TOWARDS 3D-GIS: EXPERIMENTING
WITH A VECTOR DATA STRUCTURE

V.Bric, M. Pilouk, and K. Tempfli
Department of GeoInformatics
ITC, Enschede
E-mail MORAKOT@ITC.NL, TEMPFLI@ITC.NL
Tel. +31-53-874358 Fax. +31-53-874335
The Netherlands

ABSTRACT

Spatial analysis has traditionally used two-dimensional techniques. Three-dimensional computer graphics offer more realistic presentations of geoinformation than what can be conveyed by a 2D map, especially of urban areas. Database technology offers efficient ways to handle large and complex sets of data. Development of 3D-GIS should utilize advances in CAD, GIS, and (digital) photogrammetry. The most important factor is an adequate 3D data model. Focusing on urban phenomena, we have selected a vector model. To test its practicability, we have built an experimental PC-based 3D-GIS and examined various aspects of data acquisition, database creation, querying and visualization. This paper also presents an elegant way of computing topologic space.

1. INTRODUCTION

The demand for spatial information is most pressing in urban areas. The majority of the population lives, builds, moves about and earns its living in urban areas (*eg*, 75% of the U.S. population; in Britain, some 90% of the population is concentrated on about 12% of the land; Petchenik, 1991; Rhind, 1993). Traditional sources such as maps and 2D-GISs can only partly support civic activities ranging from administration to planning, construction, facility management, security assurance, environmental management, conservation, etc. There is a rapidly increasing interest in having full 3D information available for better inventory, analysis, visualization and prediction--thus a 3D-GIS.

For developing a 3D-GIS for an urban area, which components are already available? Designing, manipulating and graphically presenting three-dimensional objects can be done by commercially available CAD systems. Especially impressive for a user is the high level of realism of 3D computer graphics, attainable by ray tracing and animation. The underlying data model of a CAD system is in essence limited to the geometry of solid objects, using either a volume or a boundary representation. Boundary representations (*eg* polyhedrons) are favoured for the visualization of solid objects, whereas a volume representation (*eg* constructive solid geometry) is likely to have advantages for object-oriented data access. Conversion between volume and boundary representation is possible (see Bric, 1993, for a detailed overview of CAD models).

CAD models, however, have ususally have no capacity to handle topology between solid objects and cannot deal with point objects, line and surface objects. CAD systems are not built for analyzing spatial relationships among various abstractions of the real world. Moreover, they do not support association with a multitude of thematic properties. Such desirable features are available in 2D-GISs. Taking into consideration that many a municipality has built already a 2D-GIS with spatial analysis tools and a wealth of thematic data, Förstner and Pallaske (1993) suggested a hybrid CAD-GIS concept. The functionality of both systems could be maintained by adding three-dimensional geometric data, as structured by a CAD volume representation, to the list of attributes of the 2D-GIS objects. There are open questions, however, concerning the flexibility of data access and problems of updating. On the other hand, extended tools are already provided by the software industry. ArcCAD (the "marriage" of ARC/INFO and AutoCAD) offers 2D analysis and 3D presentation, AutoCAD SQL (ASE link to dBASE, PARADOX, ORACLE, etc) offers the possibility to attach thematic attributes to objects structured in AutoCAD. The first solution does not provide 3D-GIS capabilities, the second allows building a 3D-GIS, but it is limited to solid objects and simple data retrieval. There does not seem to be an easy way to implement complex topologic queries. Desirable possibilities such as "walk through and query" are not (yet) provided.

An alternative is to define one single 3D data model that can cope with geometry, topology and semantics and can be handled by a single DBMS. The desired compatibility with an existing 2D-GIS model that is specifically suited for man-made objects and with a 3D boundary representation for visualization suggests a vector model. Molenaar (1992) developed such a model, the 3D formal data structure (3D-FDS).

