MODELLING FOR 3D GIS: SPATIAL ANALYSIS AND VISUALISATION THROUGH THE WEB

Siyka ZLATANOVA[*], **Klaus TEMPFLI**[**]
[*]Delft University of Technology, The Netherlands
s.zlatanova@geo.tudelft.nl
[**]International Institute for Aerospace Survey and Earth Sciences
tempfli@itc.nl

Working Group IV/2

KEY WORDS: 3D modelling, 3D topology, 3D visualisation, VRML, WWW, photogrammetric data collection

ABSTRACT

The paper presents an integrative approach to 3D modelling of complex scenes of determinate objects. The key characteristics of the proposed data model are support of 3D geometry and topology for extended spatial analysis, and facilitating interactive visualisation. The model supports the four geometric abstractions of spatial objects, i.e. point, line, surface and body. The representation based on two constructive elements (nodes and faces) is sufficient for the derivation of a large number of 3D topological relationships. The restrictions imposed on faces (connectivity, convexity, ordering) as well as the supplementary information maintained per object (radiometric parameters, behaviour) ensure the correct rendering and enable the design of realistic (photo textured), dynamic 3D worlds. The user interaction with the database residing on the Web relies on standard Internet tools — CGI scripting, VRML and HTML documents. The description of our prototype system for querying, visualising, and communicating with a DMBS is supported by several examples. The designed model is successfully linked to developed procedures for data collection. Using a digital photogrammetric system we can collect semi-automatically geometric and semantic data, and automatically extract texture from aerial photographs as necessary for texture mapping of roof facets or texture draping over the ground surface. We consider the approach a good step forward to an operational urban 3D GIS.

1 INTRODUCTION

The challenge of handling geo-information of urban areas has triggered this research. Prior to venturing into 3D GIS plenty of research had been done on semantic modelling and 2D spatial analysis. Much of it improved GIS design, found its way into software development, and promoted applications that rely on spatial information. Technologically driven and argued by application demand 3D GIS has recently been put on many R&D agendas. Sectorial research and development has already been reported on: data models supporting 3D topology (see Molenaar 1990, Pigot 1995, Pilouk 1996), extended query languages (see Mattos and DeMichiel 1994), data structures for fast visualisation (see Lindstrom et al 1996, Kofler 1998), photo texturing (see Gruber et al 1995, Leberl et al 1994), transferring and viewing 3D graphics on Internet (see Coors and Jung 1998, Doyle et al 1998, Verbree et al 1999), and new communication architectures and standards (see Orfali and Harkey 1998, Web3D Consortium 2000). The literature, however, reveals little evidence as yet about attempts to integrate the various concepts—under current technology—aiming at a more versatile system of handing geographically referenced spatial information.

This paper presents a design that aims at joining capabilities of GIS and CG. Central to the design is a new data model, which supports 3D topology (to facilitate 3D spatial analysis) and ensures correct, quick 3D visualisation on the Web. Adhering to an object view of space we choose a boundary representation for the geometric description of the complex spatial entities in urban areas. The next section gives a brief description of the conceptual data model for a 3D GIS, which is followed by outlining the logical design. The suitability of the data model to perform 3D spatial analysis is discussed on the basis of the 9-intersection concept, which was introduced by Egenhofer and Herring 1992. The Graphics User Interface (GUI) of our prototype system is demonstrated by a number of examples of remote query and visualisation of 3D spatial and non-spatial information on Web. The last section addresses data collection from aerial photographs for supplying data for an urban 3D model.

2 THE CONCEPTUAL MODEL

The core issue of conceptual 3D modelling is representing the geometry. The model presented here, named Simplified Spatial Model (SSM), follows the object paradigm and is strictly defined using set theory notions (see Zlatanova 2000). SSM distinguishes four *geometric objects* (*point*, *line*, *surface*, *body*) and two *constructive objects* (*face* and *node*). The geometric object can be associated with thematic meaning (description of semantics). The constructive objects are the smallest elements used for building the geometric objects. Thus, *nodes* describe *points* and *lines,* and *faces* describe *bodies* and *surfaces*. *Nodes* are connected with straight lines. The geometric objects are embedded in the space, i.e. a complete subdivision of the space is not required. This is to say that a curve can be represented by a set of nodes connected with straight lines. Similarly, curved surfaces can be represented by a set of planar convex faces. Each object (constructive or geometric) has a unique identifier, which is used for reference in the description. SSM complies with a strict boundary representation as the order of the constructive objects is recorded. The multi-theme concept is adopted, i.e. nodes and faces can participate in the construction of many geometric objects. *Nodes* are represented in the space by their 3D co-ordinates. *Faces* can have a polygonal boundary but must be convex and planar. Singularities between the constructive objects (i.e. *node-in-face* and *face-in-face*) are not permitted. If such a relationship occurred, the corresponding face would have to be subdivided. This construction rule is imposed in order to please the rendering engine (in our case a VR browser). The singularities *node-in-body* and *face-in-body* are permitted in order to maintain the relationship *inside* without subdivision of the bodies into unnatural parts.

