THE IMPLEMENTATION OF OBJECT-ORIENTED TIN-BASED SUBSYSTEMS FOR GIS

Alias Abdul-Rahman and **Jane E. Drummond**
Dept. of Geography and Topographic Science
University of Glasgow
Glasgow, G12 8QQ
Scotland, U.K

alias@geog.gla.ac.uk
jdrummond@geog.gla.ac.uk

Working Group IV

KEYWORDS: TIN, Spatial Data Modelling, Object-Oriented, and GIS

ABSTRACT

This paper focuses on the development of an object-oriented Triangular Irregular Networks (TINs) subsystems for GIS. It reviews the current development of 3D GIS which includes a discussion of the needs, demands, and related modelling and structures for a system. In the third section of the paper, we discuss the concept of object-oriented modelling and specific modelling for the TIN approach. The TIN structures are based on Delaunay triangulation which utilises the concepts of digital image transformation and Voronoi diagrams. The generated structures are used as the core data structures in the spatial modelling. The relevant aspects of TIN-based object-oriented modelling are introduced in section 4. Section five presents the implementation of the above concepts of the TIN-based spatial data in subsystems for the proposed object-oriented system in which a commercial OODBMS package is utilised. The subsystems are tested using photogrammetrically digitized datasets and presented visually by the display interface that we developed (called TINSoft). We also developed some GIS applications, e.g. contouring. Finally, we discuss some challenges and improvements needed for this kind of GIS system.

1. INTRODUCTION

The object-oriented (OO) approach is now being utilised in various fields including GIS. In GIS, OO techniques are used for various tasks including spatial data modelling, databasing, and system development. A research trend involving a shift from structural to OO techniques is very much in evidence in the GIS community. This trend is mainly due to the strength of the OO approach over the traditional structural technique of programming and system development (Egenhofer and Frank, 1989; and Worboys, 1995), that is OO modelling only requires representation much closer to the real world than are needed in the more abstract structured approach. Complex spatial data handling as required in GIS also contributes to this paradigm shift (Webster, 1990) as traditional relational databases could not handle efficiently such complex spatial data. It is the subject of this paper to further investigate the use of an OO system where TIN spatial data is the data structure of concern. Other work carried out by the first author is also investigating the use of TEN (tetrahedral networks) data structures.

This paper discusses some 3D GIS issues, the needs of the system, and the related problems of modelling and data structuring in section 2. OO conceptual modelling is discussed in section 3. We discuss TIN spatial data modelling in section 4, and the TIN-based subsystems for an information system in section 5. Finally, we present suggestions to improve the proposed subsystems in the summary.

2. 3D GIS

In this section we review and discuss some problems and related issues in 3D GIS software development. In GIS, 2D systems are common, widely used, and able to handle most of the GIS tasks efficiently. The same kind of system may not be able to handle 3D data as more advanced 3D applications are demanded (Raper and Kelk, 1991; Li, 1994) such as representing the full length and width of a borehole. Some examples of 3D applications areas are listed in section 2.1. 3D GIS is very much needed to generate information from such 3D data. The system is not a simple extension of another dimension (i.e. the 3rd dimension) on to 2D GIS. To add this third dimension into existing 2D GIS needs a thorough investigation of many aspects of GIS including a different concept of modelling, representation, and aspects of data structuring. Existing GIS packages are widely used and understood for handling, storing, manipulating, and analysing 2D spatial data. Their capability and performance for 2D and for 2.5D data (DTM) is generally accepted by the GIS community. Should we consider a GIS package which can handle and manipulate 2D data and DTM data as a 3D GIS system? The answer is no because DTM data is not real 3D spatial data. The third dimension of the DTM data only provides a surface attribute to the planimetric data of x, y coordinates. In fact, we hardly find any current GIS software to be in a position to handle real 3D spatial data. Although the problem has been addressed by several researchers such as Raper and Kelk (1991), Cambray (1993), Li (1994), Pilouk (1996), and Fritsch (1996), some further aspects particularly spatial data modelling using OO techniques need to be investigated. This modelling issue is addressed in this paper, see section 3 and section 4. The demand for this kind of system is discussed in the next section.

