Finite-Resolution Simplicial Complexes

Werner Hölbling, Werner Kuhn, Andrew U. Frank

Department of Geoinformation
Technical University Vienna
Gusshausstrasse 27-29, A-1040 Vienna (Austria)
frank@geoinfo.tuwien.ac.at

Abstract

Simplicial Complexes are used to model topology in Geographic Information Systems (GIS). Line intersection is an essential operation to update them. We introduce a finite-resolution line intersection method, called Zero Order Intersection, and apply it to simplicial complexes.

Any reliable implementation of a line intersection algorithm has to address the limitations of a discrete computational environment. If handled improperly, finite representation can cause drifting lines and similar effects in otherwise topologically consistent data. The Zero Order Intersection method is designed to avoid such inconsistencies. Its application to simplicial complexes results in the Discrete Simplicial Data Model which guarantees consistency and reliability of topological queries within a GIS.

Keywords

Geographic Information Systems, finite resolution, line intersection, topological data models, discrete simplicial complex.

1. INTRODUCTION

This paper focuses on adapting the insertion of line segments into a simplicial complex from the continuous to the discrete domain. The result is a reliable and robust data model for storing topological relations within a GIS, called the Discrete Simplicial Data Model (DSDM). It is based on a new method for line intersection in the discrete plane.

The Simplicial Data Model (SDM) (Frank and Kuhn, 1986) uses simplicial complexes (Giblin, 1977); (Paoluzzi et al., 1993) for modeling topology in any dimension within a Geographic Information System. In the two-dimensional space considered here, it forms an algebra on the topological primitives node, edge and trigon. Line intersection and other standard geometrical operations are used to implement it (Egenhofer et al., 1990).

Any implementation of a spatial data model is based on a grid, because numbers can only be represented with finite resolution in computer memory. Intersection algorithms often do not handle this limitation properly. As a consequence, unexpected topological inconsistencies can occur. Among them are point inversion and drifting line segments (Greene and Yao, 1986). Franklin (1984) has shown that conventional approaches like rounding to the nearest grid point or tolerances ("Epsilon-geometry") reduce the number of topological inconsistencies, but cannot avoid them completely.

Drifting line segments are a typical example of an unexpected behavior caused by continuum-based algorithms.

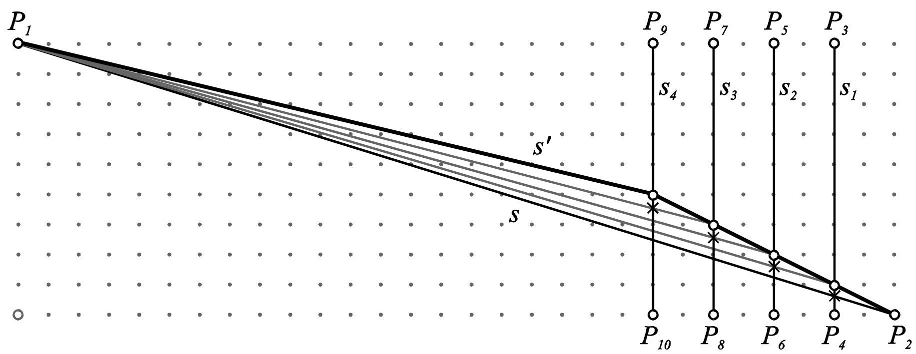


Fig. 1.1 Drifting line segments

The straight line segment s is successively intersected by the vertical straight line segments s_1, s_2, s_3, s_4 and s_5 in figure 1.1. As the intersection point between s and s_1 is rounded to the nearest grid point, the resulting segments of s are not aligned. After intersecting the longer segment with another vertical line segment s_2 and then with s_3, s_4 and s_5, we get the final scene. It is obvious, that the segments originating from s keep drifting away from it.

Users of GIS relying on topologically consistent data models face the following problem: The system is capable of storing topological information consistently, but repeated updates can introduce inconsistencies over time. For example, if a point is inserted in between an original segment and the current segment representation, the point changes its relative position from the left to the right side.

Dan Greene and Frances Yao have suggested a solution for the intersection of a set of line segments on a grid (Greene and Yao, 1986). Their idea is to avoid drifting line segments by containing them within a small neighborhood, called envelope. This envelope consists essentially of those grid points situated immediately above or below the line segment. The original line segment is replaced by its so called redrawing.

The method of Greene and Yao has already been applied to the REALM spatial data model (Güting and Schneider, 1993). However, it proved unsatisfactory for the Simplicial Data Model for two reasons:

- It produces a lot of additional line segments;
- Topological inconsistencies still remain within the envelopes.

While the first disadvantage could be accepted, the second one can not, as reliability and robustness is the primary goal of the Simplicial Data Model.

We present an alternative finite-resolution intersection method, the Zero Order Intersection. The order of a segment indicates how often a segment has been divided after its creation. Intersections with zero order segments do not cause drifting lines. The core of the method is to maintain a reference between a higher order segment and its zero order predecessor. A similar solution for the general line segment intersection problem has been developed in parallel by Guibas and Marimont (Guibas and Marimont, 1995). Zero order segments are called ursegments there.

