Realising the Foundations of a Higher Dimensional GIS: A Study of Higher Dimensional Spatial Data Models, Data Structures and Operations

PhD Research Proposal

G.A.K. Arroyo Ohori, MSc

GISt Report No. 57

GIS TECHNOLOGY

TU Delft

OTB Research Institute for Housing, Urban and Mobility Studies

Delft University of Technology

Realising the Foundations of a Higher Dimensional GIS: A Study of Higher Dimensional Spatial Data Models, Data Structures and Operations

PhD Research Proposal

G.A.K. Arroyo Ohori, MSc

GISt Report No. 57

Summary

Geographic Information Systems have had enormous advances since their inception. While early systems were mostly limited in scope to automating map making and simple analyses of two dimensional map data, modern systems enable complex spatial analyses using a variety of data sources and techniques, including spatial statistics and interactive visualisation.

Nevertheless, GIS are still relatively weak as far as the third dimension is concerned, with most systems being only able to support limited 3D storage and visualisation with the help of external modules. This lack of support is not due to an absence of applications for higher dimensional GIS. While being limited to two dimensions is acceptable to many users of geographic information, 3D GIS are highly desirable for many others. Powerful new insights can be generated with the integration of additional dimensions, such as those originating from time, scale, and other application-dependent feature spaces.

At the same time as GIS were mostly confined to 2D representations, the techniques to create and manipulate spatial data representations were developed much further within other fields, namely computer vision, computer graphics, CAD and CAM. This included the mathematical formulations to provide them with better foundations, and their extension to higher dimensions. Thus, 3D support became very widespread in these fields, to the point that is has become a robust and integral part of them.

The purpose of this research is therefore to take advantage of the developments in the modelling of spatial data that have been made within these fields, and apply that knowledge to the specific needs of spatial information science. This will provide a link between the top-down approach of GIS and the bottom-up one of computer science, and integrate their results with regards to n-dimensional data modelling. This involves realising the adaptations to these data models and data structures, and creating the operations required for their use in GIS.

ISBN: 978-90-77029-29-9
ISSN: 1560-024

OTB Research Institute for the Built Environment
TU Delft
Jaffalaan 9, 2628 BX, Delft, The Netherlands
Tel.: +31 (0)15 278 4548; Fax +31 (0)15-278 2745
Websites: http://www.otb.tudelft.nl/
http://www.gdmc.nl/
Email: GISt@TUDelft.nl

Contents

Acronyms

0D	zero dimensional
1D	one dimensional
2D	two dimensional
3D	three dimensional
4D	four dimensional
ACM	Association for Computing Machinery
AGILE	Association of Geographic Information Laboratories for Europe
AHD	Alternate Hierarchical Decomposition
BSP	Binary Space Partitioning
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CDT	Constrained Delaunay Triangulation
CGAL	Computational Geometry Algorithms Library
CGIS	Canada Geographic Information System
CRS	Coordinate and Reference System
CSG	Constructive Solid Geometry
DBMS	Database Management System
DCEL	Doubly Connected Edge List
ER	Entity-Relationship

ESRI	Environmental Systems Research Institute
FEM	Finite Method Model
GIS	Geographic Information System
IEEE	Institute of Electrical and Electronics Engineers
ISO/TC	International Organization for Standardisation Technical Committee
LoD	Level of Detail
MADALGO	Center for Massive Data Algorithmics
OGC	Open Geospatial Consortium
OO	Object-Oriented
OR	Object-Relationship
PCA	Principal Component Analysis
PLC	Piecewise Linear Complex
PSLG	Planar Straight Line Graph
STC	Space-Time Composite
STER	Spatio-Temporal Entity Relationship
TEN	Tetrahedral Network
tGAP	Topographical Generalised Area Partitioning

Chapter 1

Introduction

1.1 Motivation

Geographic Information Systems (GIS) have had enormous advances since their inception. While early systems were mostly limited in scope to automating map making and simple analyses of two dimensional (2D) map data [Coppock and Rhind, 1991], modern systems enable complex spatial analyses using a variety of data sources and techniques, including spatial statistics and interactive visualisation.

However, many of the spatial data models that originated in cartography, and were then introduced in digital form, still form the basis of GIS. Among those, we are mostly concerned with *boundary representation*, a type of vector representation in which well-defined regions are identified and represented by their limits. This is a very intuitive concept that allows for a simple and clear definition of the boundaries of an area, but is also powerful, flexible and allows for a rich set of operations.

This concept was already used in both the oldest surviving world map, the Babylonian Map of the World (700-500 BCE), and the oldest topographic map (1160 BCE), the Turin Papyrus Map (see Figure 1.1). This kind of representation is common in many types of maps, and is especially suited to those in which boundaries can be unambiguously defined, such as those that represent subdivisions of a territory (e.g. choropleth maps [Dupin, 1826]).

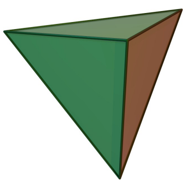
While these early maps were limited to representations of 2D space, three dimensional (3D) objects have also been known and studied since antiquity. The five Platonic solids (see Figure 1.2) were already discussed at length in Plato's Timaeus [Jowett, 1998] in ca. 360 BCE, being considered the basic elements of matter. These already show an understanding of how an object can be defined by its boundaries, composed of regular and well-defined objects of a lower dimension.

During the 1960s and 1970s the first GIS were developed, already using boundary representation to depict 2D areas, such as SYMAP [Chrisman, 1988] and the Canada Geographic Information System (CGIS) [Tomlinson, 1988]. However, despite the numerous technological

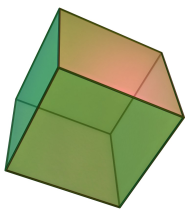
(a) The Babylonian Map of the World (700-500 BCE) depicts the known world from the perspective of Babylon [Finkel, 1995]. © Trustees of the British Museum.

(b) The Turin Papyrus Map (ca. 1160 BCE) shows different mineral deposits and rock types available in a region along the Nile river [Harrell and Brown, 1992]. Photograph by James E. Harrell, retrieved from Wikimedia Commons.

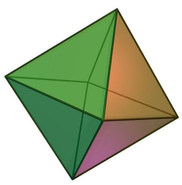
Figure 1.1: The use of boundary representations was already used in some of the most ancient maps.



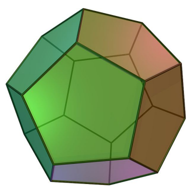
(a) Tetrahedron



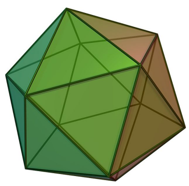
(b) Cube (hexahedron)



(c) Octahedron



(d) Dodecahedron



(e) Icosahedron

Figure 1.2: The five Platonic solids. From Wikimedia Commons.

breakthroughs that have been made since then and the fact that GIS were early adopters of this representation, they remained limited to 2D representations for an unseemly long time. Even today, GIS are still relatively weak as far as the third dimension is concerned [Gold, 2006; Schön et al., 2009; Stoter and Zlatanova, 2003], with most systems being only able to support limited 3D storage, and 3D visualisation with the help of external modules [Döllner and Hinrichs, 2000]. Meanwhile, true 3D GIS solutions are notably complex (see Zhang et al. [2011]) and face a few difficulties that still need to be overcome [Abdul-Rahman and Pilouk, 2008], such as the lack of standards in the implementation of data models (e.g. ISO 19107), as well as in data structures, operations (see Tet-Khuan et al. [2007]) and formats.

However, this slow progress is not due to a lack of applications for higher dimensional GIS. While being limited to two dimensions is acceptable to many users of geographic information, 3D GIS are highly desirable for many others. Powerful new insights can be generated with the integration of additional dimensions [Raper, 2000], such as those originating from time, scale, and other application-dependent feature spaces.

Nevertheless, as GIS were mostly confined to 2D representations, the techniques for boundary representation were developed much further within other fields, namely computer vision, computer graphics (see Kalay [2004]), Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM). This included the mathematical formulations to provide them with better foundations, and their extension to higher dimensions. Thus, 3D support became very widespread in these fields, to the point that is has become a robust and integral part of them (see Lee [1999] or Shreiner [2009]).

Since there is both a need for higher dimensional GIS, and an availability of representations developed in other fields that are able to support higher dimensional data (see Section 3.1), there is great potential in adapting these representations to the specific needs of GIS data (e.g. support for overlapping regions, holes, and complex handling of attributes and metadata) and providing the specific operations that are required for its use, such as buffering and overlays (see Albrecht [1995]). In fact, *the approaches of computer science and geographic information science to this topic can be considered as opposite and mutually complementary*. The former is bottom-up and has developed a solid mathematical foundation and much more advanced methods, while the latter is top-down and has clear requirements and applications.

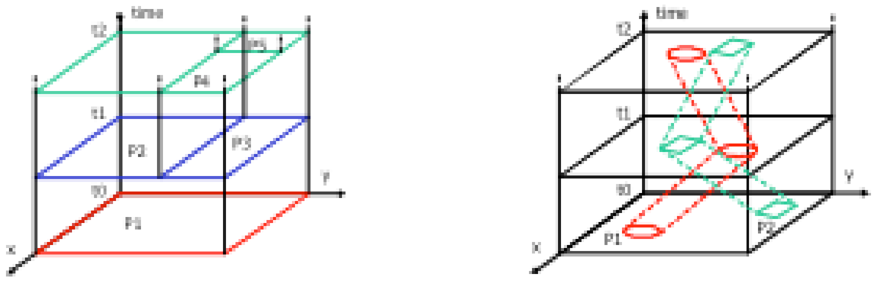
The purpose of this research is therefore to take advantage of the developments in the modelling of spatial data that have been made within the fields of computer science, computer graphics and CAD, and apply that knowledge to the specific needs of spatial information science. This will provide a link between the top-down approach of GIS and the bottom-up one of computer science, and integrate their results with regards to n-dimensional data modelling. This involves realising the adaptations to these data models and data structures, and creating the operations required for their use in GIS.

1.2 Integrated Modelling of Data in Higher Dimensions

Previously, the integration of other dimensions into a higher dimensional GIS has been mentioned. The role of spatial dimensions in this matter is clear, as they are used in a direct fashion. However, many non-spatial dimensions can be integrated and treated as spatial. Within these, time and scale are analysed within this section. Additionally, some general considerations regarding the modelling of application-dependent feature spaces are given as well.

1.2.1 Time

Among the possible other non-spatial features that can be integrated into higher dimensional GIS, time in particular has long been considered to be interlinked with space [Akhundov, 1986], and several representations for it have been developed [Peuquet, 2002]. In fact, most 2D maps can be considered to be slices of 3D space-time (usually at a predefined time), as shown in Figure 1.3.



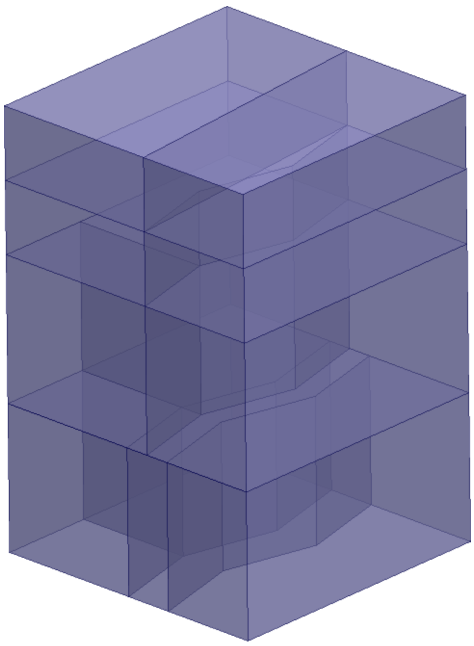
(a) The subdivision of cadastral parcels in time (b) Moving objects

Figure 1.3: Visualising time as a third dimension. From van Oosterom et al. [2006].

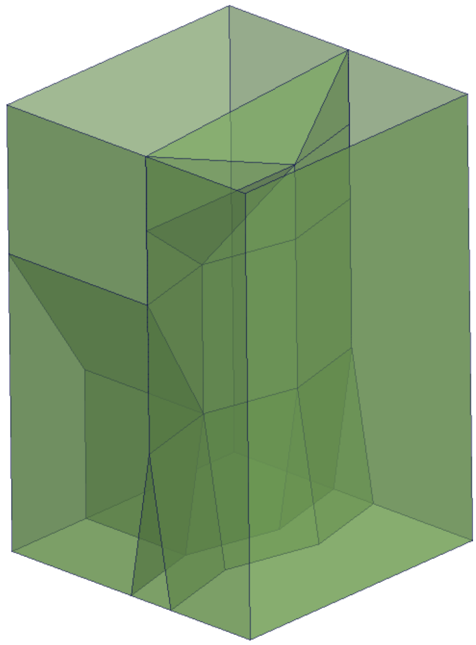
However, the question of whether time should be modelled as an additional dimension has been quite controversial within the field of geographic information science. Notable authors have had opposing views on the subject, and no consensus has been reached. While [Peuquet, 1994] argues that there are substantial differences in the manner in which space and time are modelled and a unified four-dimensional representation of space and time is not sufficient for GIS, Worboys [1994] calls for the unification of the temporal and spatial dimensions, in a model in which transaction and valid time are seen as two additional orthogonal dimensions.

1.2.2 Scale

Meanwhile, scale, despite being more of an artificial construction to define a level of detail in a map, is considered by some to be inseparable from time in the representational process due to the fact that events are only relevant at a certain spatial and temporal resolution [Raper, 2000], and can be well translated into an additional dimension due to the concept of multi-scale representations [Frank and Timpf, 1994], in which the same object or its attributes are stored at different scales, either explicitly or implicitly (vario-scale [van Oosterom and Meijers, 2011]). This is shown in Figure 1.4. Since multi-scale representations require links between the same object at different scales, a procedure to create less detailed maps from more detailed ones is required. This is made possible by the notion of (automatic) generalisation (see Weibel [1997]), which provides a set of tools and techniques to simplify different features of a map.



(a) A multi-scale representation has inter-linked objects defined at multiple scale levels.



(b) A vario-scale representation has no fixed scale levels.

Figure 1.4: Visualising scale as a third dimension. From van Oosterom and Meijers [2011].

Even so, similar arguments can be made for the exclusion of scale as an additional dimension. While space and time are continuous dimensions with well-defined units, the situation is less clear with scale in digital systems, and simple operations in 5D space-time-scale often do not lend themselves to an intuitive definition. For instance, the distance between two 5D points is hard to quantify, since the units along different axes are different. However, others like Li [1994] argue that the concept of scale is inherent in the understanding of reality, and as such, it is only natural to represent it in a 5D space-time-scale system. One way to see this is to consider geographically referenced 3D events in 2D time-scale.

1.2.3 Feature spaces

Additionally, one might consider the inclusion of other, application dependent dimensions. Some data sets are produced in a manner in which the most natural representation is multidimensional, such as sensor networks, financial transactions, and biomedical remote sensing [Schultz et al., 2001]. Remote sensing data, common in GIS, frequently uses hyperspectral images, in which a received signal strength at different frequencies is interpreted as different dimensions of an image. Thus, incorporating these additional *feature space* dimensions into a data model can enable more and better analyses of the data. Similar concepts are commonplace in other fields and provide great insights. In physics, an n-dimensional configuration space can be used to represent an entire system of n particles as a point in n-dimensional space, while in mathematics an n-dimensional parametric space describes the values of a set of parameters in a particular mathematical model.

Despite the fact that the examples mentioned above generally consist of point data, it is often desirable to analyse it in a space filling representation (e.g. by creating Voronoi regions), or has an explicit structure linking these points together (e.g. a triangulation), which makes a higher dimensional representation more practical.

1.2.4 Advantages of the integrated approach

Nevertheless and despite the arguments for or against the inclusion of other characteristics as additional dimensions of a data model, there are very practical advantages innate in choosing such an approach. Some complex time-scale-space operations can be expressed quite simply and elegantly in higher dimensional spaces. As an example, imagine a virtual globe application (e.g. Google Earth) in which we are looking at a realistic 3D city model. This is simply a perspective projection of such a city model into a 2D plane. However, since computational power is limited and such a model might be quite complex, in order to improve the performance of our application, buildings far away from our current viewpoint should be displayed in less detail than those close to us, which is equivalent to first selecting a 3D space slice at an appropriate distance dependent scale from the original four dimensional (4D) space-scale model and then projecting it into a 2D plane.

Other important advantages of the integrated approach include: avoiding inconsistencies in data (e.g. overlaps and gaps), achieving more efficient storage and indexing, having the possibility to perform more advanced topological queries, and taking non orthogonal slices of the data (e.g. in vario-scale representations).

1.3 Structure Overview

The structure of this project plan is as follows:

- Chapter 2 discusses the most relevant *Related Work in Geographic Information Systems* that tackles the problem of modelling multi-dimensional information, namely the specific cases of spatio-temporal modelling and multi-scale modelling. While these are only specific cases of spatial models and reflect the idiosyncrasies of time and scale, there are substantial advances made in modelling the specific cases of time and scale in GIS, from which we can learn how to create the data models, data structures and operations needed to support generic additional dimensions with support for GIS data.

- Chapter 3 similarly discusses the relevant *Related work in Computer Science, Computer Graphics and Other Fields*, which is related to higher dimensional data models and data structures, which will be used as a foundation for the project. Also, it discusses the types of operations that should be supported in these so that they can be used for GIS data.

- Chapter 4 describes the specific *PhD Research* that I will perform, including the objectives of my PhD research, the methodology for achieving it and how my time will be spent during the coming years.

- Chapter 5 enumerates the *Practical Aspects* related to this PhD proposal, such as the courses that I will take, lists of relevant journals and conferences and how the project will be supervised.

Chapter 2

Related Work in Geographic Information Systems

Despite the fact that Geographic Information Systems have remained largely two dimensional to this date, important research regarding higher dimensional information has been done within the field of geographic information science (mostly limited to 3D), using the characteristics that are most commonly related to space, namely time and scale. This can be seen as the general trend in GIS of analysing problems in a top-down perspective. Since there is a need to include time and scale within GIS, specific data models and data structures have been developed for this purpose. This can be contrasted with the opposite approach in other fields, as will be discussed in the next chapter.

Therefore, the work done on 2D and 3D modelling in GIS is covered in Section 2.1, and the specific cases of spatio-temporal modelling and multi-scale modelling are discussed in Sections 2.2 and 2.3. While these are only two particular examples of characteristics that might be incorporated with spatial data, they are especially important, since they are the ones most commonly used in conjunction with geographic information, and have therefore been extensively studied within the field of GIS. This entails that the data models, data structures and operations developed within these two have already been adapted to the specific needs of GIS data sets.