2.1 Who needs 3D GIS

As in the popular 2D GIS for 2D spatial data, 3D GIS is for managing 3D spatial data. Raper and Kelk (1991), Li (1994), Förstner (1995), and Bonham-Carter (1996) described some of the three dimensional applications in GIS, including:

- ecological studies
- environmental monitoring
- geological analysis
- civil engineering
- mining exploration
- architecture
- automatic vehicle navigation
- archeology
- 3D urban mapping
- landscape planning

The above applications may produce much better information if they were handled in a 3D spatial system. It appears that complex 3D spatial objects on the surface and subsurface demand better solutions (e.g. in terms of modelling, analysis, and visualization) than the existing systems can offer.

The next section reviews the modelling and data structures contributing to 3D GIS.

2.2 Modelling and Data Structuring

Much previous work done on 3D data modelling concentrated on the use of voxel data structures (Jones, 1989). This particular approach does not address spatial modelling aspects, it is only useful for the reconstruction of 3D solid objects and for some basic geometric computations. One of the problems with this model is that it needs very large computer space and memory.

Carlson (1987) proposed a model called simplicial complex. He used the term 0-simplex, 1-simplex, 2-simplex, and 3-simplex to denominate spatial objects of node, line, surface, and volume. His model can be extended to *n*-dimensions.

While Cambray (1993) proposed CAD models for 3D objects combined with DTM as away to create 3D GIS, that is a combination of Constructive Solid Geometry (CSG) and Boundary representation (B-rep).

Other attempts to develop 3D GIS can be found in Kraus (1995), Fritsch and Schmidt (1995), and Pilouk (1996). These attempts were based on the TIN data structure to represent 3D terrain objects but no reports exist on the any related aspects of OO technique for the modelling and data structure.

Data modelling and structuring of 3D spatial objects in GIS has not been as successfully achieved as in CAD (Li, 1994). Data modelling in GIS is not only concerned with the geometric and attribute aspects of the data, but also the topological relationships of the data. Topology of spatial data must be available so that the neighbouring objects can be determined. There are a number of mathematical possibilities for the determination of the topological description of objects. Within the TIN data structure, we have developed a program to determine the neighbouring triangles. The information gained from the generated TIN's neighbours is useful for further spatial analysis and applications. We also established topological relationships for linear objects as represented by TIN edges. One edge is represented by a start node and an end node. From this edge topology, a chain of edges or arcs could be easily established. For TIN data, another approach is the simplicial complex developed by Carlson. A TIN's node is equivalent to 0-simplex, TIN's edge is equivalent to 1-simplex, a TIN surface (area) is equal to 2-simplex, and 3-simplex is equivalent to a 3D TIN (tetrahedron). The simplicial complex technique checks the consistency of generated TIN structures by Euler's equality formulae, see Carlson (1987) for a detailed discussion. We explain our OO TIN approach in section 4 after elaborating OO conceptual modelling.

3. OBJECT-ORIENTED CONCEPTUAL MODELLING

Object-oriented conceptual modelling is now widely utilised in many fields including GIS. The concepts of OO such as object classification, encapsulation, inheritance, and polymorphism are able to ease the modelling of complex real world objects.

3.1 Object-Oriented Concepts

As mentioned above the object-oriented approach is now being promoted as the most appropriate method for modelling complex situations that are concerned with real-world phenomena, and thus applicable to GIS. Object-oriented concepts are considered more flexible and powerful than the traditional structural programming and other major database models such as the relational or entity-relationship model. Object-oriented concepts contribute to modelling as follows:

(a) Objects and abstraction mechanisms (classification, generation, aggregation, and association). These aspects of OO can be used for modelling real world phenomena, e.g. modelling of spatial data for geoinformation systems.
(b) Inheritance, propagation, encapsulation, persistence, Abstract Data Type (ADT), polymorphism, and overloading. These aspects of OO can be used to construct and implement the model discussed in (a).

The usefulness of these concepts in spatial modelling are explained below.

