THE FUTURE OF 3D GEO-INFORMATION MANAGEMENT

Siyka Zlatanova
Delft University of Technology, Thijsseweg 11, 2629JA, Delft, The Netherlands,
s.zlatanova@citg.tudelft.nl

Keywords: GIS, DBMS, CAD, topology, data management, visualisation, SQL

Abstract

Increasing number of applications need more advanced tools for representing and analysing the 3D world. Currently, a variety of software is already capable of handling 3D geo-information, ranging from data collection, data organisation, spatial analysis and visualisation. Among all types of systems dealing with spatial information, GIS has proven to be the most sophisticated system that operates with geometric and semantic properties of spatial objects, spatial relationships and provide means to analyse them. However, what is the future of 3D geo-management? It is the aim of this paper to analyse some new trends in organisation and management of 3D data. Some of the most important results reported by the researchers at GISt, TUDelft related to 3D modelling, editing and visualisation (based on Oracle and MicroStation) are discussed in detail.

1 Introduction

Geo-information has already proven its importance for many applications and daily use. A large number of human activities (government, industry, cadastre and land management, emergency services, tourism, traffic navigation, etc.) utilise 2D geo-data in some form (paper or digital maps) to complete different tasks. However, the world we are living in is three-dimensional and in many cases the two dimensions are not sufficient. The 3D objects presented as 2D projections may loose some of their properties and relations to other objects and may create difficulties to understand, analyse and evaluate the surrounding world in a critical for a certain activity moment. Increasing number of applications already seek for tools to model, store, analyse and visualise 3D data in an efficient and effective way. Urban (Zlatanova and Bandrova, 1998) and landscape planning, telecommunications, real estate market, 3D cadastre (Stoter, 2002), road, railway and building construction, utility management, shopping and tourism are among the most demanding ones. Moreover, maintenance, processing and visualisation of large data sets has been improving progressively in the last decade to approach the current stage when the user can immerse with a 3D model in different levels of mixture between reality and virtuality. Considering the recognised need for a

3D data and the high level of technology developments, the “logical” expectation is a variety of software dealing with the third dimension. Unfortunately, the current status of the 3D software market differs. Many vendors develop intensively extensions to their software for more effectively handling 3D data. However, the “killing” software product, capable of dealing with all kinds of 3D data and providing the functionality needed by different application is still wish. The difficulties in devising such a product arise at each stage of the 3D modelling. This paper concentrates on a limited number of those aspects, i.e. structuring and analysis of 3D data, that are closely related to 3D modelling. The paper is organised in three parts. The first part presents basic terminology and notations regarding 3D modelling. The second part concentrates on the last developments in spatial data handling achieved under the OpenGIS specifications. Several experiments organised in two case studies demonstrate the current functionality in 3D data maintenance. Final discussion summarises the current status and trends and concludes on the future of 3D GIS.

2 3D modelling

Model is a very general term to represent certain phenomena in a way readable for others. Geo-science specialists are busy with the modelling of real phenomena. A universal model to comprise all the aspects of reality is not practically realisable due to the high complexity of the real world. Different disciplines emphasise different aspects and only these aspects are included in the model. Thus a model considered good for the description of particular phenomena might be hardly appropriate for others. Different aspects and characteristics of real objects may lead also to the existence of different object definition. Furthermore, the methods to represent 3D spatial objects (e.g. vector, raster) differs significantly that also has impact on the 3D modelling. Many definitions of the term model can be found in the literature (Batty, 2001), which sometimes leads to confusions and misunderstanding.

One very general definition of a data model given for organising any type of data (Tsichritzis and Lochovsky, 1982) says that the model is a tool that provides an interpretation of the world and consist of *generating rules* and *operations.* The generating rules define the objects, their properties and mutual relationships captured in a certain time moment and reflect two aspects of the model: 1) *structure specifications* and 2) *constraint specifications.* Structure specifications establish the type and organisation of data, while constraint specifications focus accepted and allowed limitations. For example, a real building might be represented in a model as a set of planar, rectangular polygons (due to the structure specifications); however, the polygons may not intersect (due to constraint specifications). The generating rules result usually in a *data structure.* The operations describe all the actions that can be performed on the data (retrieve, delete, update, select, etc.). The larger is the set of possible operations, the more complex *data analysis* can be performed. Finally, the model with the corresponding user interface constitutes the system. The process of model production is called *modelling*. The *functionality* of the system is then qualified as the possibility to perform operations on data in order to analyse them and visualise the results.

