Integrated Modelling for 3D GIS

Morakot Pilouk

ITC

Publication
Number 40

Promotor:	Dr. Ir. M. Molenaar Hoogleraar in de Theorie van de Geografische Informatie Systemen en de Remote Sensing
Co-promotor:	Dipl.-Ing. Dr. K. Tempfli Associate Professor International Institute for Aerospace Survey and Earth Sciences (ITC), Enschede

Morakot Pilouk

Integrated Modelling for 3D GIS

Proefschrift
ter verkrijging van de graad van doctor
in de landbouw- en milieuwetenschappen
op gezag van de rector magnificus,
Dr. C.M. Karssen,
in het openbaar te verdedigen
op maandag 24 juni 1996
des namiddags te vier uur in de Aula
van de Landbouwuniversiteit te Wageningen

ITC Publication Series
Nr. 40
The research presented in this thesis
was performed at
International Institute for Aerospace Survey and Earth Sciences (ITC)
PO Box 6, 7500 AA Enschede, The Netherlands

The examining committee:
Prof. Dr. Ing W. Förstner, the Institute for Photogrammetry, University of Bonn, Germany
Prof. Ir. A. Beulens, Department of Computer Science, WAU, The Netherlands
Dr. P.J.M. van Oosterom, Cadastre, Apeldoorn, The Netherlands
Prof. Dr. J. Grasman, Department of Mathematics, WAU, The Netherlands
Prof. Dr. W. Kainz, Department of Geoinformatics, ITC, The Netherlands

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Integrated modelling for 3D GIS
Thesis Wageningen Agricultural University and ITC,
with index, ref. Summary in Dutch
ISBN 90 6164 122 5

Subject headings: 3D geographic information system, geometric data integration, spatial modelling, irregular tessellation, simplicial network.

Morakot Pilouk

Integrated Modelling for 3D GIS

Thesis
to fulfil the requirements for the degree of doctor
in the Agricultural and Environmental Sciences
on the authority of the rector magnificus,
Dr. C.M. Karssen,
to be publicly defended
on Monday 24th June 1996
at 16.00 hours in the Auditorium
of Wageningen Agricultural University

PROPOSITIONS

related to the dissertation

Integrated modelling for 3D GIS

(6) The client/server approach for GIS applications can be a good solution for providing access to all required functions and data when they are distributed over different independent subsystems. Once the required functions are available in a single GIS, the client/server architecture can be simplified to serve for only distribution of data.

(7) GIS is a discipline with a very large scope. Since the discipline is relatively new, it is still lacking a unifying theory that interrelates the many different aspects. This makes 'navigation' the very first subject in GIS, not only concerning how to read maps or to find a route, or a location, but also to move over this ocean of knowledge more peacefully. A unifying theory would provide a guideline for studying and using GIS, that is to say, facilitating navigation in this discipline, with a benefit comparable to introducing GPS in real-world navigation.

(8) At a very primitive (atomic) level, organisms have more common aspects than at a higher level of complexity. Thus to achieve a highly integrative result, we should consider integrating at the level at which things no longer appear fundamentally different.

(9) If we want to be convinced that what we experience through the computer is realistic, we must first ensure that what is stored in it is realistic.

(10) Our abilities cannot be recognized if we lack of ability to show them. Many presentation tools and media are available to us, but these require that we also have the ability to learn to use them and also that we have the ability to go out and present what has been done. When developing all these abilities, we might lose sight of the abilities we originally wanted to demonstrate.

(11) It is important that each subordinate recognizes the boss. But it is more important that the boss recognizes the importance of his subordinates.

(12) We cannot make an engine run by simply removing a gear that refuses to rotate. In most cases, just a little lubrication helps that gear to run better. If the engine does not run because two gears are in conflict, removal of either gear would not make the engine run either. A slight adjustment would help in this case. In both instances, a simple assessment of the solution still requires an expert. A non-expert solution, such as simply replacing the entire engine, is likely to be both expensive and wasteful.

Morakot Pilouk
24 June 1996

Enschede
The Netherlands

PROPOSITIONS

related to the dissertation

Integrated modelling for 3D GIS

(1) A 3D spatial model that supports high-quality representation of real-world objects and the (spatial) relationships among them is needed so that object manipulation in the database becomes highly comparable to the manipulation of real-world objects. The richness of operations, which reflects the functionality of the system, depends on the complexity of the spatial model.
• *This thesis*

(2) The integrated 3D spatial data model must be able to:
- accommodate objects of various dimensions, especially ranging from 0D to 3D
- maintain relationships among data elements and the topological descriptions between simplices and complexes
- support the modelling of objects with determinate and indeterminate spatial extent.
- permit the modelling of objects with spatial coincidence (multi-theme)
- expand to accommodate objects of more dimensions
- permit derivation of a unified data structure for the implementation.

A 3D spatial model based on the simplicial network concept meets all of these requirements.
• *This thesis*

(3) Both direct and indirect representations of spatial objects should be possible within a spatial information system. The combination of these representations can be realized in one database if an appropriate data model based on simplicial networks is used.
• *This thesis*

(4) The efficient exploitation of a 3D spatial model, based on the use of simplicial networks, requires the construction of an information system that integrates and adapts various technologies. The difficulty in constructing such a system is still a small price in comparison with the benefits for future users of 3D GIS.

(5) The existing data models in fact represent different views of reality. The database based on an integrated data model can be regarded as the integration of views. As such, an integrated database will contain excessive data for an individual application and, hence, imply longer response times than offered by a dedicated database. A remedy to this disadvantage is to use existing data models to define view-specific spatial index schemes on top of the integrated database. The possibility of defining a view-specific spatial index must also be offered to the user.
• *This thesis*

Morakot Pilouk
24 June 1996

Enschede
The Netherlands

ABSTRACT

Pilouk, M., 1996. Integrated modelling for 3D GIS. PhD Thesis, Department of Geographic Information Processing and Remote Sensing, Wageningen Agricultural University, The Netherlands, 200 pp

A three dimensional (3D) model facilitates the study of the real world objects it represents. A geoinformation system (GIS) should exploit the 3D model in a digital form as a basis for answering questions pertaining to aspects of the real world. With respect to the earth sciences, different kinds of objects of reality can be realized. These objects are components of the reality under study. At the present state-of-the-art, different realizations are usually situated in separate systems or subsystems. This separation results in redundancy and uncertainty when different components sharing some common aspects are combined. Relationships between different kinds of objects, or between components of an object, cannot be represented adequately. This thesis aims at the integration of those components sharing some common aspects in one 3D model. This integration brings related components together, minimizes redundancy and uncertainty. Since the model should permit not only the representation of known aspects of reality, but also the derivation of information from the existing representation, the design of the model is constrained so as to afford these capabilities. The tessellation of space by the network of simplest geometry, the simplicial network, is proposed as a solution. The known aspects of the reality can be embedded in the simplicial network without degrading their quality. The model provides finite spatial units useful for the representation of objects. Relationships between objects can also be expressed through components of these spatial units which at the same time facilitate various computations and the derivation of information implicitly available in the model. Since the simplicial network is based on concepts in geoinformation science and in mathematics, its design can be generalized for n-dimensions. The networks of different dimension are said to be compatible, which enables the incorporation of a simplicial network of a lower dimension into another simplicial network of a higher dimension.

The complexity of the 3D model fulfilling the requirements listed calls for a suitable construction method. The thesis presents a simple way to construct the model. The raster technique is used for the formation of the simplicial network embedding the representation of the known aspects of reality as constraints. The prototype implementation in a software package, ISNAP, demonstrates the simplicial network's construction and use. The simplicial network can facilitate spatial and non spatial queries, computations, and 2D and 3D visualizations. The experimental tests using different kinds of data sets show that the simplicial network can be used to represent real world objects in different dimensionalities. Operations traditionally requiring different systems and spatial models can be carried out in one system using one model as a basis. This possibility makes the GIS more powerful and easy to use.

Keywords: 3D geographic information system, geometric data integration, spatial modelling, irregular tessellation, simplicial network.

SAMENVATTING

Pilouk, M., 1996. Geïntegreerde modellering voor 3D GIS, Proefschrift ter verkrijging van de doctorsgraad, Vakgroep Geografische Informatieverwerking en Remote Sensing, Landbouwuniversiteit Wageningen, Nederland, 200 pag.

Een drie-dimensionaal (3D) model vergemakkelijkt het bestuderen van ruimtelijke objecten. Een geoinformatie-systeem (GIS) kan van een 3D model in digitale vorm gebruik maken om vragen over de werkelijkheid te beantwoorden. Bij de huidige stand van GIS-technologie worden in de aardwetenschappen vaak verschillende objecten in afzonderlijke systemen of sub-systemen weergegeven. Deze scheiding resulteert meestal in overtolligheid en tegenspraken wanneer verschillende componenten van de werkelijkheid, met een aantal gemeenschappelijke aspecten, vervolgens worden gecombineerd. Verhoudingen tussen verschillende objecten of tussen onderdelen van een object kunnen dan niet goed weergegeven worden. Dit proefschrift tracht daarom de integratie van de componenten met een aantal gemeenschappelijke aspecten in een 3D model te verwezenlijken. Deze integratie brengt aan elkaar gerelateerde objecten samen, vermindert overtolligheid en tegenspraken. Aangezien het model niet alleen de weergave van bekende aspecten van de realiteit moet mogelijk maken, maar ook het vervolgens afleiden van informatie uit deze weergave, is het ontwerp van het model op beide processen gericht.

Kort samengevat is het ontwerp gebaseerd op een opdeling van de ruimte door een netwerk van eenvoudige geometrie, namelijk het 'simplicial network' (netwerk van simplices, geometrische basiselementen). Reeds bekende aspecten van de realiteit kunnen vastgelegd worden in het 'simplicial network' zonder aan kwaliteit in te boeten. De eindige ruimtelijke eenheden van het netwerk bevorderen dat objecten door het model getrouw worden weergegeven. Ruimtelijke relaties tussen objecten kunnen gevonden worden via de relaties tussen de geometrische elementen waaruit ze zijn opgebouwd. Tegelijkertijd vergemakkelijken deze bouwstenen het maken van berekeningen en het afleiden van informatie die impliciet in het model aanwezig is. De conceptuele grondslag van 'simplicial networks' laat generalisatie toe naar ruimtes van willekeurige dimensies. De netwerken van diverse dimensies kunnen compatibel gemaakt worden, zodat een 'simplicial network' van een lagere dimensie opgenomen kan worden in dat van een hogere dimensie.

De complexiteit van het 3D model dat aan de bovengenoemde voorwaarden voldoet, vraagt om een passende constructiemethode. Dit proefschrift stelt een simpele manier voor. Een 3D rastertechniek wordt gebruikt voor het genereren van een 'simplicial network', waarbij de geometrie van ruimtelijke objecten als randvoorwaarde wordt gebruikt. De implementatie van het prototype in een software pakket, ISNAP, toont bouw en gebruik van zo'n 'simplicial network'. Het 'simplicial network' vergemakkelijkt ruimtelijke en niet-ruimtelijke 'queries', berekeningen, 2D en 3D visualisaties. De experimentele testen tonen aan dat het 'simplicial network' kan worden gebruikt om ruimtelijke objecten in verschillende dimensies weer te geven. Handelingen die nu nog in de praktijk op het gebruik van verschillende systemen en ruimtelijke modellen gebaseerd zijn, kunnen met de nieuwe benadering worden uitgevoerd in één systeem met gebruik van maar één model als basis. Deze mogelijkheid maakt het GIS krachtiger en tevens gemakkelijker in het gebruik.

Sleutelwoorden: 3D geografisch informatie systeem; geometrische gegevensintegratie; ruimtelijke modellering; onregelmatige opdeling; 'simplicial network'.

ACKNOWLEDGEMENTS

The completion of this thesis is the result of the cooperation between the International Institute for Aerospace Survey and Earth Sciences (ITC) and Wageningen Agricultural University (WAU) and of the generous support of a number of people, the names of all of whom will not be possible to mention.

I would like to express my sincere gratitude to both of my supervisors, Professor Dr. Martien Molenaar, Department of Surveying, Photogrammetry, and Remote Sensing (WAU) and Dr. Klaus Tempfli, Department of Geoinformatics (ITC), who set sail with me and guided me over the ocean of knowledge of Geoinformation science during these three and a half years. Their valuable knowledge in this discipline and close cooperation allowed me to reach the ultimate goal of my studies. I am grateful for having had the opportunity to conduct this research during the on-going development of a geoinformation theory by Prof. Molenaar. It is my great wish to see its completion in the near future.

Many staff members at ITC have contributed to the research in one way or the other. Prof. Ir. Richard Groot occasionally shared with me some of his experiences in geoinformation management. Prof. Dr. Wolfgang Kainz introduced me to the knowledge in spatial mathematics and let me take over some of his lectures. Mr. Christian Paresi provided some literature on system design. Jan Hendrikse, with his valuable knowledge in mathematics, assisted in conducting the proof of generalized Euler's equality. Ard Blenke was a great support in supplying me with computer hardware and software. Marga Koelen always passed information about text books in 3D computer graphics. Saskia Tempelman and the secretariat of the Geoinformatics department were always there when I needed help. Dr. Elizabeth Kosters discussed geological application with me and provided data for testing. Drs. E.S. Bos allowed me to use drawings of the new ITC building (which is about to be completed at the time of writing this) for testing. Dr. M.M. Radwan allowed me to take over part of his lectures. Dr. Edmund J. Sides kindly sent me his thesis, which was very useful. Mr. J. de Ruiter ensured that I received my allowance from the scholarship, provided by ITC and DGIS, through-out the period of my studies. Friendship and encouragement were received from Dr. Theo Bouloucos, Mr. Sokhon Phem, Mr. Rémy Ackermann, Mr. I. de Sousa, Ir. Ben Gorte, Mr. M.C. Ellis.

A valuable thesis about Delaunay network, conducted at the Norwegian Institute of Technology, University of Trondheim, was received from Dr.Ing. Terje Midtbø.

The MSc theses of Vasja Bric, Abbas Radjabi Fard and Wang Zhi Jun were related to the scope of this thesis. A case study on data structuring and 3D visualization conducted by Siyka Zlatanova was informative. There was the collaboration with PhD colleagues: Wanning Peng on developing the program ISNAP, Olajide Kufoniyi on developing a multi theme variant of the data model, Yasir Bishr for interesting discussions about federated database.

The following people were involved in the final stage of the thesis. Drs. Wan Bakx helped in the production of the colour pages. Dr. Anne Hawkins kindly edited this thesis in a significantly short time. Great contribution was received from Ann Stewart in making funds available towards publication of the thesis. Much help was received from Anneke Homan in finalizing production of the thesis.

Thanks are also due to Professor Dr. K.J. Beek, the rector of ITC, Dr. N. Rengers, the vice rector, for their strong personal involvement over the last few years toward regularizing and standardizing conditions for PhD research at ITC. To this process, much time and effort was also dedicated by Zoltán Vekerdy, Tomaso Ceccarelli, Christine Pohl, Charles Amuyunzu, my former colleagues in the group of PhD representatives. It has been a great experience for me to share their enthusiasm.

Encouragement and some financial support came from my home country: from my parents, Serm and Chongdee. During this study period, my sister and brother, Somchint and Pongsathorn, had carried out for me various matters in Thailand.

Last but not least, I could never complete this thesis without the greatest support, love and patience of my wife, Pakrairat, and the inspiration of my children, Pakawat and Patriya. I am in great debt to them in taking a lot of time from the family to concentrate on these studies during these years.

24 June 1996 Morakot Pilouk

CONTENTS

I seem to have been only a boy playing on the seashore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me.
Sir Isaac Newton (1642-1727)

We believe that if men have the talent to invent new machines that put men out of work, they have the talent to put those men back to work.
John F. Kennedy (1917-1963)

With malice towards none; with charity for all; with firmness in the right, as God gives us to see the right - let us strive on to finish the work we are in.
Abraham Lincoln (1809-1865)

1 INTRODUCTION

Exploiting digital computing technology to improve the quality of life, or prevent or mitigate hazards or disasters, first requires the construction of a model in digital form of the part of the earth and its environment concerned. Such a model, a simplified description of complex reality, can conveniently be used, stored, managed, maintained, distributed, and transported. Even a complex model may be stored on a small scale, on diskettes, tape cartridge or CD ROM, or transmitted via communication networks. A digital model contains spatial and non spatial aspects of reality and provides a basis for operation and communication among the interested parties. A model distinguishes objects. An object, or a set of objects, comprises the elements of reality under investigation. Spatial aspects are those related to shape, size and location. They pertain to geometric properties. Non spatial aspects include name, colour, function, price, ownership, and so forth, often referred to as thematic properties. Spatial aspects of reality can be well and economically represented in the form of graphics, whereas non spatial aspects, in many cases, can better be represented in text. Graphic representation facilitates rapid understanding of the situation in reality, permitting high level abstraction or description about neighbouring relationships, while the textual representation is more suitable for aspects that cannot be graphically described.

A digital model must be capable of relating these two representations. Creating such a model as an artificial construction of reality in a computing environment requires a tool set exploiting the technology both of computer graphics (CG) (Sutherland 1963, 1970, Foley et al 1992, Watt 1993) and database management (DBM). Geographic information systems (GIS, Burrough 1986, Maguire et al 1991), and computer aided design (CAD) are examples of such tools. The essential difference between GIS and CAD is the handling of the spatial aspects rather than the non spatial aspects.

1.1 Needs for 3D GIS

We live in a three dimensional (3D) world. Earth scientists and engineers have long sought graphic expression of their understanding about 3D spatial aspects of reality in the form of sketches and drawings. Graphical descriptions of 3D reality are not new. Drawings in perspective view date from the Renaissance period (Devlin 1994). 3D descriptions of reality in perspective view change with the viewing position, so their creation is quite tedious. Traditional maps overcome this problem by using orthogonal projections of the earth. However, they offer a very limited 3D impression.

These traditional drawings and maps reduce the spatial description of 3D objects to 2D. Using computing technology, however, knowledge about reality can be directly transferred into a 3D digital model by a process known as 3D modelling. A 3D description of reality is independent of the viewing position. Adequate cover of the aspects of reality under investigation requires its understanding from many different viewpoints. The disciplines of geology (Carlson 1987, Bak and Mill 1989, Jones 1989, Youngman 1989, Raper and Kelk 1991), hydrology (Turner 1989), civil engineering (Petrie and Kennie 1990), environmental engineering (Smith and Paradis 1989), landscape architecture (Batten 1989), archeology, meteorology (Slingerland and Keen 1990), mineral exploration (Sides 1992), 3D urban mapping (Shibasaki et al 1990, Shibasaki and Shaobo 1992), all draw on 3D modelling for the efficient completion of their tasks.

A 3D model is the basis of a system providing the functionality to accomplish the task in hand. Scott (1994) has summarized the work of Bak and Mill (1989), Fisher (1993), Kavouras and Masry (1987), Raper (1989), Raper and Kelk (1991), and Turner (1989), to provide a set of functions that can be expected from 3D modelling. These should provide the means for constructing a 3D model from disparate inputs, permit the maintenance of existing models, facilitate effective 3D visualization with, for example, orthographic, perspective or stereo display with hidden line/surface removal, surface illumination, texture mapping; and spatial analyses enabling the calculation of volume, surface area, centre of mass, optimal path; spatial and non spatial search and inquiry.

CAD is a typical CG tool for 3D modelling used in, for example, car, machinery, aircraft and spacecraft design, the construction industry, and architecture. CAD focuses on the geometric aspect of the model and its 3D visualization. An example would be a perspective view with hidden line and surface removal, surface illumination, ray tracing, and texture mapping. The question arises whether CAD can support all the tasks required in the disciplines listed above. Attempts have been made to use CAD for tasks in earth sciences requiring 3D modelling and functionality. However, it cannot immediately be assumed that CAD is suited to those tasks, for the following reasons.

- CAD was developed to solve problems in the design of man made objects with well or predefined shapes, sizes, spatial relationships and thematic properties. CAD does not provide the tools for data structuring, or dealing with objects lacking such well-defined shapes, sizes, spatial relationships and thematic properties. Neither is it capable of analysing spatial relationships, nor coping with the disparate data sets and uncertainty typically encountered in GIS. For example, CAD will not reliably maintain the neighbourhood relationships between objects important in earth science analyses, because these relationships may not be considered significant for the design.

- Designing an object, such as a building, is a subjective matter. All aspects of objects and their relationships have to be decided by a human designer; there is little that can be automated. Earth science applications seek to model existing objects, with shapes, sizes and interrelationships outside human control. Here, automation is desirable because of the large number of objects involved. Some relationships important for spatial analysis have to be created automatically. CAD does not usually provide a function for this kind of automation.

- CAD starts the object definition from 3D. When objects are broken down in 2D components, the relationships between them are known. Earth science applications typically model components of reality separately, mostly in 2D, and are dominated by the application view, available tools and information. The components have to be combined and their interrelationships discovered at a later stage. That is quite difficult, since CAD does not usually provide sufficient tools to derive the relationships between the separate components.

- CAD creates a complex object by combining several components possessing such simple geometry as a cube, cylinder, or sphere. The operations of transformation, union, and intersection can be readily applied to such components to obtain the complex object. Earth science applications usually treat a complex object as a whole. Decomposition into primitives is comparable to reverse engineering, the opposite of CAD. The modelling approach used by CAD may not therefore always be suitable for earth science applications. Geometric primitives of an even lower level, such as points and lines, are needed to represent complex reality beyond man made objects. These geometric primitives also determine the related operations which CAD may not be capable of providing.

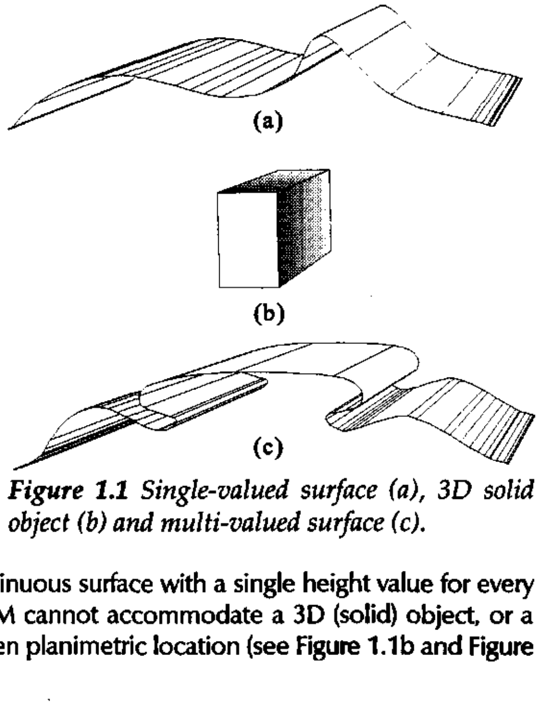


Figure 1.1 Single-valued surface (a), 3D solid object (b) and multi-valued surface (c).

A more suitable tool for earth science applications would be a GIS providing a 3D modelling capability, that is to say, a 3D GIS. At the time of writing, a GIS capable of providing the functions in the above list with full 3D modelling capability is not commercially available. Most GISs still limit their geometric modelling capability to 2D so that the 3D representation, analysis and visualization provided by CAD are not possible. Most endeavours to model the third dimension can be found in the representation of terrain relief and in digital terrain models (DTM). DTM can facilitate spatial analyses related to relief, including slope, aspect, height zone, visibility, cut and fill volume, and surface area, and the 3D visualization of a surface, as in a perspective view. However, the basis of DTM is a continuous surface with a single height value for every planimetric location (see Figure 1.1a). DTM cannot accommodate a 3D (solid) object, or a surface with multiple height values at a given planimetric location (see Figure 1.1b and Figure 1.1c, respectively).

Although raster-based systems which could be regarded as 3D GISs are available, they may not be able to maintain the knowledge about reality available in the original data set. This knowledge may be lost because of the problems of resolution and resampling. As a remedy, the original data set would have to be stored separately from the model, for example, for:

- recreating the model if the result proves to be unsatisfactory because of unsuitable mathematical definition
- creating another model with different resolution
- merging with another data set to create a new model
- archiving as a reference to, or evidence of, the model.

These activities imply the need to store original data in an appropriate structure ready for future use. Necessary information about the data should be attached to each data element. In DTM for instance, information that a line is a breakline should be kept because it will have an impact on the interpolation. Similarly, other information can be attached which influences data handling strategies.

Since neither CAD nor GISs can at present fulfil the requirements of earth science applications, further research and development of a 3D GIS would seem appropriate.

1.2 The Need for Integrated Modelling

In addition to the problem of creating a system capable of offering 3D modelling and functionality, there is a further problem concerning the type of 3D model chosen as the basis for 3D GIS. The model contains knowledge about reality, so we consider below the types of real world objects it must represent. Two kinds of real world objects may be differentiated in terms of prior knowledge about their shapes and location, as shown in Figure 1.2. In reality, objects from the two categories coexist. Traditional GIS models the objects of each category independently with the result that two separate kinds of systems or subsystems have been developed.

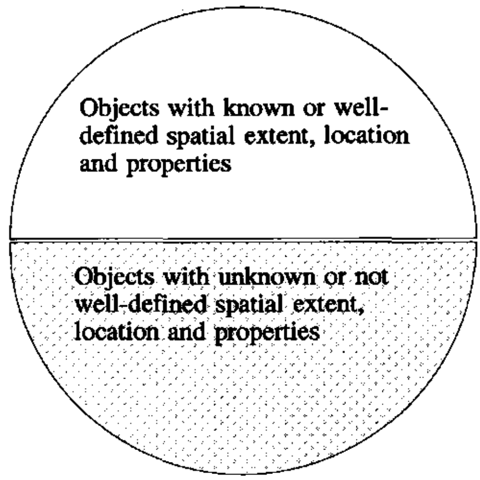


***Figure 1.2** Two types of real world objects with respect to their spatial extent.*

Raper (1989) has also defined these two categories of objects. The first category, regarded as 'sampling limited', is for objects having discrete properties and readily determined boundaries, such as buildings, roads, bridges, land parcels, fault blocks, perched aquifers. The second category, known as 'definition limited', is for objects having various properties that can be defined by means of classification, using property ranges. For example, soil strata may be classified by grain-size distribution; moisture content, colloid or pollutant in the water by percentage ranges; carbon monoxide in the air by concentration ranges, and so forth. Molenaar (1994a) regards these objects as 'fuzzy spatial objects'.

Separate modelling of these two categories of objects tends to contradict the reality, which leads to difficulties in representing their relationships. Such a question as, 'How many of the people working in a 50-storey office building are affected by polluted air generated by vehicles in nearby streets during rush hours'; cannot be answered until the two separate models are combined, as shown in Figure 1.3. Modelling them together with more accurate representation of their relationships in the 3D environment requires the integrated 3D modelling forming the general aim of this thesis.

Note also that the properties of an object may be well defined in some specific dimensions and ill defined in others. For example, given a DTM data set representing a surface, the planimetric extent of regions at the elevation of 100 metres above mean sea level cannot be defined until the result of interpolation based on a mathematical definition (for example, linear interpolation) is obtained. That is to say, although the spatial extent of this region may be known in the z-dimension, the spatial extent in planimetry (x, y) has still to be discovered. The model must contain the aspect allowing the appropriate operation, such as interpolation or classification, if the required description of the properties of an object is to be obtained.

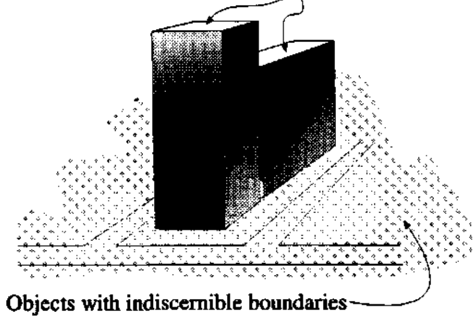


Figure 1.3 An example of two types of real world objects.

Apart from the problem of the separate modelling of the two types of objects, there remains the further problem of the separate modelling of an object's components. These components are relief and planar geometry associated with thematic properties. This separation has resulted in independent systems and data structures, DTM and 2D GIS, respectively. The consequences are data redundancy, which may lead to uncertainty when the two data sets are combined and only one data set has been updated.

DTM can facilitate several GIS analyses and visualization taking into account the third dimension. The spatial information stored in DTM and in GIS, however, can only be related through coordinates. This implies that relationships between different components may not be properly represented because of metric computation instead of topology. To overcome this, information derived from DTM must be converted into a form GIS can recognize. For example, information about a slope or height zone must first be converted into a thematic layer of GIS for further overlaying before the spatial analysis can be carried out. Imagine having information about the relief, planimetry and themes integrated into one model, so that conversion of such information as slope, height zone and so forth were no longer necessary. Such a question as, 'Which land parcels are subject to one-metre flooding?' could be answered from one model. Integrated modelling of this kind is evidently also required for 3D GIS.

1.3 Problems Associated with Integrated Modelling for 3D GIS

Establishing a 3D GIS while taking into account the integration of the necessary components and different types of objects requires the solution of the following problems related to the spatial model representing reality:

1) Design of a spatial model

- design of an integrated data model, or a scheme, permitting the derivation of a unified data structure capable of maintaining all the components of the geometric representation of real world objects, whether obtained from direct measurements or from derivations, in the same database. Each geometric component must be capable of representing a real world object differently understood by different people.

2) Construction of a spatial model

- development of appropriate means and methods for 3D data acquisition
- coordinate transformation into common georeferencing when different components are to be included into one database
- development of a data structuring method that unites the data from various inputs of multi sources into an integrated database capable of being maintained by a single database management system
- design of thematic classes to organize representation of real world objects with common aspects into the same category
- solving the uncertainty arising from discrepancies from different data sets during the integration process and converting the uncertainty into a 'data quality' statement to be conveyed to the end user.

3) Utilization of a spatial model

- utilization of existing components, such as 2D data and DTM (backward compatibility) and preparation of those components for future incorporation into the higher-dimension model (forward compatibility), to save the costs of repeating data acquisition
- development of additional spatial operators and spatial analysis functions
- development of manoeuvrable graphic visualization permitting the selection of appropriate viewpoints and representation enabling convenient, adequate uncovering of the details of objects stored in the database
- design of 3D cartographic presentation of information, including name placement, symbol, generalization, etc.
- design of a user interface and query language allowing users access to the integrated database
- development of a spatial indexing structure that speeds up data retrieval and storage processes for the integrated database, including specific (database) views for each user group and guidelines keeping these views updated according to the core database
- development of tools for navigating among different models stored in databases at different sites and computing platforms.

4) Maintenance of spatial model

- design of updating procedures, including the development of consistency rules ensuring the logical consistency and integrity of the integrated database, especially during the updating process.

1.4 Scope of This Research

It is not the intention of this research project to solve all the problems defined above, nor to achieve a fully functional 3D GIS. The scope of this thesis puts the major emphasis on the design of a 3D spatial model limited to the conceptual and logical design, and the construction of the spatial model according to that design. Because of the shortcomings of the raster approach, preference is given to the vector approach. Therefore, the study is limited to modelling in the vector domain. Apart from the design and construction issues, the exploitation of the spatial model in spatial query, analysis and visualization, are also included. It is not the aim of this thesis to address in detail:

- dynamic or temporal aspects of reality
- problems of different georeferencing during integration
- handling uncertainty from different observations and data quality
- generalizing input and output for both graphics and database
- spatial indexing to achieve highly responsive operation
- designing and optimizing a thematic hierarchy representing the organization of real world objects with common properties
- designing consistency rules for updating
- designing spatial operators for spatial analysis.

1.5 Previous Work

The status and progress of research in the 3D GIS field within the scope of this thesis and the identification of solutions and remaining problems are made clear from the following review of previous work.

The development of data models for a 3D GIS has branched in two directions. The first is the full 3D approach that looks directly into the design of a data model suitable for 3D GIS. Molenaar (1989) proposed a formal data structure (FDS) for a 3D vector map which may be regarded as a generalization of the 2D version of FDS. Shibasaki and Shaobo (1992), Rikkers et al (1993), Bric (1993), Bric et al (1994), and Wang (1994) have reported experimental use of 3D FDS.

The second approach comes from the viewpoint referred to as the 'integration of DTM and GIS'. DTM became a discipline in its own right in the late 1950s (Miller and Laflamme 1958). Fritsch (1990) has recognized the work of Makarovic (1977) as a proposer of this integration. Males (1978), though not addressing the integration issue, demonstrated the use of a triangulated irregular network (TIN) permitting the attachment of thematic information with elements of TIN in the ADAPT system.

Further steps towards this integration date from the late 1980s, when DTM became an essential part of many complex spatial analyses in GIS in erosion and slope protection, flood protection, the planning of irrigation for agriculture, the geometric correction of remotely sensed images, and so forth. Würländer (1988) investigated some strategies for integrating DTM into GIS. Sandgaard (1988) described an attempt at integrating DTM into the Dangraf system to facilitate the production of maps with contour lines. Mark and colleagues (1989) reported an approach to interfacing a GIS based on quadtree (Samet 1990) with a regular grid DTM for display or analysis. Ebner and colleagues (1990) proposed the 'subroutine interface' which was implemented in the program package HIFI-88. Subroutines for interactive editing of GIS are provided for updating DTM, for example, point insertion and deletion, and the change of coordinates in planimetry and height while databases of DTM and GIS remain separate. Ebner and Eder (1992) reported drawing on this approach to the facilitation of spatial analysis, using the HIFI-GIS interface with the SICAD-Hygris System to analyse forest damage in terms of such relief parameters as height, slope and exposition. Fritsch (1990) reported the realization of integration at the data structure level. Rather than a full 3D data structure, he suggested an approach that separates two geometric databases for terrain and situation data from another for thematic data. These three data sets are managed within one object oriented database environment. Fritsch and Pfannenstein (1992a) weighed the advantages and disadvantages of integration based on regular-grid, TIN and a hybrid of both. Fritsch and Pfannenstein (1992b) extended this comparison to the layer (organizing different themes in specific layers) and object class (organizes objects into a hierarchy) approach.

An issue in spatial modelling concerns the representation of spatial relationships. Egenhofer (1989), Jackson (1989), Kainz (1989), and Pigot (1991) have described the representation of spatial relationships between objects in 2D and 3D space, based on sound mathematical concepts.

Regarding the issue of model construction, CAD and most CG software packages provide interactive tools for the manual construction of models of objects with discernible boundaries. Manual construction is labourious and the method would not cope with large numbers of objects. For objects with indiscernible boundaries, significant progress has been made in computational geometry based on 2D and 3D Voronoi tessellation (Voronoi 1908, Thiessen 1911, Dirichlet 1850), in the construction of TINs, and tetrahedral networks (TEN). Watson (1981), Avis and Bhattacharya (1983), Edelbrunnner and colleagues (1986), Tsai and Vonderohe (1991), Midtbø (1993) have all suggested methods for the construction of TEN based on Delaunay triangulation criteria (Delaunay 1934). These methods were extensively applied long ago to the construction of TIN (Shamos and Hoey 1975, Lawson 1977, Lewis and Robinson 1978, Sibson 1978, McCullagh and Ross 1980, Lee and Schachter 1980, Bowyer 1981, Watson 1981, Mirante and Weingarten 1982, Maus 1984, Dwyer 1987, Sloan 1987, Macedonio and Pareschi 1991, etc.). *However, these developments are quite independent of GIS.*

For the issue of the exploitation of the 3D model, considerable progress has been reported in two other disciplines exploiting CG technology, namely CAD and virtual reality (VR). CAD and VR provide a realistic visualization capability, that is to say, perspective display with hidden line and surface removal, shading and surface illumination, ray tracing, and texture mapping. In addition, VR provides high interactivity within the concept of 'functional realism', allowing the user to manipulate and interact with virtual objects stored in the computer's database as in reality. For instance, the user can 'grab' a virtual object displayed on the computer screen, using the interfacing device called a 'data glove' which sends feedback to the user's hand (for example, a pulse, or vibration) as soon as the virtual object is virtually touched. Developments in this direction are also quite independent of GIS.

The status of the research in 3D GIS and the most relevant remaining problems can be summarized in the following statements.

- The full 3D approach, 3D FDS, does not support well the modelling of real world objects whose boundaries cannot be directly determined; further extension to cover this issue is needed.
- Progress made by the integration approach can only achieve solutions for surface related objects with little support from theoretical concept of spatial modelling. Extension of this approach to full 3D based on sound spatial mathematics is required.
- Efficient methods for data acquisition, data structuring, database creation and updating with respect to 3D GIS have yet to be developed.
- The incorporation into 3D GIS of independent developments in 3D visualization and 3D geometric construction, whether manual (interactive 3D graphical editing) or automatic (3D Voronoi and tetrahedral network), needs further research.

These problems lead to the following research objectives.

1.6 Research Objectives

The four main objectives of this research are to:

1. review and relate the important theoretical foundations of integrated spatial modelling
2. analyse the status and prospects of existing systems and data models for integrated geo-information with respect to terrain modelling and 3D-GIS
3. design a data model, data structure and associated operations for database creation capable of integrating objects of different nature and their various components
4. demonstrate the applicability of the proposed model.

For the last objective, the applicability of the proposed data model is demonstrated through experimental tests and the development of a software package.

1.7 The Structure of the Thesis

This thesis may be divided into four major parts. Part one includes the introduction, the elaboration of the theoretical foundations and an analysis of the status and prospects of the systems for integrated 3D geo-information. This part comprises chapters 1, 2 and 3. The second part reports the design phase. The focus is the design of the data model and data structure for the integrated 3D geo-information. This part includes chapters 4 and 5. The third part is the implementation and testing phase. It demonstrates how the design in the second part came into practice and explains the operations for database creation. This part includes chapter 6 and 7. Finally chapter 8, the concluding part, summarizes the most important achievements of the thesis.

Chapter 1 discusses the need for 3D GIS and integrated modelling. The scope of the thesis is defined. A brief review of previous work with respect to the defined scope is given, leading to the identification of the remaining problems and the objectives of the research.

Chapter 2 reviews and relates the important fundamental concepts in spatial modelling necessary for this research and also defines the terminology used in the thesis. The review follows the conceptual and logical design phases in spatial modelling. Mathematics about metric, order and topologic relations, simplicial complex and graph theory necessary for the conceptual design are summarized. This part also covers the review of current spatial modelling in general, including solid modelling, and models used for geo-information. The last part of the chapter briefly reviews relational and object-oriented approaches to the logical design phase.

Chapter 3 discusses the aspects of the system for integrated 3D geo-information. The major functional components of the system are outlined and a review of the technology supporting such functions are given. The aspects of systems are classified into evolutionary stages to make clear the direction of this research. Existing data models evidencing attempts towards integration are reviewed to reveal the need for further development.

Chapter 4 is devoted to the design of an integrated data model based on the needs for further development identified in chapter 3. The design is carried out within the scope defined in chapter 1. The development of the integrated data model is done step-by-step from lower to higher dimensions. The irregular tessellation of space and the decomposition of spatial objects into the minimal primitives are taken as the foundation of the design that follows the FDS approach.

Chapter 5 presents the logical design of the unified data structure derived from the integrated data model presented in chapter 4. The relational and object-oriented approaches are illustrated.

Chapter 6 introduces the procedures for constructing a 3D spatial model with respect to the integrated data model and data structure described in chapters 4 and 5 respectively. The constrained network construction procedure is presented which is crucial for the spatial model designed. An approach to the construction of a spatial model, based on the 3D FDS required as a structure prior to the constrained network, is also introduced.

Chapter 7 demonstrates how the integrated database can be applied in the context of GIS. The processes of database creation, graphic display, query and analysis carried out are described. Three data sets are used for this purpose; surface-based data, urban data and geological data. Some attempts to perform the spatial query directly in perspective or stereo view are also part of the demonstrations.

Chapter 8 concludes with the major findings of the research and recommendations of issues for future research.

2
FUNDAMENTALS OF GEO-SPATIAL MODELLING

This chapter reviews various concepts fundamental to spatial modelling and specifically related to geo-information. The aim is to outline the theoretical bases and fundamental concepts necessary for the design of a geo-spatial model. Since spatial theory is a relatively young discipline developed from a combination of many branches of mathematics and computer science, the terminology found in the literature is confusing. In this chapter, the terms used in this thesis are clarified. Five important components of spatial model and phases of modelling are defined. The emphasis is placed on the conceptual and logical designs of spatial models. Mathematical concepts concerning space, objects, and their interrelationships are taken as the foundation of the conceptual design of a spatial model. The concept of a simplicial complex and the theory of graphs are chosen as methods of representing objects and their interrelationships in the model. Existing spatial models are taken as examples to show the lines of further development. Relational and object-oriented approaches are considered important for the logical design of a spatial model.

2.1 Models and Their Importance for Geoinformation

In the disciplines related to geoinformation science, the word 'model' has been used in two different ways. The first meaning is in the sense of a representation, or replica, of something regarded as real or genuine, like a globe in the classroom as a replica of the earth. The second meaning refers to something used to produce a number of replicas, and may be needed for the mass production of those replicas. The word 'model' in this sense may be comparable to the word mould, or form, and has the meaning of design, plan, or scheme. The quality of the mould directly influences the quality of the replica, so that more serious attention has to be paid to the design and construction of the mould than to the replica.

Regardless of the meanings of the word model, the process of producing a model is known unequivocally as modelling. It is necessary to state clearly what the model and modelling are actually meant for.

In the context of earth science, the end product we seek is a model in the sense of a replica of some portion of the planet earth, and is called a geo-spatial model. Since the term 'spatial model' covers a large territory over many disciplines (like the modelling of human anatomy in medicine, molecular structure in chemistry, or atomic structure in nuclear physics), we add the prefix geo to indicate the scope and purpose of this earth-related model.

For the information system to utilize the geo-spatial model, it must be constructed in digital form, so that it can be maintained and exploited by a computer to perform certain tasks or operations that are:

1) less convenient in reality; for example, a distance can be obtained from a model instead of measuring from place A to B in reality, provided that places A and B are represented in the model
2) too expensive, too difficult, or practically impossible in reality; for example, a geologist may wish to see the continuou layer of sandstone lying fifteen metres under the earth's surface; removal of the upper soil to see this layer in reality is too expensive to contemplate.

The model in a digital form is in fact the database itself. Not only is a database a collection of data, it also contains relationships between data elements, and rules and operations to change the state of the data elements, regardless of how these components are stored. Components may be kept in one data set, or separately, at different places, depending on the system that manages and manipulates the model— the database management system (DBMS).

A model containing all aspects of the reality is impossible, because of reality's complexity. Only some aspects can be included in the model at a manageable level. Hence, the quality of the model is judged only in terms of its purpose and how the model will be used. If the model permits the performance of the tasks or operations as required, and with acceptable results, the quality may be regarded as good. A model constructed for a single purpose may not be able to serve tasks or operations for different purposes, unless it is an integrated model designed and constructed for multi purposes.

A single-purpose model represents only a single view of the reality (Figure 2.1). An integrated model represents various views of the reality, so the integrated model may be considered to be of higher value, since it contains more aspects of the reality and may serve more purposes.

2.2 Components of Geo-spatial Model

A model in the form of a database requires the categorization of aspects of reality into the components of the database managed and manipulated by a DBMS (Flavin 1981). The components of a geo-spatial model include the following:

1) Object types
Object types are classes of spatial entities in a geo-spatial model. In reality, they may be a road, river, city, land use, and so forth.

2) Relationships
Spatial relationships are named associations between two or more spatial objects. For example, road A *passes through* city B. 'Passes through' defines a relationship between the road A and the city B, and may be written in a predicate form as 'Pass_through (road A, city B)' (Molenaar 1994b).

3) Attributes, or descriptions
Attributes, or descriptions, are observed facts about a spatial object or relationship. An attribute or description is the smallest (non spatial) unit in the model, and has to be associated with an object type or relationship to be meaningful. An attribute or description cannot stand alone in

the model. For example, the object type 'road' has the name 'A1', indicating that it is a highway passing by several cities.

4) Conventions
A convention results in a set of rules and constraints that govern the content, structure, integrity, and operational activity of the model. A convention applies to the entire model. An example: a convention stating that 'each feature class contains objects of only one geometric type' results in a rule preventing an area object from belonging to a line feature class (Molenaar 1991).

5) Operations
A spatial operation is an action changing the state of the representation of a real world object being modelled, or deriving additional information from the current representation. Operations can be identified by events. Two types of operations can be distinguished: standard, and user-defined. Standard operations are provided for routine tasks. A user-defined operation is built by combining different types and sequences of standard operations. Standard operations include retrieve, add, delete, modify, union, intersect, difference, compare and so forth. They can be applied to different components of the model.

2.3 Phases in Geo-spatial Modelling

Before continuing this review of necessary fundamental concepts, the steps followed in geo-spatial modelling are defined.

Obtaining a geo-spatial model requires two main steps: the design phase, and the construction phase (Figure 2.1). Once the model is in place, maintenance forms an additional phase. The design phase includes all the abstraction processes, ranging from the conceptual design, the logical design, to the physical design. The product of the conceptual design is referred to as a conceptual model, or data model (Peuquet 1984, Maguire and Dangermond 1991). It comprises a general scheme describing what should be included in the model.

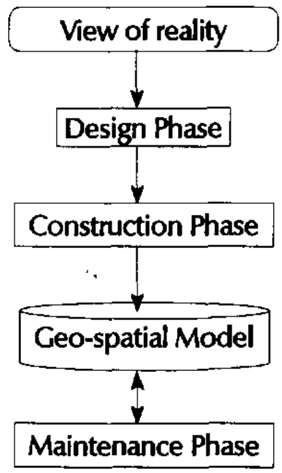


Figure 2.1 Geo-spatial modelling.

The logical design sets out all the elements needed for the construction, without stating the actual size or type of each element of the model. This design results in a logical model, or data structure. The physical design phase specifies the actual size and type of each element of the model for the implementation of the geo-spatial model. For example, a 16-bit real number may be used to store the attribute 'width.' This phase yields an internal model, or file structure, to be used by the software engineer to establish the low level communication with the hardware at the bit and byte level. Figure 2.1 and Figure 2.3 graphically illustrate this process.

Molenaar (1994b) also suggests the involvement of different disciplines in geo-spatial modelling, as shown in Figure 2.2.

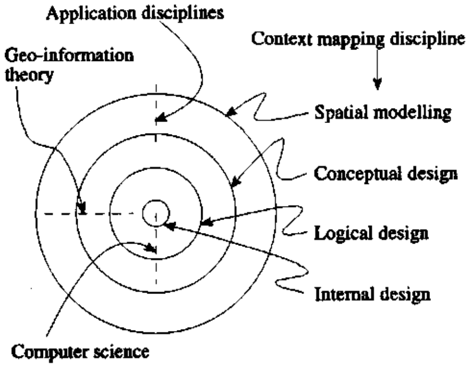


Figure 2.2 Levels of geo-spatial modelling (After Molenaar 1994b).

The five components of the geo-spatial model listed in the preceding section can be realized in these three different design phases: the object types and relations in the conceptual design phase; the attributes or descriptions of objects and relations in the logical design phase; the operations in the physical design phase.

The conventions must operate in every design phase. In the conceptual design phase, the conventions should state the allowable type of objects and relations between them to be included in the model. In the logical design phase, the conventions should state how the representation of one object is distinguished from another; an object should have a unique identifier. In the physical design phase, the conventions comprise a set of integrity and consistency rules for the operations that may change the state of the model; for example, the union of two areas sharing a common boundary has to yield only one area.

The design of the model is followed by the design and implementation of the necessary functions and the user-interface to enable the construction and exploitation of the model. The result of this implementation is a geo-spatial information system (GIS). Having constructed the model, it must be kept valid to ensure that it

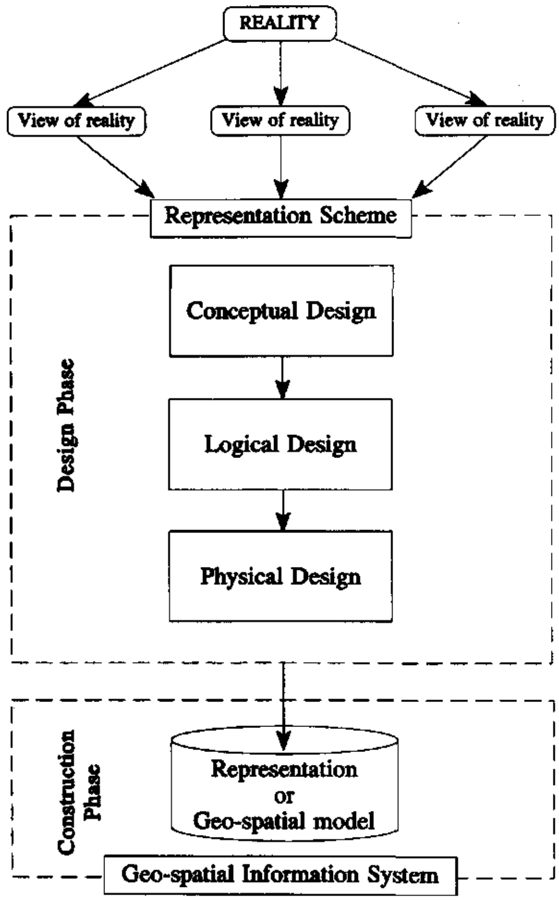


Figure 2.3 Design and construction phases for a geo-spatial model.

remains in a state comparable with the reality, which is dynamic in nature. This is the maintenance phase. The basic maintenance operations of insertion, deletion, and modification can be applied to any component of the model, that is to say object types, relations, rules, attributes and operations. A GIS should also provide functionality to maintain the geo-spatial model.

2.4 Conceptual Design of a Geo-spatial Model

The design phase deals with the abstraction of reality into the representation scheme. This phase answers two basic questions: what aspects of reality (real world objects and the relationships between them) are to be modelled; how should they be represented in the model?

A geo-spatial database represents a state of reality from a specific point of view or interest at an instant in time (if the temporal aspect itself is not the subject of the model). The reality consists of a set of various objects and the relationships between them which should be capable of representation as components of the model described in the preceding section. To be manageable, it is necessary to determine a limited number of aspects of the reality (objects together with the relationships between them) during the design phase which can be represented as the first and second components of the model (see section 2.3).

2.4.1 Definition of Space

Reality may be viewed as a space, that is to say, a collection of spatial objects and the relationships between them (Gatrell 1991). Each spatial object occupies a subspace to define its own spatial extent, which may be defined by a set of spatial locations together with a set of interest properties characterizing those locations (Smith et al 1987). Different sets of relations may define different types of space. Metric space, for example, is based on distance relationships; topological space is based on topological relationships.

For the mathematical description of space, we can rely on set theory, introduced by Cantor in 1880. Let O be a set of objects $\{o_1, o_2, o_3, ..., o_i\}$. R is a binary relation on O if $R \in O \times O$. If R is a relation on O, the relationship $(o_1, o_2) \in R$ can be denoted in prefix form by $R(o_1, o_2)$. A space S is then a collection, that is to say, a set of subsets, $\{ [O], [R] \}$, denoted $S = \{ [O], [R] \}$.

In reality, the space is an unbounded region consisting of numerous objects and relations. The space S (that is, a finite set) is only a view of reality in which the context is defined for describing the aspects of reality relevant to a particular discipline.

Having determined the collection [O] and [R], the question related to the aspects of reality to be modelled can then be answered. An example is only to include in the database the object types roads, rivers, buildings and land parcels and the relationships between buildings and land parcels, rivers and roads, roads and land parcels. In this sense, this database can be regarded as the space S.

To answer the second question, how to represent the objects of reality and the relationships between them, we have to consider some fundamental concepts of spatial modelling.

2.4.2 Abstraction of Space

There are two major abstractions of space, each of which passes on its characteristics to the spatial objects residing in that respective abstraction. The first conceptualizes space as tessellated into a contiguous set of smaller sub-spaces and is known as a field-based, or tessellation-based definition. Each individual spatial object is composed of a set of sub-spaces (for example, a raster element in the raster-based GIS).

The second abstraction treats the space as empty and homogeneous, and consists of a collection of spatial objects. It is known as an object-based, or feature-based definition (Ehlers et al 1989).

Each sub-space of the field-based space is typically understood as, and associated with, regular shapes, like a square or a cube, usually found in the raster-based geo-spatial model. Irregular shapes are also used, such as in the triangular irregular network (TIN) that subdivides the space into a set of irregular triangular shapes, as frequently used for the representation of single value surfaces.

For object-based space, the best example is the vector-based geo-spatial model, where each object is composed of several vector elements in the form of geometric primitives (such as nodes, edges, faces, or bodies).

Field-based space and object-based space have different advantages and disadvantages. The field-based representation of space offers connectivity and continuity in all directions, thus providing the freedom to visit any location in space. An intuitive example from reality is travelling in free space in an aircraft. The pilot navigates by connecting the information in his vicinity, such as landmarks, topography, or a city, to determine the travelling direction, but otherwise moves freely. This kind of approximation may be regarded as spatial interpolation.

Object-based representation does not permit such freedom. The navigation in space is limited to a confined subspace defined by each spatial object. Connectivity and continuity are defined along with the existence of spatial objects. An example from reality would be travelling along a highway by car. The highway is comparable to a confined subspace of the global space. It restricts travel to a certain direction. The explicit destination is defined for each highway, so navigation in space is just a matter of selecting the right highway. No approximation for direction is necessary in this case.

The abstraction of space is typically decided during the conceptual design phase, which is usually driven by the type of spatial operations. This thesis presents an attempt to integrate the field-based and object-based abstractions into a hybrid abstraction to allow confined and unconfined navigation in a geo-spatial model, thereby facilitating a wider range of spatial operations.

2.4.3 Abstraction of Real World Object

In the present context, the earth is the subject under consideration. It is important to bear this in mind, since some aspects of the earth have to be taken into account and included into the model. In geoinformation science, any real world object may be described geometrically and thematically (see Figure 2.4 and Molenaar 1989, Maguire et al 1991, Gatrell 1991). The terms metric and semantic have also been used (Makarovic 1984). In this thesis, the representation of a real world object is referred to as a feature where the terms spatial entity and geo-object may be found elsewhere (Peuquet 1988, Laurini and Thompson 1992, Raper 1989). A real world object that has to be described, or related to a location in reality, is referred to as a spatial object.

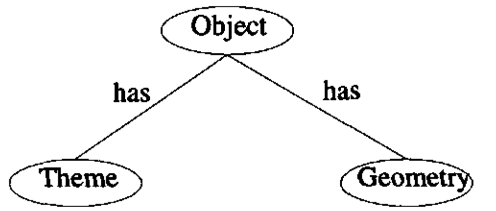


Figure 2.4: A general abstraction of the real world object (at an instant of time).

Figure 2.4 can be regarded as a general representation scheme for any spatial object. The thematic and geometric aspects may be separately modelled and considered as general component types of the model. They have, however, to be brought together at some stage. The geometric aspects are the spatial characteristics of the object such as shape, size and location. The thematic aspects are the non spatial characteristics of the object related to its state, functionality, or utility in reality.

Figure 2.4 provides an extreme level of abstraction about the aspects of the reality we want to deal with. It can only be used as a general framework for the overall modelling process. This abstraction must be further elaborated to achieve a more specific design.

Note, however, that Figure 2.4 is limited to the representation at a certain instant. Object states that change over time have not been considered. Otherwise, the dynamic aspects of the objects would have to be included as additional components of the model. The term spatio-temporal is used to indicate such a kind of model; it lies, however, outside the scope of this thesis (Langran 1992, Tansel et al 1993 for more details on this subject).

2.4.3.1 Geometric Component

Two important aspects of a real world object, location and shape, need to be included into the model which allows further derivation of the size of the object. However, they can only be correctly described if the dimension of space is taken into account. In mathematics, the dimensions of Euclidean space (see section 2.4.5.1) are expressed through the number of referential axes, which are linearly independent from each other. The distances from the origin along each axis are arranged into the *ordered-n-tuples* notation $(a_1, a_2, ..., a_n)$, which is a sequence of n real numbers used to represent a coordinate tuple in *nD-space* (denoted by R^n, see also Anton 1987).

In geoinformation science, the term 'dimension' has been used to denote various meanings. With respect to boundary representation, the dimension may indicate the data type being used to represent the object, such as point (0D), line (1D), area or surface (2D) or body (3D) (Frank and Kuhn 1986). Each object occupies a subspace and has its own dimensionality, which may be regarded as the internal dimension. The external dimension (R^n) is then the dimension of the space embedding the object. The term 'dimension' also frequently indicates both the internal and external dimension. For example, 2D may mean objects in R^2, 2.5D means 2D objects in R^3, 3D means 3D objects in R^3, 3.5D means 3D objects in R^4, and so on. In this thesis, these kinds of notations are used, dependent on the context of each part. Egenhofer and Herring (1990) also discuss the dimensionality of space and objects.

Having defined the dimension, an object's location and shape can be described. Location is defined by a set of coordinate tuples, while the description of shape can be given in different ways, for example, by a mathematical function, a verbal description, or a skeleton with radius functions (Blum 1967, Pilouk 1992). CAD is a discipline focusing on the 3D modelling of geometric aspects where several approaches have been used:

- Primitive Instancing (PI): describes an object by a set of parameters together with a shape function; for example, a rectangle can be described by its width, height and a rectangular shape function.

- Sweep Representation: applied to an object of regular shape; for example, a cylinder is the result of sweeping a circle along a straight line.

- Boundary Representation (BR): describes an object through its boundary elements, that is to say, the vertices, edges and faces for a 3D object.

- Constructive Solid Geometry (CSG): hierarchically decomposes an object into a set of components with simpler geometry. The node of each hierarchy may contain the set operator needed to combine together the components in a lower level of the geometric hierarchy. Translation and rotation parameters may also be attached to this node. For example, a solid cube with a cylindrical hole can be decomposed into a solid cube and a solid cylinder under the set operator $\cap$ (intersection) and a translation parameter to align the cylinder into the middle of the cube.

- Spatial-Partition Representation: also decomposes an object similar to CSG, but to the more primitive level known as cell. Only the set operator $\cup$ (union) is allowed to combine cells to reconstruct the object. This means that no intersection between two cells is possible. This distinguishes the spatial-partition representation from CSG. One criterion for this kind of decomposition is that the adjacent cells must share common boundary elements, such as vertices, edges, or faces. Different decomposition schemes can be used:

  - cell decomposition: decomposes an object into various types of primitives with shapes which need not be regular; for example a simple house may be decomposed into a cube and a prism

  - spatial-occupancy enumeration: decomposes an object into a set of regular cells with fixed shape and size; for example, regular-grid, pixel and voxel

- irregular tessellation: decomposes an object into one type of primitive with, however, different shapes and sizes; for example, triangular irregular network, tetrahedral network (TEN)
- hierarchical regular subdivision: subdivides a space into homogeneous zones using only one type of primitive which varies in size; for example, rectangle (quadtree), cube (octree)
- binary space-partitioning (BSP): subdivides an object into pairs of planes with arbitrary orientation and position.

In describing an integrated model, the boundary representation with irregular tessellation is used to geometrically describe an object (see chapter 4). More details about geometric modelling can be found in Requichar (1980), Mäntylä (1988), Samet (1990), Foley et al (1992), Bric (1993), Cambray (1993).

2.4.3.2 Thematic Component

Apart from geometry, objects are given a referential identifier and descriptions and may be organized into a group, or theme, to differentiate them and make reference to them more convenient. Objects having the same characteristics may be grouped together, becoming more easily distinguished from objects with other characteristics. Nevertheless, the criteria for judging whether an object belongs to a particular group are based on a specific viewpoint. Using different criteria, an object can be classified into a different group. The process of classifying objects into groups is known as thematic modelling. The term single-theme is used when the geometric description of an object is related to only one theme (Molenaar 1989), and multi-theme if the geometric description of an object relates to more than one theme (Kufoniyi 1995).

In this thesis, single-theme and multi-theme express the homogeneous and heterogeneous properties of a spatial object (see section 2.4.8.3 for more details).

Since thematic modelling is context dependent with respect to a particular application domain, no attempt to achieve the modelling of a thematic component capable of accommodating a wide range of applications has been made in this thesis.[1]

2.4.4 Object and Spatial Extent

Two kinds of spatial objects can be distinguished on the basis of knowledge about their spatial extent. The first is of the type determinate spatial extent. Objects of this type are referred to as determinate spatial objects. This type of spatial object has a discernible boundary, and is typified by houses, roads, a river, a land-parcel that can easily be sensed. Spatial objects of the second category have an indiscernible boundary which is difficult to sense. These objects are of the type indeterminate spatial extent, for example colloid in water, plume of smoke, temperature distribution, soil type, etc., and are referred to as indeterminate spatial objects. The boundaries of determinate spatial objects can be sampled and directly represented in the database. This is not the case for indeterminate spatial objects; their boundaries cannot be

[1]Consult the Tri-service data standard (1994) for guidelines on defining different themes.

directly sampled and must be derived by means of classification, or computation, using specific property values of the surrounding neighbours (for example, by interpolation or extrapolation). Therefore, the representation of indeterminate spatial objects in the database can only be indirect.

2.4.5 Spatial Relations

Spatial relations are a key issue in the design of a spatial model. Many extensive reviews and discussions can be found; Frank and Kuhn (1986), Pullar (1988), Pullar and Egenhofer (1988), Egenhofer (1989), Egenhofer et al (1989), Kainz (1989), Egenhofer (1990), Kainz (1990), Egenhofer and Franzosa (1991), Pigot (1991), Pigot (1992). In this section, only a brief review is therefore given of some important basic concepts about spatial relations.

