GEOMETRICAL AND TOPOLOGICAL MODELS FOR REAL-TIME GIS

Sisi Zlatanova (1), Daniel Holweg (2) and Volker Coors (3)

(1) Delft University of Technology, Delft
The Netherlands
(2) Fraunhofer Institute for Computer Graphics, Darmstadt
Germany
(3) Stuttgart University of Applied Sciences, Stuttgart
Germany

ABSTRACT

Many GIS, GeoDBMS and CAD systems contain data about real word aiming different applications and tasks. Traditionally, most of these applications do not require time-critical response. However, resent progress in mobile technology as well as increasing need of spatial information on 'spot', gradually change access, update, query and visualisation from relatively 'time-independent' to 'time-dependent' and even 'time-critical'. The interesting question is then whether the current developments in spatial data management offer sufficient functionality and performance to support time-critical applications.

his paper discusses the requirements to a time-critical application and addresses current commercial support and maintenance of spatial information (2D and 3D) offered by GeoDBMS. Bearing in mind the OpenGIS specifications, both geometrical and topological representation of spatial data is discussed. A special attention is paid on storage and query of 3D data since the problems in maintenance of multidimensional information has been always problematic. Some tests investigate functionality and performance with respect to time-critical requirements with both models.

1. INTRODUCTION

The slogans of today's world is 'time is money'. This shows the value of time in our all day life. Furthermore, about 80% of all data are somehow spatial related and most users have experiences with maps, being a simple and intuitive kind of visualisation for complex spatial themes. This leads to the idea of using map-based visualisations and GIS in time critical applications (e.g. emergency management).

Time-critical applications (TCA), in the context of this paper, are related to decisions that have to be made by a human decision maker in emergency situations. The IT just supports the decision maker in getting the highest quality and quantity of data and delivering it in an intuitive way to improve the quality of human decision. Often different decision makers with different competencies and tasks have to make their decision using the available data. The aim of GIS based decision support in area of TCA is appropriate data management and efficient data discovery and integration to facilitate the decision maker whenever he needs to make a decision.

In this context, it is interesting to see what is the status of GIS in support of TCA:

Firstly, the nature of GIS is changing. The integrated architecture, i.e. storing spatial data together with administrative data and the topology in DBMS (GeoDBMS) is now getting more and more mature. Many DBMS offer support of spatial data and provide spatial analysis on them. An increasing number of GIS/CAD systems supply extensions to access the spatial data stored in DBMS.

Secondly, an intensive work on data standardisation is presently observed. One of the last initiatives of the OpenGIS Consortium is the OpenLS (Open Location Services) specifications, which aims at standardising the interface for location services in order to make it available at mobile devices.

Thirdly, technology developments in mobile communications are greater than ever. Mobile devices (PDAs and mobile phones) have become daily-use tools. Cell phones incorporate functionality that few years ago were only domain of PDAs and pocket computers. PDAs and pocket computers are upgraded with communication abilities or have them out of the box. The proponents of this new class of devices often cite location-based services such as incremental guidance in a foreign city as a key benefit. Even Multimedia and 3D data are nowadays available on mobile devices.

These developments definitely have a crucial impact on the utilisation of databases for all types of TCA. In the representation of data to the end-user, Kray at all 2003, distinguish two types of critical resources, i.e. technical (speed, band width and screen resolution) and cognitive (the way the information is presented to the end user). In this paper, we concentrate on another aspect, i.e. the impact of different spatial representations that may be used in GeoDBMS.

Several critical issues play a role in achieving a good performance at database level: representation of 2D/3D spatial objects, spatial indexing and offered spatial operations. Two types of applications, Augmented Reality and Location Based Services, are focussed. The rest of the paper is organised as follows. The following sub-sections discuss the requirements to a GIS supporting TCA. Section 2 after briefly presents the geometric model of Oracle Spatial and reports the tests on the model. Sections 3 concentrates on the topological representations utilising particular 3D topological models. Last section comments obtained results and indicates topics for future research.

2. REQUIREMENTS FOR GIS BASED TIME-CRITICAL SYSTEMS

Decision makers in emergency management often have to make time-critical decisions in situations of extreme stress. A system to support this process has to integrate different kinds of spatial data, analyse it and present the relevant information in an intuitive way. On one hand such architecture has to manage different kinds of data and integrate them on the basis of spatial or semantic relationships. On another hand a set of clients is necessary to address the different decision maker anytime and anywhere.