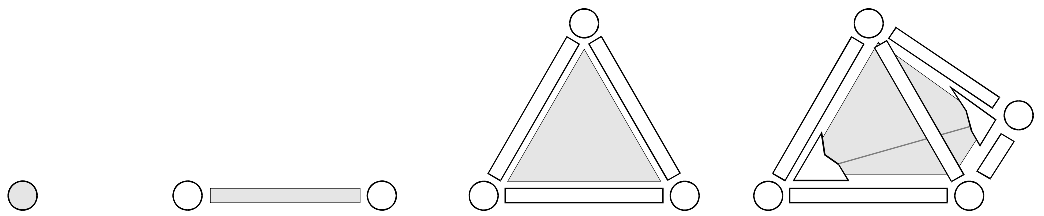
The paper is organized as follows: Section 2 gives an overview of the Simplicial Data Model. Sections 3 and 4 discuss the finite-resolution intersection method of Greene and Yao and difficulties applying it to simplicial complexes. Section 5 introduces the alternative Zero Order Intersection method. Section 6 applies the method to simplicial complexes. Conclusions are drawn in chapter 7.

2. The Simplicial Data Model

The Simplicial Data Model (SDM) applies the mathematical theory of simplexes and simplicial complexes to the modeling of geometric objects in a GIS. The goal is to establish a geometric database handling topological relations between the objects in a consistent manner. This section explains the basic concepts and operations of the topological part.

2.1 Simplex

An *n*-simplex is the convex hull to *n*+1 points in general position for dimension *n*. A 0-simplex is here called *node*, a 1-simplex *edge* and a 2-simplex *trigon*. A simplex is an elementary building block for its dimension and is used to form objects of higher complexity (Giblin, 1977).



0-simplex (node) **1-simplex (edge)** **2-simplex (trigon)** **3-simplex (Tetrahedron)**

Fig. 2.1 Simplexes

Each *n*-simplex is bounded by *n*+1 geometrically independent simplexes of dimension *n*-1, called *faces*. Simplexes are conventionally defined as open, i.e. without their faces. An edge does not contain its start and end node; a trigon does not contain its boundary edges.

2.2 Boundary, Co-Boundary

Topological relations (incidence, adjacency) represented by simplexes can be determined by the boundary and co-boundary operation.

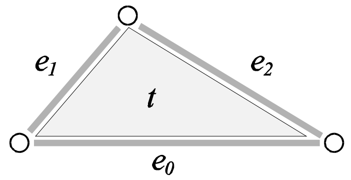
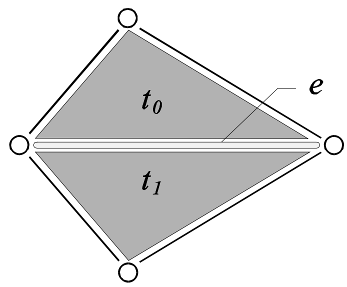


Fig. 2.2 Boundary of t



Co-Boundary of e

Boundary yields all simplexes of dimension *n*-1 which bound a simplex of dimension *n*. Co-boundary is the inverse operation, returning all simplexes of dimension *n*+1 that are bounded by a given simplex of dimension *n*.

2.3 Simplicial Complex

A finite set of *i*-simplexes (*i*=0..*n*) is called a simplicial complex $\boldsymbol{K}^n$, if it fulfills the following conditions:

completeness of incidence

The intersection of two closed simplexes is either empty or a face of both simplexes; i.e. no two simplexes of a simplicial complex overlap.

completeness of inclusion

If a simplex is part of a simplicial complex, all of its faces are, too. As a consequence, each node must be start or end node of an edge and each edge must be boundary of two trigons within a two-dimensional complex.

For the special case of closed surfaces, forming a partition, each geometrical point defined over a simplicial complex belongs to exactly one simplex. There is no undefined region inside the hull of a simplicial complex.

2.4 Operations on Simplicial Complexes

Geometric operations on simplicial complexes are straightforward, because of the convexity of simplexes. Most of them work recursively and locally. Therefore, the costs in time and space grow linearly with the number of inserted simplexes. Operations must not violate the consistency constraints of simplicial complexes. The result of any operation applied to a simplicial complex has to be another simplicial complex. Fundamental operations to build up a simplicial complex are subdivision of edges and subdivision of trigons. Nodes cannot be subdivided.

Subdivision of Trigons

A node subdivides a trigon t by three new edges into three new trigons t_1, t_2 and t_3. The edges connect the subdividing node with the trigon boundary nodes.

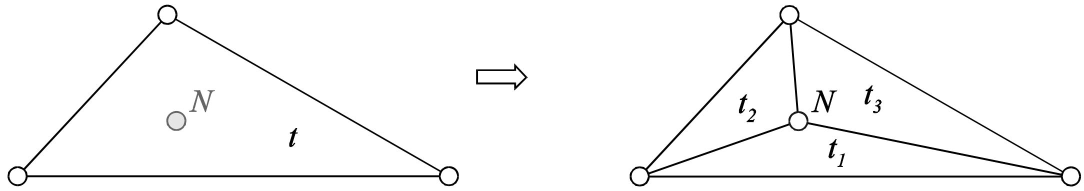


Fig. 2.3 Subdivision of trigon t

Subdivision of Edges

A node subdivides an edge by replacing it with two edges. Two another edges subdivide the trigons adjacent to the original edge into four new trigons.