3.2 The Abstraction Mechanisms

Data abstraction is a method of modelling data. Object-oriented design uses four major abstraction mechanisms: (1) classification, (2) generalization, (3) inheritance, and (4) polymorphism. In object-oriented programming, any physical or logical entity in the model is an "object". The definition of a type of object is called a "class", and each particular object of that type known as an "instance" of the class. Once a class has been defined, it can, potentially be reused in other programs by simply including the class definition in the new program. However, it is not necessary for the programmer who uses a class to know how it works, they simply need to know how to use it. The definition of operations on or between objects are called "methods", and the invocation of methods is referred to as "passing a message". Recent research in software engineering has promoted an object-oriented design method by which real world objects and their relevant operations are modelled in a program which is more flexible and better suited to describe complex real world situations (Khoshafian and Abnous, 1995). We may also consider object-orientation as a particular view of the world which attempts to model reality as closely as possible (Webster, 1990). Details on all relevant OO concepts (object, abstraction, data types, class hierarchy, inheritance, classification, aggregation, generalization and association) can be found in the OO literature such as Booch (1990), Bhalla (1991), and Stroustrup (1997). The following are some OO terms:

Classification
Classification can be expressed as the mapping of several objects (instances) onto a common class. In object-oriented approach, every object is an instance of a class (a class is a fundamental building block in OO language). Class describes the common features of a set of objects with the same characteristics; it also defines the nature of the state and behaviour, while an object records the identity and state of one particular instance of a class. Abstract Data Type (ADT) is a mechanism to create a class of spatial objects or any class in a domain of objects. An object is a basic run-time entity in an object-oriented system. This entity includes data and procedures that operate on data. Viewed from a programming stand point, objects are the elements of an OO programming system sending and receiving messages.

Generalization
Generalization in OO provides for the grouping of classes of objects, which have some operations in common, into a more general superclass. Objects of superclass and subclass are related by an *"is a"*- relation, since the object of a subclass is also an instance of a superclass.

Inheritance
Inheritance allows the building a hierarchy of types or classes that best describes the real world situation in the application field. Each class can take all or part of the structural or behavioural features from other classes, which are its parents. In turn, the newly defined class is a child of the classes from which it has inherited its features. Inheritance helps in deriving application-oriented classes without starting every definition from scratch. Also, it makes it easier to create logically complex classes from simpler classes.

Polymorphism
Polymorphism is a mechanism to define the different actions of the same named function on different classes. It is implemented by inheriting some functions from parent classes and overriding or modifying part of them. Usually, the newly created class has similar but not the same behaviour as its parents for that functional aspect. Polymorphism provides great flexibility in class derivation, for example, perimeter operation may have different implementations for different classes such class "area", class "triangle", class "polygon", etc. Each class performs the perimeter operation differently although it has the same function name.

4. OBJECT-ORIENTED TIN SPATIAL DATA MODELLING

In this section we provide a discussion of the OO TINs spatial data modelling techniques. Conceptually, the general modelling steps as depicted in Figure 1 could be used for TIN spatial data modelling. That is, the three-step approach, namely the conceptual, the logical, and the physical steps. The class schema for spatial data modelling are described below.

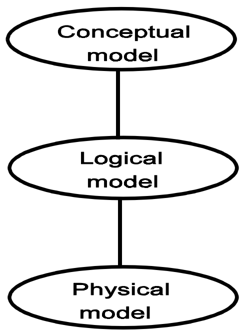

Figure 1 A typical spatial data modelling steps

4.1 The Class Schema

The schema is based on several classes, they are Spatial Objects (the super class), and four major subclasses which are Node, Edge, Polygon, and Solid.

Spatial objects
The spatial object class is a general class of the real world objects. It is the super class in the class hierarchy. We assume that all other objects are derived from this super class, see Figure 3. All terrain objects could be categorised into several sub classes such as points, lines, areas, and solids (volume) features. In OO modelling, these feature types are the classes in the modelling hierarchy.

Node
A node can be considered as the most basic geometrical unit in spatial data modelling. It may represent point entities or point objects at a particular mapping scale. Examples of point objects are wells, terrain spot heights, and the like. In geoinformation, we may represent these objects by a class called a node class. The coordinates of the nodes (including the nodes represent edges) are held by a coordinates container class, called XYZContainer class.