Figure 1 shows an example of such an architecture considering an emergency management scenario.

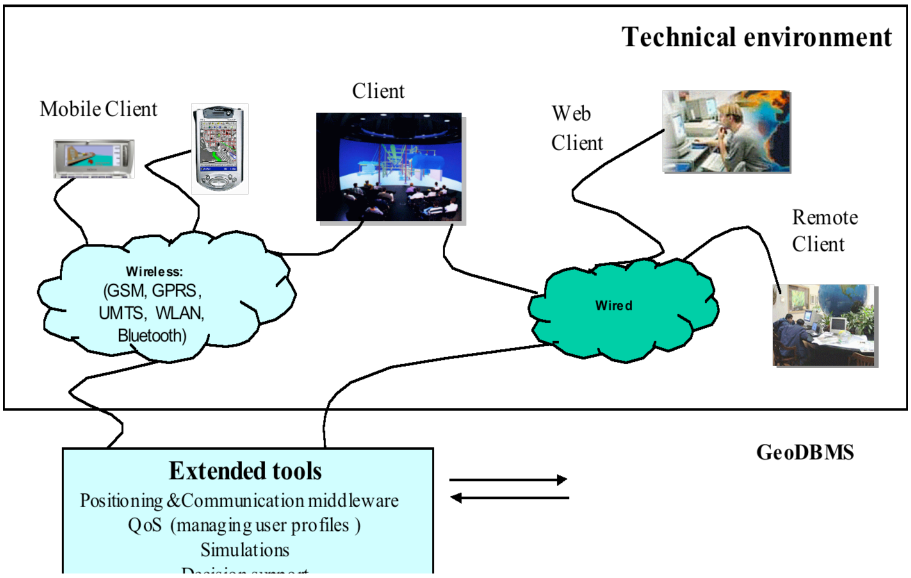

Figure 1: GIS based support in emergency management.

For such TCA it is necessary to even be independent of the location of the decision maker when he/she is asked for his decision. For example, one never knows when a case of emergency occurs and one hardly would know where a decision maker would be.

As mentioned above, there are two ways to increase the use of GIS in TCA. Firstly by decreasing the time necessary for searching, transmitting and analysing the data. Secondly by increasing the quality of visualisation to limit the time a user needs to understand the information displayed. Mobile clients and 3D visualisation are technologies supporting this idea (Holweg and Jasnoch, 2003).

A mobile client makes the integrated information available to authorized decision maker at anytime and anywhere. Moreover, mobile devices can be used as tools to modify and update the data. Therefore mobile forces will (has to) be equipped with such mobile clients. The changes transmitted via a wireless network (e.g. GSM, UMTS) are integrated into the system immediately and become available to all other users. This way of data import has the advantage of reducing the amount of data flow, since only differences have to be transmitted.

The second technology of importance for TCA is 3D visualisation. As mentioned before, 3D visualisation may decrease the time a user needs to read and understand a map. Professionals have long-term experience with 2D maps. However, we strongly believe that 3D visualisations and especially 3D maps will significantly decrease the time a user needs to understand the content of the displayed information. Since the world we are leaving is 3D, we all have by default better experience in interacting and understanding 3D worlds. It has to be taken into account that a time critical decision even for a trained decision maker is a situation of extreme stress. Therefore it is necessary to give him/her the visualisation fitting best to his current needs. Factors influencing the way of visualising information are the technical environment (available memory, battery status, screen resolution, etc.) the user has access to and the current mental situation of the users (e.g. level of emergency). The user should concentrate on the decision has to be made and not on understanding the visualisation of the data that is given to him!

2.1 Emergency management as example for a time critical application

A survey in Germany proceeded in 2002 by the Fraunhofer-Gesellschaft (Fraunhofer Gesellschaft, 2002), has shown the current situation and the potentials of GIS in the area of emergency management. Main result was that a general GIS infrastructure is already installed at most parts of the emergency management. But to improve its use it will be necessary to make better use out of the available data and to integrate all data and tools into an overall architecture (Figure 2). A support system for emergency management mainly consists out of three groups (see also Zlatanova and Holweg, 2003):

Core of the system is the *GeoDBMS*. Here the different 2D and 3D data are managed and any other kind of data can be connected to them.

Second group are the *Extended tools*. These tools use the stored data as input and create higher-level information out of them. Examples of such tools are simulation or Decision Support Tools.