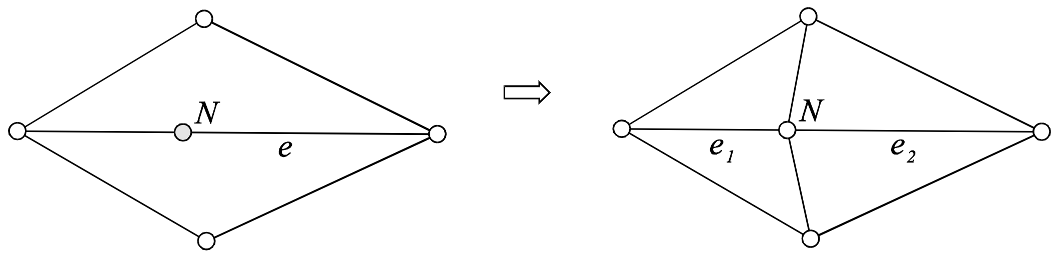


Fig. 2.4 Subdivision of edge e

The subdivision operations form the topological part of the point and line insertion operations of the Simplicial Data Model. How they are performed in detail is explained in section 6.

3. Initial Approach: Method of Greene and Yao

The method of Greene and Yao (Greene and Yao, 1986) solves the complete intersection of a set of straight line segments in discrete two-dimensional space. It is guaranteed that any resulting segment (called *redrawing*) remains within a small region (called *envelope*) of its original segment.

3.1 Envelopes

Given is a straight line segment *s* from *a* to *b* in two-dimensional discrete space. The envelope *E(s)* of the line segment *s* is the union of the set *A(s)*, containing grid points situated immediately above, below or on *s*, with the set of auxiliary points *B(s)*.

$$E(s) = A(s) \cup B(s)$$

Auxiliary points are grid points whose left and above neighbor grid point are both in *A(s)*. They are needed because intersection points falling exactly into the middle of four neighboring grid points are rounded to auxiliary points per convention.

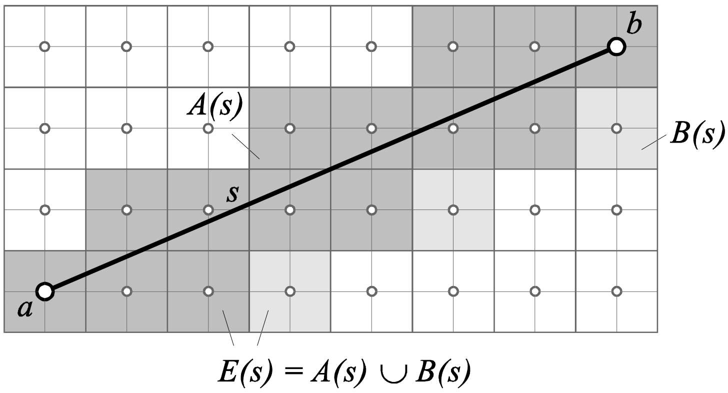


Fig. 3.1 Envelope of a straight line segment

Four squares of half unit length, situated around the same grid point, form a tile of the discrete plane. Any point within a tile is rounded to the grid point.

The envelope of a polygonal path (s_1, s_2, ..., s_n) is defined as the union of the envelopes of the individual segments:

$$E(s_1, s_2,..., s_n) = E(s_1) \cup E(s_2) \cup ... \cup E(s_n)$$

3.2 Hooked lines

A hook is a (small) vector, joining a potential intersection point situated on a line segment with a nearby grid point.

$$h_k = \langle p_k , q_k \rangle$$

A line segment together with its hooks is called a *hooked line* (Fig. 3.2).

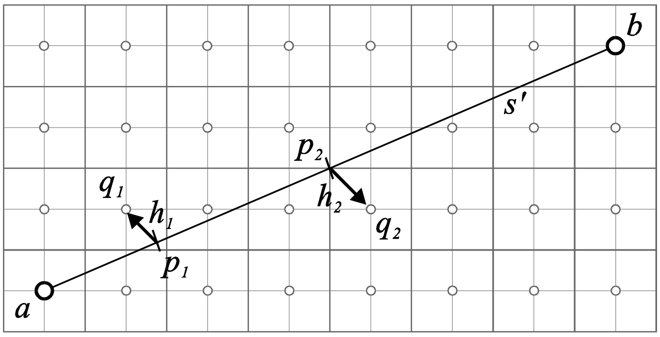


Fig. 3.2 Hooked line

After finding all intersection points of a set of line segments one has to determine potential intersections between hooks and line segments, too.

3.3 Redrawing of hooked lines

The start and end point of a hooked line must not be reconnected simply from one hook head to another. Otherwise, a grid point *c* situated between the original line segment *s* and its reconnection *r(s)* can change its relative position from the right side to the left (Fig. 3.3).

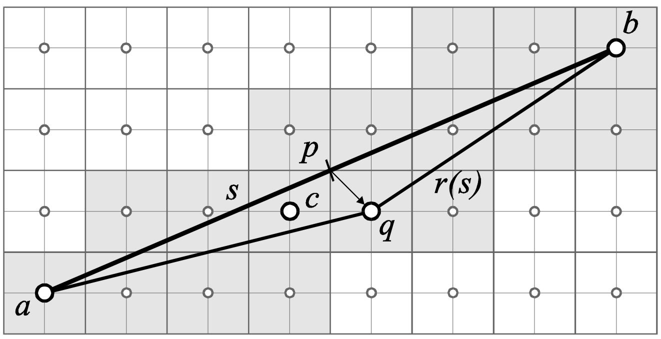


Fig. 3.3 Not allowed redrawing of a hooked line

Instead Greene and Yao give the following definition: The redrawing of a hooked line is the shortest path which lies completely inside the envelope and goes through the heads of the hooks.