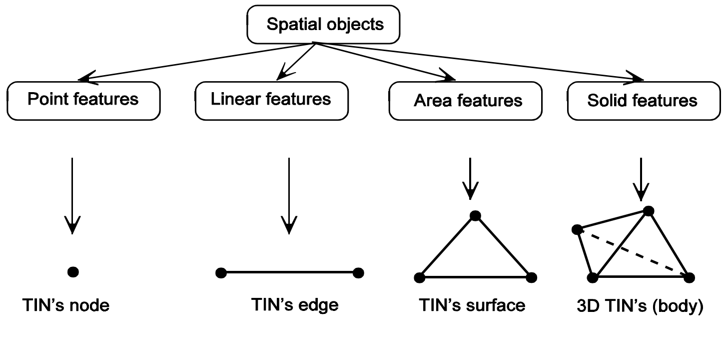

Figure 2 TINs representations for spatial objects

Edge
An edge can be represented by two nodes at each end (i.e., a start node and end node). In this study we consider two end points make a straight edge. We used this edge type to represent linear features. The arc container class, called ARCContainer holds all the

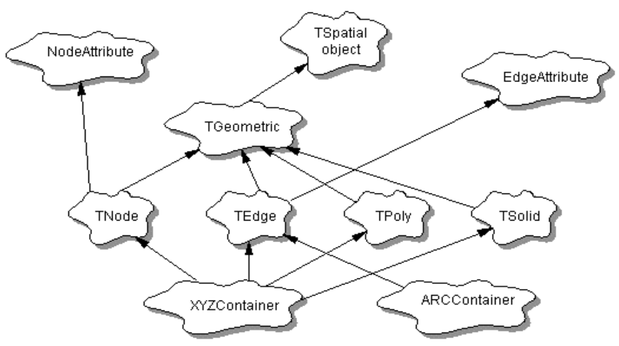

Figure 3 The class diagram (using the Booch notation)

arcs. The arcs container also serve any other class which requires arcs data in their operations for example the polygon class needs the arcs in order to form polygons.

Polygon
A polygon (sometimes known as a surface) is used to represent area features such as lakes, ponds, etc. A polygon may be constructed by chains of closed edges.

Solid (or Body)
This is a representation for solid or body features such as buildings, trees. A chain of points and lines form body objects for example, 3D TIN can be represented by a series of triangle nodes and edges as indicated in Figure 2.

The class schema in Figure 3, depicted using Booch (1990) notation is the representation of the TIN spatial data model. The schema has: four geometric classes namely (TNode, TEdge, TPoly, and Tsolid); two types of containers (geometry and attribute). The geometric containers contain the XYZ locations whereas the attribute containers are for the thematic values, e.g. names.

4.2 The POET OO Database Development

The DBMS is used to generate the OO database from the constructed TIN spatial data. In this work the schema needs to be modelled according to the POET database model (POET, 1996), that is it is required to construct all the C++ classes as classes which POET can understand. In this case, all the classes in the schema have to be compiled by the POET PTXX compiler. The PTXX compiler maps all the normal C++ classes into the several relevant PTXX schema files which in turn are used for writing application programs (runs under normal C++ compiler) as well as for populating the database. The PTXX complier also generates the OO database from the schema, see Figure 4. For database query, the OQL (Object Query Language) syntax is used, see POET (1996) for the details of the language. An example of a query which can be performed from the database is:

```
defined extent allTEdge for TEdge;
select Edge
from Edge in allTEdge
where Edge.EdgeAtr.EdgeName = “River*”
```