The third group describes the *Technical environment* to give the data to the user. Following groups of such technical environment are described:

- Client – to give access to the management at the location the emergency occurs (e.g. technical management)
- Remote client – to give access to the decision makers that are not located at the local management central, but even need full access to the data (e.g. policies)
- Mobile client – integrating mobile forces and decision makers into the system. First to allow them accessing the data for their own decisions, second to allow them updating the current data in the system
- Web Clients – to give external users (e.g. public, press) access to the data (to the subset of data that should be available for public use)

All clients have to access the same set of data and database (under restriction of the rights the different users have) to distribute consistent data. Even some of the clients will change the data during processing. **Therefore it is critical to have an effective and well-structured model for the data.**

Among all the three types of clients, the mobile clients are most demanding, since they are on the field and perhaps in dangerous surroundings. Will concentrate further on the requirements for mobile users.

Depending on the type of visualisation, we distinguish between immersive and non-immersive mobile clients. In immersive systems, the end-user observes real world augmented with supplementary information (text, virtual 3D objects, images, video). These types of systems require usually special equipment amongst which see-through head-mounded display. In non-immersive systems, the end-user observes a portable display (on PDA, cell phones, portable computers). The requirements to both types of systems are much higher compare to a desktop system (e.g. web clients, remote clients). Besides providing user-requested information (nearest parking, shortest path to exits, location of underground facilities, owner of a building, etc.), the system has to be able to locate the user and to adjust the requested data to the corresponding front-end device.

2.2 Immersive mobile client

Usually, the spatial information for immersive mobile client (e.g. AR systems) is rather simplified and relatively small amounts of data (since most of the 'objects' are already in the view) are extracted from the database (Figure 2). The objects for rendering are virtual and therefore no elaborated texture mapping (e.g. photo texturing) is required. Thus the data for AR systems can be specified as simple data sets, delivered in a very fast manner (to compensate all the lags in the system). Moreover, the rendering subsystem (for visualisation of virtual objects) should be able to 'imitate' occlusions, i.e. to create the illusion that virtual objects are partially occluded by real object. This certainly requires accuracy and consistent description of outlines of 3D objects (e.g. man-made objects such as buildings).

Many AR architectures still rely on a vision system for positioning, since the accuracy of current GPS positioning (decimetre) might not be sufficient for rendering virtual objects at a close distance (centimetres). Architectures using a vision system, however, have higher time requirements. Besides the virtual object for rendering, data (lines, features) for position of the system are needed. In this case, the spatial data has to be supplied to the mobile set within several seconds (Zlatanova and Van den Heuvel 2002).

Figure 2: Visualisation of a virtual statue in the field of view.

2.3 Non-immersive mobile client

In contracts to the AR systems, the data for visualisation in non-immersive applications (e.g. LBS) are relatively much more and texture mapping is very often recommendable. The devices are rather limited in terms of rendering capability, screen size, and limited bandwidth (Figure 3).

Currently, textured 3D models can be already rendered on PDA if a frame rate of 10 fps but mostly as videos. Still more developments (in terms of appropriate clients and performance) are needed for achieving interaction in 3D vector models. The presently low visualization performance will be surely improved following the rapid development in the range of graphic processors, exceeding even the Moore principle, which says that the performance of processors doubles every 18 month due to technology changes. On the other hand, to achieve

the same performance the size of the processor will shrink. Knowing that performance of graphic processors exceed Moore's principle, we expect hardware accelerated graphics on PDA in the near future.
The small screen size on PDA limits both 2D and 3D maps. Unfortunately, this will not change much in future. Therefore, we have to deal with the screen size during the mapping process. For example, interactive tool tips should be used instead of a map legend to save space on the screen. However, the screen size problem will be one of the main challenges in GeoVisualization of the future.

The currently available mobile network standard GPRS allows a transfer rate of 57,4 kBit/s. With third generation mobile communication networks like UMTS the bandwidth will increase significantly and transfer rates up to 384 kBit/s are expected in urban areas. However, the high data volume of 3D maps will be critical even in high bandwidth wireless networks. As the limited bandwidth of current mobile networks is one of the main bottlenecks in the visualization pipeline, specialized compression methods for 3D geometry have to be used to minimize the amount data.

Figure 3: 3D models with Nokia GL on the Communicator 9210 as one result of the TellMaris project (IST-2000-28249, http://www.tellmaris.com). © Nokia Research.