A set theoretical definition of a relation has been given in section 2.4.1. Recall that R is a relation on a set O of objects. In general, R can be further distinguished by its different basic properties that depend on the relationships between its member elements (see also Willard 1970, Stanat and McAllister 1977, Pullar and Egenhofer 1988).

- R is reflexive, if each element can be compared with itself (if and only if $(o_i, o_i) \in R$), for example 'point A' is equal to itself.
- R is symmetric, if and only if $R(o_1, o_2)$ implies $R(o_2, o_1)$. For example 'area A' is adjacent to 'area B' implies that 'area B' is adjacent to 'area A.'
- R is antisymmetric, if and only if $R(o_1, o_2)$ and $R(o_2, o_1)$ implies $o_1 = o_2$ for all $o_1, o_2 \in O$, for example if $a \leq b$ and $b \leq a$, then $a = b$.
- R is transitive, if and only if $R(o_1, o_2)$ and $R(o_2, o_3)$ implies $R(o_1, o_3)$ for all $o_1, o_2, o_3 \in O$, for example area A < area B and area B < area C then area A < area C.

For example, given a set of real number N, $<$ is a transitive relation on N, $\leq$ is a reflexive, antisymmetric, transitive relation on N, and $\neq$ is a symmetric relation on N.

It is necessary at this stage to consider the definition of functions in mathematics used later.

Given two sets A and B, a function (or map) f from A to B, denoted $f: A \rightarrow B$, is a subset of the Cartesian product A x B with the following properties:

a) For each $a \in A$, there is some $b \in B$ such that $(a, b) \in f$.
b) If $(a, b) \in f$ and $(a, c) \in f$, then $b = c$.

Each $a \in A$ must be in relationship with exactly one $b \in B$ and the relationship $(a, b) \in f$ is normally written in a prefix form as $b = f(a)$.

Comparing with relation R, every function on A is a relation R on A. However, not all relations on A are functions.

Three classes of spatial relations, namely metric, order and topology, have been distinguished, based on the type of function or relation associated with a set of objects (Egenhofer 1989).

2.4.5.1 Metric

Metric relations are built around the notion of distance function. Its mathematical description is as follows (see also Willard 1970):

Given a set M with x, y, z ∈ M and a set of real number N. A metric relation d is a function d : M x M → N with the following conditions:

a) d(x, y) ≥ 0, Distance from x to y is more than or equal to zero.
b) d(x, x) = 0; d(x, y) = 0 implies x = y, Distance from x to itself equal to zero. Distance from x to y equal to zero implies that x is equal to y.
c) d(x, y) = d(y, x), Distance from x to y equal to distance from y to x
d) d(x, y) + d(y, z) ≥ d(x, z) (triangular inequality). Distance from x to y plus distance from y to z is more than or equal to distance from x to z.

A metric space is an ordered pair (M, d) consisting of a set M together with a function d: M x M → N satisfying the above four conditions. The function d is also called the **metric** on M. Functions d : M x M → N are called distance functions. A metric space is the Euclidean n-space, denoted R^n, if the distance function is the Euclidean distance below:

$$d\,((x_1,\ \ldots,\ x_n),\ (y_1,\ \ldots,\ y_n)) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

The number 'n' defines the number of distance components between x and y (each one computed along an independent vector) and denotes the dimensionality of Euclidean space.

The distance functions available in metric spaces are used to develop the notion of continuity crucial for the development of topology.

2.4.5.2 Order

Order defines a comparative type of relationship between the objects based on a preference. Two kinds of order relations can be distinguished; **strict order** and **partial order**. Strict order is a relation < , which is transitive. This kind of relationship may be represented as a tree-like structure. Partial order is a relation ≤, which is reflexive, antisymmetric, and transitive, and may be viewed as a network structure. Every order relation has a converse relationship, for example a < b conversely implies b > a. A formal study about the use of order for spatial relationships has been reported by Kainz (1989), Kainz et al (1993). Algorithms and data structures for order operations have also been presented in Kainz (1990).

2.4.5.3 Topology

Since the eighteenth-century, topology has developed as a discipline of mathematics. The definition of topology, as the study of the properties of figures remaining invariant under topological transformation, was given by Augustus Möbius (Devlin 1994). The explanation of topology in this section, however, follows the general (point set) topology founded by Hausdorff in 1914 (see Willard 1970). Point set topology was developed from metric (distance) which is more easy to understand (Mäntylä 1988). The purpose of introducing topology is to be able to define any continuous function without mentioning distance (Willard 1970, Armstrong 1983, Pullar and Egenhofer 1988), thus adding the concept of 'neighbourhood' to location, distance, and direction (Kainz 1989). The expression of spatial relationships in the form of topology is more appropriate for handling by current computer technology, which bases arithmetic computation on a finite numbering system, and so cannot be used to completely represent continuity based on Euclidean distance (Franklin 1984, Frank and Kuhn 1986). For example, the state of a point lying inside a polygon might be changed after rotation or scaling, because of rounding errors.

The expression of continuous function is accomplished by introducing the concept of a point-set in metric space that is an open set (a set that does not include its boundary; Pigot 1991). Any open set has a continuity property (consult Willard 1970 for proof). A point-set P is an open set if every point $x \in P$ is surrounded by an ϵ-sphere of radius $\epsilon > 0$ such that distance between x to any point $y \in P$ is always less than ϵ. An example of ϵ-sphere about a point c of a set of real number is an open interval $(c-\epsilon, c+\epsilon)$

A mathematical definition of topology is as follows:

A topology on a set X is a collection T of subsets of X, called the open sets, satisfying:
a) Any union of elements of T belongs to T,
b) any finite intersection of elements of T belongs to T,
c) $\emptyset$ and X belong to T.

A topological space is denoted by (X, T). Given two topological spaces A and B, $f : A \rightarrow B$ is a continuous function if it preserves the neighbourhood relations between mapped points. This mapping is also called continuous mapping, or homeomorphism (Alexandroff 1961, Pigot 1991). The topological transformation is commonly known as rubber sheeting, in which translation, rotation and scaling are included (Pullar and Egenhofer 1988). Examples of homeomorphic mappings are transformations to correct distortion resulting from paper or film shrinkage in cartographic or photogrammetric digitizing.

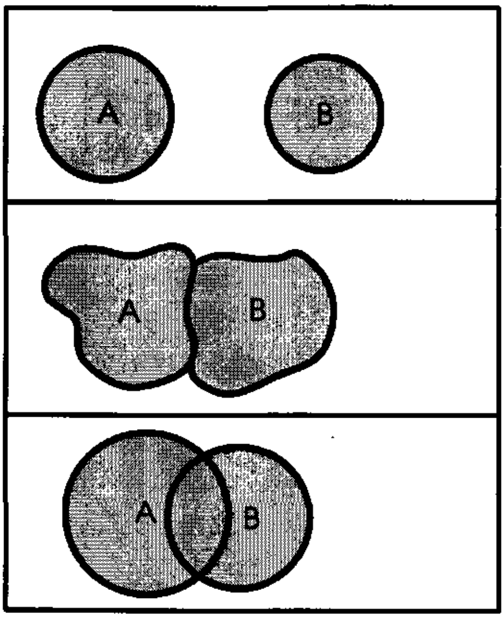
Some properties of topology have been expressed as follows:

- Topology is defined as the set of properties which are invariant under homeomorphisms (Alexandroff 1961) - one-to-one, continuous, and onto transformation (Pigot 1991).
- Topological relationships are invariant under topological transformations such as translation, scaling, and rotation (Egenhofer 1989).

The transformation that includes translation, scaling, and rotation is known in photogrammetry as geometric transformation. It defines changes in shape, size and location of an object.

Topology describes the relationship between an object and its neighbours. Topological relations can be defined through the three components of an object, that is, the interior, boundary, and exterior (Vaidyanathaswamy 1960, Pullar 1988). An elementary set operation, an intersection, is used as a mechanism to determine each type of relation. For example, if the intersection between a boundary set of object A and a boundary set of object B yields a non empty set, the relationship between the object A and the object B may be defined as 'touch.' If in addition the intersection between the interior set of the object A and the interior set of the object B also yields a non empty set, the relationship between these two objects may be defined as 'overlap.' Figure 2.5 shows some examples of topological relationships between two objects.

Intersection of the three components of two objects can be organized into a 3x3 matrix. This gives a 9-digit logic state, called a 9-intersection, which can be interpreted as relation codes (see Bric 1993). The 9-intersection gives in total 512 possible relationships, from which a set of relevant relationships can be found by a process described in Pullar and Egenhofer (1988).



Object A

A / B	Int	Bnd	Ext
Int	0	0	1
Bnd	0	0	1
Ext	1	1	1

Object A

A / B	Int	Bnd	Ext
Int	0	0	1
Bnd	0	1	1
Ext	1	1	1

Object A

A / B	Int	Bnd	Ext
Int	1	1	1
Bnd	1	1	1
Ext	1	1	1

Boundary set   Interior set   Exterior set

Figure 2.5 Example of spatial topological relationships.

The dimensionality of topological space has frequently been mentioned in the literature. Since topological space is derived from metric space, it also inherits the dimensionality defined in metric. 2D topology would mean that the topological relations are only valid for 2D metric space and certainly 3D topology would only be valid for 3D metric space. For example, a triangle may have at most three neighbours in 2D space, but there can be many more neighbours in 3D space. An important limitation is that there be no continuous mapping from the higher dimension to the lower dimension. This endeavour will therefore result in loss of information. 2D topology has been intensively studied by Egenhofer (1989). Pigot (1991), Bric (1993), Rikkers (1993) have studied 3D topology, while Pigot (1994) have studied 4D topology.

2.4.6 Application of Spatial Relations

The three types of spatial relations have been used in GISs with little realization by users concerning their categories. This section identifies some of their usages with respect to type of relations.

2.4.6.1 Spatial Indexing

A spatial database often contains a large volume of data, so a lot of time is required for the data retrieval process, particularly for non sequential data access, for example during a query operation. This process can be speeded up by heuristically limiting the search space in the database. We must know roughly the location in the file that contains the data elements or records. Storing a bounding rectangle for a set of data elements can be used as a method of giving a rough spatial index to a subset of the database. By using the metric relation to compare the coordinate set of the data to be retrieved with the coordinates of two opposite corners of the bounding rectangle, faster data retrieval can be achieved, because the search is limited within a particular rectangle at the end. Bounding rectangles of many subsets of a database may be further organized using a tree structure, further speeding up spatial access. An example is the R-tree (Guttman 1984, Samet 1990) which exploits the order relation to organize the 2D data and allows the search to proceed from coarse to fine. Navigation in the tree structure also helps avoid metric computation requiring a long access time, so it dramatically speeds up the process. Other examples of spatial indexing using an order relation are quadtree and octree (Samet 1990). Topology can also be used for spatial indexing by storing the links (for example, pointers) between data elements in the database directly. However, the storage of such information is redundant to the storage of coordinate sets based on metric relations. Problems of consistency between metric relation versus topology or order relation arise. The consistency rules must be defined and enforced to eliminate conflicts for any database operations that may change the status of the database, for example to insert, delete or modify a data element in the database. Examples defining and applying consistency rules for spatial database can be found in Kufoniyi (1995).

2.4.6.2 Spatial Analysis

Two kinds of spatial analyses may be distinguished: query-based, making preferential use of topological and order relations; computation-based, relying heavily on metric and order relations. Spatial relationships like 'touch' or 'disjoint' can be expressed in terms of metric relations. Peuquet (1986) has defined a relationship 'touch' by a distance equal to zero and never less than zero at a single location. The relationship 'disjoint' may be defined so that 'the distance from any point of object A to any point of B is greater than zero' (Egenhofer 1989). A distance relationship can be expressed in different forms, for example direction and proximity, which are commonly used in spatial modelling. Based on some referential axes, distances can be used for georeferencing in the form of a coordinate tuple and can be transformed into directions in terms of angularity. Discrete directions, for example north, east, south, west, can be used to express spatial relationships (Alia and Williams 1994). Direction may further define an orientation, for example 'from-to', useful for path finding in a network structure. Proximity can be used to represent spatial relationships like 'near', 'far' or 'within the distance of', for example in the form of a buffer zone, or distance tolerance. The distance tolerance always needs to be defined for metric operation, because of the finite state of the computer, which results in rounding-off errors. For example, intersection or touching between two straight lines may be encountered and then recorded as topological relationships in the database, if the shortest distance between the two lines is less than a predefined distance tolerance. Combining metric and order relations is typically used for the elimination of short straight line segments. The length of a straight line, that is, the distance between its two nodes, is computed and then the order relation $<$ is used to compare it with the predefined distance tolerance to decide on the deletion of this line.

Operations based on metric relationships (known as computational geometry) are time consuming if the data elements have not been organized in an appropriate structure, lacking order or topology. However, most raster operations, which are mostly simple and fast, are based on metric relationships, because the data elements have already been organized into a strict spatial order. Many relationships can be interpreted as order, for example, in front/behind, larger/smaller, greater than/less than, under/over, higher/lower, equal. For instance, the operator $\leq$ is spatially interpreted as 'is contained in' and can be used to answer a question like 'what land parcels are inside the zone A?.'

The use of topology has gained significant attention in GIS. It is used to help the navigation between data elements in the database without using sophisticated computational geometry. This is usually done by explicitly storing the topology in the database which may be initiated manually by human knowledge, or analytically derived by computational geometry (up to some accuracy). Topological relations are translated into different types of pointers from one data element to another and can therefore speed up the searching operation, because the search space is dramatically limited. For example, the incident relation 'meet' defined for polygons A and B yields an arc C that is the result of the intersection between the boundaries of A and B. The inverse relationship may be defined from the arc C to the two polygons A and B as 'left' and 'right' of the arc respectively.

$$\text{meet}(A, B) = A \cap B = C;$$
$$\text{left}(C) = A;$$
$$\text{right}(C) = B$$

The 'left' and 'right' pointers are then stored in the database directly. Navigating in the database from A to B can be achieved by starting from A and then searching for B exclusively via C.

2.4.7 Representation of Spatial Objects and Relationships

Models of the earth and geo-spatial objects may be physically created; examples are a metal or plastic globe used for teaching geography, a plaster magnet of an urban quarter. Creating a model of the earth in the computing environment requires different modelling tools. For the vector type of geoinformation, objects have to be represented in an appropriate form convenient for storage, analysis, or graphical display. Two levels of representation of spatial objects are widely used. The first has its roots in the concept of cell complex and simplicial complex (Frank and Kuhn 1986); the other in the concept of graph theory. The real world objects are mapped to different types of elements of the two representations, depending on the level of abstraction, as explained in the following sections.

2.4.7.1 Definition Level: Cell Complex and Simplicial Complex

Based on the terminology used in Moise (1977) and Giblin (1977), Frank and Kuhn (1986) described two types of elements, namely complex and simplex, that can be used for the representation of real world objects. A formalization of this concept has been provided by Egenhofer (1989). A complex constitutes a description of an object as a whole. Different types of complexes are defined by the (internal) spatial dimensions of objects, that is, 0-cell (point-object), 1-cell (line-object), 2-cell (area-object), 3-cell (volume-object), and so on. However, for any spatial dimension, there is a simplest geometric figure that can represent an object. This type of geometric figure is called a simplex. For example, every point (node), which is a geometry of dimension zero, is a 0-simplex. For spatial dimension one, a straight line segment is the simplest geometry, so it is a 1-simplex. Likewise, a triangle is a 2-simplex and a tetrahedron a 3-simplex in two and three dimensions respectively. In general, an n-simplex is the simplest geometry of dimension 'n.' A mathematical definition of simplex is as follows:

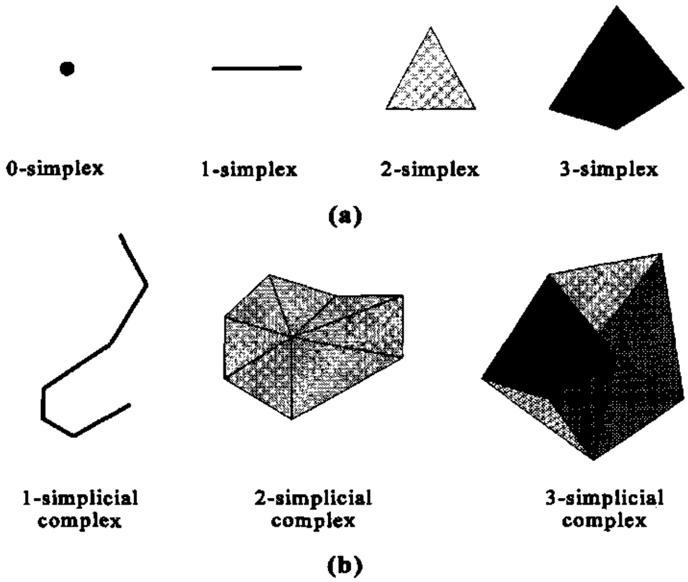


Figure 2.6 Examples of simplices (a) and simplicial complexes (b). The hull of each simplicial complex is a cell-complex.

Any simplex of dimension n, called n-simplex, is bounded by (n+1) geometrically independent simplices of dimension (n-1) (Egenhofer et al 1989).

For example, a tetrahedron is a simplex of dimension 3, that is, a 3-simplex. It is bounded by (3+1) = 4 triangles that are simplices of dimension 2. These four triangles are not components of each other and do not coincide; that is, they are geometrically independent. A triangle, that is, a 2-simplex, is bounded by (2+1) = 3 edges. Likewise, an edge is a 1-simplex bounded by two nodes that are 0-simplices. Figure 2.6 shows some examples of simplices and simplicial complexes.

If a complex object is composed of a contiguous and finite set of non overlapping simplices, it is called a simplicial complex. For example, a line object composed of a chain of straight line segments is a 1-simplicial complex. A polygon composed of a set of triangles is a 2-simplicial complex. The hull of a simplicial complex is a cell-complex. The concept of a simplicial complex is crucial for our design of an integrated data model; see chapter 4.

2.4.7.2 Description Level: Graph

A simplicial complex is a representation of geo-spatial objects at a definition level which still requires further elaboration. Understanding is usually facilitated by making an idea perceptible. Graph representation provides such a possibility. The elements of a simplicial complex can be mapped to elements of a graph which allows the idea to be visualized.

The origin of graph theory is attributed to Leonhard Euler, the Swiss mathematician, with his publication in 1736 of an answer to the question known as the 'Bridges of Königsberg' (Finkbeiner and Lindstrom 1987, Devlin 1994). It is interesting to note that the first use of a graph was to solve a problem of a spatial nature. Graph theory has been applied in many disciplines, for example, electrical engineering, artificial intelligence, information modelling. In geo-spatial modelling, graph theory has been applied to represent the topological structure of geographic databases, such as in DIME (Corbett 1979, Marble et al 1984), TIGER, ARC/INFO (1991), FDS (Molenaar 1989). It is also applied in many parts of this research (see chapter 4). Corbett (1979) has extensively applied concepts of graphs to express the topological relationships between the cell complexes that are elements of a cartographic model. Graphs have also been used as a tool to assist spatial network analysis, such as finding the optimum path, or shortest path. This section reviews some fundamental concepts of the graph.

Different graph elements have been provided for the representation of objects in the same way as elements provided by the concept of simplicial complex. Traditional elements of a graph are node, edge and face. However, these are limited to 2D representation. Additional elements of a graph are needed for the representation of objects that have a greater spatial extent than 2D; for example a 'body' element for a 3D object. In order to compare the simplicial complexes and graphs, we first look at the definition of the graph using mathematics.

2.4.7.2.1 Definition of a Graph

A *graph G* is defined by an *incident* relation between two disjoint sets *N* and *E*, where *N* is a non empty set of *i* nodes ($N = \{n_1, n_2, n_3, ..., n_i\}$) and E is a set of *j* edges ($E = \{e_1, e_2, e_3, ..., e_j\}$). If E is an empty set, a graph is called an *empty*, or *null graph*. If both *N* and *E* are finite sets, a graph is called a *finite graph*.

An edge *e* is further defined by a set of two nodes $\{n_1, n_2\}$ with *an adjacent* relationship. These two nodes can either be the same, or different. If an edge is incident with two nodes that are identical, this graph is called a *loop*. A graph that contains no loop is called a *simple graph*. If a graph contains a pair of nodes that are incident with more than one edge, the graph is called a *multigraph*. Figure 2.7 show some example of such graphs.

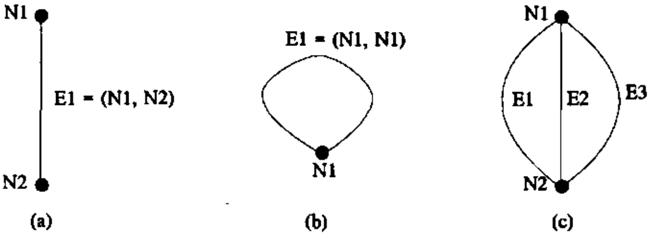


Figure 2.7 (a) a graph, (b) a loop and (c) a multigraph.

The adjacent relation can also be defined for a set of edges; the edges are said to be adjacent if all of them are incident with the same node. Here the *degree (or valence)* of a node *n* of a graph can be defined by counting the number of edges that are incident at *n*.

2.4.7.2.2 Types of Graphs

If every node of a graph is of the same degree, the graph is a *regular graph*. An edge, a triangle, a tetrahedron and a cube, are examples of graphs of this kind with nodes of degree 1, 2 and 3, respectively. A graph with *m* nodes is a *complete graph*, denoted by K_m, if each pair of distinct nodes is joined by one edge.

The degree of a regular graph is comparable to the dimension number of a simplex, as defined in the preceding section. A complete graph K_m is equivalent to a *(m-1)-simplex*. This comparison is shown graphically in Figure 2.8.

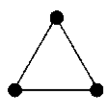
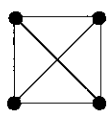
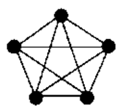
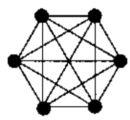
Graphics						
Simplex	0	1	2	3	4	5
Complete graph	K_1	K_2	K_3	K_4	K_5	K_6
Name	Node	Edge	Triangle	Tetrahedron	...	...

Figure 2.8 Complete graphs and simplices.

If a set of nodes of a graph *G* can be divided into two non empty and disjoint subsets *M* and *N* of *m* and *n* nodes respectively, such that an edge of *G* connects with a node of *M* and a node of *N*, this graph is called a *bipartite graph*. If each node of *M* has a distinct set of edges connecting every node of *N* and vice versa, then *G* is a *complete bipartite graph* and is denoted $K_{m,n}$ (see also Figure 2.9).

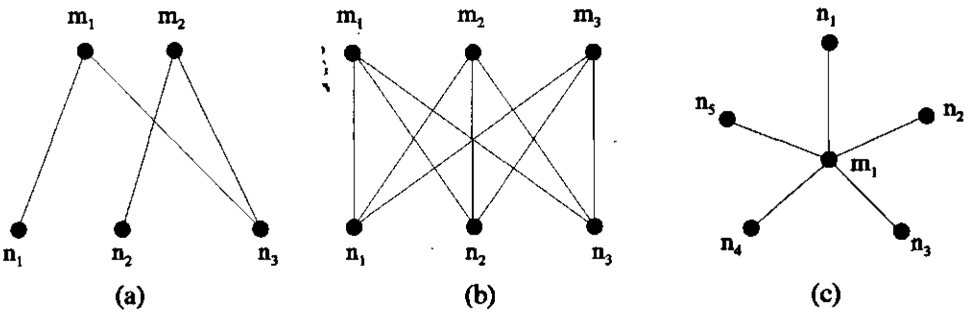


Figure 2.9 Bipartite graphs (b) $K_{3,3}$ (c) $K_{1,5}$ or star graph. Only (b) and (c) are complete bipartite graphs.

A graph can also contain subgraphs. If G_1 and G_2 are graphs with $\{N_1, E_1\}$ and $\{N_2, E_2\}$ incident relationships of nodes and edges respectively, G_2 is a subgraph of G_1 if and only if all nodes of N_2 are nodes of N_1 and all edges of E_2 are edges of E_1.

2.4.7.2.3 Similarity of Graphs

The similarity of graphs can be expressed more formally in terms of isomorphism and homeomorphism. Two graphs $G = \{N, E\}$ and $G' = \{N', E'\}$ are *isomorphic* if and only if N and N' are one-to-one correspondent and E and E' are also one-to-one correspondent.

To define the homeomorphism of two graphs, the concepts of subdivision and contraction have to be considered. Let a graph $G = \{N, E\}$ and an edge $e = \{n_1, n_2\} \in E$, then a simple subdivision G' of a graph G can be obtained by inserting a new node m of degree two on e, therefore between n_1 and n_2 and replacing the edge e with two new edges incident to two pairs of nodes $\{n_1, m\}$ and $\{m, n_2\}$. If G' results from a sequence of one or more simple subdivisions of G, G' is called a subdivision of G. Conversely, if a graph $G = \{N, E\}$, and two edges $e_1 = \{n_1, n_2\}$, $e_2 = \{n_2, n_3\} \in E$, then a contraction of G can be obtained by deleting the node n_2 (of degree two) and consequently replacing e_1 and e_2 by the new edge $e = \{n_1, n_3\}$. It is possible to perform this kind of contraction for every node of degree two of a graph. Subdivisions and contractions are inversions of each other.

The two graphs G and G' are said to be homeomorphic if they are isomorphic, or if G' is either a subdivision or a contraction of G. The concept of homeomorphism is useful for the generalization of some aspects of graphs. Some examples of isomorphic and homeomorphic graphs are shown in Figure 2.10.

An intuitive approach can be used to verify the isomorphism of graphs: if two graphs G_1 and G_2 have different numbers of nodes or edges, or if none of the nodes of G_1 graph are of the same degree as any node of G_2, then G_1 and G_2 are not isomorphic.

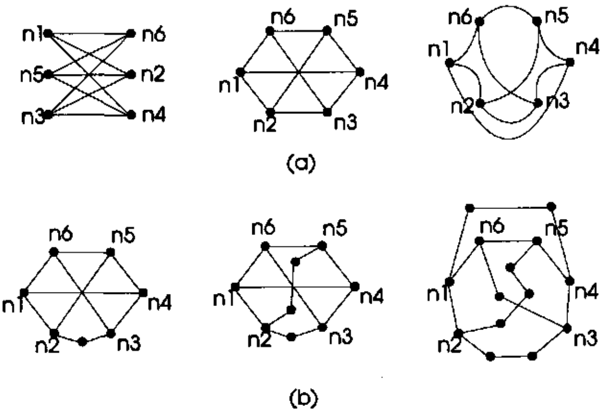


***Figure 2.10** (a) Graphs that are isomorphic; (b) Graphs that are homeomorphic to (a).*

2.4.7.2.4 Connectivity of Graphs

One aspect of graphs frequently applied to spatial modelling is connectedness. A graph can be *connected* or *disconnected*. For a *connected graph*, every edge belonging to a walk *W* that visits all the nodes and edges of the graph also belong to the graph, while the same kind of walk for a disconnected graph cannot be defined without introducing an extra edge that is not an edge of the graph. It can also be said that a connected graph has only one component, while a disconnected graph has two or more disjoint components. For example, a cadastral map may be regarded as a set of graphs used to represent different objects, such as houses and land parcels. If some individual houses situated as islands inside land parcels are contained in the map, it can be thought of as a disconnected graph. In this thesis, the number of disjoint components is expressed as *the degree of isolation*. This aspect is also important in the generalization of Euler characteristics of graphs (see chapter 4).

2.4.7.2.5 Planarity and Non Planarity of a Graph

The planarity of the graph is also commonly applied for checking the internal geometric consistency in a 2D-based spatial model (see Laurini and Thompson 1993); for example, an edge must have two different nodes stored in the same database, and a triangle must have three edges. By definition, a graph *G* is planar if and only if it is isomorphic to a *plane graph* that can be drawn on a plane with no crossing edges. Another definition states that a graph is planar if and only if it can be embedded on the surface of a sphere (Wilson 1985). The latter definition allows a planar graph to be embedded in a 3D Euclidean space so that the edges that seem to cross each other when embedding onto a plane can be placed separately around the surface of a sphere.

The algebraic approach to verifying whether a graph is planar or not is applying the Euler equality.

Theorem (Euler): Let G be a connected, plane graph with n nodes and e edges. Then G should satisfy the equation

$$n - e + f = 2$$

where f is the number of non overlapping regions separated by G. Note that each face of a planar graph is bounded by a simple circuit if the face is finite. The right side of the equation is called the *Euler characteristic* of a planar graph.

It is important to note that the outer (infinite) region has to be counted as a face; otherwise, the formula becomes

$$n - e + f = 1$$

Also, to cover a disconnected graph with i components, the Euler equality has been generalized as follows (see also Wilson 1985):

Theorem (generalized form of Euler's theorem) : Let G be a planar graph having i components, n nodes, e edges, and f faces. Then G should satisfy

$$n - e + f = i + 1.$$

The outer region should be counted only once. If it is not counted, then the formula becomes

$$n - e + f = i.$$

This generalization is equal to the summation of all results where the Euler formula has been applied to each separate component.

Another way of verifying the planarity of a graph is through checking its non planarity condition by applying Kuratowski's theorem.

Kuratowski's theorem: Any non planar graph contains a subgraph homeomorphic to K_5 or $K_{3,3}$.

K_5 is a complete graph with each node of degree 5; $K_{3,3}$ is a bipartite complete graph with each node of degree 3 (see Figure 2.8 and Figure 2.9 respectively).

2.4.7.2.6 Dual Graphs

The concept of a dual graph is also applied in spatial modelling. The most commonly known are the graphs that represent Delaunay triangulation and Thiessen polygons (Delaunay 1934, Thiessen 1911). Consider a graph G that has been embedded onto a plane. G has n nodes, m edges and f faces and G' has n' nodes, m' edges and f' faces. G' is the (geometric) dual of G if:

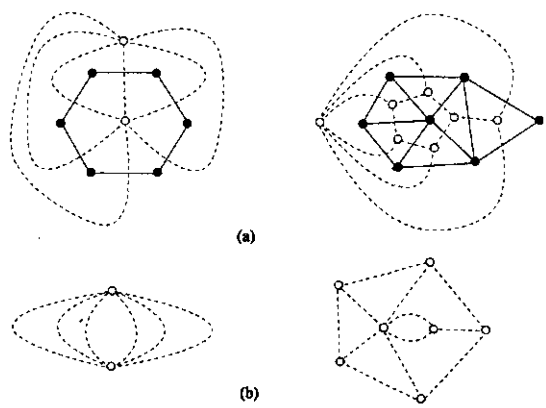


Figure 2.11 (a) Graphs and their geometric duals that are isomorphic to graphs in (b).

(i) $n' = f$, $m' = m$ and $f' = n$ and such that each node n_i' of G'

can be mapped inside each face f_i of G in a one-to-one correspondent manner, and

(ii) each edge e' of G' crosses a corresponding edge e of G, but no other edges of G, where e' joining two nodes of G', n_i' and n_j' that, respectively, lie inside two faces of G, f_i and f_j both adjoining e.

Observe also that any graph dual to G is isomorphic to G'. However, if G and H are isomorphic, G' and H' need not be isomorphic. Examples of graphs and their duals are illustrated in Figure 2.11.

2.4.8 Spatial Data Models in GIS

Based on the conceptual model of the real world presented in section 2.4, a number of spatial data models have been derived, for example DIME, ATKIS, TIGERS. In geoinformation science, data models may be categorized according to dimensionality, representation of space and thematic representation.

2.4.8.1 Multi dimension

If the aspects of reality to be modelled involve objects of various dimensions, the spatial model should provide the highest spatial dimension, so that all objects can be accommodated. For example, the data model may contain point, line, surface and body features whose dimensions range from zero to three. This kind of data model is regarded as multidimensional. The dimension number of the model is the highest dimension of the objects it can contain. The 2D data model is the most commonly used at present, but demand for 3D and 4D is increasing. The 3D data model may be regarded as equivalent to the static world, without any change over time. If the temporal component is also taken into account, the dimensionality of space is then increased to 4D, as found in the modelling of dynamic processes.

2.4.8.2 Tessellation

Tessellation is a complete and continuous subdivision of space into spatial units that may be of either regular or irregular shape. The model of the real world constructed by this approach may be regarded as a tessellation-based model. In 2D we know several types of regular tessellations, for example squares, rectangles, hexagons, equilateral triangles, where squares are the most commonly used, for example for the storage of digital images and surface models. For irregular tessellation, triangles and Thiessen polygons are the most commonly used units. Triangular irregular networks are also frequently used for the storage of surface models, while Thiessen polygons are used to represent influence zones and the proximity of objects. Both irregular tessellations are important in this research and are used to design the integrated data model as detailed in chapter 4. We therefore elaborate here on the irregular tessellations.

Two kinds of irregular tessellation are distinguished. The first is tessellation by complex geometry; the second relies on simplex geometry (see section 2.4.7.1 for the definition of simplex and complex).

2.4.8.2.1 Tessellation by Complexes

Tessellation by complexes is achieved by subdividing space into a set of cell complexes, for example, polygons and polyhedrons. The most important in this thesis is Dirichlet tessellation (Dirichlet 1850), or Voronoi tessellation (Voronoi 1908), based on the proximity of objects. If objects are represented by kernel points, a Voronoi region encompasses a set of points closer to a kernel point than to any other point in the set. This kind of tessellation can be applied to any dimension.

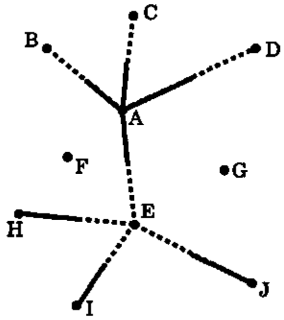


Figure 2.12 1D Voronoi tessellation.

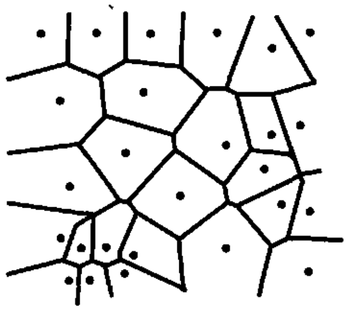


Figure 2.13 2D Voronoi tessellation.

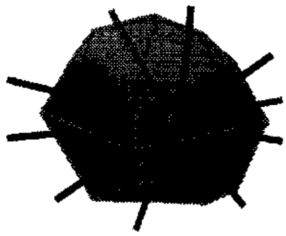


Figure 2.14 An example of Voronoi polyhedron.

Figure 2.12 is an example of 1D Voronoi Tessellation applied along any straight line connecting two points. The influent zone of each kernel point covers the distance from the point up to the middle of each line emanating from the point, and connecting to the other point. For example, between the points A and C connected by a straight line, the influence zone of A is expressed by the dark, solid part of the straight line AC, while the influence zone of C is expressed by the dotted part. Figure 2.13 is an example of a 2D Voronoi tessellation, also known as Thiessen polygons (Thiessen 1911). Voronoi polygons are shown in thick lines, while kernel points are shown as small black circles. A 3D Voronoi polyhedron may look like Figure 2.14.

2.4.8.2.2 Tessellation by Simplices

Tessellation by simplices is done by subdividing space into a set of simple objects, for example triangles and tetrahedrons. For 2D, the tessellation by triangles is known as a triangular irregular network. Its extension to 3D is a tetrahedral network (Figure 2.16). Different kinds of TINs are distinguished by the extent to which they have been considered important in this thesis (see chapter 6 for more details). Delaunay triangulation is the most popular method for TIN construction. A large number of publications have already discussed this topic in depth (Delaunay 1934, Sibson 1978, Tsai 1991; see chapter 6). With respect to graph theory, Delaunay triangulation and Thiessen polygons (2D Voronoi) are

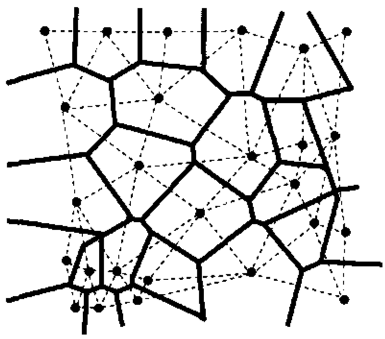


Figure 2.15 Thiessen polygons and Delaunay triangulation.

geometric duals and each can readily be derived from the other. Figure 2.15 shows both Thiessen polygons and a Delaunay triangulation in solid lines and dotted lines respectively. Likewise, Delaunay tetrahedral network and Voronoi polyhedrons are geometric duals. Figure 2.14 shows lines emanating from the node inside the Voronoi polyhedron; they are the edges of tetrahedrons.

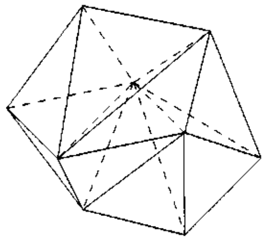


Figure 2.16 A network of tetrahedrons.

Delaunay triangulation has been used to facilitate spatial computation, for example in DTM for the derivation of contour lines, interpolation of height at given planimetry, and computation of slope and aspect.

2.4.8.3 Single-theme and Multi-theme

Two kinds of data models can be distinguished, based on the representation capability of a geometric element of the data model. If a geometric element is part of the representation of only one real world object (of one thematic class), the data model is regarded as a single-theme data model (Molenaar 1989). If a geometric element is part of the representation of more than one real world object (of more than one thematic class), the data model is regarded as a multi-theme data model (Kufoniyi 1995).

2.4.8.3.1 Single-theme

Molenaar (1989) suggested a spatial data model for vector representations of geo-spatial objects. What he called the formal data structure (FDS) of a single-valued vector map (SVVM) geometrically abstracts spatial objects, such as monuments, roads, rivers, forests and parcels to points, lines and areas. A terrain feature is described by an identifier, its geometric primitives (arc and nodes), and its (single) thematic class. Figure 2.17 shows the graph representation of this conceptual model. Although the coordinates of every node can be 3D, the model provides only 2D topology. The FDS approach provides a highly disciplined approach to geoinformation modelling. The data model is presented by a diagram and a set of conventions that provide the rules for modelling. The conventions help to prevent confusion during modelling and

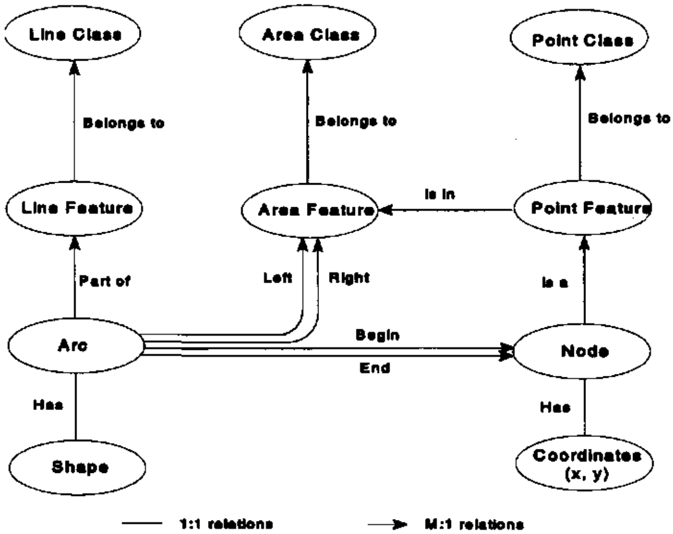


Figure 2.17 A single-theme data model (after Molenaar 1989).

subsequent processes when operations need to be applied to the data set and facilitate formulating consistency rules for updating a geo-spatial database.

2.4.8.3.2 Multi-theme

A multi-theme (or multi-valued) vector map (MVVM) refers to the integrated representation of geo-spatial objects from more than one theme in the form of point, line and area feature types in a database. A graphical representation of the multi-theme data model is shown in Figure 2.18.

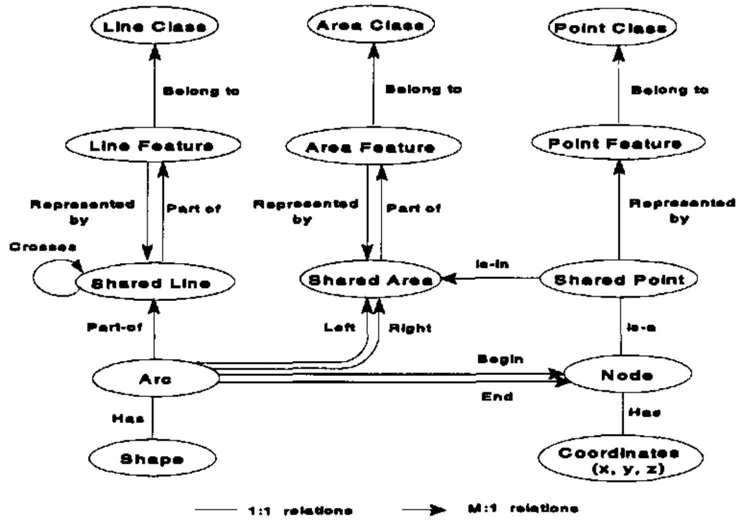


Figure 2.18 Data Model for Multi-valued Vector Maps (Kufoniyi 1992).

The multi-theme data model extends the concept of single-theme by adopting the idea that many spatial objects can overlap by sharing the same subspace. In the data model they can share the same geometry, because a reality can be viewed differently by different observers, or for different purposes. For example, soil and water can coexist at the same location during flooding, or for the analysis of the moisture content of the soil. The multi-theme data model provides additional geometric data types to group the geometric primitives shared by more than one feature. Since this group belongs to different features with different themes, it is said to have a heterogeneous (thematic) property. The feature itself belongs to only one theme, so it is said to have a homogeneous (thematic) property.

2.5 Logical Design of Geo-spatial Model

The logical design defines all the data elements needed for the representation of each spatial object. It outlines the method for the translation of the data model from the conceptual level to a logical level. Three kinds of logical data models are commonly used in information science: hierarchical, network, and relational. The hierarchical structures are those tree like structures where a record is linked to another record via a 'parent and child' relationship. Based on the strict ordering of relationships, the structure is an acyclic directed graph in which no recursive relationship is allowed.

Less strict than the hierarchical is the network structure, because it allows any kind of link between data elements, provided there are elements at both ends of the link. The operations using network data structure are fast and efficient, but the design, implementation and maintenance of this data structure are rather difficult.

The relational structure is more closely related to natural ways of thinking. Regardless of performance, relational structure offers the quickest and easiest way to logical design. Another important logical design which has become popular during the last decade is based on the object-oriented approach. The object-oriented approach extends the hierarchical and network structures by encapsulating related operations as additional attributes within each record. Logical design based on the object-oriented approach is rather involved, since we must also take into account the related operations. Only the relational and object-oriented approaches are reviewed here, since they are used in the design of a unified data structure (UNS) in chapter 5.

2.5.1 Relational Approach

A relational structure is a collection of relationships between data elements representing aspects of reality based on set or relational algebra (see Date 1986, Howe 1989, Martin 1983). The logical design of a spatial model based on the relational approach can be achieved by organizing the representations of spatial objects and relationships between them into a set of tables consisting of rows and columns (that is, records and fields). Figure 2.19 is an example of the rélational data structure of the SVVM shown in Figure 2.17.

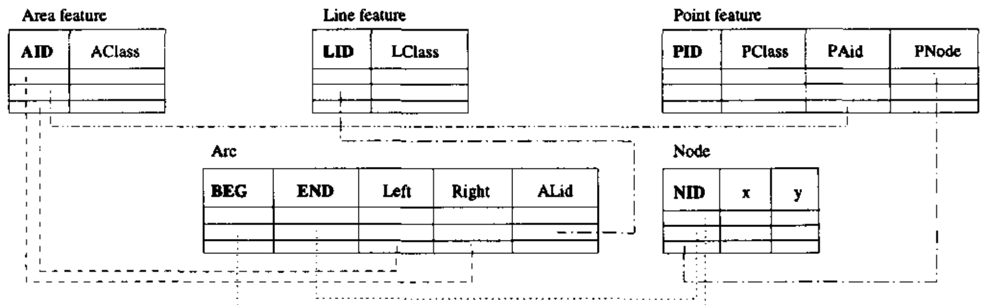


Figure 2.19 Relational data structure of SVVM (adapted from Kufoniyi 1989).

Point, Line and Area feature tables contain relationships between feature level and class level. Point, line and area features are represented by identifiers, that is, PID, LID, AID, whereas classes are represented by PClass, LClass and AClass in their respective tables. Neighbourhood relationships between each arc (indicated by BEG and END nodes) and its left and right area features are presented in the Arc table. The field ALid in this table indicates to which line feature each arc belongs.[2]

Since the relational data structure consists of several tables, a Cartesian product— a set of possible combination of rows of different tables— is used to further derive relationships between data elements across the different tables being joined together. For example, to know the coordinates of the beginning and end nodes of an arc, the tables Arc and Node are joined together. The search is carried out on the Cartesian product of these two tables to find the match between the values of BEG or END with the values NID. If the match is found, the coordinate X and Y are obtained.

It is important for this kind of data structure for all relationships to be optimized to prevent anomalies occurring with updating. Normalization is the optimization process that can be carried out stepwise, that is, from the first to the fifth normal form. In general, the third normal form is found to be acceptable. Several approaches for normalization exist, for example, non loss decomposition, entity-relationship approach (Chen 1983), dependency diagram (Smith 1985). The last mentioned is considered to be the most intuitive approach.

2.5.1.1 Normal Forms

The first normal form (1NF) is obtained after eliminating repeated groups. The second normal form (2NF) is obtained by elimination of the non identifier attributes which are not functionally dependent on the whole key. The third normal form (3NF) requires the elimination of all the functional dependency between non key attributes. The fourth normal form (4NF) deals with multi-valued dependency that occurs when an attribute value can be inferred from another record. The fifth normal form (5NF) deals with the joint dependency that occurs when some facts are stored twice in the same table. This situation may be regarded as a result of joining two pairs of attributes in a record with one attribute in common. Hawryszkiewyc (1991) discusses and provides examples of this issue.

2.5.1.2 Smith's Normalization

Normalization is commonly achieved by decomposing a preliminary table into first, second, third, fourth, and fifth normal forms; in this way, several normalized relationships in the form of tables are obtained. This seems, however, to be a very tedious task. Smith (1985) proposed a more attractive approach, whereby tables are composed from a dependency diagram. If the procedure is followed correctly, the database tables obtained are then fully normalized. The steps can be summarized in five phases:

1) identifying all the data elements
2) constructing dependency statements
3) mapping from dependency statements to dependency diagram

[2] For further elaboration about relational data structure of SWM see Bouloucos et al (1990).

4) composing relational tables from the dependency diagram
5) improving the handling of the relational table by introducing a surrogate key if necessary.

Roessel (1986) has demonstrated the applicability of this method to a spatial database. The same approach has also been used by Kufoniyi (1989), Bouloucos et al (1993), Ayugi (1992), Pilouk and Tempfli (1992), and Chhatkuli (1993), because it helps clarify and create more understanding about spatial relationships. The normalization would also be useful for the transition into object-oriented database structure, where the same kinds of relationships have to be represented. This approach will be applied in chapter 5.

2.5.2 Object-oriented Approach

The object-oriented approach was developed about two decades ago, starting from the developments of the object-oriented programming languages Simula 67 (Dahl et al 1970) and Smalltalk (Goldberg and Robson 1983). Object-orientation has been built up from the idea of encapsulating data and operation, and processing the data together. The confusion about which data must be processed by which procedure is thereby avoided. Furthermore, computer codes need to be reused, since an algorithm may be applied to different types of data in different applications. The object-oriented approach provides mechanisms allowing economic reuse of a computer code by extending the code to accommodate additional types of data without a major reprogramming effort. Moreover, it also provides abstraction mechanisms to natural model spatial objects.

2.5.2.1 Encapsulation

This is a mechanism tying together the attributes describing the state of the object and the operations retrieving information about the object, or changing the state of the object. These operations are also known as an object's behaviour, method, and dynamic properties. Encapsulation provides the object with the control determining which attributes and behaviours would be private properties and which accessible to the public. Encapsulation can ensure that an operation is applied to the right object, thus preventing ambiguity during operation. This is also known as a type-safe operation.

2.5.2.2 Classification

Each object needs to be organized into a certain class. Objects with the same kinds of properties and behaviours should be placed in the same class. Each object is said to be an instance of a class. A class is a place for defining the specification of an object. A class is said to be an abstract data type (ADT) in the sense that we have an opportunity to create new data types that fit our abstraction about a real world object.

2.5.2.3 Inheritance

When an object has been defined, the inheritance mechanism permits the propagation of the properties and behaviours to lower level objects in the same hierarchy. The propagation from one object to another object is allowed if the two objects have 'parent' and 'child' relationships. The child inherits all the properties and behaviours from the parent through a class mechanism. The inheritance is activated by deriving a new class from the existing class. If the new class is derived from only one existing class, it is known as single inheritance; it is called multiple inheritance if the new class is derived from more than one existing class. The inheritance makes the existing class reusable and thus significantly saves time in redesigning a new class, provided that the new and existing classes have something in common. If the inheritance is defined at the logical design stage, it is called static inheritance. If the inheritance needs to be defined during the construction of spatial model, it is called dynamic inheritance (Weiskamp and Flamig 1992). Dynamic inheritance is needed when multiple representation of an object is required and the representation is not known prior to the construction of the spatial model. For example, a city may be represented as a point, or as an area feature highly dependent on the user. To make the logical design flexible, the type of representation can only be decided upon during the construction of the spatial model.

2.5.2.4 Generalization and Specialization

If two or more classes have many properties and behaviours in common, a more general class can be created to become their parent. Conversely, when deriving many new classes from an existing class, each class may have, in addition to the parent, more specific properties and behaviours. The former scheme is generalization and the latter is specialization. These two mechanisms permit the streamlining of classes, making them easier to maintain. Generalization and specialization create a class hierarchy using an inheritance mechanism. The generalized class exists at the higher levels of the hierarchy, while the more specialized class exists at the lower levels of the same hierarchy. The complexity of objects increases from higher to lower levels of the hierarchy.

2.5.2.5 Aggregation

The design of a class needs to include many data types and behaviours. An aggregation process is involved. Each data type aggregated into a class may be either system or user defined. Different existing classes may be aggregated to build up an aggregated class. Objects of existing classes are components of a composite object in the aggregated class. The relationship between each component and the composite object is of the type part-of. The aggregated class does not inherit properties or behaviours from its component classes; neither is it a parent of each component class. The properties and behaviours of each component are normally inherited from their respective parents. Relationships between components are well defined. Each component is necessary for the constitution of the composite object. The composite object cannot be independently constructed. The aggregated class should have its own behaviours. The behaviours of the components are normally suppressed by the aggregated class.

An example of an aggregated class is the object 'car'. It is composed of objects from the classes wheel, engine, steering, and so forth. A car does not inherit properties or behaviours from, say, the wheel or the engine. A car resembles neither a wheel, nor an engine. A car is not complete if the component wheel, or engine, is missing.

An example in GIS can also be given. A line feature is a composite object consisting of a list of nodes. These nodes must be arranged in proper order; each pair of nodes defines a straight line segment that constitutes the geometry of the line feature. If the nodes are badly arranged, or if one of them is missing, the line feature cannot be correctly constructed.

2.5.2.6 Association

In many cases, an object can be a container, or a collection, of many other objects. The association defines a membership relationship between a group, or a set, of lower level objects (members) with a higher level object—the container. Relationships among member objects are not always clearly defined, but the relationship between each member object and the container must be clearly defined. The association is useful for the design of container classes, for example, array, linked-list, stack, or queue. The member objects need not be of the same class. Although the container in computer science is commonly used to contain the same type of objects, as for example in an array of integers, in reality a container can contain many different objects. Two kinds of container exist; homogeneous and heterogeneous. A container is complete in itself and can be independently constructed without the member objects. In other words, a member object need not constitute, nor be part of, the container. For example, an arc container may be an array containing a set of arcs. This set of arcs does not constitute the container. Whether or not this set, or subset, of arcs constitutes a line feature is not important for the container. The container's only interest is whether an arc resides within the container or not. With respect to this point, a container's own behaviour lies very much in the management of the membership of objects. The behaviours of each individual member object are normally suppressed, while the group behaviours are promoted.

The use of association and aggregation are easily confused, because both of them relate to many objects at a lower level. We can consider the case of a line feature as an example to decide whether association or aggregation of a set of arcs is more suitable. In the spaghetti model, where the relationship between any two consecutive arcs of a line feature is not considered important, the line feature may be treated as a collection of arcs. Here, the association may be adequate. In a topological model like SWM, where traversing along the line feature to support network analysis should be possible, aggregation is more suitable because the relationships between two consecutive arcs are also considered important.

2.5.2.7 Polymorphism

Polymorphism allows two or more classes to use the same attribute name without confusion about the class to which the attribute belongs. This is done by associating an object with its attribute. For example, an area feature and a line feature can have the same attribute 'length.' Length, as an attribute of the area feature, is the perimeter of the area boundary, whereas it is simply the length of the line feature. Another kind of polymorphism is called parametric polymorphism, whereby the class name is used as parameter to make the general design specific (Pohl 1993). The general design of a class is known as the template class. General dynamic behaviour can also be provided as a template function.[3]

2.6 Summary

Some fundamental and theoretical concepts related to spatial modelling have been reviewed. These concepts are important for the understanding and development of a spatial model in the different phases. Two phases of spatial modelling are distinguished; design, and construction. The maintenance phase, keeping the constructed spatial model valid, is not covered in detail here. On the basis of the scope of involvement by different disciplines in the design of spatial model Molenaar (1994b) describes, only the conceptual and logical design are within the scope of geo-information theory. The detailed internal design is left for computer scientists; it is outside the scope of this thesis.

For the conceptual design, mathematical concepts about space and relations, that is, metric, order, topology, are considered important. Concepts of simplicial complexes and the theory of graphs are used to represent real world objects in the spatial model. Existing spatial data models resulting from different conceptual design approaches, that is, SWM, MWM, irregular tessellation by simplices, are considered fundamental to the modelling approach used here. For the logical design, relational and object-oriented approaches are fundamental to the design of spatial data structures.

The elements of spatial theory provide a sound basis for the design of an integrated geo-spatial model that permits the representation of both determinate and indeterminate spatial objects. SWM, for example, is a model representing determinate objects. Simplicial complex and Delaunay triangulation provide fundamental concepts for the representation of indeterminate spatial objects. The mathematical concepts provided in this chapter can be used to support the conceptual design for both FDS and integration approaches. FDS can be formulated using cell complexes, or simplicial complexes. For example, a face is a 2-cell complex which can be decomposed into a 2-simplicial complex by using the Delaunay triangulation method. The same mathematical concepts can be used to evaluate existing data models, whether the provision of the elements for the integrated modelling of reality is sufficient or not.

[3]Further detail about object-oriented concepts can be found in, for example Alagic (1989), Egenhofer and Frank (1989), Webster (1990), Orenstein (1990), Kemp (1990), Worboy et al (1990), Molenaar (1993).

3
SYSTEMS FOR INTEGRATED 3D GEOINFORMATION

A 3D geoinformation system (3D GIS) that integrates all the necessary elements into a 3D spatial model, and provides functions efficiently to create and utilize the 3D spatial model, must be constructed to fulfill several needs from the earth science community (see chapter 1). Using a tool set provided by a 3D GIS, a 3D spatial model can be constructed to permit the virtual performance of many tasks that would otherwise be carried out in reality, or may be too difficult, expensive, or destructive to carry out at all. A high quality 3D spatial model that represents well all the relevant aspects of reality—that is to say, real world objects of interest and the relationships between them—is needed to make the results obtained from operations on the model comparable to actual operations in reality. The richness of operations, which reflects the functionality of the system, depends on the complexity of the spatial model. The spatial model is the decisive basis of the system. In present practice, many GISs tend to be developed by combining different spatial models and related functions, such as those for modelling terrain relief and 2D terrain characteristics with thematic aspects (for example, colour, ownership, price). This combination is achieved by using the digital terrain model (DTM) subsystem and typical current GIS software. The approach implies a spatial model whose components are still separately stored and loosely related. Consequently, the model is not robust and may not represent the real world aspects well.

In general, a 3D GIS should aim to integrate all the necessary elements of the spatial model and functions to create and utilize the spatial model efficiently. This chapter therefore analyses the architectures of GISs in order to assess their present status and development trend, to assist the definition of the direction of the related research and stimulate the development of 3D GIS that can readily be used. The first part of this chapter reviews some general aspects of systems for integrated 3D geoinformation, their functional components, and some related and available technological developments to support their functionality with respect to the user's perspective. The main concern is system usage, investment, maintenance, and productivity and the reliability of information. With respect to an integrated 3D spatial model as a basis of 3D GIS, besides taking into account present developments in information and communication technology, four evolutionary stages of system architecture for 3D GIS are distinguished, ranging from loosely to well-constructed systems. It is also helpful to identify the technological developments to which the present thesis contributes. The last part of the chapter reviews current attempts in conceptual design aiming at a robust 3D spatial model with the potential to be the basis of a 3D GIS. This review helps evaluate the necessity for further development of the more integrative data model.

3.1 General Aspects

The different stages of the design explained in chapter 2 lead to the actual implementation of a geoinformation system used to produce a 3D geo-spatial model in the form of a database which contains a selected set of aspects of the reality. In addition to the production role of a spatial model, a GIS should also play an interface role between human users and the database. This role includes data entry, information query and processing, and presentation. Data entry translates human knowledge into a component of the model conforming to the data model and data structure. Information query performs the reverse operation, by translating the electronic components into information made understandable to users by the presentation mechanism.

Another important role of the GIS is the maintenance and management of the database. The GIS should make updating the database possible. The management role is the handling of various requests from users and the activation of the appropriate operation on the database in response to a user's request.

Regarding integrated geoinformation, two issues require consideration: first, the spatial model itself; it must reflect all those aspects of reality that are relevant for the intended use of the GIS. The second is whether the system offers all the means needed to utilize the spatial model— a matter of system functionality. Various technologies must be adopted to serve various functions; for example, photogrammetry for acquiring some components of a spatial model, computer aided design (CAD) for graphic construction, editing and visualizing the spatial model, database technology for managing the spatial model, virtual reality for natural exploration of the spatial model, and so forth. Effective exploitation of the technological development requires the spatial model to:

- include three-dimensional (3D) aspects of reality
- allow both direct and indirect representation of the determinate and indeterminate spatial objects respectively, at the desirable level of abstraction
- be capable of accommodating data from various sources and seamlessly integrating them into one spatial model.

The first requirement results from the fact that only limited spatial analysis of our 3D real world was possible until we had a 3D model. The direct representation referred to in the second requirement is suitable for determinate spatial objects. The original observations must be maintained by the model, so that knowledge about the reality is not degraded. Such a postulate favours the vector structure as a suitable basis of the model because the vector structure is resolution independent. Regarding the indeterminate spatial objects, their geometry cannot be directly represented in the model. An indirect representation must be provided through the spatial units formed from the proximal neighbours (which are, in fact, the direct representations of determinate spatial objects). Information about the indeterminate spatial objects can be obtained from the model by means of the classification or interpolation of property values of the neighbours, using those spatial units. Various forms of the (direct) representation of determinate spatial objects (that is to say, in the form of points, lines, surfaces, bodies) should contribute to the derivation of the spatial extent of the indeterminate spatial objects. The relationships between the neighbours given by the direct representation must be used as constraints to make the derivation result accurate.

The third requirement points out the necessity of multiple representation in semantic terms (an object may have different meanings to different observers) and dimensionality (ranging from

0D to 3D). The spatial database of a GIS should have the capability of storing and maintaining multiple representations. Integrating data from various sources implies that redundancy must be minimized, and human intervention is probably required to decide how uncertainty should be resolved. Moreover, to achieve a spatial model that represents well the relationships between real world objects, the underlying data structure must permit the unification of various types and components of representation of spatial objects (Pilouk and Tempfli 1994, Pilouk et al 1994).

The system should be capable of performing both query-based and computation-based complex spatial analyses across different themes and dimensions and enable the presentation of information from arbitrary viewpoints with realistic visualization. Such requirements go beyond what 2D GIS can offer. The query-based spatial analysis is applicable to the direct representation, while the computation-based analysis facilitates the derivation of information predicting a situation in reality. This information may be fed directly into the model again to avoid repeating time consuming derivation processes.

The next section considers the functional aspects of the system for integrated 3D geoinformation.

3.2 Functional Components

The requirements listed above are supported by the five major functional components of a system for integrated geoinformation outlined below. This outline may be derived from existing 2D GISs and other systems that deal with 3D spatial modelling, such as CAD, DTM (Maguire and Dangermond 1991).

(1) Data acquisition
As a result of the limited capability of each sensor and for economic reasons, different sensors are used to acquire different types of data where georeferencing systems may not be the same and accuracy can vary. Problems of uncertainty arise when data from different sources have to be integrated.

There are different stages of data acquisition:

- primary; the acquisition of raw data taken from the real world directly, for example, from terrestrial surveys, aerial photography, satellite imagery by remote sensing, synthetic images by radar, GPS, laser profiles
- secondary; the extraction of information from the raw data or other existing documents, for example, by 2D map digitizing, or 3D digitizing in a stereo model, automated object recognition and reconstruction, interpretation, and classification.

Data input may be on-line if the data acquisition device can directly transmit data to the system database during acquisition, or off-line if the data is recorded in temporary storage and transmitted to the system database at a later stage. The system should provide adequate input channels for the different types of data acquisition.

The collection of attribute data must also be included at this stage. Such compilation may be possible during the information extraction from raw data if there is some prior knowledge about the objects; otherwise, field work may be necessary for collecting thematic data directly from the reality.