2.1 2D and 3D Modelling in GIS

Within the context of GIS, the Open Geospatial Consortium (OGC) and the International Organization for Standardisation Technical Committee (ISO/TC) 211 have established standards for 2D and 3D spatial information, such as the Simple Features specification [OGC, 2010a,b], and the ISO 19107 standard [ISO, 2003]. Additionally, they have supported other standards based on specific file formats, such as KML [OGC, 2008b], GML [OGC, 2007], CityGML [OGC,

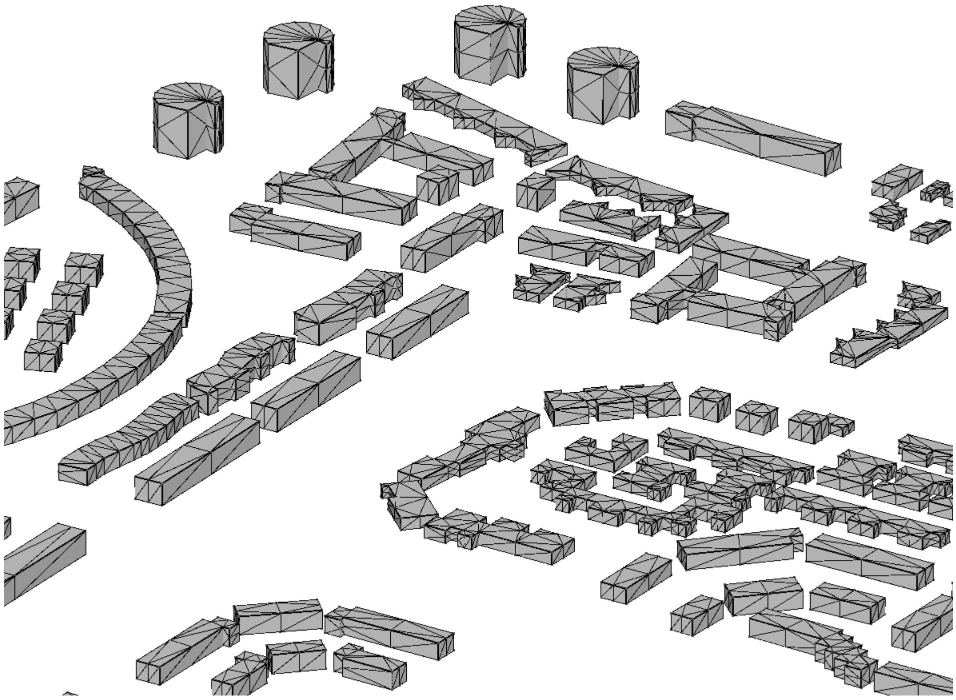


Figure 2.1: A Tetrahedral Network (TEN) model of part of the city of Rotterdam. From Penninga [2008].

2008a]. Meanwhile, GIS companies have developed and standardised their own formats, notably ESRI's Shapefiles [ESRI, 1998]. These work as data models in GIS, providing a clear description of the object modelling process and some implementation details.

Some research has been done in order to define 3D structures in GIS, which can use polyhedra [Arens et al., 2005; Pigot, 1991], regular polytopes [Thompson, 2007] or Tetrahedral Networks (TENs) [Penninga, 2008] (see Figure 2.1). However, further work is required in order to create standards that establish more advanced 3D geometries, topological primitives, relations and how to enforce the validity of such data.

2.2 Spatio-temporal Modelling

Spatio-temporal modelling concerns the creation of joint models combining spatial and temporal information, which combine knowledge related to the independent modelling of both space and time. Regarding space, it is important to consider factors such as the possibility of: raster or vector modelling, using different Coordinate and Reference Systems (CRSs), specifying orientation and direction, measuring objects, and keeping track of topology. Similarly, in temporal systems other important factors appear, such as the possibilities of: having differing temporal granularities, keeping discrete or continuous time, supporting uniform or irregular changes, keeping both transaction and valid time [Snodgrass and Ahn, 1985], modelling directed or cyclical time, and the importance of keeping historical data.

When combining these factors, and in the joint modelling of spatial and temporal information, it is important to take into account the capabilities of different models to answer spatial, temporal and spatio-temporal queries. In order to do this, a new set of factors unique to spatio-temporal modelling arise, such as the ability to keep topology in space-time, and the modelling of objects that are continuously moving and changing. A more thorough survey of spatio-temporal database models, analysed with factors such the the ones mentioned, is available in Pelekis et al. [2004].

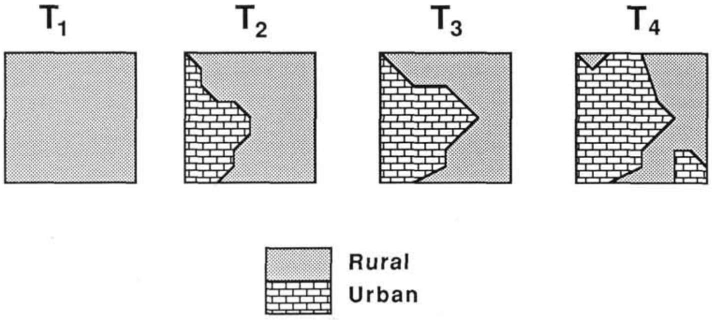
Some models use the spatial features as a base to which temporal information is attached. For instance, in the snapshot model (Figure 2.2a), temporally homogeneous sets of objects are modelled with timestamps to indicate the interval in which that particular set of objects and related attributes existed. The first example of this seems to be in the US Historical Boundary File [Basoglu and Morrison, 1978]. It is simple, but not very powerful, mainly because there are no direct relations between temporal objects. A variation of this involves storing differential changes only, as mentioned in Langran and Chrisman [1988], or time-stamping individual features with both creation and cessation time [Hunter and Williamson, 1990]. Another related possibility is keeping the current state of the map as well, which improves the query time.

In the Space-Time Composite (STC) model, regions are split into a polygonal mesh where each object shares the same attribute history. It was first described in Chrisman [1983], based on Peucker and Chrisman [1975]. It is more flexible than the snapshot model, but objects can become very fragmented, leading to difficulties when performing some operations. For instance, updating the attributes of a polygon might involve updating all the regions that the polygon is split into.

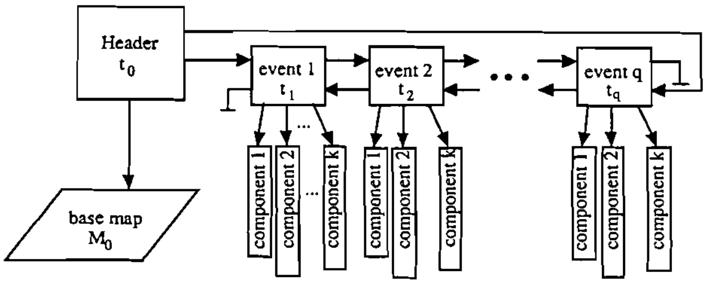
Other models are based on events instead [Peuquet and Duan, 1995] (Figure 2.2b) and maintain a list of events and a base map, with each event being linked to all changes that occurred since the last event. This makes it possible to identify individual changes and events. In the *history graph model* [Renolen, 1996], different types of events are supported, which makes it possible to model continuously changing events as well.

A different option is to keep track of space, time and semantics independently, and link objects appropriately. So called *three-domain models* are based on this concept. Examples include Yuan [1994] and Claramunt and Thériault [1995]. van Oosterom [1997] uses an identifier consisting of both a region id and time to index spatio-temporal objects.

Finally, there are some generic spatio-temporal models described at the conceptual level [Story and Worboys, 1995], which can be adapted to suit a specific application. For instance, Tryfona and Jensen [1999] describes the Spatio-Temporal Entity Relationship (STER) model, based on the Entity-Relationship (ER) model [Chen, 1976] common in the database world. It provides rudimentary support for multi-scale objects by allowing for multiple geometries to be stored in each feature. Claramunt et al. [1999], discusses an Object-Relationship (OR) model, specif-



(a) The snapshot model considers homogenous regions at a certain time, to which attributes are attached. From Langran and Chrisman [1988].



(b) Event-based models maintain a list of events, to which regions and/or changes are attached. From Peuquet and Duan [1995].

Figure 2.2: Two types of spatio-temporal models.

ically tailored to model change. Additionally, Object-Oriented (OO) models exist [Worboys et al., 1990].

2.3 Multi-scale Modelling

Multi-scale modelling can be considered as a specific instance within the topic of multiple representation [Friis-Christensen and Jensen, 2003], in which the same geographic entity or its attributes are stored more than once, either at fixed scale levels or continuous ones (i.e. vario-scale, see van Oosterom [1990]). The main problem regarding multiple representation is the fact that inconsistencies can easily arise between different representations of the same data. In multi-scale modelling, each representation is equivalent to an object at a certain scale level, which are equivalent to Levels of Detail (LoDs) in the digital world.

The data at these different scales is usually generated based on more detailed data using semi-automated processes [van Oosterom, 2009]. Here, the automation is crucial in order to ensure consistency between different scales (e.g. in web mapping [Cecconi, 2003]), and thus involves standardised procedures, which are known as generalisation methods. The specific operations that need to be done vary radically according the approach used. For instance, Weibel [1997] proposes object selection, line simplification [Douglas and Peucker, 1973], line smoothing, line segmentation and terrain generalisation as a base.

Despite the extensive research work, no tools for automatic generalisation in GIS were available until recently [Jones and Ware, 2005]. Since then, some have been developed, such as in ArcGIS 10 or 1Spatial Clarity, but still suffer from problems like the generation of topological errors or the loss of the links between generalised and non generalised objects (see Stoter et al. [2009]).

In addition to the availability of generalisation tools, it is also important to consider the development of smart data structures that are able to keep track of different scale levels in a consistent manner. This usually involves a hierarchical tree-like structure that maintains links between the same object at different scales [Frank and Timpf, 1994]. However, these links can represent several different generalisation operators, which can have important consequences at the semantic level (see Stoter et al. [2011]).

A few data structures to enable variable scale models have been developed. In particular, the Topographical Generalised Area Partitioning (tGAP) data structure [van Oosterom, 2005, 2006; van Oosterom and Meijers, 2011] is capable of storing multiple levels of detail implicitly using only a single object, making use of reactive data structures consisting of trees of generalised edges and faces to dynamically generate 2D maps based on a slicing operation, supporting smooth zooming and progressive transfer.

Chapter 3

Related work in Computer Science, Computer Graphics and Other Fields

As discussed in the previous chapter, there are quite a few existing models in GIS that are able to deal with spatio-temporal and multi-scale information. However, they are specifically tailored to the cases of time and scale, and therefore their ideas are not directly applicable to other dimensions. This certainly comes at a contrast with the approach adopted in computer science, computer graphics and CAD/CAM, which have focused on generating *generic models* in a mathematical sense, that are applicable to any type of dimension, but are usually applied to 3D space.

This has both advantages and disadvantages. While GIS models are more adapted to time and scale, they are much harder to adapt to other types of characteristics. On the other hand, models from computer science and computer graphics have a solid mathematical background and can be adapted to model all sort of different characteristics. Since for this research I aim to make extensible and adaptable data models and data structures, I will take advantage of the more advanced models that have been developed on the latter, while adapting them to the needs of GIS.

Frank [1992] discusses the existence of three distinct concepts in GIS: *spatial concepts*, that denotes the terms used by humans to understand space; *geometric data models* describe a set of abstract spatial object classes and their related operations; and *geometric data structures* are the specific implementations of a geometric data model, with certain storage structures, uses and performance. Since the purpose of this PhD research is to create spatial data models and data structures that are fit for use within GIS, the last two are of relevance to my work, and reviews of promising data models and data structures for the modelling of an arbitrary number (n) of dimensions are discussed in the following sections.

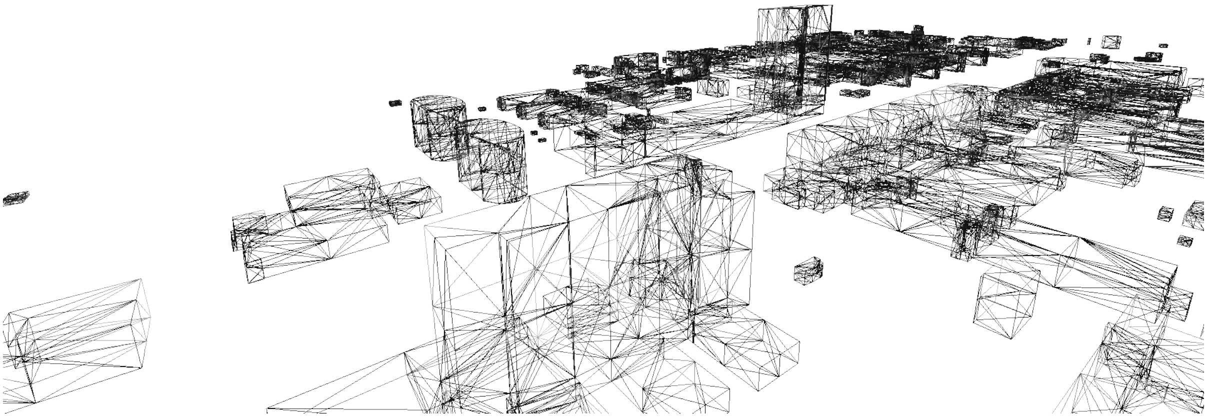


Figure 3.1: A wireframe model of the TU Delft campus. This type of model stores only the edges of an object, as surfaces and volumes are not explicitly described.

While in most contexts the terms data model and data structure are being used more or less interchangeably, they are very different within the topic of n-dimensional modelling. This is due to the fact that a data model and a data structure based upon it might have an support for a differing number of dimensions. For instance, a wireframe model (Figure 3.1) is composed of zero dimensional (0D) and one dimensional (1D) primitives embedded a higher dimensional space, but has no inherent limit on the number of dimensions it supports. Nevertheless, a data structure implementing it might be limited to three coordinates per point.

Moreover, as shown in Table 3.1, I have refined the classification from Frank [1992] to distinguish two different aspects of both data models and data structures: a *spatial definition* describing the placement and constraints of objects in space, and a *spatial connection* that states how these objects are put together. While the former is always defined, the latter can be implicit or nonexistent (e.g. a set of half-spaces representing a convex polyhedron). Making this additional distinction in classification is very useful in n-dimensional modelling, since every one of these aspects can also have a different dimensionality.

Following the aforementioned classification, within a data model there are two aspects which can be clearly distinguished: how the discretisation of space is defined (e.g. a set of points), and how they are joined together (e.g. a Bézier surface). While they are interlinked, it is often possible to change one without changing the other, as long as no fundamental features of the model are broken. For instance, consider a tuple of points joined by a plane passing through them, forming a convex polygon. If the points are no longer convex, the points can

Table 3.1: Subdivision of data models and data structures based on Frank [1992].

	Spatial definition	Spatial connection
Data model	Discretisation	Interpolation
Data structure	Embedding	Topology

still be joined (as long as they are coplanar). Likewise, they could be joined with spline curves without changing their location or order.

Similarly, we can differentiate between two different characteristics of a data structure: the topological relations that it stores and the space in which it is embedded. This distinction is sometimes not clearly made in literature, but it is important to do so since the dimensionality of the two is independent. In a common example, the winged-edge data structure [Baumgart, 1975] is often considered to be a data structure with 3D support. However, in reality, it is only capable of representing two dimensional subdivisions or 2D surfaces in higher dimensional objects. It is not able to explicitly store relations between different polyhedral objects, and thus it cannot be considered a 3D data structure.

The sections within this chapter are thus respectively intended to analyse the existing data models (Section 3.1) and data structures (Section 3.2), analysing further those that are deemed more promising with regards to their capabilities to represent n-dimensional objects. Furthermore, some potential important operations for n-dimensional data are listed in Section 3.3.

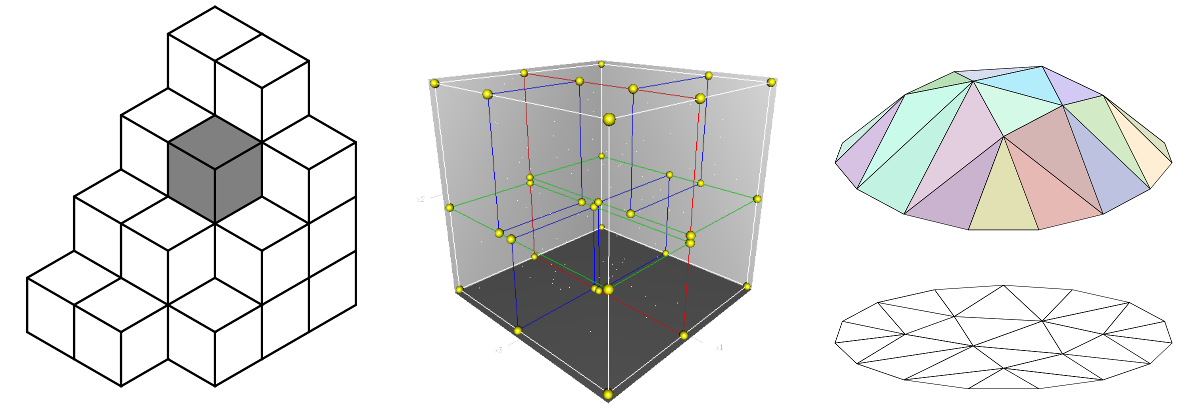
3.1 n-Dimensional Modelling

Spatial data models can be classified into different types based on their characteristics. The following classification is based on Mäntylä [1988] for decomposition and constructive models, and Lienhardt [1991] for boundary models:

Decomposition models

These types of models describe objects as a combination of basic elements put together. Among these, we can distinguish 3 techniques to perform the division of space, which are shown in Figure 3.2 and described as follows:

- **Exhaustive enumeration** representations, commonly known as *rasters*, are the most common decomposition model. They divide space into an n-dimensional grid, with objects being approximately described as grid elements that are contained in them, together with some attached characteristics. A digital image is an example of a raster representation. Since it is simple to establish an order of the elements in the grid, it is not necessary to explicitly store topology in this model. This ensures that models are always valid, unique and unambiguous. However, raster representations also have important limitations. Objects need to be approximated and



(a) Exhaustive enumeration: rasters use boxes that fit an n-dimensional grid.

(b) Space subdivision: a k-d tree recursively subdivides space in halves along different dimensions.

(c) Cell decomposition: a mesh that uses differently shaped elements.

Figure 3.2: The different types of decomposition models. From Wikimedia Commons.

the representation is highly inefficient (and gets more inefficient in higher dimensions).

- **Space subdivision** models follow a similar approach. However, instead of dividing space into boxes of a known size, space is recursively subdivided along different dimensions to yield boxes of heterogeneous sizes. This means that space subdivision can adapt to the detail present in the original data. The data structures meant to achieve this are known as Binary Space Partitioning (BSP) trees. The most common ones are quadtrees [Finkel and Bentley, 1974] and their higher dimensional analogues (e.g. octrees), which subdivide space into halves along all dimensions; and k-d (k-dimensional) trees [Bentley, 1975], which subdivide space based on the spatial distribution of the data, one dimension in each step. While this representation is more efficient than raster ones, objects still need to be approximated.
- **Cell decomposition** allows for differently shaped elements to be used. This makes it possible to represent an object more closely, but these representations lose most of the good characteristics of other decomposition models, such as easy ordering, uniqueness and simple topology. They are also difficult to use directly and are thus of limited use outside of meshes for Finite Method Models (FEMs).

Constructive models

These types of models are also based on a combination of basic elements. However, any Boolean operation is permitted to be performed between the objects [Putnam and Subrahmanyam, 1986], which are much more rich and complex. These can be classified according to the basic elements involved:

- **Half-space** models use a basic element consisting of a *half-space*, a mathematical function that defines a surface that splits space into two parts [Bieri and Nef, 1988]. The usual limit of this type of model depends on the selection of half-spaces available (e.g. planar, spherical, cylindrical, conical, etc.). They are able to express objects well and concisely, but operations on them are relatively complex and it is easy to create invalid objects (e.g. unbounded). An example of these is the regular polytopes representation of Thompson [2007].
- **Constructive Solid Geometry (CSG)** [Voelcker and Requicha, 1977] is a similar type of representation, with the distinction that the user does not have access to individual half-spaces. Instead, objects defined by a hierarchical structure of boolean operations performed on simpler elements. The advantages and disadvantages of CSG are similar to other half-space models. However, since individual half-spaces are not available, it is often also not possible to represent lower dimensional objects. An example CSG object is shown in Figure 3.3.

