This is Peer Reviewed Paper
FIG Working Week 2008

Transaction Processing on Planar Partition for Cadastral Application

Hrvoje MATIJEVIĆ, Zvonko BILJECKI, Stipica PAVIČIĆ, and Miodrag ROIĆ, Croatia

Key words: cadastre, planar partition, transaction processing, information system

SUMMARY

In this paper we present how rules for geometrical and topological consistency of planar partition can be implemented and used for on-line testing of consistency of updates in a planar partition based transaction processing system.

Upon the reception of a set of polygons as potential new version for a designated subset of the planar partition, the system does all the necessary geometry testing, does the on-line half-edge topology extraction and finally tests the new version of half-edges for internal (within itself) and external topology (the rest of the planar partition). Also, in order to prevent performance degradation in case large area becomes the subject to change, an in-memory spatial index is created on-line, and used for primary filtering. In case topological or geometrical errors are detected a log describing those is created and stored so the initial creator of the transaction or any other person can use it to correct the errors. Otherwise the transaction is either marked as valid or executed. The benefit of such an approach is its capability to receive prepared transactions from any software capable of producing polygonal data in a standard form and to automatically detect errors, description of which is logged for any later usage (primarily correcting).

Transaction Processing on Planar Partition for Cadastral Application

Hrvoje MATIJEVIĆ, Zvonko BILJECKI, Stipica PAVIČIĆ, and Miodrag ROIĆ, Croatia

1. INTRODUCTION

Since the transactions are often, and especially in cadastre, connected with legal conditions, i.e. their changes, clearly it is important to provide consistency under updates. Furthermore, unlike other well-known type of transactional information systems (banking), the cadastral information system deals with the transactions on very complex and highly structured type of data, namely spatial data. It is therefore obvious that the rules pertaining to the consistency preservation present a special challenge for the system, i.e. for implementation.

Traditional European parcel-based cadastres use planar partition to manage their geometry. Planar partition is a set of non-overlapping polygons fully covering one universal polygon.

In the early years of the development of spatial data management system (mid 1980-ties), the topologically structured data were used intensively because they made it possible to reduce the demands for permanent storage space that used to be rather expensive then, but also because they enabled much more efficient processing of some spatial queries that were executed using the algorithms of computational geometry pretty intensively. The main disadvantage of such an approach was relatively poor responsiveness, because geometrical objects had to be created on-line out of topological data. With the prices of permanent storage space going down (1990-ties) and with the available computational capabilities increasing, the focus of spatial data management system users returned to simple (purely geometrical) data structures, because they provided the processing of larger data quantity avoiding the need to have them created out of topological data every time. The maintenance of consistency in carrying out the changes has been shifted to computational geometry algorithms.

The discovery of practical rules for maintaining the consistency (Plümer and Gröger 1996), as well as further development of computational technology, especially spatial databases, have opened the possibility to return to topological data structures (Oosterom et al. 2002), this time as help, i.e. support to geometric structures and not their replacement. The return to topological data structures has manifested itself in the market of commercial spatial data management systems in two implementation approaches (Baars et al. 2004). Traditional updating of topologically structured data is performed by directly updating the tables with node/edges/faces (Oosterom 1997) provided the preconditions are fulfilled (Gröger and Plümer 1997). The drawback of approach with direct updating is the need for highly specialized operators and special software.

Alternatively, if there is an efficient system of conversion available between geometric and topological data in both directions, it is possible to perform the extraction of topology from geometry on-line, which is then used for testing of the correctness of planar partition, i.e. for indirect updating of tables with node/edges/faces (Hoel et al. 2003). This is especially interesting since the majority of standard spatial data management software support the standard form (ISO 2004b) and storage format (ISO 2004a) of Simple Features – SF, which opens the possibility for distributed preparation of changes in standard form and for testing the correctness automatically. Such a system can act partially or entirely automatically by receiving the standard format data sets that present a new state of an area affected by changes, and then by testing all the preconditions defined in advance. Afther that, the transaction can either be executed or only stored and marked as correct for later execution. The requirement for this is an integral and efficient implementation of tests of correctness of transaction.

The rest of the paper is organized as follows. First, there is a short presentation of the rules needed for maintaining the consistency of planar partition, as well as the manner in which they can be concretely implemented for the purpose of testing the changes for geometric and topological correctness. In order to provide efficient testing even of very large changes, it is necessary to use spatial index as primary filter of testing candidates. The theoretical premises and the implementation of an in-memory spatial index are given hereunder together with the presentation of concrete results of the testing done. At the end of the paper there are conclusions and possible areas of further research offered.

