A Topological Data Model for Spatial Databases *

Max J. Egenhofer, Andrew U. Frank, and Jeffrey P. Jackson

National Center for Geographic Information and Analysis and Department of Surveying Engineering, 107 Boardman Hall, University of Maine, Orono, Maine 04469, U.S.A., {MAX, FRANK, JACKSON}@MECAN1.bitnet

Abstract

There is a growing demand for engineering applications which need a sophisticated treatment of geometric properties. Implementations of Euclidian geometry, commonly used in current commercial Geographic Information Systems and CAD/CAM, are impeded by the finiteness of computers and their numbering systems. To overcome these deficiencies a spatial data model is proposed which is based upon the mathematical theory of simplices and simplicial complexes from combinatorial topology and introduces *completeness of incidence* and *completeness of inclusion* as an extension to the closed world assumption. It guarantees the preservation of topology under affine transformations. This model leads to straightforward algorithms which are described. The implementation as a general spatial framework on top of an object-oriented database management system is discussed.

1. Introduction

Traditionally, applications with spatial data are based upon coordinates and an implementation of Euclidian geometry following the model of analytical geometry. While this method is being taught so convincingly in high school as *the* geometry, its implementation in a finite computer system does not capture regular concepts of analytical geometry. Problems can be observed in many areas and impede almost any modern application of CAD, VLSI, common sense physics, spatial information systems, etc. For example, software systems for Geographic Information Systems, such as ARC/INFO, demonstrate the inherent problems of a geometric model based upon Euclidian geometry:

*This paper was published in A. Buchmann, O. Günther, S. Smith, and Y.-F. Wang (eds.) *Design and Implementation of Large Spatial*, Lecture Notes in Computer Science, Vol. 409, pp. 271-286, 1989.

- Scaling of coordinates may change topology, moving points, initially close to a line, from one side to another.
- The intersection of two lines does not necessarily lie on both lines.
- The application of two inverse geometric operations generates a geometry which may differ from the original geometry (Dobkin and Silver 1988; Hofffmann 1989).
- The crucial operation of map overlaying in Geographic Information Systems introduces so-called *gaps and slivers* (Goodchild 1978) which have to be removed with computationally expensive and conceptually dubious methods, introducing errors in the data.

Finite computers cannot provide for infinite precision as is assumed for the Cartesian representation of space used in analytical geometry. The deficiencies of computer number systems and the implementations of their algebras exclude integers and floating point reals as appropriate candidates for coordinate-based geometry (Franklin 1984). Intersections which do not lie on the intersected lines (Nievergelt and Schorn 1988) are only one of many undesired outcomes.

A geometry with 'tolerances', that would permit intersections to lie a certain distance off the particular lines and still be considered 'on' the line (Greene and Yao 1986), shows strange effects through the necessary transitivity of an equivalence relation. The fact that two points A and B are equal if the distance between them is less than a certain tolerance implies by transitivity that any point in the universe is equal to any other if enough intermediate points are introduced between them—regardless of the distance between them. In actual implementations, points may 'wander' when other points close by are introduced during the computation of map overlays (Guevara 1985).

Another problem with coordinate geometry is the complexity of standard operations and the difficulty of guaranteeing that no geometric inconsistencies are overseen (Frank 1983a). In particular, objects with holes or objects separated into non-coherent parts causes problems which are difficult to treat.

Obviously, a theory for the representation of spatial data is needed that is compatible with the finiteness of computers. The development of such a coherent, mathematically sound theory—at least for the GIS field—is one of the goals being investigated by the National Center for Geographic Information and Analysis (Abler 1987). As a contribution to such a theory, the development of a general spatial data model based upon simplicial complexes is presented. This model, using topology rather than coordinates, is better-suited for the implementation in a computer because it does not rely upon the limitation of the applications of number systems in computers. Instead, it records the connection among geometric objects with respect to their neighborhood and allows user queries about neighborhood and inclusion to be processed without the need of numeric calculations.

This spatial data model differs from traditional approaches in solid modeling using constructive solid geometry or boundary representations (Requicha

1980; Requicha and Voelcker 1983). The simplicial structure partitions the space and establishes a geometric framework from which meaningful objects can be built. Their geometry is representaed by the aggregate of simplices and complexes, located in the embedding (simplicial) space. Non-spatial properties are added in semantically richer layers built on top of this basis. This separation leads to a two-level model: (1) At a geometric level, all objects are considered *cells* without any meaning about the objects they represent. This level is the geometric framework and deals with all geometric concern. All geometric operations are defined at this level. (2) Any meaningful spatial object is composed as an aggregate of geometric parts and a collection of non-spatial properties. The concept of *inheritance* (Cardelli 1984) is employed to provide geometric operations from the objects in the geometric layer to objects at the semantic level (Frank 1987; Egenhofer 1988).