SSM is “simpler” than the conceptual 3D models reported in the literature (3D FDS and the cell model; see Zlatanova 2000 for comparison) because of omitting *arcs* as constructive objects. This is profitable in the construction, visualisation and query of urban 3D models due to the following considerations:

- *faces* can have an arbitrary number of *nodes*, which is frequently required in describing urban objects
- *bodies* can be partitions according to semantic considerations
- restrictions imposed on *faces* ensure correct display by any rendering engine
- the only two constructive objects maintained speed up traversing of the data and reduce the storage space
- the scope of topological relations is as large as the one offered by 3D FDS and the cell model.

The model described so far refers to the geometric description of the objects, which is only a part of a conceptual model for 3D GIS. Yet semantic information, radiometric properties and behaviour have to be included (see Tempfli 1998b, Zlatanova and Gruber 1998). Radiometric properties provide information about the visual appearance of the objects, e.g. colour, surface structure, reflectance and thus they are important for 3D visualisation. Introducing the *behaviour* of objects aims at describing dynamics of objects and hence is related to the interaction with the virtual 3D scene. Examples of behaviour are: a window can be open, a fountain can splash water. We maintain the parameters related to radiometric properties and behaviour in the 3D database. In the following section, the conceptual data model is translated to a logical one, further elaborating on the geometric description, radiometric properties and behaviour.

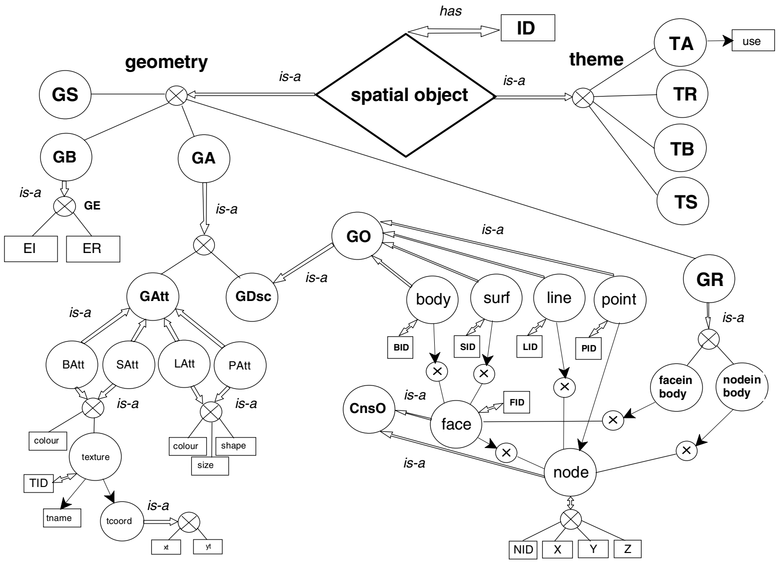

Figure 1: Simplified Spatial Schema (SSS)