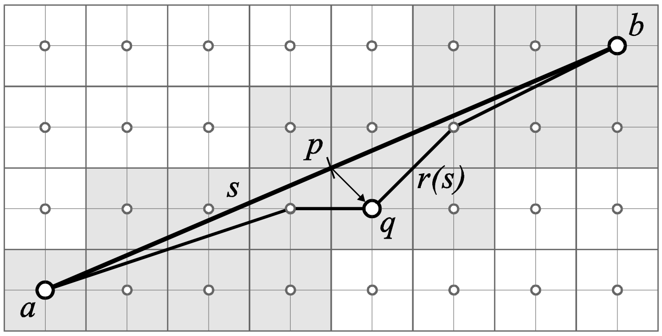


Fig. 3.4 Redrawing of a hooked line

The envelope *E(r(s))* of a redrawing *r(s)* is either the envelope *E(s)* of its base line *s* or an actual subset of *E(s)*:

$$E(r(s)) \subseteq E(s)$$

This relation becomes important in the next section. It is the reason why applying the method of Greene and Yao to simplicial complexes is not successful.

4. PROBLEMS USING ENVELOPES

The reasons why the method of Greene and Yao proved unsatisfactory for the simplicial data model are discussed in this section.

4.1 Different Envelopes

Consider a straight line *s* to be a redrawing $r^{(0)}$ of order zero, the result of a once intersected redrawing of order zero to be a redrawing $r^{(1)}$ of order one, etc. The union of the envelopes of all redrawings derived from a given redrawing by subsequent intersections is a subset of the redrawing's envelope (indices *i,j,k* denote different instances of redrawings):

$$E(r_i^{(m)}) \cup E(r_j^{(m)}) \subseteq E(r_k^{(n)}) \quad \text{with} \quad i,j,k,m \in \mathbf{N}, \quad n \in \mathbf{N}_0 \ \ and \ \ n < m$$

Instead of ordering redrawings we can also order the corresponding envelopes. Thus, $E^n()$ denotes an envelope of order *n*:

$$E^{(m)}(r_i) \cup E^{(m)}(r_j) \subseteq E^{(n)}(r_k)$$

Figure 4.1 illustrates an envelope of order zero and an envelope of order one together with their corresponding redrawings. (Another envelope of order one corresponding to r_1 is omitted for clarity). The problem is that point *P* is an element of the original envelope $E^0(e)$ but not of the derived envelope $E^1(r_2)$. This is a consequence of the fact that the union of corresponding envelopes of order $n+1$ is only a subset of a given envelope of order *n* and not its equivalent. Thus, the answer to whether a point falls on a line segment or not depends on previous intersections.

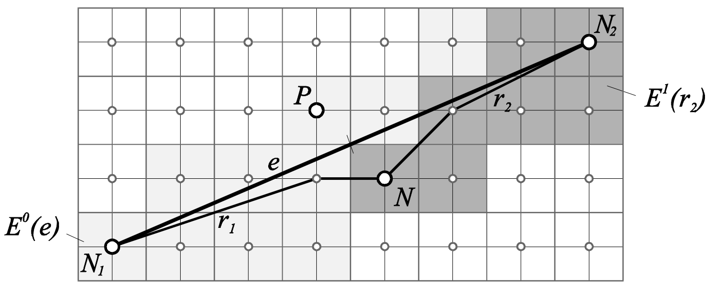


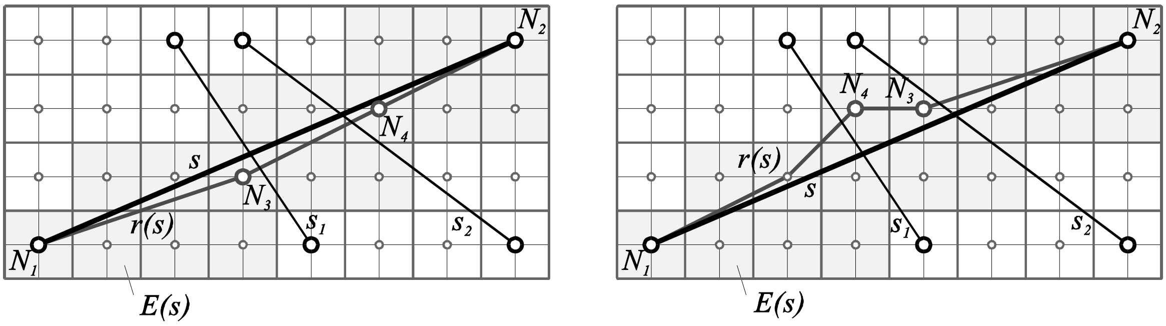
Fig. 4.1 Two answers to whether *P* coincides with *e*

Consistency within a spatial data model based on the method of Greene and Yao would require that envelopes of order zero are used for intersection tests. In this case, the relation given above changes to

$$E^0(r_i) \cup E^0(r_j) \equiv E^0(r_k).$$

4.2 Different Redrawings

Another difficulty occurs with the course of a redrawing within its envelope. Figure 4.2a illustrates a segment *s* and its redrawing *r(s)* after two successive intersections with line segments s_1 and s_2. The positions of the intersection nodes N_3 and N_4 differ from those in figure 4.2b, even though only the sequence of intersecting s_1 and s_2 with *s* has changed.



a) Intersecting *s* with s_1 and s_2 **b) Intersecting *s* with s_2 and s_1**

Fig. 4.2 Different redrawings, depending on the sequence of intersections

A test whether a point falls on a node depends on the sequence of intersections. Generally spoken, topological inconsistencies still remain within envelopes, thus:

Envelopes are not sufficient for the consistent modeling and maintenance of Discrete Simplicial Complexes.