Traditionally geo-scientist interested in objects with spatial extend and therefore differentiation between *spatial* and *non-spatial* objects is widely accepted. The model can comprise both spatial and non-spatial objects. Each of these objects can have their own characteristics, relationships and operations organised in one integrated or several independent structures. Spatial objects are represented by their *geometric* (shape, size, location) and *semantic* characteristics (called attributes) and *spatial relationships* (represented mostly by topology).

GIS was the fist system providing integrated maintenance of geometric and semantic characteristics and spatial relationships. Nowadays, many vendors dealing originally with only geometric or semantic data offer means for integrated modelling. CAD and GIS packages connect to databases with semantic information; DBMSs host spatial information (Quak et al, 2002). Apparently, the efficient geo-information management (especially when the third dimension is focussed) is rather complex task requiring high competence in different areas. Therefore leading vendors have decided to stream the efforts in GIS development by founding the OpenGIS consortium (currently consisting of more than 220 companies, government agencies and universities). The idea is to encourage technology developers to make complex spatial information and services accessible and useful with all kinds of applications. A logical consequence of the cooperation is the agreement on representation, access and dissemination of spatial information, i.e. the OpenGIS specifications (OpenGIS Consortium Inc., 1999). The specifications describe the model and are available in two variants – abstract and implementation specifications. While the abstract specifications are already completed, the implementation standards are still at developing stage.

3 OpenGIS specifications

According to the OpenGIS specifications, the spatial object (named *geographic feature*) is represented by two structures, i.e. *geometric (i.e. simple feature specifications)* and *topological (i.e. complex feature specifications)* describe the spatial properties. While the geometric structure provides direct access to the coordinates of individual objects, the topological structure encapsulates some of their spatial relationships. Thus, an application can benefit from the two representations, e.g. area, volume, distance can be completed on the geometric structure, while analysis based on neighbourhood operations can be performed on the topological structure. Currently, the attention of the vendors is toward the geometric model.

Here, we present our experiments with 3D data that give a very good overview on the current implementation status. We selected two large vendors (members of the OpenGIS consortium) that deal with spatial data, i.e. Oracle (Oracle Spatial 8i) and GeoGraphics (Bentley, 2001). GeoGraphics (extension of MicroStation) uses directly the geometric structure of Oracle Spatial. Thus, in this set-up, Oracle Spatial maintains the 3D data and GeoGraphics plays the role of a front-end engine to query, visualise, edit and post the changes back to the database. Prior discussing our tests, we will give a brief explanation of the models used by Oracle and GeoGraphics.

3.1 Geo-DBMS: Oracle

The geometric characteristics of spatial objects in Oracle Spatial 8i are defined by the *geometric type*. Currently, the supported geometric types are 2D (point, line, polygon) but 3D coordinates are accepted. The generating rules are very simple and intuitive. Lines and polygons are represented as an ordered set of coordinates (2D or 3D). Self-intersecting lines are allowed but self-intersecting polygons are not supported. Polygons with holes are maintained as well. Oracle is a object-relational DBMS and the geometric types are defined using exactly the object-oriented approach. They are defined in the *mdsys.sdo_geometry* object-relational model and contain information about type, dimension, coordinate system, holes of objects, and provide the list with the coordinates. The structure of the object is given bellow:

Name	Null?	Type
SDO_GTYPE		NUMBER
SDO_SRID		NUMBER
SDO_POINT		SDO_POINT_TYPE
SDO_ELEM_INFO		SDO_ELEM_INFO_ARRAY
SDO_ORDINATES		SDO_ORDINATE_ARRAY

