Frank, A.U., and W. Kuhn. "Cell Graphs: A Provable Correct Method for the Storage of Geometry." Paper presented at the Second International Symposium on Spatial Data Handling, Seattle, Wash. 1986.

Cell Graphs: A Provable Correct Method for the Storage of Geometry

Andrew U. Frank
Werner Kuhn

Report No. 55

Invited paper at the Second International Symposium on Spatial Data Handling

July 5-10, 1986, at Seattle, WA, USA

ABSTRACT

Several methods have been proposed for the storage of geometric properties in Geographic Information Systems, but many are based on the storage of metric data (coordinates) and analytical geometry. Because of the well-known limitations of implemenations of the algebra of computer real numbers, such systems cannot preserve topological relations such as incidence and inclusion during affine transformations.

We propose an approach in which topological relations are separately recorded and independent of metric positions. The method is based on the use of simplices, which are the simplest polyhedrons of each dimension. The zero-dimensional simplex is the point, the one-dimensional one the line, etc. In order to allow for non-straight lines as connections between points, we actually use cells, which are the homeomorphic image of simplices.

In order to store topological relations, we use two completeness principles: Completeness of incidence and completeness of inclusion. We can show that in such a geometric configuration topology is invariant to affine transformations, independently of the method selected for recording metric information.

For formal treatment we form a multi-sorted algebra (abstract data types). The axioms for this algebra must be selected such that the above-mentioned principles are maintained as invariants.

We rely on an arbitrary method to learn about the topological relations initially. This "oracle" may use the calculation of a distance and a threshold, query the user or decide randomly, but it cannot influence the consistency of the resulting geometry, as it is consulted only if the same information was not previously available and thus cannot lead to an inconsistent situation.

Reasonable performance is expected, as this method imposes a 'neighborhood' structure on the data. All operations use and change only data of objects in immediate proximity. Databases suitable for handling spatial data should permit clustering of data by proximity.

1 INTRODUCTION

Geometric objects and properties are not directly observable but are a product of an abstraction process applied when observing and describing reality. Geometric modelling in disciplines treating spatial information is done in two different modes, producing either raster or vector representations. Raster represenations are primarily used in picture processing, whereas the modelling for GIS and CAD systems is often based on vector or analytic geometry.

Several methods have been proposed for the storage of geometric properties in vector oriented Geographic Information Systems. Most are based on the storage of metric data (coordinates) and analytical geometry. Due to the well-known limitations of implementations of the algebra of computer real numbers, however, such systems cannot preserve topological relations such as incidence and inclusion during affine transformations. We propose an approach in which topological relations are separately recorded and independent of metric positions.

1.1 Scope Of Paper And Major Results

It has been recognized that systems which store only metric data are not capable of preserving topological properties such as incidences of points and lines or inclusions of points in polygons. But investigations into the theoretical reasons for this failure have been rare. We demonstrate that the preservation of topological invariants such as incidence and inclusion requires a number system which is, if not continuous, then at least countably infinite. Consequently there is no point in further investigating alternative algebras, since infinity is impossible to achieve in a digital computer.

In this paper we disregared the meaning of a geometric configuration and deal only with the geometry. The points, lines, and areas can then be used to carry the meaning in real world terms. This is a separate step which is not considered in the pure dealing with geometry in this paper.

Our approach is to search for a method which does not attempt to extract topology from stored metric data, but provides an independent representation for topological relations. For two-dimensional geometry, we propose a method based on a mesh of irregular triangles which is called a simplicial or cell complex. A simplex is the elementary topological cell in any given dimension, i.e. the zero dimensional simplex is the point, the one-dimensional one the line, the two dimensional one is the triangle, and so on. In order to allow for non-straight lines we actually use cells, which are the homeomorphic image of simplices.

In order to preserve the fundamental topological relations of incidence (i.e. when a point lies on a line or two points coincide) and inclusion (i.e. when a point lies within a polygon) during topological transformations, we extend and specialize the well-known 'closed world assumption' [REITER 1984] to two completeness principles:

- Completeness of incidence: the intersection of two simplices is either empty or a boundary simplex (of lower dimension). This rule excludes the intersection of lines at points which are not start- or end-points of the lines.

- Completeness of inclusion: all simplices of dimension n are boundary simplices of a simplex of dimension n + 1. This rules first demands that all points be endpoints of lines (i.e. no isolated, non-connected points) and second, that all lines be boundary lines of an area (i.e. there are no lines which are not part of a boundary).

The mathematical construction known as a simplicial complex fulfills these requirements.

It is important to retain this information with minimal redundancy (and further this redundancy must be known), not only because redundant data uses storage space, but more pointedly because redundant data bears in it the danger of contradiction and inconsistency.

As a method for formal treatment of this concept, we describe an abstract algebra (multi-sorted algebra) constructing simplicial complexes (precisely: cell complexes). We need (at least) the following operations:

- create an initial cell complex
- add a point to a cell complex
- connect two points in a cell complex (with a line of arbitrary shape)
- delete a point
- delete a line

For simplicity in this presentation, we do not include operations to delete points or lines nor do we fully treat the problems related to non-straight lines.

This method relies on initial knowledge of a topological relation together with its metric. If the user does not specify for a new element how it topologically relates to previously stored ones, the system makes the initial decision using the metric positions as an indication. This decision is arbitrary - we call it oracle - and may be used only once. It is only needed to complete the data and preserve consistency in a manner that does not require explicit treatment of unknown situations.

This formal treatment reveals a relatively simple, elegant solution. The goal of the exposition in this paper (given its limited size), is more the introduction of the formal methods and an attempt to convince the reader of their usefulness. A more complete treatment, including proofs, will follow in another publication.

The implementation so far has been easy, as the formal specifications were shown to be correct. We expect reasonable performance, as this method imposes a 'neighborhood' structure on the data. All operations use and change only data of objects in immediate proximity, and our database implementation already clusters data by proximity.

1.2 Contribution Towards The Larger Problem Of A Spatial Theory

The problem treated in this paper should be seen in the larger context of an emerging coherent spatial theory.

In 1983 a NASA-sponsored group of experts discussed 'Problems and Directions for Large Scale Geographic Information System Development' and concluded:
"There is at present no coherent mathematical theory of spatial relations.
..
The lack of a coherent theory of spatial relations hinders the use of automated geographic information systems at nearly every point. It is difficult to design efficient data bases, difficult to phrase queries of such data bases in an effective way,

difficult to interconnect the various subsystems in ways which enhance overall system function, difficult to design data processing algorithms which are effective and efficient. As we begin work with very large spatial data bases or global data bases the inabilities and inefficiencies which result from this lack of theory are likely to grow geometrically.