3 THE LOGICAL MODEL

To describe fully a spatial object we consider *attributes (**A**)*, *relationships (**R**), behaviour (**B**)* and *scenario (**S**)*, sticking to the traditional differentiation of the representation into a geometric (**G**) and a thematic domain (**T**). *Scenario* refers to the time related changes of the objects. *Attributes* denote semantic properties as well as geometric and radiometric properties. Associating the geometric domain with all characteristics of a spatial object that relate to its appearance (and not the semantics, which be assigned to it), we call the size and shape of a spatial object, its position and orientation, and its radiometric parameters the *geometric appearance* (**GA** in Figure 1). **GA** is composed of two groups of properties: the elements for the digital description of geometry (**GDsc**) and those for visualising the geometry (called here the *geometric attributes* **GAtt**). Thus, **GDsc** refers to position, shape and size, and **GAtt** to the radiometric parameters. The *geometric description* (**GDsc**) is achieved by *geometric objects* (**GO**) and *constructive objects* (**CO**). The two explicit relationships (i.e. *node-in-body* and *face-in-body*) are organised under **GR.** Each geometric object has its own geometric attributes. The **GAtt**'s—serving visualisation—must be different for different geometric abstractions. Bodies and surfaces can do with two parameters, i.e. *colour* and *texture.* For lines and points texture is not useful, instead *shape* and *size* are needed. Shape and size of 0-dimensional and 1-dimensional objects are introduced to enable the rendering of these objects as 3D objects. For example, a pipeline might be recorded as a line, but readability of the 3D scene will improve if tiny cylinders instead of simple lines are displayed. Depending on the way of texturing (i.e. texture mapping or texture draping), only the name of the image file or the name and the image co-ordinates have to be maintained. Behaviour refers to the parameters need to introducing dynamics. To present the types of objects and the relationships among them (see Figure 1) in a schema, we use the IFO modelling technique (see Abiteboul and Hull 1987). The IFO technique suites our purpose, i.e. following an object-oriented approach and doing a relational implementation. The Simplified Spatial Schema (SSS) can be converted directly to a relational data model (see Zlatanova 2000).

Once defined, the data model has to be evaluated. The next two sections discuss the suitability of SSS for performing spatial analysis and creating realistic 3D scenes for specifying spatial queries and visualising 3D outputs.

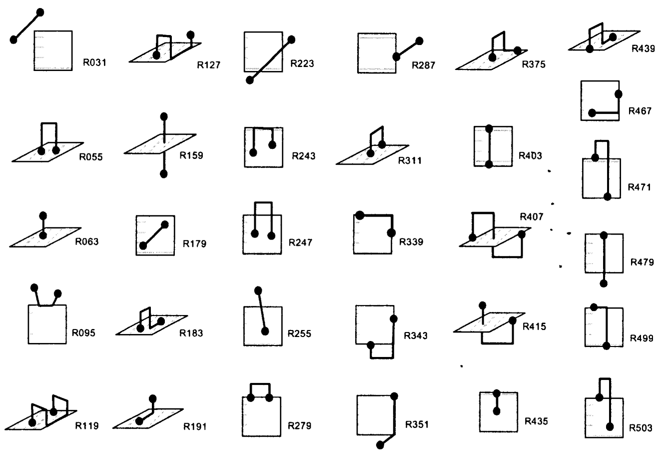

Figure 2: Surface and line in 3D: 31 relations (19 in 2D)

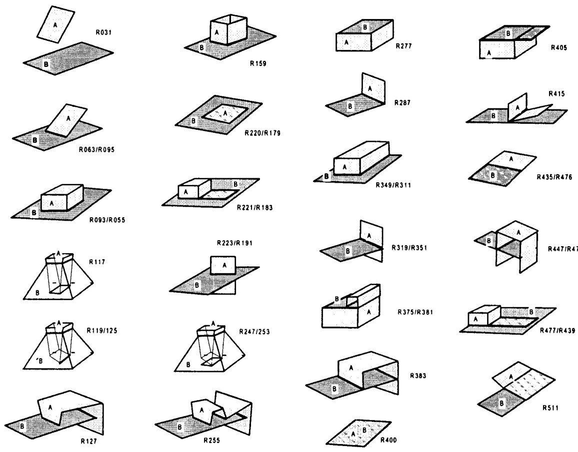

Figure 3: Surface and surface in 3D: 38 relations

4 ANALYSIS OF SPATIAL RELATIONSHIPS

The evaluation of SSS for maintaining spatial relationships utilises the framework introduced by Egenhofer and Herring. The spatial relations are clarified by investigating the intersections between the *interior*, *exterior* and *boundary* of two candidate objects. The number of all the detectable relations between two objects is 2^9=256. In reality, however, not all the relations are possible. Therefore, first, the possible relations between geometric objects in the 3D space should be singled out. The number of possible relations between simple objects was proven to be 69 (see Zlatanova 2000). Observing the 69 elementary relations, we can identify the relationships between all geometric objects. In 2D space only five groups of relationships are of interest: point and line, point and surface, line and line, surface and surface, and line and surface. The study of 3D topology is much more involved, we have to consider the relationships of body and body, surface and line in 3D space, surface and surface, body and line, and body and surface. As example, Figure 2 and Figure 3 show two of the groups of relations. The strict mathematical formulation and elaborated discussion is given in Zlatanova 2000.