3D data may be obtained by means of measurement, for example, terrestrial surveys, GPS, laser profiles, stereo digitizing, or by computation, for example, image matching, and monoplotting and should be georeferenced.

If the geometric measurements are carried out by a human operator, it is beneficial to use a topological structure for data storage. A topological structure provides an opportunity for the operator to acquire directly the spatial relationships between the spatial objects, and also to verify these relationships with the representation in the database at the time of data acquisition. Direct recording topology is more reliable than creating the relationships in a later stage. One thing for certain is that the creation of 3D spatial relationships based, for example, on metric computation is rather complicated. As yet, there are few algorithms available to automate this process; some are still in the process of research and development. The studies of 3D data acquisition by Bric (1993) and Wang (1994) show that a systematic approach that also captures some of the initial spatial relationships realized by a human operator during the time of data acquisition can significantly simplify the creation of a more complete topology at a later stage.

(2) Data structuring
Regardless of the data format, the data set obtained from the data acquisition must be transferred to this component for structuring in a proper format. The database management system must be able to recognize this format. It must also comply with the conceptual, logical, and internal design. The data structuring in the context of integrated 3D geoinformation is rather complicated and needs a more complete tool-set, for example, 3D computational geometry. This component should also make it possible to edit and clean the data before the final creation of the necessary topology between the data elements that represent the components of spatial objects. Experience from the prototype of TREVIS (ThREe-dimensional Vector Information System; Bric 1993, Bric et al 1994) and some studies about 3D data acquisition using the photogrammetric approach (Wang 1994) have pointed out that a 3D editing functionality is needed. This 3D editing function can become very efficient if it is supported by topology and advanced 3D visualization, for example, with hidden-line surface removal, or stereo (immersive) display.

(3) Data storage and management
Data organized in the required structure must be maintained in the storage media as a database. Many databases may be stored within one system. A subset of real world aspects is represented by a database in which the relationships between the data elements created during data structuring must be kept valid and up to date. The overall management of each database and its elements is the responsibility of the DBMS. This should provide the tools to deliver data to the client and ensure the validity and consistency of the database, especially when the state of the database has been changed by such operations as insertion, deletion, or modification of data elements. Apart from the task of maintaining the main database, the management of database views and indices for specific applications is also the responsibility of this module. In

3D GIS, this management task is one of the most important, because many functions, for example, hidden line and surface removal, and shading, exploit specific data structures to speed up the operations. The typical solution is to perform data conversion to reorganize the data into the required structure for each operation. This, however, creates significant redundancy and may need double storage capacity. A better solution might be to turn the operation-oriented data structures into database views or indices that consist of pointers to the main database. The consequence of this approach is the problem of maintaining the validity of the views and indices when the state of the main database changes. Integrity and consistency rules play an important role in this case.

(4) Data processing
Data processing with respect to 3D modelling is far more complex than in 2D. A large number of data processing tasks, spatial and non spatial analyses, must be handled by this component of the system. Such simple tasks as the searching of data components using certain criteria and retrieving them from the database are known as query. Other processes might involve complex computation or classification, for example, coordinate transformation, computation of slope from DTM, interpolation of contour lines or surfaces, computation of mean population per unit area, cartographic generalization, flow accumulation, or shortest path determination.

(5) Data output and information presentation
After obtaining the results from data processing, the information needs to be sent on as output to other media, or presented in an appropriate way. This output component of the system should provide a high quality cartographic and presentation capability to portray the information to users. The information may be in the form of graphics, tables, or reports. The graphic presentation of 3D geoinformation usually needs further processing to obtain an acceptable level of realism. These processes include perspective transformation, surface illumination, hidden line-surface removal, or texture mapping. Since the display media are still limited to 2D— paper, or a monitor screen— the final display still has to operate in a 2D manner, for example, by drawing pixels, line segments, and filling polygons.

Each of these output components of a system has been separately developed in the past and somehow many of these components have become disciplines of their own. The next section reviews the pertinent technologies related to each component.

3.3 Technological Supporting Functionality of 3D GIS

The emerging technologies related to 3D GIS are reviewed in this section. Those needed for common operations or routine tasks should be regarded as 'core components' of GIS. The others that strengthen and extend the functionality of a component GIS for special tasks may be regarded as 'supplementary components'. Justifying which of the following should be considered core components and which supplementary depends on the major application of GIS.

- Map digitizing, although only providing 2D information, is still an important technology for data acquisition provided that the third dimension is supplied by some other means, for example, by DTM, architecture, or engineering drawing, or a geological map showing the depth or thickness of underground strata. The completion of 3D coordinates for an object represented by a surface may be achieved by interpolating height from the DTM at every planimetric location. In many representations, especially in the case of man made objects of regular shape, like buildings, the 2D components like the footprints of the buildings, or their roof outlines shown on the 2D map, are still crucial information requiring further, but rather simple 3D completion by supplementing data in the form of height and elevation (Gruen et al 1993, Bric 1993, Wang 1994).

- Terrestrial surveys make the acquisition of 3D information possible. State-of-the-art surveying instruments are supported by portable computers for recording field data that can be directly transferred to the GIS workstation.

- Global Positioning Systems (GPS) provide another means of 3D data acquisition. GPS is based on NAV star satellites that help determine positions in the WGS coordinate system. Because the 3D coordinates can be directly obtained at the primary data acquisition stage, GPS may be considered to have potential for real-time mapping if, during data acquisition, the GPS receiver can directly transmit the coordinate measurements to a 3D GIS for further processing.

- Borehole and seismic surveys provide the means for 3D data acquisition specific to geological applications. This type of data acquisition is rather expensive and cannot provide comprehensive data to represent the real situation. Additional data from the interpretation of morphological characteristics of the terrain are normally required to improve the quality of the representation.

- Photogrammetry, in many cases, provides the means to obtain 3D information. Measurements can pass directly into a stereo model of photographs or images taken from the aircraft or satellite, and simultaneous interpretation of some spatial relationships may also be possible. The indirect approach to producing 3D information is usually based on such digital techniques as object recognition and image matching. A digital photogrammetric workstation that exploits computer graphics also makes it possible to overlay vector data with raster images, providing 3D visualization in stereo mode that helps verify the data contents. This is very convenient for change detection. A further requirement for this technology is the storage of topology in the case of direct measurement and a 3D editing capability with interactive graphic display. The term 'Computer Vision' is used specifically for the recognition of objects in the image scene in a more controlled environment, for example, in robotic design.

- Digital Terrain Model is used for the representation of relief. Two forms of DTM are typically used: regular-grid (also known as raster DTM); triangular irregular network (TIN). DTM supports various computation-based analyses in GIS, such as slope, aspect, visibility, volumes, contour lines, surface area, the derivation of a morphological feature. Because the data set of DTM represents 3D coordinates, it is ready to be part of a 3D GIS. DTM can be graphically presented in such different ways as perspective view and shaded relief maps. DTM is also used for the production of orthophotos, containing texture information. The orthophoto, together with other information resulting from query or analysis, can be draped on top of the DTM displayed in perspective view, resulting in a realistic visualization of geoinformation.

- Remote sensing is a technology for primary data acquisition producing images, mostly in digital form, from sensors on satellites such as LANDSAT, SPOT, ERS1, MOSS, or sensors mounted on aircraft which provide images with higher resolution. Information can be extracted by interpreting the raw image using manual digitizing, or digital image processing techniques, if images are provided in a digital form. 3D information can be obtained by applying photogrammetric techniques to a sequence of overlapping images, using either manual measurement or a computational approach, as previously described. Each image also supplies texture information for realistic visualization, for example, by draping the digital image on the DTM presented in perspective view.

- Digital Image Processing (DIP) is used for information extraction and spatial analysis. Raw images can be processed by certain operators and the results classified into themes. DIP is increasingly applied to enhance the functionality of digital photogrammetry for visualization enhancement to improve image interpretation.

- Computer Graphics are used mainly for graphic presentation. During the last decade, considerable advances have been made in the hardware development known as the 'graphic accelerator'. Its use for drawing operations, 2D and 3D transformations, rendering engine, graphic user interfaces (GUI), and so forth to support CG, has become affordable. A number of programming standards, such as GKS, PHIGS, OPENGL, have been made available which makes the implementation of realistic visualization easier.

- Computer Aided Design is typically applied in engineering. CAD provides many 3D editing capabilities that can be utilized by 3D GIS, for example, during 3D digitizing or editing in a later stage. ArcCAD (ESRI), which is the combination of Arc/Info and AutoCAD (Autodesk, inc.) is an example of exploiting 3D capability in CAD for GIS. For 3D GIS, CAD should be strengthened with better 3D topology. Adjustment of user-interface towards requirements in geoinformation, for example, terminology, additional functions and their operational sequence, would make it more suitable for 3D GIS. Examples of CAD that adapt to 3D GIS are MicroStation and MGE (Intergraph).

- Virtual Reality (VR) is a highly interactive and realistic 3D graphic display of information that allows the system to interact with the human user in a more natural way, thereby offering fast, intuitive understanding of information. This technology exploits highly sophisticated CG in both software and hardware, as a result of the fact that a large number of data elements have to be displayed in a very short period of time (30 frames per second) to achieve a high degree of continuation. VR techniques allow users to modify the virtual environment, navigate from place to place to match their analyses and receive more natural feedbacks and responses from the virtual objects residing in the database. Circumstances may require multimedia technology beyond visualization to provide tactile or sound feedback. There is a trend to adopt the VR as the standard user-interface (Computer Graphic World, May 1995).

- Database Management Systems have been continuously developed over a long period. Progress has been made in adding the capability of handling spatial data by both relational and object-oriented DBMSs in, for example, ORACLE version 7, PostGres, ILLUSTRA, O2. The ease of handling spatial objects in GIS seems very promising. The use of topological data structures for a spatial database that allows fast access to data elements based on spatial relations is becoming more widespread and can be found in many commercial GISs, such as Arc/Info, MGE, ILWIS, SmallWorld, GDS. This topological structure increases the efficiency of GISs. However, the capability of handling 3D spatial objects and their interrelationships awaits further developments.

- Expert systems and artificial intelligence add the capability of storing human knowledge about how to deal with complex problems based on known facts and rules in, for example, data processing and spatial analysis. Such a capability would be useful where no human expert is available.

- Computational geometry is used to compute the spatial relationships between the objects using metric computation, for example, the intersection between two lines, point-in-polygon tests, the angle between two vectors, or the circumference of a triangle. Algorithms for computational geometry are now mostly available for 2D. Further development for full 3D computational geometry would ease the topological structuring of 3D data.

In summary, we can relate each technology with the functional components of 3D GIS as shown in the table below.

	Acquiring	Structuring	Storing & Managing	Processing	Presenting
Map digitizing	•	•			
Terrestrial Survey	•				
GPS	•				
Borehole & seismic	•				
Photogrammetry	•	•		•	•
RS	•				•
DTM		•		•	
DIP	•			•	

CG				•	•
CAD	•	•			
VR				•	•
DBMS		•	•	•	
Expert System		•	•	•	
Computational geometry		•		•	

The construction of 3D GIS brings together the above technologies into its functional components. The different approaches found at the time of writing are outlined in the next section.

3.4 Evolution Stages of System Architecture

The composition of a system for integrated geoinformation always depends on the state of the art, policy and economic constraints. Technology develops, and we can consider various system architectures as stages of evolution in the handling of geoinformation. We use the following fourteen criteria to analyse the different evolutionary stages.

1) Compactness of the system
2) Common operating system (OS) or hardware platform
3) Functional access
4) Data access
5) Relationships between components of the spatial model
6) Commonness of user-interface
7) Investment cost
8) Maintenance
9) Data redundancy
10) Handling of uncertainty
11) Productivity of geoinformation
12) Potential towards automation
13) Supporting personnel
14) Size of user organization

3.4.1 Evolution Stage 1: Independent Subsystems

A GIS can be composed from a set of subsystems, as shown in Figure 3.1. Although this would seem to be an easy (but expensive) approach to constructing a GIS, it can only be regarded as a low level of integration with a low degree of unification. Each subsystem offers only a subset of all the functions of a GIS to carry out some specific tasks along the geoinformation production line. With respect to the above defined criteria, this kind of system has the following characteristics.

1) The system is composed of several subsystems, either in the form of hardware or software. Therefore, it is not compact.
2) Different subsystems may need different hardware and OSs, for example, VMS, Unix, MSDOS, MacOS.
3) The system cannot provide a central control panel. The functions of a subsystem can only be reached from the respective local control panel.
4) The data cannot be accessed from a single entry point. Data transfer from one subsystem to another may need to be done manually, for example, by using floppy disks, if the subsystems used for the consecutive operations are not connected on-line. Data conversion is likely to be required because typically each subsystem will use its own data structure.
5) Components of information are usually stored separately in the local database of each subsystem. For example, data representing man made objects may be stored in the CAD subsystem, data of terrain relief in the DTM subsystem, and data of other terrain objects in the 2D geoinformation subsystem. This segregation means that metric computation must be used to integrate data from different subsystems before topological relationships can be created.
6) The system does not provide a common user-interface. The user-interface is locally provided and depends on each subsystem. Elaborate user training is needed and operation is liable to mistakes.
7) Investment costs are high because each subsystem has to be purchased separately. Several subsystems are required to achieve the required functionality.
8) Maintenance is difficult and expensive. Different vendors may be responsible for different subsystems.
9) Data redundancy will be high, because of the separate and independent storage.
10) Dealing with uncertainty is necessary in operations that involve data sets from different subsystems.
11) Users have to cope with many problems, so the productivity is not likely to be very high.
12) The production line is difficult to automate because of the limitations mentioned.
13) This approach requires various supporting personnel, for example, an OS specialist, an application specialist, application programmer to ensure operation.
14) The size of the user organization is quite large in terms of number of personnel and space required for placing the subsystems.

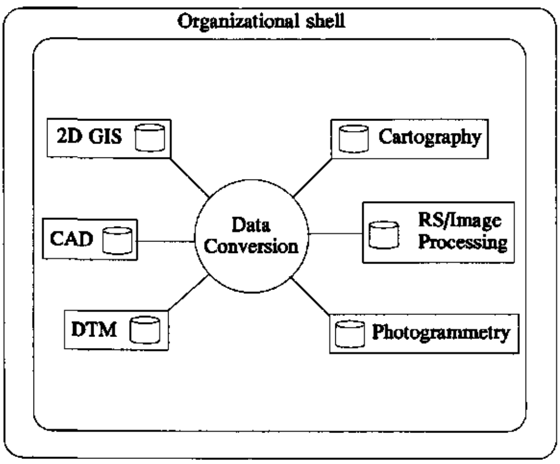


Figure 3.1 *3D GIS by Independent sub-systems.*

Figure 3.1 is a graphical illustration of this approach. Data conversion plays a central role in integrating the components of the spatial model stored separately and independently as databases in various subsystems.

3.4.2 Evolution Stage 2: Functional Integration

A system based on this architecture combines all the necessary functions into one software package. Figure 3.2 illustrates this approach. Below are listed the characteristics of the system. It:

1) is compact, because all subsystems are shrunk into functions or software modules implemented within the system
2) is based on one OS and hardware platform
3) provides a central control panel
4) accesses data from a single entry point; data transfer between software modules can take place as a background process
5) has separate data structure to store data in each module; for example, coverage data and TIN data in Arc/Info are stored in separate data sets with different data structures. Topological relationships between data elements across different data sets do not exist.
6) provides a common user-interface
7) is less expensive than the independent subsystems and the client/server (see section 3.4.3), because it is based on only one software package
8) is easy to maintain, having fewer pieces of hardware and software and only one vendor to deal with
9) still has data redundancy from different data sets
10) has problems in handling uncertainty similar to evolution stage 1
11) has better productivity than the client/server approach, because all processes are locally performed under one system shell, requiring less time for data transfer and message translation
12) has many operations which can be automated, making it more feasible to automate the whole production line. When familiar with the system, the user can optimize and streamline the operation, for example, by using script or macro language, usually provided by this kind of system, to combine basic functions, so reducing many inter-processes requiring manual operation
13) requires fewer support personnel than evolution stage 1

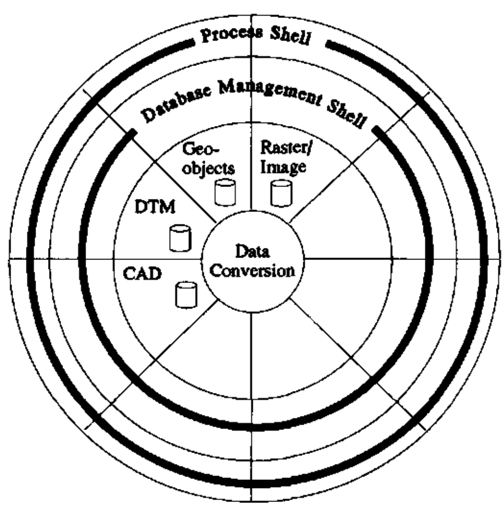


Figure 3.2 Functional integration

14) requires a smaller sized user organization in terms of number of personnel and space for accommodating the system. The vendors' organization, however, becomes larger because the design and implementation of the system require personnel from many disciplines.

Although this approach represents most of the present attempts of development of GISs, it turns out that no single system can offer all required functions yet. An example is that most of the GISs still use 2D spatial models as bases and cannot offer a 3D modelling capability. Users still have to adopt the architecture of evolution stage 1 to achieve the required functionality.

3.4.3 Evolution Stage 3: Client/Server Architecture

This approach is based on the communication between a 'client' and a 'server'. The client is a module provided to interface with the end-user. The server provides operations to process requests from, and gives feedback to, the client. The server and the user can only communicate via the client module. Typically, the client/server approach:

1) does not attempt to improve the compactness of the system; it is still composed of several independent subsystems, as in stage 1.
2) is able to carry out tasks on different hardware platforms and OSs by using a standard communication protocol. The role of the OSs on different platforms are suppressed, so the user only needs to deal with the OS and the hardware platform of the client module.
3) provides the client with a central control panel. Necessary functions can be reached from the client module. Each client function turns a user action into an appropriate request message which is sent to the server. On receiving the message, the server evaluates the request and triggers a process, if the request is valid. The client can attach to the request message data to be processed by the server. The server processes the data and sends back the result to the client. The user may not have the freedom to explore the functionality of each server unless better access to the functions of the server is provided through a mechanism called 'object link and embedding' (OLE) (see Microsoft 1993, Brockschmidt 1993). This mechanism maintains the link between each data set and its specific server application. More than one application server can be attached to a data set, and a choice of servers may be provided. On selecting the data set and server, OLE activates and transfers all control and the user-interface to the server, which allows the user to access all functions provided by the server. The user returns to the client by quitting the server. The OLE approach works well on a single OS platform. The client/server approach does not require the user to move around and enter several OS shells to reach functions that are available on different subsystems.
4) accesses data through the client, whose database is a container embedding and encompassing different data sets which may have different data structures native to specific servers. Each data set embedded in the client database may only be recognizable by a specific server (see Figure 3.3). The data is transferred from one server to another server on-line. The client application may have to provide a data conversion function locally if the destination server cannot recognize a nonnative data structure.

5) stores components of information separately in different embedded data sets. The client database can be regarded as a collection of different data sets. There are no topological relationships across different data sets.
6) has a common user interface with an intuitive graphical user interface (GUI) provided by the client; however, this is limited by the extent to which the client knows the functions of the different servers.
7) has investment costs as high as the independent subsystems approach, with the additional cost of the client application and the network connection to all server applications.
8) must have the client application upgraded in accordance with the upgrading of one of the servers or communication protocol.
9) fails to minimize data redundancy caused by the independency of the embedded data sets.
10) has to deal with uncertainty every time relationships between data elements across different data sets have to be created.
11) requires users to deal with fewer problems in functional access and data access and transfer, thereby improving productivity.
12) facilitates automation of the production line through eliminating some manual processes.
13) requires fewer supporting personnel, because users need not deal with many low level operations.
14) has smaller sized organization than the independent subsystems approach with respect to personnel, but not with respect to the space accommodating the subsystems.

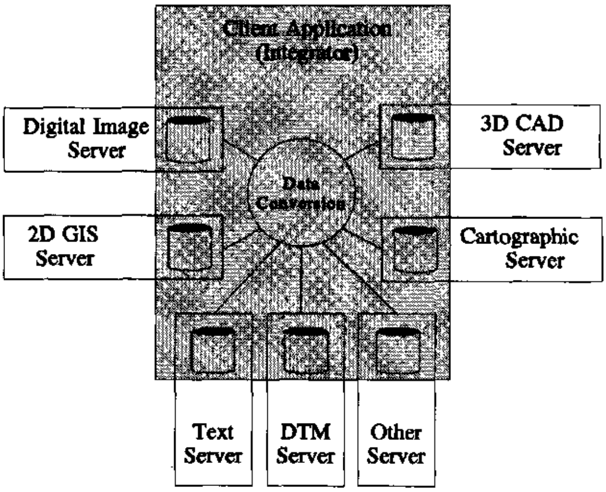


Figure 3.3 Client/server architecture

This approach was developed after evolution stage 2 in response to demands from the user community. It can be regarded as an intermediate solution because complete functionality is not yet available on a single system. The client/server approach can be a good solution for providing access to all the required functions on the different independent subsystems. The client may be provided by a third party experienced in assisting in interfacing users with systems from various vendors (for example, training, design of process flow), or systems provided through cooperation between the vendors (see Intergraph 1995).

3.4.4 Evolution Stage 4: Structural Integration

Creating a spatial model that better represents reality requires more closely related geoinformation components. Relationships between data elements need to be well defined by means of topology, which is likely to be difficult without an appropriate unified data structure (UNS). The proposed architecture is based on UNS as described in Pilouk and Tempfli (1994), Pilouk et al (1994); see also chapters 4 and 5.

3.4.4.1 General Consideration

This approach aims at a better representation of real world objects and relationships between them while maintaining the good aspects of evolution stages 1 to 3. The major considerations for the structural integration are:

- All components of the 3D spatial model must be stored in one database, so that topology can be applied to represent the spatial relationships of the real world objects. The use of topology avoids metric computation and speeds up many operations.
- Both direct and indirect representation must be possible within the system. Direct and indirect representations need different processes to present the information to the user. Allowing both representations in one database requires the system to use appropriate data structure.
- Data redundancy must be minimized. Redundancy is usually introduced by a lack of awareness of existing data. Storing redundant data provides no additional information; it consumes storage space and may lead to conflicts whose resolution requires additional operations.
- The frequency of handling of uncertainty must be minimized and be taken away from the end-user as much as possible by converting uncertainty into a data quality attribute beneficial to the end-user. The database creator has better access to the original sources and is in a better position to resolve the uncertainty.

3.4.4.2 A Proposed System Architecture

The proposed architecture for a GIS is illustrated by **Figure 3.4**. There are various layers of the system encompassing the integrated database based on UNS. The integrated database contains a spatial model accommodating both direct and indirect representations of real world objects. The database management shell embraces this database, on which various indices or database views exist. Spatial access to some specific data elements can be speeded up by a database index. This can be regarded as a specific view of the earth that is closer to the application domain or operation requirement. The existence and status of each index and view depend on the integrated database. All indices and views are updated according to the changes in the integrated database. Any changes must be directed first to the integrated database and subsequently to the indices and views. The database management shell provides functions and rules to access and update the integrated database or views. The process shell is the outer level next to the database management shell. It contains various functions to process the integrated database or views by using database management functions provided by the database management shell. The next outer shell is the generalization shell which simplifies the input from the real world to the GIS and provides the means for presenting or delivering the information stored in the GIS to the users or to the client databases (see also Richardson 1993, Peng and Molenaar 1995, Peng et al 1996).

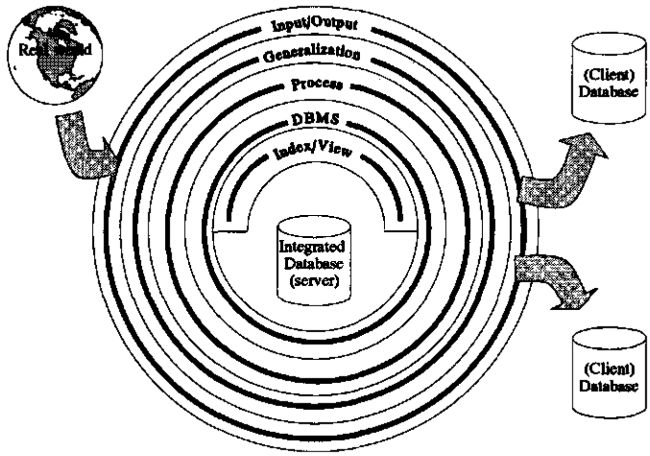


Figure 3.4 Overview of the structural integration architecture.

Figure 3.5 shows a more elaborate architecture of a 3D GIS. The focus is on the views of the integrated database. The suggestion here is to use already existing data models, for example, SWM, MWM, TIN-DTM, 3D FDS, as the underlined structures of the views. The reason is that those data models in fact represent different views of the reality. The integrated database can be regarded as the integration of views, so it may contain over extensive data for an individual application, resulting in bad access and response time. Each view may contain pointers to specific data elements in the integrated database to speed up the search operation, allowing the user fast access to frequently used data elements.

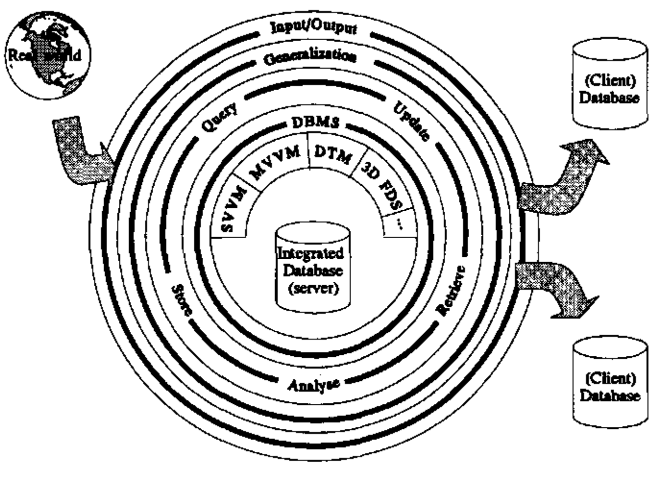


Figure 3.5 Different data structures can be used at the view level in the structural integration.

Characteristically, a system adopting the structural integration approach:

1) is as compact as in the functional integration approach
2) uses a single OS and hardware platform
3) provides a central control panel
4) provides data access from a single entry point. Each data element, that is to say, a component of the spatial model, can be accessed directly or by navigating via other data elements.
5) establishes relationships between data elements to represent spatial relationships in reality
6) can provide a common user-interface
7) has investment costs as low as for the functional integration. The investment in data storage capacity is expected to be lower than for the other architectures, through reducing redundancy.
8) is easy to maintain with respect to hardware, software and database
9) eliminates data redundancy
10) handles uncertainty only at the database creation and updating stages. Other operations only exploit the data quality information
11) anticipates high productivity
12) can support automated production because manual operations such as data transfer, conversion and resolution of uncertainty are eliminated
13) requires fewer support staff
14) requires organization size comparable to or smaller than for the functional integration architecture.

3.5 Comparison of Different System Architectures

The evolution stages of the system architectures can be compared as in the table below. The qualitative assessments presented result from the analysis in the previous section.

Criteria	Independent subsystems	Functional Integration	Client/Server	Structural Integration (Expected)
System compactness	No	Yes	No	Yes
OS & hardware platform	Multiple	Single	Multiple	Single
Functional access	Poor	Good	Good	Good
Data access	Poor	Good	Fair	Good
Data relationships	Poor	Poor	Poor	Good
User-interface	Poor	Good	Fair	Good
Investment cost	High	Low	High	Low
Maintenance	Complicate	Simple	Complicate	Simple
Data redundancy	High	High	High	Low

Handling of uncertainty	Frequent	Frequent	Frequent	Good
Productivity	Poor	Good	Fair	Good
Automation	Poor	Possible	Poor	Potential
Supporting personnel	Many	Fewer	Fewer	Fewer
User organisation	Large	Small	Small	Small

When comparing the four evolution stages, we can say the performance of structural integration is better or at least equal to any other form of system architecture with respect to all fourteen criteria.

3.6 Attempts at Structural Integration

Efforts to establish 3D GIS start from the 'Integration of DTM and GIS' reported in Males (1978), Sandgaard (1988), Ebner et al (1990), Mark et al (1989), Fritsch (1990), Pfannenstein and Reinhardt (1993), Weibel (1993), Pilouk and Tempfli (1993). The aim is to tackle the aspect of dimensionality of geoinformation, focusing especially on the integration of 2D-GIS and DTM. Early researches divide into two groups. The first group tries to build a bridge between raster and vector domains, in the 'DTM interface' approach. Ebner et al (1990) Mark et al (1989) report examples of such an approach which is still regarded as some kind of functional integration with a low degree of unification. The second group attempts what Fritsch and Pfannenstein (1992a) call 'total integration', where the separation between DTM and GIS should disappear. Examples of work in the second group are Males (1978), Sandgaard (1988), Fritsch (1990), Pilouk and Tempfli (1993).

Males (1978) presented a system called ADAPT (Area Design and Planning Tool), designed to manage geographical data in accordance with DTM. The system exploits the TIN structure by associating attributes to the triangle elements, for example, drainage and ridge lines as triangle sides, area as a connected set of triangle edges, and point data as vertices. The explanation about data structure is clear, but the data model, which should describe the topological relationships between data elements and between simplices and complexes, has not been well defined. Although the ADAPT system is not commercially well known and is limited to 2.5D, it is a good example of GIS integrating relief and other information about 2D features within one database

Sandgaard (1988) presented an approach to integrating the DTM package called TIP (Terrain Information Programs) with the DanGraf, a GIS in widespread use in Denmark. TIP is based on TIN structure and integration occurs through classifying DTM elements into points (peaks and pits), lines (breaklines and structural lines) and areas (dead, measured, and mesh). These elements are stored as a DTM feature class in the DanGraf database. The features may also be classified as DTM types, for example, roads and river banks may be classified as breaklines. The emphasis was placed on the classification scheme and operation rather than on the data model and data structure.

Fritsch (1990) investigated different approaches at the level of the data model, suggesting an object-oriented approach. The discussion of the conceptual aspects of data models and data structure is reviewed by Pfannenstein and Reinhardt (1993). Weibel (1993) discusses requirements for integration.

Pilouk and Tempfli (1993, 1994) have reported the formalization of the data model and data structure for the integration of DTM and GIS, proposing *an integrated data model and a unified* relational data structure. All the basic topological relationships between data elements and between simplices and complexes are clearly defined and used as a basis for the data model for 3D GIS developed here (see chapter 4 for details).

This section reviews some existing data models having some integrative aspects. The 2.5D and 3D data models are evaluated against the following criteria: the ability to

- accommodate objects of various dimensions, especially ranging from 0D to 3D
- maintain topological descriptions between data elements and between simplices and complexes
- support the modelling of objects with determinate spatial extent
- support the modelling of objects with indeterminate spatial extent
- permit the modelling of objects with spatial coincidence (multi-theme)
- expand to accommodate objects of higher dimensions
- permit derivation of unified data structure for the implementation.

3.6.1 2.5D Approach

The 2.5D approach can be regarded as a first integration step seeking to combine the separated data sets of 2D geoinformation and DTM. The separation of these two components of spatial objects has been inherited from 2D map production, where DTM was used to automate the production of contour lines for relief representation in a 2D map. DTM has merely been a matrix storage of height information which helps speed up the contouring process. DTM has become more important because it facilitates the derivation of a wide range of information products serving various applications, for example, contours, slope and aspect maps, relief shading, perspective view, cross-section and profile, visibility, volume and surface area. DTM can be used as a reference (mapping) surface for other terrain features, for example, for draping terrain features on a (DTM) surface and display in a perspective view, which gives a more naturalistic visualization requiring less map-reading skill.

The most traditional attempt in 2.5D integration is the 'height attributing' approach (Fritsch and Pfannenstein 1992a) that considers height as thematic information. In the case of point objects, this approach seems to be adequate. For a line or surface object, the representation of height can only be done if there is an assumption that the object is on a horizontal flat plane, or every part of the object has the same elevation. This approach can only solve problems for a limited number of applications (Ebner and Eder 1992, Fritsch and Pfannenstein 1992b). Figure 3.6 demonstrates the approach.

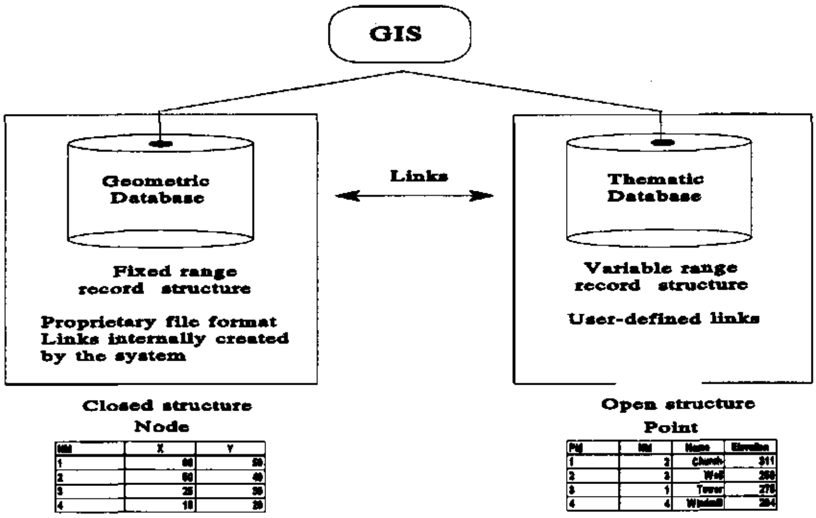


Figure 3.6 Height attributing approach maintain elevation in a thematic table.

This approach is dictated by the fact that most commercial GISs do not allow the user to have direct access to the geometric data elements kept in a fixed structure. Although the system may provide ways to retrieve this data into a database table and allow the user to utilize it, or add additional attribute columns, possibly one containing height information (z coordinate), the table is only regarded as external and is not part of the geometric components. Most commercial GIS vendors maintain the geometric components in their specific proprietary formats which are legalized and prohibited for reverse engineering, for example, studying data structures with the purpose of extending or modifying them. Therefore, the only possible way to include height information is to add it into the non spatial part of the GIS database. Furthermore, the storage of this table into the internal geometric database is not provided, or the storage mechanism declines to store extra information added by the user. Nevertheless, the existence of this approach indicates the user's need to have a system capable of handling more than 2D coordinates.

Besides the height attributing approach, Fritsch (1990) has suggested an object oriented alternative for handling a hybrid database that embraces three separate data sets: two for geometry (terrain data and situation data); one for non graphical data (attributes). Figure 3.7 illustrates the approach, which still needs further development using object-oriented programming.

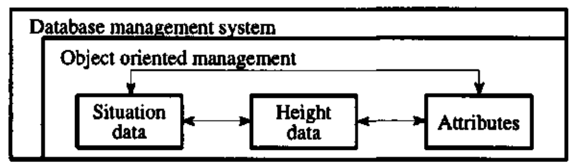


Figure 3.7 Hybrid database (after Fritsch 1990).

Kufoniyi and Bouloucos (1994) suggested a model using the multiple theme approach by taking DTM as a thematic layer and the database maintaining the links to DTM via triangle edges and vertices that are respectively isomorphic to the primitive arcs and nodes of the multi-theme 2D FDS. The data model is presented in Figure 3.8. The model is capable of handling information

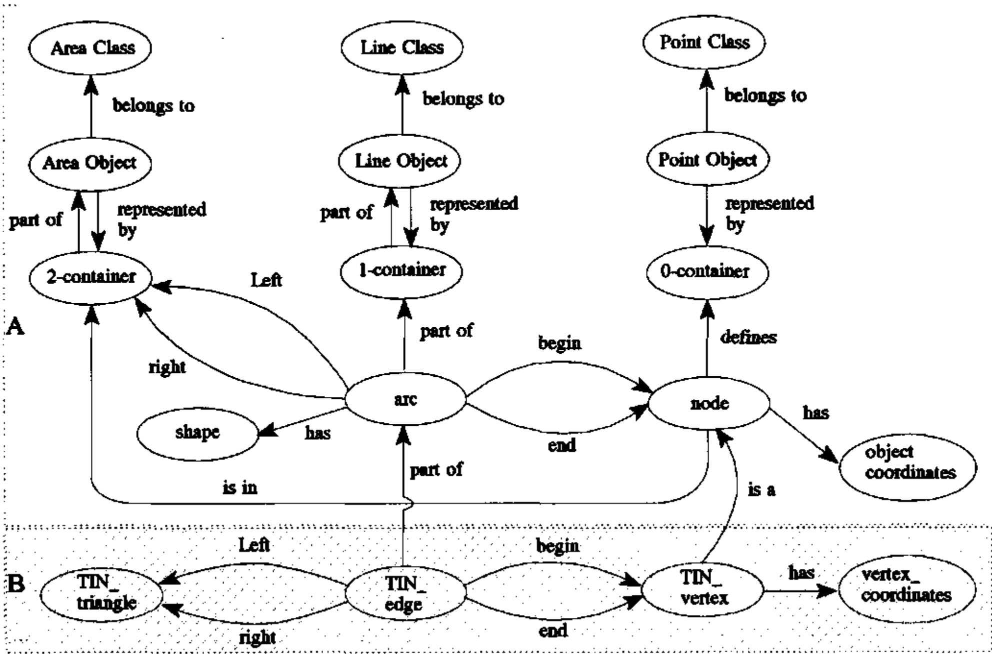


Figure 3.8 Integrated data model for multi-theme vector maps and DTM based on object-based (adapted from Kufoniyi and Bouloucos 1994).

about relief at the nodes and arcs of the features quite efficiently, because the data model provides direct links to triangle edges and vertices. However, the explicit representation of the topology between 2D geometric primitives and features is still lacking, so the problem indicated in Fritsch's data model relating 2D objects with terrain relief remains. The use of topology for navigating in the database to retrieve relief information (DTM points) inside the area features is difficult. Although, the computational geometry, the 'point-in-polygon' testing, can help overcome this problem, this computation is a time consuming process, especially when it has to be tested on a large number of points, as is the case for actual DTM data sets.

3.6.2 3D Approach

For applications that need to model solid objects, the 3D FDS, proposed by Molenaar (1990), offers a data model with 3D topology. This data model permits the representation of spatial objects in the different dimensions 0D, 1D, 2D and 3D, that is the point, line, surface and body features respectively. It is therefore possible to model terrain as a surface feature. Figure 3.9 is the graphical illustration of this data model.

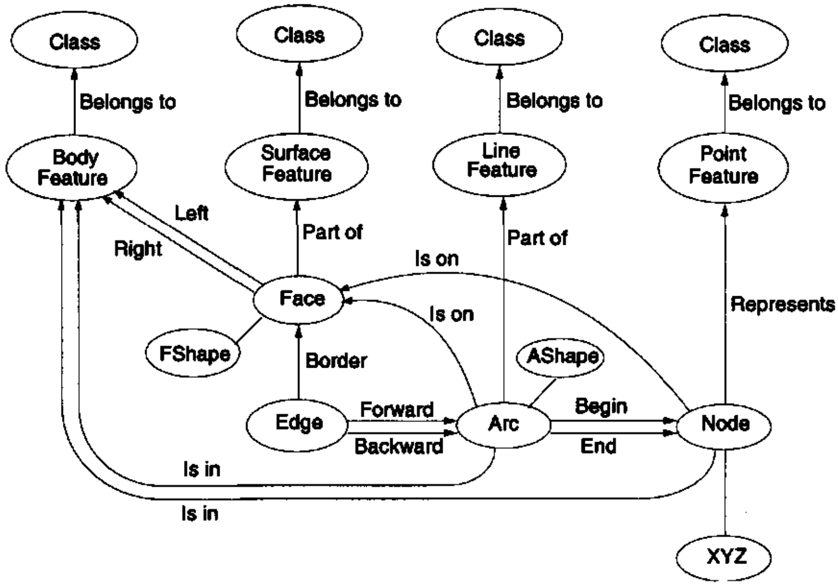


Figure 3.9 Formal data structure for 3D vector map (after Molenaar 1990).

The conceptual design of the 3D FDS is the same as for SVVM or 2D FDS. The design is based on the decomposition of a feature into identifier, geometry, and theme. Components for representing a spatial object are grouped into three levels, namely the geometric, feature and class levels, in the same way as in 2D FDS. *Faces* are 2D geometric primitives in addition to *nodes* and *arcs*, respectively of 0 and 1 dimension. *Edges* are additional geometric primitives providing the link between arcs and faces and so permitting the unique reference to *left* and *right* bodies that are 3D features. Figure 3.10 illustrates this concept.

Unless *faces* are constrained to be planar surfaces, the primitive *fshape* provides a mathematical or numerical description of a face. On the feature level, each feature has a unique identifier. The data model distinguishes four types of features: *point, line, surface* and *body*. A point feature consists of a node, a line feature of one or more arcs, a surface feature of one or more faces, and a body feature is bounded by faces. Within the concept of a single-valued vector map (see Molenaar 1989), the geometric component of each feature is then related to one specific thematic class. Shibasaki and Shaobo (1992) applied the 3D FDS to develop the digital urban space map. Bric et al (1994) and Wang (1994) showed the applicability of this model for problems where the involved

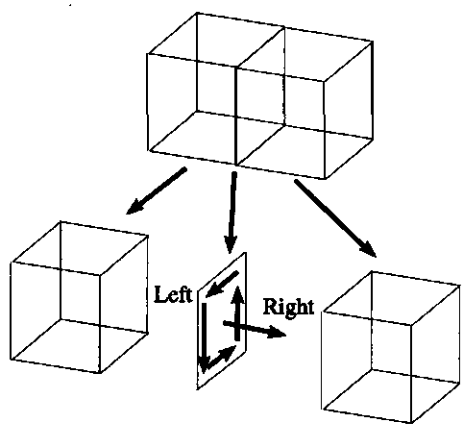


Figure 3.10 Edges and orientation of face allow the links to left and right bodies.

spatial objects are regularly shaped, and when geometry, topology and thematic attributes are known at the time of data acquisition.

The 3D FDS is suitable for direct representation and query-based spatial analysis; it does not, however, provide the 3D primitive data type and is therefore less suited to solving problems for applications that involve complex computation, for example, using the finite element method, where a 3D primitive is crucial because it is needed as a computation unit. The 3D FDS is based on single-valued concepts; however, it can readily be extended to accommodate spatial coincidence, or multi-valued aspects.

The 3D FDS has been used as a basis for implementing TREVIS, an experimental 3D GIS with limited capability. TREVIS uses a relational DBMS (dBASE IV) to store and manage all the elements of a spatial model in the form of tables according to the relational data structure, as shown in Figure 3.11. The interactive graphic module accesses these tables directly to display the spatial information and the query result in the form of wireframe graphics in a controllable

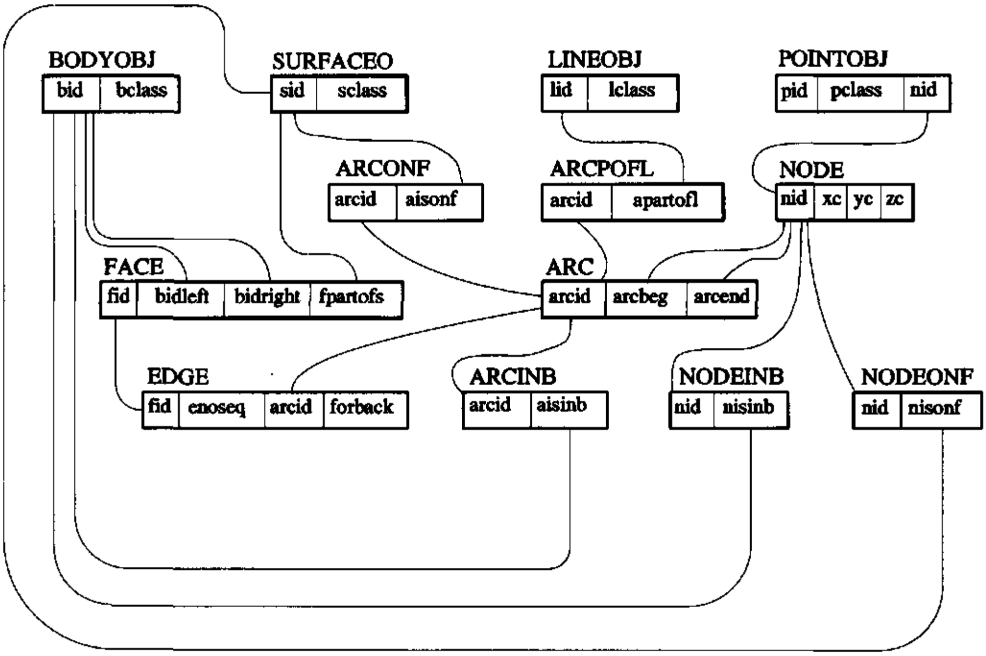


Figure 3.11Relational database structure based on 3D FDS (adapted from Rikkers et al 1993).

perspective view. Point, line, surface and body features can be used to represent spatial objects. Components of determinate spatial objects and the relationships between them can be stored in one database. If sufficient functions for the handling and analysis of spatial data were provided, TREVIS would be an attempt in evolution stage 4 suitable for serving the modelling of determinate spatial objects.

3.7 Discussion

Many technological developments have to be incorporated into the functional aspects of 3D GIS. Many different approaches to the construction of a 3D GIS can be used, ranging from loosely to well-constructed systems. These approaches are differentiated into four evolution stages of system architecture; independent subsystems, functional integration, client/server architecture, and structural integration. The proposed structural integration involves considerable effort in designing and constructing the system. This effort, however, is transferred to the vendor rather than left to the user, as in evolution stage 1. The expected superiority lies not only in the handling of uncertainty, but also in the benefits accruing from explicit relationships and minimized redundancy, with all the consequences for the user.

Current systems are in evolution stage 1, with increasing attempts concerning stage 2. There is a trend towards evolution stage 3 as a result of commercial driving forces that prefer to keep proprietary developments from the public while offering improved user-friendliness, partially fulfilling user demands. This trend is evidenced by the adoption of specifications for OLE extensions for computer-aided-design (CAD), computer-aided-manufacturing (CAM), computer-aided-engineering (CAE) and GIS by a group of vendors related to spatial information industries, namely the Design & Modelling Application Council (DMAC), which includes ANSYS Inc., Autodesk, Bentley System Inc., Cadence Design Systems Inc., Crisis in Perspective Inc., Intergraph, Microsoft, Ray Dream, SDRC, Shapeware Corp. and Spatial Technology, since the beginning of 1995 (Intergraph 1995). The evolution stage 4 promises to be an ideal system, but it can only be achieved if there is a consensus on the formal design of the spatial data model. Such a consensus could also result in the adoption of the client/server architecture on top of the structural integration.

The state of progress towards structural integration through current attempts shows that further investigation and extension of the existing data models are looked for. The essence of the matter is to be able to accommodate both direct and indirect 3D representations of real world objects within an integrated 3D spatial model, similar to the normal situation in reality, which would allow the representation and analysis of spatial relationships between the two types of objects. A 3D spatial data model with this capability should be further developed to serve as a basis for a 3D GIS adopting structural integration architecture.

4
CONCEPTUAL DESIGN

This chapter elaborates on the development of the integrated vector data model as a basis for the implementation of a core database in a 3D geoinformation system. The tessellation approach is used with the aim of extending the query space as offered in the object-based approach, described in chapter 2. The tessellation approach accommodates a wider range of complex spatial analyses that involve both computation (for example, interpolation, slope, aspect, visibility, shading, surface area, volume) and simultaneous topological navigating in the database (for example, selection, indexing, sorting). The combination of irregular tessellation and the application of topology, as found in FDS and described in chapter 2, can offer such an opportunity. They are therefore adopted to strengthen integrated 3D modelling. We first elaborate the 2.5D model which integrates terrain relief and terrain features. Since this data model is mathematically sound, it can be readily extended to cover aspects required for 3D modelling. To ensure forward compatibility, so that the 3D spatial model can be incorporated into a higher dimensional spatial model in future, a generalization for nD is also presented. The properties of the integrated data model are discussed in the framework of simplicial complexes and graph theory. This discussion leads to the definition of a simplicial network as well as its Euler characteristics.

4.1 TIN-based (2.5D) Data Model

The first stage of development of the integrated data model aims at a structural integration of the representation of terrain relief and terrain features, which is regarded as 2.5D. The terrain relief represents the geometry of the earth's surface, where terrain features are 2D representations of spatial objects. Terrain relief and terrain features information in the forms of DTM and GIS respectively are familiar tools for spatially related problem solving, decision making and hazard mitigation, such as erosion and landslide, agricultural land reformation. Since the integrated database should fulfil the requirements of typical DTM and GIS (see chapter 3), it must ensure all the functionalities of the two systems.

To design the integrated terrain relief and features data model, TIN and 2D FDS have been selected, for the reasons indicated below.

1) Efficient interpolation
Two main underlying principles adopted in the design of this data model are proximal ordering and decomposition into primitives. Digital terrain relief modelling requires interpolation, which in turn requires the proximal relationships among the given points to be known. As DTMs are in most cases based on non-gridded data, adjacency can best be expressed by triangulation. Triangulation of surveyed points in order to interpolate contour lines for topographic maps was in fact applied long before the advent of computers. Thiessen (1911) polygonal tessellation and its geometric dual, the Delaunay triangulation (see chapter 2, Pilouk and Tempfli 1992), are of special interest. This is through their establishment of a natural neighbour structure facilitating the interpolation process where different methods of interpolation (see Tempfli 1982) can be applied.

2) Fidelity and embedding of constraints

One way to increase the fidelity of surface representation is to incorporate skeleton data, such as ridge and drainage lines, in the DTM. Different approaches have been suggested, for example, by composite sampling – the combination of selective and progressive sampling (Makarovic 1977) – and constrained triangulation (Pilouk 1992). The latter is capable of maintaining skeleton data without losing their original geometry. Constrained triangulation permits the embedding of such geometric components of skeleton features as the components of triangles; for example, a line feature can be decomposed into a series of straight line segments and be embedded as edges of triangles. The same approach as for the skeleton data can be applied to embedding geometric components of other terrain features in the TIN structure without losing their original shape as obtained from observation or measurement. These features represent the human knowledge about the aspects of reality which should be recorded correctly into the database and so their information must be maintained. As a by-product, the fidelity of surface representation is also increased when more terrain features are incorporated into the TIN. This capability of embedding the terrain features is one of the most important aspects in the design of the integrated data model.

3) Locality

The locality is an important aspect of a very large data set where a large amount of time is obviously required to process the whole data set, as in data retrieval, calculation, and updating. As pointed out in the literature, TIN permits the local editing and updating of elevation data without the elaborate re-interpolation necessary in a grid-based structure (see Fritsch and Pfannenstein 1992a). The local editing of TIN structure that only involves necessary data has been demonstrated (Jackson et al 1989, Pilouk et al 1994, Midtbø 1993).

4) Convex shape

A triangle is a convex polygon. Its simplicity reduces uncertainties and consequently requires less testing, thereby offering significant advantages in many graphic and geometric operations where fast computing speed is crucial. Some examples of these operations are the inclusion of a point in a polygon, as in colour filling, and rasterization, intersection with a line and a polygon, as in hidden-line/surface removal, and finding the direction of the surface normal vector to determine the reflection of light and surface visibility.

5) Finiteness and adaptability

The number of vector elements that are the components of a triangle is fixed. It is, therefore, easy to control the consistency of its network. Nevertheless, a triangle's finiteness does not prevent the adaptability of the network to the terrain roughness, because the network of triangles can be densified without limit. The irregular shape of a triangle permits its adaptation to the irregular distribution of observations or measurements.

6) Compatibility with FDS

TIN has typically been used for the representation of surface geometry. Since it is vector-based, creating the links between geometric components and features that have links to thematic components in a similar way to that defined in FDS is highly feasible (Molenaar 1989).

For the above reasons, the design of the integrated terrain relief and features data model is as shown in Figure 4.1.

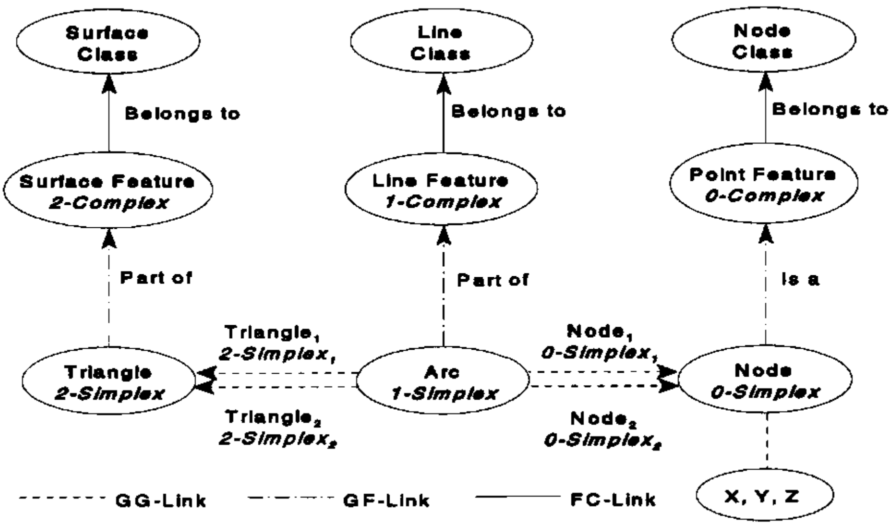


Figure 4.1 The proposed integrated 'DTM-GIS' data model.

4.2 Properties of The TIN-based Data Model

The TIN-based integrated data model can be considered as the extension of the 2D FDS of SWM on the geometric level. The geometric primitive *triangle* is added, and a data model obtained which serves the purpose of the unified handling of all surface related data. In comparison with 2D FDS (Figure 2.17), some links presented in 2D FDS are redirected in order to streamline the links. An area feature is no longer linked to *arc* directly, but to its geometric primitive *triangle*. As a result of the decomposition, an *area* (surface) feature then consists of one or more *triangles*. *Arcs* are linked to *triangles* through the *left* and *right* links, labelled in Figure 4.1 as Triangle_1 (2-Simplex_1) and Triangle_2 (2-simplex_2), respectively. The vertices of a triangle can be found through the arc-triangle and arc-node links. Each *node* is represented by one coordinate triple which consists of one *X*, one *Y*, and one *Z*. Since all nodes are assigned 3D coordinates, a plane equation can be derived from the three vertices of a triangle. This allows further derivation of information relating to terrain relief, such as elevation at any point, slope, and aspect. Additionally, the real-time visualization is no longer limited to orthogonal views (the traditional map). Popular visualizations, such as perspectives and stereo views — even with shaded relief — surface illumination or texture mapping, can be generated more efficiently, and can be combined with a cursor to access information from the '3D graphics.'

Following the FDS approach described by Molenaar (1989), the integrated terrain relief and features data model consists of three levels, described in the top-down manner:

1) Class level
This level consists of thematic class data types that maintain information related to the application, or the manner in which the features described in the next level will be used; for example, a line feature as a road, a surface feature as an industrial zone, a point feature as a city. The classes are mutually independent.

2) Feature level
This level consists of three feature data types: point-feature (0-complex), line-feature (1-complex), and surface-feature (2-complex). Each level implies the type of geometry to be used for its geometric representation. This level provides the interface to the user. Each feature type maintains a feature-class (FC) link to exactly one class data type.

3) Geometry level
This level consists of three geometric data types, that is, node (0-simplex), arc (1-simplex) and triangle (2-simplex). Each of them maintains a geometry-feature (GF) link to a feature it composes, so that a node may represent a point-feature, an arc (which is a straight line) may be a part of a line-feature, and a triangle must be a part of a surface-feature (Figure 4.1). Within the same level, there are also geometry-geometry (GG) links between two related geometric primitives; that is to say, an arc has two nodes — a beginning and an end— and it has two triangles — left and right. The node type maintains the georeference to the external space in the form of a coordinate tuple.

Note that the links represented as arrow-headed lines in Figure 4.1 only indicate that those links are possible, but not necessary. For example, some arcs are not part of any line-feature and some nodes are not point-features. Nevertheless, every node must be a vertex of a triangle and every arc must be an edge of a triangle. In comparison with FDS, a node may be isolated and an arc need not be an edge of a polygon. This interpretation becomes clear in chapter 5, where the mapping into a relational database structure is explained in detail.

In terms of simplicial complexes (Egenhofer et al 1989), this data model consists of 0-simplices (nodes), 1-simplices (arcs), and 2-simplices (triangles), the smallest data elements of 0, 1, and 2 dimensions respectively.

With respect to the transition from an object-based data to a tessellation-based model, the following requirements defining a decomposition scheme must be fulfilled:

1) A surface-feature (2-complex) is composed into a set of triangles (2-simplicial complexes).
2) A line-feature (1-complex) is composed into a set of contiguous arcs (1-simplicial complexes) that are in fact triangle edges.
3) A point-feature (0-complex) needs no decomposition; it is simply a node (0-simplex) that is a triangle vertex.

Following the syntactic approach of Molenaar (1994b), the above decomposition scheme can be mathematically described as follows:

$PartN\, [S_N, C_N]$

where PartN = part of relation at dimension N
S_N = simplex of dimension N
C_N = (complex) object of dimension N

A decomposition process that guarantees the above requirements must be made available to facilitate this. Delaunay and constrained triangulation can serve this purpose. The process is described in more detail in chapter 6.

By assigning 3D coordinates (x, y, z) to every 0-simplex, the mapping of this model in 3D metric space becomes meaningful in, for example, visualization, or the calculation of slope or surface area. Consequently, various topological operations and the derivation of topological relationships can be readily performed by using the basic binary relationships (Egenhofer et al 1989), because all objects are said to be decomposed into minimal spatial objects of their dimensions.

The data model presented in Figure 4.1 only accommodates single-theme GIS, which does not permit the sharing of the spatial region of different objects of the same dimension in a database. So the next step of the development of the integrated data model is to add the capability of handling multiple themes stored in the same database. The multi-theme concept developed by Kufoniyi (1995) described in chapter 2 can be adopted. By adding a sub-feature level that consists of three new data types between geometry and feature level, the more integrative data model as presented in Figure 4.2 is obtained (Pilouk and Kufoniyi 1994).

This further developed data model enables the sharing of spatial regions and the simultaneous representation of terrain relief. It is achieved by redirecting the links from geometry level to the sub-feature level instead of direct links to the feature level. So, instead of being an explicit part of the feature, each geometric primitive is then a part of a sub-feature and thus indirectly a part of a feature. The sub-feature is in fact an aggregation of the geometric primitives. The sub-feature level ensures the connection to the theme by maintaining the 'part of' links to the data types at the feature level. Observe that the sub-feature data types represent overlapping spatial regions being shared by more than one feature of different themes. Each sub-feature can then be a part of more than one feature and therefore has a *heterogeneous* thematic property, while the feature itself is *homogeneous,* because it represents only one object of reality that has a unique property across its spatial extent (see chapter 2). A feature still belongs to only one thematic class, so it does not lose its property of being single-valued. This convention also suits data acquisition which is usually carried out per theme.

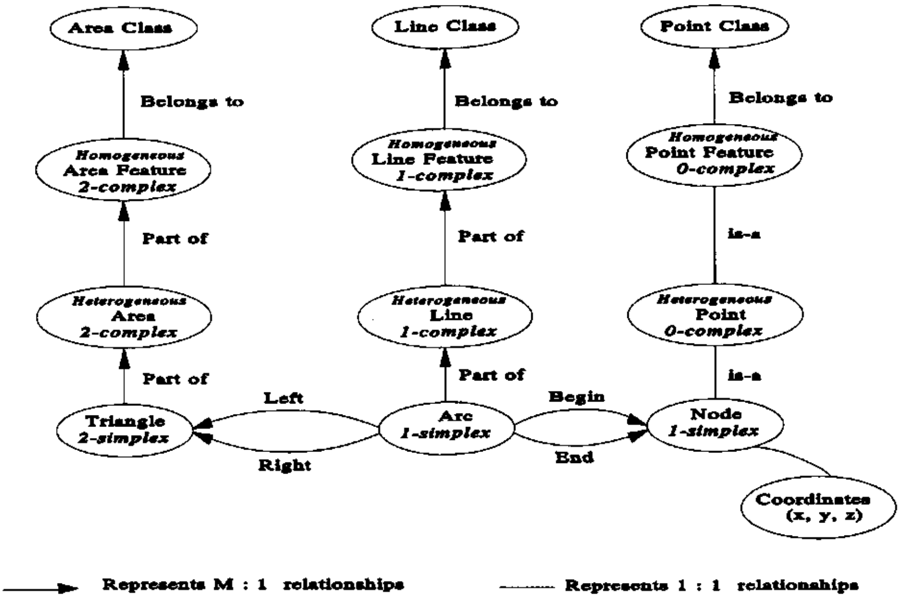


Figure 4.2 Integrated data model for DTM and multi-theme GIS (after Pilouk and Kufoniyi 1994).

The model implies the following decomposition scheme:

1) each area feature is decomposed into a set of subareas; each subarea is still a 2-complex and is therefore further decomposed into a set of triangles
2) each line feature is decomposed into a set of lines; each line is still a 1-complex and is therefore further decomposed into a set of arcs
3) no decomposition is needed for any point feature; it is considered as a 0-complex and a 0-simplex (that is to say, a node) at the same time.