This fundamental spatial data model is widely applicable. It is dimension-independent and can be used for 2-D and 3-D. Without the loss of generality, this paper is limited to a two-dimensional model. From the algorithms presented it follows that the same concepts can be applied in any higher-dimensional space as well.

The remainder of this paper is organized as follows: Section 2. presents simplices and simplicial complexes as the spatial objects of concern. *Boundary* and *co-boundary* are introduced as the fundamental operations upon complexes. A set of operations for 2-dimensional geometry inserting nodes, lines, and polygons is introduced in section 3., the algorithms of which are presented in the appendix. Results from the implementation are discussed in section 4..

2. A Model for the Representation of Spatial Data

In the mathematical theory of combinatorial topology, a sophisticated method has been developed to classify and formally describe point sets. Topology has been used for modeling spatial data and their composition for a long time. Recently, combinatorial topology was applied to spatial data models in Geographic Information Systems (GIS) (Corbett 1979; Frank and Kuhn 1986; Herring 1987), both for two-dimensional (Egenhofer 1987) and three-dimensional (Carlson 1987) geometry. The simplicity of the implementation demonstrated the simplicity of the coherent mathematical theory (Jackson 1989).

2.1 Simplex

Spatial objects are classified according to their spatial dimension. For each dimension, a minimal object exists, called *simplex*. Examples for minimal spatial objects are 0-simplices representing nodes, 1-simplices which stand for edges, 2-simplices for triangles, 3-simplices for tetrahedrons, etc.

Any n-simplex is composed of (n+1) geometrically independent simplices of dimension (n-1). For example, a triangle, a 2-simplex, is bounded by three 1-simplices (figure 1). These 1-simplices are geometrically independent if no two edges are parallel and no edge is of length 0 (Giblin 1977).

A face of a simplex is any simplex that contributes to the composition of the simplex. For instance, a node of a bounding edge of a triangle is a

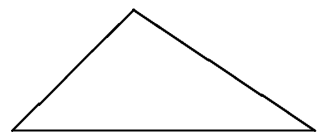

Figure 1: A 2-simplex composed of three 1-simplices.

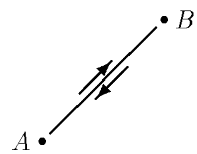

Figure 2: The two orientations of a 1-simplex.

face; another face of a triangle is any of its bounding edges. A simplex S of dimension n has $\binom{n+1}{p+1}$ faces of dimension p ($0 \leq m \leq n$) (Schubert 1968). For example, a 2-simplex has $\binom{2+1}{1+1} = 3$ 1-simplices as faces. Note that the n-simplex is a face of itself.

An ordered n-simplex s_n may be represented by its vertices in the following form:

$$s_n = \langle x_0, \cdots, x_n \rangle \tag{1}$$

For example, the two ordered 1-simplices in figure 2 can represented as

$$S = \langle A, B \rangle$$
$$T = \langle B, A \rangle$$

An *orientation* of a simplex fixes the vertices to lie in a sequence and is defined through the associated ordered simplices. The orientation of a 0-simplex is unique; the two orientations of a 1-simplex can be interpreted as the direction *from* vertex A *to* vertex B and reverse *from* B *to* A (figure 2); the orientations of a 2-simplex can be interpreted as *clockwise* or *counterclockwise*.

2.2 Simplicial Complex

A simplicial complex is a (finite) collection of simplices and their faces. If the intersection between two simplices of this collection is not empty, then the intersection is a simplex which is a face of both simplices. The dimension of a complex c is taken to be the largest dimension of the simplices of c.

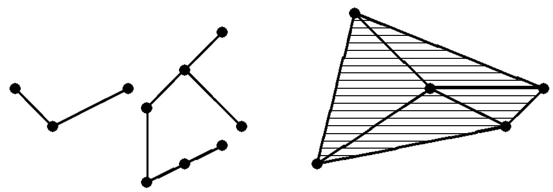

Figure 3: A 1- and a 2-complex.

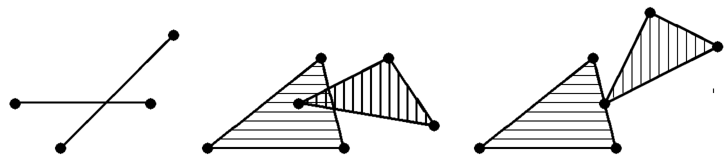

Figure 4: Three compositions which are not simplicial complexes.

The configurations in figure 3, for example, are complexes, while figure 4 shows three compositions which are not simplicial complexes. The intersection of some of their simplicies is either not a face (figures 4a and b), or not a simplex (figure 4c).

2.3 Boundary

An important operation upon an n-simplex s_n is *boundary*, denoted by ∂s_n, which determines all (n-1)-faces of s_n. The property that two successive applications of boundary give the zero homomorphism is in agreement with the geometric notion that the boundary of a simplex is a closed surface.