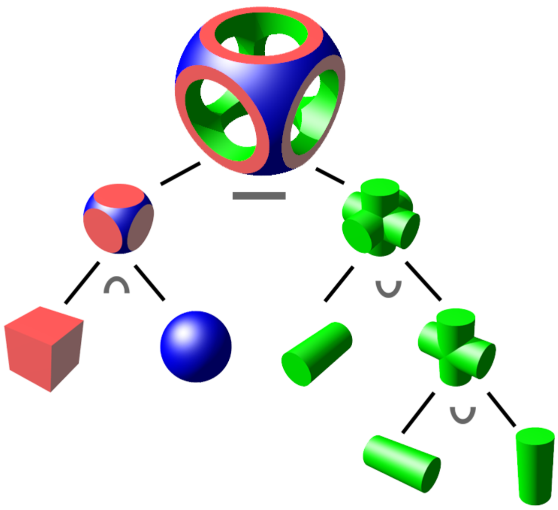


Figure 3.3: A CSG object is constructed based on a hierarchy of boolean operations performed on simple objects. From Wikimedia Commons.

Boundary models

These types of models originated from the polygonal/polyhedral models used in computer graphics. They use a construction in which higher dimensional objects are composed of lower dimensional ones defining its boundary. There are two general techniques in the creation of boundary models:

- **Incidence models** are graph-based representations in which a different types of elements are defined, usually one per dimension (i.e. vertex, edge, face, etc.). However, only the elements of the highest dimension are actually required to be explicitly defined, while others can be defined implicitly, with their characteristics stored within the higher dimensional elements that they belong to. The arcs in the graph correspond to the topological relations between the different dimensional elements (cells). A simplex based incidence model for n-dimensions will be discussed in Section 3.2.1.

- **Ordered topological models** use a single type of basic element (usually a half-edge), on which different operations act. All the cells are thus implicitly defined based on their constituting elements, which are ordered based on certain criteria. Both the attributes and the topological relations between the objects are stored within the constituting elements. Two ordered topological models for n-dimensions will be discussed in Sections 3.2.2 and 3.2.3.

Among the models presented above, boundary models seem to be the most suited to the representation of n-dimensional data, since they allow for a compact and efficient representation that does not need to conform to specific geometric constraints. These models are also the only ones that extend well to arbitrary dimensions, since they can be constructed using simple generic operations (unlike constructive models) and their size does not grow exponentially as the dimensionality of the data increases.

Despite the fact that topology is not a prerequisite in boundary models, it is required for an efficient implementation [Ellul and Haklay, 2006]. Otherwise, very simple operations like the traversal of a model become very inefficient. Therefore, during this PhD research, the focus will be on boundary models with topology.

Incidence models are most efficient when the exact configuration of the basic elements is well known. This happens when using n-simplices, which are the simplest possible elements of a certain dimension n. Thus, a 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a triangle, a 3-simplex a tetrahedron, and so on. Since, simplices are composed by a known number of lower dimensional simplices and are adjacent to a known number of n-simplex neighbours, these facts can be exploited by data structures that implement this model.

Meanwhile, ordered topological models work best when trying to represent more complex objects, in which the number of simplices that they are composed of is not known. In this case, they are much more efficient than incidence models, which would require variable-length lists to be able to represent them. Therefore, they are better suited to the representation of n-polytopes, the n-dimensional analogue of a polygon or polyhedron.

N-simplices and n-polytopes can be considered to be the extremes of a series of models that represent n-dimensional objects in a manner that has a direct relation between the object and the model. Polytopes imply that objects are stored as-is, which simplifies their creation, but could be cumbersome to use, since some operations can be quite complex (e.g. validation) and objects can be arbitrarily large. Meanwhile, simplices can be seen as the opposite approach, decomposing n-polytopes into minimal parts. Their creation can therefore be can be quite cumbersome (see Shewchuk [1998a]), but operations on them are bounded and very simple (see Karimipour et al. [2010]). However, an intermediate option is possible, involving the subdivision of n-polytopes into smaller polytopes that guarantee certain conditions, using intermediate representations that retain some of the advantages of completely unrestricted n-polytopes and n-simplices.

3.1.1 n-Simplices

Subdividing a feature into n-dimensional simplices is relatively straightforward in 2 dimensions. However, in dimensions 3 and higher, not every polytope can be subdivided without adding extra vertices (Steiner points), the simplest example of which is the Schönhardt polyhedron [Schönhardt, 1928] (see Rambau [2005]), which is shown in Figure 3.4. In fact, it is already an NP-hard problem to determine whether a polyhedron is tetrahedralisable [Ruppert and Seidel, 1992].

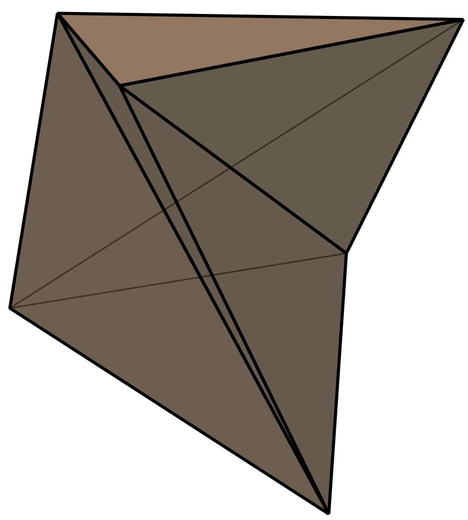


Figure 3.4: The Schonhardt polyhedron is the simplest one that cannot be tetrahedralised without adding extra vertices. From Wikimedia Commons.

The subdivision of n-dimensional space into n-dimensional simplices is generally referred to as an *n-dimensional triangulation*. Out of all possible n-dimensional triangulations, some are especially relevant for this project: Delaunay triangulations, regular triangulations, conforming Delaunay triangulations, and constrained Delaunay triangulations.

One desirable property is maximising the minimum internal angles of a triangulation, since it helps to build more robust algorithms and better looking results (e.g. in maps or shape reconstruction). This is accomplished by the Delaunay triangulation [Delaunay, 1934] (and its dual, the Voronoi diagram [Voronoi, 1908] or Dirichlet tessellation [Dirichlet, 1850]), which are also unique as long as the points from its vertices are in *general position*, i.e. for $1 \leq k \leq$

$n - 1$, no $k + 2$ vertices lie on an k-flat and no $k + 3$ points on a k-sphere [de Berg et al., 2008]. The Delaunay triangulation is defined by the *Delaunay property*, or empty circle property, in which the circum-hypersphere of the vertices of any simplex in the triangulation is empty of other points. Many algorithms for 2D Delaunay triangulations exist, including roughly five approaches: incremental insertion (walk [Devillers, 1998; Green and Sibson, 1978; Guibas et al., 1992] or jump & walk [Mücke et al., 1996]), divide & conquer [Dwyer, 1987; Guibas and Stolfi, 1985; Lee and Schachter, 1980], sweep-line [Fortune, 1987], gift wrapping [Tanemura et al., 1983] and convex hull (in a higher dimension) based [Brown, 1979; Clarkson et al., 1992]. See Su and Drysdale [1995] for a survey. In higher dimensions, there seems to be less options available, mainly gift wrapping [Dwyer, 1991], plane sweep [Shewchuk, 2000], and convex hull based [Barber et al., 1996]. If a certain quality (i.e. triangle aspect ratio) is required, a refinement algorithm may be used after the triangulation has been constructed [Ruppert, 1995; Si, 2006].

The Delaunay triangulation is actually a specific type of regular triangulations [Lee, 1991], in which every point is weighted and its weight is used for distance calculations in the neighbourhood of the point. When all weights are the same, the triangulation is the Delaunay triangulation of the point set. A popular algorithm for the construction of regular triangulations in n-dimensions is discussed in Edelsbrunner and Shah [1992]. Regular triangulations can be used to remove slivers as well [Cheng et al., 1999].

Another desirable property is to force the inclusion of certain simplices into the triangulation while preserving the Delaunay property (e.g. to conform to the shape of an object). These simplices are known as a Planar Straight Line Graph (PSLG) in 2D and a Piecewise Linear Complex (PLC) in higher dimensions. One can do so with a conforming triangulation [Hansen and Levin, 1992; Saalfeld, 1991], where additional vertices are added so that simplices from the PLC are represented by a (usually much larger) set of simplices in the triangulation. The number of additional vertices to be added is upper bounded [Edelsbrunner and Tan, 1993], but may still be quite large. Also, while there are some solutions for 3D conforming Delaunay triangulations, such as Shewchuk [1998b], Murphy et al. [2001] and Cohen-Steiner et al. [2004]); to the best of my knowledge, there are no articles that discuss the creation of conforming triangulations in dimensions higher than 3, or software that is able to do so.

Another option are Constrained Delaunay Triangulations (CDTs) [Chew, 1989], in which no additional vertices need to be added and simplices in the PLC are present and not split in the triangulation (unless there are intersecting simplices), but only a weaker property, the *constrained Delaunay property* is preserved, where every simplex in the triangulation is either a simplex from the PLC or it is *constrained Delaunay*. A simplex is constrained Delaunay if it has a circus-hypersphere that encloses no vertex of the PLC that is visible from the interior of the simplex, where visibility is occluded by the simplices in the PLC [Shewchuk, 1998a]. Efficient algorithms for the construction of CDT exist, such as Shewchuk [1996] or Agarwal et al. [2005] in 2D, Si and Gärtner [2005] in 3D, and Shewchuk [2000] in higher dimensions. However, not all PLCs can be triangulated, which is a significant problem. Shewchuk [1998a] describes a condition that guarantees that a CDT of a PLC can be created, it is sufficient but not necessary, and it can be tested with relative ease.

A good link between n-dimensional models and spatio-temporal ones is presented in Worboys [1994]. It argues that requirements often indicate the existence of at least two temporal dimensions (for transaction and valid time) which are orthogonal to each other, and thus proposes a model of spatio-temporal simplicial complexes, where each spatio-temporal simplex is composed of a spatial simplex and a bitemporal object.

3.1.2 *n*-Polytopes

Representing features as *n*-polytopes involves storing them while keeping their original structure. This construction is conceptually very simple, since no further processing of the objects is required. In this manner, a polytope models a bounded region with uniform characteristics. These polytopes can be represented based on a combination of lower dimensional primitives, such as the tesseract shown in Figure 3.5.

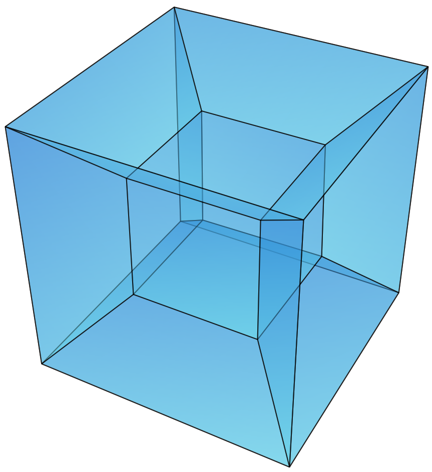


Figure 3.5: The tesseract or 4-cube is the four dimensional analogue of a cube. It can be directly represented with 4 solids (cubes), 24 faces (squares), 32 edges, and 16 vertices. From Wikimedia Commons.

When modelling GIS data, an *n*-dimensional polytope can adequately model both spatial extent and any number of non-spatial characteristics. For instance, in 4D space-time, a polytope can represent all the changes to a polyhedron in time [Hazelton et al., 1990].

3.1.3 Convex n-Polytopes or Other Intermediate Representations

One more option lies in decomposing an n-polytope into smaller polytopes that fulfil certain characteristics. The most interesting case among these involves convex polytopes, a higher dimensional analogue of convex polygons and polyhedra. A convex polytope is one that does not have any dents or holes, or formally, one for which any two points on its surface can be joined with a straight line segment that is only in the interior of the polytope.

Convex polytopes have been extensively studied and have many special properties (see Grünbaum [1967] for an extensive study), such as:

- A slice of a convex n-polytope is a convex $(n-1)$-polytope.
- A convex polytope can be modelled by a finite number of half-spaces.
- A convex polytope can be easily decomposed into n-simplices.
- The $(n-1)$ polytopes that form the n-polytope's boundary are also convex.

Based on characteristics like these, many algorithms perform much better or only work with convex polytopes [Bajaj and Dey, 1990; Kriegel et al., 1991]. Therefore, it might be desirable to split n-polytopes into convex parts, which is known as the *convex decomposition* of a polytope.

However, doing this operation is not trivial. While it has been proven that it is always possible to do so [Chazelle, 1980], many of the existing algorithms are limited to certain dimensions, such as Chazelle and Dobkin [1985] for 2D and [Bajaj and Dey, 1990] for 3D. Also, despite the fact that it is desirable to have a decomposition into a minimum number of pieces, this is known to be an NP-hard problem [Chazelle, 1984].

Nevertheless, an algorithm to perform a convex decomposition of n-dimensional polytopes has been created by Bulbul et al. [2009], based on Alternate Hierarchical Decomposition (AHD) and the application of any n-dimensional convex hull algorithm.

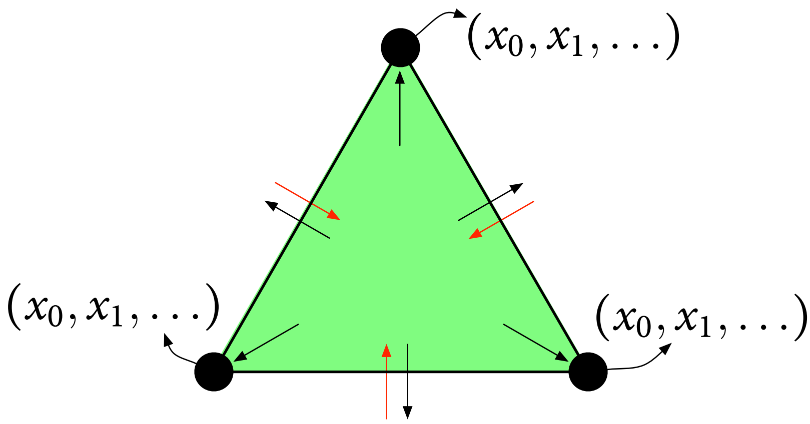
Thompson [2007] describes the use of so-called *regular polytopes*, in which polytopes are defined as unions of a finite set of convex polytopes, which are themselves described as a finite set of half spaces.

3.2 n-Dimensional Data Structures

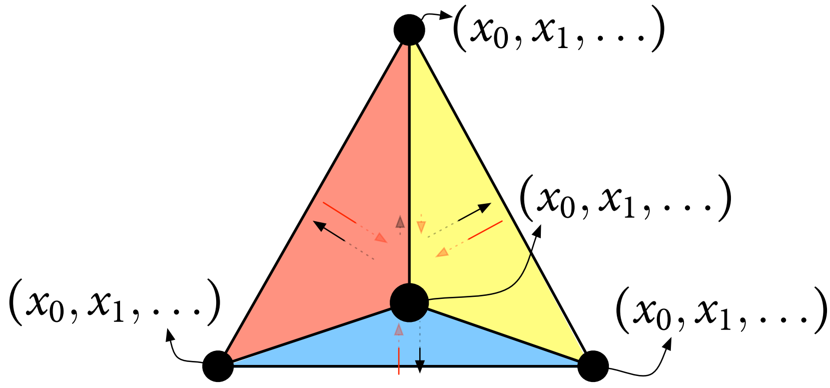
In this section, three types of data structures with support for higher dimensional data are analysed: n-simplices, the family of the quad-edge/facet-edge/cell-tuple data structures, and G-maps. Additionally, some space is devoted to data structures that currently only have support for 2D and 3D, but could be extended to work in higher dimensions.

3.2.1 n-Simplex based

The simplest data structures are meshes using n-simplices as a base, so that each structure represents a simplex. The end result is the formation of a *simplicial complex*. As long as the dimension is known, the number of pointers in the data structure is known as well. At its most basic, to represent each n-dimensional simplex it is only necessary to use $2n+2$ pointers, $n+1$ for each vertex, and $n+1$ for connections to its neighbouring simplices. However, in practice, more memory is needed in order to store information in the simplex itself or the lower dimensional simplices that conform it (e.g. vertex coordinates, normal vectors, etc.), to store the orientation of each simplex, or to specify which $n-1$-simplex of a neighbouring n-simplex is contacted, among other requirements. See Figure 3.6 for an example of this data structure in 2D and 3D.



(a) Triangle-based data structure (6 pointers per triangle)



(b) Tetrahedron-based data structure (8 pointers per tetrahedron)

Figure 3.6: n-simplex based data structures in (a) 2D and (b) 3D. Black arrows represent pointers from the element shown, while red arrows show the ones from other elements that point to this one.

This is the representation used in CGAL 2D and 3D triangulations [Boissonnat et al., 2002] and in Shewchuk's Triangle [Shewchuk, 1996] and Pyramid. For higher dimensions, the creation of higher dimensional analogues is straightforward. However, the dimension of the simplices to be represented should be known. Otherwise, the structure becomes much less efficient, as a variable length lists of pointers are required, which requires more memory and is slower.

The simplicity of this data structure facilitates the inclusion of optimisations to decrease its memory usage. For instance, Triangle uses the last two bits of a triangle's pointers to the neighbouring triangles to store which edge of the neighbour it is connected to [Shewchuk, 1996], which means that neighbouring triangles' memory addresses must be within a quarter of the total address space. Another interesting possibility is the usage of a compressed representation [Blandford et al., 2005; Isenburg and Snoeyink, 2000], or external memory algorithms [Amenta et al., 2003] to further reduce memory usage.

3.2.2 Quad-edge, facet-edge, cell-tuple

The quad-edge data structure was first described in Guibas and Stolfi [1985], storing 2D subdivisions based on edges. In this manner, vertices and faces are implicitly defined as rings of edges. Vertices are commonly referenced by one of its outgoing edges, while faces can be defined by a starting edge and starting direction from it. A subdivision and its dual are both directly represented in this data structure.

Figure 3.7 shows the half-edge algebra described for the quad-edge data structure. However, it is important to note that all of them can be described in terms of only three (e.g. *Flip*, *Rot* and *Onext*). Furthermore, if dealing with orientable manifolds, it is possible to obtain all other references from only two (e.g. *Rot* and *Onext*). Since non orientable manifolds are relatively rare in GIS, these two operations are usually sufficient, especially when combined with special handling of ambiguous cases.

Based on this half-edge algebra, edges can be seen as groups of 4 half-edges as defined by subsequent *Rot* operations on *e* (*e*, *eRot*, *eFlip* = *eRotRot* and *eRotSym* = *eRotRotRot*). The quad-edge data structure is then defined by its basic element, a *quad* (Figure 3.8), in which pointers to the *Onext* edge from each of the half-edges are stored, in what is denoted as $e[r]$`.Next`.

For the construction of subdivisions using the quad-edge data structure, only two basic operators are required. `MakeEdge` creates a new edge linked to itself (which is analogous to representing a subdivision of the sphere). Meanwhile, `Splice` receives two edges, joining them if they form two distinct loops or separating them if they form the same one. However, it is certainly desirable to have higher level functions to simplify the manipulation of objects, which can be built of `MakeEdge` and `Splice`.