The judgement of the suitability of SSS to perform spatial analysis is based upon the assumption that a specific relation has to be identified between two known objects. This is to say that the dimension of the objects (e.g. surfaces - 2D) and the type of the relation (e.g. "meet") are known in advance. The expected result is either positive (yes, the relation between the objects is the required one) or negative (no, the relations between the objects is different). This case is the easiest for analysis and implementation. The operations needed to complete inspection of any relation are among the set operations, i.e. *union*, *intersection* and *difference*. Some exceptional constellations require testing for a sequential order of nodes and belonging to a face. Since the node and the face are sets, the test for belonging to face can be converted to the operation intersection. The sequential order of the nodes in a face is explicitly specified, which allows the two sets (i.e. the set intersection and the set of nodes in a face) to be compared as ordered sets. Thus, the identification of topological relations relies entirely on the standard set operations. The benefit of this conclusion is that the operations are compatible with the standard SQL operations *select* and *join*, which is a premise for an effortless implementation in a RDBMS.

In reality, two more categories of queries are possible, which are less simple. The first one requires recognition of the relation between two objects. This is to say that the dimension of the objects is known but the type of the relation is unknown. For example, "what is the relation between the house (3D) and the garage (3D)", "find the type of interaction of the railway (1D) and the road (2D)", "what is the relation between the parcel (2D) and the lamppost (1D)".

The second category of queries refers to the identification of a relation regardless of the dimension of objects. Examples of such queries are "find all the objects, which meet", "find all the common walls in the neighbourhood". The query is relevant for a consistency check in the object reconstruction phase. Although such queries will require more complex operations, they are still supported by SSM.

5 VISUALISATON ON THE WEB

Testing the suitability of SSS for visualisation and interaction can most convincingly be done by having built a prototype. The developed system is server-oriented, i.e. using CGI scripting and VRML and HTML documents created on the fly. The VRML documents serve a twofold goal: 1) delivers the 3D graphics information obtained as a result of spatial query or/and 2) provides means to query graphically the objects observed in the 3D scene. Web and VR browsers on the client stations are utilised to interact with the 3D model(s) and specify queries. The data structured according to SSS are maintained in a RDBMS on the server.