Thus, the five parameters of the geometry for a 3D polygon with four vertices *v* (X,Y,Z), e.g. *v1* (10, 10, 0), *v2* (11, 9, 1), *v3* (11, 12, 0) and *v4* (9, 11, 1) will have the following values:

SDO_GTYPE = 3003. The first 3 indicates three-dimensional object and the second 3 indicates a polygon.
SDO_SRID = NULL. The coordinate system is not specified, i.e. decoded in the coordinates.
SDO_POINT = NULL. The described type is polygon and therefore the value is NULL.
SDL_ELEM_INFO = (1, 1003, 1). The first 1 in the sequence 1,1003,1 gives details about the geometry type (i.e. a simple polygon connected by straight lines). 1003 indicates that the polygon is an exterior ring. The final 1 specifies the geometry type, i.e. polygon. Furthermore, these particular values certify that the polygon does not contain holes.
SDO_ORDINATES = (10, 10, 0, 11, 9, 1, 11, 12, 0, 9, 11, 1, 10, 10, 0).

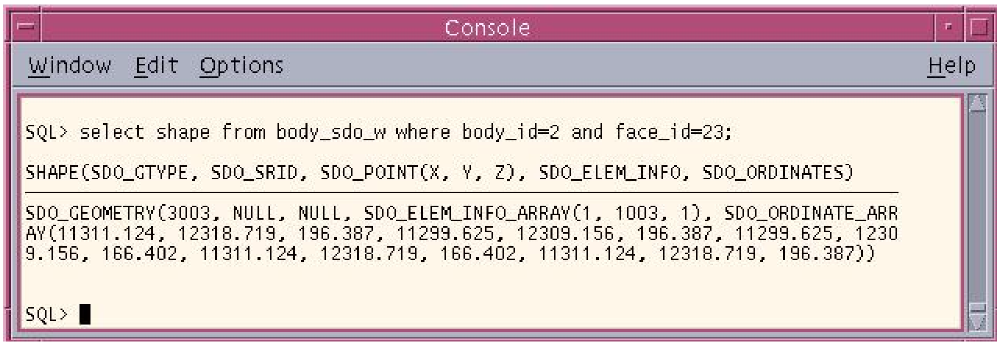


Figure 1: Representation of one polygon in Oracle Spatial 8i

Currently, the SDO_GTYPE allows decoding of 7 geometric types namely *point*, *line* or *curve*, *polygon*, *collection*, *multipoint*, *multiline* or *multicurve* and *multipolygon*. The type *collection* gives the possibility different geometric types to be organised as and considered an individual spatial objects. Figure 1 shows the representation of one 3D polygon (a face from a 3D object).

3.2 CAD: GeoGraphics iSpatial

In contract to Oracle, the definition of a spatial object in GeoGraphics cannot be done without specifying the semantic meaning (characteristics). Three levels of semantic hierarchy are maintained. *Feature* represents one or more objects from real world (e.g. the bank building, the school building). *Category* groups features with a similar theme (e.g. buildings, rivers). Finally, p*roject* refers to as the root and represents the data for the entire study area. One project can have many categories but a category may belong to only one project. To be able to distinguish between different spatial objects stored in Oracle Spatial 8i, each object has to be assigned to a feature (i.e. its semantics has to be clarified). Furthermore, edited and newly created objects cannot be posted in the database without attributing predefined features to them. Geometric characteristics of the objects are organised in one or more *spatial layers*, which correspond to geometric structure of Oracle Spatial.

4 Experiments with Oracle Spatial 8i and GeoGraphics Ispatial

To investigate the functionality of the two software products in representing, maintaining and visualising 3D spatial objects, we completed two case studies following two different approaches. In the fist case study, we had the 3D data organised in Oracle Spatial in user-defined relational tables and the task was to access, query and edit them from GeoGraphis. In the second case, the 3D data were available in a DGN file and had to be imported in Oracle Spatial.