While we can continue to make some improvements in the use of automated geographic information systems without such a coherent theory on which to base our progress, it will mean that the developments will rest on an inevitably shaky base and progress is likely to be much slower than it might be if we had a theory to direct our steps. It may be that some advances will simply be impossible in the absence of a guiding theory." [BOYLE 1983]

In a recent Ph.D. thesis at the University of California at Santa Barbara it was argued "that the lack of symbolic representation and reasoning (of spatial knowledge) is a key factor in explaining the shortcomings of Geographic Information Systems" [PAZNER 1986].

Various branches of mathematics have rigorous theories which can be applied to spatial objects, e.g. Euclidian and analytical geometry, topology and graph theory. Other sciences, most notably geography and geomorphology, have developed their own concepts, but usually lack a consistent formal definition. They generally discuss different aspects of human understanding and human perception of space.

A spatial theory can help to formally explain a human expert's work in such areas as geography, spatial analysis, urban and regional planning, computer aided design or the dispatching of forces in a command environment. It is thus a logical prerequisite to the construction of expert systems in those disciplines.

We hope that the method presented here unifies metric and topologic operations and thus provides a more coherent view of geometry. We consider this an initial contribution to the emerging spatial theory.

2 PROBLEM DEFINITION, MATHEMATICAL BACKGROUND AND TERMINOLOGY

One of the main reasons for the appearance of unexpected "slivers and gaps" in information systems is that a precise (preferably formal) definition of the information to be stored has never been attempted or has remained incomplete. Humans interpreting system output may perceive relevant properties which are actually not known to the system.

Consequently, if the system does not contain information about such properties, operations may not preserve them and the basis for the human interpretation may suddenly disappear. It is like perceiving figures in clouds: no physical law assures that they are preserved, and a gust of wind can blow them away. Unfortunately, in many GIS certain geometric properties are not much more than an accidental cloud formation.

2.1 Our Understanding Of Geometry

At least since the work of Felix Klein (Erlanger Program) [KLEIN 1872], geometry is predominantly seen as the study of invariants under certain transformations [BLUMENTHAL 1970]. The idea of invariance is deeply rooted in the human conception of space, as the process of measuring distances, angles and volumes relies on the comparison of unknown quantities with measuring tools which are considered to have invariant size. Euclid's congruence theorems constitute an early incorporation of the idea of invariance into geometry.

The nature of the shortcomings of today's geometry-handling systems, namely the loss or change of certain qualitative properties of a geometric situation under common transformations, suggests that the construction of such systems could benefit from an axiomatic (as opposed to a computational or procedural) treatment of geometry which investigates invariance properties of the necessary operations.

Such an algebraic approach to the definition of geometry further coincides with modern techniques of software specification [GUTTAG 1977] [PARNAS 1972] [ZILLES 1984]. We are using abstract algebra extensively in order to achieve both a formal and unambiguous problem description and at the same time a convenient basis for implementation.

2.2 Dimension-independent Formulation

Traditionally, many definitions for spatial objects and properties are given for situations of specified dimensions - or are at least formulated and thought of this way. Different terms are used to denote mathematically similar objects of different dimensions (e.g. point, line, area). We have observed that it is usually not too difficult to generalize definitions slightly, such that they become independent of dimension. This is a method generally used in topology, and we will apply it in the sequel.

Regardless of the attempt to generalize definitions and make them dimension-independent, the presentation in this paper concentrates on the 2-dimensional case, which is most relevant to geographic, spatial data processing.

2.3 Topology: Some Elementary Concepts

Topology has been called the general study of continuity. It investigates topological structures of point sets. Algebraic topology uses algebraic means to treat these sets, whereas analytical topology is based on real analysis, using concepts from analysis (such as open sets, neighborhood, convergence) which are independent of the algebraic structure of the point sets. Our treatment of geometry will make use of concepts from both branches.

As in geometry as a whole, interest in topology is focussed on properties which remain invariant under certain transformations. The most important transformations in topology are homeomorphisms. A homeomorphism is a bijective and bicontinuous function, i.e. both the function and its inverse are unique and continuous. Two sets are said to be homeomorphic, or topologically equivalent, if a homeomorphism exists between them. A topological invariant is a property preserved by a homeomorphism.

A metric space is obtained from a point-set X by supplying it with a distance function d: X x X -> R. This function is subject to the following conditions :

(1) d (P,Q) >= 0
(2) d (P,Q) = 0 if and only if P = Q
(3) d (P,Q) = d (Q,P)
(4) d (P,Q) + d (Q,R) >= d (P,R) .

The convex hull of a set Y of points in an n-dimensional vector space is the smallest subset in this space which contains, for all v,w in Y, the segment vw = {av + bw | a,b >= 0, a+b = 1}.

Graph theory is a special discipline within algebraic topology. It investigates the topological problems which can be described by a (finite) number of points (nodes or vertices) and their connecting lines (edges). Graphs are particularly suited to represent incidences between points and lines, but since they lack a two-dimensional concept, they are not powerful enough to treat relations involving areas. In a GIS, where areas delimited by edges are usually meaningful and questions about the inclusion of points in areas occur, it is therefore inadequate to use graphs for the representation of topological relations.

2.4 The Geometry To Be Treated

2.4.1 Objects - The objects in our geometry are called FIGURES and their ensemble is a CONFIGURATION. A figure is either a point, a line, a polygon, or an area.

A POINT is defined by its coordinates, i.e. a tuple of real numbers. Two points with the same coordinates are identical.

A LINE is a segment of a plane analytic curve which does not intersect itself. It is defined by the two points bounding it (they do not belong to the line), an analytic description of its shape and an orientation.

A POLYGON is a connected alternating sequence of lines and points.

An AREA is a connected region bounded by one or several (in the case of "islands") closed polygons.

2.4.2 Operations - Fundamental geometric operations are: to insert a point, to join two points by a line, to build polygons from lines, to define an area by its boundary and to apply affine transformations to the whole configuration. Additional operations include the subtraction of one area from another, intersection operations etc.

2.4.3 Properties - The metric properties of the figures are: the position of points, length of lines, surface of areas, distance between two points or between a point and a line, and angles between intersecting lines.

Further we have an order relation for the points on a line to orient the line.

The topological relations are: coincidences of figures, incidences of points with lines, inclusions of points in areas and intersections of lines.

2.4.4 Invariances - The properties which must remain invariant under all operations are the topological relations of coincidence, incidence (points with lines) and inclusion (points in areas). Theoretically, they are preserved by all our operations, since affine transformations are (the most general) bijective mappings which preserve incidences and inclusions.

2.5 Correctness And Consistency

By correctness we mean the correlation between reality and the facts stored in the information system, mathematically described by an isomorphism between operations on reality and on the data. As an information system can not perceive reality independently, it is not possible for the system on its own to determine, nor enforce, correctness of the data stored.

Consistency is a weaker restriction on the data, requiring only that no contradiction between the facts and rules stored in the system persist. Database theory provides more elaborate discussion of the logical problems associated [GALLAIRE 1984] [REITER 1984].

The importance of consistency is not only that it helps to avoid errors during data entry and reduces the effort to maintain the data in its correct form, but more important, it maintains certain properties for the data collection upon which the exploiting procedures may rely (e.g. an area always has a closed boundary, therefore the calculation of the size of the surface is always permitted).