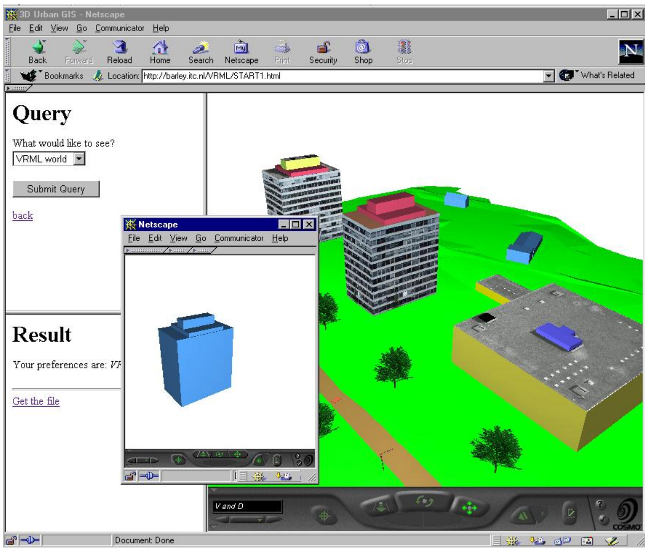

Figure 4: *Multiple- choice* menus

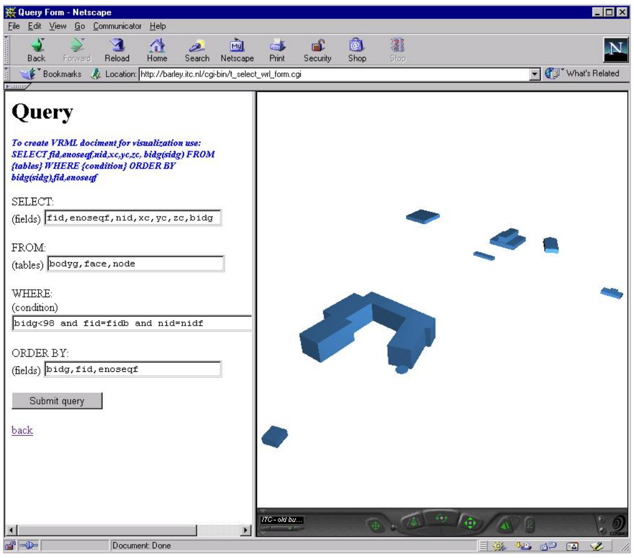

Figure 5: *One-line* SQL quires

The *multiple-choice* forms offer a very simple access to information about certain objects. The only disadvantage of the menus is the limited choice inherent in its pre-design. The user visually decides which object to query inside the VRML document by clicking on its graphic representation. The VRML document is created in such a way that it provides *point-and-click* operation (see Web3D consortium 2000). A *click* with the mouse on the building activates a CGI script, which delivers the *Query-Result* sections to the client station. The user selects the needed information from the pull down menu at the *Query* section. In the snapshot (see Figure 4) the building closer to the viewer is "equipped" with a VRML sensor and thus available for pointing.

The *one-line SQL* form enables the user with extended possibilities for querying (see Figure 5). The SQL statement is specified in an HTML fill-out form. The result of the query is displayed either in an HTML or in a VRML document. The SQL statement has to ensure sufficient data for VRML creation and efficient ordering of faces. The syntax of VRML requires restructuring the data as they are supplied by SSS. To facilitate this process the data and the order required are displayed in the fill-out form. The interface is based on a two-section framed HTML document. The left part is reserved for typing SELECT statements and right part is used to display either HTML or VRML documents. The extracted data are first visualised as text and then as 3D scene on user request. The free access to the database provides a mechanism to specify and display a wide range of spatial queries. Each request in the spatial domain (formulated by spatial or non-spatial restrictions) which can be described in one SELECT statement can also be visualised in a VRML document. Examples of such queries are “which is the highest building?”, “show the buildings in a particular area”, "show all the streets", "show all the administrative buildings". The same mechanism can be used to create DELETE, UPDATE, and INSERT forms to edit data.