Transmitting a 3D map stored as a 3 MB VRML file via UMTS will last about 60 seconds. A standard compression algorithm like gzip reduces the data volume to 1 MB and transmission time to 20s. Still, 20s is a long time to wait for the user. A recently developed Delphi compression method, which is specialized on 3D geometry, can compress this 3D map down to 180 KB (Coors and Rossignac, 2004). Transmission of this file will take less than 4 seconds.

Clearly, it is very important to extract data from the database in a very fast manner to be able to support TCA. We firmly believe that one important aspect within the long list of factors influencing the performance is organisation of the data in the DBMS.
The following sections will address current possibilities for organising data in DBMS.

3. GEOMETRICAL MODELS

Increasing number of DBMS (Ingres, Informics, Oracle, MySQL, etc.) provide maintenance of geometry types as the implementations follow (in certain degree) the OpenGIS specifications. Here, we discuss Oracle Spatial, because it constitute a large share of DBMS market and a large number of frond-ends (AutoDesk, Bentley, ESRI) make use of the model.

3.1 Geometrical representation in Oracle Spatial

The maintenance of spatial objects is done by three basic components: a schema (MDSYS) that prescribes the storage, syntax, and semantics of supported geometric data types; a spatial indexing mechanism; a set of operators and functions for performing area-of-interest queries, spatial join queries, and other spatial analysis operations and administrative utilities.

Object-Relational Model: Oracle Spatial supports an object-relational model for representing geometries. The object-relational model uses a table with a single column of MDSYS.SDO_GEOMETRY and a single row per geometry instance. A spatial object can be represented by geometry types, which are basically an ordered sequence of vertices that are connected by straight-line segments or circular arcs. Oracle Spatial supports several primitive types and geometries composed of collections of these types: points and point clusters, line strings, n-point polygons, arc line strings, arc polygons, compound polygons, compound line strings, circles, optimised rectangles. Oracle Spatial maintains only 2D objects in 3D space (with x,y,z coordinates).

The syntax for representing geometry object is only one (in contacts to OpenGIS specifications) as the type (point, line, polygon) is decoded in a parameter. Another parameter (array) contains the list with coordinates. The model is primarily designed for 2D objects with 3D coordinates. 3D objects can be represented either by simple geometry type (i.e. as a set of 3D polygons) or as a collection geometry type (i.e. 3D collection and 3D multipolygon). Using the first approach, one or two more columns have to be introduced in the relational table, to be able to specify that a polygon belongs to a particular 3D object. One 3D object is represented by several rows in the geometry table. This representation is a bit inefficient, but a 'kind' of topology (i.e. stored relationships between the faces and the 3D object) may be maintained. For example, a simple comparison between the IDs of the polygons composing the buildings can easily complete the query 'find the neighbouring building' (thus avoiding the coordinate comparison).

In the second case, a 3D object is described in one row, since all the information about the polygons is decoded in the Oracle Spatial geometry type. Although the number of records is reduced, the redundancy of coordinates cannot be avoided. Each triple of coordinates is repeated at least three times in the list of coordinates.

The two representations are recognised by front-ends for visualisation and editing. We have tested these representations with GeoGraphics (extension of MicroStation, Bentley) and ArcGIS (ESRI, direct connect and using ArcSDE) to query, edit and post the changes in the database (Stoter and Zlatanova, 2003). It is also possible to create a VRML (X3D) file that can be sent for visualisation to a mobile client (Figure 4).

Spatial indexing: To speed up the access and operations of spatial data, Oracle Spatial provides (and requires) spatial indexing. The spatial index can be an R-tree index or a quad tree index, or both. Each index type is appropriate in different situations. One even can maintain both an R-tree and quad tree index on the same geometry column. Some of the properties of R-tree indexing are: index creation and tuning is easy, relatively less storage space is required, the indexing can be up to 4 dimensions. However the geometries cannot be fine-tuned. In case of heavy update of the geometry column, the R-tree is not a good choice. Usage of quad tree allows fine-tuning of geometries and updates do not affect the performance of the quad tree. However, the performance of some operators is worse, storage space is larger and tuning is more complex. To select the appropriate indexing, analysis of the data and the operations on them has to be made in advance. The default spatial index is R-tree as the dimension is two. If a spatial index has been built on more than two dimensions, only one built-up operator (SDO_FILTER, the primary filter or index-only query) can be used, which considers all dimensions.

Spatial operators and functions: Oracle Spatial offers a large number of built-up operators and functions on the basis of geometrical model. The difference between operators and functions is that the spatial operators use spatial index and the functions not. Therefore the functions are relatively slow and the user has to apply own conditions to restrict the search. All the functions and operations work only with two dimensions, i.e. with the projection of geometry types. For example, the area of a vertical polygon (e.g. a wall of a building) will be computed as zero.