Consistency and correctness are independent. In fact, data may be correct (i.e. in accordance with reality) but inconsistent with some rules included in the data collection. This is especially true for so-called plausibility rules, which are often included as consistency rules but are sometimes violated in reality (e.g. if house style = 'cape cod' then number of stories less than 4)

2.6 Different Kinds Of Inconsistency

2.6.1 Well-formedness - the geometric configuration must be built according to certain rules. For example, an edge can only connect existing nodes. If those rules are violated, the object is not a graph and cannot be treated.

2.6.2 Topological Inconsistency - Topological inconsistency is encountered if the stored information on connectivity does not describe a possible configuration, i.e. if areas are not closed or the edges cross such that the graph is not planar.

2.6.3 Inconsistency Between Topology And Metric Data - The position of a node may not be consistent with the requirements of the (topologically consistent) simplicial complex. A node may be topologically treated as laying in one cell, whereas its metric position indicates its inclusion in another one. Nevertheless, the topology alone may be consistent, but contradicts the metric position.

3 PROBLEMS ENCOUNTERED IN GIS

Geographic information systems are often suplied with data from different sources. The integration of such data is sometimes more difficult than expected, and reveals inherent limitations of the system. Some systems simply treat lines and do not work with areas - the concept of area is only formed later in the mind of the observer. In such a system a simple geometric transformation may cause gaps to open between lines etc. Other systems primarily operate on closed polygon data, and combine different data using overlay methods. If the data layers combined are not precisely registered using accurate geodetic coordinates, spurious small areas (so-called gaps and slivers) are formed by the overlay calculations. Several methods are used to reduce these, but none seems to work completely satisfactorily.

3.1 Analytical Geometry

In order to treat spatial data in computer systems, analytical geometry is used. The basic objective of analytical geometry is the mapping from the two-dimensional plane to the space of tuples formed by two numbers (or a corresponding mapping for higher dimensions). The underlying number system is commonly the algebra of real numbers, though the choice of other algebras (fields) is conceptually possible.

The mapping from the two-dimensional plane to pairs of real numbers produces a homomorphism between the operations on both sides. Essentially, the classical constructions using ruler and compass are modelled by corresponding operations on real numbers. The results of all operations with ruler and compass (and also those of more general operations) can be computed analytically. Metric properties (distances, angles etc.) AND topological properties (incidences of points with lines, inclusion of points in polygons etc.) can be deduced from analytical computations.

Analytical geometry has the advantage of providing powerful mathematical tools which simplify common operations. Unfortunately, this mapping from points to tuples of reals and the corresponding operations require properties of real numbers which the approximations used in computers cannot provide.

3.1.1 Finiteness Of Computers - Number systems in computers are necessarily finite, as a finite sequence of bits can only represent a limited selection of different numbers.

Mapping of geometric data (coordinates) onto computer numbers has been a problem since the beginnings of computational geometry. However, the theoretical reasons for, and practical implications of this difficulty, and the shortcomings of most existing solutions, have only fairly recently gained widespread recognition (e.g. [WHITE 1983], [FRANK 1983], [FRANKLIN 1984], [CHRISMAN 1984]). In addition, researchers into computational geometry have also become aware that many of their theoretically well-founded algorithms are extremely difficult to implement, and require the consideration of many special cases [FORREST 1984].

3.1.2 Alternative Underlying Algebras - Franklin's [FRANKLIN 1984] contribution to this issue is that of having separated the concept of analytical geometry from real numbers, and of having examined other algebras. He has attempted to discern which number systems are most suitable for the performance of analytical geometry using a finite machine, and has found none to be satisfactory. His conclusion is that even most complex algebras, such as the calculation of exact rationals with a general root operator, are not powerful enough to represent geometry, and would be very inefficient to use.

In Franklin's opinion, the current practice of using real numbers of increasing precision to reduce the number of problem cases is 'part of the problem, not part of the solution'. Rounding errors, together with inevitable measuring errors in coordinate values, hinder geometric processing and produce artefacts (known as slivers and gaps) which must be weeded out by human operators.

In CAD systems, for example, these problems make it difficult to detect illegal intersections of solids automatically [SEGAL 1984]. Human experts are usually not hindered by inaccuracies in their understanding of spatial situations. However, the large volume of data in a GIS does not permit reliance on human intervention to maintain consistency in the database.

3.1.3 Minimal Requirements For A Number System For Analytical Geometry - Regular Euclidian geometry maintains that a line can be divided into smaller parts. In order to be able to perform such an operation in analytical geometry, we need a number system which is continuous: otherwise we reach a point at which a given small line segment can no longer be subdivided.

However, no finite set of numbers, as representable in a computer, can be used to model an infinite, continuous set of numbers. Therefore a computer system cannot be used to completely model analytical geometry based on Euclidian geometry.

In this article, we do not treat the question of what an analytical geometry using a finite number system (a 'geometry of the discrete plane') would look like. Obviously it would omit some axioms of Euclidian geometry.

3.2 Impossibility Of A Consistent Deduction Of Topology

It is a standard operation in analytic geometry to deduce a topological property from given metric data (point coordinates and line shapes). Consequently, computational geometry has developed efficient algorithms for such questions as: does a newly entered point coincide with one previously stored? What is the relative position of a point with respect to a line? (i.e. does the point lie to the right or to the left of the line?). Does a point lie inside or outside of a polygon?

In principle, there is no restriction on deducing incidence using a test on the distance between points or between a point and a line. If the distance between two points is zero, then the points are identical; if the distance between a point and a line is zero, then the point lies on the line.

However, if metric positions are expressed using a finite representation, this type of deduction is not consistently possible. For example, the conclusion that a point lies to one side of a line might not hold true after rotation or scaling (see example in [FRANKLIN 1984]).

It is not possible to decide with certainty whether a point lies on a line or whether two pairs of coordinates represent one and the same point. Additional precision in the coordinates (i.e. additional bits for the implementation of reals) can reduce the number of cases which cannot be correctly decided, but this is not a general solution to the problem.

We conclude that methods based on analytical geometry and using metric operations to deduce topological properties cannot preserve the invariance of these properties under such common transformations as rotations and scalings. Operations on spatial data in such systems therefore have a tendency to introduce topological inconsistencies, which are often referred to as 'slivers and gaps'.

3.2.1 Influence Of Measuring Errors - The finite approximations used in computer systems, and the inevitable measuring errors associated with data collection, have essentially the same effects. Point coordinates are approximations resulting from measuring operations (which are inherently subject to errors) and computations (which use some equally approximate coordinates of base points). Statistically speaking, coordinates are not 'estimable quantities', but the original measurements or adjusted measurements are.

This kind of problem is most obvious if we have to merge data from two different sources, which contain a number of common points and lines represented by slightly differing coordinates. Conventional processing is then plagued by discrepancies and 'slivers', small areas which are caused by small differences in the coordinate values used, but which do not correspond to reality. These must subsequently be eliminated in a tedious manual editing process.