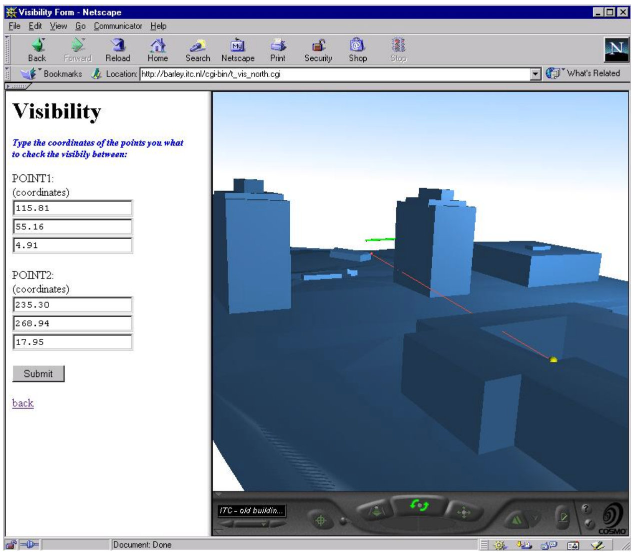

Figure 6: GUI to specify complex queries

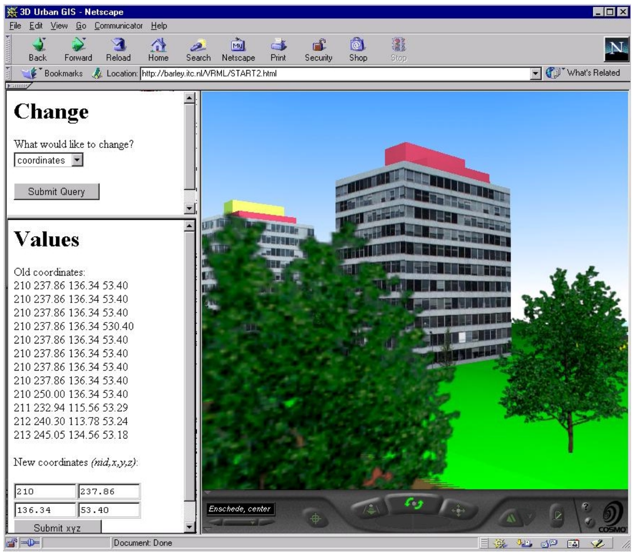

Figure 7: GUI for modification of data

Complex queries require the design of special forms. For example, the queries "check for common faces", "find common nodes" can be organised by typing (in the HTML form) the ID of the objects to be compared. Another interesting example is the visibility between two points. Suppose a mobile telephone company wants to validate the position of a transmitter. This can be translated to a query "check the visibility between the position of the transmitter and the roof of this building" or "show the range of the transmitter". To require such information, different specifications of the query are possible: 1) co-ordinates of begin and end points of the line of vision, 2) ID of the two points or 3) one point and the range of view (represented as e.g. the cone of view). In our example, we consider the case when the co-ordinates of the two are to be entered. The outcome of the query must be a set of objects, which disturb the view. Theoretically, this query requires complex 3D intersection algorithms between the line of vision and the faces forming the objects in the range of the line. Here, we present a simple solution based on a visual inspection of the actual path of a traversing line between the two points. The line of vision is drawn in the VRML world and the user can observe the obstacles (see Figure 6). The user has to type the co-ordinates of two points and as a result s/he gets a VRML document with a subset of the model surrounding the points of interest and the connecting line. In the VR browser, the user can navigate around the disturbing object, inspect and evaluate the situation. Appropriate *sensors* (e.g. extended *Touchsensor*, see Zlatanova 2000) attached to the objects provide identification information, e.g. the ID of the objects or the name of the owner (company or private person).

Using similar approach the edit operation at database level can be provided. In this case, the dynamically created VRML document is used for verification. Again the initial VRML document has to be equipped with the necessary *sensors* to detect user actions on a particular object. Currently, the form offers three options: change of co-ordinates of points, change the name of the image file used for texturing and change of the texture co-ordinates. The snapshot given in Figure 7 shows "change of co-ordinates". The operation is relevant for users who want to experience a building reconstruction. They will need to have means for pointing to a part of the building, and fields to import the new co-ordinates. The submission of the new values to the server will modify co-ordinates in the corresponding database fields and will create a VRML document for verification.

6 DATA COLLECTION

We have developed several procedures for data collection and constructed several 3D model. Focusing on 3D geometry, we can collect data from aerial photographs (stereo pairs) and structure them according to SSS and also convert data from several CAD formats, thus utilising existing city models. We use a visual/manual procedures for object recognition in images and feature extraction and automated "object reconstruction". Photo texture extraction can more easily be automated. In the following sections a brief description of the procedures will be given.

6.1 Semi-automatic 3D object reconstruction

The current procedure for 3D object reconstruction is a result of the research carried out in ITC since 1994 (see Mwewa 1998, Paintsil 1997, Pilouk 1996, Wang 1994). Initially developed for buildings, the procedure is now extended to comprise all types of objects, i.e. points, lines, surfaces and bodies. Thus, buildings are associated with bodies, lamp posts with lines, streets and parking lots with surfaces, manholes with points, etc. The assumptions that the buildings have only vertical walls, without windows and doors, and non-over-hanging roof facets are basic for the procedure. Lean to roofs, not being part of buildings, are modelled as separate surfaces. Terrain objects (streets, parking lots, etc.) are incorporated in the DTM (represented as TIN).

The main steps in the procedure are data acquisition, 3D object-reconstruction, superimposition, database updating and visualisation. Data acquisition refers to creating DTM and manual digitising of skeletal roof points using phogrammetric equipment (i.e. SocetSet and Microstation). The 3D object reconstruction involves creating roof facets and projecting roof outlines onto the DTM for computing footprints and constructing walls. The automatic reconstruction procedure distinguishes between points, lines, surfaces and bodies. Different types roofs (e.g. single-facet, multi-facet, multi-level) require different processing. Therefore the objects have to be processed either individually or grouped into similar types. The reconstruction can be validated in two ways: 1) the wire frame superimposed on the stereo model verifies the measurements and 2) the shaded model visualised by a VR browser verifies the orientation of the faces. Only the successfully reconstructed objects are recorded in the database. More details about the procedure can be found in Paintsil 1997, Tempfli 1998a.