The concept of Delaunay and constrained triangulation can still be used for the decomposition into primitives, but, only after the decomposition of features into sub-features for which the typical overlaying process in GIS can be used.

4.3 TEN-based Data Model

The TIN-based data model presented in section 4.1 is limited to applications that consider single-valued surfaces (Figure 1.1). It has no capability to serve applications that need to deal with multi-valued surfaces, or solid bodies. Applications in geology, geo-science, architecture, civil engineering, urban planning, facility management and environmental monitoring all require full 3D spatial information, so that an integrated data model that can represent multi-valued surfaces and solid objects is needed.

The TIN-based data model has to be extended to facilitate handling of 3D objects in particular in order to stretch the capability of the integrated data model in both dimensionality and computability. The triangular network can be generalized into a tetrahedral network. Delaunay triangulation can also be generalized for tetrahedronization (see chapter 6).

The general properties of a tetrahedron are the same as a triangle's (section 4.1); each is a simplex of its dimension and convex. Some important properties of their networks, for example, locality, fidelity and capability of embedding features are also the same. The latter indicates that the geometry of the features can also be maintained within the tetrahedral network (TEN), which means that TEN also has the capability of maintaining human knowledge about the real world as follows:

1) a body-feature is a contiguous set of tetrahedrons that is a subset of the TEN
2) a surface-feature is a contiguous set of triangles that are faces of tetrahedrons
3) a line-feature is a contiguous set of arcs that are edges of tetrahedrons and triangles
4) a point-feature is a vertex of at least one tetrahedron.

The above statements may be treated in addition to the set of conventions for 3D FDS; for example, self-overlapping or self-intersecting of a feature is not allowed.

For the aspect of interpolation, the bivariate interpolation methods, for example, the weighted average (Tempfli 1982), can be generalized into trivariate; that is to say, values are estimated such that $p = f(x, y, z)$.

For better understanding, we first discuss the single-valued variant as shown in Figure 4.3. Compared with the TIN-based data model in Figure 4.1, the main differences (apart from the number of data types) are the GG-links between arc and triangle. In Figure 4.1, the arc maintains the left and right links to the triangle, while in Figure 4.3, the triangle provides three links to the arc. These three links differentiate arcs as three triangle edges. This differentiation is needed to normalize the many-to-many link from triangle to arc into three many-to-one links; that is to say, a triangle has arc X as 'edge-1', while arc X can be an 'edge-Y' (Y = 1 to 3) of many triangles in 3D space. The left and right links from the arc to the triangle are eliminated since, in 3D, more than two triangles can share one arc. The geometric type 'triangle' is comparable to 'face' in 3D FDS. However, each triangle has only three edges and three nodes, so it is not difficult to determine the triangle's orientation by ordering the three edges. This order can be subsequently recorded for each instance of the triangle data type and does not consume additional storage. Such order makes it possible to omit the edge data type that keeps the information about face orientation that is important in 3D FDS, where the number of edges and nodes of the face data type can be varied. Storing the direction of each arc in the edge data type helps avoid the determination of face orientation, which may require considerable processing time. The orientation of each triangle helps to further determine the first and the second tetrahedrons situated on the positive and negative normal of the triangle respectively. (These are comparable to the left body and right body of a face in 3D FDS).

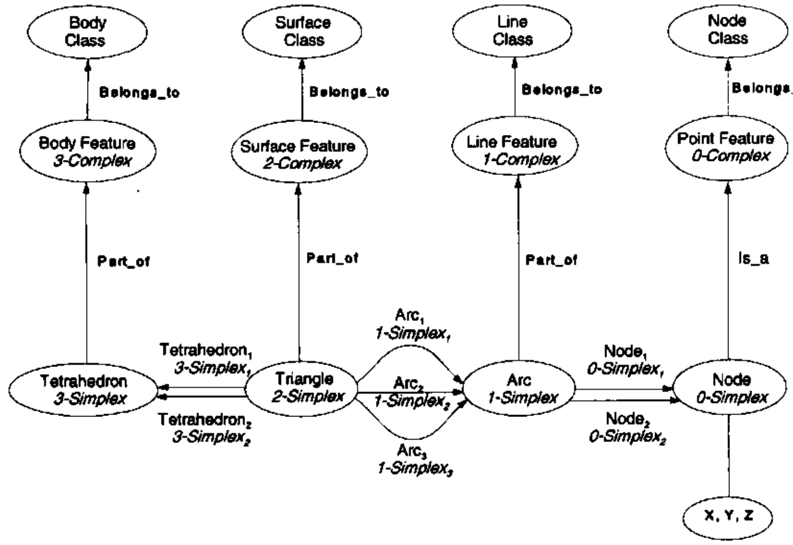


Figure 4.3 Tetrahedron-based data model.

A more precise description of the TEN-based data model can be given in terms of FDS together with simplicial complexes.

(1) An instance of the node (0-simplex) data type has x, y, and z coordinate types as its attributes. It may be a part of an instance of a point feature (0-complex) type.

(2) An arc (1-simplex) data type is defined as a straight line; it is therefore composed of only two instances of the type node, one on each end. It may be defined as a part of an instance of a line (1-complex) feature type.

(3) A triangle (2-simplex) data type is composed of three arcs. It is shared by two tetrahedrons (3-simplices), one on each side of its plane (called the 1st and 2nd tetrahedron respectively). A triangle may be a part of a surface feature.

(4) A tetrahedron (3-simplex) is a part of a body (3-complex) feature.

Observe that the tetrahedron data type does not carry any geometric description (triangular faces, edges, vertices), since its components can always be found from the geometric links with the triangle data type, being either the first or the second tetrahedron of a triangle (comparable to left or right body in FDS terminology).

To extend from single-theme to multi-theme, we must augment the sub-feature level by one more data type — the sub-body — and redirect all necessary links in the same way as for the TIN-based data model. The TEN-based version that is capable of handling 3D objects with multiple thematic representation is shown in Figure 4.4.

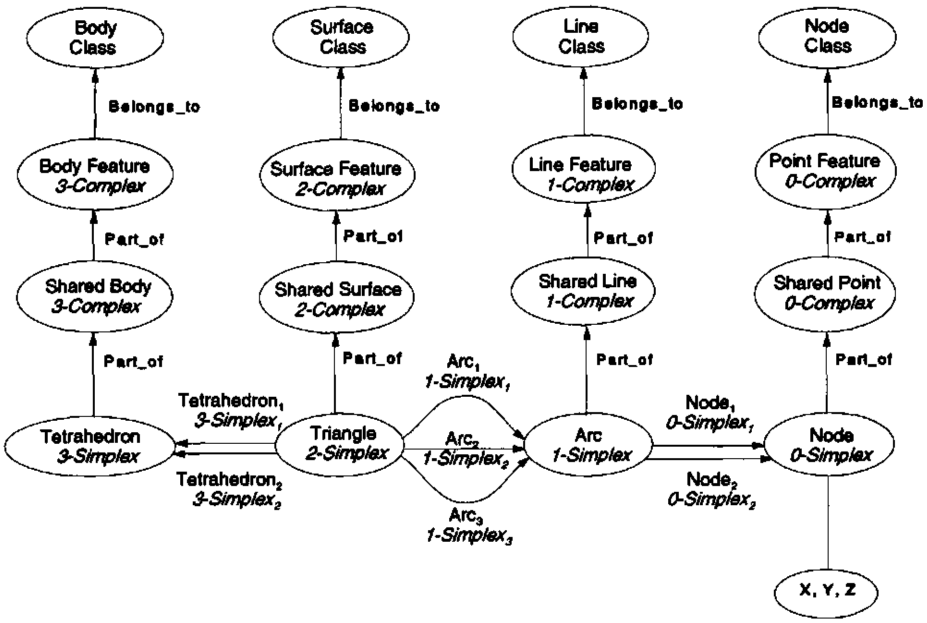


***Figure 4.4** Multi-theme tetrahedron-based data model.*

4.4 Generalized n-dimensional Integrated Data Model

Observing the similarities between the data models shown in Figure 4.1, Figure 4.2, Figure 4.3 and Figure 4.4, we can establish a general concept of an integrated data model based on irregular tessellation which may be useful for the study of multi-dimensional spatial information. In the different stages of the development, proceeding from TIN-based to TEN-based, both single-theme and multi-theme can be formalized. Theoretical support to this generalization is given by:

1) the FDS, which clearly represents relationships between the real world objects and how components of their representations are related in the spatial model
2) the simplicial complexes, which help simplify the spatial objects and systematically and consistently map them into the representations in the model
3) graph theory, which can be used to rigorously describe the representations and which also provides the mechanism to ensure the integrity of the overall representations, that is, the irregular network in this case.

An important benefit of having a theoretical basis for the tessellation-based integrated data model as a basic standard is that the compatibility across different dimensions can be established, thus:

1) It is more convenient for the user to decide what kind of data model to select; single-theme or multi-theme, and in what dimension. The user can instantiate a requirement as an input parameter to the generic data model and obtain the suitable model for the application. Users need not worry whether the databases at hand are based on or limited to a certain dimension. The generic data model makes possible the handling of data across different dimensions.
2) The user can navigate in different databases from one dimension to another dimension via the compatible links in various network structures, for example, from body, tetrahedron, triangle, arc, node and coordinates, provided that other databases also adopt the generic data model. In this sense, the generic data model can be regarded as dynamic.
3) The more efficient organizing, sharing and exchange of data and the elimination of disparity and redundancy lead to significant cost reductions. Avoidance of duplicate data collection is also feasible if the core database is widely accessible (Shepherd 1991).

Prior to the design of the generic version of the integrated data model, a set of definitions must be introduced.

4.4.1 Definitions

We limit our consideration to geometric modelling and recall the mathematical description of spatial objects following the theory of combinatorial topology described in chapter 2. This theory classifies spatial objects according to their spatial dimensions defining the spatial extent of objects. The simplest form of a geometric element for each dimension is called a simplex. For example, a node is a 0-simplex, an arc (a straight line consisting of two nodes) a 1-simplex, a triangle a 2-simplex, and a tetrahedron a 3-simplex.

Spatial position is defined by linking nodes to coordinates. Based on the concept of minimal objects and the notion that a minimal object in a higher dimension is composed of a specific number of minimal objects from lower dimensions, the following definitions can be given. (Note that some definitions in chapter 2 are repeated here for convenience.)

Definition 4.4-1: The metric dimension is defined by the number of linearly independent axes denoted by the coordinate tuple (Anton 1987).

For example, nodes are defined by coordinate pairs in 2-dimensional space, by (x, y, z) in 3-dimensional space, and by an n-tuple in n-dimensional space.

Definition 4.4-2: Any simplex of dimension n, called an n-simplex, is bounded by (n+1) geometrically independent simplices of dimension (n-1) (Amstrong 1983, Egenhofer et al 1989, Kinsey 1993) and n+1 simplices of dimension 0 (which are in fact the vertices of K_{n+1} complete graph; chapter 2; Finkbiner and Lindstrom 1987).

For example, a tetrahedron (3-simplex, K_4 complete graph) is bounded by four triangles (2-simplices) and four nodes (0-simplices); a triangle (2-simplex, K_3 complete graph) is bounded by three arcs (edges of a triangle, 1-simplices) and three nodes; an arc (1-simplex, K_2 complete graph) is bounded by two nodes. Arcs are geometrically independent if they are not parallel and none of them is of length zero.

Definition 4.4-3: Confining analysis to an n-dimensional metric space, two n-simplices are always incident at a simplex of dimension n-1.

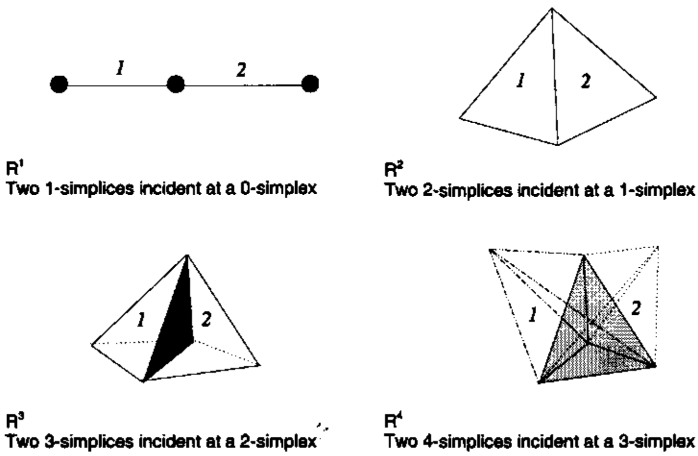


***Figure 4.5** Examples of two n-simplices incident at an (n-1)-simplex in R^n.*

The above definition can be turned into a component relation that is being shared. For example, in 1-dimensional space ($x \neq 0$), a node can be shared by at most two straight-line segments (whereas in two or higher dimensional space, a node can be shared by an infinite number of arcs); in a 2-dimensional space, an arc can be shared by only two triangles; in 3-dimensional space, a triangle can be shared by only two tetrahedrons. Similarly, in a 4-dimensional space, a tetrahedron can be shared by only two 4-simplices (see Figure 4.5 for graphic illustration).

Note that the above definitions only hold for simplices; they do not hold for complexes.

Given the above three definitions, a generic n-dimensional data model can be derived following the logic we observed when extending our model from 2D to 3D. Figure 4.6 illustrates the nD data model. The generic data model can be illustrated elegantly, and it has the advantage that objects of dimensions higher than three need not be given names. The term 'simplicial network' is, therefore, introduced to refer to the nD network. The definition of a simplicial network can be given:

Definition 4.4-4: An n-dimensional simplicial network is a network of simplices of different spatial dimensions, ranging from 0 to n-dimensions.

Definition 4.4-5: A finite set of simplices constitutes a complex that represents a spatial object.

A simplicial network should also fulfil the generalized Euler characteristic described in section 4.5.

Let us recall the similarities between simplices and complete graphs mentioned in chapter 2 (Figure 2.8). The definition of a simplicial network can be given in terms of graph theory.

Definition 4.4-6: A simplicial network is composed of a set of complete sub-graphs. The simplicial network itself need not be a complete graph. Either a simplicial network or each complete sub-graph can be, but need not be, a planar graph.

4.4.2 Single-theme and Multi-theme

The characteristic of a single-theme data model is that an instance of a feature type belongs to only one thematic class, and an instance of a geometric type (node, arc, triangle, tetrahedron) can be defined as a part of only one instance of a feature type (per theme). For a multi-theme data model, an instance of a feature type still belongs to only one thematic class, but an instance of a geometric type can be defined as a part of one or more instances of a feature type.[1]

Within the multi-theme concept, two types of complexes must be distinguished. A homogeneous complex (feature) is a set of contiguous simplices of the same dimension, all relating to only one theme. A heterogeneous complex (overlapping part) is a set of contiguous simplices of the same dimension that relate to more than one theme. A heterogeneous complex is part of two or more homogeneous complexes. By introducing homogeneous and heterogeneous complexes, we can solve the problem of 'many-to-many' relationships between geometric primitives and features. The formal definition of a multi-theme integrated n-dimensional data model can be given.

Definition 4.4-7: A spatial object is represented by a complex. A complex is a finite set of simplices. Two or more complexes can overlap; in other words, their intersection yields a non-empty but closed and contiguous set of simplices that are embedded in the network structure.

Figure 4.6 shows the nD data model for the single-theme concept. Figure 4.7 shows the corresponding multi-theme data model.

[1] For detailed description of the multi-theme concept, see Bouloucos et al (1993).

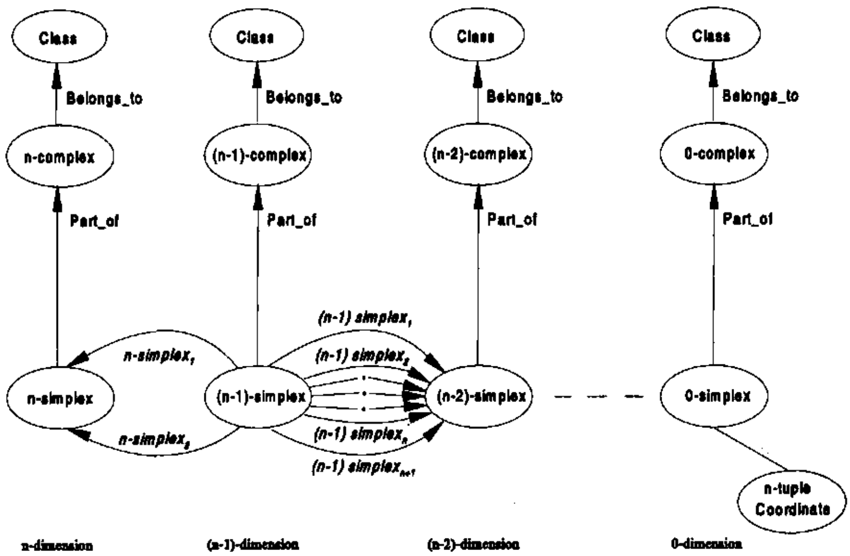


Figure 4.6 A generalized n-dimensional data model for single-theme.

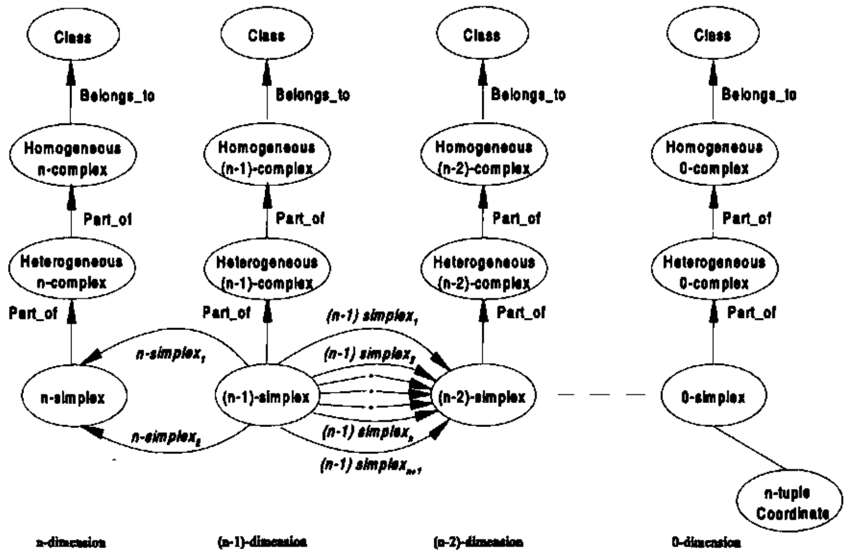


Figure 4.7 Generic multi-theme data model for n dimensions.

The multi-theme data model can be seen as an extension of the single-theme data model, as it accepts objects that share the same spatial region. This extension means two or more objects can have overlapping parts (body, surface, line, point). A typical example is of layers of soil and a volume of ground water sharing the same spatial region.

4.5 Euler's Characteristics

This section presents the consistency aspect of the integrated data models with respect to the graph theory that is crucial for ensuring the integrity of a database structured by the simplicial network formation. The Euler characteristics described below can be used to design the consistency checking mechanism. General 2D-based GIS applied Euler's equality, which has been proven to work efficiently for planar graphs. In the case of simplicial networks, the TIN-based model still complies with Euler's equality, since it is limited to 2D topology. For the tetrahedral network, even though a 3-simplex (K_4) is a planar graph, its combination may yield a non-planar one. Moreover, the objects of dimension higher than 4 are clearly non-planar. Therefore, this section presents a more general solution that can apply to n-dimensions. The first part reviews Euler's equality for planar graphs as a basis. The second part introduces the generalized concept, the formalization and some proofs.

4.5.1 Euler's Equality

We recapitulate Euler's equality for a planar graph as described in chapter 2 by the following equation:

$$n + f = e + i$$

where
- n = number of nodes
- f = number of faces
- e = number of edges
- i = degree of isolation

The degree of isolation indicates how many isolated regions are encountered. If the outer region is included in the graph, then it is also counted as a face; correspondingly, *i* should be increased by one. The above formula is applicable to the TIN-based data model.

4.5.2 The Generalized Euler Equality

To support the statement that simplicial networks of 3D and higher dimensions are non-planar, Kuratowski's theorem about the non-planarity of the graph is used.

Kuratowski's theorem: *A graph is planar if and only if it contains no sub-graph homeomorphic to K_5 or $K_{3,3}$* (Kuratowski 1930, see also chapter 2).

The above theorem implies that, if a graph contains a sub-graph that is homeomorphic to K_5 or $K_{3,3}$, then it is a non-planar graph.

We recall from chapter 2 that two graphs are *homeomorphic* (equivalent) if and only if they are isomorphic, or both of them can be obtained from the same graph by inserting or deleting nodes of degree two (a node that has only two edges connecting to it). The degree of a node is defined by the number of edges that meet at that node.

K_5 is a complete graph (Figure 4.9) where $K_{3,3}$ is a complete *bipartite* graph. Recall again that a *complete bipartite graph* is a graph where the nodes are divided into two subsets (for example, *a* and *b* in Figure 4.8), such that each node in each subset is connected to every node of the other subset, one edge per pair of nodes.

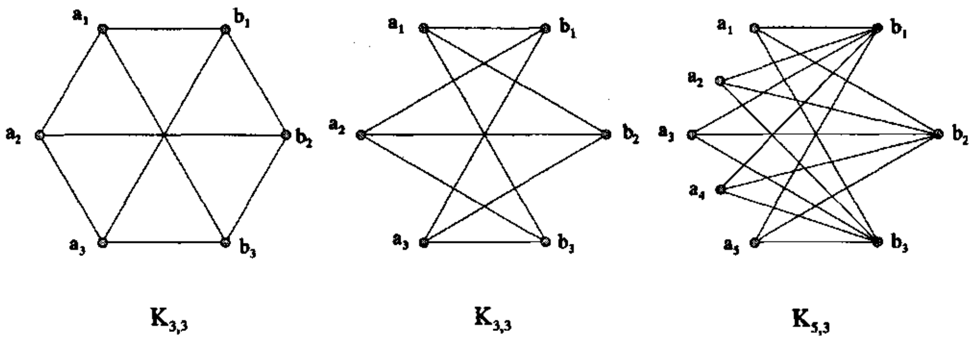


Figure 4.8 *Examples of complete bipartite graphs,* $K_{a,b}$.

By conducting a simple proof as graphically shown in Figure 4.9, it is clear that a tetrahedral network can contain sub-graphs isomorphic to K_5 or $K_{3,3}$.

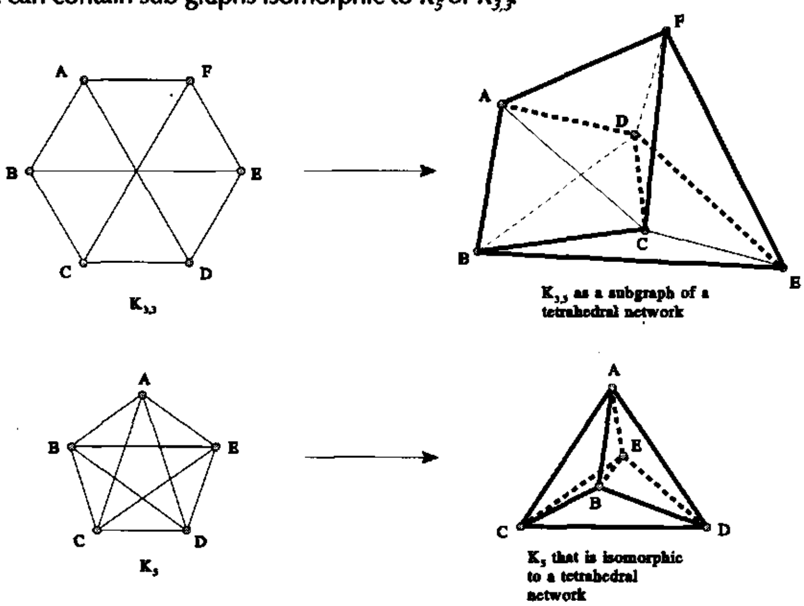


Figure 4.9 *Sub-graphs of a tetrahedral network that are isomorphic to* K_5 *or* $K_{3,3}$. *The thick lines indicate existing edges on the left side.*

The existence of such sub-graphs proves that a tetrahedral network is a non-planar graph and this also holds for any simplicial network of a higher dimension. The non-planarity implies that Euler's equality needs further generalization to be capable of application to the non-planar graph.

Sommerville (1929) has expressed an equation, similar to Euler's equality, for 3-cell complexes:

$$n - e + f - c = 1$$

where c = number of 3-cell complexes (see also Pigot 1992). Pilouk et al (1994) have presented the following equation applicable to a tetrahedral network:

$$Nodes + Triangles = Arcs + Tetrahedrons + 1$$

Note that the outer region is not included in the above formula. The variant for n-dimensions is:

$$0_{simplices} + 2_{simplices} + \ldots + k_{simplices} = 1_{simplices} + 3_{simplices} + \ldots + l_{simplices} + 1$$

where: k is even; $(0 \leq k \leq n)$
l is odd; $(1 \leq l \leq n)$
$0_{simplices}$ = number of nodes
$2_{simplices}$ = number of edges (arcs),
$k_{simplices}$ = number of simplices of dimension k
$l_{simplices}$ = number of simplices of dimension l

The above equation can be used to verify whether the simplicial network is well constructed. The imbalance indicates that the simplicial network is ill-formed, that is to say, having either free points, or intersecting edges, or faces presented in the network (Figure 4.10).

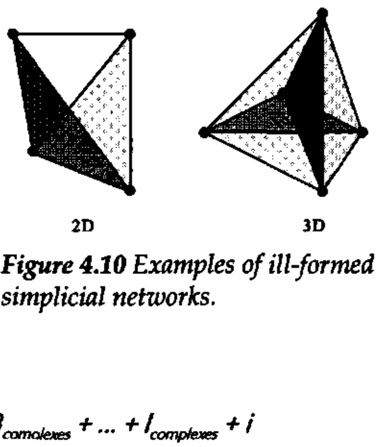


Figure 4.10 Examples of ill-formed simplicial networks.

Another variant of the generalized Euler equality for an n-dimensional complex is:

$$0_{complexes} + 2_{complexes} + \ldots + k_{complexes} = 1_{complexes} + 3_{comolexes} + \ldots + l_{complexes} + i$$

where: k is even; $(0 \leq k \leq n)$
l is odd; $(1 \leq l \leq n)$
i = degree of isolation

It is important to note that the degree of isolation must be determined correctly. There are different kinds of degree of isolation indicating the number of isolated objects. The isolated objects to be determined are:
isolated

- nodes with no connection to any arcs

- arcs (a dangling arc does not fall into this type)
- faces (a dangling face does not fall into this type)
- bodies, for example, holes in a body.

For n-dimensions, isolated objects of dimension 4 and above should also be included in *I*. Nevertheless, the type of isolation must be specified as a convention for each data model, so that an imbalance indicates that the convention is not met, and the system can issue a warning to the user. In the case of the integrated data model based on the simplicial network, the degree of isolation is equal to 1, because no isolated object other than the network itself is allowed.

The above formulae can be rewritten in a general form:

for one simplicial network with no isolation:

$$\sum_{k=0}^{k\le n} (-1)^k N_k = 1$$

for a simplicial network with *I* degree of isolation:

$$\sum_{k=0}^{k\le n} (-1)^k N_k = \sum_{k=0}^{k\le n} I_k$$

where n = dimension number
N_k = number of k-simplices
I_k = number of isolated objects or sets of mutually connected objects.

A mathematical proof of the above equation is given in appendix A.

4.6 Discussion

The developed simplicial network data model is based on four important concepts:

- formal data structure
- a constrained Delaunay network
- simplicial complexes
- graph theory.

FDS helps define representations of real world objects with respect to their relationships with geometric and thematic components. A constrained Delaunay network provides the basic concept for representing spatial units suitable for computation where existing knowledge represented in a form similar to FDS can be considered to be constraints. So, the computation result can be adapted to the situation in reality. Simplicial complexes and graph theory provide sound mathematical foundations for the simplicial network data model and rigorously support the generalization of this concept.

Both the direct and indirect representation of real world objects can be accommodated by the simplicial network data model. It permits refinement of the knowledge about the reality by deriving new information from existing facts presented as the direct representation type. The locality property of a simplicial network permits the adding of new facts into the spatial model without undue disturbance of the model as a whole. The local property applies to the elimination or updating of components of the model that no longer represent reality well. The adaptability, a property of an irregular network, makes modelling the variation aspect of the reality possible. Since the model is based on the complete tessellation of space, there are various means of navigation within the model, for example, using metric computation, order, or topology. With respect to the volume needed for storage of this kind of spatial model, the amount of data is expected to be less than that needed to store the components of a 3D spatial model separately. For example, storing terrain relief and terrain features in two separate data sets implies storing redundant elements where two representations coincide. The finiteness and convexity properties of each element of the model help simplify many operations. The data model complies with the generalized Euler characteristics, which can be used for checking logical consistency of the model with respect to its geometrical aspect. The last part of chapter 6 discusses consistency checking and gives some examples.

5
LOGICAL DESIGN

With the conceptual design of the integrated data model (IDM) presented in the previous chapter, we proceed to the logical design stage aiming at the unified data structure (UNS). This chapter explains the translation of the IDM into two kinds of UNS, using the relational and object-oriented approaches respectively. In contrast with the conceptual model, which is independent of the type of system and computing platform, the data structure comes closer to the implementation stage. The type of database management system (DBMS), which depends on a hierarchical, network, relational or object-oriented concept, has to be selected. The object-oriented approach contains the concepts of network and hierarchy and so demands more implementation effort. Not only do all objects, but also the methods of accessing each object, need to be carefully defined. Each DBMS type may only be available on one specific computing platform. The advantages and disadvantages in terms of speed and efficiency, ease of implementation, system maintenance and upgrade, and compatibility, have to be weighed to select an appropriate system for the implementation.

Since the purpose of this chapter is to describe the approach to translating the IDM presented in chapter 4 into UNSs, only a few single-theme variants of the IDM have been selected as examples. The same approach can be followed for the other variants.

5.1 Relational Approach

The reasons for using the relational approach include:

- ease of implementation; users can concentrate on the application rather than concern themselves about data access, since this is taken care of internally by the DBMS
- flexibility; the data structure can be readily extended or modified to delete or add more attribute columns, change the number of characters in a string data type, and so forth
- availability of various database management systems (Oracle, Informix, dBASE, Interbase) on different computing platforms and operating systems (PC with DOS, Windows, or UNIX; workstation, mini or mainframe with UNIX, VMS, or Windows)
- availability of software libraries and APIs (ODBC) and query languages (SQL, QBE)
- possibility of importing and exporting data to other systems, such as a spread sheet, or a word processor.

An important reason for choosing the relational approach is its maturity in providing a rigorous procedure for mapping a data model to a data structure. This process is known as normalization (see chapter 2). It is the mechanism ensuring the data integrity of the database in the face of updating anomalies. We obtain a set of skeleton tables here, using Smith's normalization procedure as presented in Roessel (1986), Kufoniyi (1989), Bouloucos et al (1990), Ayugi (1992), Pilouk and Tempfli (1992), Chhatkuli (1993). The relational approach also helps clarify and create the understanding about spatial relationships called for when establishing the object-oriented data structure in which the same kinds of relationships have to be represented.

5.1.1 Relational Data Structure for TIN-based Model

Following the five steps of Smith's normalization described in chapter 2, a TIN-based relational data structure is constructed as follows:

5.1.1.1 Constructing Dependency Statements

This step starts with the identification of the data fields to be stored in the database. In the data model in Figure 4.1, data fields are encompassed by ellipses, and the relationships are the labels on the lines connecting pairs of ellipses. The relationship between each pair of fields is analysed and then translated into a dependency statement. The list of dependency statements is given below (Pilouk and Tempfli 1993).

(1) A surface feature, which is identified by a *SID*, belongs to one *SCLASS* surface feature class.
(2) A line feature, which is identified by a *LID*, belongs to one *LCLASS* line feature class.
(3) A point feature, which is identified by a *PID*, belongs to one *PCLASS* point feature class and is represented by one *PNODE* node number.
(4) Each *NODENR* node has a position given by one *X* x-coordinate, one *Y* y-coordinate, and one *Z* z-coordinate.
(5) An arc is identified by *ARCNR*; it has one *Node1* starting node and one *Node2* ending node, and at most one *Tri1* triangle on its left side and at most one *Tri2* triangle on its right side.
(6) An *ARCNR* arc represents at most one *ALID* line feature.
(7) A triangle is identified by *TRINR* and represents at most one *TSID* surface feature.

5.1.1.2 Mapping from Dependency Statements into Dependency Diagram

From the above list of dependency statements, the corresponding dependency diagram can be drawn as in Figure 5.1. The attributes (data fields) are shown within bubbles. A line between two bubbles indicates a relationship between one data field and the other. A single-headed arrow indicates that it is a single-valued dependency; a double-headed arrow indicates a multi-valued dependency. More than 1 bubble covering a data field indicates that not all the relationships may apply to every instance of the data field. For example, an *ARCNR* should have a left and a right triangle (*tri1* and *tri2* respectively) but may not be part of a line feature. A number adjacent to a line between two bubbles indicates the dependency statement number. The indicator of the number of differently named fields having a common field type (*eg TRINR*, *tri1*, and *tri2* are of the same field type representing triangle identifiers) is the *domain flag*; it is shown as a number in a small triangle.

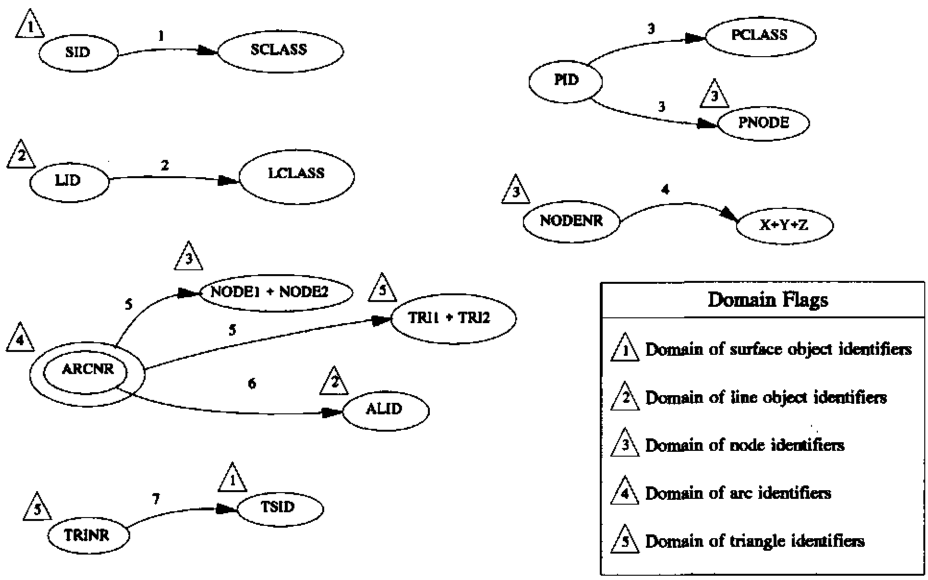


Figure 5.1 Dependency diagram of the proposed integrated DTM-GIS data model.

5.1.1.3 Composing Relational Tables from Dependency Diagram

Tables are first composed from the single-valued dependencies and then from the multi-valued dependencies. A bubble with no arrow pointing to it becomes a primary key field in one table. A target bubble becomes a data field in the same table. A bubble pointed to by an arrow and having a domain flag also becomes a foreign key field in the same table. In the case of multi-valued dependency, all the data fields with emanating arrows comprise primary keys. Special care should be taken here if there are more than three fields comprising a primary key; the table may not be practicable, since it would result in bad response times. The solution is to split the table into two by introducing a surrogate key acting as the primary key in one table and as a foreign key in the other. The following tables result (see also Figure 5.2):

R1: NODE (NODENR, x, y, z)
R2: ARC (ARCNR, node1, node2, tri1, tri2)
R3: TRISURF (TRINR, tsid)
R4: ARCLINE (ARCNR, alid)
R5: POINT (PID, pclass, pnode)
R6: LINE (LID, lclass)
R7: SURFACE (SID, sclass)

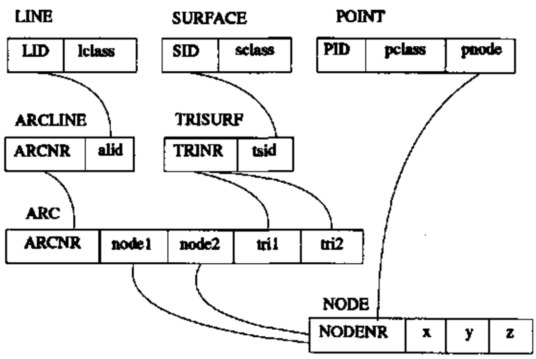


Figure 5.2 TIN-based relational data structure.

For convenience, the relational tables are labelled here by codes R1 to R7. Each table has a table name shown outside the bracket. Inside the bracket is the primary key, with its name shown in capital letters, and the set of attributes. The tables R1 and R2 represent geometric primitives and all the necessary topological relationships; for example, an arc has two nodes for its start and end, and two triangles on the left and the right side. R3 and R4 represent part-of relationships between geometric primitives and features; they are the same as R5, except that R5 also represents the thematic classification resulting from the one-to-one relationship between a node and a point feature. R6 and R7 represent thematic classes for line and area features.

5.1.2 Relational Data Structure for a TEN-based Model

The TEN-based data model can be mapped into a relational data structure by following the same procedure as for a TIN-based model. Most of the dependency statements are the same as for the TIN-based model. Some statements, however, have to be modified and some additional statements are required.

The following dependency statements result:

(1) A body feature, which is identified by a *BID*, belongs to one *BCLASS* body feature class.
(2) A surface feature, which is identified by a *SID*, belongs to one *SCLASS* surface feature class.
(3) A line feature, which is identified by a *LID*, belongs to one *LCLASS* line feature class.
(4) A point feature, which is identified by a *PID*, belongs to one *PCLASS* point feature class and is represented by one *PNODE* node number.
(5) An arc is identified by *ARCNR* and has one *NODE1* starting node and one *NODE2* ending node.
(6) Each *NODENR* node has a position given by a one *X* x-coordinate, one *Y* y-coordinate, and one *Z* z-coordinate.
(7) A triangle is identified by *TRINR* and represents at most one *TSID* surface feature; it has at most two tetrahedrons *TET1* and *TET2* attached to it, one on each side of the facet. It has at most 3 edges, *EDGE1*, *EDGE2* and *EDGE3*.
(8) An *ARCNR* arc represents at most one *ALID* line feature.
(9) A tetrahedron is identified by *TETNR* and represents at most one *TBID* body feature.

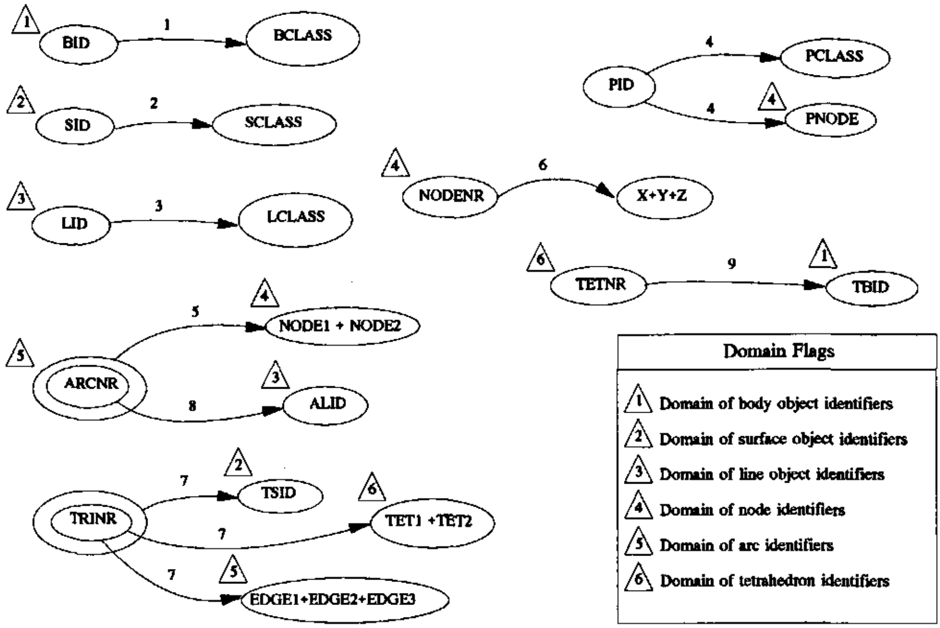


***Figure 5.3** Dependency diagram of the tetrahedron-based data model.*

Figure 5.3 shows the dependency diagram derived from the above list of dependency statements.

The following ten relations (tables) are obtained from the normalization process. The links are shown in Figure 5.4.

R1: Node (NodeNr, x, y, z)
R2: Arc (ArcNr, node1, node2)
R3: Triangle (TriNr, tet1, tet2, edge1, edge2, edge3)
R4: Tetra (TetNr, tbid)
R5: TriSurf (TriNr, tsid)
R6: ArcLine (ArcNr, alid)
R7: Point (Pid, pclass, pnode)
R8: Line (Lid, lclass)
R9: Surface (Sid, sclass)
R10: Body (Bid, bclass)

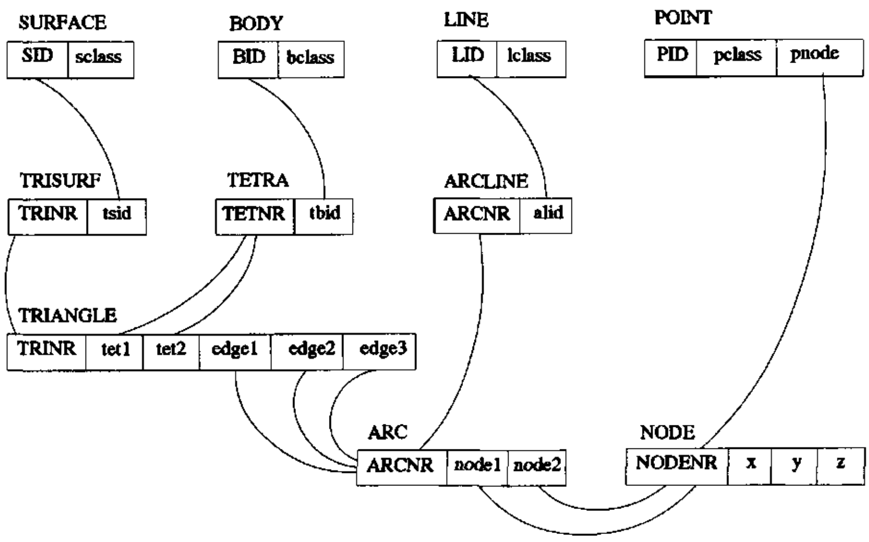


Figure 5.4 A TEN-based relational data structure.

R1, R2 and R3 can be regarded as geometry tables. R4, R5 and R6 are geometry-feature tables. R7 is a geometry-feature-class table. R8, R9 and R10 are class tables. Note that table R4 maintains no other information than a tetrahedron number (*TetNr*) and an identifier of the body feature (*TBID*) to which it belongs. We can only search for the geometric components of a tetrahedron of interest via the R3 (Triangle) table by matching the attribute value of the tetrahedron (*TetNr*) with either attribute value of *tet1* or *tet2*. Once the match is found, the next step is to get each of the three attribute values of *edge1*, *edge2* and *edge3* of the R3 table as a key to search for the match with the *ArcNr* in the R2 (Arc) table. If the match is found, we must get the attribute values of *node1* and *node2* and use each of them to search for the match with the *NodeNr* in the R1 (Node) table to get the coordinates x, y, and z for each respective node. In this way, we can use this database for 3D interpolation and for responding to a wide range of queries.

5.1.3 Relational Data Structure for an n-dimensional Data Model

The generic integrated n-dimensional data model was presented in the previous chapter, so mapping into the corresponding n-dimensional UNS will not be further elaborated here, since the procedure followed for the TIN-based can also be applied to a TEN-based data model. This section is restricted to the end results, included here for the sake of completion.

By observing the number of tables in the TIN-based and TEN-based UNSs and applying mathematical induction, it is possible to intuitively predict the number of tables necessary for the n-dimensional UNS. There are *n* geometric tables, *n* geometry-feature tables, *1* geometry-feature-class table (*ie*, for the link between node and point feature and class), and *n* feature-class tables. The set of relational tables will resemble the following:

Geometry
G_1 (S_0Nr, Crd_1, Crd_2, Crd_3, ..., Crd_n);
G_2 (S_1Nr, S_{01}, S_{02});
G_3 (S_2Nr, S_{11}, S_{12}, S_{13})
G_4 (S_3Nr, S_{21}, S_{22}, S_{23}, S_{24})
...
G_{n-1} ($S_{(n-2)}$Nr, $S_{(n-3)1}$, $S_{(n-3)2}$, $S_{(n-3)3}$, ..., $S_{(n-3)(n-1)}$)
G_n ($S_{(n-1)}$Nr, $S_{(n-2)1}$, $S_{(n-2)2}$, $S_{(n-2)3}$, ..., $S_{(n-2)n}$, S_{n1}, S_{n2})

Geometry-feature
GF_1 (S_1Nr, $c_1$id)
GF_2 (S_2Nr, $c_2$id)
GF_3 (S_3Nr, $c_3$id)
GF_4 (S_4Nr, $c_4$id)
...
GFn (S_nNr, $c_n$id)

Geometry-feature-class
GFC (C_0id, C_0class, S_0Nr)

Feature-class
FC_1 (C_1id, C_1class)
FC_2 (C_2id, C_2class)
FC_3 (C_3id, C_3class)
FC_4 (C_4id, C_4class)
...
FC_n ($C_n$id, $C_n$class)

where Crd_i is a coordinate component,
S_i represents the simplex or geometric primitive of i dimension,
C_i represents the complex or feature of i dimension.
S_{ij} represents the number j of i-simplex
C_{ij} represents the number j of i-complex

5.2 Object-oriented Approach

Although a relational database approach yields several advantages, certain important aspects are still lacking, which the object-oriented approach promises to fulfil. These aspects are:

- the relational approach is based on the Cartesian product. The joint operation on several tables causes long response times, particularly for the large amounts of data commonly found in GIS. The object-oriented approach includes the hierarchical and network data structures that can efficiently represent topology and facilitate navigation among different elements in the database, and so is likely to have better response times

- more complete and precise control over each individual object, especially where considerable ambiguity exists, as may happen when there are many different types of objects stored within the same database
- re-usability and extendibility of database management API (Application Program Interface) with no modification to the source code. These requirements mostly come from the community of developers, where the source code needs to be protected and hidden from users for commercial reasons. The API is implemented and compiled into a computer object code that encapsulates the objects, their attributes and accessing methods together in a form similar to software libraries, where only function names, methods of calling and function parameters are provided. The re-usability and extendibility of the API are provided through the inheritance mechanism. Users can further develop the API to fit their requirements by deriving new classes from existing classes, using an object-oriented compiler, such as C++, Smalltalk, or Object Pascal.

This section reports a study applying the object-oriented concept to the structured geoinformation based on the integrated data model. The focus is on the definition of objects and the design of object class hierarchies.

An object-oriented approach affords many alternatives to the design and implementation with respect to different abstractions of the real world. Worboys et al (1990), Kainz and Shahriari (1993) have presented similar designs in which the thematic class is defined as a parent that passes all aspects onto the geometric class. Their approach may be considered too rigid if multiple representations are needed. Multiple representation requires different types of geometry to be chosen for the representation of an object, depending on the level of abstraction. If the geometric representation for an object has already been fixed at the design stage, it would not be possible for the user to select any other kind of geometric representation. If for example it is decided to represent a road as a line-feature, it would not be possible to represent the road later as a band, as might be needed for abstraction on a larger scale, since the band is an area-feature. The whole hierarchy would have to be redesigned for every different level of abstraction, which could result in many classes. This approach may therefore be regarded as an *ad hoc* solution.

The selected approach follows as strictly as possible the conceptual model defined by (Molenaar 1989). This conceptual model offers a natural way of handling geoinformation, especially when considering the aspect of object creation that relates to data acquisition. This approach does not fix the geometric representation of a feature at the design stage. It divides the components of a feature into two hierarchies. The inheritance hierarchy is used for the thematic attributes and the aggregation hierarchy for the geometric attributes. These two hierarchies are only combined at runtime (the construction phase of the spatial model), thus allowing the user to select different types of geometric representation for a feature. The requirement for multiple representation can be fulfilled. Comparing the two approaches, the latter is more versatile, but it is more difficult to implement and requires highly skilled software engineering.

The translation of the IDM into object definitions follows, using the abstraction mechanisms of the object-oriented paradigm presented in chapter 2, namely classification, specialization, aggregation, and inheritance. The top-down approach starts from the most generalized class and proceeds to the most specialized class. Part of the implementation to UNS is illustrated using the C++ object-oriented programming language.

5.2.1 The Object-oriented Definition of a Spatial Object

We recall that the abstraction of real world objects consists of two parts: geometric and thematic (see Figure 5.5). The geometric component contains information about shape, location and topology, while the thematic component contains human knowledge about other properties of the object (colour, name, ownership, function, and so forth). When applying the concept using the relational approach, the geometric and thematic attributes are linked through a feature identifier (see section 5.1.1). In an object-oriented approach, the geometric and thematic components are realized as objects that can be tied together by a 'feature object' through the encapsulation mechanism.

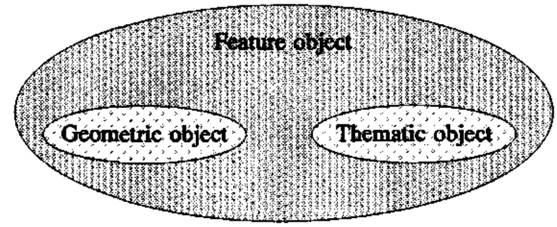


Figure 5.5 Object-oriented definition of a real world object. Feature object encapsulates geometric component and thematic component.

Since the object-oriented approach uses terminology in a similar way to our normal descriptions of the reality in the conceptual design stage, it is important to note that the discussion in this section is limited to the logical design stage in object-oriented programming terminology. The term 'class' is an abstract data type (ADT), whereas the term 'object' is used to refer to an instance of a class in object-oriented programming terminology. The terms class and ADT are used interchangeably.

5.2.2 Object-oriented Design Based on IDM

Classification

Molenaar (1993) provides a rationale for arriving at an object-oriented design. We are concerned with the logical design of an object-oriented data structure which defines a scheme for the storage of information about the spatial object. The representation of a real world object can be translated into three ADTs; ADTFeature, ADTGeometry and ADTTheme. The ADTFeature class at the top level of the hierarchy aggregates the ADTGeometry and ADTTheme classes at the lower level. The ADTFeature class defines the storage of a representation of a real world object as a whole. The storage of geometric and thematic descriptions about the real world object are defined by the ADTGeometry and ADTTheme classes respectively. A feature is an instance of an ADTFeature class. Likewise, a geometric object is an instance of an ADTGeometry class and a thematic object is an instance of an ADTTheme class.

Each of the three ADTs is considered as a general class with its own hierarchy and which still has to be defined further. A general class is subdivided into more specific classes to any desired level of refinement. The common status and behaviour of subclasses characterize their general class. Given the three general classes, we have to deal with three class hierarchies which have to be related at an appropriate time based upon the user's requirements and context (Molenaar 1993). Among the three general classes, the ADTFeature class has a 'primus inter pares' position. It must be considered to be at a higher level, since it constitutes and defines an aggregation hierarchy from the other two classes.

Thematic class

The ADTTheme class defines a data structure for the storage of thematic information which is highly related to the application domain in geoinformation, such as land use, transportation networks, water bodies, and the like. This class consists of information about common attributes and behaviours of descendant thematic objects. The purpose of having this class is to facilitate the definition of an inheritance hierarchy, minimizing redundancies and allowing re-usability between thematic information.

There follows a basic ADT for the storage of thematic information.

class :	ADTTheme
description:	General representation of thematic components of a spatial object
parent:	None
attributes :	code, name, texture, colour, ...
methods :	create, delete, show code, modify code, show name, modify name, ...
constraints :	

Geometric class

The geometric description of a spatial object is stored and maintained in the ADTGeometry class. This class defines a hierarchy of geometric primitives which comprise the geometric descriptions of a spatial object. The class provides a general data structure for the storage of components, describing the shape of each feature, its georeferencing scheme and its topological relationships with other features. One important aspect is that every geometric object has to be referenced to the ADTFeature class. This relates to the everyday life situation, where subordinates should always know their superiors. An ADTFeature object is comparable to the boss who represents and rules the group of subordinates. This assumption helps us make the organization more natural and efficient.

A basic ADTGeometry class is defined as follows:

class :	ADTGeometry
description:	General representation of geometric components of a spatial object
parent:	None
attributes :	identifier, reference to a feature class object (part of), ...

methods :	create, delete, show identifier, modify identifier, show feature, modify feature, display graphics, ...
constraints :	on creation, requires input of object identifier and reference to a feature object from the creator; on creation, sends a request to the reference feature object to update the geometric container of the feature, ...

Feature class

The ADTFeature class plays a central role in the representation of the real world. This class provides the interface between the users and the system. The class is also the entry point for the user to retrieve or store all components of the feature. The ADTFeature class is an aggregate class. Any instance of this class is a composite object, consisting of two components; an ADTGeometry object, and an ADTTheme object. In other words, both ADTGeometry and ADTTheme objects form 'part-of' an ADTFeature object.

class :	ADTFeature
description:	general representation of a spatial object
parent:	none
attributes :	identifier (to interface with users), reference to ADTTheme object, reference to the collection of ADTGeometry objects, ...
methods :	create, delete, add to geometric container a reference to ADTGeometry object, delete from geometric container a reference to ADTGeometry object, (graphics) display geometry, display thematic properties, ...
constraints :	on creation, requires identifier and reference to ADTTheme object from the creator, ...

5.2.3 Specialization of Classes

As we have said, each of the above classes can be further refined as more detailed objects. The following sections show the construction of class hierarchies using the specialization mechanism, resulting in inheritance hierarchies.

5.2.3.1 Thematic Hierarchy

The class ADTTheme can be specialized as various subclasses, such as road, railway, river, control point. The construction of this hierarchy is very subjective, depending on the user's point of view and application. There are, however, many advantages in modelling thematic information using an object-oriented approach, especially when an object has to be related to several themes at the same time. The object-oriented approach provides a straightforward solution to this representation through the multiple inheritance mechanism. A class can inherit properties from more than one parent class. Such a class then represents a combination of themes.

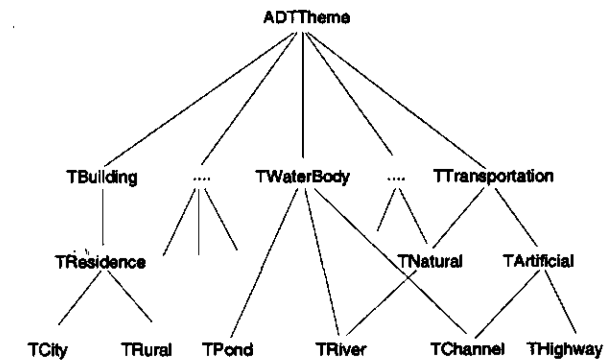


Figure 5.6 Thematic class hierarchy (adapted from Egenhofer et al 1989).

An example, taken from Figure 5.6, is the TRiver class which can be seen as 'is-a' TWaterBody and TNatural transportation network at the same time. By aggregating an object that belongs to the thematic hierarchy (for example, class TRiver) as the component of an object that belongs to the feature hierarchy (for example, line feature class), the object belonging to the feature hierarchy automatically carries multiple thematic information. However, ambiguity may arise in such a case; two parent classes may have attributes or methods of the same name and so need special attention and appropriate resolution. The designer of this class must decide from which parent the new class inherits the properties, otherwise the ambiguous properties have to be completely overridden. It may be necessary to resolve the ambiguity by setting up consistency rules as detailed by Kufoniyi (1995). Egenhofer and Frank (1989) and Kainz and Shahriari (1993) have reported some other examples emphasizing the construction of the thematic hierarchy. The specialization of thematic class is not elaborated further here.

5.2.3.2 Geometric hierarchy

The class ADTGeometry can be specialized for each geometric primitive or simplex—node, arc, triangle, tetrahedron and so forth, as shown in Figure 5.7. These classes inherit properties from the parent class ADTGeometry; each of them contains only the additional status and behaviours differing from its ancestor's.

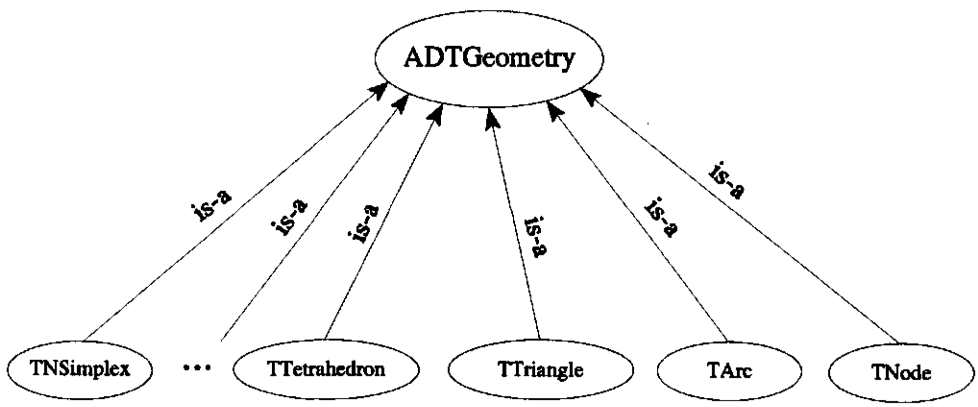


Figure 5.7 Geometric class hierarchy.

class :	TNode
description:	0D geometric component of the representation of a spatial object
parent :	ADTGeometry
attributes :	x, y, z coordinate, perspective transformed coordinate (xp,yp)
methods :	get coordinates, modify coordinates, (graphics) display, 2D, 3D, perspective transformation, ...
constraints :	requires 3D coordinates, reference to ADTFeature object, and a geometric identifier from the creator
class :	TArc
description:	1D geometric component of representation of a spatial object
parent :	ADTGeometry
attributes :	references to two TNode objects as begin and end nodes, references to two TTriangle objects on its left and right sides.
methods :	create, delete, (graphics) display 2D, 3D, ...
constraints :	requires identifier, references to two TNodes and ADTFeature objects from creator, ...
class :	TTriangle
description:	2D geometric component of representation of a spatial object
parent :	ADTGeometry
attributes :	references to three TNode objects as its vertices, three TArc objects as its edges, three TTriangle objects as neighbour triangles, slope, parameters of plane in normal form (a, b, c, d), ...
methods :	create, delete, get edges, get neighbours, get slope, get plane parameters, interpolate elevation for a given x,y coordinate, interpolate locations for a given z coordinate, (graphics) display 2D, display 3D, shade, ...
constraints :	requires a geometric identifier, references to three TArc objects, and reference to an ADTFeature object, ...
class :	TTetrahedron
description:	3D geometric component of representation of a spatial object
parent :	ADTGeometry
attributes :	references to four TTriangle objects as its faces, four TArc objects as its edges, four TNode objects as its vertices, four TTetrahedron objects as its neighbours, ...
methods :	create, delete, get vertices, get edges, get faces, get neighbours, interpolate value for a given x,y,z coordinates, interpolate contour surface, (graphics) display 3D, shade, ...
constraints :	requires a geometric identifier, references to four TNode objects, and reference to an ADTFeature object, ...

5.2.3.3 Feature Hierarchy

The class ADTFeature is specialized into four specific classes: point, line, area and body, as shown in **Figure 5.8**. Each derived class has its specific behaviours and attributes in addition to the behaviours and attributes of the parent class ADTFeature. A simple example is the draw operation. Drawing a point may only require drawing a pixel on a screen, while drawing a line, or an area, requires additional operations. The topology has to be used to navigate in the database to obtain all the nodes and their links before the pixels can be drawn along the line, or along the boundary of the area. The specialization also helps streamline the handling of the geometry and topology of each particular subclass. The design of related functions can be concentrated on specifically for each one in turn, with no fear of their interfering with each other, even if the functions of the different objects have the same function names.