Dobkin and Laszlo [1987] created the the *facet-edge data structure*, which can be seen as a 3D equivalent of the quad-edge data structure, incorporating a relation between edges and faces as a basic element instead of only an edge. This is done by defining an order not only for

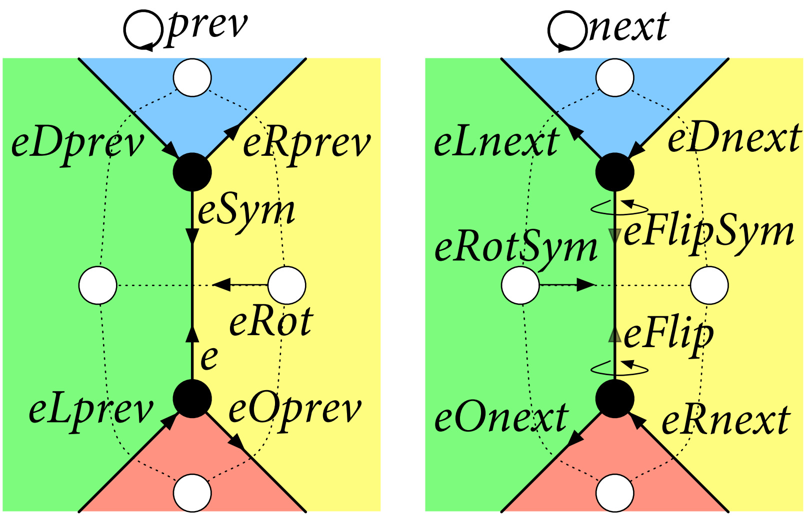


Figure 3.7: The edge functions of the quad-edge data structure, in terms of the Edge *e*. In this nomenclature, *O* refers to the origin of the edge, *D* to its destination, *L* to its left face, *R* to its right face. *eSym* is an edge with the same direction but opposite orientation, while *eFlip* is an edge with the same orientation but opposite direction. To navigate between a subdivision and its dual, *eRot* is used, defined as an oriented edge from *eRight* to *eLeft*. *eOprev*, *eOnext*, *eDprev*, *eDnext*, *eLprev*, *eLnext*, *eRprev* and *eRnext* are the previous or next edges of *e* with reference to a certain face or vertex, according to a predefined counterclockwise rotation direction.

the traversal of an edge, but for an adjoining face as well. The result is a so-called *facet-edge pair* (Figure 3.9).

In order to traverse a facet-edge data structure, four basic operators exist: *Enext*, *Fnext*, *Clock* and *Rev*. These are shown in Figure 3.10. Obtaining the dual of a subdivision is very simple and elegant, the facet ring and the edge ring are swapped, so that the old facet ring becomes the new edge ring, and vice versa.

As with the quad-edge data structure, facet-edge pairs are partitioned into groups of eight. Each group consists of the four possible orientation combinations for the facet and edge rings of a facet-edge combination and its dual. Regarding operations, there is a simple `Make_facet_edge` function to create a new facet-edge pair, and two splice functions receiving two facet-edge pairs. `Splice_facets` works on the facet rings, combining them if they are distinct or separating them if they are identical, while `Splice_edges` performs an analogous operation of the edge rings. Additionally, since none of these operations alter the incidence relations between polyhedra and vertices, it is necessary to have a function that does so, which is called `Transfer`.

Finally, Brisson [1989] extended the concepts of the quad-edge and facet-edge data structures to arbitrary dimensions with the cell-tuple data structure. In it, a subdivided n-manifold

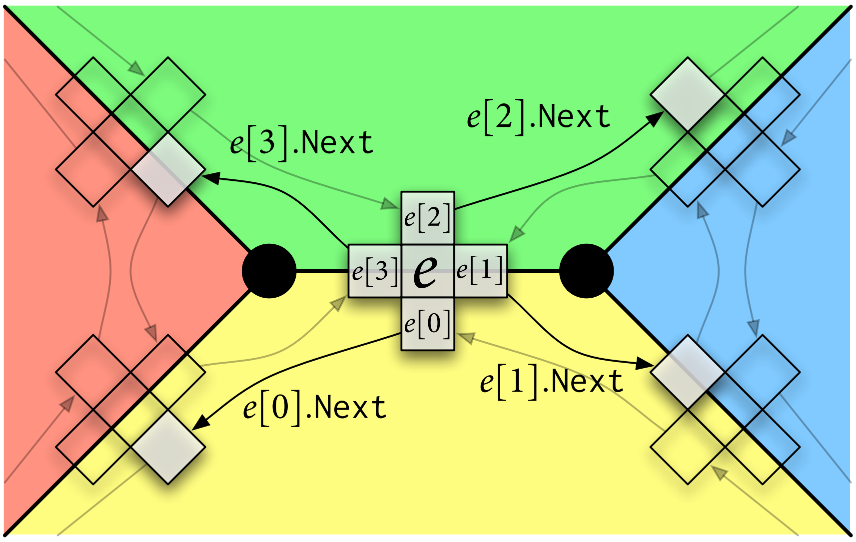


Figure 3.8: The basic element of the quad-edge data structure is the quad, in which the *Onext* element from each of the 4 half-edges that define an edge is stored, denoted as $e[r]$.`Next`. Note that it is possible to circle around a vertex or a face by subsequent applications of `Next`.

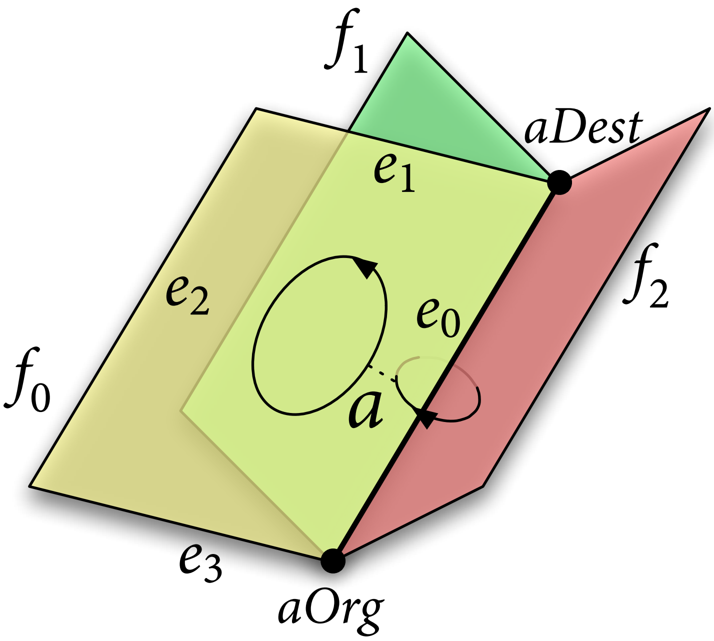


Figure 3.9: In the facet-edge data structure, the basic element is an edge-face pair conformed by an edge ring and a facet ring around the directed edge a going from *aOrg* to *aDest*. Note how the orientation specified an order for the incident edges (e_0, e_1, e_2, e_4) and faces (f_0, f_1, f_2).

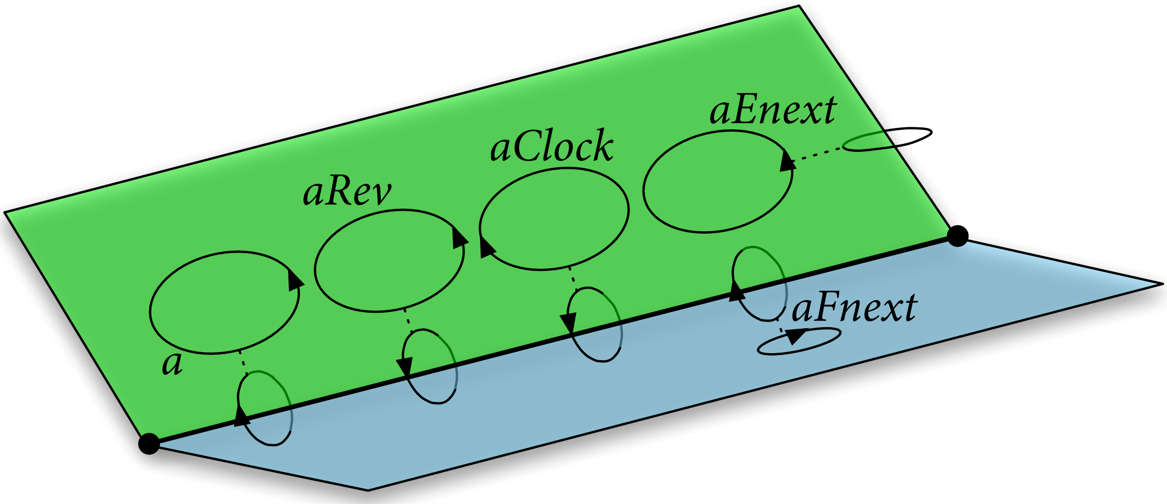
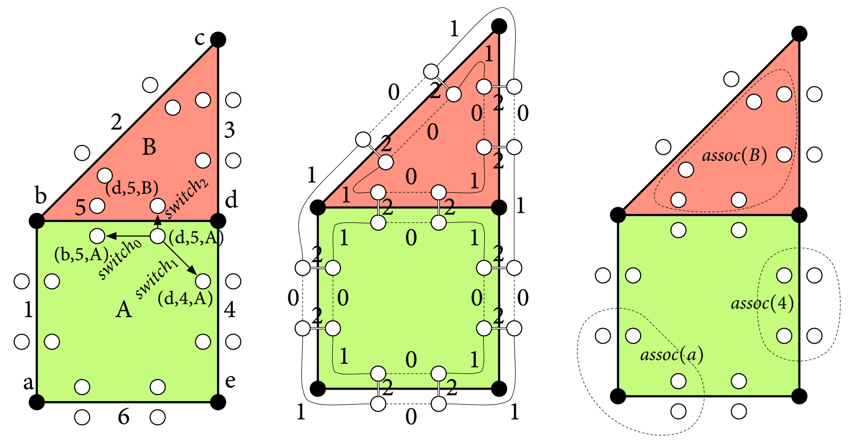


Figure 3.10: The four basic operators in the facet-edge data structure. *Enext* moves the edge ring to the next edge according to the orientation defined in the facet ring, *Fnext* moves the facet ring to the next face according to the orientation defined by the edge ring, *Clock* changes the orientation of the facet and edges rings, while *Rev* changes the orientation of the facet ring only.

is represented as a set of $n + 1$ tuples, where each tuple represents a half-edge and each element of a tuple the cells of ordered dimension that the object is part of. Two basic operations on a cell-tuple are defined: $switch_k$ for $0 \leq k \leq n$ obtains a different cell-tuple in which the k-dimensional element that the cell-tuple belongs to is changed, and $assoc(c_\alpha)$ obtains a set of related cell-tuples sharing the element c_α. Examples of the cell-tuple and the $switch_k$ and $assoc(c_\alpha)$ operators are shown in Figure 3.11.

Meanwhile, ordering in the cell-tuples data structure is based on the property that given a $(k-2)$-cell in the boundary of a $(k+1)$-cell, it is possible to order the k- and $(k-1)$-cells between them around the $(k-2)$-cell. This order alternates between $(k-1)$-cells and k-cells. As with the quad-edge and facet-edge data structures, moving a representation and its dual are directly represented, and $switch_k$ in the original is equivalent to $switch_{d-k}$ in the dual. Therefore, applying any operation on a reversed cell-tuple is equivalent to applying it to its dual.

Implementation-wise, it is best to consider a pairs of cell-tuples with their corresponding $switch_k$ operators. If using in a database or another method in which relational operators are possible, it is efficient to implement $assoc(c_\alpha)$ by looking for all cell-tuples that have a certain element in a known position. An example of the cell-tuple data structure stored in a database in shown in Figure 3.12.

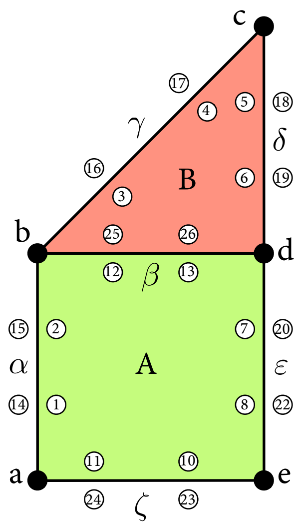


(a) A $switch_k$ operation on a cell-tuple traverses the structure to obtain a neighbouring element that differs from the first one only on its k-dimensional cell.

(b) $switch_k$ operators connect all cell-tuples in a subdivision. The numbers show the dimension of the $switch_k$ operators displayed.

(c) $assoc(c_\alpha)$ returns a set of related cell-tuples that share a certain common element.

Figure 3.11: Cell-tuples and the $switch_k$ and $assoc(c_\alpha)$ operators.



(a) The objects being represented. Cell-tuples are numbered, vertices are assigned lowercase characters, edges greek characters, and faces uppercase characters.

cell-tuple	*0-cell*	*1-cell*	*2-cell*	*switch*$_0$	*switch*$_1$	*switch*$_2$
1	a	α	A	2	11	14
2	b	α	A	1	12	15
3	b	γ	B	4	25	16
⋮	⋮	⋮	⋮	⋮	⋮	⋮
26	d	β	B	25	6	13

0-cell	*x*	*y*
a	0.0	0.0
b	0.0	1.0
c	1.0	2.0
d	1.0	1.0
e	1.0	0.0

2-cell	*colour*
A	green
B	red

(b) The database tables representing these objects. The primary key of each table is underlined.

Figure 3.12: The cell-tuple data structure implemented in a database.

3.2.3 G-maps

Edmonds [1960] defines the concept of a combinatorial (or topological) map, a data structure for 2D subdivisions and surfaces of polyhedral objects. It has bases in combinatorial topology and was formalised under the name constellations by Jacques [1970].

A combinatorial map consists of 3 elements (D, σ, α):

- A finite set of half-edges D (darts when referring to G-maps).
- A permutation σ (for *sommet*, French for vertex), which returns the next dart, by turning around the vertex in a predefined direction.
- An involution α (for *arête*, French for edge), which returns the other dart of the same edge. It is considered an involution since reversing its direction has no effect ($\alpha = \alpha^{-1}$).

Additionally, the function ϕ, where $\phi = \sigma \circ \alpha$ returns the next dart of the same face. Sometimes ϕ is used instead of σ, so that a combinatorial map might also be (D, ϕ, α). A sample combinatorial map using (D, σ, α) is shown in Figure 3.13. Note that vertices are only implicitly represented, usually being linked to one of its incident half-edges.

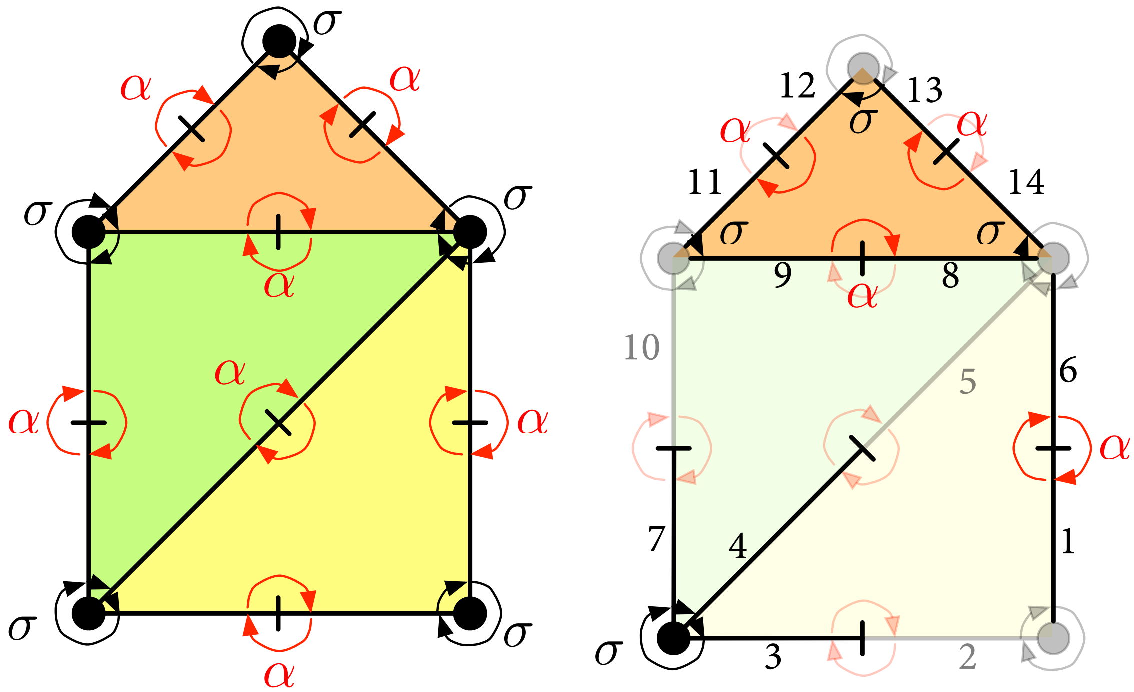
Afterwards, this concept was extended to 3D with the V-map data structure in Lienhardt [1988], adding the definition of a volume. Two important changes were made: every half-edge within every face is one dart, and an additional permutation γ joins adjacent darts of different faces, rotating around an edge in a way that $\alpha\gamma$ is an involution. An simple V-map is shown in Figure 3.14, and a more complex one that demonstrates the interaction between different volumes in Figure 3.15.

This concept was further generalised to n-dimensions with the G-map (or n-G-map) data structure in Lienhardt [1989] and Lienhardt [1994]. The main difference with respect to the V-map is that the σ and γ permutations between the different darts are removed, with all relations now being expressed by a tuple of involutions, each involution corresponding to a certain dimension. A G-map is thus defined by a $(d+2)$-tuple $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_d)$, where d is the maximum dimension of the objects to be represented, B is a set of darts, and α_n is an involution that connects objects of dimension n. In this manner, α_0 joins half-edges into edges, α_1 connects consecutive edges within a face, α_2 connects consecutive faces within a volume, and so on. An example 2-G-map (two dimensional G-map) is shown in Figure 3.16 and a 3-G-map in Figure 3.17.

In order to traverse a G-map, the operator $< \alpha_n, \alpha n + 1, \ldots, \alpha n + m > (d)$ obtains all the connected components

A sample implementation in memory (used in GOCAD), together with some validation algorithms, is given in Lévy and Mallet [1999], while a database implementation is available in Thomsen et al. [2008].

However, unless additional information is added, it is hard to adequately model many features common in GIS, e.g. holes, objects without explicit topological relations, and similarly hard to compare objects without ancillary data structures. In order to learn about the

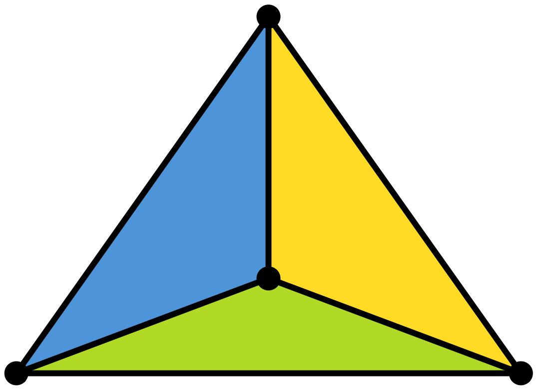


(a) A combinatorial map (D, σ, α) consists of a set of half-edges D, permutations σ that return the next dart by turning around a vertex (black arrows), and involutions α that return the opposite dart of the same edge (red arrows).

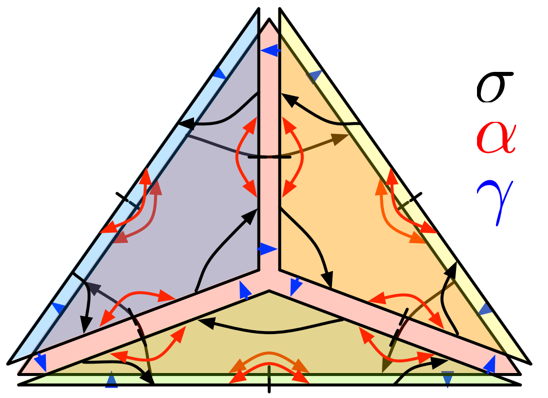
(b) Moving around a vertex is done by successive σ operators, obtaining the half-edges in a face by successive α operators, and getting the half-edges around a face by alternating σ and α operators.

Figure 3.13: A combinatorial map.