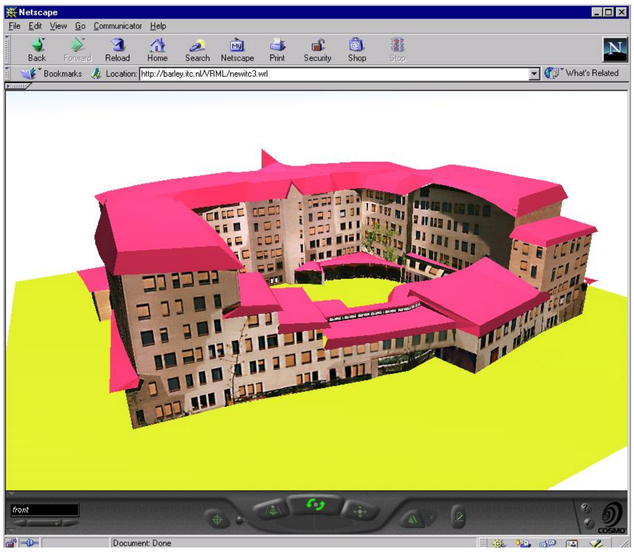

Figure 8: ITC building: texture mapping

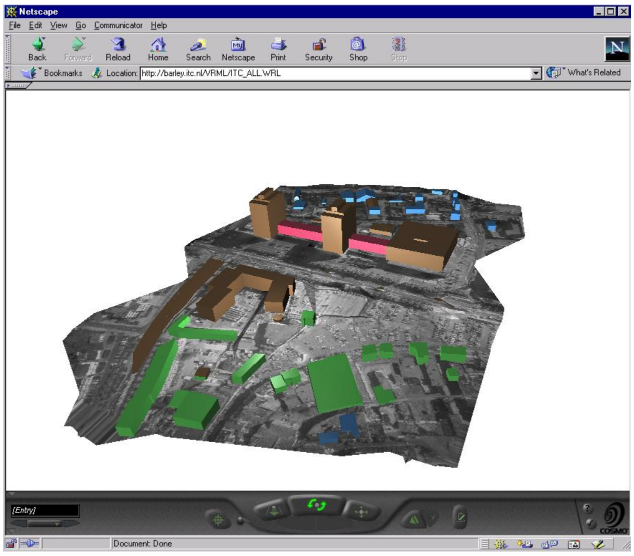

Figure 9: Enschede: texture draping

6.2 Photo texture extraction

Maintenance of two mechanisms for texturing are considered in our model, i.e. texture mapping and texture draping. Texture mapping, which requires correspondence between image and object co-ordinates, provides the precise mapping of images onto the "geometry". The mechanism is appropriate for objects where the exact fit is recommendable, e.g. roofs and façades of buildings (see Figure 4 and Figure 8). Texture draping is less precise and therefore applicable only for objects that are observed from a large distance, e.g. terrain surface (see Figure 9). For texture mapping, we distinguish three possible relations between a surface and an image file. First, a surface is mapped with image parts that belong to one image. Second, a surface is mapped with parts of originally separate photo images but united in one image file. Third, the faces of the surface are mapped with approximately the same parts of an image, i.e. the image file is one but the texture co-ordinates vary per face. These three cases can be readily organised in SSS.

Data storage for draping is simpler, since the draping mechanism does not require referencing between image and face co-ordinates; only the name of the image file has to be specified. The image, however, has to be oriented in a way appropriate for the draping mechanism adopted by VRML. Since SSS does not maintain additional parameters for this type of texturing, the image has to be pre-processed to meet the default values (for scale, shift and rotation) of VRML and then stored in the database.

One of the problematic issues in texture mapping is the automatic co-ordinates reference of object-image. In the case of nearly vertical images, we have used the approach of Sithole 1997 for extracting parts of images per face. The prototype software automatically selects the best image (of a stereo pair) for texturing and corrects for occlusions. The approach can be applied for collecting both textures for roofs (see Figure 4) and textures for faces constituting surface objects (on the terrain). An extension of the procedure for façade images still has to be developed. For the examples shown here, the façade images of the models are processed manually in VR modeller (*Medit*) and then imported in SSS by in-house developed converting software.

7 CONCLUSIONS

In this paper we presented an approach for integrated modelling for 3D GIS, aiming at 3D model which supports 3D topology and is coupled with an user interface for querying and 3D visualisation on the Web. By linking the model design to procedures for data collection we completed the chain of the 3D modelling process.

The data model presented here (maintaining geometric, radiometric and thematic characteristics of objects) was successfully tested for both performance of spatial analysis and 3D visualisation on the Web. The hitherto completed research let us draw a number of conclusions. The major difference with similar "topological models" is the maintenance of less constructive objects. The omission of arcs, however, does not disturb the recognition of topological relations and speeds up the visualisation process. The provided organisation of radiometric characteristics (i.e. texture) ensures both precise texture mapping and texture draping in a very condense way. Maintaining behaviour as a geometric characteristic allows dynamics of objects to be permanently stored at database level.