The addition of estimates for the errors associated with the coordinate values can help to reduce the number of cases which cannot be easily decided, but cannot solve all such cases.

4 A TWO LEVEL APPROACH

In order to separate the problems related to metric relations and to clearly restrict their influence, we will divide the problem into two levels:

1. a layer that stores the necessary topological relations, the operations assume that the user knows the topological relations of newly stored objects

2. a layer that maintains a consistent geometric configuration, without receiving the topological relations. It deduces them from approximations to positions using 'oracles'.

5 CELL COMPLEXES

This section describes the data structure which we use to store topological relations explicitly. We call this structure a Cell Complex. Our terminology is mostly based on [MOISE 1977] and [GIBLIN 1977]. Elementary topological notions have been introduced in section 2.3. For some intuitively clear concepts (such as "in general position") we give no definitions here and refer the reader to the two references mentioned. We will first introduce the basic notions of a simplex and a simplicial complex on which the definitions of cells and cell complexes are founded. As a means of building aggregates of primitive elements, chains will be introduced at the end of this section.

All definitions refer to a two-dimensional space. However, the concepts introduced have their direct equivalents in higher dimensions. Generalization is therefore straightforward.

5.1 Simplices

A simplex is the elementary geometric object in a given dimension, i.e. a building block from which all figures in this dimension can be constructed.

The 0-dimensional building block is a point. It is called 0-simplex or sometimes vertex or node.

A 1-simplex (also called segment or edge) is the convex hull (see section 2.3) of two nodes in general position.

Similarly, a 2-simplex (also called triangle) is defined as the convex hull of three nodes in general position.

A face of an n-simplex is the convex hull of a nonempty subset of n nodes in general position (in the n-dimensional space). Thus a triangle has three edges and three nodes as its faces. Together they form its boundary.

5.2 Simplicial Complexes

As mentioned before, simplicial complexes are a generalization of graphs. They include graphs as a special case in which the elements of the complex are 0- and 1-dimensional building blocks only (nodes and edges, but no triangles). The defining characteristic of a simplicial complex is the intersection condition:

A simplicial complex is a collection K of simplices such that

(1) K contains all faces of all elements of K

(2) The intersection of two simplices in K is either empty or else a face of both simplices.

This means that if a simplex is an element of a simplicial complex then all its faces are elements as well. Second, the simplices in a simplicial complex do not "overlap".

5.3 Cells

Simplices are defined as convex hulls but we would like to represent non-convex areas and non-straight lines as well. This can be achieved by a slight generalization of the simplex concept.

5.3.1 Definition Of Cells - An n-cell is a space homeomorphic to an n-simplex. Loosely speaking, if the faces are also homeomorphic to the corresponding faces of th n-simplex, the n-cell is called a combinatorial n-cell. In the subsequent discussion, n-cell stands for combinatorial n-cell. The notions of node, edge and triangle will be taken over from the simplex terminology and used alternatively for (combinatorial) 0-, 1-, and 2-cells.

We establish the convention that every n-cell is open, i.e. does not contain its boundary. By doing so we achieve the condition that every point of the plane belongs exclusively either to a 0-cell or to a 1-cell or to a 2-cell. This disjoint partition of all points in the plane allows for easier semantics of the operations on cell-complexes.

5.3.2 Metric Description - So far, we have used only topological concepts to define n-cells. The point sets forming these cells can however be described by metric properties. For 0-cells or nodes, this is trivial: The two reals associated with a 0-simplex and consequently with a 0-cell are called the coordinates of the 0-cell.

For 1-cells or edges, we use equations to describe shape. A convenient form is the parametrization using the arc length s as a parameter. This simplifies computations and avoids the singularities which occur when using explicit equations.

The shapes and coordinates of bounding cells are sufficient for the metric description of 2-cells.

5.3.3 Orientation - Every cell has an orientation. This is quite natural for an edge, where the direction leads from the start to the end-node, making two orientations possible. We will define an orientation as an ordering of the cell's nodes, and consider all even permutations of this ordering to be equivalent.

Based on this definition, nodes have one (unique) orientation. It may be somewhat surprising that triangles (and cells of higher dimensions) have, like edges, exactly two possible orientations. For a triangle, we may define the clockwise sequence of nodes as positive orientation and associate with it a positive value for the surface area.

5.4 Cell Complexes

Cell complexes are defined as collections of cells in the same way that simplicial complexes have been defined above.

A cell complex is connected if any two given nodes are connected by a path of edges.

The join of a node n in a cell complex consists of all edges that are bounded by n.

We will require that our cell complexes be closed surfaces. Informally, this means that isolated nodes, edges or triangles are not permitted, and that every edge bounds exactly two triangles.

5.5 Chains

A path in a cell complex can be represented as a chain of edges - this is a one-dimensional chain (1-chain). The notion can be generalized to an n-chain which consist of n-cells, and to formal sums of cells which are not connected (e.g. the above-mentioned join of a node can be seen as a 1-chain).

A chain c of dimension n is defined as a formal sum

$l_1 s_1 + l_2 s_2 + l_3 s_3 + \ldots + l_n s_n$
where the li are integers (positve or negative)
and the si are all cells of dimension n.

Given a fixed set of cells, the chains form an abelian group with the operation addition, fulfilling the following axioms [GIBLIN 1977]:

A is an abelian group with elements a if:

1. for all a,b,c in A : a + (b + c) = (a + b) + c
2. there exists a neutral element 0 in A such that, for all a in A : a + 0 = 0 + a = a
3. for each a in A there exists b in A such that a + b = b + a = 0
4. for all a,b in A : a + b = b + a

A chain can represent a path of edges (and by analogy a path between nodes), and thus it is important to indicate whether we forward on the edge (from start to end) or backward (from end to start). The coefficient li is defined as 1 for forward and -1 for reverse. Note that we are permitted to travel over an edge more than once, so the coefficient of an edge may be a number other than -1, 0 or 1.

Using chains supplies us with a regular, well-understood algebraic structure on which to build more complex operations.

6 A MULTISORTED ALGEBRA FOR CELL COMPLEXES

The informally described concepts of cells and cell complexes, and the corresponding operations, can be described as a multisorted algebra or abstract data type [GOGUEN 1978] [GUTTAG 1977] [PARNAS 1972] [ZILLES 1980] [ZILLES 1984]. This is a very precise method of formally describing the meaning of data types. It is based on the familiar concept of an algebra (e.g. the algebra of real numbers) and extends it to new data types. It seems to be one of the few methods useful to formally capture the meaning (semantics) of a newly defined data type (i.e. without relying on the connotations of the names of variables and procedures).