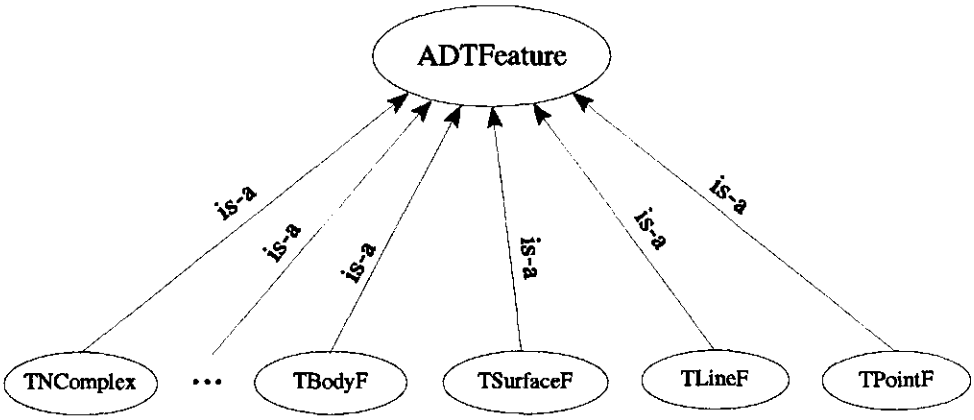


Figure 5.8 Feature class hierarchy.

class :	TPointF
description:	0D representation of a spatial object
parent :	ADTFeature
attributes :	...
methods :	create, delete, display geometric (draw node, 2D, 3D), display thematic, ...
constraints :	on creation, requires an identifier, references to ADTGeometry and ADTTheme objects from the creator, ...
class :	TLineF
description:	1D representation of a spatial object
parent :	ADTFeature
attributes :	bounding rectangle, ...
methods :	create, delete, display geometric (draw all component arcs as 2D, 3D), display thematic, ...
constraints :	on creation, requires an identifier, references to ADTGeometry and ADTTheme objects from the creator, ...
class :	TSurfaceF
description:	2D representation of a spatial object
parent :	ADTFeature
attributes :	bounding rectangle (cube)
methods :	create, delete, display geometric (draw all component triangles as 2D, 3D, shade), display thematic, ...
constraints :	on creation, requires an identifier, references to ADTGeometry and ADTTheme objects from the creator, ...
class	TBodyF

description:	3D representation of a spatial object
parent :	ADTFeature
attributes :	bounding box (cube)
methods :	create, delete, display geometric (draw all component tetrahedron as, 3D, shade), display thematic, ...
constraints :	on creation, requires an identifier, references to ADTGeometry and ADTTheme objects from the creator, ...

5.2.4 Aggregation of Objects

The ADTFeature class forms an aggregation hierarchy by taking objects belonging to the geometric and thematic hierarchies as its components (see Figure 5.9). This is a stage of assembling or manufacturing an instance of the ADTFeature class. Subclasses of this class, for example, TPointF, TLineF, TSurfaceF, TBodyF, are also of aggregate types; a TLineF object may consist of many TArc objects. For each subclass of the ADTFeature, the actual aggregation has to be done at runtime. This is because it is not possible to know at the design phase which specific class in the thematic hierarchy will be its thematic component. The dynamic referencing mechanism is the solution to this problem. The technique is first to define the aggregation, using the reference to a generic class (ADTTheme). During runtime, the user selects the more specific class (for example, class TRoad). Dynamic inheritance and aggregation take place here. The class that aggregates the class TRoad into the class TLineF is, in fact, derived at runtime. The TLineF object knows at that moment that its thematic component is of the specific class TRoad, which is the descendant instead of the generic class ADTTheme. The reference to class ADTTheme is then changed to class TRoad.

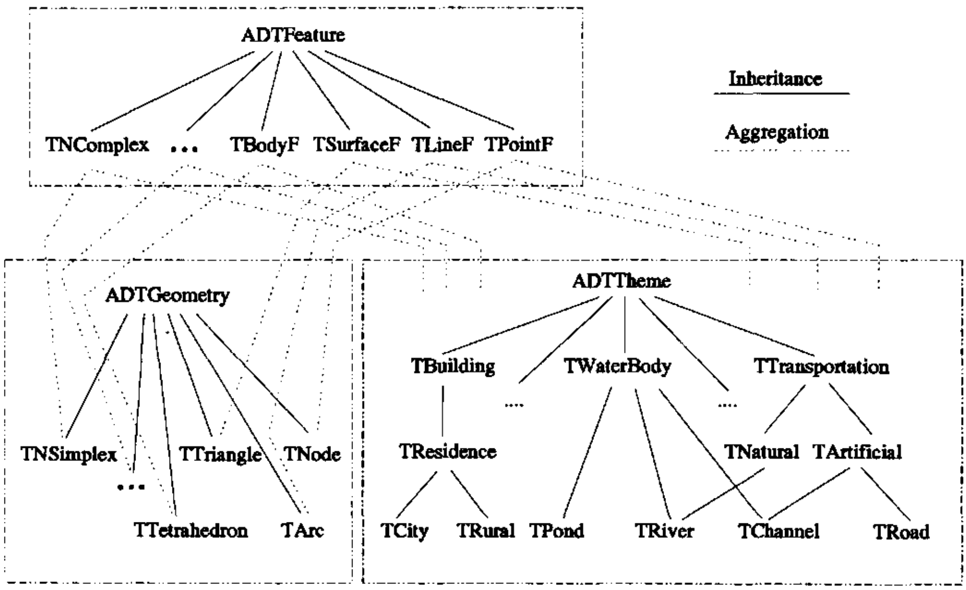


Figure 5.9 Relationship between class hierarchies.

5.2.5 Creation of Objects

In addition to the classes defined above, the system must provide container classes, each of which is specific to the objects of the ADTTheme, ADTGeometry and ADTFeature classes. The objects for each class should be created in an appropriate sequence. ADTTheme objects are the first to be created and registered into the container of ADTTheme. In practice, users should first define their own thematic hierarchies according to the purpose of the application. For example, if the geoinformation is to serve the management of a road network, the thematic hierarchy should start from 'general road' and then specialize down to 'primary road', 'secondary road', 'highway', 'superhighway', and so on.

The ADTFeature objects are created next. Every instance of this class must be registered into the container of ADTFeature. The user defines which theme is to be represented by which kind of feature. The notions of scale and resolution govern the choice. For example, an application using small scale maps may represent towns as point features (represented by TPointF class), while on a larger scale they may be represented by area features (represented by TSurfaceF class). To comply with this presumption, the definition of ADTFeature class must consider the specialized classes of ADTGeometry and ADTTheme. Molenaar (1993) discusses this issue in detail.

The ADTFeature object created at this stage has to be considered incomplete, because of the lack of geometric content (see Figure 5.10). Completion can only occur when the reference to the geometric container has been established and the geometric container filled with all necessary references to ADTGeometry objects (see Figure 5.11).

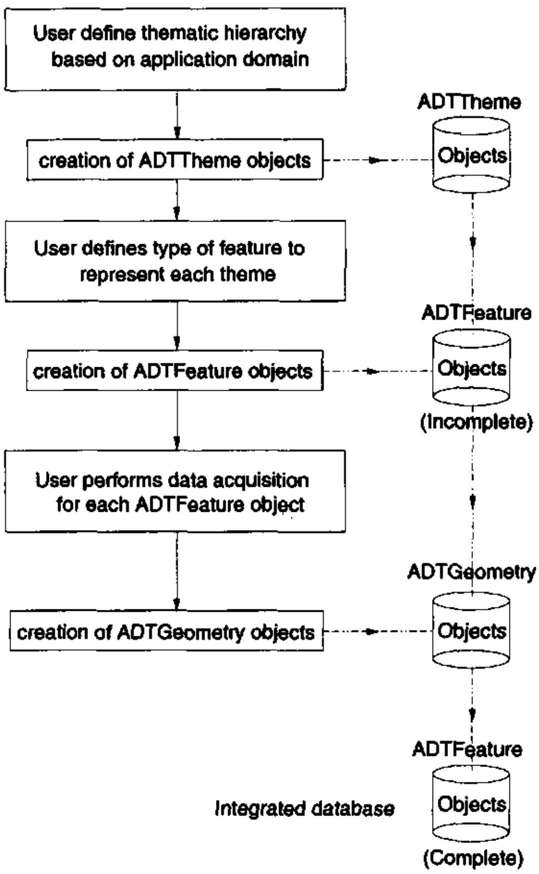


Figure 5.10 Steps to create objects.

The ADTGeometry objects are the last to be created. The reason for this is that ADTFeature objects are not georeferenced before the stage of data acquisition. The specialized class of ADTFeature defines the specialized class of ADTGeometry object to be captured. When the user decides that a river will be represented by a line feature, the ADTGeometry objects to be captured are of the TArc class, and certainly not of the TTriangle class. Because the TArc object has references to two TNode objects, the user is forced to capture (create) two TNode objects prior to the creation of the TArc object. This engenders strict discipline in collecting data with expected high consistency.

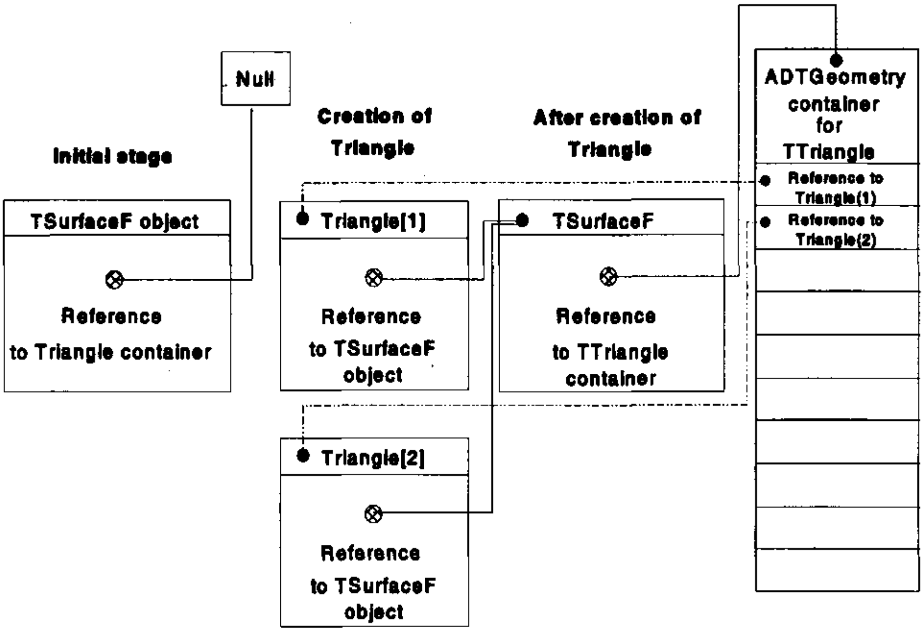


Figure 5.11 Referencing scheme.

5.2.6 Behaviour of Objects in the Database

By defining the hierarchies and relationships between the objects as outlined above, every object can respond to the message it receives from another object (whether self-activated or not). This kind of operation is efficient and consistent, since the appropriate operation is specific. For example, a user may wish to display the area features in a perspective view. Using a broadcasting mechanism, the user sends a message, such as 'Draw-3D,' to all objects belonging to the class TSurfaceF. On receiving this message, each TSurfaceF object then reacts to it by sending another 'Draw-3D' message to all of its component TTriangle objects (by searching in its geometric container). After each TTriangle object has received the message, it is sent to all three vertices, the TNode objects. The message asks the TNode objects to make a perspective transformation and then, using the transformed coordinates, to draw straight lines between themselves, perhaps adding colour-fill or shading if so requested.

Considering the aspect of spatial access, we observe that TLineF and TSurfaceF objects include references to geometric containers of classes TArc and TTriangle respectively. Taking a TSurfaceF as an example, and given a spatial location, the spatial search operation can be coarse to fine using, for example, the bounding rectangle of the TSurfaceF object as spatial index. The containment test is then performed in a simplified and fast manner. On receiving a positive result, the spatial search is then limited to TTriangle objects which are components of the TSurfaceF object. The search can then be performed using a reference to the TTriangle container that is one component of the TSurfaceF. This TTriangle container, which is specific to the TSurfaceF, contains a series of references to TTriangle objects. The references to the object offer a fast way of accessing the object component, that is, the attributes and methods.

Regarding the interfacing of system and user, the objects of classes under the ADTFeature hierarchy should provide all the necessary interfaces. During the database operation (deleting, modifying and so on), the objects of classes under the ADTGeometry hierarchy should not be directly accessible to users and should be under the complete control of each specialized ADTFeature object.

The example of implementation given as C++ object-oriented programming language is presented in Appendix B. The focus is on the aspects of object creation, dynamic referencing and inheriting. For simplicity, a fixed size array is chosen as the container of references to each specific ADTGeometry object. Other versions of C++ offer more powerful container class libraries which can be used for the real implementation. We have implemented part of this definition in our software ISNAP (Integrated Simplicial Network Applications Package). The experimental investigation has demonstrated the feasibility of the design, thereby stimulating further investigation into the matter, for example response time and efficiency in spatial search operation. The implementation using this logical design in a commercial OODBMS environment still needs further exploration.

5.2.7 Comparison with Other OO Approaches

In comparison with the approach presented by Webster and Omare (1991), Worboys et al (1990), Kainz and Shahriari (1993), that presented here offers a more flexible structure where the users have the freedom to select different types of geometric representation per thematic class with respect to the scale of data acquisition. As an example, a city can be represented by an area object when the data is acquired from a map of scale 1: 50000, or a point object if acquired from a map of 1: 500000. The other approaches mentioned above have only adopted the inheritance hierarchies. For example, Webster and Omare (1991) defined a point feature as a supertype (parent or ascendant) of the node class, where a geometric class is a subtype (child or descendant) of a thematic class, which is similar to the approach used by Kainz and Shahriari (1993). Worboys et al (1990) defined a district class as a specialization (child) of a polygon class where a feature object class is a child of a geometric class. In both cases, the consequence is that only one type of geometric representation is allowed in a hierarchy of that feature object. This restriction might prove too stringent and so the rigid inheritance approach can only be used as a logical design for a particular application. The whole object hierarchy has to be redefined when the database has to be upgraded to use a more precise geometry.

The rigid inheritance approach described above may not be suitable for UNS. In UNS the possibility of having multiple geometric representations per class helps minimize the number of features stored in the database. The approach suggested in section 5.2.4 of using the aggregated hierarchy permits the selection of the type of representation according to whatever is available at the time of data acquisition. For instance, there may be many cities on the map with different possible representations, such as point and area, depending on their sizes. If the application only needs cities to be represented as points, each point object can be derived from the area object residing in the database, for example by using a cartographic generalization function of that area object.

5.3 Discussion

The translation from the IDM into a relational and an object-oriented UNS has been presented. Note, however, that only the necessary attributes are included in the relational UNS. This approach offers a quick and simple way of implementing an integrated database. Although good performance in terms of response time may not be obtained, the realization of all the necessary relationships between the data elements is facilitated. The control of database consistency depends on this minimum set of relationships between the data elements, even when the object-oriented approach is used. Significant performance gain in terms of response time is expected from the object-oriented approach, because links and pointers are used for navigating in the database instead of Cartesian products, as in the relational approach. Joining several tables together results in a long response time during a data retrieval and search operation in a large database. Most relational DBMSs offer a simple solution by creating an index file that can be thought of as a reordering of the records, using criteria on a selected column of each table. A typical indexing method is a binary tree (B-tree, Knuth 1973), which may not, however, be suitable for indexing spatial data. The object-oriented approach permits the implementation of a more suitable spatial indexing method, such as Grid File, R-tree (Guttman 1984, Oosterom 1990). This method, however, requires a greater effort in implementing the index structure. Some DBMSs, such as Illustra (1994), claim to have offered a solution by providing R-tree to support efficient access to spatial data. Note that the index structure provides additional relationships among data elements. Most of these relationships can be inferred from the minimum set of relationships obtained from normalization in the relational approach and so they may be considered redundant, which asks for special care during the updating of the database.

It is worth mentioning that a UNS derived from the IDM can be managed by a single DBMS. Users only deal with one system and one user-interface. The time required for studying the use of the different commands of different databases (even for the same kind of operations) can be reduced, allowing users to concentrate on the actual application.

6
CONSTRUCTION OF THE MODEL

The construction of the integrated 3D geo-spatial model based on the conceptual design explained in chapter 4 has to follow certain steps. All data elements that are components of the model must be organized in a structure that conforms to the logical design explained in chapter 5. The first part of this chapter suggests steps for the construction of the model. The second part presents relatively simple operations for the most important steps of this construction, focusing on the constrained network formation. The data structuring of 3D features according to 3D FDS is also included in this chapter, since the features of 3D FDS are used as constraints in network construction. The checking of internal consistency for the network formation is discussed in the last part of this chapter.

6.1 Steps for the Construction of an Integrated 3D Spatial Model

This section gives an overview of how to achieve an integrated 3D geo-spatial model based on the simplicial network (SN) concept.

Taking into account the fact that existing data should be utilized as components of the new 3D spatial model, the construction steps are:

1) Structure the 2D data of determinate spatial objects according to SWM or MWM (see chapter 2); these are called 2D FDS data.
2) Obtain terrain relief data in the form of a grid or TIN.
3) Introduce the height component for each node of the 2D FDS data that have only the planimetric component by means of interpolation, using the DTM data set.
4) Convert the DTM to TIN if it is in the form of a grid. Redundant data with respect to relief representation may be eliminated to reduce the volume of data.
5) Combine the nodes that are vertices of TIN with the nodes of the 2D FDS data.
6) Embed all features into TIN by performing constrained triangulation, using terrain features as constraints. The result is recorded in 2.5D UNS.
7) Structure the 3D data of determinate spatial objects according to 3D FDS; this is called 3D FDS data.
8) Obtain 3D representation of indeterminate spatial objects. These may be in the form of a voxel array or tetrahedral network. Convert into a tetrahedral network if it is in the form of voxel array
9) Combine all nodes from 2.5D UNS, 3D FDS, and TEN.
10) Perform constrained tetrahedronization using data from 2.5D UNS and 3D FDS as constraints. Record the output in the form of 3D UNS.

The integrated 3D database obtained at step 10) as shown in Figure 6.1.

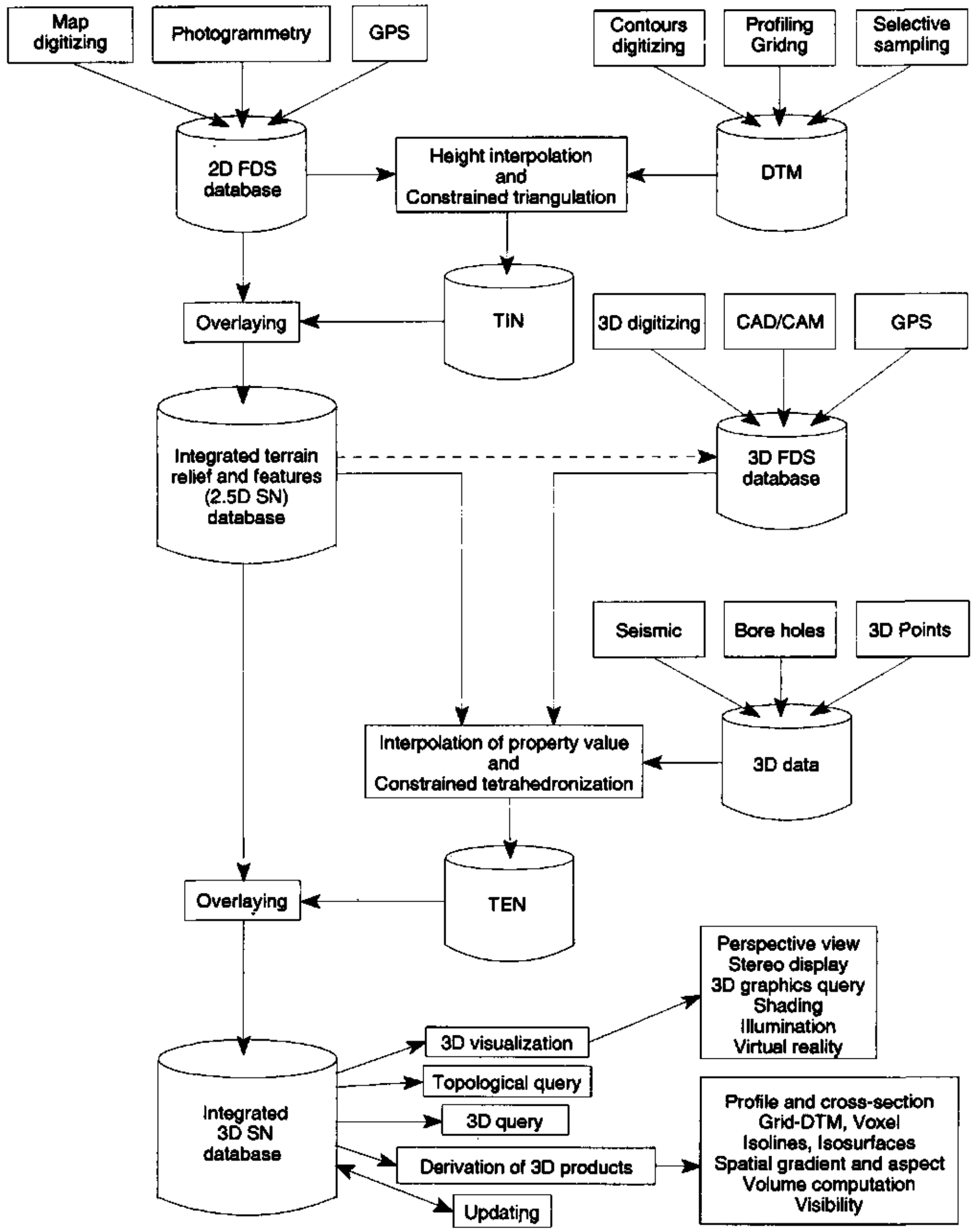


Figure 6.1 Steps in constructing an integrated 3D spatial model.

Each step needs a specific data structuring operation to create the necessary relationships between data elements and organize the data into a unified structure. With respect to the scope of this thesis, only the data structuring for a simplicial network and 3D FDS are explained here. The explanation for the latter is important since 3D features that have to be incorporated into a 3D simplicial network must first be structured according 3D FDS.

6.2 Construction of a Simplicial Network with Constraints

Direct and indirect representation (see chapter 1) are the key issues in the integrated 3D spatial model. Direct representation entails the incorporation and representation of known features without alteration. Indirect representation entails the accurate derivation of object boundaries. The approach to constructing SN is built up from the 2D situation, where a triangulation process is used for the irregular tessellation of space. However, not all triangulations can fulfil the requirement for indirect representation. Among the various possible criteria for triangulation—arbitrary triangulation (connecting pairs of points by straight lines), optimal triangulation (minimum sum of edge lengths), greedy triangulation (shorter edge by local minimum, see Manacher and Zobrist (1979), Lingas (1986)), Delaunay triangulation (empty circumcircle, Delaunay (1934))—Delaunay produces the most compact, equiangular triangulation (Watson and Philip 1984). In general, it offers triangular spatial units that well facilitate interpolation. Each triangle is defined by three natural neighbours; the centre of their circumcircle is the shared vertex of the Thiessen polygons. Figure 6.2a illustrates this explanation as well as the empty circumcircle criterion.

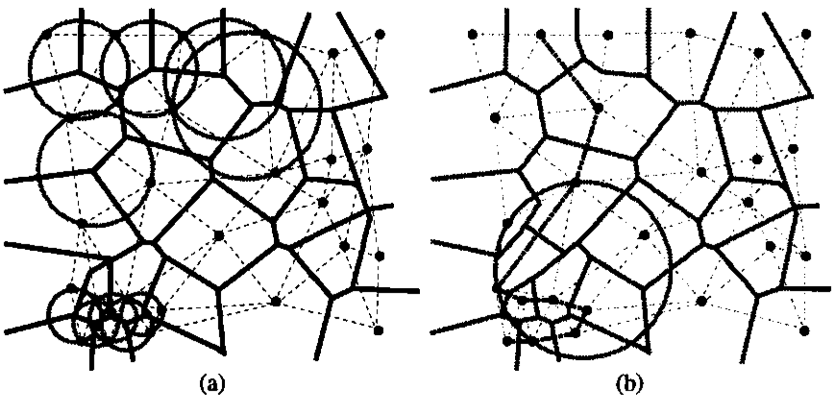


Figure 6.2 Thiessen polygon and Delaunay triangulation, (a) unconstrained and (b) constrained (thick dotted line). In (a), no point is encompassed by the circumcircle of each triangle while in (b) the circumcircle of a triangle with constrained edge may encompass other points.

For the requirement of direct representation, all geometric components of known features must be embedded into the structure of the simplicial network, that is to say, all nodes as triangle vertices and arcs as triangle edges; a contiguous set of triangles forms a surface feature. Nodes are automatically embedded as triangle vertices. The remaining problem is to maintain linear features which are either line features or boundaries of surface features. To achieve the embedding of all geometric components of features, the network formation process must provide a mechanism to strictly preserve the connectivity of each arc whose geometry is represented by the link between its beginning and end nodes. In its original form, Delaunay's concept cannot accommodate this requirement. One therefore has to introduce a constrained network construction, resulting in local relaxation of the Delaunay empty circumcircle criterion in the vicinity of constrained features. Figure 6.2b illustrates this situation.

6.2.1 Vector Approach

Various techniques for Delaunay triangulation have been reported during the past two decades, for example radial sweep (Mirante and Weingarten 1982), divide-and-conquer (Shamos and Hoey 1975, Lewis and Robinson 1978, Lee and Schachter 1980, Dwyer 1987), locally equiangular (Sibson 1978), incremental insertion (Lawson 1977, Lee and Schachter 1980, Bowyer 1981, Watson 1981, Sloan 1987, Macedonio and Pareschi 1991, Tsai and Vonderohe 1991), triangulation growth (Green and Sibson 1978, Brassel and Reif 1979, McCullagh and Ross 1980, Maus 1984). Most attempts have aimed at improvements to speed up Delaunay triangulation. Maus (1984), Sloan (1987), Larkin (1991), Tsai and Vonderohe (1991), Midtbø (1993) have sought improvements to achieve linear-time operation—the time spent for triangulation varies linearly with the number of points involved. An efficient data structure for fast indexing and localization is critical in vector-based triangulation.

Delaunay triangulations can also be generalized to tetrahedronization (Kraak 1992, Kraak and Verbree 1992). A tetrahedral network which complies with the Delaunay condition of 'empty circumspheres' (see Figure 6.3) can be derived from the Voronoi polyhedral tessellation in the same way as Delaunay triangles can be derived from Thiessen polygons. Watson (1981), Avis and Bhattacharya (1983), Edelsbrunner, Preparata and West 1986), Tsai and Vonderohe (1991) and Midtbø (1993) suggest a vector approach to tetrahedronize a set of points. Kanaganathan and Goldstein (1991) report a comparative study of different algorithms. Field (1986), Field and Smith (1991), Joe (1989), Lawson (1985) report studies of the properties and other related issues of tetrahedral networks. Most methods in the vector approach construct tetrahedrons directly rather than derive them from the Voronoi polyhedrons, since that has proven to be simpler. Nevertheless, the difficulty of the vector approach is that some data sorting and sophisticated spatial indexing is required prior to building Delaunay tetrahedrons.

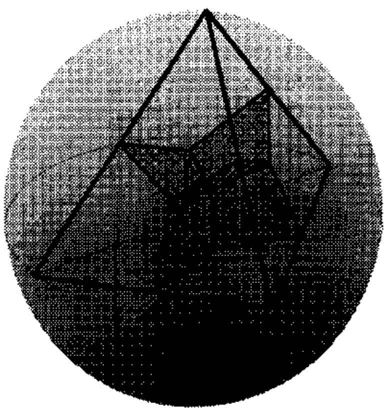


Figure 6.3 Empty circumsphere criterion for formation of Delaunay tetrahedral network. The shaded polygons are part of Voronoi polyhedrons--geometric dual of tetrahedral network.

The embedding of features within the simplicial network implies significant complications. Hardly any solutions for the fast embedding of constraints, for example close to linear-time, have been reported. Most vector approaches refine the simplicial network to include the constraints after the completion of the strict Delaunay network formation and obviously significant extra computing time is indicated. Examples of such attempts found in constrained triangulation are given by Lee and Lin (1986), Bernal (1988), Chew (1989), De Floriani and Puppo (1988), Tsai (1993).

A 2D constrained network construction using the vector approach can be summarized as follows:

1) creation of a regular-grid index structure for the input, *ie* a set of nodes including the nodes of the constrained edges. This process can be viewed as rasterization with very coarse resolution, where each grid element contains a subset of all input nodes
2) construction of a convex-hull
3) initial Delaunay triangulation of the convex-hull
4) densification of the network by inserting nodes one by one until the list is finished
5) embedding constrained features by inserting the constrained edges one by one and performing local retriangulation to fix the new edges. The local retriangulation involves:
   - identifying existing edges that are intersected with the new edge
   - deleting intersected edges and triangles having these edges as their components
   - creating new edges and new triangles linking with the constrained edge.

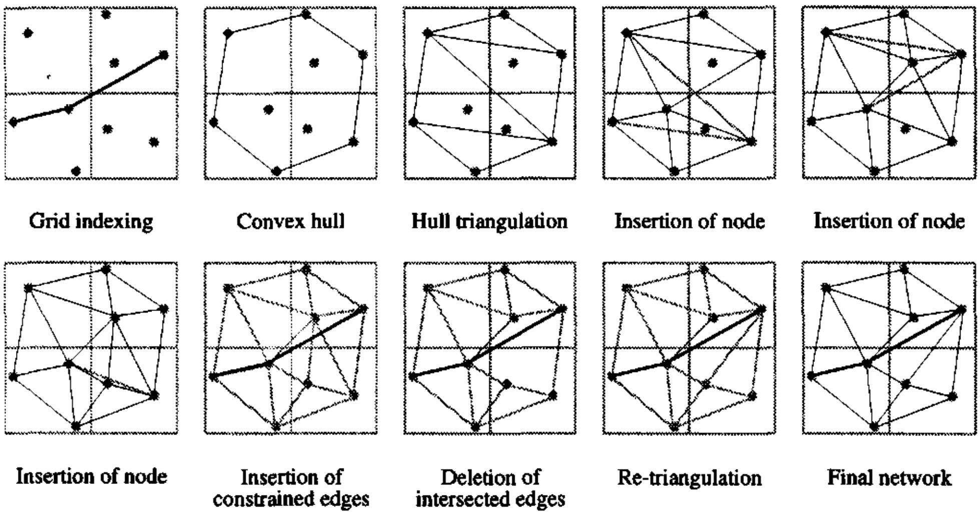


***Figure 6.4** A vector approach for constrained simplicial network formation.*

Figure 6.4 illustrates this process. Tsai (1993), De Floriani and Puppo (1988) explain the algorithm from step 1) to 4) in detail. Note that steps 4) and 5) already imply the operations of insertion and deletion, similar to the updating of the simplicial network in the maintenance phase (see also chapter 7).

The construction of a 3D simplicial network requires constrained tetrahedronization. No algorithm for the vector approach is currently available. Although the approach described above can be intuitively generalized to 3D, investigation of many issues is still needed. For instance, the insertion of a constrained face into a tetrahedral network seems to involve many steps. The face also needs first to be triangulated. Triangles that are components of the face have to be inserted one by one into the tetrahedral network. Further identification of intersected triangles with the constrained face has to be defined prior to local re-tetrahedronization. Complicated computational geometry seems to be required for this operation. Furthermore, the implementation of a vector approach needs careful design of the spatial index structure to speed up the operation. This could be complicated and is beyond the scope of this thesis. Oosterom (1990) and Samet (1990) provide good documentation about the various spatial index structures.

6.2.2 Raster Approach

This section suggests a relatively simple alternative to the vector approach, using a raster approach which can cover both unconstrained and constrained network constructions. The advantage of the raster approach over the vector approach is twofold:

1) it does not require any extra index structure, since the raster is itself a good index structure
2) complicated computational geometry using floating point operations can be avoided, so that the speed of network formation for both unconstrained and constrained is significantly higher.

Another superiority over the vector approach is that the constraints can be added prior to the formation of the network, so that less computing time is required. Since this approach makes extensive use of Boolean operations, its simplicity also allows for fast implementation. Moreover, it is simpler to generalize for 3D and higher dimensions. The obvious disadvantage of the raster approach is its resolution dependence, that is to say, the large storage and memory consumption at very fine resolution.

Pilouk (1992), Pilouk and Tempfli (1992) have reported the raster approach for unconstrained Delaunay triangulation. Tang (1992) reports similar possibilities, including constrained triangulation. For tetrahedronization, the few raster-based algorithms reported are the octree-based (Shephard and Schroeder 1990), mathematical morphology (Cheng et al 1994), and 3D Distance transformation (Pilouk et al 1994). The last two also present a method of tetrahedronization with constraints in which tetrahedrons are derived from Voronoi polyhedrons.

The raster approach to the formation of a simplicial network described in this thesis is based on the influence zones of the data points, that is to say, the Voronoi tessellation obtained by a process known as Distance Transformation (DT). The DT in the raster domain requires the organization of the input in a raster form. Although the operation is based on a raster, the output of the network formation is obtained directly in a vector format. Because of the resolution dependency, special care must be taken during the input phase when converting vector data into raster format. This issue is further discussed in section 6.2.2.3.

Construction of an integrated 3D spatial model based on the simplicial network by raster approach involves the following steps:

- rasterization of nodes
- incorporation of constraints
- Voronoi tessellation
- simplicial network formation
- composition of features.

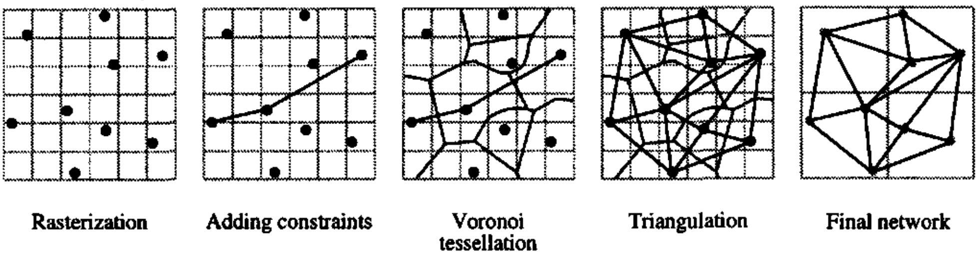


Figure 6.5 Overview of raster approach for constrained Delaunay triangulation.

An overview of the raster approach for 2D simplicial network formation with constraints is shown in Figure 6.5. Fewer operational steps are needed in comparison with the vector approach (see Figure 6.4). The rest of this section elaborates each step in more detail.

6.2.2.1 Overview of Distance Transformation

DT is, in general terms, a process to approximate Euclidean distance from any location in space to the nearest reference location. In a raster domain, each array element represents a location in space. These elements corresponding to given points (nodes) are regarded as references and called kernel elements. There are many different methods for DT including City block, Chessboard, Octagonal, Chamfer, Square-Euclidean. Square-Euclidean (Danielsson 1980) offers the closest approximation to true Euclidean distances; however, the method has the following major problems:

1) the speed of computation is rather low as a result of the many computational steps
2) the quick increment of the distance value by squares requires many bytes to hold the value of square distance
3) large memory and storage are required for the distance image.

The method that best optimizes speed, memory and accuracy is Chamfer distances, proposed by Borgefors (1986). The 2D and 3D variants are Chamfer 3-4 and Chamfer 3-4-5, respectively (see Figure 6.6).

Prior to performing the DT in a finite state computing machine, the input array must be initialized as follows:

- value of kernel elements zero
- value of all other elements the same, close to the highest integer value; for example FFFF in hexadecimal.

The reasons underlying this initialization can be interpreted such that the distance at the reference locations (that is, the kernel elements) is zero and the distances of all other elements are to be determined by the Chamfer distance approach.

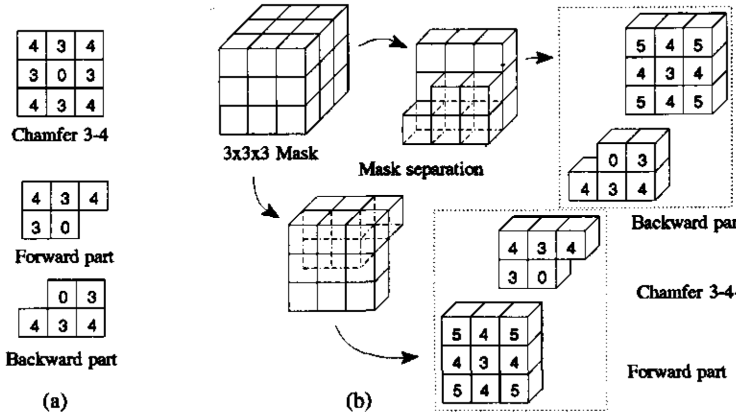


Figure 6.6 (a) Chamfer 3-4 and (b) Chamfer 3-4-5 masks for 2D and 3D distance transformation.

The DT based on the Chamfer distance is carried out by selecting the minimum distance candidate and discrete distance accumulation, starting from a distance value of zero at the kernel. The distance value at the current location is the result of adding the distance value of the previous location that is closer to the kernel, with the selected distance candidate. The candidates are selected by the array elements that coincide with the elements of the Chamfer mask. This mask is a small symmetric array of selected dimensions, 3x3 for 2D and 3x3x3 for 3D. Each element of the mask contains a specific distance approximation value which is added to the value of the element of the input array that coincides with this element of the mask.

The result of this addition is a distance candidate. For 2D DT there are 9 candidates, whereas for 3D DT there are 27 candidates. The approximation of distance for each element of the mask must be done so that it is a good approximation of the true Euclidean distance. Various approaches to the approximation of the true Euclidean distance are given in Borgefors (1986). The name of a Chamfer mask indicates the way the distance approximation is made. For Chamfer 3-4, two approximate values are distinguished; 3 for those connected to the central element of the mask (which gets the value zero) by an edge, and 4 for those connected by vertices. For Chamfer 3-4-5, three approximate values are distinguished; 3 for those connected to the central element by faces, 4 for those connected by edges, and 5 for those connecting by vertices.

In practice, the operation is carried out by scanning the input array in two passes. The mask is split into two parts (Figure 6.6), one for the forward and the other for the backward pass. In the forward scan, the mask is moved systematically through the entire input array. For each location of the mask, the minimum distance is selected from the distance candidates and assigned to the array element coinciding with the central element of the mask. The backward scan follows the reverse route using the second part of the Chamfer mask. For each pass, the splitting of the mask yields 5 distance candidates for 2D and 14 distance candidates for 3D. Since the operation can be integer-based, it is expected to be fast.

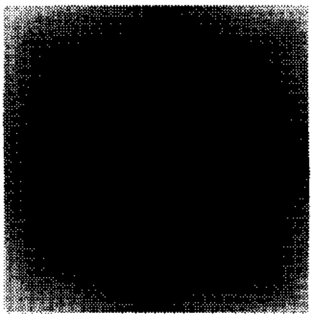


Figure 6.7 A distance image of one kernel element.

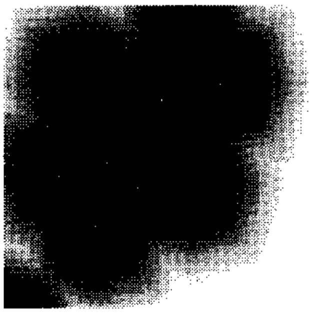


Figure 6.8 Distance image with no edge constrained.

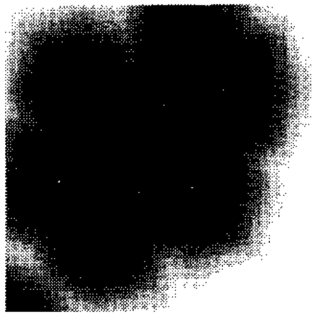


Figure 6.9 Distance image with constrained edges.

Figure 6.7 shows the result of DT by using only one kernel element located at the middle of the image. The variation of the brightness of the pixels implies different distance values. The brighter pixel value indicates a larger distance from the kernel element. Notice that the darkest pixel value (that is, zero) is the kernel element. Figure 6.8 shows the results of DT using 9 kernel elements, whereas Figure 6.9 has, in addition, kernel elements that are part of two constrained edges (see also Figure 6.5) shown as dark strips in the image. We can notice from the last two images that there is a region encompassing each kernel element. Two adjacent regions meet at the frontier located midway between their kernel elements. This kind of frontier is marked by the bright pixels. Each region is known as a Voronoi region, or Thiessen polygon, which indicates the influence zone of each kernel element.

6.2.2.2 Voronoi Tessellation

Exploiting the Voronoi regions implicit in the distance image requires the representation of these regions to be made explicit. This is achieved by a parallel processing as shown in Figure 6.10. This process simultaneously performs DT and Voronoi tessellation (VT). The outputs obtained are the distance image and Voronoi image. The distance image holds the distance values, whereas the Voronoi image holds the identifiers of Voronoi regions. The array elements of a Voronoi region have the same value.

The VT is a process to determine the influence zone of a kernel K such that each element in the influence zone of K is closer to K than to any other kernel. The examples of K in Figure 6.10 are the elements indicated as 01, 02 and 03. The initialization of VT is in fact a copy of the original image. The parallel process needs both initialized distance and Voronoi images. The Chamfer mask is used for the DT. As the mask is moved (in the forward and backward scan) the Voronoi array is built up, element by element, by keeping track of the kernel that influences the selected distance candidate. The identifier of this kernel is recorded to the element of the Voronoi array that coincides with the central element of the Chamfer mask in DT. In this way it is possible to determine the distance at each array element which has been accumulated from which kernel. The process is illustrated in Figure 6.10 where the three adjacent Voronoi regions, obtained in the final stage of the process, create a triangular relationship, which is the principle of the raster approach to Delaunay triangulation.

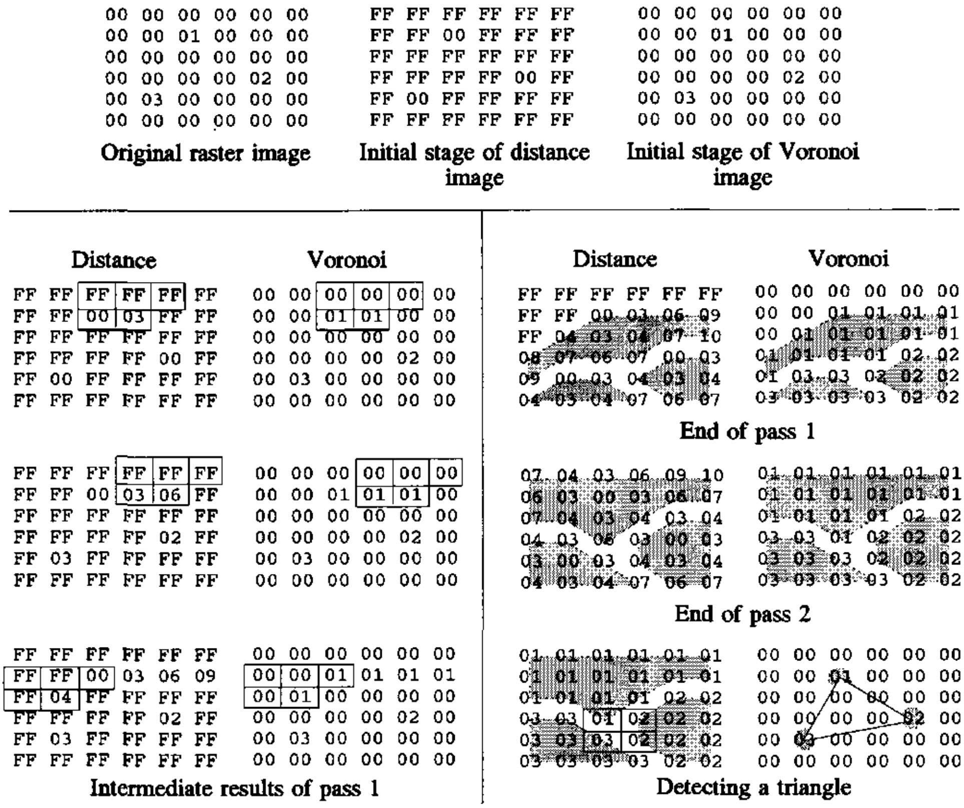


Figure 6.10 Parallel processing of distance transformation and Voronoi tessellation.

6.2.2.3 Rasterizing the Set of Nodes

The purpose of this process is to arrange a set of nodes as kernel elements in the input array to DT to facilitate further processing for simplicial network formation. The rasterization is also known as vector-to-raster conversion, and is achieved by means of the conformal coordinate transformation. This involves translating, scaling, and rotation using one of the transformation formulae in Appendix C. The dimension of the array accommodating the set of nodes (shown in the node table in Figure 6.12) is based on the highest dimension of space taken for the representation of reality. For the formation of the simplicial network at a later stage, a special criterion for rasterization must be defined so that no two nodes are mapped onto the same array element. In other words, mapping from a list of nodes (in the node table) onto a set of array elements and vice versa must have one-to-one correspondence to avoid loss of information. This correspondence can be achieved by satisfying the following two conditions:

1) the list of (input) nodes must not contain:

- for 2D rasterization, duplicated (x, y) coordinates
- for 3D rasterization, duplicated (x, y, z) coordinates
- in general for nD rasterization, duplicated n-coordinate tuples.

2) The appropriate scale factor used must be capable of separating the two closest nodes in the list into two elements of the array.

Given a list of distinct nodes and an appropriate scale factor conforming with these conditions, all nodes can be guaranteed to participate in the further process and be embedded in the simplicial network.

To ensure the first condition is satisfied, the list of nodes must be examined prior to rasterization, to eliminate any duplicated nodes that may have been introduced while digitizing in the data collection process. This error is generally difficult to prevent, unless the digitizing software provides some safeguard mechanism. To meet the second condition, the user may provide a scale factor if the smallest distance is known. The distance D_{min} between the two closest nodes may be previously determined; otherwise, the scale factor S and raster cell-size P can be determined from (see also **Figure 6.11**):

$$S = \sqrt{2} / D_{min}$$

$$P = 1 / S$$

where $S = S_x = S_y = S_z =$ Scale factors.
$P =$ raster cell-size
D_{min} = the smallest distance between the two closest nodes in the list

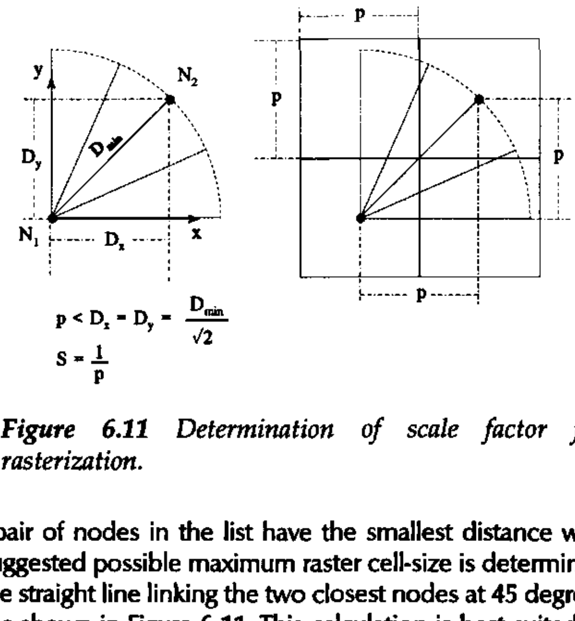


Figure 6.11 *Determination of scale factor for rasterization.*

Note that the value $D_{min}/\sqrt{2}$ is not the only possible cell-size that can separate the two closest nodes in the raster array. It is, however, applicable in any case, particularly when more than one pair of nodes in the list have the smallest distance with possible different directions. The suggested possible maximum raster cell-size is determined using the most critical direction of the straight line linking the two closest nodes at 45 degrees from one of the coordinate axes, as shown in **Figure 6.11**. This calculation is best suited to software implementation.

Having derived the data extent (the bounding rectangle or cube) of the node data set, the array dimension can be defined by multiplying the range of coordinates by the scale factor.

$$Dim_x = R_x \cdot S$$
$$Dim_y = R_y \cdot S$$
$$Dim_z = R_z \cdot S$$

$$R_x = X_{max} - X_{min}$$
$$R_y = Y_{max} - Y_{min}$$
$$R_z = Z_{max} - Z_{min}$$

where Dim_x, Dim_y, Dim_z = dimensions of the array in x, y and z directions respectively,
R_x, R_y, R_z = ranges of x, y and z,
X_{max}, Y_{max}, Z_{max} = the largest x, y and z coordinate values,
X_{min}, Y_{min}, Z_{min} = the smallest x, y and z coordinate values.

In the case where the smallest distance between the two closest nodes in the list is not known, it can be automatically computed. The 'sledge hammer' approach would be to compute the distance between every pair of nodes and obtain D_{min} by means of distance comparison. This would be very time consuming if there were a large number of nodes in the list. A better alternative requiring less computing time is iterative rasterization using a small scale factor (a very large raster cell-size) in the very first iterations. On detecting an element of the array that is being occupied by more than one node, the distance between each pair of nodes sharing the same array element is computed and compared. When the smallest distance has been obtained, the scale factor and array dimensions are computed as shown above to carry out the final rasterization.

6.2.2.4 Algorithms to Incorporate Constraints

The constraints are the known features from data acquisition (for example, fault lines, a ground water level, the ground surface, the drainage network, a railroad). The integrated data models require their geometry to be embodied as part of the simplicial network. The original geometry of these features has to be preserved to prevent loss of spatial knowledge.

Using the raster approach to construct a 2D simplicial network with constraints, the strategy to ensure the connectivity of the constraints is defined at the preparation phase, that is to say, the rasterization. Tang (1992) presented an algorithm that performs the rasterization of a straight line in such a way that the adjacency of its beginning and end nodes is explicitly presented in the input array, thus ensuring the creation of their adjacent Voronoi regions. This is done by propagating the identifiers of the beginning and end nodes of the line, and having them mapped onto the input array, towards each other. The frontiers of this propagation eventually meet at the middle of the line. Two adjacent kernel elements result, one carrying the identifier of the beginning node and the other carrying the identifier of the end node. Note also that each array element resulting from the mapping of this line is also considered as a kernel element, so that a straight line in raster form is in fact a series of kernel elements presented in the input array.

Figure 6.12 illustrates this process. The arc #1 (see Arc table of this figure) has node #4 and #6 as beginning and end node respectively. In the raster image, each pixel in the first half of the rasterized arc receives value 4 while each pixel of the second half receives value 6. Observe that there is adjacency of pixels #4 and #6 at the middle of the arc #1, as for pixels #6 and #7 at the middle of the arc #2.

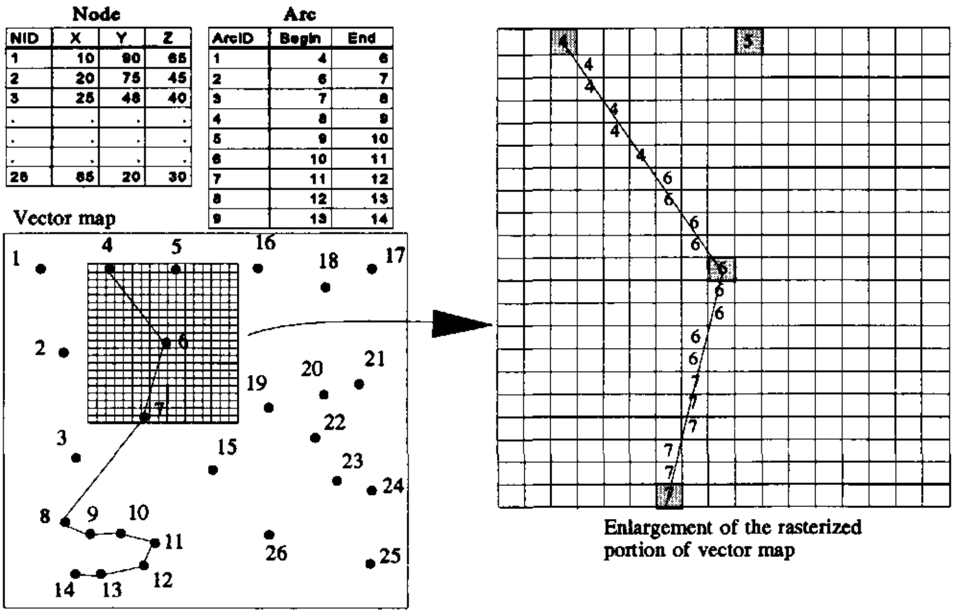


Figure 6.12 Rasterized line features using 'Half-line' algorithm.

Considering the underlying principle of Tang's algorithm, that is to say, the influence zone, we can see that a node of a straight line only extends its influence up to the middle of the line. This characteristic fits the concept of Voronoi tessellation well, so the algorithm can be regarded as a 1D Voronoi tessellation performed on a 1-simplex. With respect to this concept, the constraint edge, that is, a 1-simplex, is tessellated into two 1D Voronoi regions.

Figure 6.13 shows the results of DT and VT of a 2D case where the input contains a set of 4 nodes and a constrained edge. Without the constrained edge, the Voronoi regions and their triangular relationships would be realized as shown in Figure 6.13b. Incorporating the constrained edge by using the 1D Voronoi tessellation, the Voronoi regions and their triangular relationships are adapted to the constraint as shown in Figure 6.13d. We notice also that the two adjacent 2D Voronoi regions encompass the constraint, thus containing two 1D Voronoi regions. The two 1D Voronoi regions coexist with the two 2D Voronoi regions. In other words, the two 1D Voronoi regions of the constraint edge remain unchanged after the DT and VT in 2D. This implies that a 1D Voronoi tessellation has an invariant property under a 2D Voronoi tessellation.

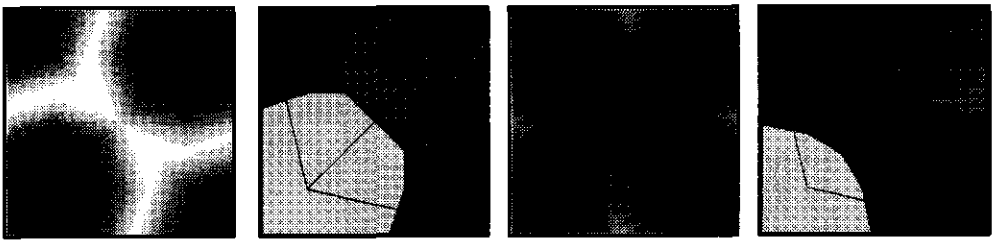


Figure 6.13 *(a) Distance image of four data points with no constraint. (b) Four Voronoi regions corresponding to the distance image in (a). The dark lines show the triangular relationships of three adjacent Voronoi regions. (c) Distance image of four data points with one constrained edge connecting the two points at the upper-left and lower-right corners of the image (the darkest strip running through the two corners). (d) Four Voronoi regions corresponding to the distance image in (b). The thick line shows the adjacency link conforming to the constrained edge.*

We can try to utilize this invariant property for the higher dimension DT and VT. For constrained tetrahedronization, the general aim is to maintain the geometry of every surface feature which has been tessellated into a set of triangles. More specifically, the tetrahedronization must not alter the geometry of any triangle that is part of a surface feature. With this reasoning, we can anticipate the effect of 2D Voronoi tessellation on the 3D DT and VT. By incorporating the 2D Voronoi regions into a 3D array, all elements of the 2D Voronoi regions become the kernel elements of the 3D array. These kernels force 3D VT to create 3D Voronoi regions embracing existing 2D Voronoi regions. The adjacency among the 2D Voronoi regions is preserved and enforces the adjacency of the encompassed 3D Voronoi regions. This also holds when incorporating solely adjacent 1D Voronoi regions into the 3D array. The 3D Voronoi regions embracing adjacent 1D Voronoi regions are also adjacent.

We can conclude:

Theorem 6-1 : n-dimensional Voronoi regions are invariant under m-dimensional distance transformation and Voronoi tessellation where $m \geq n$.

Theorem 6-2 : Given $m > n$, the m-dimensional distance transformation on adjacent n-dimensional Voronoi regions creates m-dimensional Voronoi regions that are also adjacent while maintaining the adjacency among the n-dimensional Voronoi regions.

The above two theorems imply that the incorporation of constrained features must be dimension-wise, that is, it proceeds from a lower to a higher dimension. With respect to the formation of a 3D simplicial network with constraints, the following steps to incorporate constraints can be taken, distinguished by different cases:

a) 1D constraints are part of 2D simplicial network:
- perform 1D VT on constrained arcs
- incorporate the tessellated 1D constraints into the input array for 2D VT

- perform 2D simplicial network formation
- perform 1D VT on every edge, using all edges of triangles as constraints
- incorporate the tessellated edges into the input array for 2D VT
- perform 2D VT again
- let the result from 2D VT serves as 2D constraints for 3D VT.
- perform 3D VT
- perform 3D simplicial network formation.

b) 1D constraints are part of 3D simplicial network but are not part of 2D simplicial network:

- perform 1D VT on constrained arcs,
- incorporate the tessellated 1D constraints into the input array for 3D VT,
- perform 3D VT,
- perform 3D simplicial network formation

Figure 6.17 illustrates this process.

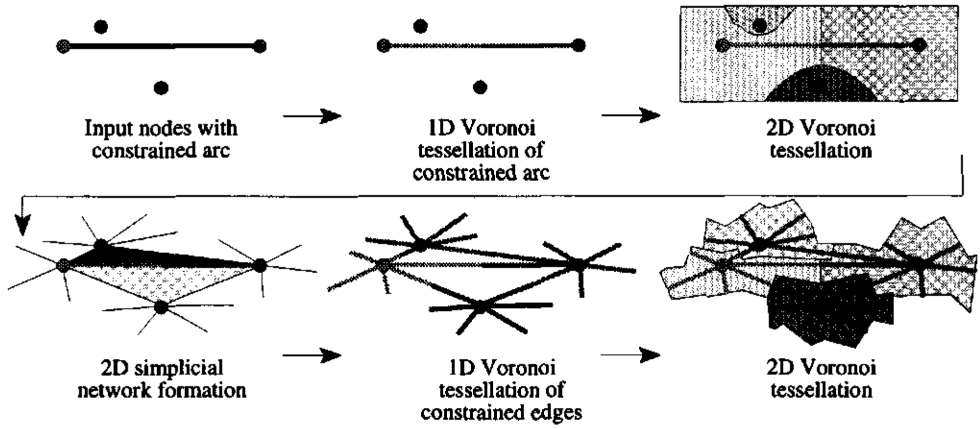


Figure 6.17 Process to incorporate 1D and 2D constraints for simplicial network formation.

6.2.2.5 Algorithms for Irregular Network Formation

For 2D simplicial network formation, scanning the Voronoi image using a 2x2 mask with predefined conditions generates the Delaunay triangulation. A triangle is detected in a situation where at least 3 of the 4 elements of the mask are different. By combinatorial mathematics (Finkbeiner and Lindstrom 1987, Liu 1986), this is a 3-selection (3-combination) from a mask, which is a set of at most 4 distinct elements. The 3-selection is a subset of the mask. We can apply the following definition to determine the number of combinations (triangles in this case):

The number of k-selections from an n-element set is denoted by C(n, k),

$$C(n, k) = \frac{n!}{(n-k)!\,k!}; \quad where\ 0 \le k \le n$$

If the mask contains only 3 distinct pixels, it becomes:

$$C(3,\ 3) = \frac{3!}{(3 - 3)!\ 3!} = 1$$

which yields only 1 triangle.

If all 4 elements of the mask are different, it is:

$$C(4,\ 3) = \frac{4!}{(4 - 3)!\ 3!} = 4$$

which yields 4 triangles to be formed. This leads to 4 intersecting triangles. The situation occurs when 4 points are situated on a circle, hence, 4 Voronoi regions meet at the centre of this circle, as shown in Figure 6.9.

To overcome this problem, only the combinations from the two opposite diagonals of the mask are selected, which yield only two non-intersecting triangles.

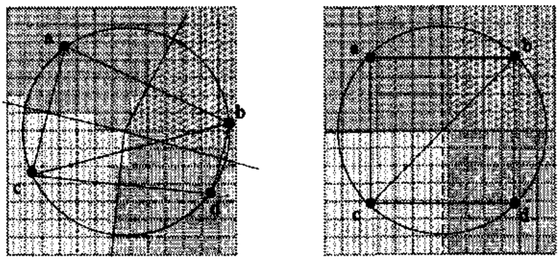


Figure 6.9 Four points situate on a circle causing four Voronoi regions meet at its centre.

The above analysis leads to the conditions i) and ii) for triangulation. Let a, b, c and d be the contents of the 2x2 mask at an instance:

Condition i) The elements in the upper triangle of the mask must be different (elements number 1, 2, 3 in Figure 6.10). This allows 4 possibilities:

a b c c	a b c b	a b c d	a b c a

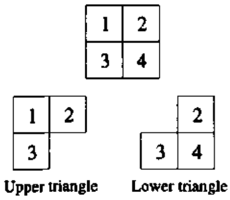


Figure 6.10 2x2 mask elements numbering.

or

Condition ii) The elements in the lower triangle of the mask must be different (elements numbered 2, 3, 4 in Figure 6.10). This allows 4 possibilities

a a b c	a b a c	a b c d	a b c a

Condition iii) The 2 elements on the perpendicular diagonal must be different (elements numbered 1, 4 in Figure 6.10). This condition prevents a faulty formation of a triangle caused by a sliver raster Thiessen polygon. A sliver polygon may be caused by a situation as shown in Figure 6.11.

The consequence of this condition is the rejection of the following situation:

a b
c a

The union of the permissible situations in condition i) and ii) yields 6 distinct possibilities. One of these is rejected by condition iii), so that only 5 possibilities remain.

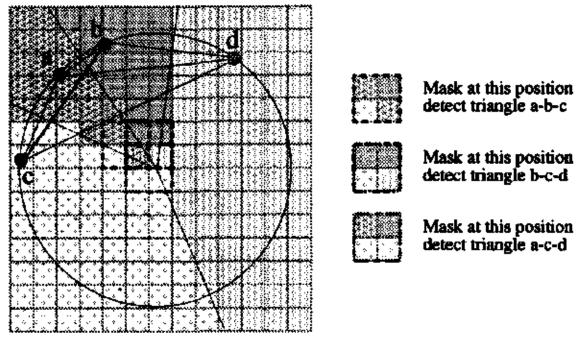


***Figure 6.11** Faulty triangulation caused by sliver Voronoi region.*

a b	a b	a a	a b	a b
c c	c b	b c	a c	c d

For 3D, the formation of a 3D simplicial network is done by using the 2x2x2 mask to scan the 3D Voronoi array once. A tetrahedron is detected when the contents of 4 elements of the mask from the total of 8 are different. By combinatorial mathematics, this is a 4-selection from a set of 8 distinct voxels, that is:

$$C(8, 4) = \frac{8!}{(8 - 4)!\ 4!} = 70$$

This means that there are 70 possible intersecting tetrahedrons.