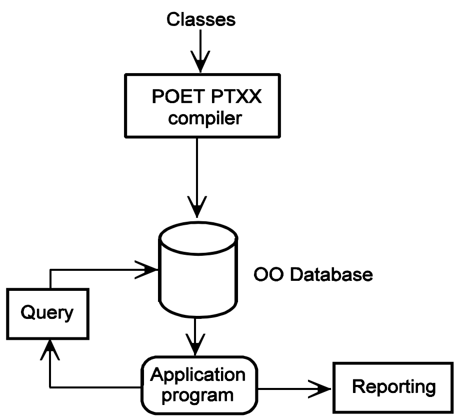

Figure 4 The POET database development flows

5. OBJECT-ORIENTED TIN-BASED SUBSYSTEMS FOR GIS

5.1 The Subsystems

The OO TIN GIS is based on several fundamental concepts and aspects of spatial data which have been discussed in the previous sections. Basic components in the system are data input processing, TIN data construction, TINs database, transformation operations, data output, and user-interface. Rasterization forms a major operation in the data input component. Figure 5 shows the other major components of the proposed system

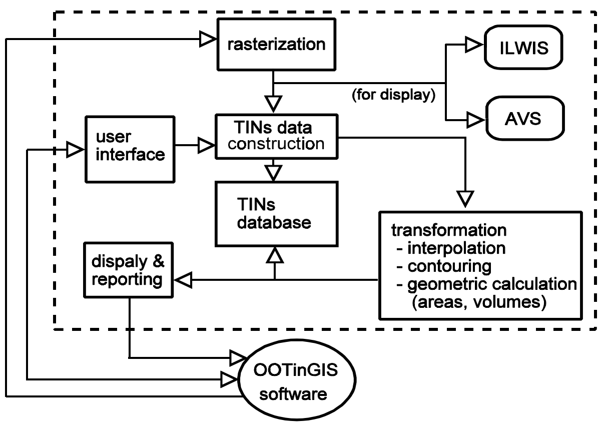

Figure 5 The proposed system for the TIN-based spatial data

which includes the use of other commercial software, i.e. ILWIS™ (Integrated Land and Water Information System) and AVS™ (Advanced Visualization System). These two packages are only for display purposes especially for validating the output from the rasterization process. We also developed a simple user interface as part of the software development. Besides our own written programs for databasing purposes, we also used a commercial database package, called POET™ OODBMS as mentioned. The DBMS package is for the development of the OO TIN spatial database.

5.2 The Test and Results

We tested the subsystems (or their components) by using photogrammetrically digitized datasets (other types of datasets such as data acquired by field survey, or by map digitizing are also possible as input to the subsystems). Figure 6 shows the study area (Drumbuie, near Kyle of Lochash, north-west Scotland). Other diagrams visually illustrate some of the output from the subsystems.

Figure 6 The study area (orthophoto image) from where points and lines were extracted using 3D digitizing (with stereo mate)