5. Revised Approach: Zero Order Intersection

This section introduces the Zero Order Intersection method. Its key features are the redefinition of envelopes and the use of non-subdivided line segments for calculating intersection points. Collinearity information for line segments is preserved and can be stored explicitly.

5.1 Unique Path

Greene and Yao's envelope of a line segment consists of those grid points situated immediately above or below the line segment. Thus, an intersection point can be rounded to two different grid points. The choice depends on previous subdivisions. To avoid this ambiguity, we replace envelopes by unique paths.

If any point of a line segment s in $\boldsymbol{R}^n$ *can be rounded to a unique grid point in* $\boldsymbol{Z}^n{}_m$ *, the union of all related points in* $\boldsymbol{Z}^n$ *is called unique path U(s) in* $\boldsymbol{Z}^n$*; n denotes the dimension, m the resolution of the discrete space* $\boldsymbol{Z}^n{}_m$*.*

This definition only includes the characteristics of a unique path, not how to produce it. Snap-rounding all potential intersection points of a given line segment (Guibas and Marimont, 1995) is one possible approach.

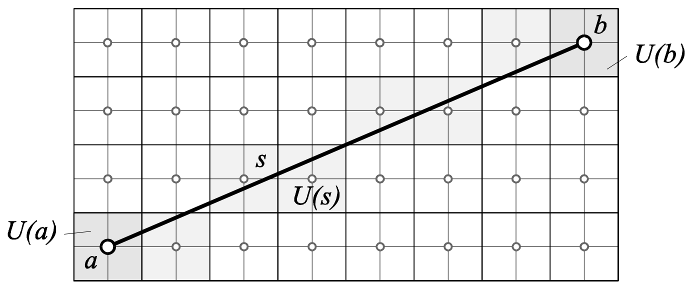


Fig. 5.1 Unique paths of two points and a line segment

5.2 Zero Order Intersection

The intersection point P of two line segments in $\boldsymbol{R}^n$ is rounded to the nearest grid point N in $\boldsymbol{Z}^n$. In general, the subdivided segments are not aligned. Continued intersections can cause drifting line segments (see section 1).

The order of a line segment denotes how often it has been subdivided. A zero order line segment is also called root segment, a once subdivided line segment has order one, etc. (Fig. 5.2). Instead of ordering line segments we can also order their corresponding unique paths.

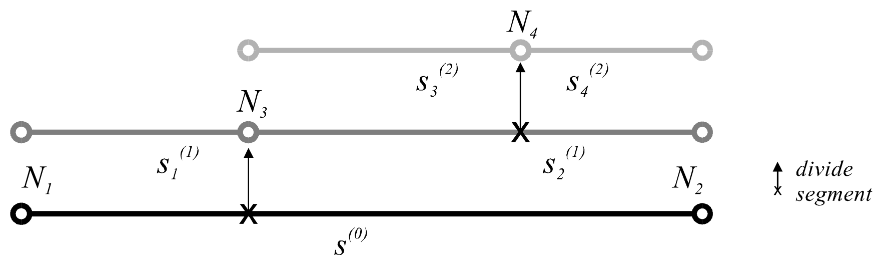


Fig. 5.2 Order of line segments

The zero order intersection uses root segments instead of current segments to calculate intersection points, avoiding drifting line segments. A subdivided line segment refers to its root segment by a reference (Fig. 5.3). Such references are the only additional data to maintain.

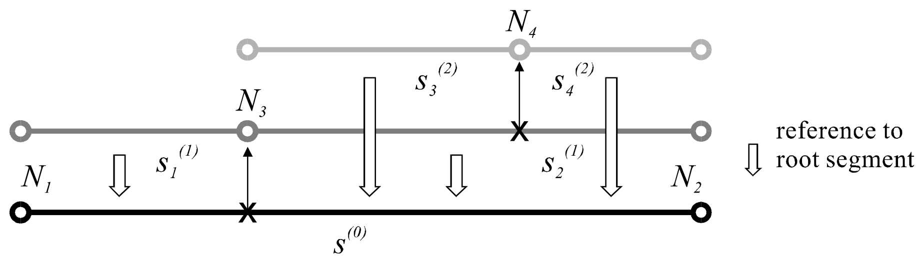


Fig. 5.3 Zero order straight line segment intersection

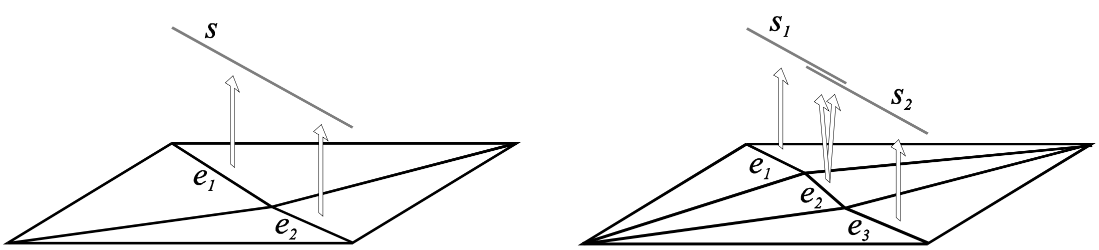
The zero order intersection method calculates unique intersection points which are independent of previously performed subdivisions.

6. APPLYING ZERO ORDER INTERSECTION TO THE SIMPLICIAL DATA MODEL

The Simplicial Data Model allows for a consistent representation of topological relationships. However, topological inconsistencies still arise from repeated intersections of line segments, due to the finite resolution of computational number systems. In order to overcome this limitation, this section discusses how simplicial complex operations and the Zero Order Intersection Method are combined to form consistent point and line insertion operations resulting in the Discrete Simplicial Data Model.