In addition to the basic issue of adequate data modelling and providing tools for analysis and visualization, the third important aspect is data acquisition. Photogrammetry can offer accurate and economical techniques for acquiring geometric data of topographic objects, such as buildings, streets, urban green, etc. Data collection techniques that are supported by today's commercially available analytical and digital stereo-plotters produce vector data.

Our present investigation has not aimed at a fundamental analysis of information requirements for the various civic activities and by what kind of data model they could best be served. We want to explore the prospects of the 3D-FDS and identify problems in using it. To start with, we limited ourselves to (simple) buildings, an important component of an urban scene. To this end, we built a rudimentary 3D-GIS and we will report here on our approaches and first experiences with manual extraction of houses from images, editing and building topology, querying the databases, and visualizing objects and query results. For the implementation of queries, we developed an algorithm for computing the topologic space.

2. DATA MODEL AND STRUCTURE

Molenaar (1988) suggested a model for vector-structured 2D representations of a terrain. What he called the formal data structure of a single-valued vector map distinguishes three data types: terrain features, their geometric primitives, and their thematic attributes. These components and the links among them are shown in figure 1. Although the coordinates can be three-dimensional, the model provides only 2D topology. As such, it suffices for many applications, being merely (topographic) surface-related without detailed interest in phenomena above or below it.

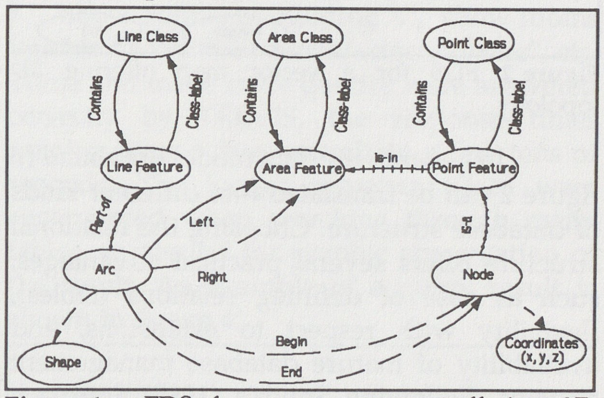

Figure 1 : FDS for a vector map offering 2D topology

To extend to full 3D, a metamorphosis of *area feature* is necessary (see figure 2). Introducing the two-dimensional geometric primitive *face* supports both surface features and body features, a surface feature consisting of one or more faces, a body feature being bound by faces (*eg* a park in a city could be modelled as a surface feature). A body is the smallest solid object meaningful to a user, *eg* a room in a building. A traffic-light could be a point feature, a power-line a line feature. A face is a polygon in space bounded by a closed chain of arcs; it can be a plane facet or mathematically simple curved surface - as defined by *fshape*. In the present investigation, we limited ourselves to simple objects and thus planar faces. *Edge* is another additional geometric primitive, needed to provide a link between arcs and faces that assures distinction between left and right bodies with respect to a face. Edges add a sense of direction to arcs. Arcs need not be straight lines; their shape is defined by *ashape*, but here we do not consider curved arcs. The 3D-FDS can be seen as an extension of the edge-based boundary representation known from solid modelling. Considering single-valued vector maps, Molenaar (1992) linked each feature to one thematic class (see figures 1 and 2).