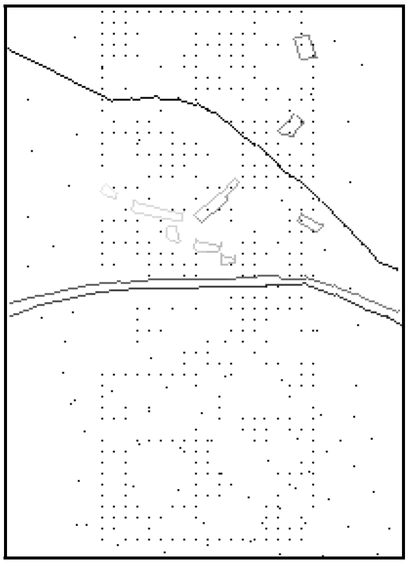

Figure 7 The rasterised points and lines of selected features

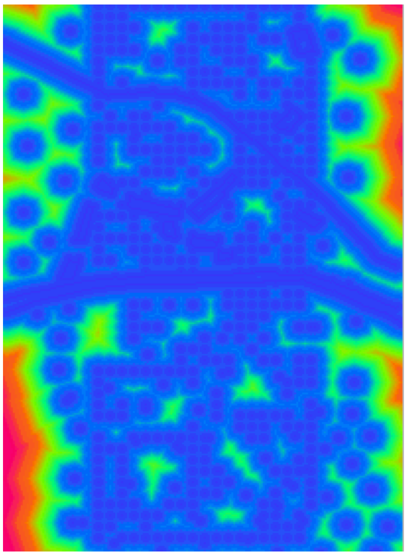

Figure 8 The distance transform (DT) image of the area

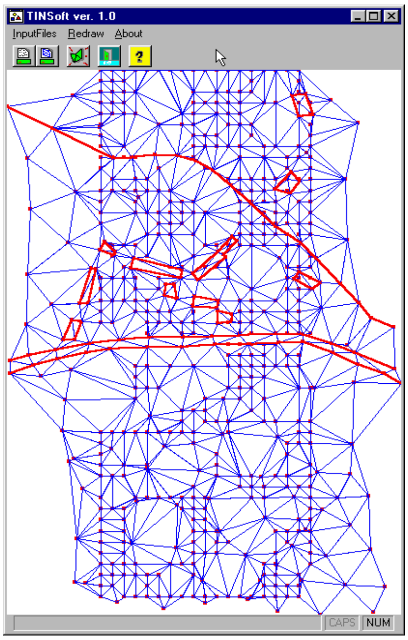

Figure 9 The generated TINs incorporated with constrained edges.

Figure 7 shows the output from the rasterization component. Figure 8 is the intermediate process of 2D TIN data construction component, that is the distance transformation computation output. The generated TINs are in Figure 9 whereas Figure 10, the simple display interface, illustrates the perspective view of the 3D objects such as buildings and trees sitting on top of the TIN surfaces. Contours of the terrain surfaces could also be derived from the TIN database as clearly draped on the terrain surface.

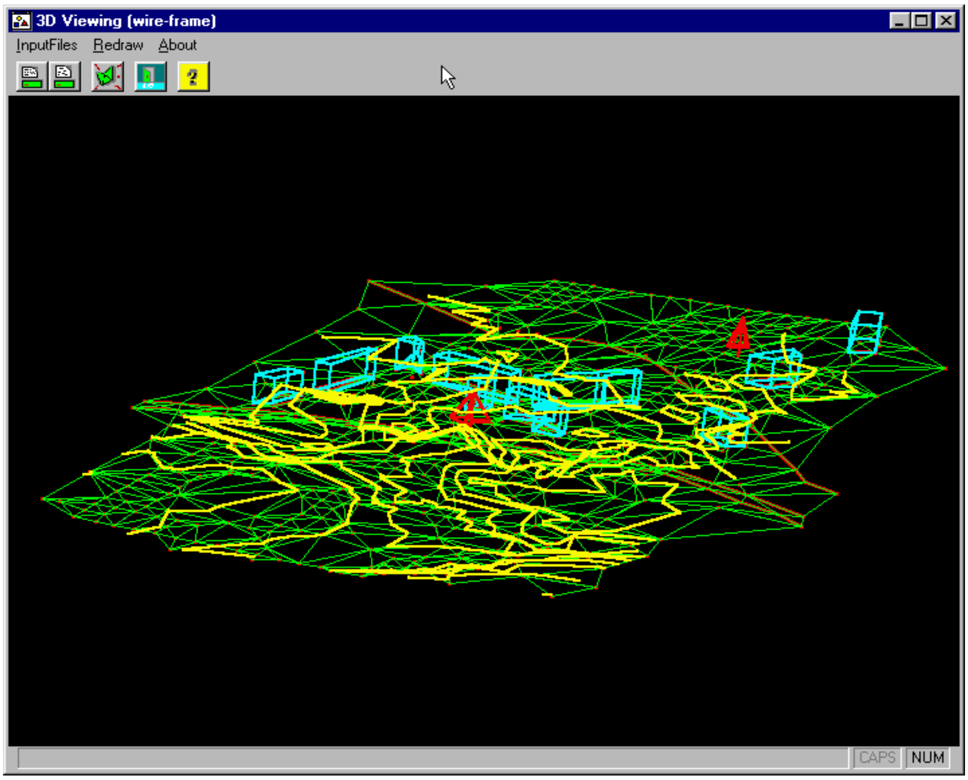

Figure 10 Perspective view of 3D objects (buildings) and trees draped on TINs surfaces with the derived contours (4-metre intervals).