6.1 Linking Topology and Metrics

Operations on the Simplicial Data Model affect both topology and metrics. Topological relations in the plane are represented by the simplexes node, edge and trigon and their boundary relations. Points, lines, areas and other geometric primitives carry metric information. Geometric objects may overlap, simplexes must not.



a) 1:1 references **b) 1:1 and 1:m references**

Fig. 6.1 References from topology to metric

There is a one-to-many relationship between a simplex and its associated geometric objects. A node may be referred to by several points, an edge by several line segments, a trigon by several areas.

The Simplicial Data Model uses coordinates stored with nodes for topological decisions (equal, left, right, inside,…) and coordinates stored with points for metric calculations (distance, area, …). Thus, node coordinates are considered to be relative positions rather than preserving metric information, as is the case in other graph-based spatial data models (e.g. Tiger (Cowen et al., 1990); Realm (Güting and Schneider, 1993)).

Fundamental operations to build up a geometric database based on the Simplicial Data Model are point and line insertion. Both operations assume an initial simplicial complex in two-dimensional space which comprises the minimum bounding rectangle of the area of interest, divided into two trigons.

6.2 Point Insertion

Inserting a point into a Simplicial Database requires to decide on which simplex of the underlying simplicial complex it falls. A point falls on a simplex if its coordinates are elements of the unique path of the simplex. Three cases are possible:

1. The point falls on an existing node: The simplicial complex remains unchanged; the point is associated to the node.
2. The point falls on an existing edge: The point subdivides the edge and is associated to the subdivision node.
3. The point falls on an existing trigon: The point subdivides the trigon and is associated to the subdivision node.

6.3 Line insertion

A line segment can recursively be inserted into a simplicial complex (Frank and Kuhn, 1986); (Egenhofer et al., 1990). The algorithm consists of several critical tasks. Among them are the search for the simplex on which a new point falls, the insertion of the point itself and a function which enforces an edge referring to the straight line segment, following a specific direction. Applying the Zero Order Intersection method to the algorithm requires updates of the references between simplexes and associated geometric objects:

Initialization

Insert the start point and the end point (according to section 6.2).
Let the node of the start point be the current node.

Recursion

If the current node is equal to the end node then stop.
Enforce the next edge referring to the straight line segment.
Let the edge node adjacent to the current node be the new current node.

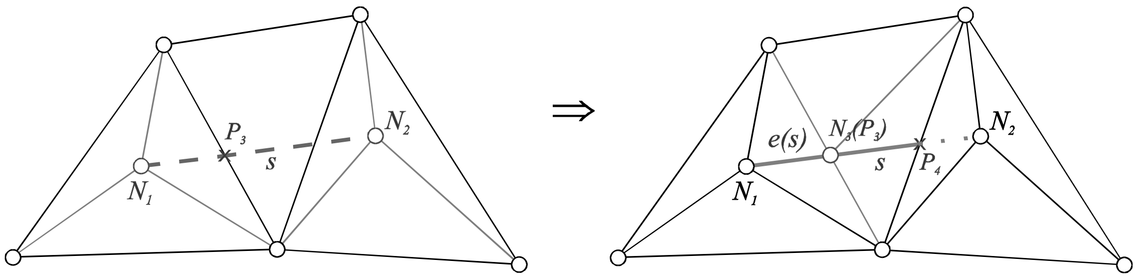
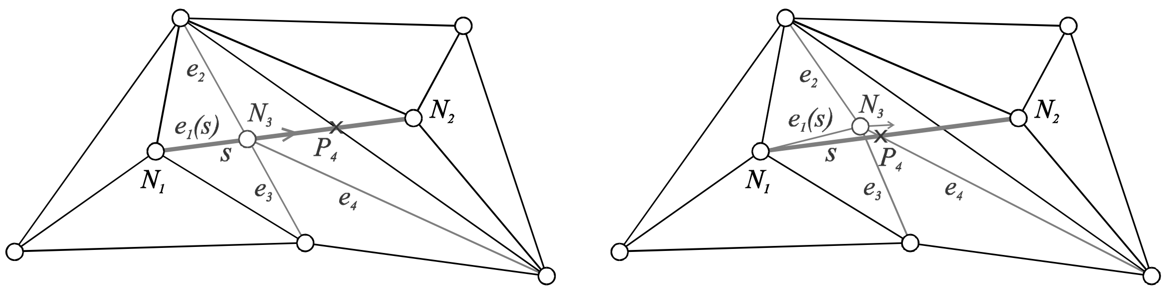


Fig. 6.2 Recursive insertion of a straight line segment

The core of the algorithm is to enforce the next edge referring to the inserted line segment. Let us consider three possible cases in detail:

1. If the segment passes through a node adjacent to the current node, the edge incident to both nodes is the solution.
2. If the segment crosses an edge incident to the current node, this edge has to be subdivided by a new node. The edge incident to both nodes is the solution.
3. If the segment crosses an edge opposite to the current node, this edge has to be subdivided by a new node. The edge incident to both nodes is the solution. An edge is opposite to a node, if its start node and end node are adjacent to each other and also to the node.