Some important functions are validation, area/length computation, distance, a new geometry of two intersecting geometries, etc. The validation function checks a number of rules for correct description of geometry types: the order of the points, repetitive points, self-intersecting polygons, etc. The validation is limited to 2D, i.e. validity of 3D shape (intersecting polygons, closed volume, etc.) cannot be checked.

3.2 Tests with geometrical model

We have tested the performance of several functions and operations (SDO_RELATE, SDO_WITHIN_DISTANCE, SDO_AREA, SDO_INTESECT, SDO_UNION, etc.) that might be useful for time-critical applications using with several data sets (2D and 3D) organised in Oracle Spatial geometry model. For example, SDO_WITHIN_DISTANCE is the most suitable for the queries needed for an AR system based on a vision system. Given a position and radius of interest, the function returns all the objects within this radius. We have implemented a SQL/PL function Field-of-View (FOV) that further limits the number of objects with respect to the direction and angle of view.

Figure 4 shows a VRML file created on the fly as the FOV function is executed. The green sphere is the point of interest (e.g. the user of the AR system). The position, direction and the angle of the FOV are known (i.e. obtained from the GPS receiver and the inertial system). The radius of interest can be specified with respect to the 3D model that is used (in case of many objects it can be reduced to 200-300 m.). The function is executed on a database level for several different distances and performance is excellent. For example, an area of interest less than 700m (actually much larger that can be seen by the user) can be extracted within 3 seconds (out of 21 000 buildings represented as set of polygons).

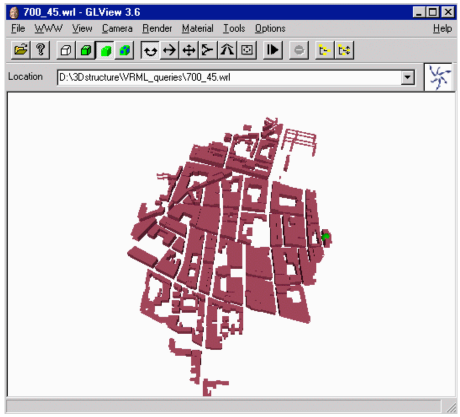

Figure 4: Field-of-View: a PL/SQL function extracting data in a given distance and angle, along given direction.

4. TOPOLOGICAL MODELS

There is a long-term discussion on utilisation of 3D topological models for TCA. Currently, there is not a system (GIS or DBMS) that supports natively 3D topology. All the 3D topological models that have been developed and implemented are usually user-defined and mapped in relational representations. Being not supported by the vendor, they are usually relatively slower than the geometric representation. However, several 3D topological models have been already used for TCA. In this paper we focus two implementations: Simplified Spatial Model (tested for AR application) and the Urban Data Model (tested for LBS).

SSM: For the purpose of visual AR system developed within UbiCom project (http://bscw.ubicom.tudelft.nl/), a 3D model was used, a modification of SSM (Zlatanova and Heuvel 2002). Each physical object is associated with four abstractions namely *point*, *linestring*, *surface* and *body*, that are built of simpler elements, i.e. *node* and *face*. Nodes describe faces, linestrings (e.g. pipe lines) and points (e.g. trees, lampposts). The order of the nodes in the face is maintained as wheel. The orientation of the faces is anticlockwise looking at the objects (e.g. buildings) from outside. Faces represent surfaces (e.g. streets, parking lots) and polyhedrons (e.g. buildings). The 3D coordinates are stored with the nodes. All other references are to the ID of the low-level elements. Details on facades of buildings or on the ground are maintained as loose lines with a link to the polygon they belong to.

The conceptual schema mentioned above is implemented using different approaches. The first straightforward approach is the **relational implementation**. For each object a separate relational table is created. The implementation of the table for nodes is trivial: one column for the identifier of the node and the three columns for the (geodetic) co-ordinates of the points. The rest of the tables have one column for ID of the high-dimensional object, one column indicating the order of low-dimensional object and a third column contains the ID of the constituting low-dimensional object.