The algebraic interpretation of the boundary operation is particularly useful for the subsequent formal investigations. Suppose that the representation of the ordered n-simplex s_n is as introduced in equation 1 $s_n = \langle x_0, \cdots, x_n \rangle$, then the boundary of s_n is determined by

$$\partial s_n = \sum_{i=0}^{n} (-1)^i \langle x_0, \cdots, \widehat{x_i}, \cdots, x_n \rangle \tag{2}$$

where $\widehat{x_i}$ denotes that the vertex x_i is to be omitted (Schubert 1968). The bounding simplices form a chain which is an element of a free Abelian (i.e., additive) group, with $-\langle x_0, \cdots, x_n \rangle = \langle x_n, \cdots, x_o \rangle$ and $\langle x_0, \cdots, x_n \rangle - \langle x_0, \cdots, x_n \rangle = 0$. Hence, the boundary of a simplicial complex c_n can be determined as the sum of the boundaries of all its simplices s_n.

$$\partial c_n = \sum \partial s_n \text{ if } s_n \in c_n \tag{3}$$

Figure 5 illustrates the following example.

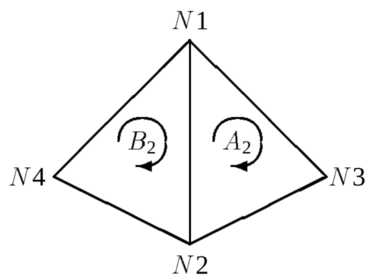

Figure 5: The boundary of the 2-complex c_2, calculated as the sum of the boundaries of the two 2-simplices A_2 and B_2.

The two neighboring 2-simplices A_2 and B_2 have the following boundaries (Table 1):

	s_2	∂s_2
A_2	$\langle N1, N3, N2\rangle$	$\langle N3, N2\rangle - \langle N1, N2\rangle + \langle N1, N3\rangle$
B_2	$\langle N1, N2, N4\rangle$	$\langle N2, N4\rangle - \langle N1, N4\rangle + \langle N1, N2\rangle$

Table 1: Simplices and corresponding boundaries illustrated in figure 5.

Then the complex C_2 formed by A_2 and B_2 has the following boundary:

$$\begin{aligned} \partial C_2 &= \partial A_2 + \partial B_2 \\ &= \langle N3, N2\rangle - \langle N1, N2\rangle + \langle N1, N3\rangle + \langle N2, N4\rangle - \langle N1, N4\rangle + \langle N1, N2\rangle \\ &= \langle N3, N2\rangle + \langle N2, N4\rangle + \langle N4, N1\rangle + \langle N1, N3\rangle \end{aligned}$$

2.4 Co-Boundary

The co-boundary of a simplex s_n, denoted by γs_n, is introduced as the set of all (n+1)-simplices which are bounded by s_n. The orientation of s_n is not significant.

$$\gamma s_n = \bigcup s_{n+1} \text{ if } s_n \in \partial s_{n+1} \tag{4}$$

For instance, the co-boundary of a 1-simplex is the set of the two bounding 2-simplices. The co-boundary of a complex c_n is then the union of the co-

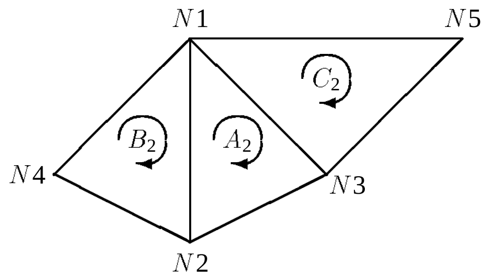

Figure 6: The 2-simplices C_2 and A_2 as the co-boundary of the 1-simplex $\langle N1, N3\rangle$.

boundaries of the n-simplices of c_n.

$$\gamma c_n = \bigcup \gamma s_n \text{ if } s_n \in c_n \tag{5}$$

Figure 6 illustrates the following example calculating the co-boundary of the 1-simplex $s_1 = \langle N1, N3\rangle$ in the 2-complex $\langle A_2, B_2, C_2\rangle$.
The simplex $\langle N1, N3\rangle$ (or its inverse $\langle N3, N1\rangle$) is contained in the boundaries of the two complexes A_2 and C_2 (Table 2):

	s_2	∂s_2	$\partial s_2 \cap s_1$
A_2	$\langle N1, N3, N2\rangle$	$\langle N3, N2\rangle + \langle N2, N1\rangle + \langle N1, N3\rangle$	$\langle N1, N3\rangle$
B_2	$\langle N1, N2, N4\rangle$	$\langle N2, N4\rangle + \langle N4, N1\rangle + \langle N1, N2\rangle$	$\emptyset$
C_2	$\langle N1, N5, N3\rangle$	$\langle N5, N3\rangle + \langle N3, N1\rangle + \langle N1, N5\rangle$	$\langle N1, N3\rangle$

Table 2: Simplices and corresponding boundaries illustrated in figure 6.