6. SUMMARY

We have introduced several subsystems for OO TIN-based GIS. The generated products (i.e. the output) from the subsystems indicate the workability of these components. Although the system is not yet complete as one whole GIS system, but the results produced from the components are promising. An integration of the subsystems with the commercial OODBMS needs to be investigated so that seamless integration could be achieved. Thus, from the perspective of GIS software development, the object-oriented TIN data modelling can be used to develop a geo information system.

ACKNOWLEDGEMENTS

Thanks are due to the following organisations which have contributed one way or another in our research work, they are LH System Inc. (Helava-Leica) for the digital photogrammetric software, Advanced Visualization Systems Inc., (AVS) for 3D raster data visualization package, and POET Inc. The generous assistance from Dr. Ron Poet of Glasgow University's Department of Computer Science and Mr. Prasad Jeevanigi on POET's compiler error messages must also be acknowledged.

REFERENCES

Bhala, N., 1991. Object-oriented data models: a perspective and comparative review. Journal of Information Science, Vol. 17, pp. 145-160.

Bonham-Carter, G. F., 1996. Geographic information systems for geoscientists: modelling with GIS. Computer Methods in the Geosciences*,* Vol. 13, Pergamon Publications, 398 p.

Booch, G., 1990. Object oriented analysis and design with applications. 2nd. Edition, Addison-Wesley, 589 p.

Bric, V., 1993. 3D vector data structures and modelling of simple objects in GIS. M. Sc. Thesis, ITC, Enschede, The Netherlands, 107 p.

Cambray, de B., 1993. Three-dimensional (3D) modelling in a geographical database. Proceedings of AutoCarto 11, Minneapolis, USA, pp. 338-347.

Egenhofer, M. and Frank, A. U., 1989. Object-oriented modelling in GIS: inheritance and propagation. Proceedings of AutoCarto 9, Baltimore, pp. 588-598.

Förstner, W., 1995. GIS - the third dimension. Workshop on Current Status and Challenges of Geoinformation Systems, IUSM working group on LIS/GIS, University of Hannover, Germany, pp. 65-72.

Fritsch, D., 1996. Three-dimensional geographic information systems - status and prospects. International Archives of Photogrammetry and Remote Sensing (ISPRS), Vienna, Austria, Vol. 31, Part 4, pp. 215-221.

Fritsch, D. and Schmidt, D., 1995. The object-oriented DTM in GIS. Proceeding of 45th. Photogrammetric Week, Stuttgart, pp. 29-34.

Jones, C. B., 1989. Data structures for three-dimensional spatial information systems in geology. International Journal of Geographic Information System (IJGIS), Vol. 1, no. 3, pp. 15-31

Khoshafian, S., and Abnous, R., 1995. Object orientation : concepts , analysis, languages, databases, graphical user interfaces, standards, Second Edition, John Wiley, 504 p.

Kraus, K., 1995. From digital elevation model to topographic information system. Proceeding of 45th.Photogrammetric Week, D. Fritsch and D. Hubbie (eds.), Stuttgart, Germany, pp. 277-285.

Li, R., 1994. Data structures and application issues in 3D geographic information systems. Geomatica, Vol. 48, No. 3, pp. 209-224.

Pilouk, M., 1996. Integrated modelling for 3D GIS. International Institute of Aerospace Survey and Earth Sciences (ITC), Publication No. 40, Enschede, The Netherlands, 200 p.

POET , 1996. POET™ C++ programmer's guide. Release 4, 560 p.
http://www.poet.com

Raper, J. and Kelk, B., 1991. Three-dimensional GIS. In: Geographical Information Systems: Principles and Applications*.* D. J., Maguire, M. Goodchild and D. Rhind (edts.), Longman Geoinformation, pp. 219-317.

Stroustrup, B., 1997. The C++ programming language, 3rd. Edition. Addison-Wesley, 910 p.

Webster, C., 1990. The object-oriented paradigm in GIS. International Archive of Photogrammetry and Remote Sensing (IAPRS), Vol. 28, Part 3/2, Comm. III, Wuhan, China.

Worboys, M. F., 1995. GIS: a computing perspective. Taylor and Francis Publication, 376 p.