Next possibility is creating **object-oriented views** from the relational tables. Views are especially appropriate for retrieval of standard data sets, e.g. the geometry needed for composing a VRML file. The last possibility is **object-oriented implementation**. Practically, this is a two-step procedure, i.e. creating objects and creating tables. We use two extended

Oracle data types, i.e. *varrays* and *nested tables.* While *varrays* are recommended for objects which elements are always retrieved in their completeness, nested tables are said to be suitable for accessing and retrieving individual elements of an object. We have implemented and tested both representations.

All the reconstructed 3D objects are recorded in these representations, but for performance test another relatively large data set. The basic query used for the test is 'extract objects needed by the AR system for the accurate positioning' for given position and direction of view. The performance has shown advantages of relational representation and object-oriented views compare to nested tables and variable arrays. Further appropriate spatial indexing and tuning of the database are recommendable for the object-oriented implementations. The best timing for 600 buildings extracted from 1600 buildings is 10 seconds (Zlatanova and Heuvel 2002).

UDM: In the Urban Data Model each object is associated with one of the four abstractions named *body, surface, line*, and *point*. If a feature has a zero-dimensional spatial extension it is represented as a point. The geometry of a body or a surface is given as a boundary representation described by planar convex *faces*. Each face is defined by a set of *nodes* with coordinates in 3D space (Figure 5). Two convex planar polygons are adjacent if they share at least two nodes. Because most graphics hardware is optimised for rendering triangles, an arbitrary polygon is decomposed into triangles. As a triangle is always convex, every face can be created using such a triangulation (Coors and Flick, 1998).

The main advantage of that model is the very efficient storage of geometry data. The restriction of faces as convex polygons or triangles introduces a storage overhead in comparison to arbitrary polygons. However, due to this restriction face adjacency could be derived directly from the database without explicitly storing edges. This reduces the number of stored objects by a factor of 2.

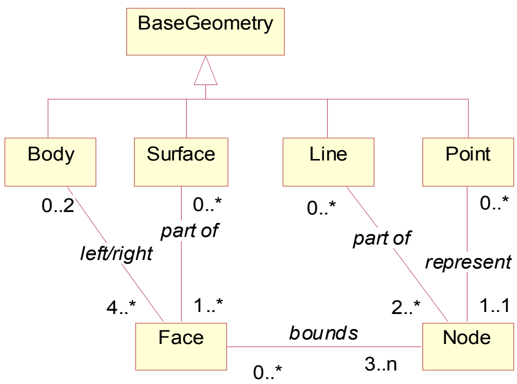

Figure 5: Geometry and topology in UDM.

The conceptual model was mapped to a relational model. The *Node* table is trivial: one column for the identifier of the node and three columns for the coordinates of the points. The relationship between a face and its constituent nodes is many to many (m:n) in the conceptual model. A face as a convex polygon has three or more nodes. Each node bounds several faces (6 on average in a regular 3D model). In the implementation, a face is restricted to a triangle. Every polygon can be triangulated without any loss of information. With this restriction, the three columns *node1*, *node2* and *node3* define a face in counterclockwise order. The face orientation, which is essential for visualization purposes, is implicitly given by the counterclockwise order of the nodes. Looking from the outside of the object, that is the normal vector that points to the outside. Each face bounds one or two bodies. For each face, these bounded bodies are referenced in the *body_left* and *body_right* column.

The proposed data model is used to manage an area-wide city model of Darmstadt. The data was captured by a German telecommunication company in order to simulate the their mobile network. In the data model, building-part is represented as a body-Feature, while each roof is a surface-feature. Each area, that is part of a roof, has a building-part body on it's left or right side. There are only five roof types supported: flat, shed, gable, hipped, and pyramidal roof. Other roofs are generalized to one of these types to speed up the data capturing process. The height of the roofs accurate with a 10% error tolerance. Wall and floor plan of a building part are modelled as surfaces in a similar way. A building consists of one or more building-parts. The number of building parts mainly depends on the building geometry and the semantic of its parts. For example, the castle of Darmstadt has several parts that have a meaning in the real world such as 'bell tower' besides 'part of the castle'. If these semantics should be stored, it is necessary store these entities as features, not only as geometry. Other relevant data are single walls, trees and other vegetation with a height of more than 3 meters.