Then the co-boundary of $\langle N1, N3\rangle$ is

$$\gamma\langle N1, N3\rangle = \{A_2, C_2\}$$

2.5 Completeness Axioms

All spatial objects are located in the same world which is represented by the fundamental geometric structure and closed in analogy to the closed world assumption (Reiter 1984) of non-spatial mini-worlds. Within this fundamental level, two completeness principles are guaranteed (Frank 1983a; Frank and Kuhn 1986):

- Completeness of incidence: The intersection of two n-simplices is either empty or a face of both simplices. Hence, no two geometric objects must exist at the same location. For example, though an edge may represent both a part of a state boundary and a part of the border of a nation, the geometry of the edge will be recorded only once.

- Completeness of inclusion: Every n-simplex is a face of an (n+1)-simplex. Hence, in a 2-dimensional space every node is either start- or end-node of an edge, and every edge is the boundary of a triangle.

The simplicial structure is a complete partition of the space. In the case of a 2-dimensional world, this structure establishes a triangulated irregular network (TIN) (Peucker and Chrisman 1975). The characteristics of this space are similar to subdivisions established by regular tilings of space; however, the simplicial structure is more flexible and can represent exact locations and exact topology.

Hierarchical structures can be superimposed (Bruegger 1989), such that a structure like a quadtree (Samet 1984) results in the two-dimensional space. Other possible structures currently investigated at the University of Maine are posets or lattices which are frequently necessary to model geographic data appropriately (Kainz 1988, Saalfeld 1985). These non-hierarchical structures permit that the same geometric parts may be components of several objects. For example, the polygon of a state consists of 2-cells which are also part of its counties.

3. Fundamental Operations

A characteristic of the simplicial algebra is its simplicity: only a small set of operations is necessary. The operations are closed within the simplicial structure, i.e., an operation manipulating a simplex can produce only a spatial object that is a simplicial complex. Updates like the insertion of a new node must be consistent, i.e., the structure of the triangulated network and the completeness axioms must be guaranteed after each modifiation. The principle of all operations is that they guarantee consistency, i.e., each successfully completed operation will preserve the simplicial structure. The fundamental operations for inserting new 0-, 1-, and 2-cells in a 2-dimensional simplicial space will be presented and illustrated step by step.

Subsequently, it is assumed that the universe is established by an outer void and no geometry will exist outside of this universe. By this assumption, all operations are reduced to the insertions of geometric objects inside the universe.

The underlying data structure establishes the relationships among 0-, 1-, and 2-cells: each 1-cell is bound by two 0-cells; each 2-cell is bound by three 1-cells; a 0-cell bounds several edges; and each 1-cell bounds two 2-cells. This structure allows for the derivation of adjaceny through the operations boundary and co-boundary. These operations are fundamental for the following algorithms and their implementation is assumed.

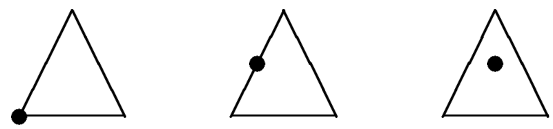

Figure 7: The three cases of node insertion: (a) on a 0-cell, (b) on a 1-cell, and (c) in a 2-cell.

3.1 Node Insertion

The addition of a 0-cell c_0 to an existing network may appear as three possible cases: (1) c_0 coincides with an existing 0-cell; (2) c_0 falls on an existing 1-cell; or (3) c_0 falls within an existing 2-cell. Figure 7 shows these three constellations.

3.1.1 Node Coincidence

The first case is trivial because the object to be added exists already. It is part of the assurance of the completeness of incidence according to which each geometric object cannot exist twice.

3.1.2 Node on Edge

The addition of c_0 which falls on an existing 1-cell c_1 involves in general two 2-cells. The insert operation can be broken down into the following steps:

- Storage of c_0 (figure 8a).
- Connceting c_0 with each 0-cell of the co-boundary of c_1 by storing a new 1-cell (figure 8b).
- Insertion of 2-cells, each made up of two newly inserted 1-cells and one in the boundary of the co-boundary of c_1 (figure 8c).
- Deletion of the 2-cells which formed the co-boundary of c_1 (figure 8d).
- Deletion of the 1-cell c_1 (figure 8e).

3.1.3 Node in Polygon

The addition of a 0-cell c_0 which falls within an existing 2-cell c_2 consists of the following operations:

- Storage of the 0-cell c_0 (figure 9a).
- Insertion of the three 1-cells connecting c_0 with the 0-cells in the boundary of c_2 (figure 9b).