2. PLANAR PARTITION

Researchers have been working on testing and searching the relation between advantages and disadvantages of topological data structures ever since the mid 1970-ties. To start with, the reader is referred to one of standard books on GIS or papers e.g. (Theobald 2001).

Although there are more formal definitions, as the one in (Molenaar 1998) where the term geometric partition has been used, a simple and easily understandable definition of planar partition is given in (CGAL 2003): A partition of a polygon P is a set of polygons such that the interiors of the polygons do not intersect and the union of the polygons is equal to the interior of the original polygon P.

In order to retain the unambiguousness we now bring further in the text the most important definitions for the expressions used. Polygon is a two-dimensional geometric element that is defined as the surface closed by one external and by an arbitrary number of internal boundaries. Each polygon’s boundary is made of continuous and closed sequence of lines where every line element is the connection of polygon’s two successive vertices. Equivalent topological elements are face for polygon, ring for boundary, edge for line, and node for vertex. The realization of topology is a process of adding geometric information to the topological elements, which leads to the creation of geometric elements. Out of the realized geometric element the extraction of topology can be performed using computational geometry algorithms.

The full description of the theoretical background and general guidelines for possible implementation of planar partition are given by Plümer and Gröger (1996). Since the material referred to is rather extensive, it will not be repeated but recommended to the readers. There are two facts to be mentioned as important products of the paper. First, the statement that in the relational database the planar partition can be presented with two relations:

- geometry (Node, X, Y)
- topology (Edge, Node1, Node2, Face).

This is correct as well as the observation that other more efficient data structures can be derived from these simple relations, which will be described later in the chapter on implementation. The other fact refers to seven integrity constraints for the correctness of planar partition. The constraints are the following:

- PG1.
  a) for each node in the topological relation there is a node in geometric relation and
  b) no two nodes in geometry have equal coordinates (uniqueness of position),
- PG2. each node has got at least two incident edges,
- PG3. for each edge there are exactly two incident nodes at its ends,
- PG4. no two edges have common points except at the ends,
- PG5. each edge has got exactly two incident faces,
- PG6. each face has got exactly one cycle as a window,
- PG7. no edge point lies on the inner of the face.

The integrity constraints PG1-PG3 and PG5 can be easily implemented at the relational database level, and the constraint PG6 is conditioned by correct realization of geometry from topology depending on the data structure and the chosen algorithm. The constraints PG4 and PG7 are tested by using computational geometry algorithms.

Finite precision in computing the algorithms of computational geometry prevents us from using here the algorithms for testing the constraints PG4 and PG7 in a simple original form. The solution for this problem is given by Milenkovic (1988) through the normalization of geometric data. The normalization of data is defined as slightly changing the structure and parameters of data, getting thus into the configuration for which the algorithms of computational geometry always yield unambiguous results in the conditions of finite precision of computations.

Five rules according to (Milenkovic 1988) have to be met for the polygonal regions to be normal. If the system consists of nodes being arranged as the coordinate pairs of final accuracy that present the points in the plane, and the edges being arranged as the node pairs that present the oriented lines in the plane, and the tolerance value ε is known, then the conditions of correctness are as follows:

1. no two nodes are closer than ε,
2. no node is closer than ε to neither of edges that it is not the end node of,
3. no two edges overlap, except in end nodes,
4. For each node, the angularly sorted list of edges containing that node alternates between incoming edges and outgoing edges,

5. for each point in the plane the topological winding number is 0,1 or not defined.

It should be emphasized that meeting the first two prerequisites ensures correctness of tests based on computational geometry algorithms but it does not provide the correctness of the set of polygons that can be created out of the set of nodes and edges.

3. TRANSACTIONS ON PLANAR PARTITION

Transaction is a set of operations initiated by the application program that are executed on one or more data servers with ACID properties guaranteed by the run-time system of these servers (Weikum and Vossen 2002). In the rest of the text a change on the planar partition is considered a transaction.

3.1 Data Model and Definition of Transaction

Two basic variants of storing the topological data for planar partition are winged-edge (Baumgart 1972) and half-edge, and they come originally from the domain of solid modeling. The basic difference between these two data structures is revealed in the method of storage. With winged-edge all pointers are stored together (single database row) and with half-edge there is a row per half-edge, yielding two rows per edge. Good comparison of these two data structures can be found in (Kettner 1998).