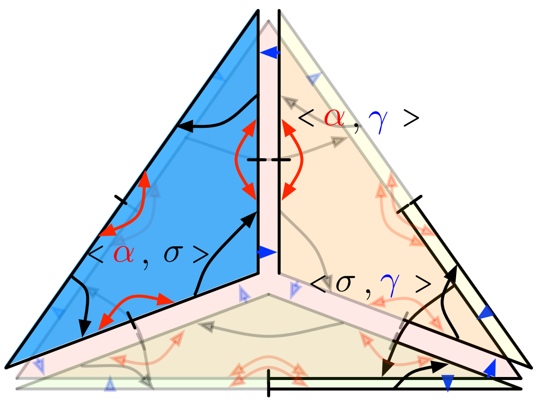
existing n-dimensional data structures and the challenges they represent with regards to GIS data, a sample implementation of G-maps was made, which was later modified to show the significant differences that are required in order to correctly support this type of data. The original and modified data structure is shown in Figure 3.18.



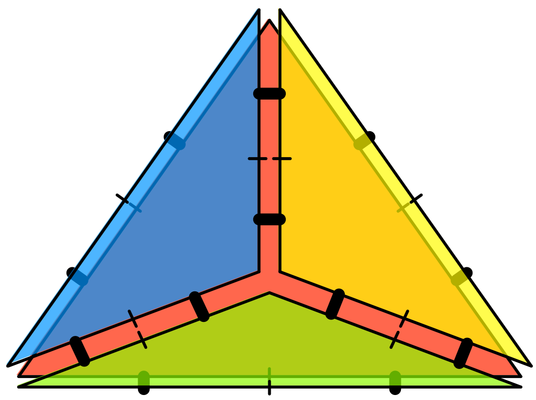
(a) The tetrahedron that is being represented.



(b) A V-map $(B, \alpha, \sigma, \gamma)$ consists of a set of threads B (half-edges), involutions α that return the opposite dart of the same edge, permutations σ that return the next dart by turning around a vertex (blue), and permutations γ that return the equivalent dart in a neighbouring face.

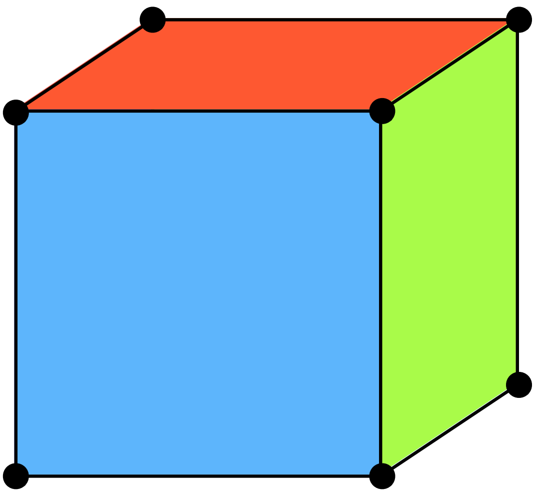


(c) Getting a vertex is done with alternating σ and γ operators, an edge with alternating α and γ operators and a face with alternating α and σ operators.

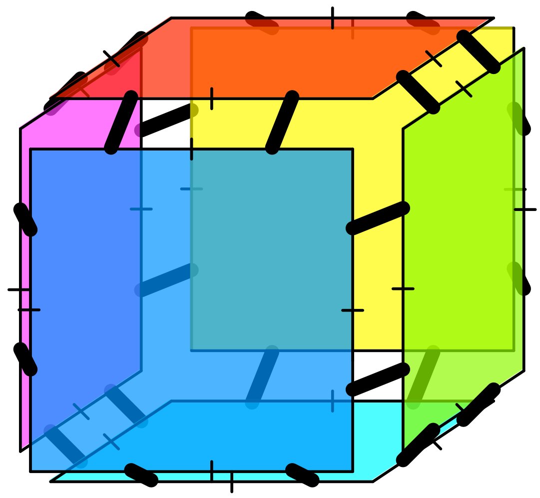


(d) A more compact manner to represent a V-map, in which the α and σ operators and not shown, and thick lines represent the γ operator.

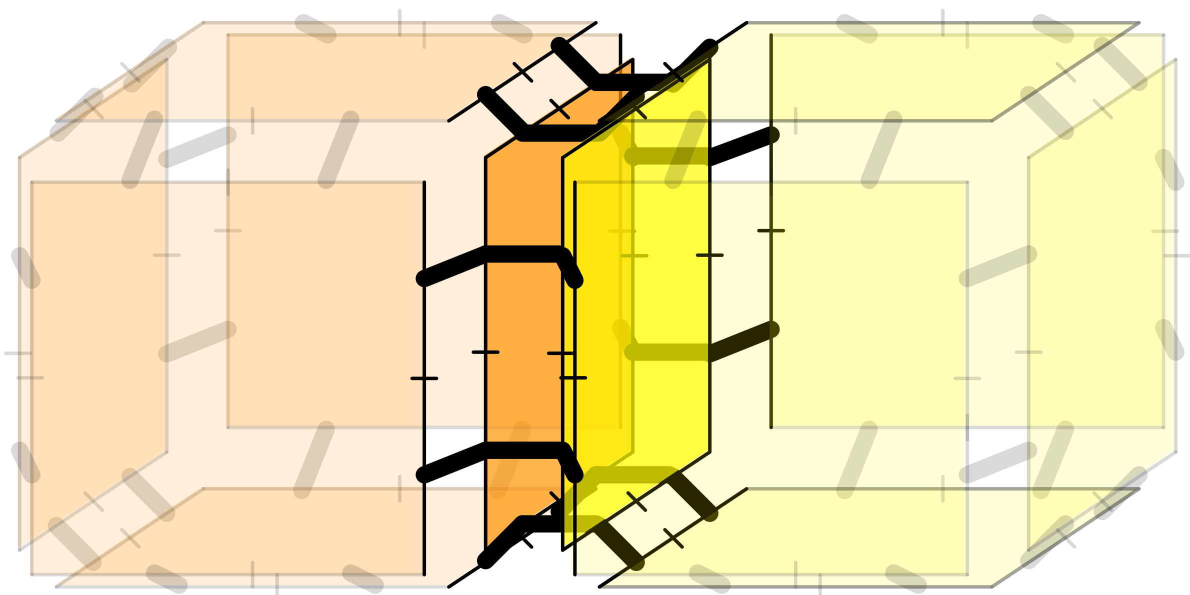
Figure 3.14: The V-map representation of a tetrahedron



(a) The cube that is being represented.

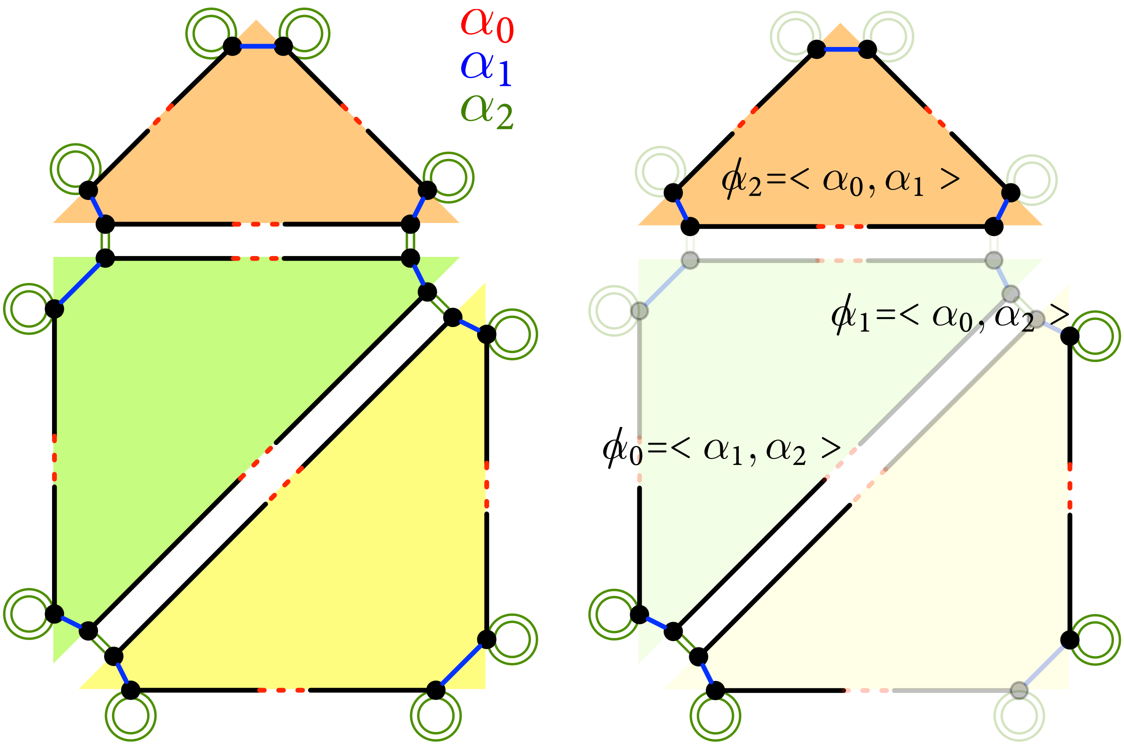


(b) A V-map representation of the cube.



(c) A V-map representation of two adjacent cubes. Since the γ operator (thick lines) can connect multiple threads, subsequent γ operators can return any thread from the set, and is therefore a permutation and not an involution.

Figure 3.15: The V-map representation of a cube and a pair of adjacent cubes



(a) A G-map is defined as a $(d + 2)$-tuple $G = (B, \alpha_0, \alpha_1, \ldots, \alpha_d)$, where d is the maximum dimension of the objects to be represented, B is a set of darts, and α_n is an involution that connects objects of dimension n.

(b) Moving around a vertex is done by successive σ operators, obtaining the half-edges in a face by successive α operators, and getting the half-edges around a face by alternating σ and α operators.

Figure 3.16: A 2-G-map representation of a 2D subdivision

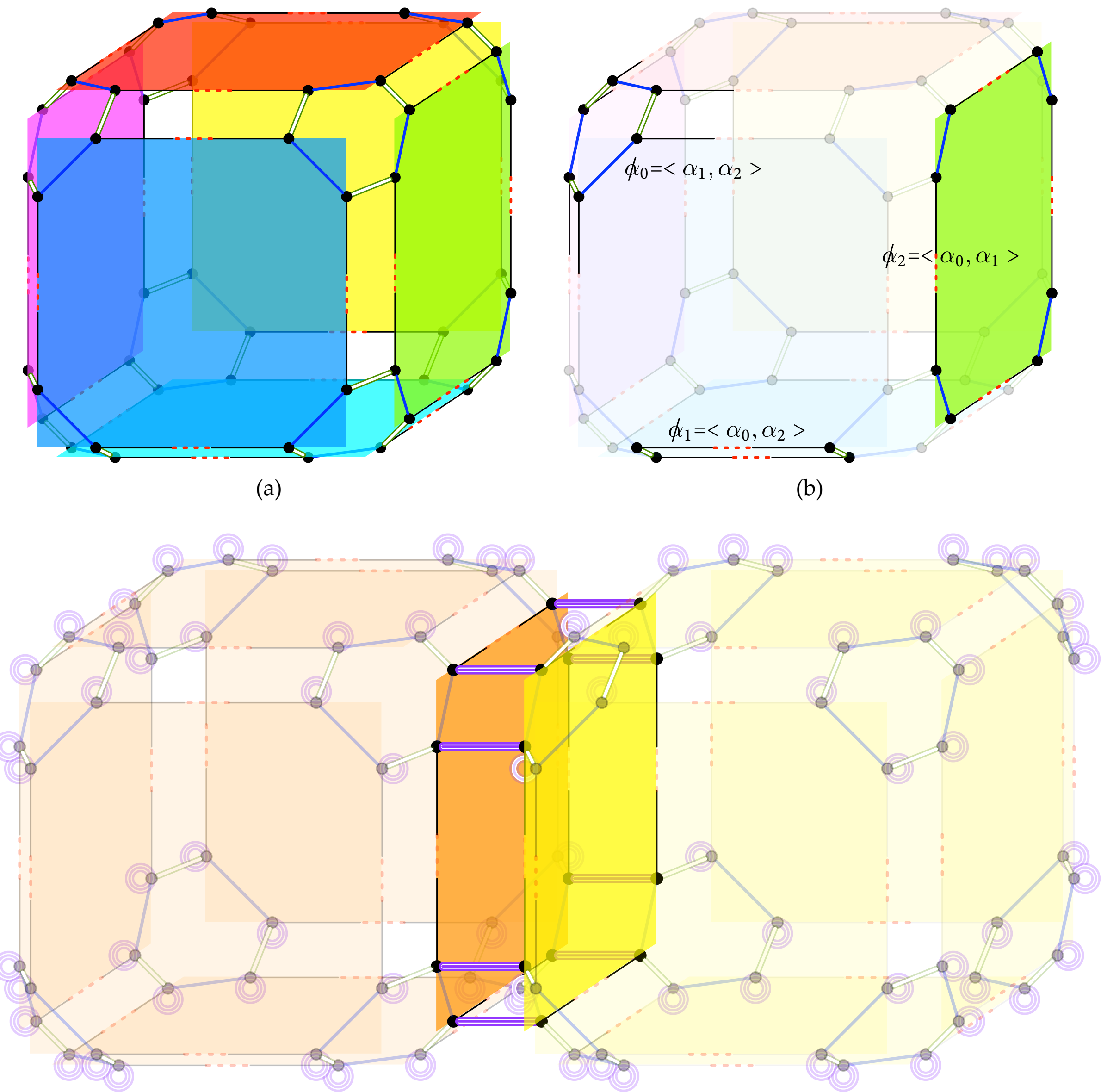


Figure 3.17: A 3-G-map representation of a cube and a pair of adjacent cubes

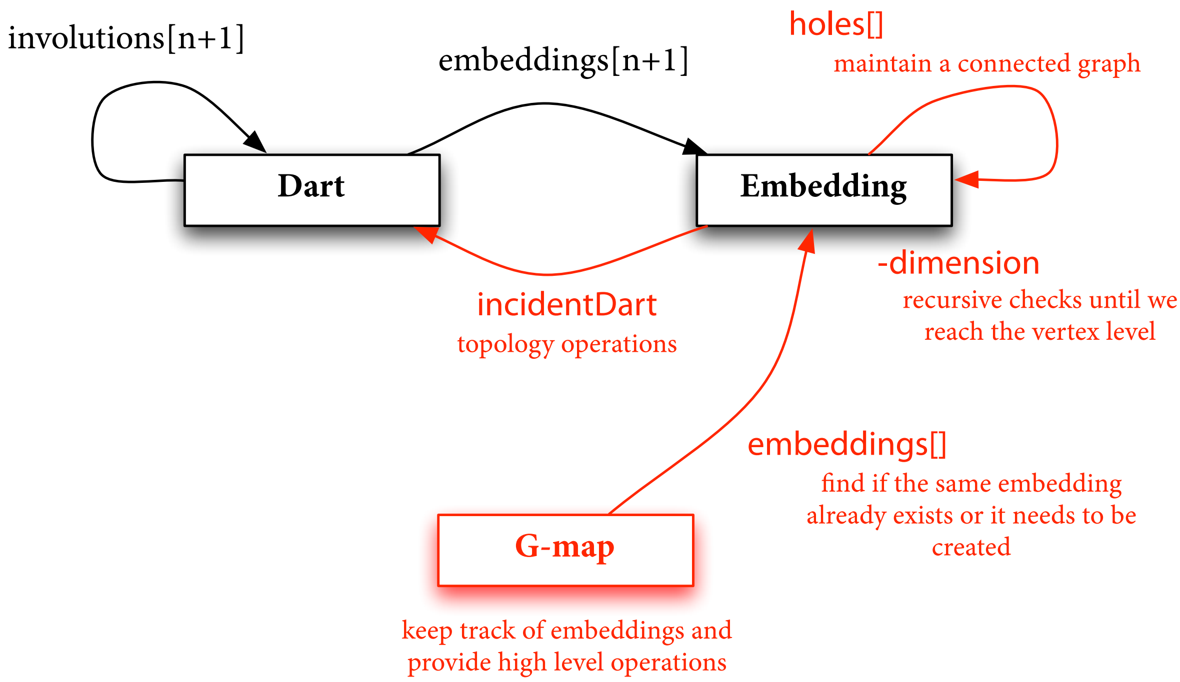


Figure 3.18: A straightforward version of the G-maps data structure (black) and the changes required to support GIS data (red).

3.2.4 Data structures for 2D and 3D

There are prominent data structures that are in widespread use in 2D and 3D, but that have not been generalised to higher dimensions. An additional option to consider would be the extension of any of these to higher dimensions. The most important data structures of this type are:

- The winged-edge data structure [Baumgart, 1975] is commonly used to represent 2D subdivisions and surfaces of 3D subdivisions. It is very similar to the quad-edge data structure, and can actually be considered as an expression of the *eOprev*, *eOnext*, *eRprev* and *eLnext* operations from the half-edge algebra from the quad-edge data structure.
- The Doubly Connected Edge List (DCEL) data structure [Muller and Preparata, 1978] is also commonly used to represent 2D subdivisions and surfaces of 3D subdivisions. It stores an explicit record for each vertex, half-edge and face. In a similar manner, higher dimensional cells could be added to the representation.
- The face-adjacency graph representation [Ansaldi et al., 1985] stores faces as nodes in a graph, with edges and vertices represented as arcs and hyperarcs.
- The radial-edge data structure [Weiler, 1986] is the most common of a family of data structures that are able to deal with non manifold surfaces.
- Selective geometric complexes [Rossignac and O'Connor, 1989] store vertices, edges and faces, along with their extent, boundary and their adjacencies, if necessary.

3.3 Operations on n-Dimensional Data

In addition to the data models and data structures, it is also important to study the different operations that are needed for GIS data. The higher dimensional data models and data structures used within this project should be able to support these operations. This will involve the implementation of the basic operations for GIS data, plus some additional operations to deal with higher dimensions in an easy manner.

The following operations have been considered so far. However, the list of not exhaustive, and will grow once a study of the operations in GIS data has been conducted.

Basic Operations: Creation, Deletion and Updates

As with all kinds of data, the basic operations involve the creation, deletion and update of a single or a set of spatial features. Convenience functions to perform these operations at a high level should be constructed.

Traversal

Traversing neighbouring polytopes is the most common operation that is performed on spatial data, and a base for most other operations. In many implementations[Lévy and Mallet, 1999], markers inside the data structure are used to ensure that each element is only visited once. However, it is also possible to do so using an auxiliary data structure, or in some cases, without one. For instance, de Berg et al. [1997] discusses a technique to traverse a 2D subdivision without using memory.

Selection

Selecting a number of features based on spatial, temporal, topological, scale-dependant, or thematic properties, among others. This selection can be used for further processing later on.

Defining Relations Between Objects

It is important to establish the different relations that are possible between any number of spatial objects. These could be based on the 4-intersection [Egenhofer and Franzosa, 1991] and 9-intersection [Egenhofer et al., 1994] models.

Embeddings into Lower Dimensions

It is important to consider the embeddings of higher dimensional spatial information into lower dimensions in order to be able to analyse and visualise the data in an easier manner. Specifically, since it is not possible to visualise high-dimensional spatial information directly while making it intuitively understandable to a human, it is important to consider how to create 2D or 3D representations of the data. This can be seen as a subproblem of dimensionality reduction. However, unlike non spatial data, it is crucial to preserve certain properties of the data.