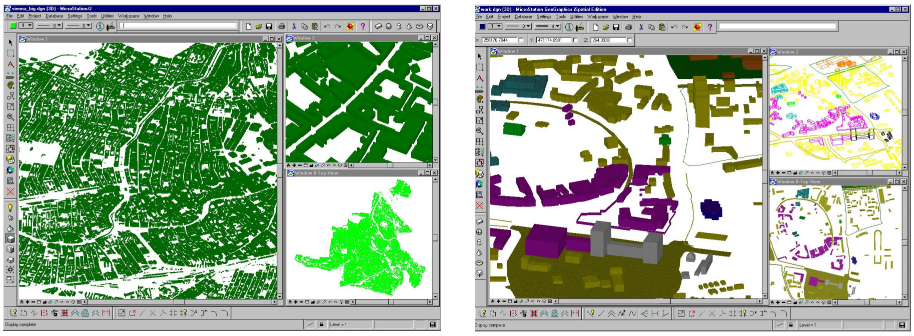


Figure 2: The data sets used in the experiments: Vienna (left) and Enschede (right)

4.1 Case Study 1: 3D data organised in user defined relational tables

A set of 21 000 buildings from the city of Vienna (Figure 2, left) was used for the test. The data were initially organised in a 3D topological structure (Zlatanova 2000) mapped into several relational tables in Oracle. An operation written in PL/SQL (i.e. a high-level programming language in Oracle) converted the data from the topological to the geometric model of Oracle Spatial 8i. Similar topology-geometry procedure is described in Oosterom et al 2002. Table 1 shows two possible descriptions of 3D objects within the geometric model of Oracle. In the first representation (Figure 3), each building has unique identifier (ID), stored in the column BODY_ID. The column

FACE_ID contains the unique ID of the face. The geometry of each face is organised according to the object-oriented model (SDO_GTYPE=3003, 3D polygons) of Oracle in the column SHAPE. Apparently, several records (at least 4) represent every building in the relational table. Although a bit inefficient, this structuring facilitates some types of queries (due to the stored relationships between the faces and the 3D object). For example, the query "find the neighbouring building" can be completed by comparing the IDs of the faces composing the buildings and thus avoiding the coordinate comparison.

Table 1: Descriptions of BODY_SDO table by: 3D polygons and a 3D collection.

Name	Null?	Type	Name	Null?	Type
MSLINK	NOT NULL	NUMBER(10)	MSLINK	NOT NULL	NUMBER(10)
BODY_ID		NUMBER(10)	SHAPE		MDSYS.SDO_GEOMETRY
FACE_ID		NUMBER(10)			
SHAPE		MDSYS.SDO_GEOMETRY			

In the second representation (Figure 4), the MSLINK column includes the ID of he building and the SHAPE column contains the 3D coordinates of all the polygons composing one building. Thus, every building is described as a collection of polygons (iSDO_GTYPE=3004, 3D collection). Although the number of records is reduced (i.e. one building is represented by only one record), the redundancy of coordinates cannot be avoided. Each triple of coordinates is repeated at least three times in the list of coordinates (i.e. in SDO_ORDINATES).

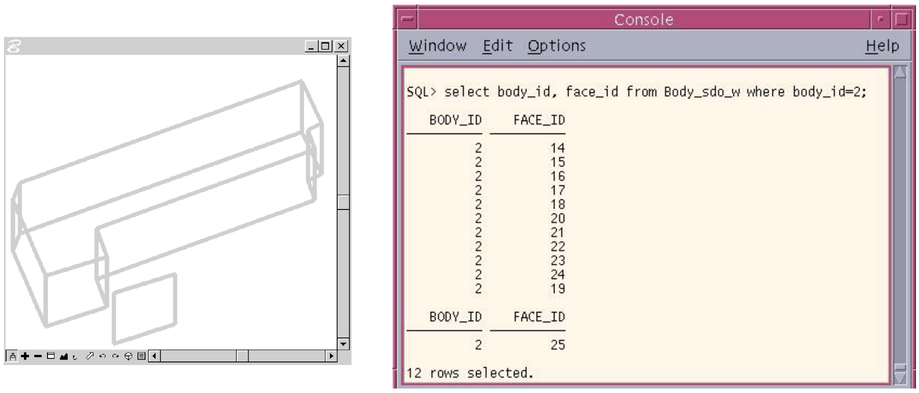
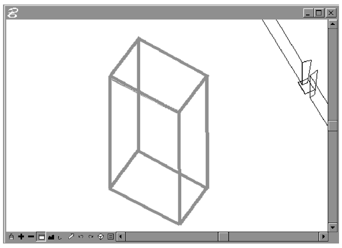