To ensure proper formation of the 3D simplicial network, that is, the network of tetrahedrons, a Boolean approach is used to set up a set of conditions to form tetrahedrons. The general aim is to allow the creation of at most 6 non-intersecting tetrahedrons at one instance if all 8 elements of the mask are different (see Figure 6.13).

To achieve this requirement, six primary conditions are attached to the 2x2x2 mask for the formation of a unique set of non-overlapping tetrahedrons. Note that the numbers encompassed by circles in Figure 6.13 correspond to the following conditions (1) to (6). The numbering system of the mask is shown by the numbers encompassed by small cubes.

(1) $1 \neq 3 \neq 4 \neq 5$ (2) $1 \neq 2 \neq 4 \neq 5$ (3) $3 \neq 4 \neq 5 \neq 7$
(4) $2 \neq 4 \neq 5 \neq 6$ (5) $4 \neq 5 \neq 6 \neq 8$ (6) $4 \neq 5 \neq 7 \neq 8$

The line drawn between two elements of the mask shows a possible edge of a tetrahedron to be formed by this mask. While scanning the 3D Voronoi array, the mask looks for the boundaries of the 4 adjacent Voronoi polyhedrons. On detecting this situation, a tetrahedron is formed if one of the above 6 conditions is fulfilled.

Nevertheless, the above 6 conditions are not sufficient to prevent the erroneous formation in the case where more than 4 nodes are situated on the surface of a sphere. This situation is comparable to the 2D case shown in Figure 6.9 and Figure 6.11. Additional conditions are therefore added to prevent overlapping or intersecting tetrahedrons. Conditions (a) to (i) listed below correspond to the letters placed on the edges of tetrahedrons in Figure 6.13.

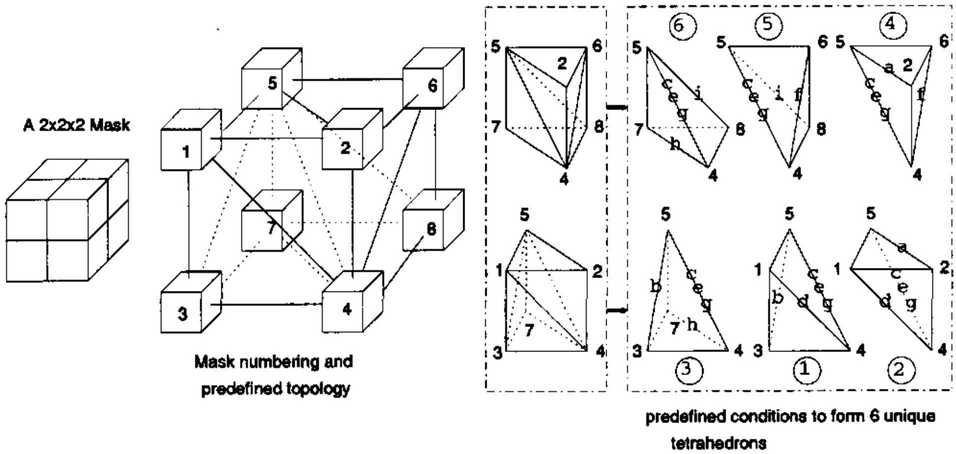


Figure 6.13 *A 2x2x2 conditional mask for forming a tetrahedral network.*

(a) 1 ≠ 6,	*(b) 1 ≠ 7,*	*(c) 1 ≠ 8,*
(d) 2 ≠ 3,	*(e) 2 ≠ 7,*	*(f) 2 ≠ 8,*
(g) 3 ≠ 6,	*(h) 3 ≠ 8,*	*(i) 6 ≠ 7.*

Apart from the conditions above, another three must be added. These three conditions are needed to prevent the formation of a tetrahedron because of raster peculiarities. Similar to the broken appearance of an inclined line in a 2D raster image, in 3D an inclined plane appears as a staircase. This causes a problem when four points are situated on an inclined plane (relative to the 2x2x2 mask) and by chance on the circumference of a circle. Then, four adjacent Voronoi polyhedrons would be detected. Without the three conditions given below, a flat tetrahedron would be formed. This problem does not occur if there are completely horizontal or vertical planes (relative to the mask), since the previous conditions take care of such a constellation:

(j) not (1 ≠ 5 and (1 = 4) and (5 = 8)),
(k) not (3 ≠ 4 and (3 = 5) and (4 = 6)),
(l) not (2 ≠ 4 and (2 = 5) and (4 = 7)).

Combining the first, the second and the third sets of conditions leads to the following algorithm:

if *(1) and (b) and (c) and (d) and (e) and (g) and (l)*
or *(2) and (a) and (c) and (d) and (e) and (g) and (k)*
or *(3) and (b) and (c) and (e) and (g) and (h) and (j)*

or *(4) and (a) and (c) and (e) and (f) and (g) and (j)*
or *(5) and (c) and (e) and (f) and (g) and (i) and (l)*
or *(6) and (c) and (e) and (g) and (h) and (i) and (k)*
then
increase number of tetrahedrons (for memory allocation)
form a tetrahedron.

Since there are several alternatives in designing this mask (for example, a cube can be decomposed into five or six tetrahedrons), the mask in Figure 6.13 takes into account the compatibility with 2D triangulation. This provides an easy way of combining a TIN with TEN without any conflict. The design of the 2x2x2 mask is based on the principle that a cube can be cut by three different planes, each plane passing through two diagonally opposite edges of the cube, taking a pair of edges for every coordinate axis. The three planes intersect each other along a diagonal of the cube and divide the cube into six tetrahedrons.

The topology of the resulting simplicial network is documented by tables as shown in chapter 5.

6.2.2.6 Composition of Features

To complete the construction of the 3D spatial model, the topological relationships between the simplices and the complexes must be established by classifying (assigning) each simplex as part of the complex it constitutes. This can be achieved by performing an overlay process between the data sets containing features and the simplicial network respectively. As a result of the convex property of simplices, the centroid of each simplex can be used for the containment test against the complex. This significantly simplifies the overlaying process. Note that the overlaying process requires both data sets to be correctly structured in advance. The most favourable data structure for the features is that derived from the variants of FDS (see chapter 2), because of the compatibility.

6.3 Data Structuring for 3D FDS

The construction of a 3D spatial model based on a simplicial network requires the incorporation of 3D features as constraints (see Figure 6.1). The constraints are preferably structured in the database according to 3D FDS. This database provides the geometric components of the constraints involved in the network construction and the thematic components for the composition of features by the overlaying process. A relational data structure derived from 3D FDS has been presented by Rikkers et al (1993), Bric (1993), Bric et al (1994), as shown in Figure 3.11.

In Bric (1993) the capability of 3D FDS has been explored by building TREVIS an experimental 3D GIS (see also chapter 3). Various queries about topological relationships between features in 3D space, for example neighbourhood, adjacency and inclusion, have been tested with satisfactory results. The data sets used for the experiment were, however, structured manually. The complexity of constructing a database of 3D features was realized and simplification of the process by capturing necessary spatial relationships at the data acquisition phase was proposed.

Since aerial photographs were to be used as the data source, strict photogrammetric digitizing procedures were suggested. Further investigation into the design of photogrammetric digitizing procedures has been carried out by Wang (1994), placing the main focus on urban scale application. Database creation involves the reconstruction of 3D features representing buildings, houses, and surface features, for example roads, land parcels, and terrain relief. Each digitized feature was assigned a specific code indicating for which data the structuring strategy was suitable, as shown in Table 6.1.

Table 6.1 Examples of output format from photogrammetric data collection process

Type	Code	Description/Example	Purpose
Body feature	B1	Roof outline	Construct the body by plane sweep vertically to intersect with DTM
Surface feature	S1	Roof facet boundaries (ridge and drainage)	Replace the roof outline after obtaining the body
Surface feature	S2	Land parcel	Make part of terrain surface
Line feature	L1	Road, railway	Make part of terrain surface
Point feature	P1	Location of a tree, lamppost	Make part of terrain surface

Note that only a limited number of codes were made available. Two major groups of codes can be distinguished, the first contributing to the representation of buildings and the second contributing to the representation of terrain surfaces. Since the aerial photographs are limited to near vertical view, roofs of buildings and houses, 3D objects of interest can be captured quite easily. The knowledge and experience of the operator carrying out the digitizing are important for the correct interpretation of the situation shown in the aerial photograph.

The data structuring can now proceed as shown in Figure 6.14. All possible topological relationships are also recorded in parallel with the reconstruction of the geometry using the data structure shown in Figure 3.11.

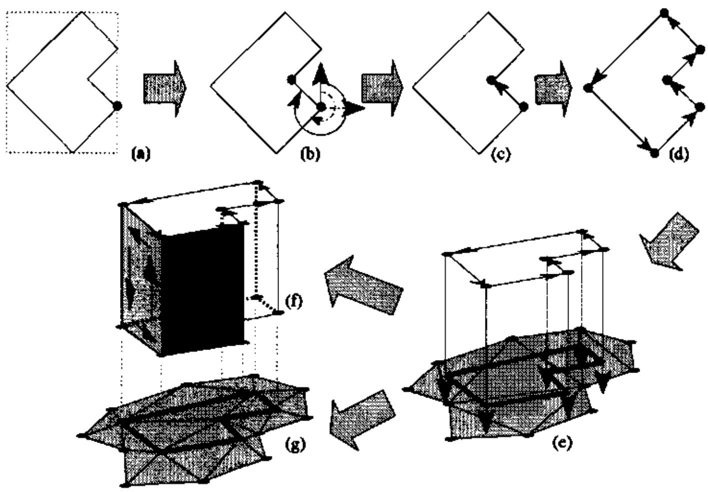


Figure 6.14 Steps for data structuring for 3D FDS.

Representation of the terrain surface and outline of the roofs are used to construct the 3D features representing the buildings and houses. The footprints obtained are incorporated in the 2.5D simplicial network, that is, a TIN which represents the terrain surface and 2D features shown in Figure 6.14 (e) and (g), respectively. In this way, the topological relationships between terrain surface and 3D features can be established and maintained within the 3D FDS database. In addition to the construction of geometry in Wang (1994), the process shown in Figure 6.14 also includes face orientation, fulfilling the 3D visualization requirement for the normal vector of each visible face of a body feature to point towards the outside of the body.

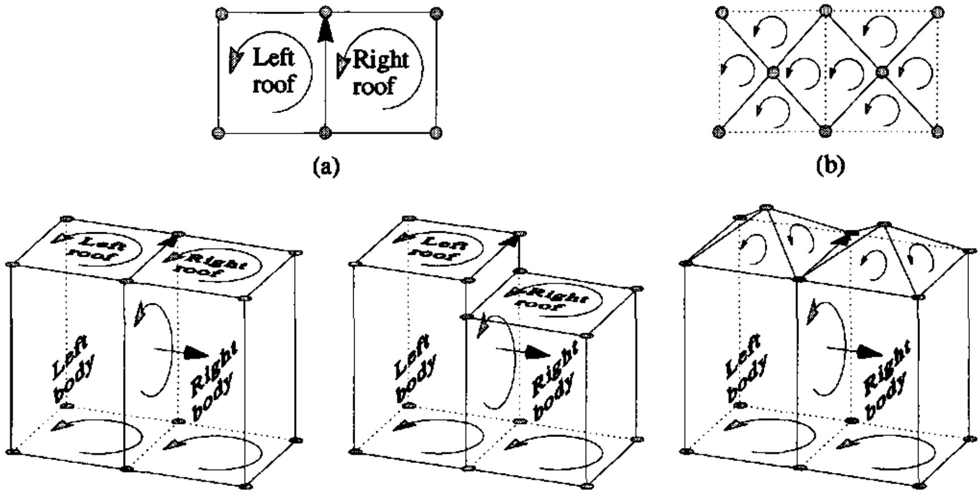


Figure 6.15 (a) Outline of two adjacent roofs, (b) details of the roof, (c) two adjacent flat roofs with the same elevation, (d) similar to (c) but different elevation, (e) replacing the outline of the roofs by their facets after reconstructing the main geometry.

Note however that not all topological relationships as shown in Figure 3.9 (for example a node in a body) can be established by this data structuring through the lack of information during data collection. Some adjacency may be capable of being directly inherited from 2D topology (see Figure 6.15 (a) and (c)). Nevertheless, the relationships may not always be straightforward, for example as shown in Figure 6.15 (d). This implies that further investigation of this issue is still needed.

6.4 Consistency Checking

During the data structuring process described in section 6.1, it is necessary to ensure data consistency at various stages. The different kinds of consistency to be considered are:

1) Attribute-domain; for example length must be numeric
2) Range consistency; for example, a date must lie within 1 and 31 for January
3) Valid links; for example a node of the arc must exist in the node table
4) Cardinality; for example an arc has two nodes, this means both of them should not be nil

5) Internal consistency between the metric and topological representation; for example if a node is topologically represented to be in a body feature, the metric computation must also ensure that the node is in the body feature
6) Semantic consistency which considers relationships between objects at the application level; for example a road cannot run coincidentally with a river. However, this level is very much subject to human knowledge and experience. (A commercial system that claims to be capable of checking this is System 9.)
7) Indexing consistency; in case the index structure is used to access spatial data, the index must point to the correct data element. In general, the index must be updated after and according to the change of the core database.

Laurini and Thompson (1992) suggest two stages for the consistency checking for a spatial database:

1) during database creation, checking may be global and for all objects
2) during changes in the database, for example insertion, deletion and updating, checking may be locally performed for each newly created object, or those affected by the changes.

In this thesis, only the internal consistency of a simplicial network is considered, using the generalized Euler equality given in chapter 4. We recall from chapter 4 the generalized Euler equality for a simplicial network:

$$\sum_{k=0}^{k\leq n} (-1)^k N_k = 1$$

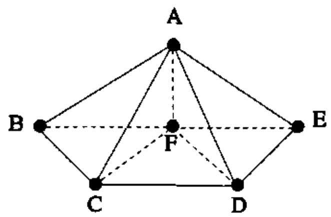
where n = dimension number
N_k = number of k-simplices

For application in 3D GIS, it is more convenient to translate the above equation into:

Nodes - Arcs + Triangles - Tetrahedrons = 1

For example, if the simplicial network consists of 3 tetrahedrons, 8 triangles, 11 arcs and 6 nodes as shown in Figure 6.17, instantiating these values into the above equation yields:

6 - 12 + 10 - 3 = 1



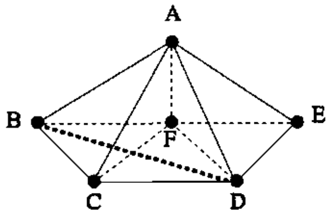
No.	Node	Arc	Triangl	Tetrah
1	A	AB	ABC	ABCF
2	B	AC	ACD	ACDF
3	C	AD	ADE	ADEF
4	D	AE	ABF	
5	E	AF	ACF	
6	F	BC	ADF	
7		BF	AEF	
8		CD	BCF	
9		CF	CDF	
10		DE	DEF	
11		DF		
12		EF		

***Figure 6.17** Example of a simplicial network of tetrahedrons.*

The above equation should be applied immediately after the network formation is finished. Two situations require evaluation:

1) If the equality is not met, the network is in error. This often occurs through nodes being neglected, particularly likely when there are duplicate nodes in the node list; or a chosen raster cell-size is too large, so that the two closest nodes cannot be separated into two raster elements. Another cause may be an extra arc intersecting another arc, which consequently produces intersecting simplicial elements like triangles or tetrahedrons (marked in the table as *) in the network as shown the Figure 6.16.

Counting the number of simplicial elements as shown in Figure 6.16, there are 6 nodes, 13 arcs, 13 triangles and 6 tetrahedrons (note that BCDF is a flat tetrahedron). Applying the generalized Euler equality yields:



No.	Node	Arc	Triangl	Tetrah
1	A	AB	ABC	ABCF
2	B	AC	ACD	ACDF
3	C	AD	ADE	ADEF
4	D	AE	ABF	*ABCD
5	E	AF	ACF	*ABDF
6	F	BC	ADF	*BCDF
7		BF	AEF	
8		CD	BCF	
9		CF	CDF	
10		DE	DEF	
11		DF	*BCD	
12		EF	*BDF	
13		*BD	*ABD	

Figure 6.16 An incorrect form of a simplicial network.

$$6 - 13 + 13 - 6 = 0$$

which indicates the error that is likely to be caused by a faulty implementation of the network formation algorithm.

2) If the equality is met, a further consistency check, for example a test for valid links or intersection, is still needed because, unfortunately, the equality can be spuriously compensated. For instance, if there is a rule defined in the system such that the existence of a flat tetrahedron (for example BCDF) is not allowed, then there will be 5 tetrahedrons instead of 6 and the equality holds despite the fact that the network is in error.

Since this kind of error cannot be detected by using the Euler equality, special attention has to be paid to the design of the algorithm to form the network to prevent this kind of situation. Otherwise, it is advisable to test each tetrahedral candidate to be recorded into the database during the network formation. Testing against neighbours sharing a component with the candidate is sufficient to detect this kind of error.

6.5 Discussion

We have presented methods for data structuring, database creation and internal consistency checking for the building of an integrated 3D database based on IDM and UNS. We have also outlined a procedure for acquiring 3D urban data and structuring them according to 3D FDS, an intermediate step in the construction of an integrated 3D spatial model. For generating a simplicial network with constraints, a vector approach is very cumbersome. We have therefore proposed and developed a raster approach, which simplifies the implementation. Note, however, that this algorithm has not been designed to adapt locally to different densities of nodes. Special care should be taken during the rasterization process where the selection of an appropriate raster cell-size is suggested, to avoid information loss. Although raster processing is simple and fast, it requires a large amount of memory and storage space. This problem is becoming less significant with the rapid development of both computing power and storage capacity.

Since the raster approach forms a simplicial network from Voronoi regions, we have developed an approach that is valid for nD, incorporating constraints into the simplicial network using invariant property of Voronoi regions under Voronoi tessellation. This completes the geo-spatial modelling ranging from the design to the construction phase.

Creating an integrated 3D database requires many steps. When contemplating a large area with highly detailed information, the task may seem impossible. We therefore suggest incremental construction of the integrated database. This is one of the most important aspects the simplicial network structure offers. Also, more detailed information can be contained through network refinement; this can be done locally. Although the proposed data structuring is attained by raster processing, the end result is a vector structure that does not depend on scale, or level of precision. The simplicial network can be refined as necessary, so it can be used to model broad ranges of real world objects.

To certify the integrity of a generated simplicial network, the generalized Euler equality derived in chapter 4 can be used for the internal consistency checking, valid for the geometric aspect of the simplicial network. Further checks may still be necessary. However, more in-depth investigation is needed to cover this aspect for the maintenance of the simplicial network.

7
APPLICATIONS OF THE MODEL

Having designed the integrated data model, unified data structures and introduced methods of construction, demonstrating the 3D spatial model's applicability is the last objective. This is achieved through various steps of spatial data processing, using both simulated and real data. The approach is stepwise, starting from the 2.5D application that integrates terrain relief and terrain features. Examples of spatial query and 3D visualization exploiting an integrated spatial model based on a simplicial network are also presented. For the full 3D application, two kinds of data sets are used as examples. The first uses the data of an urban area consisting of roads and buildings. The second uses a simulated boreholes data. The latter is used to demonstrate the modelling of subsurface objects typically found in geological applications.

7.1 Integration of Terrain Relief and Terrain Features

The combined use of terrain relief and terrain features has been often referred to as 'integration of DTM and GIS' as mentioned in chapter 1. This section intends to demonstrate that the concept of a simplicial network can offer a solution for this problem. The general aim is to create an integrated spatial model representing the earth's surface and 2D representation of spatial objects related to this surface in a 3D space. Simulated data sets are used for this demonstration. The data sets consist of:

- Surface features shown as: land use and soil maps in Figure 7.1 and Figure 7.2, respectively
- Line features shown in Figure 7.1 and Figure 7.2
- Measurements for terrain relief modelling (Figure 7.3).

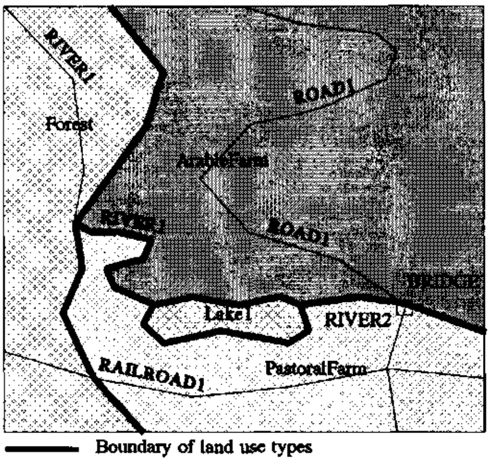


Figure 7.1 Land use map.

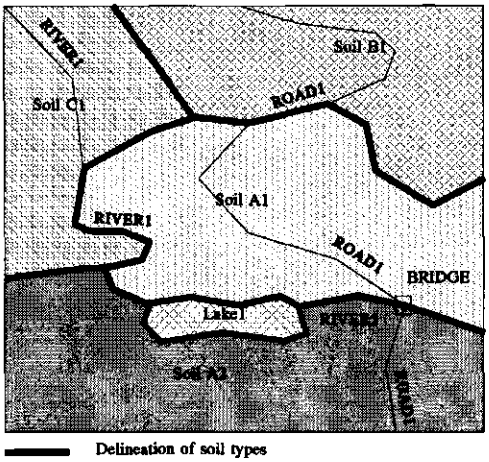


Figure 7.2 Soil map.

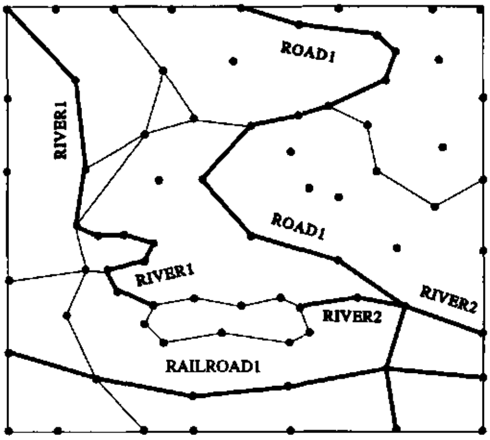


Figure 7.3 DTM points. Line features and boundaries of area features to be used as constraints.

The data sets of land use, soil, line features and relief may be used to answer the questions listed below:

- Which sections of the Road1 may need side slope protection?
- Where is a suitable location for dam construction?
- What is the total length of the railroad that may be damaged by a 1-metre flooding from Lake1 this year?
- How large is the surface area of the soil type A1 between elevation 100 and 200 metres?
- Where are the forest areas with a terrain slope more than 30%?
- How large is the catchment area generating run-off into the River1?

7.1.1 Creating an Integrated Database

The maps shown in Figure 7.1 and Figure 7.2 are in the form of SVVM. They should be combined by overlaying in the first step of the integration. This yields a multi-theme data set shown as a multi-valued vector map (MVVM) in Figure 7.4. The multi-theme data set is used as a basis to create relationships between features and primitives at a later stage. Since there are common features in both land use and soil data sets, that is, roads, rivers, railroads and lakes, redundancy and uncertainty problems arise. For example, the digitizing of Road1 in land use and soil data sets may be carried out separately. Overlaying these two data sets may introduce slivers (small polygons) along Road1. This problem, as shown in Figure 7.5, should be solved before proceeding further.

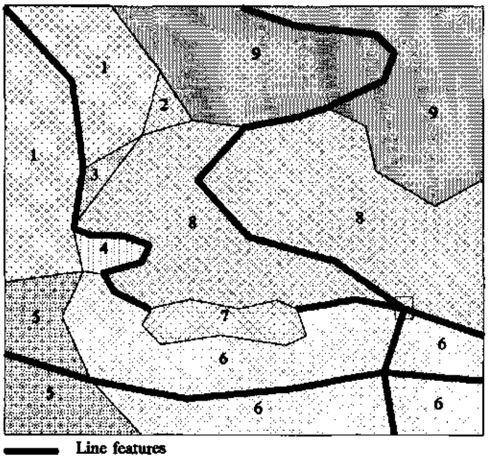


Figure 7.4 Overlaying of land use and soil maps result in a multi-theme map.

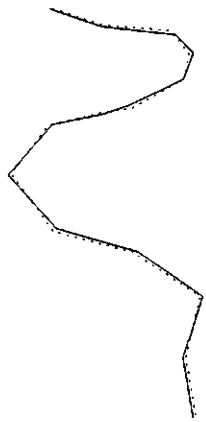


Figure 7.5 Slivers caused by uncertainty in data acquisition may occur after overlaying of different data sets. A solid line and dotted-line show two different sources of data for the same line feature.

Shi (1994) has discussed the handling of this kind of uncertainty. It should be noted that uncertainty for both planimetry and height are likely if all data points have 3D coordinates. However, it has been assumed here that this problem has been solved, allowing us to proceed further.

The next step is to ensure that all digitized points have 3D coordinates. If some of these points have only 2D coordinates, a preliminary Delaunay triangulation may be constructed using only the nodes that have 3D coordinates. Linear features that are already in 3D should be involved as constraints in the triangulation process, to ensure better fidelity of the surface representation. The TIN resulting from the constrained triangulation in this step can be used to introduce the third dimension to all 2D data points by means of interpolation. Each 2D point can be tested with the point-in-triangle algorithm. The plane equation can be computed from the triangle obtained from the positive result of this test. The height information is obtained at the point of intersection between the triangle plane and the vertical line passing through this point.

When all the nodes with 3D coordinates have been obtained, they should be used for constrained triangulation. All line features and boundaries of area features should be used as constraints, as explained in chapter 6. The resulting TIN, shown in Figure 7.6, is then overlayed with the multi theme data set, shown in Figure 7.4. The relationships between triangle components, that is, faces, edges, nodes, and the point, line and surface features can be created after this process. It is worth mentioning that the algorithm used for overlaying TIN with a polygon map can be simplified to point-in-polygon testing. The centroid of each triangle can be used as a point to be tested against a set of polygons.

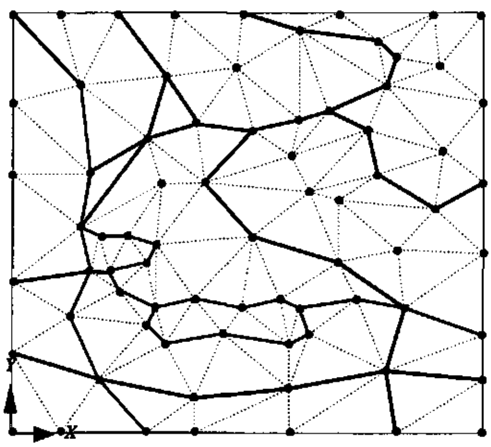


Figure 7.6 TIN resulted from constrained triangulation using all digitized points. Line features and boundaries of area features are used as constraints.

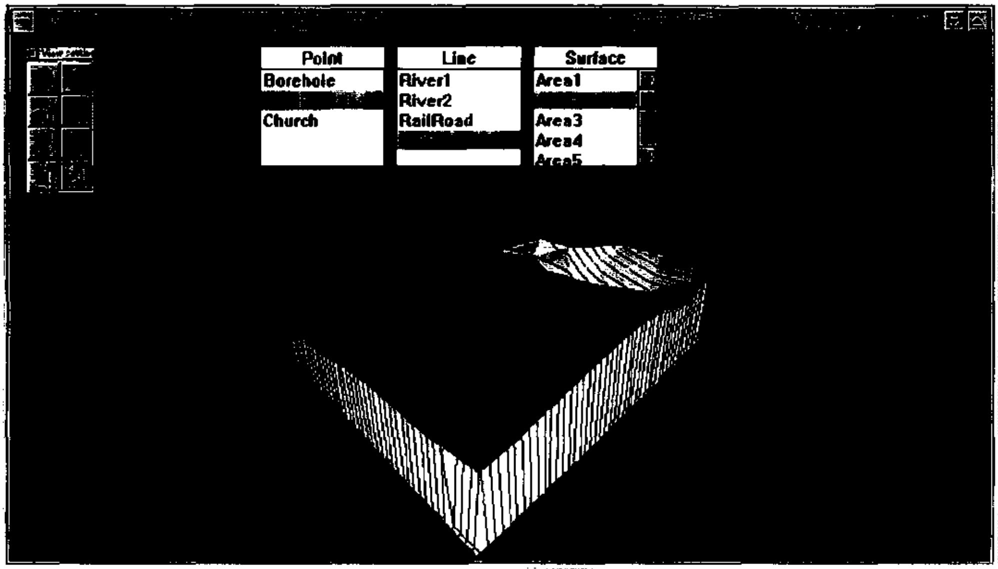


Figure 7.7 The constrained TIN presented in a perspective view shows aspect of relief of this data set. The query results can be directly presented in this view. Contour lines can also be derived directly from this database.

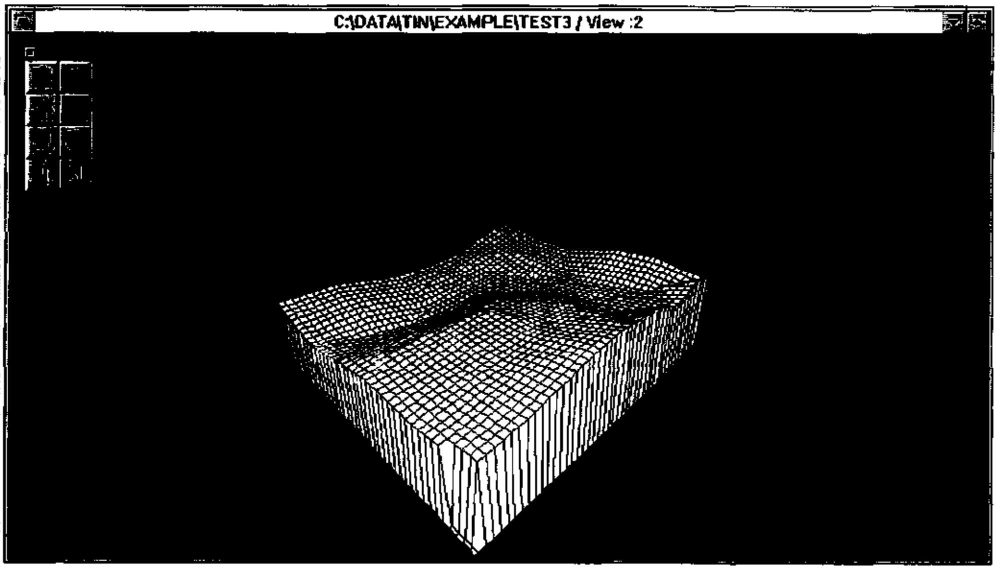


Figure 7.8 Regular-grid DTM derived from simplicial network database.

The database obtained at this stage is called a 'simplicial network integrated database' (SNIDB) for convenience. Figure 7.7 presents the content of the SNIDB in perspective view that creates understanding about relief. Data structured in this way permit direct presentation of query results in this kind of view. Regular-grid DTM can also be derived from this database, as shown in Figure 7.8.

7.1.2 A Spatial Query Example

Having the database in an SNIDB scheme extends the query space typically provided by a 2D GIS or a DTM significantly. Many complex queries requiring many steps when using a typical 2D GIS can be simplified. For example, a road engineer looking for soil material of type A1 to use for road construction might ask, 'What is the volume of soil A1 within the arable farm area and at a depth of 3 metres under the average elevation of this area?' Using the SNIDB, the system just looks for all the triangles that are part of the polygons having feature code = 8, computes the mean elevation using this set of triangle vertices and then computes the summation of volume under each triangle with a depth of 3 metres below this mean elevation. The volume under each triangle can be computed using the formula shown in Figure 7.9. The process is illustrated in Figure 7.10.

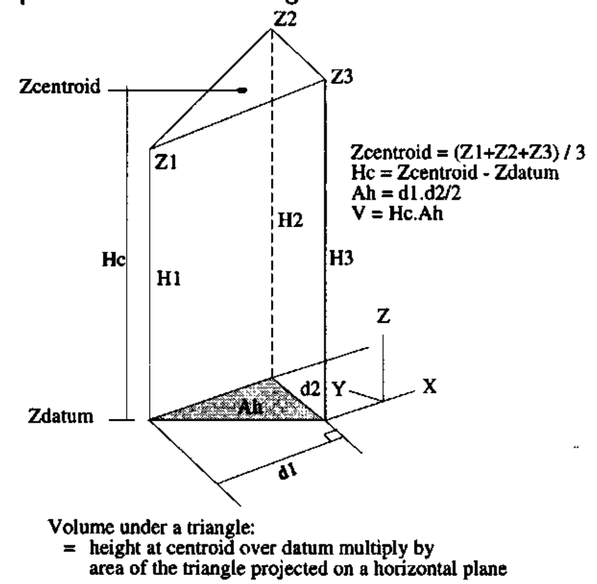


Figure 7.9 Computation of volume above datum and under a triangle.

Without the SNIDB, a typical user of several databases:

(a) overlays the soil database and the land use database to obtain the overlapping area of soil A1 and arable farm area

(b) solves uncertainty, for example in the form of slivers

(c) overlays the results obtained from (b) with the DTM database to obtain the clip DTM within the area of soil A1 and arable farm area

(d) uses the clipped DTM, calculates the mean elevation within the area

(e) calculates the volume.

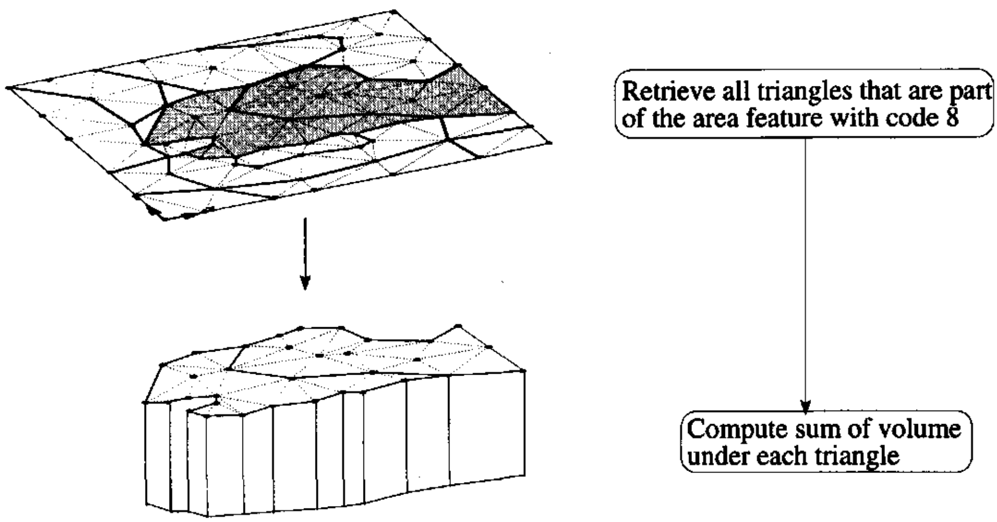


Figure 7.10 Computing summation of volume under and area tessellated into a set of triangles.

The above requires two subsystems, that is, 2D GIS and DTM. Steps (a) to (b) are carried out in 2D GIS whereas steps (c) to (e) have to be done in the DTM subsystem. Two overlaying processes are required, entailing extra time to process the query. The integration into SNIDB takes the responsibility of overlaying processes away from the user. Although constructing the SNIDB might take more time, the user gains the response time during the query process, which seems to be more reasonable, because the query tends to take place more frequently than the database construction. For example, the same kind of question may be asked again for a different area, elevation and date, requiring a repeat of the process from (a) to (e). This repetition is needed for every different set of parameters given.

The overlaying process normally requires computational geometry at the level of geometric primitives. A large amount of operating time may be required, especially for a large data set. The second problem is the solving of uncertainty, which requires knowledge about the source of data and the appropriate solution to be taken. The SNIDB approach performs preoverlaying at the database construction level and therefore turns the overlaying process during a query into a spatial search which can be speeded up by the use of topology. The uncertainty only needs to be solved once and is then converted into data quality that can be stored as an attribute in the database. These features make the GIS more convenient to use.

7.2 Integrating with 3D Features

Should the model need to cover the full three-dimensional representation of buildings and other man made objects as shown in Figure 7.11, the simplicial network can also meet this requirement. The assumption is that the representation of 3D objects is given in the form of 3D FDS. Since the 2D simplicial network is fully compatible with the 3D FDS, the complexity of integration is reduced. As a result of compatibility, a feature belonging to the 2D simplicial network can readily be considered as a feature of 3D FDS. A surface feature (that is, part of the terrain surface) can be related with a 3D feature via their footprints. This can be done by embedding the footprints of 3D features within the simplicial network representing terrain surface by means of constrained triangulation, as shown in the chapter 6. The footprints of the 3D features carry the links to the terrain surface and the 3D features themselves. In this way the 3D topology between the surface and 3D features is established, which permits the integrated use of the two types of data within a 3D FDS database. Figure 7.12 shows an example of terrain data. Figure 7.13 shows the data in the form of a simplicial network that has been constructed by constrained triangulation.

Figure 7.11 Scanned aerial photograph of the study area of the city centre of Enschede.

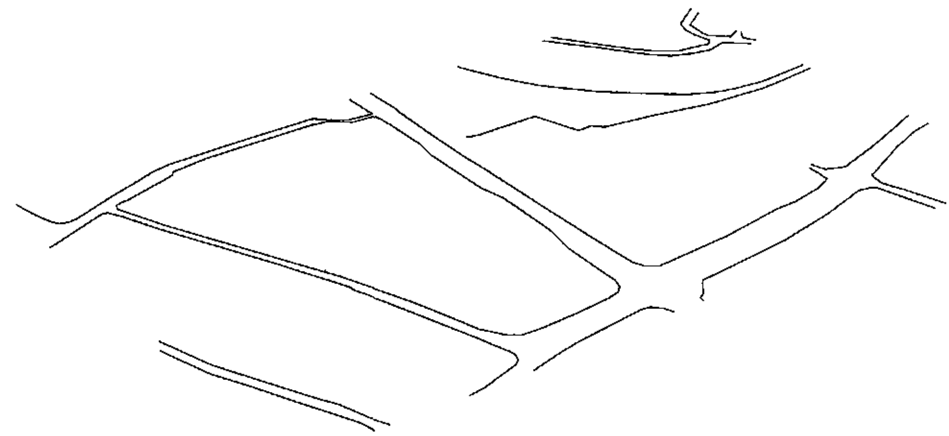


Figure 7.12 2D features on the terrain surface of the study area. The data was photogrammetrically digitized using a Matra T10 Digital Photogrammetric Workstation.

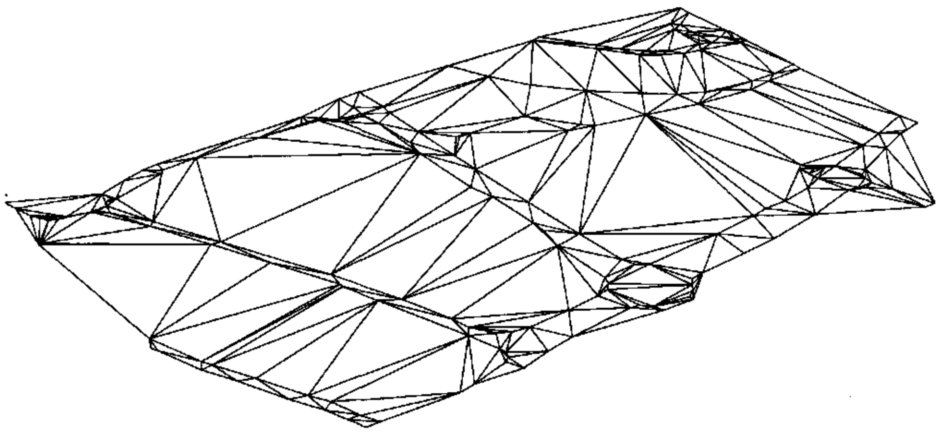


Figure 7.13 The result of triangulation applying 2D features as constraints. The constrained triangulation was carried out using the raster approach implemented in the ISNAP program.

Figure 7.14 shows the data of 3D objects. The data has been digitized by the stereo photogrammetric approach using a Matra T10 digital photogrammetric work station. Only the outlines of the roof of each building were digitized manually. The wall and footprint of each building were obtained automatically by vertically projecting the outline of the roof onto the TIN-DTM, shown in Figure 7.13, using a set of programs developed by the author and Wang (1994). The 3D objects are maintained using the 3D FDS scheme.

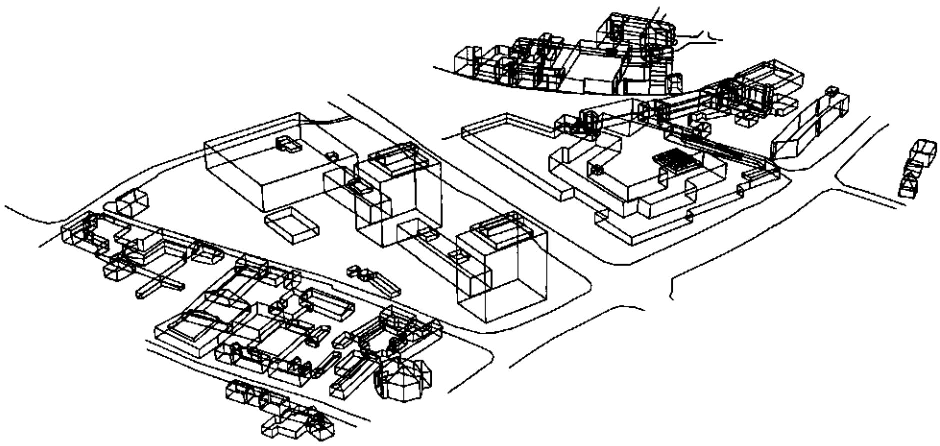


Figure 7.14 Buildings with walls and footprints resulting from vertically projecting the roof outlines onto the TIN-DTM .

The footprint of each building is then retrieved from the 3D FDS database, as shown in Figure 7.15.

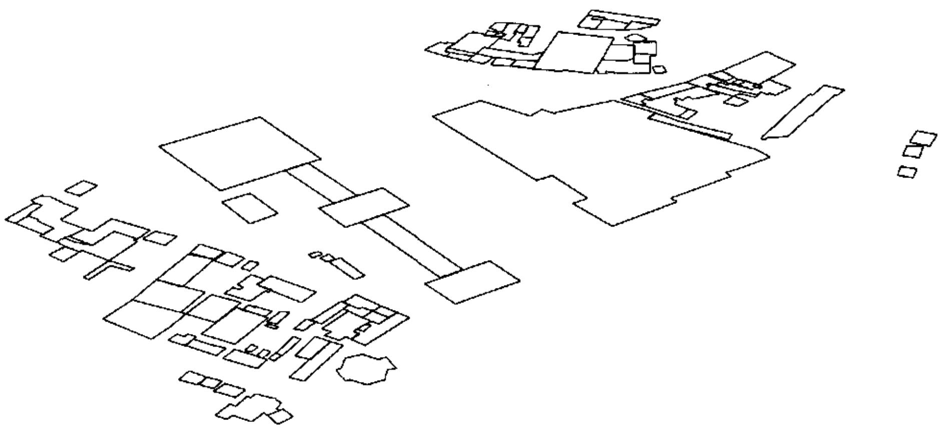


Figure 7.15 All footprints of the buildings extracted from the 3D FDS database.

By mean of constrained triangulation or local updating of a 2.5D simplicial network as shown in appendix E, all footprints of the buildings are embedded onto the terrain surface that is represented by a simplicial network, as shown in Figure 7.16. By means of overlaying, the part-of relationships can be established between triangles and a surface feature representing the footprint of a building and embracing them. All triangles affected by retriangulation are also subjected to updating the part-of relationships with the corresponding surface features.

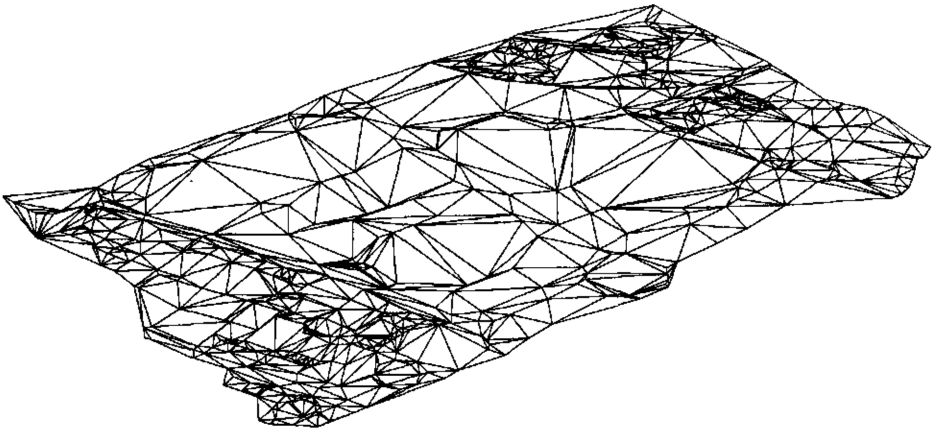


Figure 7.16 A 2D simplicial network representing a terrain surface of part of the central area of Enschede.

Figure 7.17 shows the final result of the integration between the surface and 3D objects.

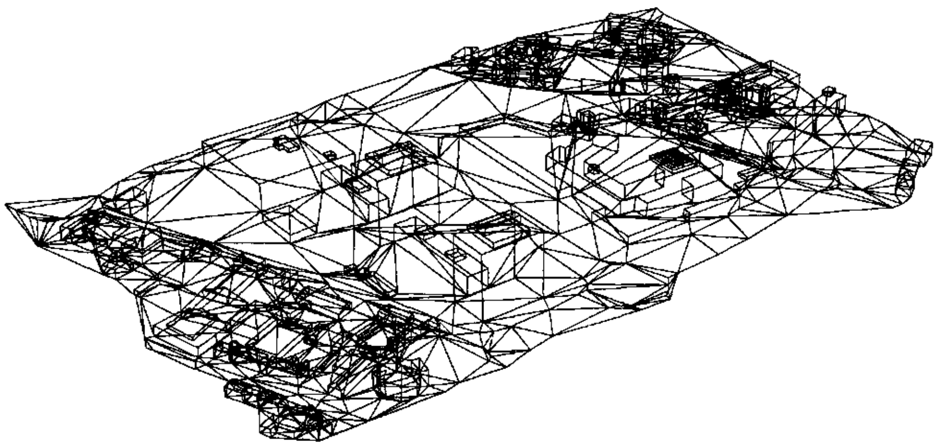


Figure 7.17 Merging of the representations of terrain surface and the 3D objects.

This kind of database may be used to answer such questions as:
- Which buildings are suitable for the placing of antennas for mobile telephones?
- Which buildings are visible from point A located on the top of building B?
- Will the noise from the high-speed trains passing nearby be heard in this city?

If the representation of a terrain surface, in which the footprints of 3D features may be included, is stored separately from the database storing 3D features, the user may have to face several problems before a query can be carried out, for example:

- transformation to a common coordinate system
- solving uncertainty with the footprint of a building in the 3D database if the footprint is on the 2.5D database also.

7.3 Integrating with Geo-scientific Data

Geo-sciences and engineering, including geology, air, soil and water pollution control, civil and geotechnical engineering, require the investigation of objects of interest at levels beyond the scope of data sampling. Mathematical models involving finite element analysis are typically used. The SNIDB can facilitate such a requirement, because each geometric primitive has a finiteness property. A spatial model in the form of a 3D simplicial network may be used to answer such questions as:

- Does the clay layer extend over the entire area under the construction site?
- Which buildings are potential sources of chemical disposal into the ground?
- Will these buildings be affected by excavating the soil to 10 metres depth from the ground surface?

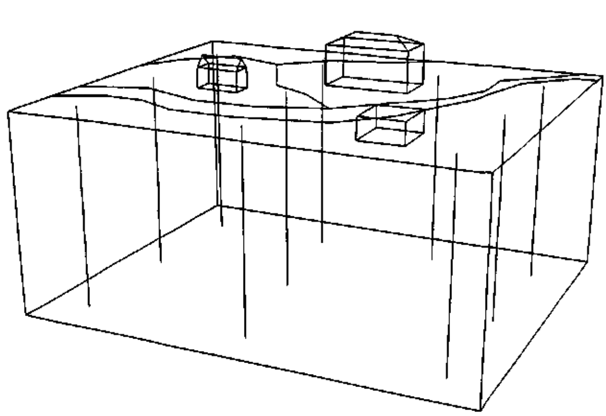


Figure 7.18 An example of borehole locations for sampling geological data. Vertical lines indicate boreholes.

Data obtained from boreholes, as shown in Figure 7.18, usually have at least one measurement in addition to the coordinates (x, y, z). The data can be stored within the SNIDB, as it is for further processing in the spatial analysis. In this case, the SNIDB must apply the tetrahedral network to structure such data. All point, line, and surface objects encountered at this stage must be involved in the tetrahedronization as constraints. The SNIDB can support a data base query for those objects as well as for some finite element analyses such as volume, bearing capacity, soil strata, and the like.

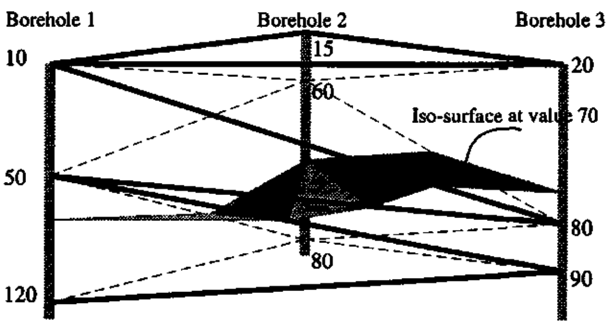


Figure 7.19 Derivation of iso-surface from a tetrahedral network constructed using three boreholes. The numbers in this figure are property values used for interpolation of the surface.

Figure 7.19 shows how an iso-surface can be derived from a TEN. The surface can be obtained in the form of a TIN.

Figure 7.20 is a wireframe plotting of the 3D simplicial network generated from the simulated borehole data, using the ISNAP program (see section 5.2.6). Figure 7.21 shows the derivation of upper and lower surfaces bounding a soil stratum from the generated 3D simplicial network.

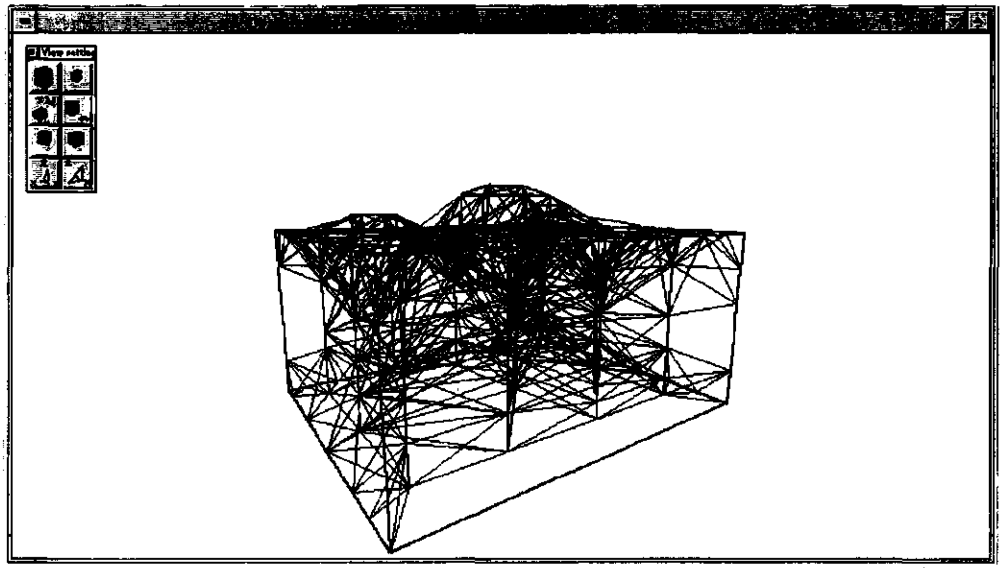


Figure 7.20 *3D simplicial network generated from simulated borehole data.*

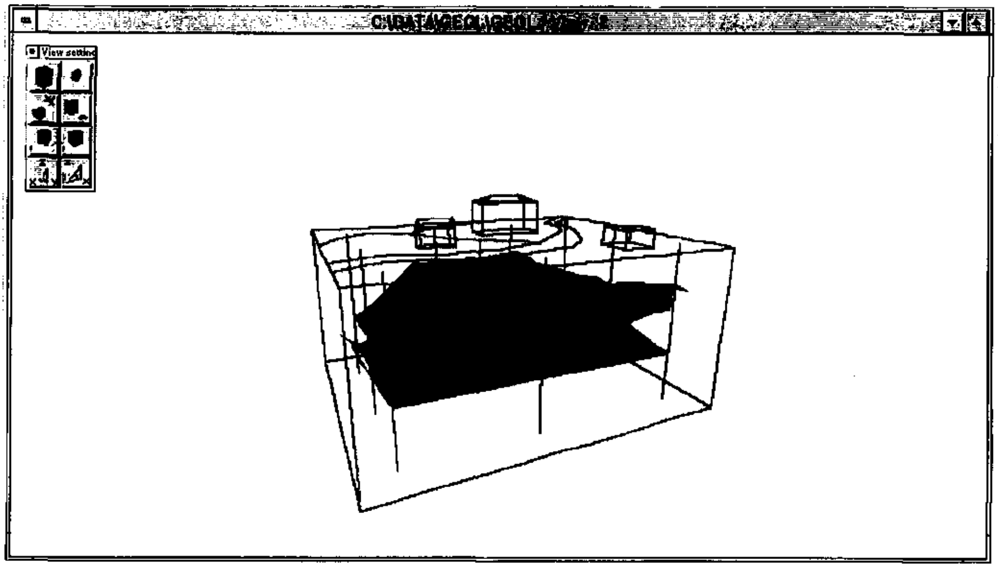


Figure 7.21 *Derivation of contour surfaces from a 3D simplicial network.*

7.4 Spatial Operators

The applicability of the designed spatial data model also depends on the user interface (for example, the query language) and the availability of spatial operators. Both of these define the functionality of the system. The spatial operators define the operation at a low level, while the spatial query language defines the operation at a relatively high level capable of being well understood by human beings.

The spatial operators are comparable to those found in mathematics, set and logic algebra. The basic operators are union, intersection, or, and, xor, and so forth. These basic operators can be combined to build more sophisticated functions. Different kinds of spatial relations, that is, metric, order and topology, are also essential to the design of spatial operators. Metric operators are built around the computational geometry, for example point-in-polygon, point-in-body, intersection of lines, and intersection of surfaces. Metric operators can be used to derive topological relationships. With respect to the simplicial network data model, these problems can be reduced to the level of a simplex. Point-in-polygon and point-in-body can be reduced to point-in-triangle and point-in-tetrahedron respectively. The intersection of bodies can be simplified to intersections between tetrahedrons, which can be further reduced to intersections between simplices of a lower dimension, for example between triangles, edges, as shown in Figure 7.22.

Order operators are those used to compare and arrange spatial elements. Topological operators are those defined by topological relationships, like containment, touch, coincidence, disjoint, left, right. For example, a body feature A is contained in another body feature B if all tetrahedrons of A are contained in B. Body A is a neighbour of Body B if a tetrahedron of A is a neighbour of a tetrahedron of B. Bodies A and B are coincident if all tetrahedrons of A are tetrahedrons of B, and vice versa. The spatial operator is essential to spatial analysis. It is a link between the spatial query language and the spatial analysis function.

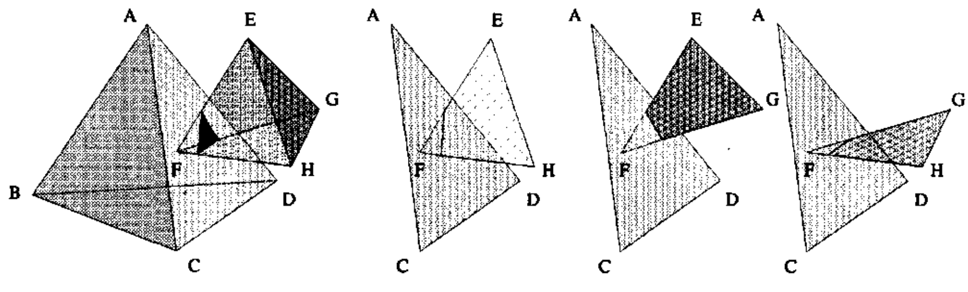


Figure 7.22 The intersection of two tetrahedrons can be reduced to intersections between triangles or intersections between edges and triangles.

The concept of a simplicial network also helps to simplify the development of 3D computational geometry. All spatial elements in the data model have finite properties. The triangle and tetrahedron are convex geometric shapes, making many complex computations simpler. Many 2D operations can be readily generalized into 3D. For example, the algorithm for point-in-triangle can be generalized into point-in-tetrahedron. An algorithm to compute the area of a triangle can also be generalized into computing the volume of a tetrahedron. Generalization into n-dimensions is implied.

Figure 7.23 (a) shows an algorithm for point-in-triangle testing. A point P_1 situated inside a triangle is always in the negative direction of the normal vector of each edge of the triangle (given that the normal of each edge points outwards from the triangle). This is not the case for a point situated outside the triangle. This assumption also holds for testing if a point is contained in a tetrahedron as shown is Figure 7.23 (b). Note that this algorithm is not valid for a non convex polygon or polyhedron. The line-intersection test can be used by counting the odd or even numbers of intersections of the line emanating from the point (see the dashed lines in Figure 7.23 (a)) with the boundary of polygon or polyhedron. An odd number indicates that the point is inside the polygon or polyhedron, while an even number indicates that the point is outside.

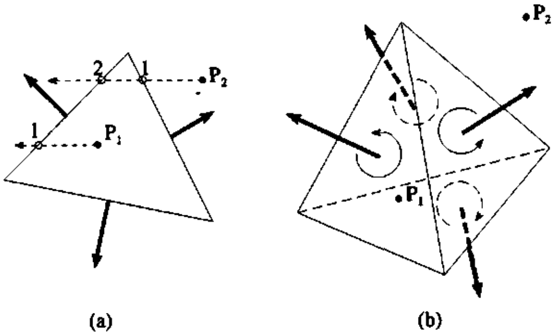


Figure 7.23 Point-in-triangle and point-in-tetrahedron testing.

Spatial operators to calculate some properties of the a spatial object, for example volume, surface area, can be designed more easily. The volume of a complex object is the summation of the volumes of all the tetrahedrons that are part of the object. The surface area of the complex object can be computed from the summation of the area of all the triangles that are part of the boundary of the object.

7.5 Graphic Visualization

In 3D geoinformation, visualization is one of the most important components of the system. Realism and interaction are necessary for information to be quickly understood. The key is speed of data processing, which relies on the power of the system and an appropriate data structure. SNIDB permits the visualization of the representations of both determinate and indeterminate spatial objects. The representation of determinate spatial objects can be displayed directly, while the derivation of the boundaries is needed for indeterminate spatial objects prior to their graphic visualization. A simplicial network supports different types of graphic visualizations, as described below.

7.5.1 Wireframe Graphics

Wireframe graphics give a relatively low level of realism. They only make use of nodes and arcs stored in SNIDB. Without interactivity, wireframe graphics seem not to be very useful for complex, or large amounts of data (see Figure 7.17, Figure 7.20). The operation to display 3D wireframe graphics consists of transforming all the coordinates of the nodes into a perspective system relating to the observer and the viewing distance to the objects.

With all the coordinates in the perspective system, the next step is to use arc and node topology to draw the straight lines connecting the beginning and end nodes of each arc. The wireframe graphic is then obtained. Examples of wireframe graphics are shown in many of the figures of section 7.2. When there is a need to differentiate different type of graphically displayed information, different styles, colours and line thickness can be used.

Visualization using wireframe graphics can be further improved by adding stereoscopic vision capability. A simple and economic approach is anaglyphic stereo, using red and blue (or green) filters to separate two parallax images from the viewer's left and right eyes. The parallax images are displayed using different camera positions along the line parallel to the eye-base of the viewer. The blue (or green) shade is used for the left image and the red shade for the right image. The viewing glasses must have red and blue (or green) colours in the opposite sense of the displayed images. The perception of depth helps to resolve visual ambiguity on a 2D display. Figure 7.24 is an example of an anaglyphic stereo pair. Note that red-blue (or green) glasses are needed to obtain a 3D effect.