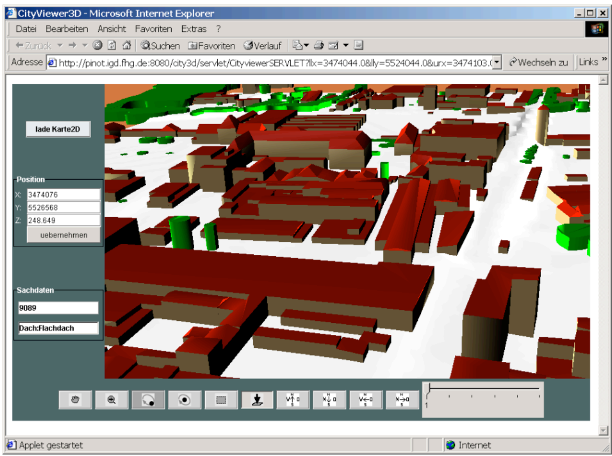

Figure 6: Example of 'window' query from Darmstadt data set.
The result of the query is a 3D model.

The proposed data model was implemented as a relational data model in Oracle. There are about 20.000 body features (buildings), 57.000 surface feature, 420.000 faces, and 250.000 nodes in the data model. The pure building geometry without terrain data and textures is about 83MB data in our implementation. This data was successfully queried for different applications (Figure 6).

4. CONCLUSIONS

Currently, the representations using the geometrical model have showed better performance compare to the topological models, which is not a surprise. First of all, the nature of the queries that were performed was pure geometric, i.e. select object within given area that is related to 3D coordinates of objects. In the topological model the 3D coordinates are stored in one table, which means that all the other tables have to be traversed to obtain them. In contract, in the geometric model, they are organised in one table (one or more records).

Second, the geometric model is integrated within the DBMS (and thus optimised), while the topological model is organised in user-defined objects and tables.

Third, DBMS maintain spatial indexing, which is not applicable for topological models. This is to say that presently, the geometric model is more appropriate for TCA compare to the topological model. However, the topological has a number of advantages compare to the geometric model, e.g. avoids redundant storage, easier to maintain consistency, efficient for certain query operations (e.g. find neighbours). We estimate that the two representations (geometrical and topological) will have to be in use for quite long time. Topological models will be used to perform specific topological operations and the geometric model will be used to 'realise' the geometry for visualisation.

Research and developments are needed in both models.

Geometrical models maintain only 2D objects and most of the functions are using only the two coordinates (x,y) of the objects. The geometrical models have to be extended to:

- Comprise 3D-object with the corresponding 3D functions (validation of 3D object, volume, etc.).
- Maintain operations and functions making use of all the 3 coordinates
- Have more operations, i.e. larger utilisation of spatial index
- Provide a consistent conversion to topological models
- Maintain more generic functions for data reduction, e.g. generalisation, back-face culling, field-of-view.

Currently, no DBMS maintain 3D topological models. Several implementations of 2D topology exist, as some main-stream vendors offer native support of 2D topology (e.g. Oracle Spatial 10g). Still many issues left to be addressed by the researchers:

- Metadata for topological models, i.e. possibilities to maintain many topologies in one DBMS (Oosterom et al. 2002)
- Index for topological models
- Operations on topological models, including conversion to geometrical models. We expect better performance of the 9-intersection model implemented in the topological model compare to the geometrical model.

We believe that once implemented in DBMS, the 3D topological model will contribute largely to the entire functionality and performance of the DBMS. It should not be forgotten that the main advantage of topology is maintenance. If one moves a point in a topological model, the model is still valid. In a geometry model (e.g. in Oracle Spatial object-relational model) one should have to move the same point several times to get a valid model because coordinates are redundantly stored

The functions currently available for the geometrical model will not be redundant. They will be still needed when building the topological model. For example if, a new polygon is introduced in the model, its relationships with neighbouring objects has to be established. If an intersection of geometries is detected, new objects have to be created and organised in the model.

Based on tests and experiments performed in GeoDBMS, we can conclude that the support of spatial data in DBMS has significantly improved in the last several years. We expect further progress in GeoDBMS for TCA in very short terms.

REFERENCES

Coors, V., and S. Flick (1998): Integrating Levels of Detail in a Web-based 3D-GIS, in: 6th ACM Symposium on Geographic Information Systems (ACM GIS 98), Washington D.C., USA, 1998

Coors, V. and J. Rossignac (2004): Delphi: geometry-based connectivity prediction in triangle mesh compression, The Visual Computer Vol. 20, Spinger, Heidelberg, Germany.

Fraunhofer Gesellschaft (2002): Gemeinsame Studie (GS) Marktanalyse Katastrophen- und Notfallmanagementsysteme.

Holweg, D., and U. Jasnoch (2003): GIS-Unterstützung im Katastrophen- und Notfallmanagement, In: Kartographische Schriften Band 7 Visualisierung und Erschließung von Geodaten – Seminar GEOVIS 2003, Kirschbaum Verlag.

