutoCAD
- run model construction software (DXF to FDS file formats)
- run software for conversion of FDS file format to DXF format

- import DXF file with 3D model into packages for visualization (3DStudio, AutoCAD, etc.)
- generate VRLM file from the DXF file to use VR browsers to ‘walk through’

The case study has shown that 3D FDS offers fast and automatic way for building a 3D model. The manual method of digitizing allowed various roofs to be interpreted and categorized into different groups by the operator. The introduced coding system could be used to distinguish and decomposed complex objects. The model provided all the necessary data for visualization, rendering and texturing. The initial work on tools for visualization (see [10], [153]) producing graphics from dBASE files, showed that visualization of spatial analyses was possible. Additional experimentation work on data queries using dBASE IV, which is not discussed in this paper, showed good possibilities for spatial analysis for models which can be described using primitives *face*, *arc* and *node* (see [2]).

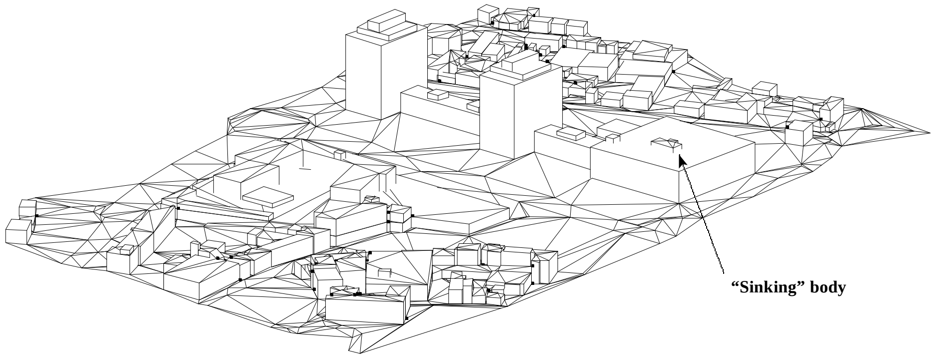

Figure 10: Wireframe of ITC model

4 Conclusions and Further Development

The approach used for reconstructing buildings is one possible manner for building 3D models using manual 3D digitizing from aerial images. In this context, it is one successful attempt to reduce manual measurements. Furthermore, this approach establishes an appropriate and efficient way of digitizing for creating 3D topology. Although encouraging results, there are still many tasks which remain for exploration and further development. For example:

- The coding system for digitizing should be more precise. For example, the measurements with code number B03 were additionally separated into two new groups for the case study. The required differentiation depends on the complexity of the scene and the required level of detail.

- Elaborating the coding system for colouring and rendering purposes. The software developed so far does not provide possibilities for colour and texture coding.

- Developing the algorithms for processing complex roofs. Existing software does not offer solutions for each type of roofs.

- Defining the procedures for controlling data consistency during both data collection data structuring.

The aim of the prototype software was to prove the concept for constructing a 3D model supporting topology. The developed programs are dependant to a certain degree, on the used photogrammetric equipment and commercial software. Since the digitizing (DEMETER) and visualizing software (AutoCAD, 3DStudio) supports DXF file format, software for conversion from DXF to 3D FDS format and vice-versa was developed. However, the diversity of the data import and export formats considering 3D FDS can be enlarged.

The investigations on 3D modelling using 3D FDS are still at an infant stage. Various aspects of the entire process of reconstruction, maintaining, utilizing and validating the model has to be studied and explored (see [14]). Conversion algorithms from other 3D representations into 3D FDS are of interest in order to create 3D topology from existing 3D CAD models. Methods for incorporation of data from various sources, e.g. 2D GIS, DTM, digitized maps, photogrammetric and survey measurements, etc. have to be investigated. Tools for 3D operations like interactive 3D editing with realistic visualization, 3D overlay for visual comparison with the original image, etc. are still lacking. Rules and methods for data updating in order to ensure data consistency and handle uncertainty of the data are another important issue. Algorithms for 3D visualization of the results from various spatial analysis appear necessary to be developed.

References

[1] Bric, V. "3D Vector Data Structure and Modelling of Simple Objects in GIS", MSc-thesis, ITC, 1994

[2] Bric V., Pilouk M. and Tempfli K. "Towards 3D GIS: Experimenting With a Vector Data Structure", in: ISPRS, Athens, 1994

[3] Cambray, B. " 3D Modelling in Geographic Database", in: AutoCarto, 11th International Conference on Computer Assisted Cartography, Minneapolis, 1993

[4] Kofler, M., H. Rehatschek and M. Gruber, "A Database for a 3D GIS for Urban Environments Supporting Photo-Realistic Visualization", in: ISPRS, Commission III, Vienna, Austria, 1996

[5] Kufoniyi, O. "Spatial Coincidence Modelling Automated Database Updating and Data consistency in vector GIS", PhD thesis, ITC, 1995

[6] Leberl F., M Gruber, P. Uray and F. Madritsch, "Trade-offs in the Reconstruction and Rendering of 3-D objects", Mustererkennung'94, Wien 1994

[7] Li, R. and Chen, Y. "Approaching 3D Data Structures in Geographic Information Systems", 1994

[8] Molenaar, M. "A Topology for 3D Vector Maps", in: ITC Journal, 1992

[9] Pigot, S. " Topological Model for 3D Spatial Information Systems", in: Technical papers, ACSM-ASPRS, Baltimore, 1991

[10] Pilouk, M. "Integrated Modelling for 3D GIS", PhD Dissertation, ITC, 1996

[11] Pular, D. and M. Egenhofer, "Toward Formal Definition of Topological relationships Among Spatial Objects"", in: Processing of the Third international Symposium on Spatial Data Handling, Sydney, 1988

[12] Shibasaki, R. and E. Shimizu, "Three Dimensional (3D) Digital Map for An Urban Area", in: ISPRS, Proceedings of the Symposium on Cartographic and Data Base Applications of Photogrammetry and Remote Sensing, Japan, 1990

[13] Tempfli, K. and M. Pilouk, "Vector Data Models for 3D-GIS", in: Proceedings of Third International Colloquium of LIESMARS on Integration, Automation and Intelligence in Photogrammetry, Remote Sensing and GIS, Wuhan, China, 1994

[14] Tempfli, K and M. Pilouk, "Practical Photogrammetry for 3D GIS", in: XVIII ISPRS, Commission III/IV, Vienna, 1996

[15] Wang, Z. "Digital Photogrammetric Data Acquisition for 3D GIS", MSc-thesis, ITC, 1994