Figure 3: 3Dobject represented as a set of polygons: in GeoGraphics one of the polygons is shifted (left) and part of the relational table BODY_SDO in Oracle (right)

Microstation GeoGraphis interprets these two representations in a different manner. In the first case the building is visually one object, but practically, it is a set of individual polygons (Figure 3 left). The entire building can be selected only by placing a fence around all the polygons. In the second case, the building is interpreted as a "group", i.e. a single click of the mouse will highlight the entire building (Figure 4, left). In order to edit the object, however, the group has to be "dropped" into the constructing individual polygons. To send the changes back to the database, grouping of the objects will be again required. Otherwise, the object will be considered a set of several new polygons.

The steps to assess the data and query them are described in details in Zlatanova et al 2002. Basically, three major steps have to be followed:

1. Making reference between a spatial layer (in GeoGraphics) and the relational tables (in Oracle Spatial). Since the table with the geometric types (in Oracle Spatial) already exists, it needs to be declared as a spatial layer in GeoGraphics.
2. Creating semantics, i.e. features and categories
3. Linking features with the spatial objects. Running an appropriate script within Oracle is one of the easiest ways to complete this operation in case of many objects.



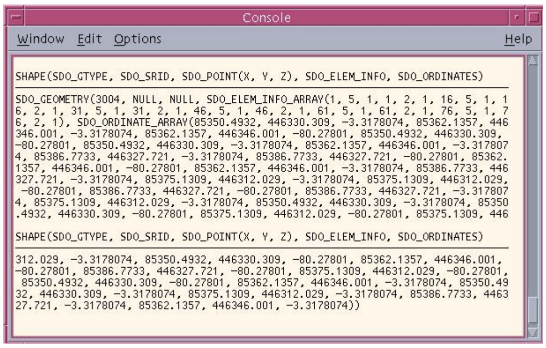


Figure 4: 3D object represented as a collection of polygons: in GeoGraphics (left) and the geometric representation in Oracle (right)

4.2 Case Study 2: 3D data organised in DGN file

The data for the second case study are obtained from a semi-automatic procedure for 3D reconstruction developed in ITC, Enschede. The current procedure is an extension to the one presented in (Tempfli, 1998). The manual digitising points characterising roofs of buildings in a photogrammetric stereo model (from aerial photographs) creates a "skeletal point cloud". The 3D reconstruction then consists of automatically computing and assembling all the faces (roof faces and walls) of the building from this point cloud. The model obtained in this way contains planar closed polygons, which normal vector points towards the outside of the building (to ensure correct 3D visualisation). The procedure is capable of processing a number of objects (not only buildings) as the reconstruction rules for other topographic objects are in most cases simpler than those for buildings. All reconstructed objects are organised in a topological data structure 3D FDS (Molenaar, 1990). In addition, a new DGN file with the 3D reconstructed model is created. This procedure (running on *SocetSet*, Leica) was applied to reconstruct the centum of Enschede (Figure 2, right). Another adaptation of the same idea for *SoftPlotter* (Automatric Inc., 1999) is presented in Vermeij and Zlatanova 2001, which allowed the reconstruction of several buildings in the campus area of the TUDelft. The 3D objects of these two procedures were also successfully imported in Oracle Spatial by the following steps:

1. Creating of features and categories.

2. Selecting the entire geometry (polygons or group of polygons) per spatial objects in GeoGraphics and attaching a feature to it.
3. Posting the spatial objects to the database.