There are established techniques that allow for dimensionality reduction in a manner in which certain properties are preserved. For instance, the Johnson-Lindenstrauss lemma states that a small set of points from high-dimensional Euclidean space may be embedded into a space of much lower dimension in such a way that distances between the points are nearly preserved [Johnson and Lindenstrauss, 1984]. One more example is Principal Component Analysis (PCA) [Pearson, 1901], which applies orthogonal transformations to data in order to convert correlated variables into uncorrelated ones, after which the dimensions with lowest variance can be discarded.

Some of the most useful operations that could be implemented involve slicing, projections, and the creation of isosurfaces [Lévy et al., 2001].

Visualisation

An important part of the project will deal with the visualisation of higher dimensional data. During the course of the research for this proposal, a simple visualiser for 3D data has been developed, which will be extended in order to work with the developed data structures and support higher dimensional data.

Validation

GIS data in particular needs certain validity criteria which are more strict than in other fields. For instance, volumes should be closed, correctly oriented and comply to certain standards (e.g. ISO 19107 [ISO, 2003]).

Chapter 4

PhD Research

4.1 Research Objectives

The creation of models that integrate space and other dimensions, such as time, scale, and application dependent feature spaces (e.g. full waveform remote sensing data) is a complex task, which needs to be tackled from multiple perspectives and spans the work of multiple people. This PhD research will be focused on three main axes that serve as foundations of such a model: the development of adequate n-dimensional data models, data structures, and algorithms to support operations on higher dimensional data.

The main research objective of this PhD project is therefore:

"The realisation of a data model, data structure and the basic algorithms required for the operations in a higher dimensional Geographic Information System"

where higher dimensional refers to four and more dimensions, but is not limited to any particular number.

Two small programs were written as part of the research performed for this proposal. One implements a modified version of the G-maps data structure with support for an arbitrary number of dimensions and GIS data. Another is a 3D visualisation engine. A practical vision for the future will be to be able to join these two programs, enabling real-time visualisation, modification and analysis of higher dimensional GIS data.

4.1.1 In scope

The scope of the PhD work has been defined as follows:

- All the research performed will take into account the possibility that any number of dimensions could be added in the future. However, case studies will be mainly focused on time and scale as the dimensions to be modelled. Nevertheless, different kinds of time could be considered as different dimensions.

- Existing n-dimensional data models and data structures will be studied, implemented, and adapted order to support GIS data. This will include the implementation of the basic operations needed to use such data. Creating implementations will make it possible to gain extensive knowledge about the inner workings of these data models and data structures, which will be taken into account for the creation of a final model.

- Since the human understanding of space is limited to the 3D Euclidean space in which we live, the ability to create 2D of 3D visualisations of higher dimensional data is not only very important, but necessary for this project as well. This will help us to have an intuitive understanding of the model, which will help the project as a whole.

- While the availability of automatically generalised data is of fundamental importance to the feasibility of any multi-scale system, it is a complete topic by itself and is considered to be outside the scope of this PhD project. However, some literature will be studied in order to understand the state of the art and the possibilities of using automatically generalised data.

- The research will be mostly limited to data conforming to the object model (as opposed to the field model), and thus does not necessarily conform to a grid (i.e. raster).

- Extend the definitions of a valid object in GIS to an n-dimensional valid object.

- Research will be focused into the creation of a boundary representation model with topology (as opposed to non topological models, like Simple Features [OGC, 2010a,b]). While other options exist (i.e. volume representations, no explicit topology), this concept seems to be the most promising, as previously discussed in Sections 3.1 and 3.2.

4.1.2 Partially in scope

There are some topics that are outside the scope of the project, but whose study is helpful in order to better understand the necessities of data structures for GIS and the project as a whole. They will not be studied in detail, but some research will be done on them. This will help to bring new knowledge to the GIS domain. The following objectives regarding these related topics have been defined:

- Research which existing spatio-temporal models exist and their characteristics, including their related spatio-temporal operators.

- Research which existing multi-scale and vario-scale models exist and their characteristics. Also, how a certain model at a predefined scale is best obtained from each.

- Compare the existing spatio-temporal and multi-scale models and establish what are the most important desirable characteristics that should also be present in an integrated model for GIS data.
- Research about generalisation of maps, both manual and automatic. This includes generalisation operators.

4.1.3 Out of scope

Moreover, there are themes which are considered to be completely outside the scope of the project. No work will be fone in the following:

- Handling raster data, field modelled data, or non topological data.
- Implementing (automatic) generalisation operators.

4.2 Methodology by topic

Five main themes within the PhD project have been identified. In the following sub-sections, the key activities that will be made in each are detailed:

4.2.1 Fundamental Knowledge

FK1 Learn about general topology and algebraic topology.

FK2 Learn about general visualisation algorithms.

4.2.2 n-Dimensional Data Models

DM1 Research which n-dimensional models for the representation of spatial information exist and their characteristics.

DM2 Compare the existing n-dimensional models and select one or a few that are considered most promising for further investigation in the context of an integrated n-dimensional GIS.

DM3 Taking into account the research on n-dimensional data structures and operations on n-dimensional data, select one or a few n-dimensional models that should be used for implementation and testing.

4.2.3 n-Dimensional Data Structures

DS1 Research what are the necessary characteristics for a data structure, taking into account the research on spatio-temporal and multi-scale modelling, and existing spatial ones that work in lower dimensions.

DS2 Research which n-dimensional data structures exist and their characteristics.

DS3 Investigate the advantages and disadvantages of using each data structure in the context of a higher dimensional GIS.

DS4 Analyse the changes that would be required in the aforementioned data structures in order to support the characteristics of spatial data, such as attaching attributes, and supporting holes, disconnected or overlapping objects.

DS5 Analyse how a database implementation of the aforementioned data structures would look like, and the challenges of creating a database implementation of it. Based on it, define what the optimum level of database integration for its usage should be (i.e. anywhere from only using a database as storage and operating solely in memory, to doing everything in a DBMS).

DS6 Taking into account the research on n-dimensional data models and operations on n-dimensional data, select a few data structures that should be implemented and tested.

4.2.4 n-Dimensional Operations

O1 Taking into account the research on spatio-temporal and multi-scale models, make a list of basic operations that should be supported based on existing classifications. This should include the elementary operations (e.g. traversal, marking, creation, modification, deletion, etc.) on which more complex ones can be built, and more complex ones to be able to query and manipulate data easily (e.g. selection, slicing, projections, etc.).

O2 Research about visualisation techniques and tools for higher dimensional data, as well as the current visualisation technologies and software for 3D data.

4.2.5 Implementation

I1 Obtain and/or generate test data sets, both in high dimensional Euclidean space, and combining 3D space, time and scale. There should be a variety of data sets with different characteristics, such as n-dimensional point clouds, data that can be represented as polytopes, building models, generated random data, etc. Some possibilities for test data sets include those from: 2D + time of the City of Rotterdam, standard computer graphics 3D mesh simplification (e.g. Stanford bunny), generated 3D + time data of the TU Delft campus, higher dimensional Voronoi diagrams of point set data, etc.

I2 Implement a few of the combinations of the data structures and data models (e.g. g-maps representing convex polytopes), including the necessary changes that should be made to them to support spatial data.

I3 Create a database implementations of the aforementioned data structures and models, with the database integration level that was established during the research on the data structures,

I4 Implement an interactive visualiser for n-dimensional data, based on techniques for dimensionality reduction (e.g. slicing, projections, creation of iso-surfaces) and standard 3D visualisation libraries and software.

I5 Implement the basic operations required for n-dimensional data, including those necessary for visualisation, those that come from 3D spatial GIS, spatio-temporal analysis, multi-scale systems, and those deemed necessary for an integrated system.

4.2.6 Testing and Analysis

TA1 Test and benchmark the built implementations with the obtained data sets.

TA2 Analyse the advantages and disadvantages of using each data model and data structure in the context of a higher dimensional Geographic Information System.

4.3 Time Planning

I have defined 6 distinct phases within the project. In each of them, several activities will be performed in parallel. The phases, their duration and their related activities are presented in the following table. The keys in parenthesis refer to activities in the methodology section.

Phase	Time period	Activities
1	June-October 2011	Initial (broad) literature review (FK1, FK2, DM1, DS1, DS2) Basic implementation of G-maps for GIS data (DS2, DS3, DS4, I2) Initial work on a 3D visualiser (FK2, I4) Research proposal
2	November 2011-February 2012	Continued literature review (FK1, FK2, DM1, DS1, DS2) Define the needed n-d operations (O1) Article about bridging computer science/GIS Implement more high-level operations on G-maps (O1) Obtain n-d data sets (I1) Loading and viewing 3D data in visualiser (O2, I4)
3	2012	Focused literature review (DM1, DS1, DS2) Test other data structures (DM2, DM3, DS3, DS6, I2, TA2) Comparison of n-d data structures (DS1, DS3, DS4, DS6) Article about n-d data structures Connect n-d data structure and visualiser (I2, I4) Loading and storing n-d data (DS5, I3)
4	2013	n-d operations for GIS data (O1, I5) Investigate database implementation (DS5, I3) Definition of the data structure to use (DM3, DS6, I2, TA2) Visualisation of n-d data (I4, I5)
5	2014	Formalisation of the developed ideas
6	January-May 2015	Work on dissertation Prepare for PhD defence

4.4 Expected Results and Deliverables

During this PhD project, I estimate that I will participate in the creation of:

- 6-8 conference/workshop articles
- 2-3 journal articles
- Dissertation

Additionally, the software developed during this project will be made open source to ensure its application and continued development.

Chapter 5

Practical Aspects

5.1 Courses

In this section, a list of courses that I have taken or I will take in the near future. These will help me acquire new knowledge for the project and for my scientific development.

PhD Start Up

Introduction on project and process management within the PhD, communication and presentation skills, and cultural issues. Gives an opportunity to know other PhD students within the university. June 20-22, 2011, in Soest, The Netherlands.

MADALGO & CTIC Summer School on High-dimensional Geometric Computing

Introduction to key-problems and techniques in high-dimensional geometric computing. It covers topics such as linear programming, algorithms for spaces with low intrinsic dimension, high-dimensional combinatorics and similarity search. August 8-11, 2011, in the Center for Massive Data Algorithmics (MADALGO), Aarhus, Denmark.

PROM-4: Scientific Writing in English

Improving the ability to write academic English, using an article or thesis chapter. September-December 2011, in the Faculty of Technology, Policy and Management.

Data Visualisation

Models of the visualisation process, colour models and use of colour, information visualisation, representation and processing of data, volume visualisation, medical visualisation, interactive visual data analysis, visualisation of vector fields and flows, feature extraction, and virtual reality for visualisation. November 2011-January 2012, in the Faculty of Electrical Engineering, Mathematics and Computer Science.

5.2 Supervision

Peter van Oosterom will be acting as promotor, Jantien Stoter as co-promotor and Hugo Ledoux as daily supervisor. Supervision for this PhD project will consist of bi-weekly meetings with Jantien Stoter and Hugo Ledoux, sporadic (2-4 times a year) meetings with Peter van Oosterom, plus other meetings when deemed necessary.

5.3 Software

This is a list of software that could be helpful for the project. It covers convenience libraries that can be used to construct more advanced functionality, and software that can perform part of the tasks of the project, which can be used both directly or to get new ideas and inspiration.

5.3.1 Useful Software and Libraries

CGAL

Library for robust geometric operations, including many packages for convex hulls, meshes, triangulations [Boissonnat et al., 2002], polygon and polyhedron operations, etc. Mainly 2D, with some 3D functionality (e.g. triangulations), and very limited operations in higher dimensions. Available at `http://www.cgal.org/`.

GDAL/OGR

Libraries for input and output of spatial data. GDAL works with raster formats, while OGR deals with vector ones. Mainly 2D formats. Available at `http://www.gdal.org/`.

GEOS

Library to model and manipulate 2D geometries. Available at `http://trac.osgeo.org/geos/`.

Hull

Arbitrary dimensional convex hulls, Delaunay triangulations, alpha shapes, and volumes of Voronoi cells using exact arithmetic and an incremental algorithm. It is available at `http://www.netlib.org/voronoi/hull.html`.

LaTeX, XeLaTeX and BibTeX

Typesetting and bibliography management software. There are several distributions available. I will be using MacTex, available at `http://www.tug.org/mactex/`.

MeshLab

Processing and viewing software for 3D triangular meshes. It is available at `http://meshlab.sourceforge.net/`.

PostgreSQL/PostGIS
PostgreSQL is a Database Management System (DBMS), available at `http://www.postgresql.org/`. Good 2D and some 3D spatial extensions for it are provided by PostGIS, available at `http://postgis.refractions.net/`.

Qhull
Computes convex hulls, Delaunay triangulations, Voronoi diagrams, half-space intersections about a point, furthest-site Delaunay triangulations, and furthest-site Voronoi diagrams in up to seven dimensions using the Quickhull algorithm [Barber et al., 1996]. No constrained triangulations or triangulation of non-convex surfaces. Fast, uses robust floating point arithmetic. Available at `http://www.qhull.org/`.

QGIS
GIS with good 2D visualisation functionality. It can use PostgreSQL and PostGIS as a back-end. Limited 3D support is available through osgEarth. Available at `http://www.qgis.org/`.

TetGEN
Generates the Delaunay tetrahedralisation, Voronoi diagram, and convex hull for three-dimensional point sets using Shewchuk's predicates. Can detect but not work with intersecting PLCs. Available at `http://tetgen.berlios.de/`.

Triangle
Very fast exact Delaunay triangulations, constrained Delaunay triangulations, conforming Delaunay triangulations, Voronoi diagrams, and high-quality triangular meshes in 2D. The details of the involved algorithms and their implementation, including a set of very fast robust floating point predicates are in Shewchuk [1997]. Available at `http://www.cs.cmu.edu/~quake/triangle.html`.

XCode
Good IDE for Mac, including debugging and profiling tools. Mainly meant for development in C/C++ and Objective-C. Available at `http://developer.apple.com/xcode/`.

5.4 Relation with Other Projects

This PhD work will be the foundation of the project 5D Data Modelling: Full Integration of 2D/3D Space, Time and Scale Dimensions. It will also involve the work of Jantien Stoter, Hugo Ledoux, and in the future, another PhD student. The funding is provided by an NWO Vidi grant with project code 11300.

Additionally, this project is related to the project Vario-scale Geo-information, involving the work of Peter van Oosterom, Martijn Maijers, and in the future, another PhD student. Cooperation will be established between the two whenever possible. The funding is provided by an STW grant with project code 11185.

Publications should include the relevant acknowledgements from these two organisations, including project numbers.

5.5 Conferences and Workshops

I will attend 1-2 conferences or workshops a year, which will be useful to gain new knowledge, present relevant findings, and establish cooperation with interested parties.

Hereafter a list of relevant conferences is presented. These could be used to publish articles, acquire new knowledge, get feedback and new ideas for the project, and meet people involved in the field. The list of conferences is organised by topic.

5.5.1 Computational Geometry

Annual Symposium on Computational Geometry (SoCG)
Computational Geometry. SoCG12: June 17-19, 2012 in Chapel Hill, United States. SoCG13: January, 2013, in Rio de Janeiro, Brazil.

European Workshop on Computational Geometry (EuroCG)
Computational geometry. 28th EuroCG: March 19-21, 2012, in Assisi, Italy. 29th EuroCGL: March, 2013, in Braunschweig, Germany.

5.5.2 Computer Graphics, Imaging and Visualisation

Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)
Computer graphics. SIGGRAPH 2011: August 7-11, in Vancouver, Canada. SIGGRAPH 2012: August 5-9, in Los Angeles, United States.

International Symposium on Visual Computing (ISVC)
Computer vision, computer graphics, virtual reality and visualisation. ISVC11: September 26-28, in Las Vegas, United States.

5.5.3 General

Conference on Spatial Information Theory (COSIT)
Spatial information theory, and related topics in cognition, psychology, linguistics, anthropology, geography, planning, computer science, artificial intelligence, and mathematics. COSIT11: September 12-16, 2011, in Belfast, United States.

5.5.4 GIS and Spatial Information

3D GeoInfo
Emerging topics in the field of 3D geo-information. 3D GeoInfo 2012: May 16-17, 2012, in Quebéc City, Canada.

ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS)
Research in geo-spatial data and knowledge. ACM SIGSPATIAL GIS 2011: November 1-4, 2011, in Chicago, United States.

AGILE Conference on Geographic Information Science (AGILE)
Research in geoinformation science, geomatics and spatial analysis. AGILE 2012: April 25-27, 2012, in Avignon, France.

Free and Open Source Software for Geospatial (FOSS4G)
Free and open source solutions for spatial information. FOSS4G 2011: September 12-16, in Denver, United States.

International Conference on Geographic Information Science (GIScience)
Fundamental research related to GIS. GIScience 2012: September 18-21, 2012, in Columbus, United States.

International Symposium on Spatial Data Handling (SDH)
GIS science. SDH 2010: May 26-28, 2010, in Hong Kong, China. SDH 2012: August, 22-24, 2012, in Bonn, Germany.

5.6 Journals

Hereafter a list of relevant journals is presented. The articles written during the project can be submitted to these.

ACM Transactions on Graphics (TOG)
Computer graphics, related to SIGGRAPH. Impact factor: 5.07, h-index: 91.

Computational Geometry: Theory and Applications
Computational geometry. Impact factor: 0.92, h-index: 26.

Computers & Geosciences
New computation methods for the geosciences. Impact factor: 0.53, h-index: 49.

Computers & Graphics
Research and applications of computer graphics techniques. Impact factor: 0.46, h-index: 32.

Discrete & Computational Geometry (DCG)
Computational geometry. Impact factor: 1.26, h-index: 30.

GeoInformatica
Computer science and geographic information science. Impact factor: 0.51, h-index: 24.

IEEE Computer Graphics and Applications (CG&A)
Computer graphics. Impact factor: 0.82, h-index: 47.

International Journal of Computational Geometry and Applications (IJCGA)
Computational geometry. Impact factor: 0.58, h-index: 23.

International Journal of Geographical Information Science (IJGIS)
GIS science and geocomputation. Impact factor (2010): 1.01, h-index: 51.

ISPRS International Journal of Geo-Information (IJGI)
Open access journal on geo-information. Impact factor: n/a, h-index: n/a.

Journal of Spatial Information Science (JOSIS)
Open access journal on spatial information science. Impact factor: n/a, h-index: n/a.

Bibliography

Alias Abdul-Rahman and Morakot Pilouk. *An Overview of 3D GIS Development*, chapter 2, pages 15–23. Springer Berlin / Heidelberg, 2008.

Pankaj K. Agarwal, Lars Arge, and Ke Yi. I/O-efficient construction of constrained Delaunay triangulations. In Gerth Brodal and Stefano Leonardi, editors, *Proceedings of the 13th Annual European Symposium*, volume 3669 of *Lecture Notes in Computer Science*, pages 355–366. Springer Berlin / Heidelberg, 2005.

Murad Davudovich Akhundov. *Conceptions of space and time: sources, evolution, directions*. MIT Press, 1986.

Jochen Albrecht. *Universal Analytical GIS Operations. A Task-Oriented Systematization of Data Structure-Independent GIS Functionality Leading Towards a Geographic Modeling Language*. PhD thesis, University of Vechta, 1995.

Nina Amenta, Sunghee Choi, and Günter Rote. Incremental constructions con BRIO. In *Proceedings 19th Annual Symposium on Computational Geometry*, pages 211–219, San Diego, USA, 2003. ACM Press.

Silvia Ansaldi, Leila de Floriani, and Bianca Falcidieno. Geometric modeling of solid objects by using a face adjacency graph representation. *IEEE Computer Graphics and Applications*, 19 (3):131–139, 1985.

Colin Arens, Jantien Stoter, and Peter van Oosterom. Modelling 3d spatial objects in a geo-dbms using a 3d primitive. *Computers & Geosciences*, 31:165–177, 2005.