- Insertion of three 2-cells, each made up of one bounding 1-cell of c_2 and two newly inserted 1-cells (figure 9c).
- Deletion of c_2 (figure 9d).

3.2 Line Insertion

The insertion of a 1-cell is a recursive operation. It can be decomposed into the following steps:

- Insertion of the two bounding 0-cells A_0 and B_0 (figure 10a).
- Retrieval of the smallest 2-complex C_2 inside of which A_0 is completely contained (figure 10b). C_2 is determined with the application of the co-boundary operation, first around A_0 which results in the 1-cells bounded by A_0, and then for the 1-cells yielding the 2-cells.
- Intersection of the boundary of C_2 with the line from A_0 to B_0 (figure 10c) and insertion of the resulting 0-cell D_0 (figure 10d).
- Recursively repeating the retrieval of the boundary of the 2-complex around the node inserted last, the intersection with the line connecting the start and end node of the 1-cell, and the storage of the intersection as new node (figure 10e and f) until the intersection coincides with the end node (figure 10g).

3.3 Polygon Insertion

In a two-dimensional model, polygons are described by a sequence of 1-cells which must form one or more closed 1-spheres, representing its boundary. The insertion of a polygon is initially defined by the insertion of its boundary. This first step is accomplished by adding 1-cells using the operation described above. Figure 11 shows the insertion of a triangle.

At this stage, the simplicial structure is sound; however, there are no links between the 2-complex and the 2-simplicies of which it is comprised. The correct association between the complex and the simplices is accomplished with the following recursive operation:

- Selection of the inner 2-cell of an arbitrary 1-cell in the boundary (figure 12a).
- Reducing the set of bounding 1-cells left by the found one.
- Adding the found 1-cell to the set of visited 1-cells.
- Searching the next 1-cells in the boundary of the current 2-cell without considering already visited and bounding 1-cells.
- Searching the neigboring 2-cells which were not yet visited (figure 12b).
- Adding the found 2-cells to the result.

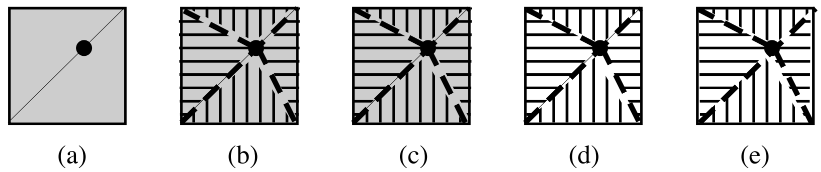

Figure 8: The insertion of a node on an existing edge.

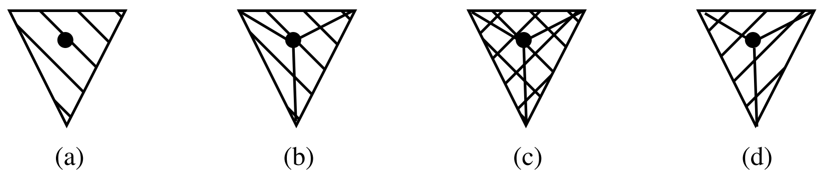

Figure 9: The insertion of a node in an existing triangle.

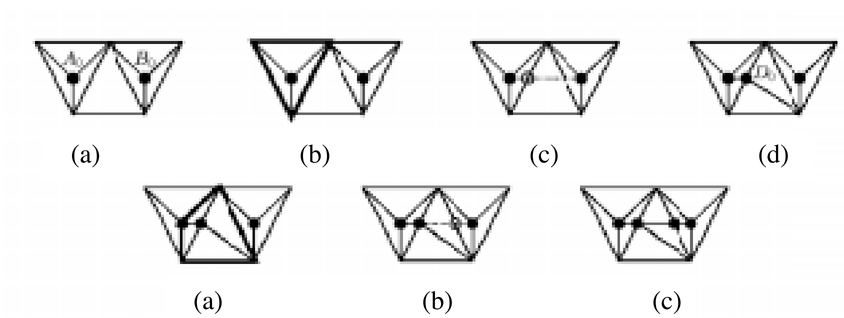

Figure 10: The insertion of a n1-cell.

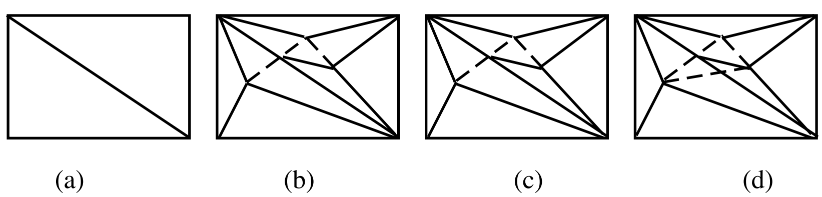

Figure 11: Part 1 of the nsertion of a 2-complex: boundary insertion.