In both cases studies, after completing the requirements of the both structures (in GeoGraphics and Oracle) it was possible to query, visualise and edit the objects. The query can be performed on the basis of the semantic characteristics of the objects as they are defined in GeoGraphics. For example, query on feature "buildings" will result in visualising all the buildings. If a feature (e.g. "the hospital") is attached to only one spatial object, then only that object will be extracted from the database. Apparently, this is quite convenient for editing and updating large 3D models. Rendering of thousands of polygons can be easily avoided.

5 Analysis of the current implementations

Our experiments clearly showed that significant progress in 3D modelling is made, i.e. storage and query of 3D spatial data is possible, although limited. The agreement on a model that can be used by different applications (types of software) for different purposes contributes greatly to the positive achievements. The model, however, is still rather simple: spatial objects are represented by their coordinates and the semantic properties are maintained by the front-end engine (but again in the database). Further developments are needed for unifying the semantic representation as well. The simplicity of the spatial model encourages many developers to use it. However, since information about spatial relationships is not maintained (in *mdsys.sdo_geometry* object), complex analysis cannot be performed. The current status of the 3D modelling (refer to chapter 2) can be summarised as follows:

Generating rules (objects, geometry, topology, reflectance). Our test revealed that the implemented geometric types of Oracle could easily represent 3D objects. The Z-coordinate it is not an attribute anymore. 3D object can be stored as a set of polygons (i.e. several rows in a relational table) or as one object, i.e. collection (one row in a relational table). Prior the real 3D geometry type is introduces, it is worth mentioning that the definition of geometric types in Oracle Spatial 8i permits better representation of 3D spatial objects. Stoter and Oosterom, 2002 propose new values of *mdsys.sdo_geometry* that allow more compact record. The array with coordinates is suggested to have two sections, i.e. a list of coordinates and references to the list. Such a structuring of the SDO_ORDINATES array will significantly reduce its length.

Spatial relationships represented by topology (especially 3D topology) need further development and implementation. Since different topologies may be appropriate for different applications, Oosterom and al 2002 propose their organisation and maintenance to be completed on a higher level, i.e. in a meta-data table. Topology-geometry and geometry-topology operations will ensure the consistency in both representations.

The support of parameters to describe physical properties of 3D objects is still missing. Physical properties play a critical role for realistic visualisation of 3D models. For example, many consider utilisation of real world images a solution to representing geometric details (Gruber et al, 1995). Such images, however, require strict organisation and maintenance in the model. Currently, the feature description (provided by GeoGraphics iSpatial) permits properties of lines (e.g. colour, width, gaps width, type line) to be specified, but no properties of polygons are considered. For example, the colour of the polygon (in a rendering mode) is selected with respect to the colour of the line. 3D photo-true visualisation is practically not possible due to lack of a mechanism to store texture and texture parameters.

Operations (retrieve, edit, post, spatial operations). As mentioned above, 3D objects stored in a database, permit the user to extract and concentrate on only a limited set of data and thus to reduce the time for loading and manipulation. Thus large 3D models can be easily edited and updated on a regular basis (as 2D maps). Presently, the operations are restricted to the geometric types defined in Oracle Spatial (e.g. polygons, line and vertices). In some cases this may require longer editing (processing) time. For example in the case shown in Figure 4, left, a shift of one face (wall) will change only the position of the selected polygon, which may differ from the intensions of the user. Furthermore, objects defined as spheres, cylinders, cubes and all types of extruded shapes (widely used in architectural modelling) have to be simplified to points, lines and polygons for editing and storing in the database.

Real possibilities of 3D spatial analysis in GeoGraphis iSpatial and Oracle Spatial 8i are still missing. Tools in GeoGraphics iSpatial to create 2D topological layers or tools in Oracle Spatial 8i to perform spatial operations (e.g. compare, intersect, within_dispance, area, length, validate_polygon) are provided but they operate with only 2D data.