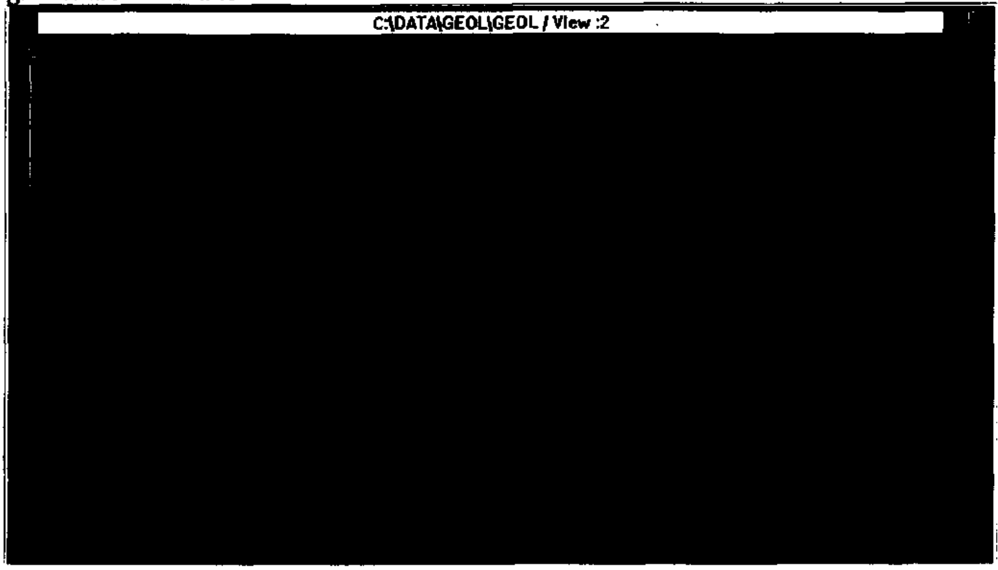


Figure 7.24 An example of wireframe images displayed in anaglyphic stereo mode.

7.5.2 Hidden Line and Surface Removal

The 3D visualization with wireframe graphics can be significantly improved by applying a hidden line and surface removal operation. For this purpose, many algorithms are available in computer graphics and CAD (Beaty and Booth 1982, Foley et al 1992). One of the most efficient and relatively simple algorithms is known as a 'z buffer' (or 'depth-buffer') which is the raster-based operation that only stores the pixels belonging to the visible part of the objects in the scene. For each location on the buffer that is a 2D array, only the pixel nearest to the viewer is stored. Each pixel value indicates the identifier of the facet. A simplicial network provides information in the form of a triangle that can be used for the purpose. However, some extra spatial index structure (for example, a BSP-tree) needs to be constructed on top of the core database to speed up and ease the operation. For example, if the depth sorting algorithm is used, all triangles need to be sorted according to the distance from the viewer. The remote triangles are displayed first, while the triangle closest to the viewer is the last to be displayed. Each triangle must be filled by background colour, while its boundary is drawn in foreground colour. In this way, part of the objects that should be invisible are overwritten. Only the visible parts remain on the display device. The z-buffer is also a kind of index structure in the form of a regular grid.

7.5.3 Surface Shading and Illumination

When the hidden line and surface removal operation described above has been applied, surface shading and illumination can take place next in the sequence. Colour can be assigned to each triangle and then displayed by a filling operation during the hidden line and surface operation. For surface illumination, the lighting model must be used to compute the colour intensity for each triangle. The intensity depends on the amount of light reflecting from the facet to the viewer. The lighting models available are Gaurad shading or Phong illumination (Foley et al, 1992). The general light geometry is shown in Figure 7.25.

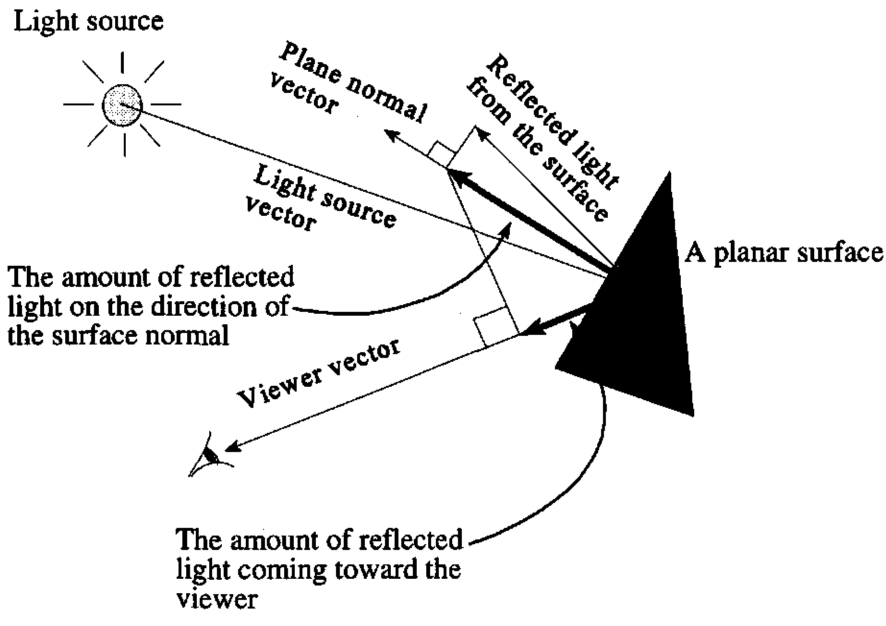


Figure 7.25 Principle of surface illumination.

A simplicial network provides the planar surface that is ready to be used for the calculation of a plane's normal vector. Since the order of vertices of the facet determines the direction of the normal vector, it is advisable to store this information systematically in the database. For upwards so as to be able to interact naturally with simulated sunlight. To calculate the amount of light reflecting from a triangle to the viewer, the reflected light from the surface is first projected onto the normal vector of the triangle. This projected light is then projected onto the viewer vector which determines the intensity of light the viewer perceives from this light source. Some other factors also influence how the viewer sees the shade and colour intensity of the triangle. Such a factor is type of material, which determines the roughness and shininess attributes of the surface features. When there is more than one light source, such as those reflecting from other surfaces nearby, the summation of the individual reflectances can be taken to be the total intensity.

7.5.4 Texture Mapping

If the texture information for each facet is available, it may be used to fill the surface during display instead of normal colour filling. Illumination can still be applied to improve realism. The hidden line and surface removal operation needs to be applied beforehand. Since texture mapping is a raster operation, the texture array and the array of pixels indicating visible facets must be stored in parallel in the buffer memory during the operation. This operation requires powerful hardware and software because of the great deal of memory and large number of resampling operations needed.

Incorporating texture information into a simplicial network is also possible. Texture information can be provided mathematically as a function, or as an image which is typically in a raster form. Only the latter is considered here. For the raster data structure, incorporating texture information in the form of an image is quite straightforward. Incorporating texture information (known as 'texture mapping') can be challenging. Knowledge of computational geometry, photogrammetry and digital image processing are needed. The texture mapping helps improve the visualization of geoinformation.

The process for texture mapping using an image involves the solving of image transformation relative to parameters of the camera used to capture the scene. If the vector data is completed with 3D coordinates, it can be transformed to match the camera orientation and then superimposed onto the image by taking into account the visibility of each vector element. The image can then be segmented by the vector elements of SNIDB, that is to say, the nodes, edges, and triangles. After the segmentation, the texture information can be stored along with each element of SNIDB. Each node has a pixel value stored as an additional attribute. For an edge, a set of pixels along the edge must be stored with the pixel size or scale factor. Eventually, this edge needs to be stretched or contracted, depending on the perspective transformation. Pixel values can be interpolated to fill the gaps between pixels (see Figure 7.26).

For a face or a triangle, all the pixels falling inside the face or triangle during superimposition must be stored. In order to normalize the camera parameters, this set of pixels may be resampled to a coordinate system that is orthogonal to the face or the triangle on the same scale. The storage of each segmentation (a face or a triangle) is then in the form of a rectangular image in which the image size is the same as the bounding rectangle of each segmentation. When the data has to be graphically viewed using different transformation parameters, the image must be re-sampled to map onto each facet. The affine transformation can be applied by taking the corners of the bounding rectangle as the control points to determine the transformation parameters. The pixels that have data values are then mapped onto the facet in a perspective view. To eliminate the gap in the resulting view, four adjacent pixels can be used as vertices of a square drawn as a quadrangle in the perspective view. The colour of the quadrangle can be determined, for example, by taking the average value of the four pixels. The process is shown in Figure 7.27.

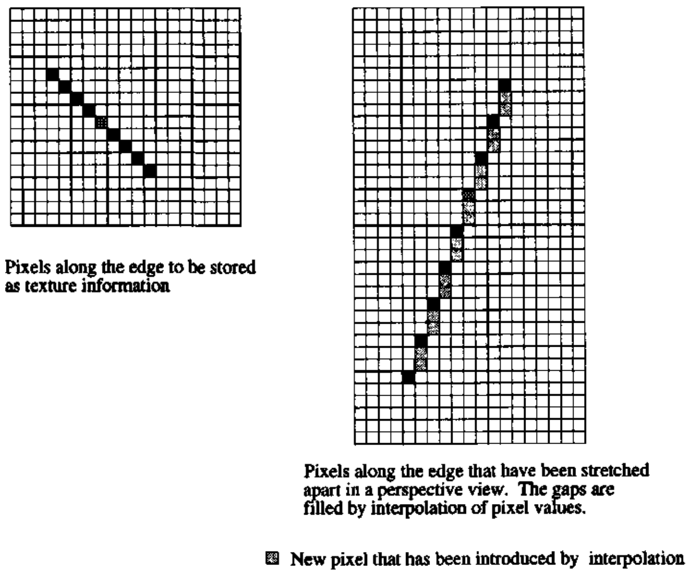


***Figure 7.26** Resampling operation along an edge.*

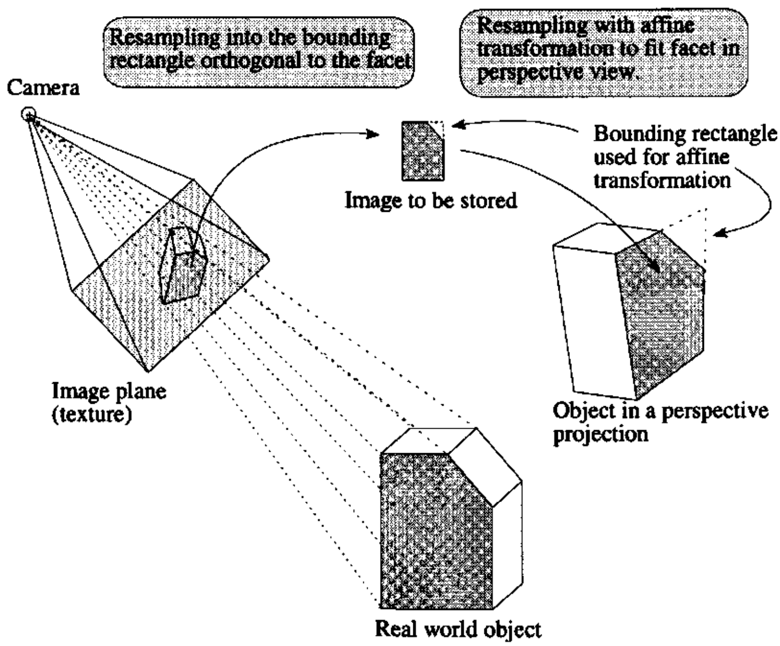


***Figure 7.27** Operation of texture mapping with respect to a face.*

With respect to the DBMS aspect, many modern and commercially available DBMSs (for example, dBASE V, Illustra) are already capable of storing and managing an image as the attribute of a record. The previous explanation would therefore suggest it to be feasible to incorporate the texture information into the integrated database. The data structure in a relational form may look like the following table.

Face Id	Texture file	Name of texture array	Dimension (W x H)	Pixel size (mm)
347	FW1.DAT	Front_wall	30 x 50	0.2

The bounding rectangle can also be derived directly from the dimension of the texture image, using the lower-left corner of the image as the origin. The upper-right corner is just the addition of the width and height of the image to the origin, which can be started from (0, 0). The transformation problem is limited to 2D and is relative to each face. Detailed discussion about digital image transformation can be found in Wolberg (1990).

7.6 Virtual Reality

VR can be used to explore the content of information stored in an SNIDB or 3D FDS database. VR provides highly interactive, realistic and dynamic visualization. It uses almost all the visualization techniques described in section 7.5, so that powerful hardware and software are needed. VR allows the continuous change of viewing position and tries to provide ways of interacting with the representation of spatial objects as happens in reality. Understanding the spatial model can be readily achieved if the information content stored in the database is displayed appropriately. The storage of 3D coordinates and boundary representations in SNIDB and 3D FDS are compatible with many VR systems. This means VR technology can be adapted to access information stored in SNIDB directly. Thematic attributes of each feature can be translated into specific colour, shade, type of material or texture, to be graphically rendered in each scene. Spatial relationships stored in the SNIDB provide constraints for the virtual environment. For example, adjacent objects should remain close together at any viewing distance or direction in the virtual world. An experiment using the constructed 3D spatial model in the VR environment has been conducted. The 3D spatial model shown in Figure 7.17 was converted into the VRML (virtual reality modelling language) and could be viewed by many VR systems (see appendix G for an example of scene). It is expected that future 3D GIS will have built-in VR functionality for interactive visualization and other kinds of responses, such as sound.

7.7 Discussion

The concept of a simplicial network can be applied to the integrated modelling of reality, for example the integration of terrain relief and terrain features. That has been a serious problem in geoinformation science. The spatial model resulting from this integration supports operations typically needing both DTM and 2D GIS. Queries about features and relief information can occur together. A simplicial network representing the earth's surface and 2D representation of terrain objects can also be integrated into a database of 3D FDS for urban application. For applications in the geo-sciences, for example geology, or environmental monitoring, spatial objects presented in a database of 3D FDS can be incorporated into the 3D simplicial network as constraints to facilitate better derivation of the representation of spatial objects with indeterminate spatial extent. The database in the form of a simplicial network also facilitates various kinds of 3D visualization, even when high interactivity and realism (for example virtual reality) are required, provided there is an appropriate extension of the data structure to accommodate more attributes, for example texture, or colour. Also, spatial index structures suitable for each kind of operation must be built on top of this core database for efficiency in terms of response time. Although conceptually, the simplex elements of the simplicial network data model help simplify many complex operations, the limitations on applying the SNIDB are still the lack of 3D spatial operators. Further development is indicated. These operators are metric, order and topological operators for the computation of volume, surface area, testing of containment, intersection, touch, disjoint, coincidence, and so forth. Such set operators as union, intersection, difference, or, xor, are also needed. Once these operators are available, the applicability of SNIDB will be extended significantly.

8
CONCLUSIONS AND RECOMMENDATIONS

Research and development within the scope of 3D-GIS is now extensive. This thesis can only deal with some parts of it. The emphasis here is on the conceptual and logical design of a 3D spatial model and how it can be constructed. Some examples utilizing the integrated 3D spatial model with respect to the design introduced in this thesis are also given. The conclusions drawn from this research and the recommendations for further research follow.

8.1 Conclusions

Several problems associated with 3D GIS were identified in chapter 1. The scope of this thesis, however, restricts the emphasis to various stages of the design and construction of a 3D vector spatial model. This kind of model permits the integration in one database of two kinds of real world objects: determinate and indeterminate spatial objects and their components. This integration permits better representation of the spatial relationships between the two types of spatial objects. Determinate spatial objects—objects with discernible boundaries, like buildings and roads— can be represented directly by the elements of the model. Indeterminate spatial objects—objects with indiscernible boundaries, like soil strata, temperature, and mineral deposits—require indirect representation. Given a specific type of property and a given property value, or the property range, the boundary of an indeterminate spatial object can be derived from the surrounding neighbours. In a vector spatial model, the neighbours may be represented as a point, line, surface, or body feature.

When the boundary of an indeterminate spatial object has been derived, this object then becomes a determinate object capable of being visualized and allowing further spatial analysis (computation of volume, surface area, relationships with other spatial objects, etc). To make this possible, the neighbours must form a spatial unit permitting the performance of operations (interpolation, classification) to make the boundary of the indeterminate spatial object explicit. To obtain an accurate result—the (derived) boundary of the indeterminate spatial object—the characteristics of the neighbours must be taken into account as constraints. For example, underground discontinuities like geological faults, obtained from interpreting seismic data, may have to be incorporated directly into the spatial model so that the derivation of orebody from drillhole samples of mineral deposits can be obtained more accurately.

The review of the current situation indicates that existing systems do not provide adequate 3D modelling tools for earth science applications needing to model the relationships between determinate and indeterminate spatial objects. Moreover, the components of spatial objects are often represented in separate spatial models, such as in models of terrain relief in DTM and terrain features in typical 2D GIS. The consequence of these is the difficulty of accurate representation of the relationships between objects in a spatial model. It is evident that the key problem is the lack of a spatial data structure suitable for this kind of modelling which also permits the adaptation of various available technological developments to be implemented as functions of 3D GIS. Therefore, an appropriate spatial data model has to be developed to make it possible to derive such a spatial data structure. Although attempts towards the design of a 3D spatial data model have been made, this aspect of integrated modelling of the two types of spatial objects had not been adequately addressed. The main objective of the research was, therefore, to design a data model suitable for the integrated modelling of the two kinds of spatial objects and accordingly to suggest a simple method of constructing a spatial model as well as to demonstrating its utilization.

The study commences with a review of all the necessary fundamental concepts of geo-spatial modelling incorporated into the design, construction and maintenance phases. Although different theories and concepts abound, only those supporting the design of a spatial model are reviewed and ordered with respect to the conceptual and logical design phases. By relating and bringing some order into those theories and concepts, the study also contributes to the further development of spatial theory.

Conceptual design

Since a 3D GIS needs to adapt various technological developments for its functionality, a review of these technological developments was carried out. In addition to this, the present architecture of the geoinformation systems and future development trends are analysed and differentiated into four evolution stages: independent subsystems, functional integration, client/server, and structural integration. The independent subsystem is the common approach, since a GIS has been developed whereby available subsystems in the form of hardware and software are taken as components of a GIS. These subsystems evolve into software modules of a GIS in the next evolution stage, the functional integration. However, few of the GISs developed at this stage can provide all the functions the users require. The client/server architecture, which is evolution stage 3, emerges offering an intermediate solution. This architecture makes use of communication technology to exchange information between independent subsystems connected on-line. Nevertheless, the architecture of the systems in these three evolution stages still relies on various independent data structures specific to functions or subsystems.

Since separate data storage requires different DBMSs, many problems persist. These problems are summarized in chapter 3. The thesis anticipates evolution stage 4, the structural integration, that is expected to offer solutions with all functions relying on a common database. This database provides the information necessary for all the operations in geo-spatial modelling. A unified data structure is the basis of the system. The system provides various database views and spatial index structures specific to functions or operations on top of the unified data structure. The client/server approach can be adopted on top of the structural integration architecture which allows each developer to concentrate on a set of functions of 3D GIS. The review of some attempts towards structural integration with respect to 3D GIS shows that the design of spatial data model is needed to permit the derivation of a unified data structure for a 3D GIS employing a structural integration architecture to accommodate both direct and indirect representations of spatial objects.

To contribute to the development of 3D GIS adopting architecture based on structural integration, the design of an integrated data model and the development of the method to construct the spatial model were carried out. The simplicial network data model (SNDM) is the result of the conceptual design. The SNDM provides general concepts valid for spatial models ranging from 2.5D to nD. The SNDM has the following properties:

1) Theoretical aspects:
i) The components of the model are distinguished into geometric, feature and thematic class levels in the same way as FDS.
ii) Complex spatial objects are decomposed into simplices. The Delaunay concept of taking complex objects as constraints is used in the decomposition method to provide spatial units suitable for the indirect representation. All simplices contribute to the geometry of the simplicial network.
iii) A simplicial network as well as its components can be described using graph theory. Each simplex is a complete graph, therefore, a simplicial network is a network of complete graphs with different degrees of nodes. This concept makes the simplicial network a sound and consistent structure. Each network has mathematical characteristics that accord with a generalized Euler equality.

2) Practical aspects:
i) The network provides basic computation units suitable for finite element analysis.
ii) The network accommodates both direct and indirect representations. The direct representation implies a high fidelity property of representing spatial objects.
iii) The network has a locality property, so it is suitable for use as a structure for the storage of a large database where a spatial model can be maintained without large perturbations to the model as a whole.
iv) The irregularity of the network makes it adaptable to spatial variation in reality. This makes the spatial model versatile.
v) The network is a complete tessellation of space, allowing more freedom to navigate within the spatial model using various means, such as topology, order, metric computation or their combinations. For example, the derivation of iso-lines or iso-surfaces makes use of a combination of different means to navigate in the spatial model, while query about features make use of topology as a navigation means.

Each component of a simplicial network has the following properties:
i) convex shape
ii) irregular shape
iii) finiteness, therefore, it is verifiable against a complete graph
iv) simplest geometry in its internal dimension,

These properties make it possible to automate many operations ranging from the construction of a 3D spatial model (for example, the constrained Delaunay network formation) to the derivation of information as required by applications in earth sciences (for example, the computation of spatial gradient, iso-lines, iso-surfaces) as indicated in chapter 1.

Logical design

With respect to the logical design, a unified data structure (UNS) can be derived from the SNDM. This allows for the handling of a spatial model by a single DBMS. Two different logical designs using the relational and object-oriented approaches ensure that a SNDM is feasible. The UNS provides elements for storing the necessary information for various operations. Regardless of the speed of the spatial operations, a relational UNS can be implemented using many commercially available DBMSs that already provide the basic operations to create, retrieve and update a database and its elements. Normalization using Smith's method is applied to obtain relational UNS, providing better updating the database. Typical relational DBMSs do not provide functions for spatial operations. These operations have to be implemented in addition to the basic database operations.

The logical design using the object-oriented approach shows that SNDM can be implemented differently for better performance that is not well provided for by the relational approach as a result of the unsuitable indexing method. The implementation using C++ in the ISNAP program demonstrates the practicability of the object-oriented approach. Instead of relating components of a spatial model by join operations and Cartesian products, as is typical in the relational approach, relationships among the components of a spatial model can be implemented as pointers. Spatial searches can be more efficient, as can be seen in operations like the derivation of grid DTM and contour lines.

Construction of a 3D spatial model

For practical use, a method of constructing such a spatial model must be available. The constrained network construction is the method of constructing a spatial model based on SNDM. Incorporating representations of determinate spatial objects as constraints into the simplicial network is the most important issue. For 2D network construction, both raster and vector approaches are available and ready for use. Since no simple method for 3D network construction with constraints is available, generalizing a 2D algorithm for 3D is feasible. The vector approach is, however, very complicated to generalize for 3D network construction. Generalizing the raster approach was achieved within this study. The general method valid for nD is devised to incorporate constraints into the simplicial network. The method is based on the invariant property of Voronoi regions under Voronoi tessellation, using distance transformation. The geometry of line and surface features can be embedded within the 3D simplicial network as required. A further achievement is the construction of a spatial model based on 3D FDS required as an intermediate structure for storing features to be used as constraints in the 3D network construction. This achievement is, however, limited to man made objects such as buildings and roads extracted by photogrammetric digitizing from a stereo model. The construction of this kind of model is achieved by using the digitized outline of the building roofs and DTM, as explained in chapter 6.

Implementation

The object-oriented UNS and method for constrained network construction were implemented in the ISNAP program. Some functions that are only available on separate systems (2D GIS and DTM) can now be implemented in ISNAP, together with some additional functions required for 3D modelling. In short, ISNAP has the following functionality:

- 2D Delaunay network construction with constraints
- 3D Delaunay network construction

- Graphic display:
  - orthogonal, perspective and stereo views
  - wireframe or surface illumination
  - hidden line and surface removal
- Query of point, line and surface features that can be performed in any display view
- Derivation of contour lines, contour surfaces
- Derivation of regular-grid DTM.

Apart from ISNAP, other development in the field of 3D GIS have been carried out. TREVIS has been developed to explore the capability of the relational approach and 3D FDS for 3D GIS and uses as tool for 3D visualization with some 3D editing capability. TREVIS can perform various kinds of queries using functions provided by a commercial relational DBMS, dBASE IV.

Testings

The applicability of the SNDM is demonstrated through three tests. These tests were conducted using both TREVIS and ISNAP as tools for constructing the model: query, process, and visualization. The first test is specific to the problem of the integrated modelling of terrain relief and terrain features typically handled separately by DTM and 2D GIS. ISNAP was used to perform constrained triangulation, overlaying, query of features, deriving contour lines and regular-grid. Basic GIS and DTM functions could be performed on one database. It can be concluded that the integrated modelling of terrain relief and terrain features is achievable using SNDM.

The second test shows that the representation of a surface in the form of a simplicial network can be integrated into a spatial model in the form of a 3D FDS that contains the representation of 3D spatial objects. The study area was the central area of Enschede, consisting of different kinds of buildings. The terrain of the study area was represented in the form of simplicial networks that facilitate the construction of the geometry of 3D representation of buildings. Representations of the footprints of these buildings were incorporated into a simplicial network of the surface, successfully integrated with 3D objects representing buildings and stored within a 3D FDS database in the subsequent process.

The third test demonstrates the construction of a 3D simplicial network and the derivation of the boundary of indeterminate spatial objects from this network. A tetrahedral network was constructed from simulated borehole data using the ISNAP program. Iso-surfaces that are the assumed boundaries of a soil layer (for instance, clay) were derived from this tetrahedral network and then visualized in perspective and stereo mode. Determinate objects such as buildings and roads could be incorporated into this network for visualization purposes, or for complex analysis.

The test results shown in chapter 7 are obtained from TREVIS and ISNAP. This ensures SNDM can be implemented and the resulting spatial model is practicable. The model can fulfil various requirements, that is integrated modelling of determinate and indeterminate spatial objects, supporting complex queries, various kinds of visualizations. Boundaries of indeterminate spatial objects can be derived and visualized.

The ISNAP program demonstrates that the 3D spatial model based on UNS can facilitate various operations. Tasks in the scope of 3D GIS (described as the functionality of ISNAP) that typically require many different systems and databases can be carried out using ISNAP and an integrated database. This makes the ISNAP a simple example of 3D GIS adopting the structural integration architecture. Various functions are available and reachable from one control panel with a common user-interface, making the system more convenient to use and significantly reducing time in dealing with many different systems. Since one database can facilitate many operations, data redundancy due to storing duplicate data in different databases (for example databases of terrain relief and terrain features) is eliminated. Accessibility to each data element is improved because all components of spatial model are stored in one database. Users need not deal with uncertainty during spatial query or computation. This means many requirements stated in chapter 3 can be fulfilled. ISNAP also demonstrates that various technological developments, construction of spatial model, query and deriving information from the model, 2D and various 3D visualization techniques and so forth, can be integrated into one system that uses SNDM.

8.2 Recommendations

A simplicial network data model has been proposed. The unified data structure and method to construct the corresponding spatial model have been designed and implemented. Some of the practicability of the constructed model has been demonstrated. The objectives of this thesis have been achieved. However, further investigations and developments still need to be carried out.

- Development of tools for 3D operations

Tools for 3D operations with respect to 3D GIS are still lacking. Some examples of these tools are:
- interactive 3D editing with realistic visualization
- 3D overlay
- implementation of point-in-tetrahedron testing
- conversion between a 3D irregular network and 3D regular grid useful for many operations
- a virtual reality interface for conveniently exploring content of a 3D database.

- Implementation of 3D constrained network construction

Although the concept of constrained network construction using the raster approach has been generalized for n-dimensions, only the constrained triangulation was implemented in ISNAP. The constrained tetrahedronization still needs to be implemented.

- Development of 3D spatial index

Many operations in 3D GIS, for example for realistic visualization, require data to be organized in a specific structure for efficiency. These are task-oriented index structures. The integrated database can only provide data in a basic structure and so requires the index structure appropriate for each task to be built on top of it. There are still requirements to identify index structure that provide efficient operation for each task. An object-oriented approach is potential for this kind of development; however, further studies, implementation and experiments are still needed. Such a study should include how to incorporate various database views and spatial index structures with the core database.

• Further tests for applicability of the simplicial network data model with some evaluations, for example against handling complex queries, finite element analysis, visualization.

• Methods to handle uncertainty covering 3D cases, for example to resolving lines or planes that coincide in 3D space,

• Maintenance of a 3D simplicial network spatial model; this is the problem of updating the spatial database which also requires the development of consistency rules.

• Comparative study raster and vector approaches for constrained network construction with respect to speed, memory and storage usage and overall efficiency.

• High quality 3D cartographic presentation of 3D spatial model, including:
- design of 3D symbols
- design of artificial texture
- text and name placement in 3D space
- use of 3D database for pictorial maps
- 3D graphic generalization.

BIBLIOGRAPHY

Alagic, S.,1989, *Object-oriented database programming*, Springer Verlag, New York, 320 pp.

Alexandroff, P., 1961, *Elementary concepts of topology*, Dover Publications, Inc., New York.

Alia, A., and Williams, H.,1994, *Approaches to the representation of qualitative spatial relationships for geographic databases*, In: Molenaar, M., and De Hoop, S., (Eds.), *Advanced geographic data modelling*, Netherlands Geodetic commission, pp. 204-216.

Anton, H., 1987, *Elementary linear algebra*, Fifth edition, John Wiley & Sons.

Arc/Info, 1991, *Surface modelling with TIN*, Arc/Info user's guide, ESRI, U.S.A.

Armstrong, M.A., 1983, *Basic topology*, Springer, New York.

Avis, D., and Bhattacharya, K.B., 1983, *Algorithms for computing d-dimensional Voronoi diagrams and their duals*, Advances in Computing Research, 1, pp. 159-180.

Ayugi, S.W.O., 1992, *The multi-valued vector map*, M.Sc. Thesis, ITC, Enschede, The Netherlands.

Bak, P.R.G., and Mill, A.J.B., 1989, *Three dimensional representation in a geoscientific resource management system for minerals industry*, In: Raper, J., (Ed.) 1989, *Three dimensional applications in geographical information systems*, Taylor & Francis, London, pp. 155-182.

Barbalata, J.C., and Lebel, R., 1992, *Digital elevation model for photogrammetric measurements of soil erosion*, International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Part B4, Commission IV, Washington, D.C., U.S.A., pp. 831-835.

Batten, L.G., 1989, *National capital urban planning project: development of a 3-D GIS*, Proceedings of GIS/LIS '89, ACSM/ASPRS, Falls Church, pp. 781-786.

Beaty, J.C., and Booth, K.S., 1982, *Tutorial: computer graphics*, Second Edition, IEEE Computer Society Press, Silver Spring, MD.

Bernal, J., 1988, *On constructing Delaunay triangulation for a set of constrained line segments*, Technical Note 1252, National Institute of Standards and Technology, United States of Commerce.

Blum, H, 1967, *A transformation for extracting new descriptors of shape*, Proceedings of Symposium on Models for Perception of Speech and Visual Form, MIT Press, Cambridge, Mass., pp. 362-380.

Borgefors, G., 1984, *Distance transformations in arbitrary dimensions*, Computer Vision, Graphics, and Image Processing, 27, pp. 321-345.

Borgefors, G., 1986, *Distance transformations in digital images*, Computer Vision, Graphics, and Image Processing, 34, pp. 344-371.

Bouloucos, T., Ayugi, S.W.O., and Kufoniyi, O., 1993, *Data structure for multi-valued vector maps*, Proceedings Fourth European Conference on Geographical Information Systems (EGIS'93), Genoa, Italy, pp. 237-245.

Bouloucos. T, Kufuniyi, O., and Molenaar, M., 1990, *A relational data structure for single valued vector maps*, International Archives of Photogrammetry and Remote Sensing, Vol. 28, Part 3/2, Commission III, Wuhan, China, pp. 64-74.

Bowyer, A., 1981, *Computing Dirichlet tessellation*, Computer Journal, 24, pp. 162-166.

Brassel, K.E., and Reif, D., 1979, *Procedure to generate Thiessen polygons*, Geographical Analysis, 11, pp. 289-303.

Bric, V., and Pilouk, M. 1994, *Computation of topologic space*, In preparation, ITC, Enschede, The Netherlands.

Bric, V., Pilouk, M., and Tempfli, K., 1994, *Towards 3D-GIS: Experimenting with a vector data structure*, International Archives of Photogrammetry and Remote Sensing, Vol. XXX, Part 4, Athens, Georgia, USA, pp. 634-640.

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

Brockschmidt, K., 1993, *Programming for Windows with Object Linking and Embedding 2.0*, Microsoft Press.

Burrough, P.A., 1986, *Principles of geographical information systems for land resources assessment*, Clarendon Press, Oxford University Press, 194 pp.

Cambray, B. de, 1993, *Three-dimensional (3D) modelling in a geographical database*, Proceedings 11th International Symposium on Computer Assisted Cartography (AUTOCARTO 11), Minneapolis, pp. 338-347.

Cantor, G., 1880, *Über unendliche, lineare Punktmannigfaltigkeiten*, Math. Ann. (B) 17, pp. 355-388.

Carlson, E., 1987, *Three dimensional conceptual modeling of subsurface structures*, Technical Papers, Vol. 4, ASPRS-ACSM Annual Convention, Baltimore, Maryland, pp. 188-200.

Chen, P. P-S., (Ed.) 1983, *The entity-relationship approach to information modelling & analysis*, International Conference. Proceedings, North-Holland.

Chen, X., Ikeda, K., Yamakita, K., and Nasu, M., 1994, *Raster algorithms for generating Delaunay tetrahedral tessellations*, International Archives of Photogrammetry and Remote Sensing, Commission III, Vol. 30, Part 3/1, Munich, Germany, pp. 124-131.

Chhatkuli, R.R., 1993, *Modelling data quality parameters in a multiple-theme vector data structure and its implementation in a geographic information system*, M.Sc. Thesis, ITC, Enschede, The Netherlands.

Chew, L.P., 1989, *Constrained Delaunay triangulations*. Algorithmica, 4, pp. 97-108.

Collin, W.J., 1992, *Data structures: an object-oriented approach*, Addison-Wesley Publishing Company, Reading Massachusetts, 624 pp.

Corbett, J.P., 1979, *Topological principles in cartography*, Technical paer 48, U.S. Department of Commerce, Bureau of the cencus, 50 pp.

Danielsson, P.E., 1980, *Euclidean distance mapping*, Computer Graphics and Image Processing, 14, pp. 227-248.

Dahl, O-J, Myrhaug, B., and Nygaard, K., 1970, *SIMULA common base language*, Norwegian Computing Center S-22, Oslo, Norway.

Date, C.J., 1986, *An introduction to database systems*, Vol. 1, Addison-Wesley, Reading, Mass.

Devlin, K., 1994, *Mathematics: The sciences of patterns*, Scientific American Library, New York, 215 pp.

De Floriani, L., and Puppo, E., 1988, *Constrained Delaunay triangulation for multi resolution surface description*, Proceedings of the 9th International Conference on Pattern Recognition, Rome, Italy, pp. 566-569.

De Floriani, L., and Puppo, E., 1992, *An on-line algorithm for constrained Delaunay triangulation*, CVGIP: Graphical Models and Image Processing, 54, pp. 290-300.

Delaunay, B., 1934, *Sur la sphére vide*, Bulletin of the Academy of Sciences of the USSR, Classe des Sciences Mathématiques et Naturelles, 8, pp. 793-800.

Dirichlet, G.L., 1850, *Über die reduction der positiven quadratischen formen mit drei unbestimmten ganzen zalen*, J. Reine u. Angew. Math., 40, pp. 209-227.

DLG-E, *Digital Line Graph-Enhanced*, U.S. Department of the Interior, U.S. Geological Survey.

Dwyer, R.A., 1987, *A fast divide-and-conquer algorithm for constructing Delaunay triangulations*, Algorithmica, 2, pp. 137-151.

Ebner, H. and Eder, K., 1992, *State-of-the-art in digital terrain modelling*, Proceedings Third European Conference on Geographical Information Systems (EGIS'92), Volume. 1, Munich, Germany, pp. 681-690.

Ebner, H., Hossler, R., and Wurlander, R., 1990, *Integration of an efficient DTM program package into geographical information systems*, International Archives of Photogrammetry and Remote Sensing, Vol. 28, Part 4, Commission IV, Tsukuba, Japan, pp. 116-121.

Edelsbrunner, H., Preparata, F.P., and West, D.B., 1986, *Tetrahedrizing point sets in three dimensions*, Technical Report UIUCDCS-R-86-1310, Department of Computer Science, University of Illinois, 1304 W. Springfield Avenue, URBANA, Il 61801.

Ehlers, M., Greenlee, D., Smith, T., and Star, J., 1991, *Integration of remote sensing and GIS: Data and data access*, Photogrammetric Engineering & Remote Sensing, Vol. 57, No. 6, pp. 669-675.

Ehlers, M., Edwards, G., and Bédard, 1989, *Integration of remote sensing with geographic information systems: a necessary evolution*, Photogrammetric Engineering & Remote Sensing, Vol. 55, No. 11, pp. 1619-1627.

Egenhofer, M.J., 1991, *Extending SQL for cartographic display*, Cartography and Geographic Information Systems, Vol. 18, No. 4, pp. 230-245.

Egenhofer, M.J., 1990, *Interaction with geographic information systems via spatial queries*, Journal of Visual Languages and Computing, Vol. 1, No. 4, pp. 389-413.

Egenhofer, M.J., 1989, *A formal definition of binary topological relationships*, Technical Report No. 101, NCGIA/Department of Surveying Engineering, University of Maine, Orono, ME, USA.

Egenhofer, M.J., and Frank, A.U., 1989, *Object-oriented modelling in GIS: Inheritance and propagation*, Proceedings 9th International Symposium on Computer Assisted Cartography (AUTOCARTO 9), Baltimore, Maryland, pp. 588-589.

Egenhofer, M.J., Frank, A.U., and Jackson, J.P., 1989, *A topological data model for spatial databases*, NCGIA Technical Report, No. 104.

Egenhofer, M.J., and Franzosa, R.D., 1991, *Point-set topological spatial relations*, International Journal for Geographical Information Systems, Vol. 5, No. 2, pp. 161-174.

Egenhofer, M.J., and Herring, J.R., 1990, *A mathematical framework for the definition of topological relationships*, Proceeding of the Fourth International Symposium on Spatial Data Handling, Zurich, Switzerland, pp. 803-813.

Egenhofer, M.J., and Herring, J.R., 1992, *Categorizing binary relationships between regions, lines, and points in geographical databases*, Technical report, Department of Surveying Engineering, University of Main, USA.

Field, A.D., 1986, *Implementing Watson's algorithm in three dimensions*, In Proceedings of ACM Symposium on Computational Geometry, pp. 246-259.

Field, A.D., and Smith, W.D., 1991, *Graded tetrahedral finite element meshes*, International Journal for Numerical Methods in Engineering, 31(3), pp. 413-425.

Finkbeiner, D.T., and Lindstrom, W.D., 1987, *A primer of discrete mathematics*, W.H. Freeman and Company, New York, 363 pp.

Fisher, T.R., 1993, *Use of 3D geographic information systems in hazardous waste site investigations*, In: Goodchild, M.F., Parks, B., and Steyaert, L., (Eds.), *Environmental Modeling with GIS*, Oxford University Press, New York.

Flankin, W.M., 1984, *Cartographic errors symptomic of underlying algebra problems*, in: Marble, D., et al. (Eds.), Proceedings of the International Symposium on Spatial Data Handling, Zurich, Switzerland.

Flavin, M., 1981, *Fundamental concepts of information modelling*, Yourdon Press Computing Series, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA, 128 pp.

Flowerdew, R., 1991, *Spatial data integration*, in Maguire, D.J., Goodchild, M.F., and Rhind, D.W., (Eds.), *Geographical information systems principles and applications*, Longman Scientific & Technical, pp. 375-387.

Foley, J.D., van Dam, A., Feiner, S.K., and Hughes, J.F., 1992, *Computer graphics: principles and practice*, Second Edition, Addison-Wesley, U.S.A., 1175 pp.

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

Frank, A.U., 1992, *Spatial concepts, geometric data models and data structures*, Computer & Geosciences, Vol. 18, No. 4, pp. 409-417.

Frank, A.U., and Kuhn, W., 1986, *Cell graphs: a provable correct method for the storage of geometry*, Proceedings of the Second International Symposium on Spatial Data Handling, Seattle, Washington, USA., pp. 411-436.

Fréchet, M., 1906, *Sur quelques points du calcul fonctionnel*, Rendiconti di Palermo 22, 1-74.

Fritsch, D., 1990, *Towards three-dimensional data structures in geographic information systems*, Proceedings First European Conference on Geographical Information Systems (EGIS'90), Volume. 1, Amsterdam, The Netherlands, pp. 335-345.

Fritsch, D., and Pfannenstein, A., 1992a, *Integration DTM data structures into GIS data models*, International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Part B3, Commission III, Washington, D.C., USA., pp. 497-503.

Fritsch, D., and Pfannenstein, A., 1992b, *Conceptual models for efficient DTM integration into GIS*, Proceedings Third European Conference on Geographical Information Systems (EGIS'92), Volume. 1, Munich, Germany, pp. 701-710.

Gatrell, A.C., 1991, *Concepts of space and geographical data*, In: Maguire, D.J. , Goodchild, M.F., and Rhind, D.W., (Eds.), *Geographical Information Systems*, Vol. 1: Principles, Longman, UK.

Giblin, P., 1977, *Graphs, surfaces and homology*, Chapman and Hall, London.

Goldberg, A., and Robson, D., 1983, *Smalltalk-80: the language and its implementation*, Addison-Wesley, Reading, Massachusetts.

Golda, Y.V., 1992, *The "flowing" accumulation method and its application for earth surface analysis*, International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Part B4, Commission IV, Washington, D.C., USA., pp. 836-842.

Gorte, B., and Koolhoven, W., 1990, *Interpolation between isolines based on the Borgefors distance transform*, ITC Journal Special Issue Remote Sensing and GIS, 1990-3. pp. 245-247.

Green, P. J., and Sibson, R., 1978, *Computing Dirichlet tessellations in the plane*, Computer Journal, 21, pp. 168-173.

Gruen, A., Streilein, A., Stallmann, D., and Dan, H., 1993, *Automation of house extraction from aerial and terrestrial images*, Conference ASIA, Wuhan, China.

Guptill, C., and Morrison, J.L., 1995, *Elements of spatial data quality*, Elsevier Science Ltd.,

Guttman, A., 1984, *A dynamic index structure for spatial searching*, Proceedings of the SIGMOD Conference, Boston, pp. 47-57.

Gütting, R., 1988, *Geo-relational algebra: a model and query language for geometric database system*, in Schmidt, J., Ceri, S., and Missikoff, M., (Eds.), Advances in Database Technology-- EDBT '88, International Conference on Extending Database Technology, Venice, Italy, Lecture Notes in Computer Science, Springer Verlag, New York, Vol. 303, pp. 506-527.

Hausdorff, F., 1914, *Grundzüge der mengenlehre*, Leipzig. Reprinted by Chelsea, New York (1949) 88 pp.

Hawryszkiewycz, I.T., 1991, *Introduction to system analysis and design*, second edition, Prentice Hall, Australia.

Hearnshaw, H.M., and Unwin, D.J., 1994, *Visualization in geographical information systems*, Wiley and Sons, 243 pp.

Herring, J.R., 1989, *A fully integrated geographic information system*, Proceedings 9th International Symposium on Computer Assisted Cartography (AUTOCARTO 9), pp. 828-837.

Herring, R.J., and Egenhofer, M.J., 1990, *A mathematical framework for the definition of topological relationships*, Proceedings of the Forth International Symposium on Spatial Data Handling, Zurich, Switzerland, pp. 803-813.

Herring, J., Larsen, R., and Shivakumar, J., 1988, *Extensions to the SQL language to support spatial analysis in a topological data base*, Proceedings GIS/LIS '88, San Antonio, Texas, pp. 741-750.

Hesse, W., and Leahy, F.J., 1990, *Authoritative topographic-cartographic information system (ATKIS). The Project of the State Survey Authorities for the Creation of Digital Landscape Models and Digital Cartographic Models*, Landesvermessungamt Nordrhein-Westfalen, Bonn, 29 pp.

Howe, D.R., 1989, *Data analysis for database design*, second edition, Edward Arnold A division of Hodder & Stoughton, London, 317 pp.

Illustra, 1994, *Illustra, relational databases and spatial data*, An Illustra Technical White Paper by Malcolm Colton, Oakland, CA, 6 pp.

Intergraph, 1995, *New OLE extensions for CAD/CAM/CAE and GIS adopted*, Press Releases, Intergraph Corp., Huntsville, http://www.intergraph.com/press95/dmpr.html.

Jackson, J., 1989, *algorithms for triangular irregular networks based on simplicial complex theory*, ASPRS-ACSM Annual Convention, Baltimore, MD, USA., pp. 131-136.

Jackson, M.A., 1983, *System development*, Prentice Hall, 418 pp.

Joe, B., 1989, *Three-dimensional triangulations from local transformations*, Siam Journal on Scientific and Statistical Computing, 10(4), pp. 718-741.

Jones, C.B., 1989, *Data structures for three-dimensional spatial information systems in geology*, International Journal of Geographical Information Systems, Vol.3, No. 1, Taylor & Francis, London, pp. 15-31.

Kainz, W., 1989, *Order, topology and metric in GIS*, ASPRS-ACSM Annual Convention, Baltimore, Vol. 4, pp. 154-160.

Kainz, W., 1990, *Spatial relationships--topology versus order*, Proceedings of the Fourth International Symposium on Spatial Data Handling, Zurich, Switzerland, Brassel, K., and Kishimoto, H., (Eds.), Vol. 2, pp. 814-819.

Kainz, W., Egenhofer, M., and Greasley, I., 1993, *Modeling spatial relations and operations with partially ordered sets*, International Journal of Geographical Information Systems, Vol. 7, No. 3, pp. 215-229.

Kainz, W., and Shahriari, N., 1993, *Object-oriented tools for designing topographic databases*, Proceedings GIS/LIS'93, pp. 341-350.

Kanaganathan, S., and Goldstein, N.B., 1991, *Comparison of four-point adding algorithms for Delaunay-type three-dimensional mesh generators*, IEEE Transactions on Magnetics, 27(3), pp. 3444-3451.

Kavouras, M., and Masry, S., 1987, *An information system for geosciences, design considerations*, Proceedings 8th International Symposium on Computer Assisted Cartography (AUTOCARTO 8), Baltimore, MD, pp. 336-345.

Kemp, Z., 1990, *An object-oriented model for spatial data*, Proceedings 4th International Symposium on Spatial Data Handling, Vol. 2, Zurich, Switzerland, pp. 659-668.

Kinsey, L.C., 1993, *Topology of surfaces*, Springer Verlag, New York, 276 pp.

Knuth, D.E., 1973, *The Art of computer programming*, Vol. 3: Sorting and Searching, Addison-Wesley, Reading.

Kraak, M-J., 1992, *Working with triangulation-based spatial data in 3D space*, ITC Journal 1992-1, pp. 20-33.

Kraak, M-J., and Verbree, E., 1992, *Tetrahedrons and animate maps in 2D and 3D space*, In Proceedings of the 5th International Symposium on Spatial Data Handling, pp. 63-71.

Kufoniyi, O., 1995, *Spatial coincidence modelling, automated database updating and data consistency in vector GIS*, Ph.D. Thesis, Wageningen Agricultural University, The Netherlands, 206 pp.

Kufoniyi, O., 1989, *Editing of topologically structured data*, M.Sc. Thesis, ITC, Enschede, The Netherlands.

Kufoniyi, O., and Bouloucos, T., 1994, *Flexible integration of terrain objects and DTM in vector GIS*, Proceedings International Colloquium on Integration, Automation and Intelligence in Photogrammetry, Remote Sensing and GIS, Wuhan, pp. 111-122.

Kuhn, W., and Frank, A.U., 1991, A formalization of metaphors and image-schemas in user interfaces, in Mark, D., and Frank, A., (Eds.), Cognitive and linguistic aspects of geographic space, Kluwer Academic Publ., Dordrecht, pp. 419-434.

Langran, G., 1992, *Time in geographic information systems*, Taylor & Francis, London.

Larkin, B.J., 1991, *An ANSI C program to determine in expected linear time the vertices of the convex hull of a set of planar points*, Computers & Geosciences, 17, pp. 431-443.

Laurini, R., and Thompson, D., 1992, *Fundamentals of spatial information systems*, Academic Press, London, 680 pp.

Lawson, C.L., 1985, *Some properties of n-dimensional triangulation*, External Report, JPL Publication 85-42, National Aeronautics and Space Administration.

Lawson, C.L., 1977, *Software for C1 surface interpolation*, In Rice, J., (Ed.), Mathematcal Software III, Academic Press, Newyork, USA, pp. 161-194.

Lawson, C.L., 1972, *Generation of a triangular grid with application to contour plotting*, California Institute of Technology, Jet Pollution Laboratory, Technical Memorandum No. 299.

Lee, D.T., and Lin, A.K., 1986, *Generalized Delaunay triangulation for planar graph*, Discrete & Computational Geometry, 1, pp. 201-217.

Lee, D.T., and Schachter, B.J., 1980, *Two algorithms for constructing a Delaunay triangulation*, International Journal of Computer and Information Sciences, 9, pp. 219-242.

Lewis, B.A., and Robinson, J.S., 1978, *Triangulation of planar regions with applications*, Computer Journal, 21, pp. 324-332.

Li, R., 1993, *Three-dimensional GIS: a simple extension in the third dimension?*, ACSM/ASPRS Annual Convention, New Orleans, USA, Vol. 3, pp. 218-227.

Lingas, A., 1986, *The greedy and Delaunay triangulations are not bad in the average case*, Information Processing Letters, 22, pp. 25-31.

Liu, C.L., 1986, *Elements of discrete mathematics*, Second edition, McGraw-Hill, 433 pp.

Macedonio, G., and Pareschi, M.T., 1991, *An algorithm for the triangulation of arbitrarily distributed points: applications to volume estimate and terrain fitting*, Computer & Geosciences, 17, pp. 859-874.

Maguire, D.J., and Dangermond, J., 1991, *Functionality of GISs*, In: Maguire, D.J., Goodchild, M.F., and Rhind, D.W., (Eds.), Geographical Information Systems, Vol. 1, Principles, Harlow: Longman Scientific & Technical, pp. 319-335.

Maguire, D.J., Goodchild, M.F., and Rhind, D.W., (Eds), 1991, *Geographical information systems: principles and applications*, Longman Scientific & Technical.

Makarovic, B., 1984, *Structures for geo-information and their application in selective sampling for digital terrain models*, ITC Journal 1984-4, pp. 285-295.

Makarovic, B., 1977, *Composite sampling for DTMs*, ITC Journal.

Males, R.M., 1978, *ADAPT - a spatial data structure for use with planning and design models*, In: Dutton, G., (Ed.), First International Symposium on Advance Study on Topological Data Structures for Geographic Information Systems, Vol. 3, 19 pp.

Manacher, G.K., and Zobrist, A.L., 1979, *Neither the greedy nor the Delaunay triangulation of a planar point set approximates the optimal triangulation*, Information Processing Letters, 9, pp. 31-34.

Mäntylä, M., 1988, *Solid modelling*, Computer Science Press, Rocville, Maryland, 401 pp.

Marble, D.F., Calkins, H.W., and Peuquet, D.J., 1984, *Technical description of the DIME system*, Basic Readings in Geographic Information Systems, SPAD Systems, Ltd. USA., pp. 57-64.

Mark, D.M., Lauzon, J.P., and Cebrian, J.A., 1989, *A review of quadtree-based strategies for interfacing coverage data with digital elevation models in grid form*, International Journal of Geographical Information Systems, Vol.3, No. 1, Taylor & Francis, London, pp. 3-14.

Martin, J., 1983, *Managing the data-base environment*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Maus, A., 1984, *Delaunay triangulation and the convex hull of n points in expected linear time*, BIT, 24, pp. 151-163.

McCullagh, M.J., and Ross, C.G., 1980, *Delaunay triangulation of a random data set for isarithmatic mapping*, The Cartographic Journal, 17, pp. 93-99.

Meier, A., 1986, *Applying relational database techniques to solid modeling*, CAD, Vol.18, No.6, pp. 319-326.

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

Microsoft, 1993, *Object linking and embedding: OLE 2.0 design specification*, Microsoft Corporation.

Midtbø, T., 1993a, *Incremental Delaunay tetrahedrization for adaptive data modelling*, Proceedings Fourth European Conference on Geographical Information Systems (EGIS'93), Genoa, Italy, pp. 227-236.

Midtbø, T., 1993b, *Spatial modelling by Delaunay networks of two and three dimensions*, Dr. Ing. Thesis, Norwegian Institute of Technology, University of Tronheim, Norway, 147 pp.

Miller, C.L., and Laflamme, R.A., 1958, *The digital terrain model - theory and application*, Photogrammetric Engineering, pp. 433-442.

Mirante, A., and Weingarten, N., 1982, *The radial sweep algorithm for constructing trianguled irregular networks*, IEEE Computer Graphics and Applications, 2, pp. 11-21.

Moellering, H., 1991, *Spatial database transfer standards: current international status*, Elsevier Applied Science, 247 pp.

Moise, E.E., 1977, *Geometric topology in dimension 2 and 3*, Springer Verlag, New York.

Molenaar, M., 1994a, *A syntax for representation of fuzzy spatial objects*, In: Molenaar, M., and De Hoop, S., (Eds.), *Advanced geographic data modelling: spatial data modelling and query languages for 2D and 3D applications*, Netherlands Geodetic Commission, No. 40, Delft, The Netherlands, pp. 155-169.

Molenaar, M., 1994b, *A syntactic approach for handling the semantics of fuzzy spatial objects*, European Science Foundation, GISDATA, Baden, Austria, 15 pp.

Molenaar, M., 1993, *Object hierarchies and uncertainty in GIS or why is standardisation so difficult?*, Geo-Informations-Systeme, Vol. 6, No. 4, pp. 22-28.

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

Molenaar, M., 1991, *Formal data structures, object dynamics and consistency rules*, Digital Photogrammetric Systems, Herbert Wichmann Verlag GmbH, Karlsruhe, pp. 262-273.

Molenaar, M., 1990, *A Formal data structure for 3-D vector maps*, Proceedings First European Conference on Geographical Information Systems (EGIS'90), Volume. 2, Amsterdam, The Netherlands, pp. 770-781.

Molenaar, M., 1989, *Single valued vector maps - a concept in GIS*, Geo-Informations-Systeme, Vol. 2, No. 1, pp. 18-27.

Molenaar, M., 1988, *Single valued polygon maps*, International Archives of Photogrammetry and Remote Sensing, Vol. 27, Part B4, Commission IV, Kyoto, Japan, pp. 592-601.

Ning, S., 1992, *On the principles and the approaches of implementing the strict digital geometric rectification for SPOT imagery*, International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Part B3, Commission III, Washington, D.C., USA., pp. 32-34.

Oosterom, P. van., 1990, *Reactive data structures for geographic information systems*, Ph.D. Thesis, Leiden University, The Netherlands, 197 pp.

Orenstein, J. A., 1990, *An object-oriented approach to spatial data processing*, Proceedings of the 4th International Symposium on Spatial Data Handling, Vol. 2, Zurich, Switzerland, pp. 669-698.

Peng, W., and Molenaar, M., 1995, *An object-oriented approach to automated generalization*, Proceedings of GeoInformatics '95, Vol. 1, Hong Kong, pp. 295-304.

Peng, W., Tempfli, K., and Molenaar, M., 1995, *Automated generalization in a GIS context*, to be published in Proceedings of GeoInformatics '96, Florida, USA, 11 pp.

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

Petrie, G., and Kennie, T., 1990, *terrain modelling in surveying and civil engineering*, Whittles, Latheronwheel, UK.

Peuquet, D.J., 1988, *Representations of geographic space: toward a conceptual synthesis*, Annals of the Association of American Geographers, 78, pp. 375-94.

Peuquet, D.J., 1986, *The use of spatial relationships to aid spatial database retrieval*, Proceedings of the Second International Symposium on Spatial Data Handling, Seattle, WA.

Peuquet, D.J., 1984, *A conceptual framework and comparison of spatial data models*, CARTOGRAPHICA, Vol. 21, No. 4, pp. 66-113.

Pfannenstein, A., and Reinhardt, W., 1993, *Data analysis in geographical information systems in combination with integrated digital terrain models*, Proceedings Fourth European Conference on Geographical Information Systems (EGIS'93), pp. 1341-1349.

Pigot, S., 1992, *A topological model for a 3-D spatial information system*, Proceedings of 5th International Symposium on Spatial Information Handling, Charleston, S.C., pp. 344-360.

Pigot, S., 1991, *Topological models for 3D spatial information systems*, Proceedings 10th International Symposium on Computer Assisted Cartography (AUTOCARTO 10), Technical Papers, ACSM-ASPRS, Annual Convention, Vol. 6, Baltimore, Maryland, USA., pp. 369-391.

Pilesjo, P., Michelson, D.B., and Hall-Konyves, K., 1992, *Digital elevation models for identification of potential wetlands*, International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Part B4, Commission IV, Washington, D.C., USA., pp. 817-822.

Pilouk, M., Radjabi Fard, A., and Tempfli, K., 1994, *Local updating of TIN for the integrated DTM and GIS data structure*, International Archives of Photogrammetry and Remote Sensing, Vol. XXX, Part 4, Athens, Georgia, USA, pp. 460-466.

Pilouk, M., and Kufoniyi, O.,1994, *A relational data structure for integrated DTM and multitheme GIS*, International Archives of Photogrammetry and Remote Sensing, Commission III, Vol. 30, Part 3/2, Munich, Germany, pp. 670-677.

Pilouk, M., Tempfli, K., Molenaar, M., 1994, *A tetrahedron-based 3D vector data model for geoinformations*, In: Molenaar, M., and De Hoop, S., (Eds.), *Advanced geographic data modelling: spatial data modelling and query languages for 2D and 3D applications*, Netherlands Geodetic Commission, No. 40, Delft, The Netherlands, pp. 129-140.

Pilouk, M., and Tempfli, K., 1994, *An object oriented approach to the unified data structure of DTM and GIS*, International Archives of Photogrammetry and Remote Sensing, Vol. XXX, Part 4, Athens, Georgia, USA, pp. 672-679.

Pilouk, M., and Tempfli, K., 1994, *Integrating DTM and GIS using a relational data structure*, GIS'94, Vol. 1, Vancouver, Canada, pp. 163-169.

Pilouk, M., and Tempfli, K., 1993, *An integrated DTM-GIS data structure: a relational approach*, Proceedings 11th International Symposium on Computer Assisted Cartography (AUTOCARTO 11), Minneapolis, Minnesota, USA, pp. 278-287.

Pilouk, M., and Tempfli, K., 1992, *A digital image processing approach to creating DTMs from digitized contours*, International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Part B4, Commission IV, Washington, D.C., USA., pp. 956-961.

Pilouk, M., 1992, *Fidelity improvement of DTM from contours*, M.Sc. Thesis, ITC, Enschede, The Netherlands.

Pohl, I., 1993, *Object-oriented programming using C++*, Benjamin/Cummings Publishing Company, Inc., California, 496 pp.

Pullar, D., 1988, *Data definition and operators on a spatial data model*, ACSM-ASPRS, Annual convention, Vol. 2, pp. 196-202.

Pullar, D., and Egenhofer, M., 1988, *Towards formal definitions of topological relations among spatial objects*, Proceedings of the Third International Symposium on Spatial Data Handling, Sydney, Australia, pp. 225-242.

Raper, J., 1993, *Three dimensional GIS for the 1990*, Seminar on 3 Dimensional GIS Recent Developments, ITC, Delft, The Netherlands, pp. 4-5.

Raper, J. F., 1989, *The 3-dimensional geoscientific mapping and modelling system: a conceptual design*, In: Raper, J.F. (Ed), *Three dimensional applications in geographic information systems*, Taylor & Francis, London.

Raper, J.F., and Kelk, B., 1991, *Three-dimensional GIS*, In: Maguire, D.J., Goodchild, M.F., and Rhind, D.W., (Eds), *Geographical information systems principles and applications*, Longman Scientific & Technical, pp. 299-316.

Requicha, A.A.G., 1980, *Representation for rigid solids: theory, methods, and systems*, Computing Surveys, Vol. 12, No. 4.

Richardson, D.E., 1993, *Automated spatial and thematic generalization using a context transformation model*, Ph.D. Thesis, R&B Publications, Canada, 149 pp.

Rikkers, R., Molenaar, M., and Stuiver, J., 1993, *A query oriented implementation of 3D topologic datastructure*, Proceedings Fourth European Conference on Geographical Information Systems (EGIS'93), Genoa, Italy, pp. 1411-1420.

Roessel, J.W. van, 1986, *Design of a spatial data structure using the relational normal forms*, Proceedings of the 2nd International Symposium on Spatial Data Handling, Seattle, pp. 251-272.

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

Roushannejad, A.A., 1993, *Mathematical morphology in automatically deriving skeleton lines from digitized contours*, M.Sc. Thesis, ITC, Enschede, The Netherlands.

Samet, H., 1990, *The design and analysis of spatial data structures*, Reading, Addison-Wesley, Massachusetts.

Sandgaard, J., 1988, *Integration of a GIS and a DTM*, International Archives of Photogrammetry and Remote Sensing, XVI Congress, Commission III, Kyoto, Japan, pp. 716-725.

Scott, M.S., 1994, *The development of an optimal path algorithm in three-dimensional raster space*, M.Sc. Thesis, Department of Geograhy, University of South Carolina, 108 pp.

Shamos, M.I., and Hoey, D., 1975, *Closest-point problems*, Proceedings of the 16th Annual Symposium on the Foundations of Computer Science (Washington: IEEE), pp. 151-162.

Shephard, M.S., and Schroeder, W.J., 1990, *A combined octree/Delaunay method for fully automatic 3-D mesh generation*, International Journal for Numerical Methods in Engineering, 29, pp. 37-55.