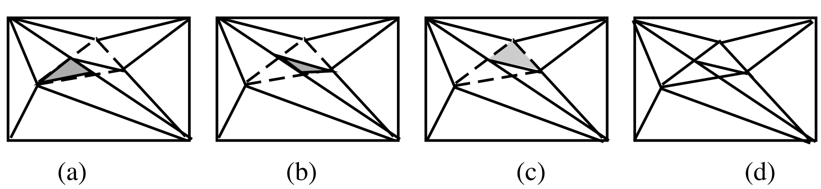

Figure 12: Part 2 of the nsertion of a 2-complex: aggregation of 2-simplices.

- Recursively searching for the next neighboring 2-cells through the boundary operator until all 1-cells have been visited (figure 12c).

The set of bounding 1-cells is not empty if the newly added 2-cell consists of several, non-coherent parts; therefore, the algorithm must loop until the last 1-cell left in the set of bounding cells is processed.

4. Implementation

The simplicial data strucutre was implemented on top of PANDA (Egenhofer and Frank 1989), an object-oriented database management system. A particular component of PANDA is the Field Tree (Frank 1983b), a spatial indexing structure organizing spatial data according to their adjacency, and supporting for spatial search techniques and fast retrieval. The spatial search technique was employed for the determination of coincidence of 0-cells.

The implementation gave evidence that a database-like system is necessary for an appropriate and efficient implementation of the operations. In particular, frequent calculations of boundary and co-boundary can be achieved efficiently only if a database-like structure exists.

The transaction concept of database management systems is employed transfering the geometric structure in an atomic operation from one into another consistent state (Härder and Reuter 1983). Operations which are—for whatever reason—unsuccessful, are aborted and the simplicial structure will be reset to its initial state before the operation.

The implementation verified the assumption that a large number of small 2-cells is created. From loading one layer of a USGS 7 1/2 minute topographic map, a linear increase of 1-cells by a factor of three was counted compared to the data set.

5. Conclusion

A spatial data model has been presented which is based upon the mathematical theory of simplices and simplicial complexes. These principles of combinatorial topology have been made available to a wide range of applications with spatial data, such as CAD/CAM and Geographic Information Systems. The major contribution of this approach is that inherent problems of implementations of Euclidian geometry are overcome by recording topological properties explicitly.

The simplicial model structures the embedding space in cells, from which meaningful spatial objects can be composed. Completeness of incidence and completeness of inclusion guarantee that the space contains each cell only once and that no isolated cells exist. Only a small set of geometric operations is neccessary for the manipulation of the data collection. Operations and algortihms were presented for inserting 0-cells, 1-cells, and 2-cells in a two-dimensional world.

The simplicial structure was implemented on top of an object-oriented database management system. The simplicity of the implementation demonstrated the value of a coherent mathematical theory.

6. Acknowledgement

This research was partially funded by grants from NSF under No. IST 86-09123 and Digital Equipment Corporation (Principal Investigator: Andrew U. Frank). Jeffrey P. Jackson was supported by an Undergraduate Research Experience grant from NSF under No. IRI 86-09123. The support from NSF for the NCGIA under grant number SES 88-10917 is gratefully acknowledged. Thanks to Werner Kuhn, currently at ETH Zürich, for his valuable contribution in the initial stage of this work. Robert Franzosa improved our mathematical skills in algebraic topology. David Pullar and Eric Carlson helped with the implementation.

Appendix

7. Algorithms for Fundamental Operations

The following operations act upon 0-, 1-, and 2-cells. Chains are a data structure for cells and form an Abelian group with an operation for addition. Sets and its conventional operations are prerequisits. Coercion from set to cell and from chain to set are assumed.

Database operations exist to *store* an n-cell, and *access* its n-simplices, boundary, and co-boundary. *Have* operations store a cell only if it did not exist before. *Make* operations link the a chain of (n-1)-cells and create an n-cell.

7.1 Boundary

```
boundary (c: cell): chain ==
  result := emptyChain;
  FOR EACH simplex IN c DO
    FOR EACH boundingSimplex IN simplex DO
      addCellToChain (result, boundingSimplex,
        getOrientation (simplex, c));
```

7.2 Co-Boundary

```
coBoundary (c: cell): set of cells ==
  result := emptySet;
  FOR EACH simplex IN c DO
    FOR EACH boundedSimplex OF simplex DO
      addCellToSet (result, boundedSimplex);
```

7.3 Node Insertion

7.3.1 Node On Node

```
storeC0Onc0 (c0: 0-cell): boolean ==
  location (retrieveC0 (location (c0))) = location (c0);
       -- True if a 0-cell already exists at the same location.
```