All topological rules refer to the edges and the edges are (and not half-edges) together with nodes the basis of planar graph closing the faces. This is the argument in favor of using some form of full edge data structure. However, since polygons, instead of non-structured set of lines, are the means for managing the planar partition and polygons being an ordered sequence of vertices from which it possible to create half-edges almost directly, makes them convenient for this purpose. Half-edges have been therefore used as topological data structure. The basic geometric segment of the described model are now two relations presenting the variant of half-edge:

- geometry (Node, X, Y)
- topology (Node1, Node2, Face, Ring, Type).

The first three values are pointers to referenced objects and the other two are number and type of the ring in the face (internal/external). The initial state for further considerations makes a set of polygons of planar partition stored in the correct condition in the form of geometry and topology (of half-edges and nodes).

Semantic definition of transactions in planar partition needed in this consideration is the transition from the current state into the new state of data set that determine the polygons of planar partition (geometry and topology) placed within the boundaries of the involved area. The polygon corresponding to the union of all polygons affected by the transaction in the current state will be called involved area. It can also be defined as: each polygon of the set of polygons making the current state of the involved area must be in !DISJOINT (not disjoint) relation with at least one polygon of the same set. Furthermore, apart from the contents of the involved area that will be changed, the boundary of the involved area can also be changed indirectly. The boundary of the involved area is in the topological sense a set of half-edges whose faces are not a part of the involved area, and whose complementary half-edges are part of the involved area (Fig. 1).

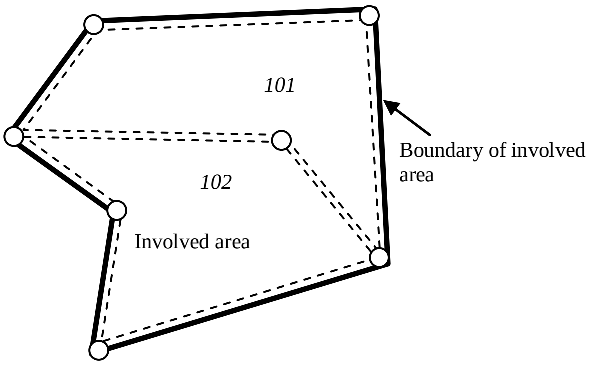

Fig. 1 Involved area and its boundary for the change on parcels 101 and 102

One transaction is a set of delete/insert operations related to topology by which all half-edges are deleted within the involved area and by which new half-edges of the involved area are inserted. Also, the half-edges on the boundary of the involved area that are split because of the changes of half-edges complementary to them must be additionally taken into consideration. Now, a set of prerequisites needs to be found which would ensure that after replacing the polygons (i.e. the derived half-edges and nodes) for the involved area the planar partition would remain in correct state.

3.2 Consistency Rules and Half-edges

Let us presume at the beginning that a set of data presenting future state of the involved area that is a part of topologically and geometrically correct planar partition, and consisting of new polygons and points, has been tested to meet the following prerequisites:

- no new node is located at the distance smaller than chosen tolerance from any existing or any other new node (identically PG1),
- all nodes are of the order >1 (identically PG2),
- all half-edges have got the existing nodes or the nodes that are to be created via this transaction as their ends (PG3),
- all polygons realized from new topology (half-edges) are simple (PG6),
- all new half-edges are extracted from new polygons (M1).

For this set of prerequisites we shall use further on the expression first order rules for the sake of simplicity. Readers are invited to notice that PG5 and PG7 have not been fulfilled, and that PG4 has been indirectly fulfilled through the polygon simplicity condition. To continue, let us introduce the expression of complementary half-edge:

A half-edge is complementary to the other half-edge if its first node is identical to the second node of the other half-edge, and if its second node is identical to the first node of the other half-edge.

Furthermore, it is obvious that, if a half-edge a is complementary to the half-edge b then b is also complementary to a. Let us now have the complementarity rule for half-edges valid:

The involved area (half-edges, nodes and faces that they make) is correct after the change/transaction if the prerequisites PG (1,2,3,6) and M1 are met and if for each new half-edge there is one complementary half-edge.

In order to prove that, we show that it is not possible to have half-edge appear for which there is no complementary half-edge, and to retain the prerequisites from the presumption at the same time. To start with, let us consider a simple change/transaction of two polygons making involved area which gets rearranged by a transaction (Fig. 2).