Shepherd, I.D.H., 1991, *Information integration and GIS*, In: Maguire, D.J., Goodchild, M.F., and Rhind, D.W., (Eds), *Geographical information systems principles and applications*, Vol. 1, Longman Scientific & Technical, New York, USA, pp. 337-360.

Shi, W., 1994, *Modelling positional and thematic uncertainties in integration of remote sensing and geographic information systems*, Ph.D. Thesis, International Institute for Aerospace Survey and Earth Sciences, Enschede, The Netherlands, 147 pp.

Shibasaki, R., Shimizu, E., and Nakamura, H., 1990, *Three dimensional (3D) digital map for an urban area*, International Archives of Photogrammetry and Remote Sensing, Vol. 28, Part 4, Commission IV, Tsukuba, Japan, pp. 211-220.

Shibasaki, R., and Shaobo, H., 1992, *A digital urban space model - a three dimensional modelling technique of urban space in a GIS environment*, International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Part B4, Commission IV, Washington, D.C., USA., pp. 257-264.

Shmutter, B., and Doytsher, Y., 1988, *An algorithm to assemble polygons*, ACSM-ASPRS Annual Convention, St. Louis, Missouri, pp. 98-105.

Sibson, R., 1978, *Locally equiangular triangulations*, Computer Journal, 21, pp. 243-245.

Sides, E.J., 1992, *Modelling three-dimensional geological discontinuities for mineral evaluation*, Ph.D. Thesis, University of London, 281 pp.

Slingerland, R., and Keen, T.R., 1990, *A numerical study of storm driven circulation and 'event bed' genesis*, Proceedings of Symposium on Structures and Simulating Processes, Freiburger Geowissenschafliche Beitrage, 2, pp. 97-99.

Sloan, S.W., 1987, *A fast algorithm for constructing Delaunay triangulations in the plane*, Advanced Engineering Software, 9, pp. 34-55.

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.

Smith T.R., Menon, S., Star, J.L., and Estes, J.E., 1987, *Requirements and principles for the implementation and construction of large-scale geographical information systems*, International Journal of Geographical Information Systems, 1:13-32.

Smith, D.R., and Paradis, A.R., 1989, *Three-dimensional GIS for the earth sciences*, In: Raper, J.F., (Ed.), *Three dimensional applications in geographical information systems*, Taylor & Francis, London, pp. 149-155.

Sommerville, D.M.Y., 1929, *An introduction to the geometry of N dimensions*, Dover publications, Inc., New York, 196 pp.

Stanat, D.F., and McAllister, D.F., 1977, *Discrete mathematics in computer science*, Prentice-Hall, Englewood Cliffs, NJ.

Sutherland, I.E., 1963, *Sketchpad: A man-machine graphic communication system*, TR-296, MIT Lincoln Laboratory, Lexington, Mass.

Sutherland, I.E., 1970, *Computer displays*, In: Beatty, J.C., and Booth, K.S., (Eds.),1982, *Tutorial: Computer Graphics*, IEEE Computer Society Press, Silver Spring, MD, pp. 4-20.

Takahashi, M., and Yokokawa, T., 1992, *The automatic selection system of transmission line routes based on DTM*, International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Part B4, Commission IV, Washington, D.C., USA., pp. 883-885.

Tang, L., 1992, *Raster algorithms for surface modelling*, International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Part B3, Commission III, Washington, D.C., USA., pp. 566-573.

Tansel, A.U., Clifford, J., Gadia, S., Seger, A., and Snodgrass, R., 1993, *Temporal databases, theory, design, and implementation*, Benjamin/Cummings Publishing Company, Inc., California.

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

Tempfli, K., 1982, *Notes on interpolation and filtering*, ITC Lecture Note, 3rd edition, ITC, Enschede, The Netherlands.

Tempfli, K., and Makarovic, B., 1978, *Transfer functions of interpolation methods*, ITC Journal 1978-1. pp. 50-78.

Thiessen, A.H., 1911, *Precipitation averages for large areas*, Monthly Weather Review, July, 39, pp. 1082-1084.

TIGER, *Topologically integrated geographic encoding and referencing system*, U.S. Department of Commerce, Bureau of the Census.

Tri-service data standard, 1994, *CADD/GIS Technology Center.*

Tsai, V.J.d., 1993, *Delaunay triangulations in TIN creation: an overview and a linear-time algorithm*, International Journal Geographical Information Systems, Vol. 7, No. 6, Taylor & Francis Ltd., pp 501-524.

Tsai, V.J.d., and Vonderohe, A.P., 1993, *Delaunay tetrahedral data modelling for 3-D GIS applications*, Proceedings GIS/LIS '93 Conference, Minneapolis, Minnesota, Vol. 2, pp. 671-680.

Tsai, V.J.d., and Vonderohe, A.P., 1991, *A generalized algorithm for the construction of Delaunay triangulations in Euclidean n-space*, Proceedings GIS/LIS '91 Annual Conference, Atlanta, GA, Vol. 2, pp. 562-571.

Turner, A.K., 1989, *The role of 3-D GIS in subsurface characterization for hydrogeological applications*, In: Raper, J.F., (Ed.), *Three Dimensional Applications in Geographic Information Systems*, Taylor & Francis, London, pp. 115-127.

Vaidyanathaswamy, R., 1960, *Set topology,* Chelsea Publishing Company, New York.

Voronoi, G., 1908, *Nouvelles applications des parameters continus á la théorie des formes quadratiques,* Deuxiéme Mémoire: Recherches sur les parallelloedres primitifs. Journal fur die Reïne und Angewandte Mathematik, 134, pp. 198-287.

Wang, Z.J., 1994, *Digital photogrammetric data acquisition for 3D GIS,* M.Sc. Thesis, ITC, Enschede, The Netherlands, 88 pp.

Watson, D.F., 1981, *Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes,* Computer Journal, 24, pp. 167-172.

Watson, D.F., and Philip, G.M., 1984, *SURVEY: Systematic triangulations,* Computer Vision, Graphics and Image Processing, Vol. 26, pp. 217-223.

Watt, A., 1993, *3D computer graphics,* Addison-Wesley Publishing Company Inc., UK, 500 pp.

Webster, C.J., 1990, *The object-oriented paradigm in GIS,* International Archives of Photogrammetry and Remote Sensing, Commission III, Vol. 28 Part 3/2, Wuhan, China, pp. 947-984.

Webster, C.J., and Omare, C.N., 1991, *A formal approach to object-oriented spatial database design,* Proceedings Second European Conference on Geographical Information Systems (EGIS'91), Vol. 2, Brussel, pp. 1210-1218.

Weibel, R., 1993, *On the integration of digital terrain and surface modeling into geographic information systems,* Proceedings 11th International Symposium on Computer Assisted Cartography (AUTOCARTO 11), Minneapolis, pp. 257-266.

Weiskamp, K., and Flamig, B., 1992, *The Complete C++ Primer,* 2nd edition, Academic Press, Inc., USA., 540 pp.

Willard, S., 1970, *General topology,* Reading, Addison-Wesley, Massachusetts, USA. 369 pp.

Wilson, R.J., 1985, *Introduction to graph theory,* Third Edition, Longman Scientific & Technical, UK.

Wolberg, G., 1990, *Digital image warping,* Los Alamo, IEEE Computer Society Press, 318 pp.

Worboys, M.F., Hearnshaw, H.M., and Maguire, D.J., 1990, *Object-oriented data modelling for spatial databases,* International Journal of Geographical Information Systems, Taylor & Francis Ltd., Vol. 4, No. 4, pp. 369-383.

Würländer, R., 1988, *Undersuchung zur Integration von digitalen geländemodellen in raumbezogene informationssystemme,* Diplomarbeit, Technische Universität München.

Youngman, C., 1989, *Spatial data structures for modeling subsurface features,* In: Raper, J.F., (Ed.), *Three Dimensional Applications in Geographic Information Systems,* Taylor & Francis Ltd., pp. 129-136.

APPENDIX A

A Proof of a Generalized Euler's Equality

Given the equation

$$Nodes + Triangles = Arcs + Tetrahedrons + 1$$

the variant for *n*-dimension is:

$$0_{simplices} + 2_{simplices} + \ldots + k_{simplices} = 1_{simplices} + 3_{simplices} + \ldots + l_{simplices} + 1$$

where: k is even; $(0 \leq k \leq n)$
l is odd; $(1 \leq l \leq n)$
$0_{simplices}$ = number of nodes
$2_{simplices}$ = number of edges (arcs),
$k_{simplices}$ = number of simplices of dimension k
$l_{simplices}$ = number of simplices of dimension l

Another variant for an n-dimensional complex is:

$$0_{complexes} + 2_{complexes} + \ldots + k_{complexes} = 1_{complexes} + 3_{comolexes} + \ldots + l_{complexes} + i$$

where: k is even; $(0 \leq k \leq n)$
l is odd; $(1 \leq l \leq n)$
i = degree of isolation

Different kinds of degree of isolation can be indicated by the number of isolated objects. The isolated objects are:

- isolated nodes that have no connection to any arcs
- isolated arcs (dangling arc does not fall into this type)
- isolated faces (dangling face does not fall into this type)
- isolated bodies (*eg* holes in a body)

For n-dimensions, isolated objects of 4-dimensions and above should also be included.

The above formulae can be written in a general form:

for only one simplicial network with no isolation:

$$\boxed{\sum_{k=0}^{k \leq n} (-1)^k N_k = 1}$$

for a simplicial network with *I* degree of isolation:

$$\sum_{k=0}^{k \le n} (-1)^k N_k = \sum_{k=0}^{k \le n} I_k$$

where n = dimension number
N_k = number of k-simplices
I_k = number of isolated objects or sets of mutually connected objects.

The following theorems supported by a set of proofs are introduced to prove the above formulae.

Theorem A-1: In a complete graph with n nodes (k_n) the number of complete subgraphs with odd node size (k_m, $1 \le m \le n$, m odd) is equal to the number of complete subgraphs with even node size (k_m, $2 \le m \le n$, m even) plus 1.

The above theorem defines the total number of all components of a simplex. Recall that a k_1 complete graph is a 0-simplex, a k_2 complete graph is a 1-simplex and so on. Therefore a k_n complete graph is an (n-1)-simplex.

Proof: A complete graph k_n has $\binom{n}{m}$ complete subgraphs of node-size m with $1 \le m \le n$.

Since this is comparable to a problem of distinct selection found in combinatorial mathematics, we conduct the proof by using the binomial theorem:

$$(a+b)^n = (a+b)_1(a+b)_2 \ldots (a+b)_n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k \qquad \ldots\ldots(I)$$

For a detailed algebraic proof, see Finkbeiner and Linstrom, 1987.
Here we conduct a simple numerical proof:
Let n = 2

$$\begin{aligned}(a+b)^2 &= \binom{2}{0} a^{(2-0)} b^0 + \binom{2}{1} a^{(2-1)} b^1 + \binom{2}{2} a^{(2-2)} b^2 \\ &= \left(\frac{2!}{(2-0)!0!}\right) a^2 + \left(\frac{2!}{(2-1)!1!}\right) ab + \left(\frac{2!}{(2-2)!2!}\right) b^2 \\ &= a^2 + 2ab + b^2\end{aligned}$$

Replacing *b* with *(-b)* in equation (i) yields:

$$(a+(-b))^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k}(-b)^k$$

$$(a-b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k}(-1)^k b^k$$

$$(a-b)^n = \sum_{k=0}^{n} (-1)^k \binom{n}{k} a^{n-k} b^k \qquad .....(ii)$$

$$(a-b)^n = \sum_{\substack{k=0 \\ k \text{ even}}}^{k\le n} \binom{n}{k} a^{n-k} b^k - \sum_{\substack{k=1 \\ k \text{ odd}}}^{k\le n} \binom{n}{k} a^{n-k} b^k \qquad .....(iii)$$

From (iii), put $a = b = 1$, therefore the left side of the equation, $(a-b)^n$, yields 0. Exchanging the left and right side of the result yields:

$$\sum_{\substack{k=0 \\ k \text{ even}}}^{k\le n} \binom{n}{k} - \sum_{\substack{k=1 \\ k \text{ odd}}}^{k\le n} \binom{n}{k} = 0 \qquad .....(iv)$$

Observe when $k = 0$, the equation (iv) expands as follows:

$$\sum_{\substack{k=0 \\ k \text{ even}}} \binom{n}{k} + \sum_{\substack{k=2 \\ k \text{ even}}}^{k\le n} \binom{n}{k} - \sum_{\substack{k=1 \\ k \text{ odd}}}^{k\le n} \binom{n}{k} = 0 \qquad .....(v)$$

and note that

$$\sum_{\substack{k=0 \\ k \text{ even}}} \binom{n}{k} = \binom{n}{0} = \frac{n!}{(n-0)!0!} = 1$$

thus equation (v) becomes:

$$1 + \sum_{\substack{k=2 \\ k \text{ even}}}^{k \le n} \binom{n}{k} - \sum_{\substack{k=1 \\ k \text{ odd}}}^{k \le n} \binom{n}{k} = 0$$

rearranging the above equation:

$$\sum_{\substack{k=1 \\ k \text{ odd}}}^{k \le n} \binom{n}{k} - \sum_{\substack{k=2 \\ k \text{ even}}}^{k \le n} \binom{n}{k} = 1 \qquad ....(vi)$$

and this completes the proof for one simplex only.

A numeric example is:
Given a 3-simplex (k_4), therefore n = 4:

k	0 (right side)	1 (left side)	2 (left side)	3 (left side)	4 (left side)
binomial	$\binom{4}{0}$ $\frac{4!}{(4-0)!0!}$	$\binom{4}{1}$ $\frac{4!}{(4-1)!1!}$	$\binom{4}{2}$ $\frac{4!}{(4-2)!2!}$	$\binom{4}{3}$ $\frac{4!}{(4-3)!3!}$	$\binom{4}{4}$ $\frac{4!}{(4-4)!4!}$
sign	+	-	+	-	+
result	1	4	6	4	1

$$\left(\binom{4}{1}+\binom{4}{3}\right) - \left(\binom{4}{2}+\binom{4}{4}\right) = \binom{4}{0}$$

$$(4+4)-(6+1) = 1$$

Continuing the proof for a simplicial network, we still remain in the binomial theorem.

Theorem A-2: The merging of two k_n complete graphs by mutual connection results in a loss of at least one k_m complete subgraph (a common part; and, consequently, its components that are k_{n-2}, k_{n-3}, ..., k_0 complete subgraphs), where $1 \le m < n$.

Proof: Equation (vi) can be write in the most compact form:

$$\sum_{k=1}^{k\le n}(-1)^k\binom{n}{k} = 1 \qquad ....(vii)$$

Note that the equation (vii) is still for one complete graph only. Consider the merging of two complete graphs, K_{1n} and K_{2n}. If the two graphs are not mutually connected then this case will satisfy the following condition:

$$K_{1n}+K_{2n} = \sum_{k_1=1}^{k_1\le n}(-1)^{k_1}\binom{n}{k_1} + \sum_{k_2=1}^{k_2\le n}(-1)^{k_2}\binom{n}{k_2} = 2$$

If there are i complete graphs, K_{1n}, K_{2n}, K_{3n}, ..., K_{in}, the equation becomes:

$$K_{1n}+K_{2n}+K_{3n}+\ldots+K_{in} = i$$

$$\sum_{j=1}^{i} K_{in} = \sum_{j=1}^{i}\sum_{k_j=1}^{k_j\le n}(-1)^{k_j}\binom{n}{k_j} = i$$

Therefore, i denotes the degree of isolation as has been previously mentioned.

In the case of mutual connection, if the common part of K_{1n} and K_{2n} is K_m, where $1\le m \le n$, then the result of this merging reduces one of the K_m. Note that K_m is a complete subgraph of both K_{1n} and K_{2n}, thus it also satisfies equation (vii). The merging then satisfies the following equation:

$$\begin{aligned} K_{1n}+K_{2n}-K_m &= \sum_{k_1=1}^{k_1\le n}(-1)^{k_1}\binom{n}{k_1} + \sum_{k_2=1}^{k_2\le n}(-1)^{k_2}\binom{n}{k_2} - \sum_{k_m=1}^{k_m\le m<n}(-1)^{k_m}\binom{m}{k_m} \\ &= 2-1 = 1 \end{aligned}$$

We should note also the right hand side of the above two equations; the reduction of the common part certainly brings this number back to 1.

Before continuing the proof, let us look at the definition of the Euler trail and circuit.

Definition A-1: Let G be a graph or multigraph. An Euler trail of G is a trail that covers each edge exactly once. An Euler circuit is an Euler trail that ends at its starting vertex (Finkbeiner and Lindstorm 1987).

We can generalize the above definition by generalizing an edge to a complete graph.

Definition A-2: Let G be a graph or multigraph composed of mutually connected subgraphs. A trail of G is a trail that covers each complete subgraph exactly once. A circuit is a trail that ends at its starting complete subgraph.

We can think of the formation of a simplicial network as the merging of i complete graphs, K_{1n}, K_{2n}, K_{3n}, ..., K_{in}. Recall that every two mutually connected complete graphs of node-size *n* share one complete graph, K_m, of node-size (n-1). In case the merging is a trail, there are always *i-1* of K_m, *ie* K_{1m}, K_{2m}, K_{3m}, ..., $K_{(i-1)m}$. The equation can be written:

$$K_{1n}+K_{2n}+K_{3n}+\ldots+K_{in}-(K_{1m}+K_{2m}+K_{3m}+\ldots+K_{(i-1)m} = i-(i-1) = 1$$

$$\sum_{j=1}^{i} K_{jn} - \sum_{j=1}^{i-1} K_{jm} = \sum_{j=1}^{i}\sum_{k_j=1}^{k_j\le n}(-1)^{k_j}\binom{n}{k_j} - \sum_{j=1}^{i-1}\sum_{k_j=1}^{k_j\le m<n}(-1)^{k_j}\binom{m}{k_j} = 1$$

If the merging is a circuit, the number of K_m is equal to i; K_{1m}, K_{2m}, K_{3m}, ..., K_{im}.

$$K_{1n}+K_{2n}+K_{3n}+\ldots+K_{in}-(K_{1m}+K_{2m}+K_{3m}+\ldots+K_{im}) = i-i = 0$$

$$\sum_{j=1}^{i} K_{jn} - \sum_{j=1}^{i} K_{jm} = \sum_{j=1}^{i}\sum_{k_j=1}^{k_j\le n}(-1)^{k_j}\binom{n}{k_j} - \sum_{j=1}^{i}\sum_{k_j=1}^{k_j\le m<n}(-1)^{k_j}\binom{m}{k_j} = 0$$

Nevertheless, at the closing of a circuit, the merging of the complete subgraphs at the start and the end that reduce the K_{im} eliminates one of the complete subgraphs of K_{im} that has node-size b, denoted as K_b such that $1 \le b \le m$. Since K_b is a part of other subgraphs, its absence causes the network to be corrupted and the right hand side of the above equation become zero. Therefore, K_b must be reinserted into its old place. Note also that K_b still satisfies equation (vii) and therefore brings the right hand side of the equation back to one again, as shown below.

$$K_{1n}+K_{2n}+K_{3n}+\dots+K_{in}-(K_{1m}+K_{2m}+K_{3m}+\dots+K_{im})+K_b = i-i+1 = 1$$

$$\sum_{j=1}^{i} K_{jn} - \sum_{j=1}^{i} K_{jm} + \underset{1\le b<m}{K_b} =$$

$$\sum_{j=1}^{i}\sum_{k_j=1}^{k_j\le n}(-1)^{k_j}\binom{n}{k_j} - \sum_{j=1}^{i}\sum_{k_j=1}^{k_j\le m<n}(-1)^{k_j}\binom{m}{k_j} + \sum_{k=1}^{k\le m<n}(-1)^{k}\binom{m}{k} = 1$$

To prove that K_n contains K_m and in the same manner both K_n and K_m contain K_b, where $1 \le m < n$. We instantiate *n* and *m* to *n* in equation (vii). After expansion we obtain:

(a) For K_n

$$(-1)^n\binom{n}{n}+(-1)^{n-1}\binom{n}{n-1}+(-1)^{n-2}\binom{n}{n-2}+(-1)^{n-3}\binom{n}{n-3}+\dots$$
$$+(-1)^3\binom{n}{3}+(-1)^2\binom{n}{2}+(-1)^1\binom{n}{1} = 1$$

(b) For K_m, let m = n-1

$$(-1)^{n-1}\binom{n-1}{n-1}+(-1)^{n-2}\binom{n-1}{n-2}+(-1)^{n-3}\binom{n-1}{n-3}+\dots$$
$$+(-1)^3\binom{n-1}{3}+(-1)^2\binom{n-1}{2}+(-1)^1\binom{n-1}{1} = 1$$

To come to a conclusion, we establish the hypothesis such that:
Every

$$\binom{n}{k} \textit{ contains } \binom{n-m}{k-m} \textit{ where } k\le n \textit{ and } 1\le m\le k$$

To ascertain the above hypothesis, let us consider the following equation:

$$\binom{n}{k} = \sum_{a=k}^{n} \binom{a-1}{k-1} \qquad .....(viii)$$

where $1 \le k \le n$.

By induction on k using the values (n-1), (n-2), (n-3), ..., 3, 2, 1 for equation (viii) observe the results:

$$let\ k=n\ then \binom{n}{n} = \binom{n-1}{n-1} = 1$$

$$let\ k=n-1\ then \binom{n}{n-1} = \binom{n-2}{n-2} + \binom{n-1}{n-2}$$

$$let\ k=n-2\ then \binom{n}{n-2} = \binom{n-3}{n-3} + \binom{n-2}{n-3} + \binom{n-1}{n-3}$$

$$let\ k=n-3\ then \binom{n}{n-3} = \binom{n-4}{n-4} + \binom{n-3}{n-4} + \binom{n-2}{n-4} + \binom{n-1}{n-4}$$

$$\cdots$$

$$let\ k=3\ then \binom{n}{3} = \binom{2}{2} + \binom{3}{2} + \binom{4}{2} \ldots + \binom{n-3}{2} + \binom{n-2}{2} + \binom{n-1}{2}$$

$$let\ k=2\ then \binom{n}{2} = \binom{1}{1} + \binom{2}{1} + \binom{3}{1} \ldots + \binom{n-3}{1} + \binom{n-2}{1} + \binom{n-1}{1}$$

$$let\ k=1\ then \binom{n}{1} = \binom{0}{0} + \binom{1}{0} + \binom{2}{0} \ldots + \binom{n-3}{0} + \binom{n-2}{0} + \binom{n-1}{0}$$

note the terms expressed in large bracket that assert the above hypothesis.

If we delete the terms in (a) that are common to the terms in (b) and at the same time introduce those that remain.

$$
\begin{aligned}
&(-1)^{n-1}\binom{n-2}{n-2} + \\
&(-1)^{n-2}[\binom{n-3}{n-3} + \binom{n-2}{n-3}] + \\
&(-1)^{n-3}[\binom{n-4}{n-4} + \binom{n-3}{n-4} + \binom{n-2}{n-4}] + \\
&\qquad\qquad \ldots + \\
&(-1)^{3}[\binom{2}{2} + \binom{3}{2} + \binom{4}{2} + \ldots + \binom{n-3}{2} + \binom{n-2}{2}] + \\
&(-1)^{2}[\binom{1}{1} + \binom{2}{1} + \binom{3}{1} + \ldots + \binom{n-3}{1} + \binom{n-2}{1}] + \\
&(-1)^{1}[\binom{0}{0} + \binom{1}{0} + \binom{2}{0} + \ldots + \binom{n-3}{0} + \binom{n-2}{0}] = 1
\end{aligned}
$$

We can also rewrite equation (a) by replacing each binomial with equation (viii) and excluding the terms which have (a-1) = m:

$$
\begin{aligned}
&(-1)^{n}\sum_{a=c}^{n}\binom{a-1}{c-1} + (-1)^{n-1}\sum_{a=c}^{n-1}\binom{a-1}{c-1} + (-1)^{n-2}\sum_{a=c}^{n-2}\binom{a-1}{c-1} + (-1)^{n-3}\sum_{a=c}^{n-3}\binom{a-1}{c-1} + \ldots \\
&+ (-1)^{3}\sum_{a=c}^{3}\binom{a-1}{c-1} + (-1)^{2}\sum_{a=c}^{2}\binom{a-1}{c-1} + (-1)^{1}\sum_{a=c}^{1}\binom{a-1}{c-1} = 1; \quad \textit{where } (a-1) \neq m
\end{aligned}
$$

In a general form, it is:

$$
\sum_{k=1}^{k\leq n}(-1)^{k}\sum_{a=k}^{n}\binom{a-1}{k-1} = 1; \quad \textit{where } (a-1) \neq m \qquad \ldots.(ix)
$$

In the case of a simplicial network:

$$\sum_{j=1}^{i} (K_{jn} - K_{jm}) + \underset{1 \le b < m}{K_b} = \sum_{j=1}^{i} \sum_{k_j=1}^{k_j \le n} (-1)^{k_j} \sum_{a=k_j}^{n} \binom{a-1}{k_j-1} = 1; \textit{ where } (a-1) \ne m \quad .....(x)$$

The above equation is generalized for both trail and circuit by subtracting i common parts from the merging and adding back the missing part, denoted by K_b. By expanding this equation, we get the alternate positive and negative terms for the number of simplices of each dimension (k-1) correspondingly. Recall that a complete graph K_n is an (n-1)-simplex.

For convenience, equation (x) is simplified by organizing all complete subgraphs that have the same node-size *k* into a group, denoted by N_k.

$$\sum_{k=1}^{k \le n} (-1)^k \sum_{j=1}^{i} \left(\sum_{a=k}^{n} \binom{a-1}{k-1} \right)_j = 1; \textit{ where } (a-1) \ne m$$

$$\textit{let } N_k = \sum_{j=1}^{i} \left(\sum_{a=k}^{n} \binom{a-1}{k-1} \right)_j ; \textit{ where } (a-1) \ne m$$

Hence, we can rewrite equation (x) that expresses the generalized Euler's characteristics of a simplicial network consisting of complete subgraphs with the largest node-size *n* as follows:

$$\sum_{k=1}^{k \le n} (-1)^k N_k = 1 \quad ....(xi)$$

If there are *l* subgraphs not mutually connected in such a network, the equation can be derived by taking a summation of equation (xi) and again grouping all complete subgraphs that have the same node-size, thus obtaining:

$$\sum_{k=1}^{k \le n} (-1)^k N_k = \sum_{k=1}^{k \le n} l_k \quad ....(xii)$$

To express equation (xi) and (xii) in terms of simplicial complexes, we reduce k by 1, recalling again that a K_n complete graph is an (n-1)-simplex:

for only one simplicial network with no other isolation:

$$\sum_{k=0}^{k \le n} (-1)^k N_k = 1$$

for a simplicial network with *I* degree of isolation:

$$\sum_{k=0}^{k \le n} (-1)^k N_k = \sum_{k=0}^{k \le n} I_k$$

where
- n = dimension number
- N_k = number of k-simplices
- I_k = number of isolated objects or sets of mutually connected objects.

This asserts the validity of the generalized Euler's equality.

APPENDIX B

An Example of the Implementation of a Unified Data Structure Using C++ Object-oriented Programming

```
class ADTTheme
{ protected:
         char *name;
  public:
         ADTTheme(const char Name[]);
         char *className(void) { return name; }
};

class ADTTFeature
{ friend ADTGeometry;
  protected:
         int ID;
         ADTTheme *Theme;
         ADTGeometry **Collection[10];
         void addCollection(ADTGeometry *aGeometry);
         void decCollection(ADTGeometry *aGeometry);
  public:
         ADTFeature(int id, ADTTheme *aTheme);
         ADTTheme &theme(void) { return *Theme; }
};

class ADTGeometry
{ protected:
         int ID;
         ADTFeature *Feature;
  public:
         ADTGeometry(int iden, ADTFeature *aFeat, ADTGeometry *aGeom);
         int identify(void) { return ID; }
};

class TNode : public ADTGeometry
{ protected:
         float x, y, z;
         int xp,yp;
  public:
         node(int iden, float X, float Y, float Z, ADTFeature *aFeat);
         float transform(void);
         void display(void);
};

class TArc : public ADTGeometry
{ protected:
```

```
        TNode *begin, *end;
        TTriangle *left, *right;
 public:
        TArc(int iden, TNode * Begin, TNode * End, ADTFeature *aFeature);
        void display2D(void);
        void display3D(void);
};

class TTriangle : public ADTGeometry
{ protected:
        TNode *N1, *N2, *N3;
        TArc *E1, *E2, *E3;
 public:
        TTriangle(int iden, TNode *n1, TNode *n2, TNode *n3, ADTFeature *aFeature);
        float slope(void);
        float elevation(float X, float Y);
        void draw2D(void);
        void draw3D(void);
};

class TPointF : public ADTFeature
{ public:
        TPointF(int id, ADTTheme *aTheme) : ADTFeature(id, aTheme);
};

class TLineF : public ADTFeature
{ public:
        TLineF(int id, ADTheme *aTheme) : ADTFeature(id, aTheme);
};

class TAreaF : public ADTFeature
{ public:
        TAreaF(int id, ADTTheme *aTheme) : ADTFeature(id, aTheme);
};
.
.
main()
{ ADTTheme *Th1(new ADTTheme("Control Point"));
 ADTTheme *Th2(new ADTTheme("Road"));
 ADTTheme *Th3(new ADTTheme("River"));
 ADTTheme *Th4(new ADTTheme("RailRoad"));
 ADTTheme *Th5(new ADTTheme("ArableLand"));
 TPointF *PF1(new TPointF(1, Th1));
 TLineF  *LF1(new TLineF(2, Th2));
 TLineF  *LF2(new TLineF(3, Th3));
 TLineF  *LF3(new TLineF(4, Th4));
```

```
TAreaF *AF1(new TAreaF(5, Th5));
TNode *N1(new TNode(1,100.24,150.56,50.68,PF1));
TNode *N2(new TNode(2,200.62,300.23,55.97,0));
TNode *N3(new TNode(3,300.17,215.35,52.85,0));
TNode *N4(new TNode(4,400.57,435.85,43.89,0));
TArc  *A1(new TArc(1,N1,N2,LF1));
TArc  *A2(new TArc(2,N2,N3,LF2));
TArc  *A3(new TArc(3,N3,N1,LF3));
TArc  *A4(new TArc(4,N3,N4,LF3));
 TTriangle *T1(new TTriangle(1,N1,N2,N3,AF1));
cout << "Area feature 1 " << (*AF1).theme().className() << "\n";
cout << "Line feature 1 " << (*LF1).theme().className() << "\n";
cout << "Node 1 identifier " << (*N1).identify() << "\n";
cout << "Arc 1 feature " << (*A1).PartOf().theme().className()  << "\n";
cout << "Arc 2 feature " << (*A2).PartOf().theme().className()  << "\n";
cout << "Arc 3 feature " << (*A3).PartOf().theme().className()  << "\n";
cout << "Triangle " << (*T1).PartOf().theme().className()
       << "\n";
cout << "Line 3 N primetives " << (*LF3).NPrime() << "\n";
cout << "Line feature 3 geometric 1 "
  << (*LF3).Geometric(0).PartOf().theme().className() << "\n";
cout << "Line feature 3 geometric 2 "
  << (*LF3).Geometric(1).PartOf().theme().className() << "\n";
cout << "Line feature 3 geometric 1 "
  << (*LF3).Geometric(0).identify() << "\n";
cout << "Line feature 3 geometric 2 "
  << (*LF3).Geometric(1).identify() << "\n";
cout << "Point feature 1 geometric 1 "
  << (*PF1).Geometric(0).identify() << "\n";
cout << "Arc 4 feature " << (*A4).PartOf().theme().className()  << "\n";
(*A4).PartOf(LF2);
cout << "change to "
  << (*A4).PartOf().theme().className() << "\n";
 ....
}
```

APPENDIX C

Rasterization Formulae

Rasterization is a conformal transformation from a world coordinate system into a coordinate system specific to the array, which is comparable to the coordinate system of most computer devices, *eg* image on graphic monitor, plotter. With respect to rasterization, the transformation involves only scaling and translation. In the case of 2D, only the planimetric (x, y) component of the coordinates is used. With respect to rasterization, the transformation involves only scaling and translation. In the case of 2D, only the planimetric (x, y) component of the coordinates is used and the transformation formula can be given in matrix notation as follows:

$$\begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} T_i \\ T_j \end{bmatrix}$$

where i, j = indices of the array; $1 \le i \le i_{max}$, $1 \le j \le j_{max}$
i_{max}, j_{max} = dimension of the array
a_{mn} = elements of transformation matrix where $1 \le m \le 2$ and $1 \le n \le 2$;
X, Y = world coordinate tuple;
T_i, T_j = translation vector.

For each given point *P* with (X, Y) coordinates and denoted as *P(X, Y)*, the first step of the transformation is the translation to the origin (0, 0). This is done by subtracting from all coordinate components the minimum coordinate value (X_{min}, Y_{min}) obtained from the point data set. During the transformation, the origin needs to be relocated to match the origin of the coordinate system specific to the computer device at the location comparable to the top-left corner of an image. This is done by using a negative scale factor for the Y-component in the transformation matrix. The result is a negative value of j that needs to be changed to a positive value by translating this value j_{max} units in a positive direction, thereby obtaining a positive offset from the origin of device coordinate system. It can be expressed as follows:

$$\begin{bmatrix} i \\ j \end{bmatrix} = truncate \left(\begin{bmatrix} S_x & 0 \\ 0 & -S_y \end{bmatrix} \begin{bmatrix} X - X_{min} \\ Y - Y_{min} \end{bmatrix} \right) + \begin{bmatrix} 1 \\ j_{max} + 1 \end{bmatrix}$$

where S_x, S_y = scale factors for X and Y respectively,
X_{min}, Y_{min} = minimum coordinate obtained from the point data set.

Upon obtaining the index (i, j) of the 2D array for *P*, the identifier (id) of *P* is then assigned to this array element (denoted by *A(i, j)*).

$$A(i, j) = id \ of \ P(X, Y)$$

For rasterization in 3D, a set of (x, y, z) coordinates are transformed into a 3D array. The transformation can be done using the formula:

$$\begin{bmatrix} i \\ j \\ k \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_i \\ T_j \\ T_k \end{bmatrix}$$

where i, j, k = indices of the array;
a_{mn} = elements of transformation matrix where $1 \le m \le 3$ and $1 \le n \le 3$;
X, Y, Z = world coordinate tuple;
T_i, T_j, T_k = translation vector.

Similar to the 2D case, given a point *P* with world coordinates (X, Y, Z), denoted by P(X, Y, Z), the first step translates this point to the origin (0, 0, 0). The next step is to apply the scaling and at the same time transform from the right-handed to the left-handed system (which is specific to the 3D array and straightforward for further mapping into a 1D array performed internally by a computer) by applying the negative scaling factor for the Z-component and translating the result by positive k_{max} units, where k_{max} is the size of the k-component of the 3D array. This can be shown in matrix notation as follows:

$$\begin{bmatrix} i \\ j \\ k \end{bmatrix} = truncate\left(\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & -S_z \end{bmatrix} \begin{bmatrix} X-X_{min} \\ Y-Y_{min} \\ Z-Z_{min} \end{bmatrix} \right) + \begin{bmatrix} 1 \\ 1 \\ k_{max}+1 \end{bmatrix}$$

where S_x, S_y, S_z = scale factors for X, Y and Z respectively,
X_{min}, Y_{min}, Z_{min} = minimum coordinate obtained from the point data set.

Having obtained the index (i, j, k) of the 3D array for *P*, the identifier (id) of *P* is then assigned to this array element (denoted by *V(i, j, k)*.

$$V(i, j, k) = id\ of\ P(X, Y, Z)$$

APPENDIX D

Updating Procedure for a 2.5D Simplicial Network

Although it is not the intention of this thesis to elaborate on the updating of a simplicial network which is in the maintenance phase, the approach to the updating of a surface stored in SNIDB is presented in this section to provide the element for integrating 2.5D SNIDB into the 3D FDS presented in section 7.2.

The SNIDB facilitates the simultaneous updating of information about terrain features and terrain relief without disturbing the whole network because of the locality property of the simplicial network. Updating feature data also influences the representation of surface and vice versa. Three different operations are differentiated for each type of features, *ie* insertion, deletion and modification (Pilouk et al 1994).

Updating integrated terrain information means altering the data elements that are contained in the SNIDB. The user can do so through the representation of objects, *ie* point, line and surface features. The geometric component must be under the control of the system or the database administrator via a set of algorithms and procedures provided by the database designer. Integrity rules and constraints should be taken into account during the design of algorithms and be well implemented in the procedures. The system takes the user's action as input, uses this set of procedures to evaluate and decide on the appropriate action and eventually gives feedback to the user. With this kind of constraint, the user's updating actions can be anticipated and differentiated into three major types with respect to each feature: insertion, deletion, and modification. The following subsections elaborate on each of the operations per type of feature. The updating scheme, which is based on the idea of local updating of TIN, is comparable to that presented by Egenhofer et al (1989). Local TIN updating is preferred to global updating for reasons of computational efficiency.

D1 Updating a Point Feature

Altering the geometry of a point feature affects directly its primitive, a node, and indirectly the arcs and triangles associated with this node. We can distinguish three cases:

D1.1 Insertion of a Point Feature

When inserting a new point feature into the database, for example, the user may issue a command like "INSERT POINT A with identifier = 1234, class label = 'Town B', X = 1234.350, Y = 3845.230, Z = 750.245". The point feature set is then the first one that is subject to change. The system needs to evaluate whether the point identifier or these coordinates already exist. If this is not the case, the point can be inserted. Figure D1 shows all three situations that can possibly occur.

Case (a): The coordinates of the new node coincide with those of an existing node (within a predefined tolerance). The insertion attempt is rejected.

Case (b): The node falls on an arc (within a predefined tolerance). The arc is then split into two arcs. Both triangles, to the left and right of the arc, are each also split into two new ones.

Case (c): The node falls inside a triangle. Three major steps are then required:

- 1) adding the node to the node set
- 2) using the three vertices of the triangle and the new node to decompose the triangle into three new smaller triangles. The old triangle is then deleted from the triangle set and the three new triangles have to be added to this set
- 3) adding three new arcs to the arc set taking into account the relationships with the left and right triangles.

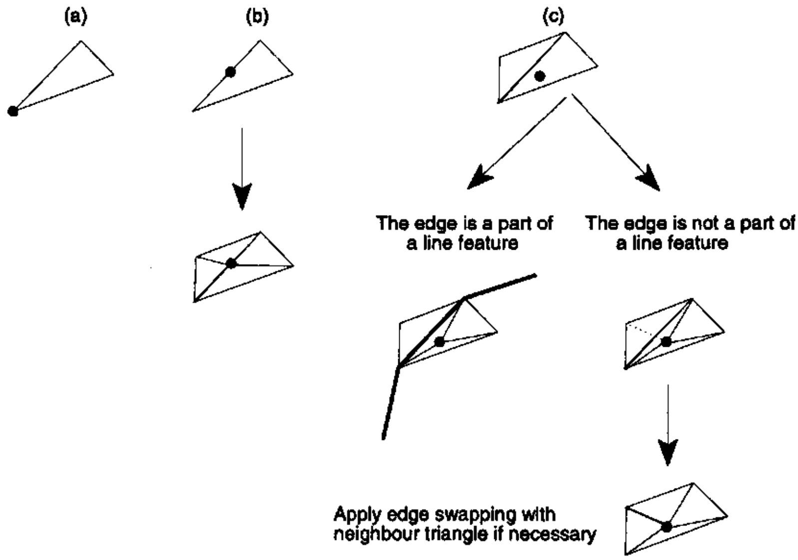


Figure D1 The three cases of node insertion (a) node coincide, (b) node on edge, (c) node in triangle.

For cases (b) and (c), if there is a requirement to follow the Delaunay triangulation, further refinement such as swapping some of the edges between the neighbours of the existing triangles may be necessary. However, when an edge of the existing triangle is a part of a line feature or boundary of a surface feature, this edge must remain unchanged; otherwise, the chain of the linear feature would be destroyed, implying loss of integrity.

D1.2 Deletion of a Point Feature

When the user decides to delete a point feature, it must first be correctly identified. In the relational DBMS, this can be done by using an SQL statement with search criteria (for example, delete where pclass = 'town B') or using a pointing device to pick up the point feature from the displayed graphics. The point feature can be deleted immediately from the point feature set. Then the status of the node associated with this point feature needs to be evaluated. If the node is not a component of any line or surface feature, then this node is subject to deletion upon the decision of the user. The user may consider maintaining this node for the purpose of surface representation. If not, the node is deleted from the node set. Subsequently, all arcs and all triangles linked to this node should be deleted. This results in a polygon which needs to be locally re-triangulated. Figure D2 illustrates this operation.

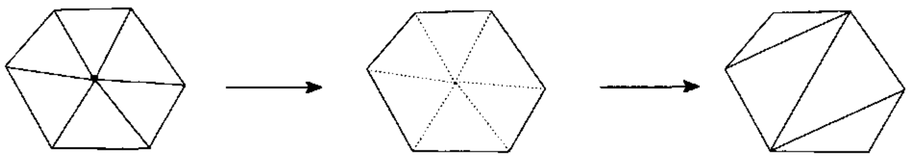


Figure D2 Deletion of a point feature.

D1.3 Modification of a Point Feature

Modification can pertain to a change of the thematic description, to a change of the z-coordinate (elevation) or to a change of (x,y). Changing the thematic description only does not affect the topology, but changing the geometric description may invalidate the topology. Moreover, since the underlying data model is based on 2D topology, a change in the z-coordinate does not affect the topology either. Only the change in planimetry can affect the topology, so re-triangulation may be required. A radical change of the planimetry of the point feature is undesirable and should be avoided because it can cause severe disturbance to both topology and relief representation in a large portion of the network. If a large planimetric discrepancy in the location of the point is found, it is better to delete the original point and insert a new one. Changing the planimetric location of a point feature may be permitted only in the case of a small change and in such a way that the topology is not disturbed. The respective tolerance can be derived from given accuracy specifications of the database.

D2 Updating a Line Feature

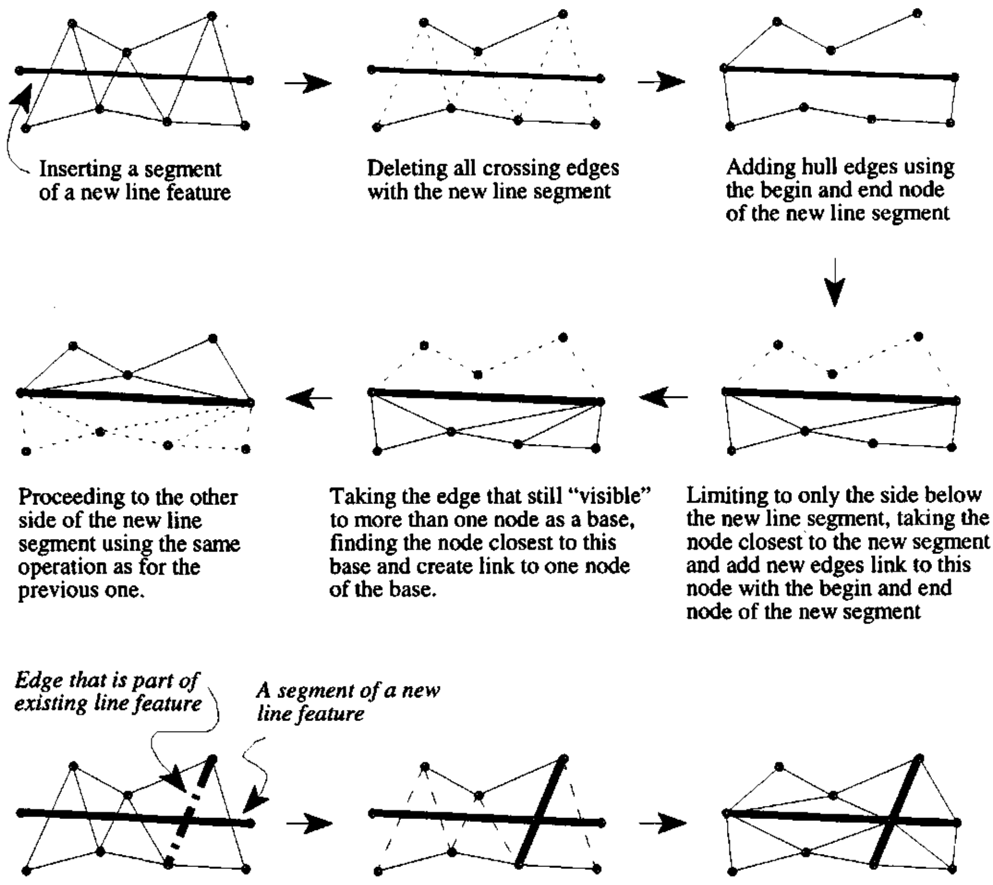
Geometrically changing a line feature affects all arcs and nodes that are components of the line feature. Also, triangles that share these arcs and nodes are inevitably subject to re-triangulation.

D2.1 Insertion of a Line Feature

Insertion of a line feature first requires insertion of all the nodes that constitute the new line. A straightforward procedure is first to insert each node, as described in section D1.1. The algorithm proposed by De Floriani and Puppo (1988) for insertion of a constrained edge, as shown in Figure D3. The procedure is taken for each arc that is part of the new line feature. Two cases must be distinguished during the determination of all existing arcs that cross the new arc such that:

- an arc will be deleted if it is not part of any existing line feature
- if a crossing arc is part of an existing line feature, the point of intersection with the new arc must be taken to decompose these two arcs into four arcs first, as shown in the

lower part of Figure D3. The point of intersection may be treated as a new node to be added into the node set as in the case of insertion of a node on an edge shown in Figure D1.



If a crossing edge is part of an existing line feature, the point of intersection between this edge and the segment of the new line must be taken to decompose these two edges into four edges before processing as normal.

***Figure D3** Insertion of a line feature.*

For the second case, the height at the intersection point must also be determined. This can be regarded as some sort of uncertainty, since heights determined from two different arcs can be different. The average height may be taken if there is no better solution. It should be noted that the local re-triangulation does not necessarily produce optimal triangles according to the Delaunay empty circumcircle criterion.

D2.2 Deletion of a Line Feature

When deleting a line feature, we can distinguish between deleting the thematic and the geometric descriptions. First the identification of the line feature is required, then the line feature is deleted from the line feature set. The next step is to eliminate the relationships between the arcs and a line feature, which means freeing each related arc from geometrically representing this line feature. After this operation, the geometric description remains in the arc set (that is, the vertices of the line are still there, but the linking arcs no longer compose a line feature). If the user also decides to remove the geometric description of the line feature from the SNIDB (for example, if it is not considered necessary for relief representation), this set of arcs should be deleted from the arc set. As a consequence, the set of triangles that are in relationships as left or right of this set of arcs are also deleted from triangle set. Lastly, all nodes that are components of these arcs must be deleted from the node set. This results in a polygon bounded by those arcs which are components of the deleted triangles and leads to a situation comparable to that described in the last part of section D1.2 (see also Figure D4). Thus, re-triangulation has accordingly been achieved.

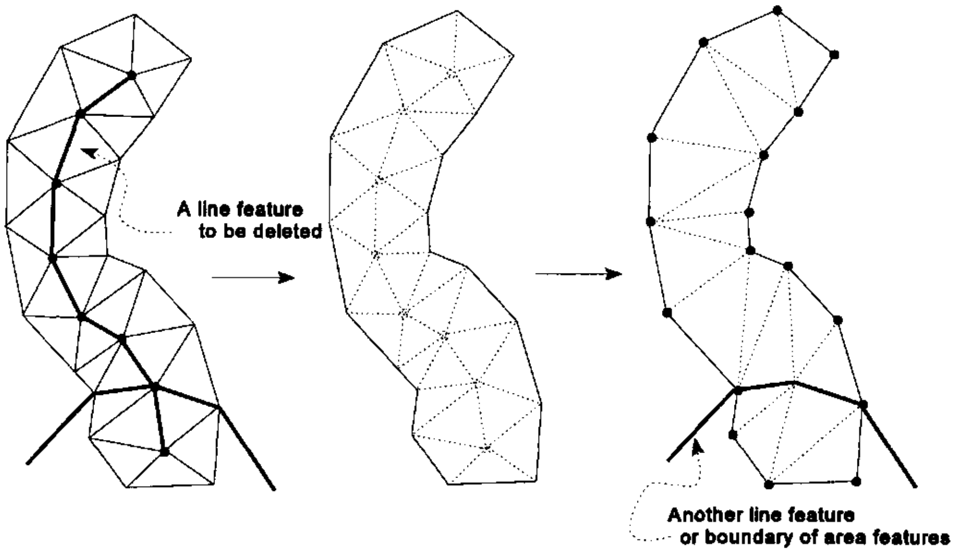


Figure D4 Deletion of a line feature.

D2.3 Modification of a Line Feature

The same considerations hold for modifying a line feature as described for a point feature in section D1.3. Modifying the thematic description does not affect the topology or the relief representation. Thus, it can be done directly in the line feature set. Modifying the geometric description concerns the nodes that are components of the line feature and is carried out according to the explanation in section D1.3.

D3 Updating a Surface Feature

A surface feature is defined by its thematic description and its boundary polygon. The boundary has a chain of arcs and nodes as its components.

D3.1 Insertion of a Surface Feature

After the user has issued a command to insert a surface feature into the database, the surface feature set is first updated. New surface features will be added. Next, all new nodes (and arcs) that are components of the boundary have to be inserted, following the procedure described in section D2.1. The triangle set is the last to be updated; this takes place in such a way that all triangles lying completely inside the boundary of the surface feature have to be reclassified into the new surface feature.

D3.2 Deletion of a Surface Feature

To delete a surface feature, the user needs first to identify the surface feature. The considerations are the same as for the deletion of a point or a line feature. Deleting the thematic description of a surface feature does not disturb the topology. A surface feature can be deleted directly from the surface feature set. Consequently, all triangles that are part of the surface feature have to be reclassified into 'unclassified'. This update is performed on the triangle set. The geometric description remains unchanged. If there is a need to delete the boundary nodes of the surface feature (if they are considered irrelevant to the relief representation), the procedure described in section D2.2 can readily be applied. If the two adjacent areas need to be resolved into one, the common boundary of the two areas need not be deleted; only the reclassification into the new designated class is required. The procedure is described in the next subsection.

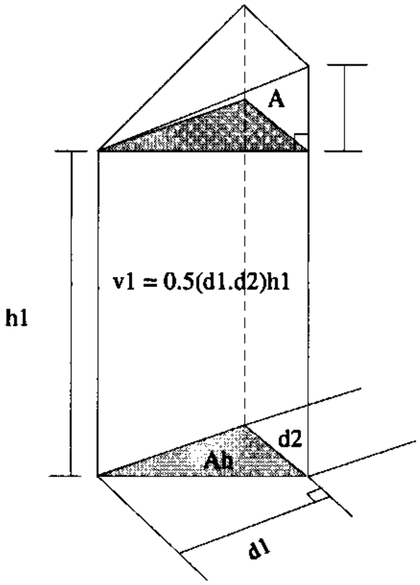
D3.3 Modification of a Surface Feature

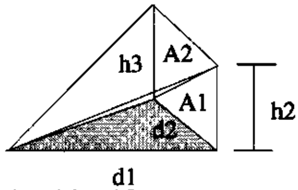
The modification of a surface feature may occur in two aspects; thematic or geometric descriptions. Changing the thematic description is straightforward; only the class name attribute of this surface feature has to be replaced. Modification of the geometric description of a surface feature has the same principles and precautions as outlined for point and line features.

APPENDIX E

Computation of Volume Under a Triangular Surface

Proof of the formula to compute the volume under a triangle with a given datum.





$A = A1 + A2$

$A1 = 0.5(h3.d2)$

$A2 = 0.5(h2.d2)$

$A = 0.5(h2+h3)d2$

$A2 = 0.5h2d2+0.5h3.d2-0.5h3.d2$

$A2 = 0.5h2.d2$

$V2 = A2.d1 / 3$

$V3 = A1.d1 / 3$

$V4 = V2+V3$

$V4 = d1(A1+A2)/3$

$V4 = d1A/3=0.5(h2+h3)d1d2/3$

$V4 = (h2+h3)d1d2/6$

$V = V1+V4$

$V = 0.5d1d2h1+d1d2h2/6+d1d2h3/6$

$V = 0.5d1d2(3h1+h2+h3)/3$

$V = 0.5d1d2(h1+(h1+h2)+(h1+h3))/3$

$Ah = 0.5d1d2$

$V = Ah(h1+(h1+h2)+(h1+h3))/3$

Conclusion: Volume under a triangle = height at centroid * area of the horizontal triangle

The formula is given in the above figure.

Examples of Scenes from ISNAP and a Virtual Reality Browser

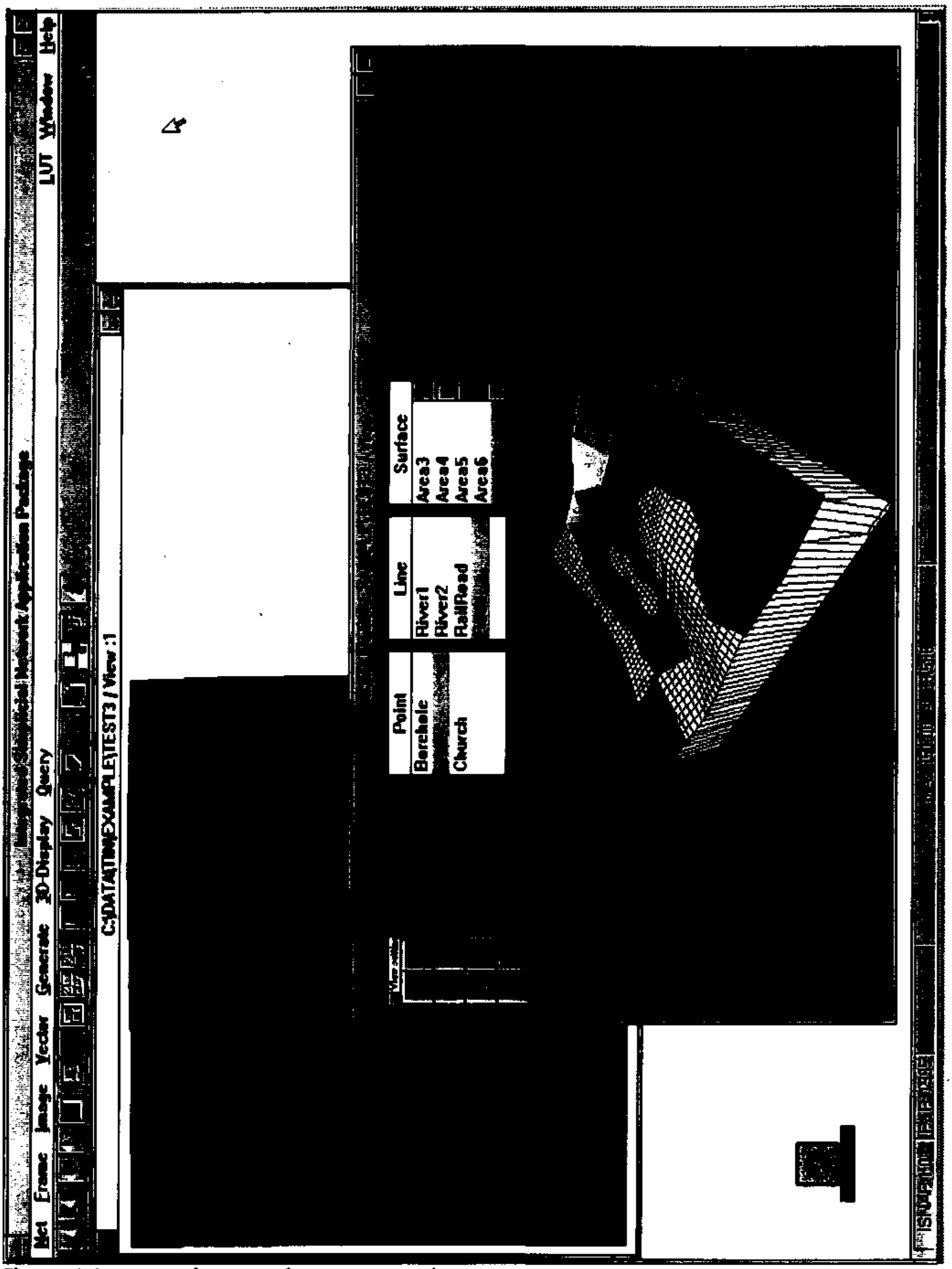


Figure 1 An example scene from ISNAP software.

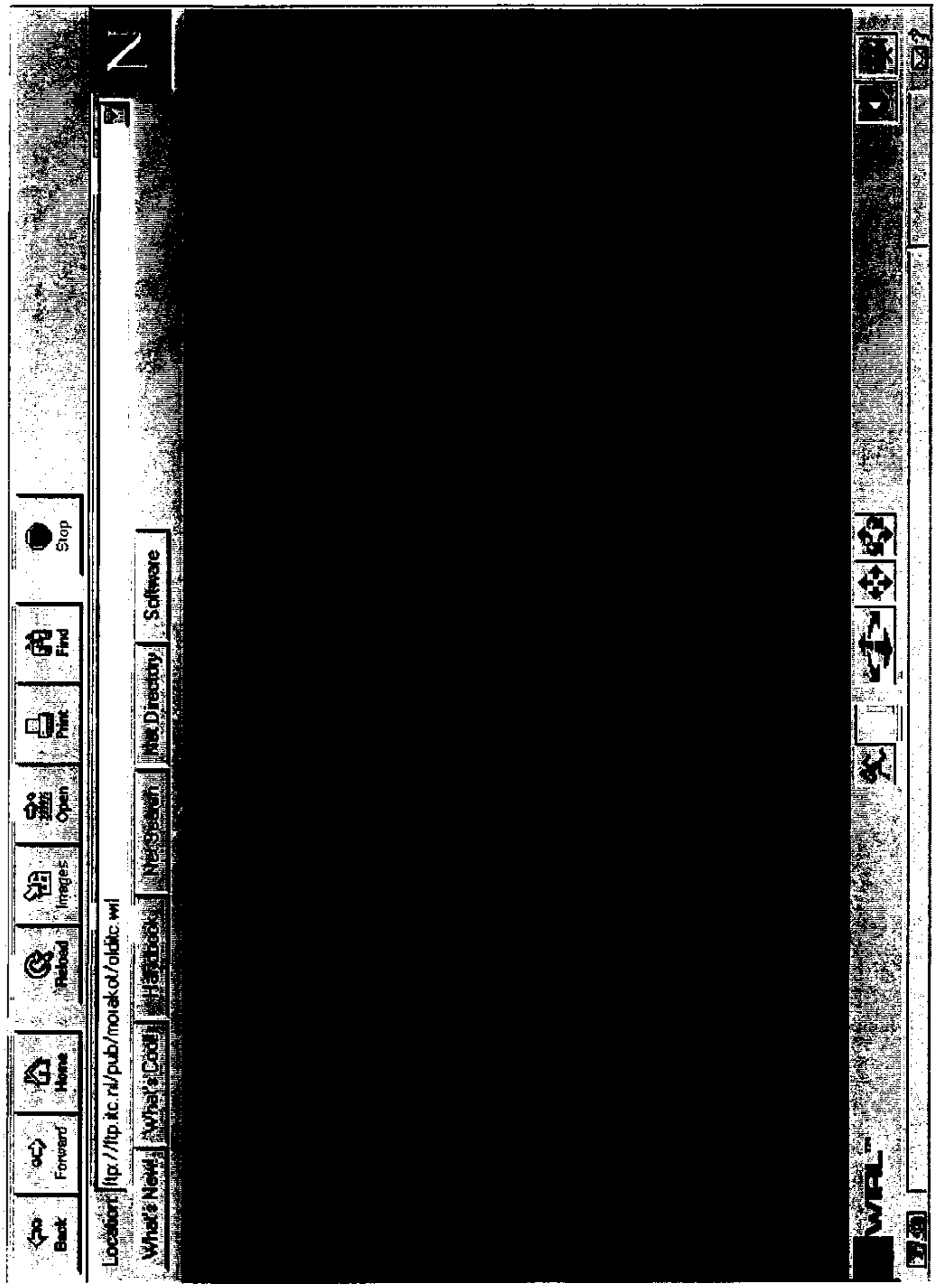


***Figure F2** An example of scene taken from a virtual reality browser.*

CURRICULUM VITAE

Morakot Pilouk, born on 18 January 1964, received his B.Eng. degree in irrigation engineering from Kasetsart University, Bangkok, Thailand, in 1985. From 1985 to 1989, he worked for an engineering consulting and construction company as a project engineer in construction projects, and as a civil engineer in the design and construction section of the Royal Forest Department of Thailand. In August 1989, he became a student at the International Institute for Aerospace Survey and Earth Sciences (ITC), Enschede, The Netherlands where he graduated from the postgraduate diploma course in photogrammetry with distinction, in August 1990. In January 1992, he received the MSc degree in integrated map and geoinformation production at ITC. This degree was also awarded with distinction. In October 1992, he started his PhD research project reported in this thesis. The research was carried out in cooperation with the department of Geographic Information Processing and Remote Sensing, Wageningen Agricultural University, The Netherlands, and funded by ITC. He is the author and coauthor of many scientific articles. He is married to Pakrairat and has two children, Pakawat and Patriya.