Multisorted algebras (or abstract data types) are based on common mathematical concepts and they can be used in formal proofs to show, for example, that the defined operations preserve certain desirable properties. They are, on the other hand, similar enough to computer programming that the translation of their formal definitions into a program expressed in a procedural language (we use Pascal) is easy and straightforward. A description of an application in terms of abstract data types allows the detection of weaknesses or inconsistencies in concepts at an early stage, before any program code has been written. Our experience leads us to believe that the time and effort spent on formal definitions are easily recouped through faster program development with fewer errors.

An abstract data type (ADT) is described in three parts:
1. the sorts (data types) involved, e.g. integer, point, chain;
2. the operations, usually written as functions, together with the data types of their arguments and results;
3. the axioms, which define equivalences between sequences of operations. The axioms are understood as preceeded by all quantifiers for all variables (read "for all x, y, z... ").

This section will deal exclusively with the topological relations between cells, and will not include any provisions for metric data. The positions of points and the parameters describing the shape of a line are included in another layer of our description. This permits a complete separation of concerns, and clearly identifies the point at which the problems of realizing metric operations on finite computers arise.

6.1 Operations On The Cell Complex

The structure which we use to represent topological relations explicitly is a cell complex, as explained in a previous section. Since a cell complex is nothing more than an irregular, complete triangulation of the plane, any operations on it have to preserve this invariant, i.e. they can only densify the triangulation. (We do not yet treat deletion operations.) Thus, an algebra on cell complexes is very simple; it consists of an operation to create an initial complex and of two possible subdivision operations:

```
new:    .  -> complex
subdiv1 :   node x edge x complex -> complex
   --  an edge is split by adding a node and in consequence
   --  the two adjacent triangles (identified by their common edge)
```

-- are split into four

subdiv2 : node x triangle x complex -> complex
-- a node is added inside a triangle and in consequence
-- the triangle is split into three triangles

Figure 1: (a) subdiv1, (b) subdiv2

Indeed the last two operations can be seen as one, namely:

subdiv : cell x cell x complex -> complex

The axioms for these operations are the conditions for the simplicial (or cell) complex (see above section 5.2).

These two operations (new and subdiv) alone already constitute the complete set of operations manipulating the topological structure. The addition of an edge to a complex is split into the repeated addition of nodes which then, by the rules governing the simplicial complex, triggers the addition of the desired edge.

Additionally, some observer operations may be defined, such as the "join" which finds all edges incident with a node.

These operations are only useful if the inherent topological of the newly inserted cell are known. This is often the case for constructive operations. For example, the algorithm constructing the mid-point of a line 'knows' that the new point must lie on the line. Such knowledge is preserved in the cell complex and remains available later.

6.2 Creation Of Nodes (0-cells) And Edges (1-cells)

Subdivisions of a cell complex will require some elementary operations to create the new nodes and edges. These are defined here. Note however that they are only used by the subdivision operations: this is to assure the consistency of the complex, as otherwise arbitrary cells could be created, violating the intersection condition and/or the purity of a complex.

6.2.1 Nodes - Nodes are distinct points in which we are interested. Every node is different from any other known node (this is essentially the unique name assumption [REITER 1984] mentioned before).

```
Operations on nodes:
new:   . -> node
  -- to create a new node
equal: node x node -> boolean
  -- this is not a test on equality of coordinates, but on the
  -- equality of the entity

axiom:     let n := new ();
equal (n, n) =  true
equal (new(), new()) = false
  -- any two nodes created are different, a node is only equal to
  -- itself
```

6.2.2 Edges - An edge is a line segment connecting two nodes. It consists of an infinite number of points, but has only two nodes (i.e. distinct points of interest). Edges cannot exist without the adjacent nodes, and there may not be more than one edge between two nodes. This is asserted by the definition of equality of two edges but also by the restricted use of the new-operation in the operations on the complex

```
new: node x node -> edge
  -- create an edge between two nodes
  -- this implicitly defines the orientation of the edge
start: edge -> node
  -- get the start-node of the edge
end:   edge -> node
  -- get the end-node of the edge
equal: edge x edge -> boolean
  -- test two edges for equality

axioms:
start (new (n1, n2)) = n1
end (new (n1, n2) ) = n2
equal (e1, e2) = if (start(e1)=start(e2) and end(e1)=end(e2))
                  or (start(e1)=end(e2) and end(e1)=start(e2))
                  then true else false
  -- the equal operation on nodes is abbreviated by "="
  -- two edges are equal if they run between the same nodes
  -- this excludes multiple edges between the same pair of nodes
```

6.2.3 Generalized Cell - To simplify the description of dimension-independent operations, we introduce some generalized operations for cells. The dimension operation allows us to determine what kind of cell we are dealing with (i.e. whether it is a node, an edge, or a triangle). Another general operation (boundary) will be introduced below.

```
dimension:  cell -> integer

axioms:
   dimension (node) = 0
   dimension (edge) = 1
   dimension (triangle) = 2
```

6.3 Operations On Chains

In order to treat special aggregates of cells (e.g. the three edges bounding a triangle, or a path of edges between two nodes), we define the ADT chain. Its formal definition very much resembles an algebra of polynomials:

```
new : integer x cell -> chain
   -- create a new chain with one element
null: . -> chain
   -- make an empty chain
is_empty: chain -> boolean
   -- test if a chain is empty
dimension: chain -> integer
   -- get the dimension of the chain
sum:  chain x chain -> chain
   -- this is the addition operation of the abelian group
coeff:    cell x chain -> integer
length:    chain -> integer

axioms:
dimension (new (i,c)) = dimension (c)
  -- the dimension of a chain is the dimension of the first cell
  -- entered
dimension (null) = error
coeff (c, new (i,c)) = i
coeff (c, null) = 0
is_empty (ch1) = if coeff (c1, ch1) = 0 then true
  -- a chain is empty if the coefficient is zero for all cells
sum (ch1, ch2) = if dimension (ch1) <> dimension (ch2) then  error
  -- to add two chains of different dimensions is an error
dimension (sum (ch1, ch2)) = dimension (ch1)
  -- the dimension of the sum of chains is the same as
  -- the dimension of each of them
coeff (c1, sum(ch1,ch2)) =  coeff (c1,ch1) + coeff (c1,ch2)
length (ch1) = integer.sum for all c1: coeff (c1, ch1)

it follows:
is_empty (null) = true
is_empty (new (i,c)) = false
sum (null, ch1) = sum (ch1, null) = ch1
sum (ch1, ch2) = sum (ch2, ch1)
length (null) = 0

it does not follow:
length (sum (ch1, ch2)) = length (ch1) + lenght (ch2)
```

6.3.1 The Boundary Relation On Cells - The important relation between the cells in a cell complex is 'bounding', a generalization of the incidence relation between edges and nodes in a graph. Each cell is bounded by several cells of lower dimensions. We will see that this relation between the cells representing a configuration is sufficient for the deduction of all topological relations of the figures (points, lines and areas).