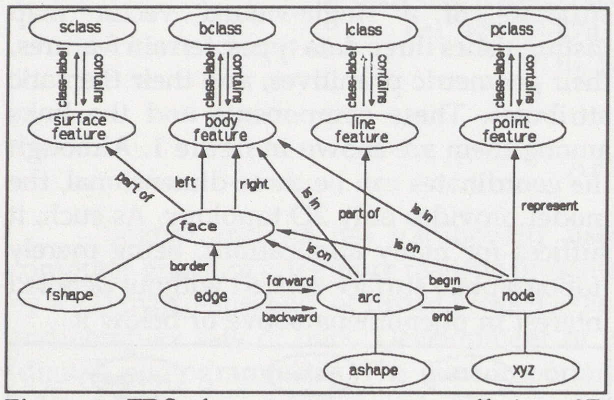

Figure 2 FDS for a vector map offering 3D topology

In general, the data model presented in figure 2 can be translated into different kinds of database structure. Choosing the relational structure offers several practical advantages, such as ease of defining relations (tables), flexibility with respect to extensions, and availability of mature database management systems. Following Smith's (1985) approach yields fully normalized tables (see figure 3), thus ensuring avoidance of redundancy, which will safeguard database integrity in data updating processes. For implementation, we selected the widely available dBASE IV DBM-software (see also Pilouk and Tempfli, 1993).

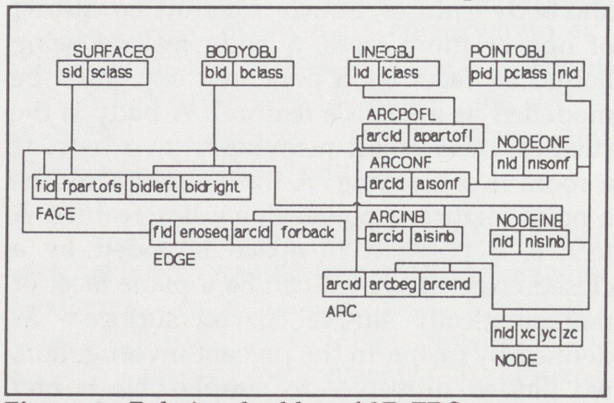

Figure 3 : Relational tables of 3D-FDS

3. TOPOLOGY

To investigate topologic relationships among spatial objects in 3D and implement queries, a formal approach is needed because of the vast number of existing complex relationships. Egenhofer and Herring (1992) developed a method for the analysis of binary topologic relationships for 2D, which was applied by Meij (1992) to 3D. By distinguishing between the interior, the boundary and the exterior of an object, we obtain nine possible intersections of two objects. Allowing only binary relationships implies that there are as many as $2^9 = 512$ different topologic relationships possible between two objects. Not all of them are relevant, however, when considering real objects. In order to determine the minimum set of relevant relationships, we developed and tested an algorithm based on set algebra (see Bric and Pilouk, 1994). Tedious manual elimination can now be replaced by an elegant computation of topologic space.

4. TREVIS

For building the experimental 3D vector GIS (TREVIS, Bric 1993), standard components were used: A PC 486/33, 210Mb HD, colour graphics monitor with an ATI VGA 24XL graphics card and a monochrome monitor with a Hercules card, and a mouse; dBASE IV with SQL and compiler; AutoCAD; Zeiss C120 analytical plotter equipped with Kork digital mapping software. The program shell and display graphics were programmed in Turbo Pascal, the queries in dBASE IV.

Access to the system is from a shell. It combines the 3D graphics and the database modules and offers various utilities. The queries that were programmed can be selected from a menu. The results are shown on the monochrome monitor if they are text. Graphic query results are shown on the colour monitor, superimposed on the display of all objects.

A minimum set of tools were developed for visualization, providing the following capabilities. Objects are presented as wireframe drawings in perspective views, either monoscopically or stereoscopically using the anaglyph principle. The graphics are produced directly from the dBASE files. For viewing purposes, data are kept in video memory; it is therefore very easy and fast to change the viewing position, the target position and the scale. This allows inspection of an object from all sides and with different enlargements. For large datasets, we still have to solve the clipping and panning problem. All viewing functions are also available for displaying query results; depending on the mode of display, the answers are shown in either a different color or with a higher intensity than the underlying objects.

A 3D cursor allows the user to move around in the perspective view or in the anaglyph stereo-model and snap to the nearest node or to the middle of the nearest arc. The result of snapping, a node or arc identifier, can readily be used as input to a query. Some queries use this possibility by asking the user for graphic or textual input. When operating in anaglyph mode, a "stereo-cursor" (like a photogrammetric floating mark) can be used instead of the 3D cursor; this provides an even better pointing facility. Also available are rudimentary editing functions. Using the 3D cursor, simple 3D objects can be constructed. The result of "digitizing" goes into two files with the same content as the NODE and ARC tables. Further development is still needed to obtain full editing capabilities and to fill the database with clean data.