7.3.2 Node On Edge

```
storeC0OnC1 (c0: 0-cell, c1: 1-cell): 1-cell ==
  storeC0 (c0);
        -- Storage of c0.
  FOR EACH simplex IN boundary (coBoundary (c1)) DO
    makeC1 (simplex, c0);
        -- Connecting c0 with each 0-cell of the
        --  co-boundary of c1 by storing a new 1-cell.
  c1Set := coBoundary (c0) + boundary (coBoundary (c1));
  FOR EACH closedChain IN c1Set DO
    makeC2 (closedChain);
        -- Insertion of four new 2-cells
  FOR EACH c2 in coBoundary (c1) DO
    deleteC2 (c2);
        -- Deletion of the 2-cells which formed the
        --  co-boundary of c1.
  deleteC1 (c1);
        -- Deletion of the split 1-cell.
```

7.3.3 Node In Polygon

```
storeC0inC2 (c0: 0-cell, c2: 2-cell): 0-cell ==
  storeC0 (c0);
        -- Storage of the 0-cell c0.
  FOR EACH simplex IN boundary (c2) DO
    makeC1 (simplex, c0);
        -- Insertion of the three 1-cells connecting c0
        --  with the 0-cells in the boundary of c2.
    c1Set := coBoundary (c0) + boundary (c2);
  FOR EACH closedChain IN c1Set DO
    makeC2 (closedChain);
        -- Insertion of three 2-cells, each madeup of
        --  one bounding 1-cell of c2 and two newly
        --  inserted 1-cells.
  deleteC2 (c2);
        -- Deletion of c2.
```

7.4 Line Insertion

```
inkC0 (c0Start, c0End: 0-cell, c1Set: set of 1-cells) ==
  c1 := boundary (coBoundary (coBoundary (c0)));
        -- Retrieval of the 2-cell around the first
        -- 0-cell stored and calculation of its boundary.
  s0 := storeC0onC1 (intersection
          (makeLine (c0Start, c0End), c1), c1);
        -- Intersection of the boundary with the line
        --  from c0Start to c0End.
  c1Set := c1Set + getC1Between (s0, c1);
    IF NOT c0Equal (s0, c0End)
      THEN linkc0 (s0, c0End, c1Set);
        -- Recursively repeating the retrieval of the
        --  boundary around the node inserted last,
```

```
        --  the intersection with the line,
        --  and the storage of the intersection as new node.

toreC1 (c0Start, c0End: 0-cell): set of 1-cells ==
  result := emptySet;
  haveC0 (c0Start);
        -- Insertion of the two bounding 0-cells
  havec0 (c0End);
        --  c0Start and c0End.
  linkc0 (c0Start, c0End, result);
        -- Connecting c0Start and c0End.
```

7.5 Polygon Insertion

```
mushroom (c1Memory, c11Memory, c1Set: set of 1-cells, c1: 1-cell,
          c2: 2-cell, c2Memory: set of 2-cells) ==
  c11Memory := c11Memory - c1;
        -- Reducing the set of bounding 1-cells
        --  left by the found one.
  c1Memory := c1Memory + c1;
        -- Adding the found 1-cell
        --  to the set of visited 1-cells.
  nextC1 := boundary (c2) - c1Set - c1Memory;
        -- Searching the next 1-cells in the boundary of
        -- the current 2-cell without considering already
        -- visited and bounding 1-cells.
  IF nextC1 <> empty THEN
    coB := coBoundary (c1) - c2Memory;
        -- Searching the neigboring 2-cells which were
        --  not yet visited.
    c2Memory := c2Memory + coB;
        -- Adding the found 2-cells to the result.
    mushroom (c1Memory, c11Memory, c1Set, nextC1, coB, c2Memory);
        -- Continuing recursively.


storeC2 (c1set: set of 1-cells): set of 2-cells ==
  result := emtpySet;
  FOR EACH c1 IN c1set DO
        -- Storage of the bounding edges.
    storeC1 (getStart (c1), getEnd (c1));
  c1Memory := c1set;
  c11Memory := c1Set;
  WHILE c11Memeory <> empty DO
    b1 := getFirst (c11Memory);
        -- Selection of the 1-cell in the boundary.
    c2 := coBoundaryPos (b1, orientation (b1, c11Memory));
        -- Getting the ‘inner’ 2-cell.
    mushroom (c1Memory, c11Memory, c1Set, b1, c2, result);
        -- Mushrooming to collect all 2-cells.
```

References

R. Abler (1987) The National Science Foundation National Center for Geographic Information and Analysis. *International Journal of Geographical Information Systems*1(4): 303-326.

B. Bruegger and A. Frank (1989) Hierarchies over Topological Data Structures, *ASPRS-ACSM Annual Convention*, Baltimore, MD, pp. 137-145.

L. Cardelli (1984) A Semantics of Multiple Inheritance. in: G. Kahn, D. McQueen, and G. Plotkin (Eds). *Semantics of Data Types.* Lecture Notes in Computer Science 173: 51-67. New York, NY, Springer-Verlag.