We can define an operation 'boundary' on a cell, which returns a chain of the bounding cells. The dimension of the boundary chain is always one less than the dimension of the given cell. The boundary chain of a node is empty.

```
boundary: cell -> chain
```

axioms :
dimension (boundary (c)) = dimension (c) - 1
is_empty (boundary (c)) = true if dimension (c) = 0
boundary (c) = if dimension (c) = 1 then
sum (new(-1,edge.start (c)), new (+1,edge.end (c)))

Given a chain of edges we can construct a chain of nodes by considering each edge as a chain of two nodes, the start-node with a coefficient of -1, the end-node with coefficient +1. In a similar fashion, we can construct boundaries of areas, by the enumeration of the edges used for its limit; the coefficient is +1 if, during an anticlockwise cycle around the areas, the edge is traversed in its forward direction, -1 if traversed in its reversed direction.

We can extend the boundary operation on chains as follows:

boundary: chain -> chain

axiom:
boundary (sum(ch1,ch2)) = sum (boundary (ch1), boundary (ch2))
is_empty (boundary (chain.null)) = true
is_empty (boundary (ch1)) = if dimension (ch1)=0 then true

We can test whether a chain of edges is closed by forming its boundary chain of nodes. The chain of nodes of a closed chain of edges must be empty. Such a closed 1-chain is called a 1-cycle, but the concept easily generalizes to closed 2-chains (polyhedrons).

is_cycle: 1-chain -> boolean

axiom:
is-cycle (ch) = is_empty (boundary (ch))

6.3.2 Triangles - A straightforward definition of triangles requires the concepts of chain and boundary. A triangle (2-cell) is create from a chain of three edges which its boundary. The coefficients in the chain define the orientation of the triangle.

new: 1-chain -> triangle
boundary: triangle -> 1-chain
equal: triangle x triangle -> boolean

axioms
new (ch) = error if not length (ch) = 3 or not is_cycle (ch)
boundary (new (ch)) = ch
is_cycle (boundary (new (ch))) = true
equal (t1,t2) = chain.equal (boundary (t1), boundary (t2))

6.4 Abstract Implementation

The abstract data types defined above can be combined to form the operations on the cell complex. In such an abstract implementation we explain how operations of the higher level (the cell complex) are built from the operations of the lower level (nodes, edges, chains, and triangles). Then we are able to formally prove that the axioms defined for the upper level operations follow from the axioms for the lower level operations.

Unfortunately, the proofs are presently quite tedious and lengthy, but we hope future work will provide us with more compact formulation. Work on these proofs was very instructive and draw our attention to many details of the axioms showing us how they interact.

7 EXTENDING THE ALGEBRA BY METRIC

The abstract algebra formulated for cell complexes in the previous section assumes that the topological relations between new cells and the previously established cell complex are known. It copes with the problem of storing a known topology explicitly, but it does not include operations to determine topology, e.g. to find out which subdivisions of the complex are required when storing a new figure. When a new figure is added to a configuration, the cell complex which records this figure's topological relationship to existing figures has to be updated. This process uses knowledge of the metric properties of existing cells, i.e. coordinates of nodes and shapes of edges, as well as additional information for cases in which we do not rely on metrics.

The concept determining topology in view of uncertain metric information, i.e. by using the "oracle" only once and in a potentially arbitrary way without loss of consistency, has been mentioned before. This section defines the related operations. At the same time, the steps needed to update the cell complex when storing a new point or line are explained. Axioms for the operations in this section are omitted.

7.1 Integer Vs. Real For Coordinate Representation

Coordinate values can be represented in a computer either as integer or real values. Either way, there is only a finite number of different values available. Given a number of bits per value, the same number is available whether reals or integers are used, but the distribution of these bits over the number space is different.

If we select reals, we must bear in mind that the operations on computer reals do not always follow the standard laws of arithmetic (e.g. a -b + b is not necessarily a). If we use integers, we lack a division operation. Whatever the choice, these limitations require some judicious decisions to avoid major problems. In what follows, we will assume that coordinate values are expressed as 'values' and point out where limitations of finite number schemes may cause problems.

7.2 The Operations On Points

Storing a point associates the node with a tuple of coordinate values which determine its position in metric space.

operations:
create: node x value x value -> point

7.2.1 Distance Between Points - A function to determine the distance between two points is needed. We use an approximation to the regular Euclidian distance.

```
distance: point x point -> real
```

axioms: see above section 2.3

It is presently not known, whether a Euclidian metric using

sqrt (sqr (delta x) + sqr (delta y))

as a distance function can be realized using finite approximations to values.

7.3 Lines

Lines are the metric counterpart of edges. For each line, a corresponding edge is created. These edges consist of start- and end-point, with all intermediate points being accessed using a parameter form:

```
create: point x point x form -> line
start : line -> point
end   : line -> point
point_of_line : line x real -> point
    -- this is a form for a parametric representation
    -- the real value varies between 0..1
```

7.3.1 Distance Between Line And Point - The distance between a line and a point is defined as the minimum distance between the point and a point on the line. This definition not only represents the regular concept of distance between an infinite straight line and a point, but also makes it applicable to non-infinite and non-straight lines .

```
distance: point x line -> real

axiom:
distance (p, l) = minimum (distance (start (l), end(l),
                          point_of_line (s,l))
   -- distance between line and point is defined using the minimum
```

7.4 Oracles

In order to determine whether two given positions refer to the same point or not, or to determine whether a point lies on a line, we need additional input. This can either come from a human user who knows more about the real situation, or from an algorithmic decision procedure. We rely - in absence of other information - on the distance, but we could easily include considerations for the type of data, the quality of the source, etc. Heuristics may also be used. A reasonable implementation is along the following lines: the calculated distance is compared to a threshold, and if it is larger, it is decided the the positions refer to Different points, if it is smaller then we can assume that the two positions indicate the same point.

Small variations in the coordinate values, introduced by measuring or rounding operations, may influence this decision. This does not endanger consistency, because an oracle may be used only once and the outcome of it is recorded in the way the point is stored in the cell complex. The oracle is never used if the cell complex already contains an indication and the cell complex records all previously deduced information.

Besides the decision as to whether two points are the same, we also need a decision as to whether a point lies on a line. This is also decided using distance.

same-point: position x positon -> boolean
point-on-line: positon x line -> boolean

With this limited use of the oracle, no consistency constraints are imposed on the oracle. The thresholds may be changed arbitrarily without influencing the overall consistency of the stored configurations. If other heuristic methods are used for the oracle, they may also not affect the consistency - indeed a purely random decision would still maintain consistency. However, if the decisions are made such that metric information is contradicted, we may observe contradiction between metric and topologic data (a small amount of such contradiction will be inevitable, as introduced by measuring and rounding errors).

7.5 Operations On The Cell Complex And Their Use Of Oracles

Here we define operations on the cell complex to find out about the topological relations between new figures and figures already represented in the complex. The operations assume a disjoint partition of the plane by triangles, edges and nodes: The position of a new point can lie either on a node, on an edge or inside a triangle. The same is true for the infinite number of points forming a line.