5. QUERIES

By navigating through the 3D model, it is possible to find whether a particular question posed to the system can be answered. For the designer it is very important to know which questions can be answered (and which cannot). Unfortunately there is no formalized procedure available for determining the complete query space of a data model, but analytic thinking helps. The response time to a query depends on the data structure used for implementing the model and, of course, on the hardware and DBM software employed. For experimental testing we created two databases.

The first database we created contained several geometrically simple objects. The tables were filled according to sketches of the artificial objects with manually numbered faces, ordered edges and nodes. The most time-consuming step was filling the EDGE table. The database was used to test several sets of topologic queries ranging from simple ones such as "coordinates of a node", "nodes of a body feature", "faces of a surface feature", "give class of body X", "give body in which point X is lying", etc, through to standard ones such as "which lines touch line X", "which surfaces share arc X", "which bodies touch face X", "give line features that bound surface X", "give body object on which line X is lying", etc, to complex queries such as "where is point X", "is surface X an island of surface Y", "does line X pass through bodies", etc. We also tested some metric and application queries: "distance between nodes X and Y", "give all point features within 10 units of line feature X", "show outline of building X", "show rooms touching room X by walls/edges/corner", etc. These and many more queries were answered correctly by TREVIS, the response times ranging from a few seconds to a few tens of seconds. The longer response times were encountered when searching through many tables, as needed for graphic presentation of the result. An example of a query result is shown in figure 4.

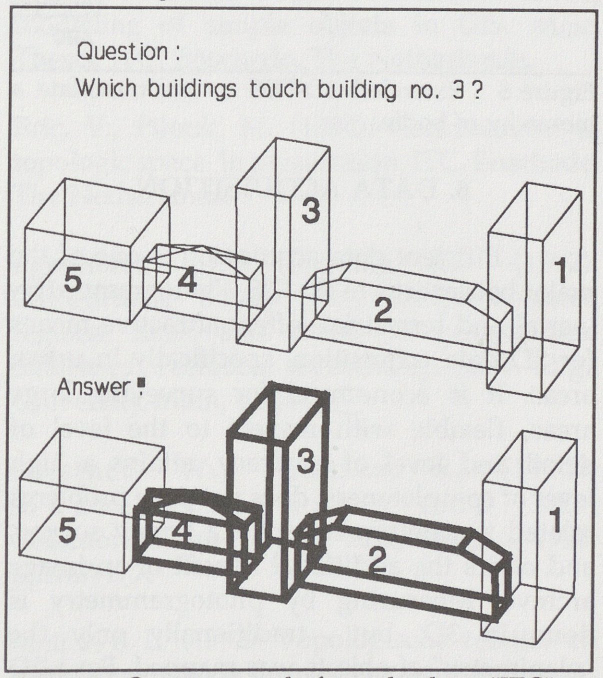

Figure 4 : Query example from database "ITC"

The second database, also created from artificial data, had to accommodate a hierarchy of bodies, *ie*, buildings, floors, rooms. To achieve this, the original 3D-FDS was extended to include higher thematic levels (see figure 5). The rooms were defined as body features; for the floors and buildings, new tables FLOR and BUIL were constructed. To avoid redundancies in FLOR and BUIL, the tables FLORDES (floor description) and BUILDES (building description) were introduced.

The efficiency of querying has not yet been tested against that of an alternative extension of the model, *ie*, adding attributes in BODYOBJ instead of introducing more tables.