E. Carlson (1987) Three Dimensional Conceptual Modeling of Subsurface Structures. in: *ASPRS-ACSM Annual Convention*, Baltimore, MD.

J. P. Corbett (1979) *Topological Principles of Cartography*. Technical Report 48, Bureau of the Census, Department of Commerce.

D. Dobkin and D. Silver (1988) Recipes for Geometry and Numerical Analysis, Part 1: An Empirical Study, *Fourth ACM Symposium on Computer Geometry*, pp. 93-105.

M. Egenhofer (1987) Appropriate Conceptual Database Schema Designs For Two-Dimensional Spatial Structures. *ASPRS-ACSM Annual Convention* 5: 167-179. Baltimore, MD.

M. Egenhofer (1988) *Graphical Representation of Spatial Objects: An Object-Oriented View*. Orono, ME, Department of Surveying Engineering, University of Maine.

M. Egenhofer and A. Frank (1989) PANDA: An Extensible DBMS Supporting Object-Oriented Software Techniques. in: T. Hrder (Ed.). *Database Systems in Office, Engineering, and Science.* Informatik-Fachberichte 204: 74-79. New York, NY, Springer-Verlag.

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

A. Frank (1983b) *Problems of Realizing LIS: Storage Methods for Space Related Data: The Field Tree.* Technical Report, Institute for Geodesy and Photogrammetry, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

A. Frank (1987) Overlay Processing in Spatial Information Systems. in: N. Chrisman (Ed.). *AUTO-CARTO 8, Eighth International Symposium on Computer-Assisted Cartography*: 16-31. Baltimore, MD.

A. Frank and W. Kuhn (1986) Cell Graph: A Provable Correct Method for the Storage of Geometry. in: D. Marble (Ed.), *Second International Symposium on Spatial Data Handling*, Seattle, WA, pp. 411-436.

W. R. Franklin (1984) Cartographic Errors Symptomatic of Underlying Algebra Problems. *International Symposium on Spatial Data Handling*: 190-208. Zurich, Switzerland.

P. J. Giblin (1977) *Graphs, Surfaces, and Homology*. New York, NY, Halsted Press, John Wiley and Sons.

M. Goodchild (1978) Statistical Aspects of the Polygon Overlay Problem. in: G. Dutton (Ed.). *Harvard Papers on Geographic Information Systems* Vol. 6. Reading, MA, Addison-Wesley.

D. Greene and F. Yao (1986) Finite-Resolution Computational Geometry, *27th IEEE Symposium on the Foundations of Computer Science*, pp. 143-152.

J. A. Guevara (1985) A Fuzzy and Heuristic Approach to Segment Intersection Detection and Reporting. *AUTO-CARTO 7*. Washington, D.C.

T. Härder and A. Reuter (1983) Principles of Transaction-Oriented Database Recovery. *ACM Computing Surveys* 15(4).

J. Herring (1987) TIGRIS: Topologically Integrated Geographic Information Systems. in: N. Chrisman (Ed.), *AUTO-CARTO 8 Eighth International Symposium on Computer-Assisted Cartography*, Baltimore, MD, pp. 282-291.

C. Hoffmann (1989) The Problems of Accuracy and Robustness in Geometric Computation. *IEEE Computer* 22(3): 31-42.

J. Jackson (1989) Algorithms for Triangular Irregular Networks Based on Simplicial Complex Theory. *ASPRS-ACSM Annual Convention* 4: 131-136. Baltimore, MD.

W. Kainz (1988) Application of Lattice Theory to Geography. in: D. Marble (Ed.). *Third International Symposium on Spatial Data Handling*: 135-142. Sydney, Australia.

J. Nievergelt and P. Schorn (1988) Line Problems with Supra-Linear Growth (in German). *Informatik Spektrum* 11(4).

T. Peucker and N. Chrisman (1975) Cartographic Data Structures. *The American Cartographer* 2: 55-69.

R. Reiter (1984) Towards a Logical Reconstruction of Relational Database Theory. in: M. Brodie, J. Mylopoulos, and J. Schmidt (Eds). *On Conceptual Modelling*: 191-233. New York, NY, Springer-Verlag.

A. Requicha (1980) Representations of Rigid Solids: Theory, Methods, and Systems. *ACM Computing Surveys* 12(4): 437-464.

A. Requicha and H. Voelcker (1983) Solid Modeling: Current Status and Research Directions. *IEEE Computer Graphics and Applications* 3(7): 25-37.

A. Saalfeld (1985) Lattice Structures in Geography, *Auto-Carto 7*, Washington, D.C., pp. 482-489.

H. Samet (1984) The Quadtree and Related Hierarchical Data Structures. *ACM Computing Surveys* 16(2): 187-260.

H. Schubert (1968) *Topology*. Boston, MA, Allyn and Bacon, Inc.