Notice that there is one more case in the discrete plane as opposed to the continuous plane. In Figure 6.3a, an edge incident to the current node N_3 is not a possible candidate for the next subdivision, since the segment must exactly pass through N_3 on the continuous plane. Figure 6.3b illustrates the same example on the discrete plane. The segment s crosses the edge e_4 incident to the (rounded) current node N_3. The intersection point P_4 must be rounded to the nearest grid point, too.



a) on the continuum **b) on the grid**

Fig. 6.3 Find next edge to subdivide

6.4 Anomalies on the Grid

Implementing simplicial complexes on a grid can cause situations that are not possible in the continuous plane:

Point inversion

In figure 6.4, rounding the intersection point P to the nearest grid point situates the associated node N on the wrong side relating to the edge e. This violates the simplicial complex consistency constraints.

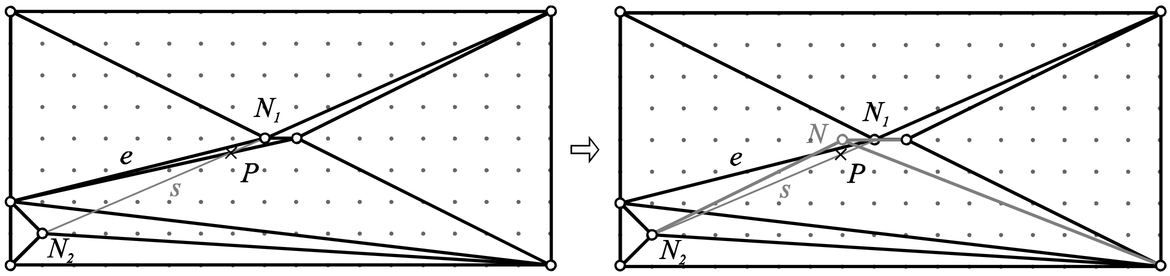


Fig. 6.4 Inversion of a point *P*

Greene, Hobbes and Guibas (Guibas and Marimont, 1995) solve such situations by the so called *snap rounding*. All line segments which cross the pixel environment of a node are subdivided by the node.

Snap rounding can be adapted for simplicial complexes as follows: Instead of testing whether a line segment passes through the pixel environment of a newly inserted node, use test nodes. Test nodes are those in the 4-neighborhood of a given node which are not (yet) inserted into the simplicial complex.

An inserted test node can be removed if it is adjacent to the given node. Otherwise some line segments must cross the pixel environment of the node and the test node becomes part of the simplicial complex itself. Furthermore, the given node and the test node have to be connected to each other by a unit line segment, intersecting all line segments crossing the unit line segment. After applying this process to all test nodes, the pixel environment of a just inserted node is resolved in a consistent manner. The amount of inserted test nodes indicates whether the selected grid resolution fits the scale of inserted geometric objects.

Double Nodes

If a line segment crosses an edge, the subdivision node can coincide with another node close to the edge. Since nodes sharing the same grid position are not allowed within a simplicial complex, the newer node together with its incident simplexes must be kept from being stored in the simplicial database. References between eliminated simplexes and geometric objects have to be transferred to valid simplexes, accordingly.

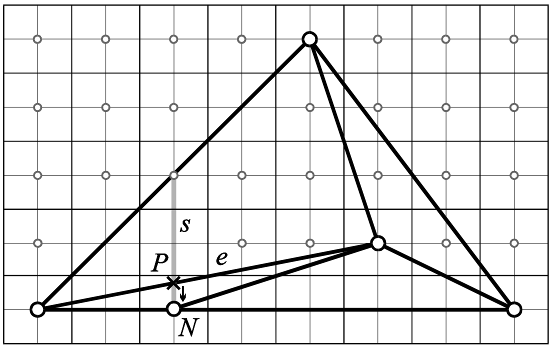


Fig. 6.5 Double nodes

Figure 6.5 illustrates how a double node comes into existence. The intersection point *P* between the new inserted line segment *s* and the existing edge *e* is rounded to the existing node *N*.

Splitting line segments to different edges

When a line segment crosses an edge, the position of the subdivision node is calculated with the line segment and a root segment referred to by the edge. The resulting edges inherit the reference to the root segment, preserving collinearity information. As an edge may point to several root segments, the position of the subdividing node must be calculated with each of them. Thus, line segments sharing a common edge can refer to different edges after such a subdivision process.

6.5 Modeling Collinearity

References between edges and straight line segments allow for a consistent modeling of collinearity information. Three edges e_1, e_2, e_3 are aligned in figure 6.6, since they refer to the same straight line segment *s*.

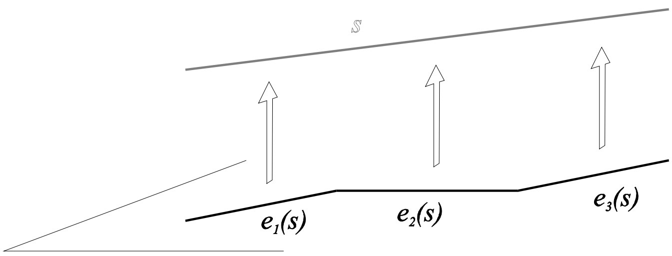


Fig. 6.6 Collinearity of edges

The collinearity of three points follows directly from the Zero Order Intersection definition:

- The points represented by three nodes of a discrete simplicial complex are collinear if all edges of the shortest path between the nodes refer to a common straight line segment.
- If a point is collinear to other points and the edges of the shortest path between the point nodes do not refer to a common straight line segment, they can be *declared* as collinear by inducing a straight line segment to which the edges refer.

7. CONCLUSIONS

We have proposed a robust method for line segment intersections in the discrete plane, called Zero Order Intersection. The application to the Simplicial Data Model has resulted in the Discrete Simplicial Data Model (DSDM) which enables consistent modeling of topological relations within a GIS in discrete computational environment.