7 Conclusions

It is apparent that the OpenGIS consortium has opened a new page in the history of the geo-management (beneficial also for 3D modelling). The first step (i.e. support of spatial objects on a database level) will certainly change the nature of GIS: instead of the traditional, desktop, monolith system, GIS will become an aggregation of CAD, GIS and DBMS. In this conglomerate of different systems, DBMS will play the critical role of integrated (spatial and non-spatial) data container inheriting all the achievements of the DBMS technology from the last several decades, e.g. high-level data management, data share, security, etc. We expect further extension of the capabilities of DBMS to maintain and analyse geo-data (in both geometry and topology domain). CAD and VR system will serve as powerful front-end engines ensuring extended graphics interface for 3D query, visualisation and navigation though the model.

References

1. *Batty, M*. 2001, Models in planning: technological imperatives and changing roles, in: JAG, Volume 3, Issue 3
2. *Bentley* 2001, Bentley MicroStation GeoGraphics ISpatial edition (J 7.2.x) (20 01). URL: http://www.bentley.com/products/geographics/
3. *Gruber, M., M. Pasko and F. Leberl*, 1995, Geometric versus texture detail in 3D models of real world buildings, in: Proceedings of automatic extraction of man-made objects from aerial and space images, Birkhauser Verlag, Basel, pp. 189-198

4. *Leica Helava Systems*, 2002, URL: http://www.lh-systems.com/
5. *Molenaar, M. 1990,* A formal data structure for 3D vector maps, in: Proceedings of EGIS'90, Vol. 2, Amsterdam, The Netherlands, pp. 770-781
6. *Oosterom, P. v, J. Stoter, W. Quak and S. Zlatanova*, 2002, The balance between geometry and topology, in: Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling, D.Richardson and P.van Oosterom (Eds.), Springer-Verlag, Berlin, pp. 209-224
7. *Open GIS Consortium, Inc.* 1999: The OpenGIS abstract specification, topic 1: Feature geometry. *Technical Report Version 4 (99-101.doc),* OGC, URL: http://www.opengis.org/techno/abstract.htm
8. *Oracle Spatial 8i*, 2002, User's guide and reference, available at: http://technet.oracle.com/doc/inter.815/a67295/toc.htm
9. *Quak, W., P. Oosterom and T. Tijssen*, 2002, Testing current DBMS products with realistic spatial data, In: Proceedings of AGILE 2002, 2-4 October, Praag, Czech republic (to be published)
10. *Automatric, Inc*., 1999, Softplotter 2.0 User's Guide, Bangor, URL: http://www.autometric.com/NEW/Products/Production/sp.html
11. *Stoter J.E.*, 2002, 3D Cadastres, state of the art: from 2D parcels to 3D registrations, GIM International, the world magazine for Geomatics, February, 2002.
12. *Stoter, J. and P. van Oosterom,* 2002, Incorporating 3D geo-objects into a 2D geo-DBMS, in: Proceedings of ASPRS/ACSM, 19-26 April, 2002, Washington, USA
13. *Tsichritzis, D. and F. Lochovsky,* 1982, Data Models, Prentice-Hall, New Jersey
14. *Tempfli, K. (1998)*: 3D topographic mapping for urban GIS, ITC Journal 3/4, pp. 181-190
15. *Vermeij, M. and S. Zlatanova* 2001, Semi-automatic 3D building reconstruction using Softplotter, in: Proceedings of the International Symposium on "Geodetic, Photogrammetric and Satellite technologies: development and integrated applications", 8-9 November, Sofia, Bulgaria, pp.305-314
16. *Zlatanova, S.*, 2000, 3D GIS for urban development, PhD thesis, ITC publication, The Netherlands, 222 p.
17. Zlatanova, S. and T. Bandrova, 1998, User requirements for the third dimensionality, E-mail seminar of Cartography 1998: Maps of the future, Vol. 1, Sofia, pp. 61-72
18. *Zlatanova, S., A. Rahman, M. Pilouk,* 2002, 3D GIS: current status and perspectives, in: Proceedings of ISPRS, 8-12 July, Ottawa, Canada, CDROM, 8p.