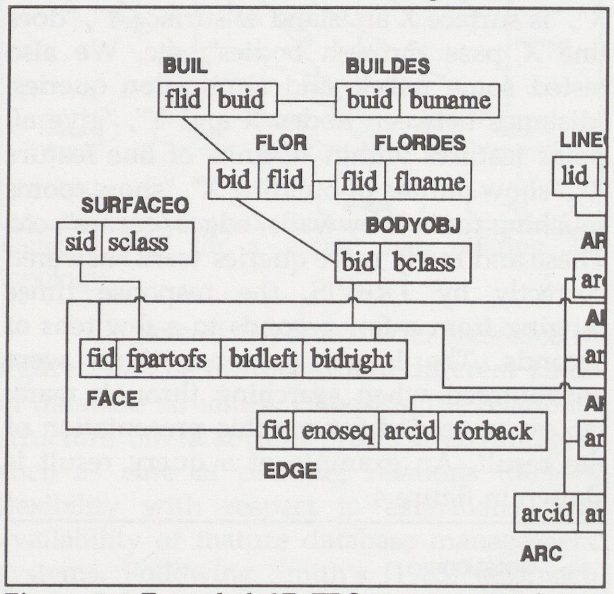

Figure 5 : Extended 3D-FDS to accommodate a hierarchy of bodies

6. DATA ACQUISITION

Efficient data acquisition is one of the major bottlenecks in 3D-GIS. Photogrammetry--aerial and terrestrial--offers attractive means for 3D data acquisition, specifically in urban areas. It is economical for surveying large areas, flexible with respect to the level of detail and level of accuracy, attains a high level of completeness, does not pose problems related to safety, allows for a timely survey, and offers the additional benefit of an image archive. Measuring by photogrammetry is done in 3D, but traditionally only the "planimetry" of objects was mapped. For a 2D description of a building, it was sufficient to measure the outline of the roof and project it orthogonally to a reference plane (datum). For a 3D description, however, we need explicitly all faces of a house, *ie*, its roof facets, its walls, its footprint, and these in terms of geometry and topology. This implies considerably more work in measuring and/or building topology. In view of the enormous demand for up-to-date data, automation of data acquisition will be decisive for the broad introduction of 3D-GISs. An account of what is already possible today in terms of (semi-) automation was given by, for example, Förstner and Pallaske (1993). Our present interest is on finding the best strategy for collecting and structuring house data with means available also outside laboratories. We are testing two approaches: (a) a "measure it all" procedure and automated conversion from 3D spaghetti to 3D-FDS and (b) measure in a structured way to reduce the measuring effort and develop a tailored procedure for filling the database tables.

Procedure (a) is of interest in view of a general input function to TREVIS. Many existing 3D data could easily be converted to a spaghetti structure where every face of an object is given by all its edges. Also semi-automatic extraction of houses from images would deliver closed polygons for detectable faces. Subsequent manually building of topology would be too cumbersome for large data sets, but if automated it would serve many potential data sources for a 3D-FDS.

In procedure (b), not all edges of every face are measured, but only all roof edges only (without duplication). This can be done on any photogrammetric plotter with 3D digitizing software, using aerial photographs. Constructing the walls, footprint of a building and non-visible corners is done computationally, at least for simple objects. To this end we need a DTM, produced either automatically (by a digital plotter, *eg*, Traster T10) or semi-automatically (by an analytical plotter, *eg*, DSR1 and COPS, (see Tempfli, 1986)). The vertices of a polygon outlining a roof are projected orthogonally onto the DTM, thus producing the footprint and the walls of a building. For the time being we assume vertical walls (and rectangular buildings in case of incomplete roof outlines) and also use AutoCAD for checking and editing if necessary. Surface, line and point features such as streets and their furniture can be digitized by known manual procedures. The surface model is exported from AutoCAD using the 3DFACE command to produce the NODE, ARC, and EDGE tables. The last step is tracing the faces and filling the FACE and BODYOBJ tables. This general procedure allows for different variants and must include consistency checks. The experiment following procedure (b) is being applied to the city of Lublijana, Slovenia. The phase of data acquisition is complete. The second phase, to convert these data into 3D-FDS, still has to be done.