Our first approach to avoid line drifting by applying the concepts of envelopes and redrawings (Greene and Yao, 1986) failed for two reasons:

- The union of the envelopes of the parts of an edge is only a subset of the envelope of the original edge. Thus, the original edge cannot be reconstructed from the two edges resulting from a subdivision.
- The course of a redrawing depends on the sequence of intersections. Thus, the answer to whether a point lies on a redrawing or not depends on previous intersections. In dynamic update situations, this is not acceptable.

The key idea of the alternative Zero Order Intersection method is to calculate intersection points only with zero order line segments, i.e. with line segments not yet subdivided. The additional costs compared to the continuous Simplicial Data Model are constant because

- no additional intersections have to be calculated;
- no additional storage space is necessary iff collinearity information is required only for edges associated to geometric objects.

A prototype of the Discrete Simplicial Data Model has been successfully specified and implemented with the Gofer functional programming language (Jones, 1991).

Figure 7.1 illustrates the solution of the drifting line segment example from section 1 calculated with the Simplicial Data Model prototype.

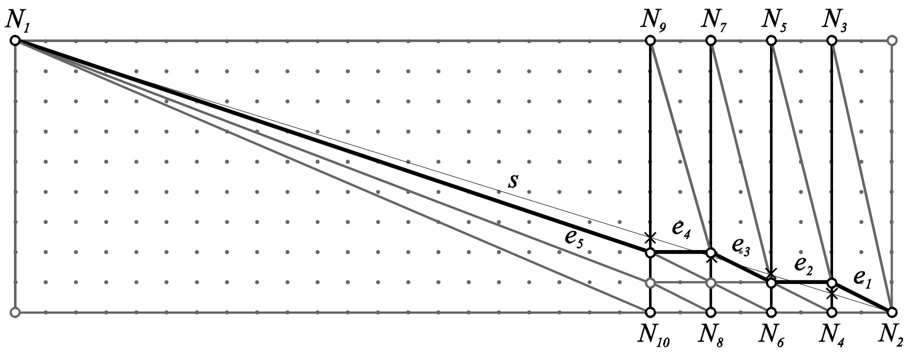


Fig. 7.1 Multiple line intersection with the Discrete Simplicial Data Model

An original straight line segment *s* is subdivided by four successively inserted straight line segments (N_3,N_4), (N_5,N_6), (N_7,N_8) and (N_9,N_{10}). The resulting edges e_1,..,e_5 associated to the line segment *s* do not drift away from *s* anymore.

Based on the line intersection example in this paper and our initial experiences with the prototype implementation, we believe that Zero Order Intersection also works using general curves instead of straight line segments. Essentially, the path of the curve in the grid plane must be unique.

Inserting a point into a simplicial complex is an expensive operation, since the simplex to the given point position has to be found. Hierarchies over partitions (Samet, 1988) and simplexes (Bruegger and Frank, 1989); (Paoluzzi et al., 1993) can improve the costs used for searching. However, the right method has not yet been chosen.

Other extensions of the Discrete Simplicial Data Model include simplexes and associated geometric objects of the third dimension. This seems feasible, since simplicial complexes are defined in any dimension.

References

Bruegger, B. P., and Frank, A. U. 1989. “Hierarchies over Topological Data Structures.” *ASPRS-ACSM Annual Convention*, Baltimore, MD, 137-145.

Cowen, D. J., Shirley, W. L., and White, T. 1990. “An Evaluation of the Use of Digital Line Graphs and TIGER Files for Use in Multipurpose Geographical Information Systems.” *4th International Symposium on Spatial Data Handling*, Zurich, Switzerland, 621-631.

Egenhofer, M. J., Frank, A. U., and Jackson, J. P. 1990. “A Topological Data Model for Spatial Databases.” Symposium on the Design and Implementation of Large Spatial Databases, A. Buchmann, O. Günther, T. R. Smith, and Y.-F. Wang, eds., Springer-Verlag, New York, NY, 271-286.

Frank, A. U., and Kuhn, W. 1986. “Cell Graphs: A Provable Correct Method for the Storage of Geometry.” *Second International Symposium on Spatial Data Handling*, Seattle, WA, 411-436.

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

Giblin, R. J. 1977. *Graphs, Surfaces and Homology*, Chapman and Hall, London.

Greene, D., and Yao, F. 1986. “Finite-Resolution Computational Geometry.” *27th IEEE Symp. on Foundations of Computer Science*, 143-152.

Guibas, L. J., and Marimont, D. H. 1995. “Rounding Arrangements Dynamically.” *11th Ann. Symp. on Computational Geometry*, Vancouver, B.C. Canada, 190-199.

Güting, R. H., and Schneider, M. 1993. “Realms: A Foundation for Spatial Data Types in Database Systems.” *SSD’93*, Singapore, 14-35.

Jones, M. P. 1991. “An Introduction to Gofer.” , Yale University.

Paoluzzi, A., Bernardini, F., Cattani, C., and Ferrucci, V. 1993. “Dimension-Independent Modeling with Simplicial Complexes.” *ACM Transactions on Graphics*, 12(1), 56-102.

Samet, H. 1988. “An Overview of Quadtrees, Octrees, and related Hierarchical Data Structures.” Theoretical Foundations of Computer Graphics and CAD, R. A. Earnshaw, ed., Springer-Verlag, Berlin Heidelberg, 51-68.

TABLE OF CONTENTS