Kray, Ch., Ch. Elting, K. Laakso and V. Coors (2003): Presenting route instructions on mobile devices, IUI'03, January 12-15, 2003, Miami, Florida, USA, 9 p.

Oosterom, P. van, J. Stoter, W. Quak and S. Zlatanova (2002): The balance between geometry and topology, in: Advances in Spatial Data Handling, Springer-Verlag, Berlin, pp. 209-224.

Oracle Spatial 9i, 2002, User's guide and reference, available at: http://technet.oracle.com/doc/

Stoter, J. and S. Zlatanova (2003): Visualisation and editing of 3D objects organised in a DBMS, Proceedngs of the EuroSDR Com 5. Workshop on Visualisation and Rendering, 22-24 January, Enschede, The Netherlands, 16 p.

Vries, M. de and S. Zlatanova (2004): Interoperability on the Web: the case of 3D geo-data, in: P. Isaias, P. Kommers and M. McPherson (eds.), Proceedings of the IADIS International Conference e-Society 2004, Avila, Spain, July 16-19, pp. 667-674.

Zlatanova, S. and D. Holweg (2003): 3D Geo-information in emergency response: a framework, , in: Proceedings of the Fourth International Symposium on Mobile Mapping Technology (MMT'2004), March 29-31, Kunming, China 6 p.

Zlatanova, S. and F. van den Heuvel, 2002: 3D GIS for outdoor AR applications, in: Proceedings of the Third International Symposium on Mobile Multimedia Systems & Applications, ISBN 90-9016467-7, 6 December, Delft, The Netherlands, pp. 117-124.

CVS OF THE AUTHORS

Sisi Zlatanova has graduated as a surveyor at the University of Architecture, Civil Engineering and Geodesy (UACG), Sofia, Bulgaria in 1983 and has obtained her PhD degree on '3D GIS for urban modelling' at the Graz University of Technology, Graz, Austria in 2000. She worked as a software designer at the Central Cadastre in Sofia, Bulgaria (1984-1989), an assistant-professor at UACG (1989-1995) and a researcher at the International Institute for Geo-information Science and Earth Observation (ITC), Enschede, The Netherlands (1995-1999). Since 2000 she is working at section GIS Technology, OTC, Delft University of Technology. Her research interests are in 3D GIS: 3D object reconstruction, 3D data structures and 3D geo-databases, 3D spatial relationships (topology) and 3D visualisation (VR and AR).

Daniel Holweg received his diploma 2001 in geodesy at the Technical University of Darmstadt. Since then he is working as a researcher at the GIS department at Fraunhofer Institute for Computer Graphics (Fraunhofer-IGD) in Darmstadt. Since November 2003 he leading the GIS department and became Managing Director of InGeoForum. His research activities lie in the domain of GIS based visualizations to support decision makers and the use of 3-dimensional GIS. He was involved in several national and international applied research and development projects.

Volker Coors is Professor in Geomatics and Computer Science at Stuttgart University of Applied Sciences, Germany. His current research is focused on 3D Geographic Information Systems, in particular compression and visualization of 3D urban models. He holds a Diploma degree with honors in Computer Sciences and a doctoral degree in Computer Graphics from Technical University of Darmstadt. From 1997 to 2002 he worked as a researcher at Fraunhofer Institute for Computer Graphics, department GIS.

CO-ORDINATES

Sisi Zlatanova
Delft University of Technology
Research Institute for Housing, Urban and Mobility Studies
GIS Technology
Jaffalaan 9
2628 BX Delft
The Netherlands
Tel. +31-15-2782714
Fax +31-15-2782745
E-mail z.zlatanova@otb.tudelft.nl
Website www.otb.tudelft.nl
www.gdmc.nl

Daniel Holweg
Fraunhofer Institute for Computer Graphics
Fraunhoferstr.5,
64283 - Darmstadt
Germany
Tel. +49 6151 155-415, +49 6151 155-420
Fax +49 6151 155-444
E-mail daniel.holweg@igd.fraunhofer.de
Website http://www.igd.fraunhofer.de/igd-a5

Volkers Coors
Department of Geomatics, Computer Science and Mathematics
Stuttgart University of Applied Sciences
Schellingstr. 24,
70174 Stuttgart
Germany
Tel. +49 (0)711 9980130
E-mail volker.coors@hft-stuttgart.de