Chanderjit Bajaj and Tamal K. Dey. Convex decomposition of polyhedra and robustness. Technical report, Purdue University, 1990.

C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, December 1996.

Umit Basoglu and Joel Morrison. The efficient hierarchical data structure for the US historical boundary file. In Geoffrey Dutton, editor, *Harvard Papers on Geographic Information Systems*, volume 4. Addison-Wesley, 1978.

Bruce G. Baumgart. A polyhedron representation for computer vision. In *Proceedings of the May 19-22, 1975, National Computer Conference and Exposition*, pages 589–596, 1975.

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

H. Bieri and W. Nef. Elementary set operations with d-dimensional polyhedra. In *Computational Geometry and its Applications*, volume 333 of *Lecture Notes in Computer Science*, pages 97–112. Springer Berlin / Heidelberg, 1988.

Daniel K. Blandford, Guy E. Blelloch, David E. Cardoze, and Clemens Kadow. Compact representations of simplicial meshes in two and three dimensions. *International Journal of Computational Geometry and Applications*, 15(1):3–24, 2005.

Jean-Daniel Boissonnat, Olivier Devillers, Sylvain Pion, Monique Teillaud, and Mariette Yvinec. Triangulations in CGAL. *Computational Geometry: Theory & Applications*, 22:5–19, 2002.

Erik Brisson. Representing geometric structures in d dimensions: topology and order. In *Proceedings of the 5th annual symposium on Computational geometry*, pages 218–227, New York, NY, USA, 1989. ACM.

K. Q. Brown. Voronoi diagrams from convex hulls. *Information Processing Letters*, 9(5):223–228, 1979.

Rizwan Bulbul, Farid Karimipour, and Andrew U. Frank. A simplex based dimension independent approach for convex decomposition of nonconvex polytopes. In *Proceedings of Geocomputation 2009*, 2009.

Alessandro Cecconi. *Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping*. PhD thesis, University of Zurich, 2003.

Bernard Chazelle. *Computational Geometry and Complexity*. PhD thesis, Carnegie-Mellon University, 1980.

Bernard Chazelle. Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM Journal on Computing*, 13(3):488–507, 1984.

Bernard Chazelle and David P. Dobkin. Optimal convex decompositions. In *Computational Geometry*, pages 63–133. Elsevier, 1985.

Pin-Shan Chen. The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.

Siu-Wing Cheng, Tamal K. Dey, Herbert Edelsbrunner, Michael A. Facello, and Shang-Fua Teng. Sliver exudation. In *Proceedings of the 15th Annual Symposium on Computational Geometry*, 1999.

L. Paul Chew. Constrained Delaunay triangulations. *Algorithmica*, 4(1):97–108, 1989.

Nicholas R. Chrisman. The role of quality information in the long-term functioning of a geographic information system. *Cartographica*, 1983.

Nicholas R. Chrisman. The risks of software innovation: A case study of the harvard lab. *The American Cartographer*, 15(3):291–300, 1988.

Christophe Claramunt and Marius Thériault. Managing time in gis: An event-oriented approach. In James Clifford and Alexander Tuzhilin, editors, *Proceedings of the International Workshop on Temporal Databases*, pages 23–42, 1995.

Christophe Claramunt, Christine Parent, Stefano Spaccapietra, and Marius Thériault. Database modelling for environmental and land use changes. In John Charles Harold Stillwell, Stan Geertman, and Stan Openshaw, editors, *Geographical Information and Planning*, Advances in Spatial Science, chapter 10, pages 181–202. Springer Berlin / Heidelberg, 1999.

Kenneth L. Clarkson, Kurt Mehlhorn, and Raimund Seidel. Four results on randomized incremental constructions. In Alain Finkel and Matthias Jantzen, editors, *Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science*, volume 577 of *Lecture Notes in Computer Science*, pages 461–474. Springer Berlin / Heidelberg, 1992.

David Cohen-Steiner, Éric Colin de Verdière, and Mariette Yvinec. Conforming Delaunay triangulations in 3d. *Computational Geometry: Theory & Applications*, 28:217–233, 2004.

J.T. Coppock and D.W. Rhind. The history of GIS. In D.J. Maguire, M.F. Goodchild, and D.W. Rhind, editors, *Geographical Information Systems: Principles and Applications*, volume 1, chapter 2, pages 21–43. Longman Scientific & Technical, 1991.

Mark de Berg, Marc van Kreveld, René van Oostrum, and Mark Overmars. Simple traversal of a subdivision without extra storage. *International Journal of Geographical Information Science*, 11(4):359–374, 1997.

Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3 edition, 2008.

Boris N. Delaunay. Sur la sphère vide. *Otdelenie Matematicheskikh i Estestvennykh Nauk*, 6: 793–800, 1934.

Olivier Devillers. Improved incremental randomized Delaunay triangulation. In *Proceedings of the 14th Annual Symposium on Computational Geometry*, pages 106–115, 1998.

Gustav Lejeune Dirichlet. Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Journal für die Reine und Angewandte Mathematik*, 40:209–227, 1850.

David P. Dobkin and Michael J. Laszlo. Primitives for the manipulation of three-dimensional subdivisions. In *Proceedings of the 3rd Annual Symposium on Computational Geometry*, pages 86–99. ACM, 1987.

Jürgen Döllner and Klaus Hinrichs. An object-oriented approach for integrating 3d visualization systems and gis. *Computers & Geosciences*, 26(1):67–76, February 2000.

David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, December 1973.

Charles Dupin. Carte figurative de l'instruction populaire de la France. *Jobard*, 1826.

Rex A. Dwyer. A faster divide-and-conquer algorithm for constructing Delaunay triangulations. *Algorithmica*, 2(1–4):137–151, 1987.

Rex A. Dwyer. Higher-dimensional Voronoi diagrams in linear expected time. *Discrete & Computational Geometry*, 6:343–367, 1991.

Herbert Edelsbrunner and N. R. Shah. Incremental topological flipping works for regular triangulations. In *Proceedings of the 8th Annual Symposium on Computational Geometry*, pages 43–52. ACM, 1992.

Herbert Edelsbrunner and Tiow Seng Tan. An upper bound for conforming Delaunay triangulations. *Discrete & Computational Geometry*, 10(2):197–213, 1993.

J. Edmonds. A combinatorial representation of polyhedral surfaces. *Notices of the American Mathematical Society*, 7, 1960.

Max J. Egenhofer and R. D. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Systems*, 5(2):161–174, 1991.

Max J. Egenhofer, David M. Mark, and John Herring. The 9-intersection: Formalism and its use for natural-language spatial predicates. Technical Report 94-1, National Center for Geographic Information and Analysis, 1994.

Claire Ellul and Muki Haklay. Requirements for topology in 3d gis. *Transactions in GIS*, 10(2): 157–175, March 2006.

ESRI. *Shapefile Technical Description*. Environmental Systems Research Institute, July 1998.

I.L. Finkel. A join to the map of the world: a notable discovery. *British Museum Magazine: the-5*, Winter:26–27, 1995.

Raphael Finkel and J.L. Bentley. Quad trees: A data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, 1974.

Steven Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2(1–4):153–174, 1987.

Andrew U. Frank. Spatial concepts, geometric data models, and geometric data structures. *Computers & Geosciences*, 18(4):409–417, 1992.

Andrew U. Frank and Sabine Timpf. Multiple representations for cartographic objects in a multi-scale tree: An intelligent graphical zoom. *Computers & Graphics*, 18(6):823–829, 1994.

Anders Friis-Christensen and Christian S. Jensen. Object-relational management of multiply represented geographic entities. In *Proceedings of the 15th International Conference on Scientific and Statistical Database Management*, pages 150–159. IEEE Computer Society, 2003.

Christopher M. Gold. What is gis and what is not? *Transactions in GIS*, 10(4):505–519, 2006.

P. J. Green and R. Sibson. Computing Dirichlet tessellations in the plane. *The Computer Journal*, 21(2):168–173, 1978.

Branko Grünbaum. *Convex polytopes*. John Wiley, 1967.

Leonidas J. Guibas and Jorge Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi. *ACM Transactions on Graphics*, 4(2):74–123, 1985.

Leonidas J. Guibas, Donald E. Knuth, and Micha Sharir. Randomized incremental construction of delaunay and voronoi diagrams. *Algorithmica*, 7:381–413, 1992.

Andrew J. Hansen and Peter L. Levin. On conforming delaunay mesh generation. *Advances in Engineering Software*, 14:129–135, 1992.

James E. Harrell and V. Max Brown. The oldest surviving topographical map from ancient egypt (turin papyri 1879, 1899 and 1969). *Journal of the American Research Center in Egypt*, 29: 81–105, 1992.

N.W.J. Hazelton, F.J. Leahy, and I.P. Williamson. On the design of temporally-referenced, 3-d geographical information systems: development of four-dimensional gis. In *Proceedings of GIS/LIS'90*, 1990.

Gary J. Hunter and Ian P. Williamson. The development of a historical cadastral database. *International Journal of Geographical Information Systems*, 4(2):169–179, 1990.

Martin Isenburg and Jack Snoeyink. Face fixer: Compressing polygon meshes with properties. In *Proceedings of SIGGRAPH 2000*, pages 263–270, July 2000.

ISO. *19107: Geographic Information – Spatial Schema*. International Organization for Standardization, March 2003.

A. Jacques. Constellations et graphes topologiques. *Combinatorial Theory and Applications*, 2: 657–673, 1970.

W. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

Christopher B. Jones and J. Mark Ware. Map generalisation in the web age. *International Journal of Geographical Information Science*, 19(8–9):859–870, 2005.

Benjamin Jowett, editor. *Plato's Timaeus*. Project Gutemberg, 1998.

Yehuda E. Kalay. *Architecture's New Media: Principles, Theories and Methods of Computer-Aided Design*. MIT Press, 2004.

Farid Karimipour, Mahmoud R. Delavar, and Andrew U. Frank. A simplex-based approach to implement dimension independent spatial analyses. *Computers & Geosciences*, 36(9): 1123–1134, September 2010.

Hans-Peter Kriegel, Holger Horn, and Michael Schiwietz. The performance of object decomposition techniques for spatial query processing. In *Proceedings of the 2nd International Symposium on Advances in Spatial Databases*, 1991.

Gail Langran and Nicholas R. Chrisman. A framework for temporal geographic informations. *Cartographica*, 25(3):1–14, 1988.

Carl W. Lee. Regular triangulations of convex polytopes. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Mathematics*, pages 443–456. American Mathematical Society, 1991.

D. T. Lee and B. J. Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer and Information Sciences*, 9(3):219–242, 1980.

Kunwoo Lee. *Principles of CAD/CAM/CAE*. Prentice Hall, 1999.

Bruno Lévy and Jean-Laurent Mallet. Cellular modeling in arbitrary dimension using generalized maps. Technical report, ISA-GOCAD, 1999.

Bruno Lévy, Guillaume Caumon, Stéphane Conreaux, and Xavier Cavin. Circular Incident Edge List: A data structure for rendering complex unstructured grids. In *Proceedings of the IEEE Visualization Conference 2001*, San Diego, CA, USA, 2001. IEEE Computer Society.

Zhilin Li. Reality in time-scale systems and cartographic representation. *The Cartographic Journal*, 31(1):50–51, 1994.

Pascal Lienhardt. Extension of the notion of map and subdivisions of a three-dimensional space. In Robert Cori and Martin Wirsing, editors, *Proceedings of the 5th Symposium on the Theoretical Aspects of Computer Science*, volume 294 of *Lecture Notes in Computer Science*, pages 301–311. Springer Berlin / Heidelberg, February 1988.

Pascal Lienhardt. Subdivisions of n-dimensional spaces and n-dimensional generalized maps. In *Proceedings of the 5th annual symposium on Computational geometry*, pages 228–236. ACM, 1989.

Pascal Lienhardt. Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Computer-Aided Design*, 23(1):59–82, 1991.

Pascal Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry and Applications*, 4(3):275–324, 1994.

Martti Mäntylä. *An introduction to solid modeling*. Computer Science Press, New York, USA, 1988.

Ernst P. Mücke, Isaac Saias, and Binhai Zhu. Fast randomized point location without preprocessing in two- and three-dimensional delaunay triangulations. In *Proceedings of the 12th Annual Symposium on Computational Geometry*, pages 274–283, New York, NY, USA, 1996. ACM.

D. E. Muller and F. P. Preparata. Finding the intersection of two convex polyhedra. *Theoretical Computer Science*, 7(2):217–236, 1978.

Michael Murphy, David E. Mount, and Carl W. Gable. A point-placement strategy for conforming delaunay tetrahedralization. *International Journal of Computational Geometry and Applications*, 11(6):669–682, 2001.

OGC. *OpenGIS Geography Markup Language (GML) Encoding Standard*. Open Geospatial Consortium, 3.2.1 edition, August 2007.

OGC. *OpenGIS City Geography Markup Language (CityGML) Encoding Standard*. Open Geospatial Consortium, 1.0.0 edition, August 2008a.

OGC. *OGC KML*. Open Geospatial Consortium, 2.2.0 edition, April 2008b.

OGC. *OpenGIS Implementation Specification for Geographic Information - Simple Feature Access - Part 1: Common Architecture*. Open Geospatial Consortium, 1.2.1 edition, August 2010a.

OGC. *OpenGIS Implementation Specification for Geographic Information - Simple Feature Access - Part 2: SQL option*. Open Geospatial Consortium, 1.2.1 edition, August 2010b.

Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

Nikos Pelekis, Babis Theodoulidis, Ioannis Kopanakis, and Yannis Theodoridis. Literature review of spatio-temporal database models. *The Knowledge Engineering Review*, 19(3):235–274, September 2004.

F. Penninga. *3D Topography A Simplicial Complex Solution-based Solution in a Spatial DBMS*. PhD thesis, Delft University of Technology, 2008.

Thomas K. Peucker and Nicholas R. Chrisman. Cartographic data strutures. *The American Cartographer*, 2(1):55–69, 1975.

Donna J. Peuquet. It's about time: A conceptual framework for the representation of temporal dynamics in geographic information systems. *Annals of the Association of American Geographers*, 84(3):441–461, 1994.

Donna J. Peuquet. *Representations of Space and Time*. Guilford Press, 2002.

Donna J. Peuquet and Niu Duan. An event-based spatiotemporal data model (estdm) for temporal analysis of geographical data. *International Journal of Geographical Information Science*, 9(1):7–24, 1995.

Simon Pigot. Topological models for 3d spatial information systems. In *Proceedings of Auto-Carto 10*, pages 368–392, 1991.

L.K. Putnam and P.A. Subrahmanyam. Boolean operations on n-dimensional objects. *IEEE Computer Graphics and Applications*, 6(6):43–51, June 1986.

Jörg Rambau. On a generalization of schönhardt's polyhedron. *Combinatorial and Computational Geometry*, 52:501–516, 2005.

Jonathan Raper. *Multidimensional geographic information science*. Taylor & Francis, 2000.

A. Renolen. History graphs: conceptual modeling of spatio-temporal data. *GIS Frontiers in Business and Science*, 2, 1996.

J. Rossignac and M. O'Connor. SGC: A dimension-independent model for pointsets with internal structures and incomplete boundaries. In M. Wosny, J. Turner, and K. Preiss, editors, *Proceedings of the IFIP Workshop on CAD/CAM*, pages 145–180, 1989.

Jim Ruppert. A delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18(3):548–585, May 1995.

Jim Ruppert and Raimund Seidel. On the difficulty of triangulating three-dimensional non-convex polyhedra. *Discrete & Computational Geometry*, 7(1):227–253, 1992.

A. Saalfeld. Delaunay edge refinements. In *Proceedings of the 3rd Canadian Conference on Computational Geometry*, pages 33–36, 1991.

Bianca Schön, Debra Laefer, Sean W. Morrish, and Michela Bertolotto. Three-dimensional spatial information systems: State of the art review. *Recent Patents on Computer Science*, 2: 21–31, 2009.

E. Schönhardt. über die zerlegung von dreieckspolyedern in tetraeder. *Mathematische Annalen*, 1:309–312, 1928.

Roger A. Schultz, Thomas Nielsen, Jeff R. Zavaleta, Raynal Ruch, Robert Wyatt, and Harold R. Garner. Hyperspectral imaging: A novel approach for microscopic analysis. *Cytometry*, 43 (4):239–247, April 2001.

Jonathan Richard Shewchuk. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996.

Jonathan Richard Shewchuk. Adaptive precision floating-point arithmetic and fast robust geometric predicates. *Discrete & Computational Geometry*, 18:305–363, 1997.

Jonathan Richard Shewchuk. A condition guaranteeing the existence of higher-dimensional constrained delaunay triangulations. In *Proceedings of the 14th Annual Symposium on Computational Geometry*, pages 76–85. ACM, 1998a.

Jonathan Richard Shewchuk. Tetrahedral mesh generation by delaunay refinement. In *Proceedings of the 14th Annual Symposium on Computational Geometry*, pages 86–95. ACM, June 1998b.

Jonathan Richard Shewchuk. Sweep algorithms for constructing higher-dimensional constrained delaunay triangulations. In *Proceedings of the 16th Annual Symposium on Computational Geometry, Hong Kong*, pages 350–359, 2000.

Dave Shreiner. *OpenGL Programming Guide*. Addison-Wesley, 2009.

Hang Si. On refinement of constrained delaunay tetrahedralizations. In *Proceedings of the 15th International Meshing Roundtable*, pages 509–528, 2006.

Hang Si and Klaus Gärtner. Meshing piecewise linear complexes by constrained delaunay tetrahedralizations. In *Proceedings of the 14th International Meshing Roundtable*, September 2005.

Richard T. Snodgrass and Ilso Ahn. A taxonomy of time databases. In *Proceedings of the 1985 ACM SIGMOD International Conference on Management of Data*, 1985.

P. A. Story and Michael F. Worboys. A design support environment for spatio-temporal database applications. In Andrew U. Frank and Werner Kuhn, editors, *Spatial Information Theory: A Theoretical Basis for GIS*, volume 988 of *Lecture Notes in Computer Science*, pages 413–430. Springer Berlin / Heidelberg, 1995.

Jantien Stoter and Siyka Zlatanova. 3d gis, where are we standing? In *Proceedings of the ISPRS Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis*, October 2003.

Jantien Stoter, Dirk Burghardt, Cécile Duchêne, Blanca Baella, Nico Bakker, Connie Blok, Maria Pla, Nicolas Regnauld, Guillaume Touya, and Stefan Schmid. Methodology for evaluating automated map generalization in commercial software. *Computers, Environment and Urban Systems*, 33(5):311–324, 2009.

Jantien Stoter, Thomas Visser, Peter van Oosterom, Wilko Quak, and Nico Bakker. A semantic-rich multi-scale information model for topography. *International Journal of Geographical Information Science*, 25(5):739–763, 2011.

Peter Su and Robert L. Scott Drysdale. A comparison of sequential delaunay triangulation algorithms. In *Proceedings of the 11th Annual Symposium on Computational Geometry*, pages 61–70. ACM, June 1995.

Masaharu Tanemura, Tohru Ogawa, and Naofumi Ogita. A new algorithm for three-dimensional voronoi tessellation. *Journal of Computational Physics*, 51(2):191–207, 1983.

Chen Tet-Khuan, Alias Abdul-Rahman, and Sisi Zlatanova. 3d spatial operations in geo dbms environment for 3d gis. In *Computational Science and Its Applicaction — ICCSA 2007*, volume 4705 of *Lecture Notes in Computer Science*, pages 151–163. Springer Berlin / Heidelberg, 2007.

R.J. Thompson. *Towards a rigorous logic for spatial data representation*. PhD thesis, Delft University of Technology, 2007.