The implementation of the data model using a RDBMS and a web-oriented GUI indicated the feasibility of an easy, low-cost access to the database. Most components of the system are freeware, some investments are only necessary for the organisation of the server side. The user interface on the client side relies on standard Web techniques (i.e. HTML, VRML). The proposed system architecture demonstrates that a "working" 3D GIS can be obtained by using existing technology achievements.

We expect benefit of the designed system for "urban users", because of

- possibilities to query graphically objects in a virtual environment
- visualisation of 3D spatial queries in a virtual environment
- virtual reality navigation and exploration of 3D worlds
- possibilities to visualise outcomes of queries in different file formats
- access to the information from any office via the Internet
- provision of moderate means to edit data over the Web and visualise the changes.

REFERNENCES

Abiteboul, S. and R. Hull, 1987, IFO: A formal semantic database model, ACM Transactions on Database Systems, Vol.12, No.4, December 1987, pp. 525-565

Coors, V. and V. Jung, 1998, Using VRML as an Interface to the 3D data warehouse, Proceedings of VRML'98, New York

Doyle, S., M. Dodge and A. Smith, 1998, The potential of Web-based mapping and virtual reality technologies for modelling urban environments, Computers, Environment and Urban Systems, Vol. 22, No. 2, pp. 137-155

Egenhofer, M. J. and J. R Herring, 1992, Categorising topological relations between regions, lines and points in Geographic databases, The 9-intersections: formalism and its use for natural language spatial predicates, Technical report 94-1, NCGIA, University of California

Gruber, M., M. Pasko and F. Leberl, 1995, Geometric versus texture detail in 3D models of real world buildings, Automatic extraction of Man-made objects from aerial and space images, Birkhauser Verlag, Basel, pp. 189-198

Kofler, M., 1998, R-trees for the Visualisation of large 3D GIS databases, PhD thesis, TU, Graz, Austria

Leberl, F. and M. Gruber, 1996, Modelling a French village in the Alps, Proceedings of the 12th Spring Conference, Budmerice, Slovak Republic

Lindstrom, P. D. Koller, W. Ribarsky and L. Hodges, 1996, Real-time, continuos Level of Detail rendering of height fields, Proceedings of SIGGRAPH'96, New Orleans, Louisiana, USA, pp. 109-118

Mattos, N. and L. DeMichiel, 1994, Recent design trade-offs in SQL3, ACM SIGMOD record, Vol. 23, No.4, 6 p.

Molenaar, M., 1990, A Formal Data Structure for 3D Vector Maps, Proceedings of EGIS'90, Vol. 2, Amsterdam, The Netherlands, pp. 770-781

Mwewa, L., 1998, Object-oriented roof model construction from skeletal point clouds: for building reconstruction, MSc. Thesis, ITC, The Netherlands

Orfali, R. and D. Harkey, 1998, Client/Server programming with Java and CORBA, John Wiley&Sons, Canada

Paintsil, J., 1997, 3D Topographic data by aerial photogrammetry, MSc thesis, ITC, The Netherlands

Pigot, S., 1995, A topological model for a 3-dimensional Spatial Information System, PhD thesis, University of Tasmania, Australia

Pilouk, M., 1996, Integrated modelling for 3D GIS, PhD thesis, ITC publication, The Netherlands

Sakseau, J., 1998, 3D object reconstruction, Project report, ITC, The Netherlands

Tempfli, K., 1998a, 3D topographic mapping for urban GIS, ITC Journal 3/4, pp. 181-190

Tempfli, K., 1998b, 3D topographic mapping, Lecture notes PHM112, Geoinformatics, ITC, Enschede

Tempfli, K. and M. Pilouk, 1996, Practical photogrammetry for 3D-GIS, Proceedings of ISPRS, Vol. XXXI, Part B4, Vienna, Austria, pp. 859-867

Verbree, E., G. van Maren, R. Germs, F. Jansen and M.J. Kraak, 1999, Interaction in virtual world views—linking 3D GIS with VR, Int. J. Geographical Information Science, Vol13, no 4, pp. 385-396

Wang, Z., 1994, Digital Photogrammetric Data acquisition for 3D GIS, MSc thesis, ITC, The Netherlands

Web3D Consortium Inc., The VR repository, http://www.web3d.org/vrml/vrml.htm (January 2000)

Zlatanova, S. 2000, 3D GIS for urban development, PhD thesis, ITC publications (to be published)

Zlatanova, S. and M. Gruber, 1998, 3D GIS on the Web, in: Proceedings of ISPRS, Com. IV, 7-10 September, Stuttgart, Germany, pp. 691-699