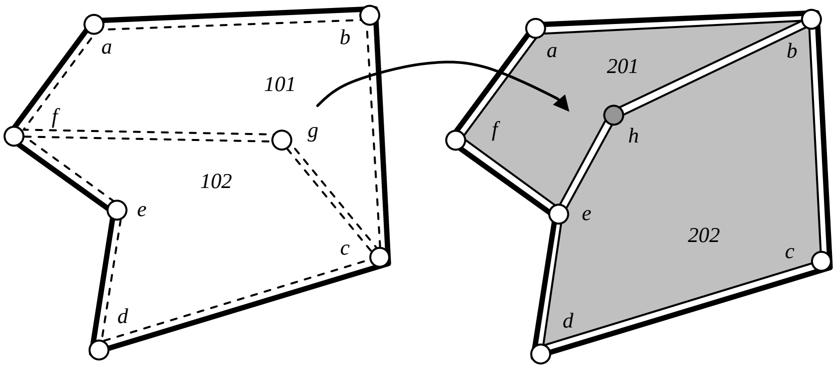

Fig 2. Involved area before and after the correct transaction

Let us consider now the situation on half-edges as a result of a simple irregularity on new node h. If the vertex of the polygon 201 moves from h to h' (Fig. 3) then the polygon 201 moves away from 202 and empty space appears between them which disrupts one of the basic conditions of planar partition, a complete coverage, i.e. formally PG5. The other case, when the vertex of the polygon 201 moves from h to h'', the other basic condition of the planar partition is disrupted, the non-existence of polygon overlapping, i.e. PG7.

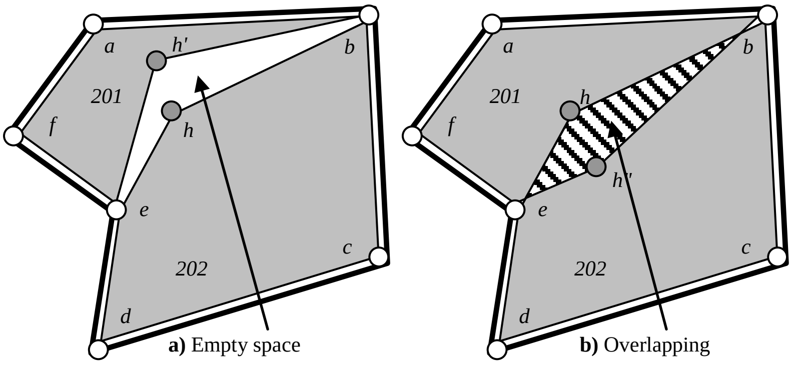

Fig 3. Two types of incorrectness after the transaction

Important here is the fact that in both cases, although the first order rules are met, the transactions have resulted in incorrect state in which for the half-edges eh' and h'b of the face 201, i.e. for the half-edges bh and he of the face 202 there are no complementary half-edges. It is now obvious that if any vertex of any polygon creating a new state of the involved area should move from the node participating in any other polygon, although the first order rules are met, the complementarity rule of half-edges is not met.

Finally, it is to be concluded from the previous argumentation that the new state of the involved area is correct after the transaction if the first order rules are met and if the complementarity rule of half-edges is met.

4. IMPLEMENTATION CONSIDERATIONS

Further in the text there is the description of the implementation given, that is based on the above presented theoretical considerations. In concrete case there has been Oracle10g SDBMS (Spatial Database Management System) used, the entire business logic has been coded in PL/SQL and is executed on the database server. In order to avoid frequent usage of many orders of magnitude slower permanent storage subsystem (HDD) during the execution of transaction testing, all the data to be tested if first loaded into in-memory copies and all algorithms are executed on that data. For the purpose of in-memory data storage the PL/SQL tables (index-by tables) have been used that provide direct access to data via index value. For the purpose of presenting the orders of magnitude, the tests have been executed on a database engine installed on a PC with one Intel dual core processor working on 2GHz clock speed and with 1GB RAM and a SATA HDD. The Digital Cadastral Map maintenance transaction processing system called *Vektoria@DKP* using the described engine for testing the transactions is currently being implemented on sites in Croatia and Bosnia and Herzegovina.

4.1 Algorithms Needed for Implementation

The implementation receives a set of polygons and possibly a set of new points (geometry) as input data and the following needs to be executed. Uniqueness of points position test, the extraction of topology (half-edges) out of geometry together with the normalization of data, checking of the simplicity of polygons and finally checking of the correctness of the set of derived topological data (half-edges) within itself, and on the edge of the involved area.