7.5.1 Determine In Which Cell A New Point Lies - We need an operation to determine the cell which occupies a given position. This is similar to a "pick" operation in a computer graphics program: its result is either a node, an edge or a triangle. If the oracle must be used, the oracle alone may indicate more than one candidate. In this case, one candidate must be selected, either by rules or by random choice. For a fast implementation, it may be reasonable to first search for coincident points; if none are found, then we search for incident lines, and only if none of these are found, we check for inclusion in a triangle.

containsPos: point x complex -> cell

Once this cell is determined, the cell complex can be subdivided accordingly using the operation introduced initially. If the cell is a node, no subdivision occurs (because a node representing this position already exists). Otherwise, depending on whether an edge or a triangle occupies the position of the new point, the corresponding subdivision operation, as defined in the previous section (see section 6.1), is invoked. Once this has been accomplished, the cell complex contains the topological relation decided by the oracle, and it will not be necessary to invoke the oracle again. The oracle operations do not contribute directly to the consistency of the subdivision operations, which are independent of them.

7.5.2 To Add A Line To A Cell Complex - In order to add new (straight) lines to our configuration, we need operations to determine their situation within the complex. The problem of storing lines is solved recursively.

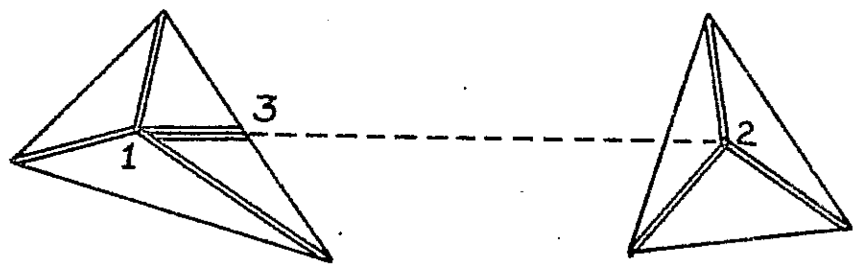

Figure 2: adding a straight line to a cell complex.

First, the two endpoints are stored using operations to determine which cells must be subdivided and to include the appropriate edges to keep the cell complex consistent.

A test then determines whether the new line is now included in the cell complex; if it is, the recursion stops.

For the start-point of the new line, we determine which cell contains the beginning of the line. This amounts to determining which cell contains the direction of the line's tangent at its start point:

containsDir: line x complex -> triangle

The triangle found must contain the start-node of the line in its boundary.

Next we determine the intersection point of the new line with the boundary of this triangle (there must be an intersection) and add this point to the cell complex.

intersect: line x chain -> point

We have now stored the new line from the start point to this new intermediate point and we apply the same steps recursively until the complete line is stored.

The two operations 'containsDir' and 'intersect' are part of the oracle. Because their implementations using approximations for coordinate values cannot be exact, we must avoid their repeated usage, as this may introduce inconsistency. In this proposal, they are only used once: when a line is initially stored. The topological information is stored in the structure of the cell complex afterwards.

8 IMPLEMENTATION STRATEGIES

A cell graph can be implemented either as a specialized datastructure using pointers or using a suitable database management system. We use the PANDA DBMS [FRANK 1982], which is based on the extended network or entity-relationship data model [CHEN 1976]. In principle, the use of a purely relational database management system should work. Performance of the necessary operations to retrieve the boundaries of cells, however, might be a problem, as these involve costly relational join operations.

The system should contain some support to quickly find cells with a given location (spatial search), as demanded by databases suitable for GIS [FRANK 1984]. We use an operation to find all cells of a given dimension which lie (completely or partially) in a certain region bounded by a rectangle with sides parallel to the coordinate axis. The result is a chain of nodes, edges or triangles, depending on the dimension.

within: rectangle x dimension x complex -> chain

The PANDA database system improves the performance of this operation by using physical clustering of data on the storage disk.

9 AREAS OF APPLICATION

Cell graphs, forming an irregular triangulation (tesselation) of a surface have a wide field of applications. Triangular irregular networks (TIN) have been studied [PEUKER 1973] for modelling of surface elevation models, and have been found to provide more accuracy with less data storage. TIN have been used in engineering applications to determine the surface runoff of urban areas in order to design sewer systems and for soil conservation studies [GRAYMAN 1978] [JETT 1979] [GRAYMAN 1982].

Cell graphs can be seen as an improved modelling structure for surfaces as they allow the inclusion of lines, whereas triangular irregular networks in the past were built only from points. It seems advantageous for many applications to include lines, as these are obviously part of the reality to be modelled. In combining data from different sources, lines often form one layer (e.g. roads), delimit areas in another layer (e.g. land use), or are break lines for the surface elevation model.

10 EXTENSIONS AND FUTURE WORK

10.1 Treatment Of Objects With Meaning

Cell graphs must be completed with a layer to treat meaning of the cells stored. The separation into one layer which treats geometry and another which deals with the meaning of the objects treated reduces the complexity of the necessary programs and makes them independent of each other. It is clearly desirable, for example, for the oracle operations to include heuristics which make use of the known meaning of the objects. The application of methods from artificial intelligence are appropriate here. We intend to build rules upon which decision can be based in a PROLOG-like language [CLOCKSIN 1981] which can access data in the database [FRANK 1984a].

10.2 Deletion Operations

We currently treat deletions as the removal of all meaning from a cell, but do not delete the cell completely. It may be wothwhile to study how the deletion of cells would work. Such operations could be implemented, either executing immediately or as a later clean-up operation, removing all unnecessary cells from the cell graph.

10.3 Dealing With Errors In Observations

Some of the problems associated with coordinate values, which are statistically speaking 'non-estimable quantities', could be resolved by storing the relative measurements between the pooints (as is legally required under the common law for property descriptions) and recomputing the coordinate values for points whenever better approximations can be obtained. In such a system, coordinate values can be change over time. If this happens, it may be necessary to check whether the expressed topological relations between nodes and edges and the corresponding relations between metric positions agree. Discrepancies may not be resolved automatically, but can be listed for individual attention by the user.

10.4 Extension To Three Dimensions

Cell graphs can be extended to three dimensions and might be usable to model the geometry of geological layers.

10.5 Provisions To Deal With Hierarchies

The major performance problem we expect stems from the division of cells into an ever-increasing number of ever-smaller pieces. This problem is not new to cell graphs, but occurs naturally in many situations where hierarchical or similar subdivisions on several levels are applied (e.g. subdivision of counties down to single parcel level). It should be possible to treat aggregates on higher levels as objects in their own right, without incurring the penalty of reconstructing them each time from all their components. We expect that the theory of partially ordered sets and lattices will provide some hints on how to solve this problem.

11 EARLIER AND RELATED WORK

The work done at the Bureau of the Census which led to the DIME and TIGRE data structures is directly related to the proposed cell graphs. Since the task of the census was to edit and maintain the large files describing statistical units throughout the U.S., the need for a mathematical concept was felt early on. In [CORBETT 1979], concepts of combinatorial topology were applied to the problem of error detection in DIME files. It is difficult to judge why this work did not have more influence on later research.