7. EXPERIENCE AND OUTLOOK

Developing 3D vector GISs is still in its infancy. The transition from 2D to 3D considerably increases complexity, effort of data collection and data structuring and data volumes. The 3D-FDS offers attractive possibilities of spatial analysis for applications where the real world can be described by solid objects, surface, line and point features. The data structure can be simplified, depending on the intended application, eg, if only body features are needed. The 3D-FDS supports a wide range of queries, from simple class retrieval to metric queries and complex topologic ones. A thorough analysis of the query space would still have to be done.

Photogrammetry can deliver 3D vector data. Manual extraction of houses does not pose much problem, but time efficiency does. Until further progress is made in semi-automatic extraction, strict digitizing rules should be followed to minimize measuring, especially in a case of complex buildings and when a high level of detail is required. Even more cumbersome, unless automated, is the subsequent step, *ie*, detecting gaps and completing the body features, while taking into account measuring inaccuracies, and structuring the data into the tables NODE, ARC, etc. In 3D, automating the building of topology is significantly more demanding than in 2D. We have not yet looked into updating of 3D databases in all aspects; obviously it will require comprehensive consistency checks before adding or deleting data. Other issues of practical interest are conversion algorithms from other 3D representations to 3D-FDS (in particular CAD to 3D-GIS), how to incorporate existing 2.5D databases and digitized floor plans in TREVIS.

Who would not like to view the constructed model of the real world in terms of virtual reality? A little less sophistication, however, may also suffice; because of display speed, the user should be able to change the level of realism, thus applying different rendering procedures when needed. The anaglyph display combined with the stereo cursor can satisfy minimum requirements, but as soon as the objects become complex, the wireframe presentation can easily cause confusion. Polarization for stereo viewing would increase the costs, but eliminate the "ghosting effect" and allow the use of colours. Semi-transparent walls would make pointing easier. In contrast to 2D computer graphics, where a wealth of tools is available, the 3D developer has to rely much more on his own capabilities. An open digital photogrammetric system would be an excellent platform for developing and running a 3D vector GIS.

8. REFERENCES

Bric, V. (1993): 3D vector data structures and modelling of simple objects in GIS. M.Sc. Thesis, ITC, Enschede, The Netherlands.

Bric, V., Pilouk, M. (1994): Computation of topologic space. In preparation, ITC, Enschede, The Netherlands.

Egenhofer, M.J., Herring J.R. (1992): Categorizing binary relationships between regions, lines, and points in geographical databases. Technical report, Dep. Surv. Eng., University Main, USA

Förstner, W., Pallaske, R. (1993): Mustererkennung und 3D-Geoinformationssysteme. ZPF, 61.Jg., 5/1993, pp.167-177.

Meij v/d L. (1992): Topologische relaties en bevragingen in de formele datastructuur voor drie-dimensionele vectorkaarten.
Scriptie, LU Wageningen, The Netherlands

Molenaar, M. (1988): Single valued polygon maps, IAPRS, Vol.27, Part B4, Kyoto, pp.592-601.

Molenaar, M. (1992): A topology for 3D vector maps. ITC Journal, 1992-1, pp.25-33.

Petchenik, B.B. (1991): New directions for national mapping. URISA, Vol.3, No.1, pp.77-79.

Rhind, D. (1993): Policy on the supply and availability of Ordnance Survey information over the next 5 years. Mapping Awareness and GIS in Europe, Vol. 7, No.1

Pilouk, M., Tempfli, K. (1993): An integrated DTM-GIS data structures: a relational approach, AUTO CARTO 11, pp. 278-287.

Smith, H.C. (1985) : Data base design: Composing fully normalized tables from a rigorous dependency diagram, Communication of the ACM, Vol.28, No. 8, pp.826-838.

Tempfli, K., (1986) : Composit/progressive sampling - a program package for computer supported collection of DTM data, ACSM-ASPRS Annual Convention, Washington D.C.