Andreas Thomsen, Martin Breunig, Edgar Butwilowski, and Björn Broscheit. Modelling and managing topology in 3D geoinformation systems. In P. van Oosterom, S. Zlatanova, F. Penninga, and E. Fendel, editors, *Advances in 3D Geoinformation Systems*, Lectures Notes in Geoinformation and Cartography, chapter 14, pages 229–246. Springer Berlin / Heidelberg, 2008.

R.F. Tomlinson. The impact of the transition from analogue to digital cartographic representation. *The American Cartographer*, 15(3):249–261, 1988.

Nectaria Tryfona and Christian S. Jensen. Conceptual data modeling for spatiotemporal applications. *GeoInformatica*, 3(3):245–268, 1999.

Peter van Oosterom. *Reactive Data Structures for Geographic Information Systems*. PhD thesis, Leiden University, 1990.

Peter van Oosterom. Maintaining consistent topology including historical data in a large spatial database. In *Proceedings of Auto-Carto 13*, 1997.

Peter van Oosterom. Variable-scale topological data structures suitable for progressive data transfer: The GAP-face tree and GAP-edge forest. *Cartography and Geographic Information Science*, 32(4):331–346, 2005.

Peter van Oosterom. The tGAP structure: minimizing redundancy and maximizing consistency and offering access at any LoD. In K. Toth, A. Illert, and K. Murray, editors, *Proceedings of the INSPIRE Multiple- Representation and Data Consistency Workshop, Ispra*, November 2006.

Peter van Oosterom. Research and development in geo-information generalisation and multiple representation. *Computers, Environment and Urban Systems*, 33(5):303–310, 2009.

Peter van Oosterom and Martijn Meijers. Towards a true vario-scale structure supporting smooth-zoom. In *Proceedings of the 14th ICA/ISPRS Workshop on Generalisation and Multiple Representation, Paris*, 2011.

Peter van Oosterom, Hendrik Ploeger, Jantien Stoter, Rod Thompson, and Christiaan Lemmen. Aspects of a 4d cadastre: A first exploration. In *Proceedings of the XXIII International FIG congress*, 2006.

H.B. Voelcker and A.A.G. Requicha. Geometric modeling of mechanical parts and processes. *Computer*, 10(12):48–57, 1977.

Georgy Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *Journal für die Reine und Angewandte Mathematik*, 133:97–178, 1908.

Robert Weibel. Generalization of spatial data: Principles and selected algorithms. In Marc van Kreveld, Jürg Nievergelt, Thomas Roos, and Peter Widmayer, editors, *Algorithmic Foundations of Geographic Information Systems*, volume 1340 of *Lecture Notes in Computer Science*, pages 99–152. Springer Berlin / Heidelberg, 1997.

K. Weiler. The radial edge data structure: a topological representation for non-manifold geometric boundary modeling. In *Proceedings of the IFIP WG 5.2 Working Conference*, May 1986.

Michael F. Worboys. A unified model for spatial and temporal information. *The Computer Journal*, 37(1):26–34, 1994.

Michael F. Worboys, Hilary M. Hearnshaw, and David J. Maguire. Object-oriented data modelling for spatial databases. *International Journal of Geographical Information Systems*, 4(4): 369–383, 1990.

May Yuan. Wildfire conceptual modeling for building gis space-time models. In *Proceedings of GIS/LIS'94*, pages 860–869, 1994.

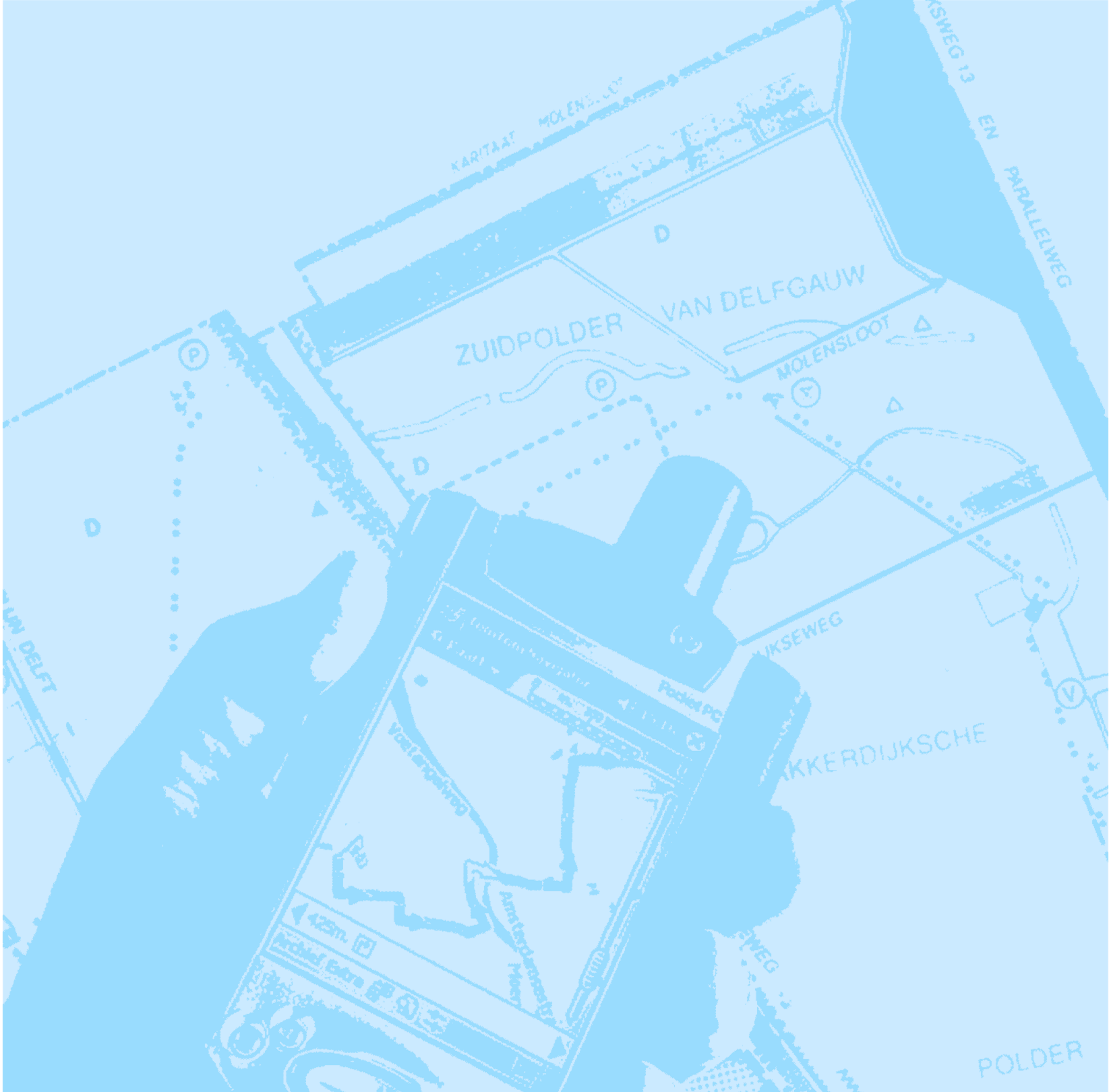
Yeting Zhang, Qing Zhu, Gang Liu, Wenting Zheng, Zhonghua Li, and Zhiqiang Du. GeoScope: Full 3d geospatial information system case study. *Geo-Spatial Information Science*, 14 (2):150–156, 2011.

Reports published before in this series

1. GISt Report No. 1, Oosterom, P.J. van, Research issues in integrated querying of geometric and thematic cadastral information (1), Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 29 p.p.
2. GISt Report No. 2, Stoter, J.E., Considerations for a 3D Cadastre, Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 30.p.
3. GISt Report No. 3, Fendel, E.M. en A.B. Smits (eds.), Java GIS Seminar, Opening GDMC, Delft 15 November 2000, Delft University of Technology, GISt. No. 3, 25 p.p.
4. GISt Report No. 4, Oosterom, P.J.M. van, Research issues in integrated querying of geometric and thematic cadastral information (2), Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 29 p.p.
5. GISt Report No. 5, Oosterom, P.J.M. van, C.W. Quak, J.E. Stoter, T.P.M. Tijssen en M.E. de Vries, Objectgerichtheid TOP10vector: Achtergrond en commentaar op de gebruikersspecificaties en het conceptuele gegevensmodel, Rapport aan Topografische Dienst Nederland, E.M. Fendel (eds.), Delft University of Technology, Delft 2000, 18 p.p.
6. GISt Report No. 6, Quak, C.W., An implementation of a classification algorithm for houses, Rapport aan Concernstaf Kadaster, Delft 2001, 13.p.
7. GISt Report No. 7, Tijssen, T.P.M., C.W. Quak and P.J.M. van Oosterom, Spatial DBMS testing with data from the Cadastre and TNO NITG, Delft 2001, 119 p.
8. GISt Report No. 8, Vries, M.E. de en E. Verbree, Internet GIS met ArcIMS, Delft 2001, 38 p.
9. GISt Report No. 9, Vries, M.E. de, T.P.M. Tijssen, J.E. Stoter, C.W. Quak and P.J.M. van Oosterom, The GML prototype of the new TOP10vector object model, Report for the Topographic Service, Delft 2001, 132 p.
10. GISt Report No. 10, Stoter, J.E., Nauwkeurig bepalen van grondverzet op basis van CAD ontgravingsprofielen en GIS, een haalbaarheidsstudie, Rapport aan de Bouwdienst van Rijkswaterstaat, Delft 2001, 23 p.
11. GISt Report No. 11, Geo DBMS, De basis van GIS-toepassingen, KvAG/AGGN Themamiddag, 14 november 2001, J. Flim (eds.), Delft 2001, 37 p.
12. GISt Report No. 12, Vries, M.E. de, T.P.M. Tijssen, J.E. Stoter, C.W. Quak and P.J.M. van Oosterom, The second GML prototype of the new TOP10vector object model, Report for the Topographic Service, Delft 2002, Part 1, Main text, 63 p. and Part 2, Appendices B and C, 85 p.
13. GISt Report No. 13, Vries, M.E. de, T.P.M. Tijssen en P.J.M. van Oosterom, Comparing the storage of Shell data in Oracle spatial and in Oracle/ArcSDE compressed binary format, Delft 2002, .72 p. (Confidential)
14. GISt Report No. 14, Stoter, J.E., 3D Cadastre, Progress Report, Report to Concernstaf Kadaster, Delft 2002, 16 p.
15. GISt Report No. 15, Zlatanova, S., Research Project on the Usability of Oracle Spatial within the RWS Organisation, Detailed Project Plan (MD-NR. 3215), Report to Meetkundige Dienst – Rijkswaterstaat, Delft 2002, 13 p.
16. GISt Report No. 16, Verbree, E., Driedimensionale Topografische Terreinmodellering op basis van Tetraëder Netwerken: Top10-3D, Report aan Topografische Dienst Nederland, Delft 2002, 15 p.
17. GISt Report No. 17, Zlatanova, S. Augmented Reality Technology, Report to SURFnet bv, Delft 2002, 72 p.
18. GISt Report No. 18, Vries, M.E. de, Ontsluiting van Geo-informatie via netwerken, Plan van aanpak, Delft 2002, 17p.
19. GISt Report No. 19, Tijssen, T.P.M., Testing Informix DBMS with spatial data from the cadastre, Delft 2002, 62 p.
20. GISt Report No. 20, Oosterom, P.J.M. van, Vision for the next decade of GIS technology, A research agenda for the TU Delft the Netherlands, Delft 2003, 55 p.

21. GISt Report No. 21, Zlatanova, S., T.P.M. Tijssen, P.J.M. van Oosterom and C.W. Quak, Research on usability of Oracle Spatial within the RWS organisation, (AGI-GAG-2003-21), Report to Meetkundige Dienst – Rijkswaterstaat, Delft 2003, 74 p.
22. GISt Report No. 22, Verbree, E., Kartografische hoogtevoorstelling TOP10vector, Report aan Topografische Dienst Nederland, Delft 2003, 28 p.
23. GISt Report No. 23, Tijssen, T.P.M., M.E. de Vries and P.J.M. van Oosterom, Comparing the storage of Shell data in Oracle SDO_Geometry version 9i and version 10g Beta 2 (in the context of ArcGIS 8.3), Delft 2003, 20 p. (Confidential)
24. GISt Report No. 24, Stoter, J.E., 3D aspects of property transactions: Comparison of registration of 3D properties in the Netherlands and Denmark, Report on the short-term scientific mission in the CIST – G9 framework at the Department of Development and Planning, Center of 3D geo-information, Aalborg, Denmark, Delft 2003, 22 p.
25. GISt Report No. 25, Verbree, E., Comparison Gridding with ArcGIS 8.2 versus CPS/3, Report to Shell International Exploration and Production B.V., Delft 2004, 14 p. (confidential).
26. GISt Report No. 26, Penninga, F., Oracle 10g Topology, Testing Oracle 10g Topology with cadastral data, Delft 2004, 48 p.
27. GISt Report No. 27, Penninga, F., 3D Topography, Realization of a three dimensional topographic terrain representation in a feature-based integrated TIN/TEN model, Delft 2004, 27 p.
28. GISt Report No. 28, Penninga, F., Kartografische hoogtevoorstelling binnen TOP10NL, Inventarisatie mogelijkheden op basis van TOP10NL uitgebreid met een Digitaal Hoogtemodel, Delft 2004, 29 p.
29. GISt Report No. 29, Verbree, E. en S.Zlatanova, 3D-Modeling with respect to boundary representations within geo-DBMS, Delft 2004, 30 p.
30. GISt Report No. 30, Penninga, F., Introductie van de 3e dimensie in de TOP10NL; Voorstel voor een onderzoekstraject naar het stapsgewijs introduceren van 3D data in de TOP10NL, Delft 2005, 25 p.
31. GISt Report No. 31, P. van Asperen, M. Grothe, S. Zlatanova, M. de Vries, T. Tijssen, P. van Oosterom and A. Kabamba, Specificatie datamodel Beheerkaart Nat, RWS-AGI report/GIST Report, Delft, 2005, 130 p.
32. GISt Report No. 32, E.M. Fendel, Looking back at Gi4DM, Delft 2005, 22 p.
33. GISt Report No. 33, P. van Oosterom, T. Tijssen and F. Penninga, Topology Storage and the Use in the context of consistent data management, Delft 2005, 35 p.
34. GISt Report No. 34, E. Verbree en F. Penninga, RGI 3D Topo - DP 1-1, Inventarisatie huidige toegankelijkheid, gebruik en mogelijke toepassingen 3D topografische informatie en systemen, 3D Topo Report No. RGI-011-01/GISt Report No. 34, Delft -2005, 29 p.
35. GISt Report No. 35, E. Verbree, F. Penninga en S. Zlatanova, Datamodellering en datastructurering voor 3D topografie, 3D Topo Report No. RGI-011-02/GISt Report No. 35, Delft 2005, 44 p.
36. GISt Report No. 36, W. Looijen, M. Uitentuis en P. Bange, RGI-026: LBS-24-7, Tussenrapportage DP-1: Gebruikerswensen LBS onder redactie van E. Verbree en E. Fendel, RGI LBS-026-01/GISt Rapport No. 36, Delft 2005, 21 p.
37. GISt Report No. 37, C. van Strien, W. Looijen, P. Bange, A. Wilcsinszky, J. Steenbruggen en E. Verbree, RGI-026: LBS-24-7, Tussenrapportage DP-2: Inventarisatie geo-informatie en -services onder redactie van E. Verbree en E. Fendel, RGI LBS-026-02/GISt Rapport No. 37, Delft 2005, 21 p.
38. GISt Report No. 38, E. Verbree, S. Zlatanova en E. Wisse, RGI-026: LBS-24-7, Tussenrapportage DP-3: Specifieke wensen en eisen op het gebied van plaatsbepaling, privacy en beeldvorming, onder redactie van E. Verbree en E. Fendel, RGI LBS-026-03/GISt Rapport No. 38, Delft 2005, 15 p.
39. GISt Report No. 39, E. Verbree, E. Fendel, M. Uitentuis, P. Bange, W. Looijen, C. van Strien, E. Wisse en A. Wilcsinszky en E. Verbree, RGI-026: LBS-24-7, Eindrapportage DP-4: Workshop 28-07-2005 Geo-informatie voor politie, brandweer en hulpverlening ter plaatse, RGI LBS-026-04/GISt Rapport No. 39, Delft 2005, 18 p.
40. GISt Report No. 40, P.J.M. van Oosterom, F. Penninga and M.E. de Vries, Trendrapport GIS, GISt Report No. 40 / RWS Report AGI-2005-GAB-01, Delft, 2005, 48 p.
41. GISt Report No. 41, R. Thompson, Proof of Assertions in the Investigation of the Regular Polytope, GISt Report No. 41 / NRM-ISS090, Delft, 2005, 44 p.
42. GISt Report No. 42, F. Penninga and P. van Oosterom, Kabel- en leidingnetwerken in de kadastrale registratie (in Dutch) GISt Report No. 42, Delft, 2006, 38 p.
43. GISt Report No. 43, F. Penninga and P.J.M. van Oosterom, Editing Features in a TEN-based DBMS approach for 3D Topographic Data Modelling, Technical Report, Delft, 2006, 21 p.
44. GISt Report No. 44, M.E. de Vries, Open source clients voor UMN MapServer: PHP/Mapscript, JavaScript, Flash of Google (in Dutch), Delft, 2007, 13 p.
45. GISt Report No. 45, W. Tegtmeier, Harmonization of geo-information related to the lifecycle of civil engineering objects – with focus on uncertainty and quality of surveyed data and derived real world representations, Delft, 2007, 40 p.

46. GISt Report No. 46, W. Xu, Geo-information and formal semantics for disaster management, Delft, 2007, 31 p.
47. GISt Report No. 47, E. Verbree and E.M. Fendel, GIS technology - Trend Report, Delft, 2007, 30 p.
48. GISt Report No. 48, B.M. Meijers, Variable-Scale Geo-Information, Delft, 2008, 30 p.
49. GISt Report No. 48, Maja Bitenc, Kajsa Dahlberg, Fatih Doner, Bas van Goort, Kai Lin,Yi Yin, Xiaoyu Yuan and Sisi Zlatanova, Utilty Registration, Delft, 2008, 35 p.
50. GISt Report No 50, T.P.M. Tijssen en S. Zlatanova, Oracle Spatial 11g en ArcGIS 9.2 voor het beheer van puntenwolken (Confidential), Delft, 2008, 16 p.
51. GISt Report No. 51, S. Zlatanova, Geo-information for Crisis Management, Delft, 2008, 24 p.
52. GISt Report No. 52, P.J.M. van Oosterom, INSPIRE activiteiten in het jaar 2008 (partly in Dutch), Delft, 2009, 142 p.
53. GISt Report No. 53, P.J.M. van Oosterom with input of and feedback by Rod Thompson and Steve Huch (Department of Environment and Resource Management, Queensland Government), Delft, 2010, 60 p.
54. GISt Report No. 54, A. Dilo and S. Zlatanova, Data modeling for emergency response, Delft, 2010, 74 p.
55. GISt Report No. 55, Liu Liu, 3D indoor "door-to-door" navigation approach to support first responders in emergency response - PhD Research Proposal, Delft, 2011, 47 p.
56. GISt Report No. 56, Md. Nazmul Alam, Shadow effect on 3D City Modelling for Photovoltaic Cells - PhD Proposal, Delft, 2011, 39 p.


ZUIDPOLDER
VAN DELFGAUW
MOLENSLOOT
PARALLELWEG
AKKERDIJKSCHE
POLDER