The work proposed can also be seen as a continuation of my previous work [FRANK 1983] [FRANK 1984]. It attempted to construct a mathematically sound base for modelling geometry using graph theory, and led to a classification of geographic features which can be used to express consistency constraints for spatial data bases.

12 CONCLUSION

We have presented an extension to the triangular irregular networks which includes line objects. The cell graphs are firmly based on mathematical topology: in particular, on the theory of simplices and simplicial complexes. The presentation stressed formal definitions building multi-sorted algebras (or abstract data types).

We have further discussed the limitations of geometric operations using finite computers and approximations to real numbers. The cell complex is a method wereby these limitations are confined to a number of oracles, which are implemented using approximations. This identifies levels at which limitations influence the treatment, and makes it possible to guarantee that an oracle is used only once and thus cannot introduce inconsistency into the cell graph. The cell graph stores a most complete record of the topological situation in its structure, which is independent and not influenced by metric operations such as map transformations. Therefore cell graphs avoid the 'gaps and slivers' often found during query processing in overlay systems. The treatment based on multi-sorted algebra permits formal reasoning and simplifies later the implementation.

13 ACKNOWLEDGEMENTS

The work by the first author was partially supported by a grant from Digital Equipment Corp. to the University of Maine and by a Faculty Summer Research Award from University of Maine. The Swiss Federal Institute of Technology, and especially Prof. R. Conzett helped and encouraged us. We are further grateful for extensive discussions with Bruce Palmer and Peter Krasznai, who were interested in and supportive of these ideas. BITNET/EARNET was extensively used for communication between the authors, and the importance of this facility is acknowledged. Finally, we have to thank Max Egenhofer and Linda Langley for great help with the preparation of the manuscript.

14 REFERENCES

Blumenthal, L.M, Menger, K., Studies in Geometry, Freeman and Company, San Francisco, 1970

Boyle, A.R., et al., Final Report of a Conference on the Review and Synthesis of Problems and Directions for Large Scale Geographic Information System Development, April 1983, NASA Contract NAS2-11346, available from ESRI, Redlands, CA, 1983

Chen, P., The entity-relationship model: towards a unified view of data, ACM TODS, Vol. 1, No.1, 1976

Chrisman, N.R., Concepts of space as a guide to cartographic data structures, in: Dutton, G. (ed.), First International Advanced Study Symposium on Topological Data Structures for Geographic Inforamtion System, Harvard University, Cambridge

(Mass.), Vol. 7, 1979

Chrisman, N.R., A many dimensional projection of ODYSSEY, the Harvard Geographic Information System, Harvard Laoratory for Computer Graphics and Satial Analysis, 1979

Chrisman, N.R., On storage of coordinates in geographic information systems, Geo-Processing, Vol. 2, pp. 259, 1984

Clocksin, W.F., Mellish, C.S., Programming in Prolog, Springer Verlag, New York, 1981

Corbett, J.P., Topological Principles in Cartography, Technical Paper 48, Bureau of the Census, Dec 1979

Forrest, R., et.al., Guest Editor's Indroduction too Special Issues on Computaional Geometry, ACM Transaction on Graphics, Vol. 3, No. 4, Oct 1984

Frank, A., PANDA - Pascal Network Database (in German), Report Nr. 62, Institute for Geodesy and Photogrammetry, Swiss Federal Instute of Technology, Zurich (Switzerland), 1982

Frank, A., Data Structures for Land Information Systems - Semantical, Topological and Spatial Relations in Data of Geo-Sciences, Ph. Thesis, Swiss Federal Institute of Technology, Zurich (Switzerland), 1983

Frank, A., Requirements for database systems suitable to manage large spatial databases, in: D. Marble, et al. (Eds), Proceedings of the International Symposium on Spatial Data Handling, Zurich (Switzerland), 1984

Frank, A., Extending a Database with Prolog, in: L. Kerschberg (Ed.), Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island (SC), Oct 1984

Franklin, W.M., Cartographic errors symptomatic of underlying algebra problems, in: D. Marble, et al. (Eds), Proceedings of the International Symposium on Spatial Data Handling, Zurich (Switzerland), 1984

Gallaire, H., et al., Logic and databases: a deductive approach, ACM Computing Surveys, Vol. 16, No. 2, June 1984

Giblin, P., Graphs, Surfaces and Homology, Chapman and Hall, London, 1977

Goguen, J.A., An initial algebra approach to the specification, correctness, and implementation of abstact data types, in: Yeh, R.T. (Ed.), Current Trends in Programming Methodology, Vol. 4, Data Structuring, Prentice-Hall, 1978

Grayman, W.M., Koch, P.D., ADAPT Sewer Design Geographic Information System, Proceedings of Conference on Computing in Civile Engineering, Atlanta, Georgia, 1978

Grayman, W. M., et.al., Use of Integrated Spatial Data and Modeling Capabilities for Urban Runoff Analyses, INternational Symposium on Urban Hydrology, Hydraulics, and Sediment Control, Lexington, Kentucky, 1982

Guttag, J., Abstract data types and the development of data structures, Comm. of the ACM, June 1977

Jett, S.C., et.al., Geographic Information Systems in Hydrologic Modeling, Proceedings, Hydrologic Transport Modeling Symposium, New Orleans, Louisiana, 1979

Klein, F, Vergleichende Betrachtungen ueber neuere geometrische Forschungen, Erlangen, 1872

Moise, E.E., Geometric Topology in Dimension 2 and 3, Springer Verlag, New York, 1977

Parnas, D.L., A technique for software module specification with examples, Comm. ACM, Vol. 15, No. 5, May 1972

Pazner, M., Geographic Knowledge Base Desing and Implementation, Ph. D. thesis, Univ. California at Santa Barbara, 1986 (Draft)

Peuker, T.K., Geographic data structures report after year one, ONR Contract N00014-73-C-0109, Dept. Geography, Simon Fraser University, Burnaby, B.C., 1973

Reiter, R. Towards a logical reconstruction of relational database theory, in: Brodie, M.L., et al. (Eds), On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages, Springer Verlag, New York, 1984

Segal, M., Sequin, C., Consistent Calculations for Solid Modelling, Memo Computer Science Division, University of California, Berkley, 1984

White, M.S., Tribulations of automated cartography and how mathematics help, in: Wellar, B.S. (ed.), Proceedings of the Sixth International Symposium on Automated Cartography, Ontario, Oct 1983,

Zilles, S. N., An introduction to data algebras, in: Bjorerner, D., (Ed.), Abstract Software Specifiactions, Lecture Notes in Computer Science, Vol. 86, Springer Verlag, New York, 1980

Zilles, S.N., Types, algebras and modelling, in: Brodie, M.L., et al. (Eds), On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages, Springer Verlag, New York, 1984