Two algorithms of computational geometry needed for this are calculation of the Euclidian distance between two points in the plane and determination of the distance of a point from a line (Morrison 1991, Paeth 1990).

Extraction of topology is an operation yielding a set of half-edges out of a set of polygons and nodes. The procedure is rather trivial and consists of spatially querying the set of points (nodes) for each vertex of the polygon in order to find the one that is spatially identical to the vertex (within the tolerance). Node pairs for successive polygon vertices together with the data for the current ring of the polygon make records for consecutive half-edges of the face of this polygon. The realization of polygon from the face is again a trivial procedure, and it consists of directly acquiring the node coordinates and storing them in a predefined format (SF) according to the sequence from the face record using the identification of nodes from half-edges. There is no spatial querying here.

4.2 Optimizing of Spatial Tests (Spatial Indexing)

In order to execute primary filtering of possible pairs during testing of a spatial relationship, a spatial index is used with spatial databases. However, the functionality of spatial index is lost with in-memory calculation. With smaller input data set it is of no importance, but if the input data set is large, the usage of spatial index considerably improves the performance.

Therefore, the approach with spatial indexing of in-memory data is used here. In order to perform tests for the distance between points, a very simple procedure is described by (Zalik 1999). In brief, all points are stored into the memory (into the collection with the possibility of direct access by using an identifier), this is so called geometry data structure – GDS. For the entire area (here involved area) an imaginary division into fields of identical dimensions is created that cover the whole area, which was called by the author a uniform planar subdivision – UPS. When being entered into GDS it is defined for each point in which field of UPS it is placed and this information is stored. In order to obtain a good storage demand and obtained efficiency ratio, index structure can be organized in such a way that there is a record together with the indicator of the first point of GDS in this field. In GDS each record contains a pointer for the next record (the pointer for the first record is in UPS) for each field in which there is at least one point (Fig. 4).

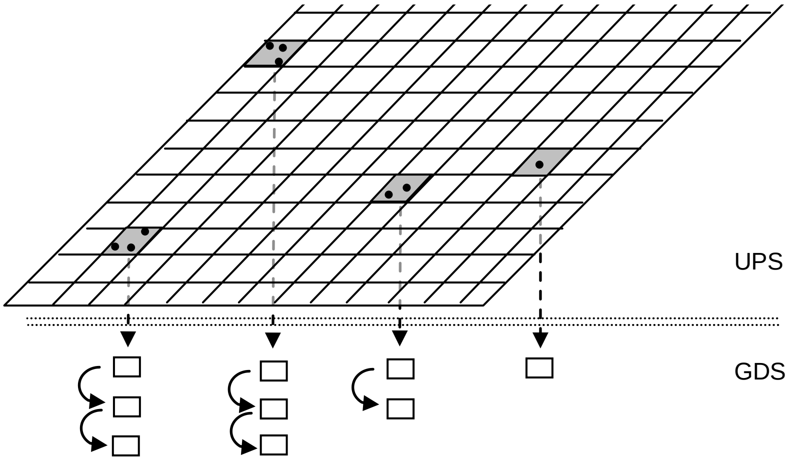

Fig. 4. Relationship between UPS and GDS (modified from Zalik 1999)

When filtering the candidates for testing, after the position has been calculated, i.e. the filed in which the test point is located, it is sufficient to take the record of the first GDS element from UPS and perform testing for it and for all the others (obtained by means of the pointer from the previous one). Additionally, the tolerance for distance tests needs to be regarded for the points near or at the edge of the field. The easiest way to deal with this is to copy the reference for the point near or at the edge of the field to the neighboring field which ensures it will be tested with points from this field also.

With indexed linear elements Zalik (1999) used already created UPS by determining for each line in which fields UPS is located (through which it passes). Now that a set of lines for separation testing is to be checked, it is sufficient to check for each UPS field which lines go through it. In order to avoid multiple testing of the same line pairs they are marked as tested. This, of course would require a calculation of algorithm for finding fields through which lines pass, which would unnecessary complicate initialization of the indexing subsystem.

In order to make the approach as simple as possible, i.e. the initialization of index, in our approach only one more level of UPS has been introduced for indexing linear elements. The fields of this level (further UPS1) are obtained by grouping, i.e. joining the fields of already created UPS for the points (further UPS0). The information in which UPS1 field the UPS0 field actually belongs is not physically stored, but determined on-demand. The creation of index for the lines (UPS1) is now reduced to calculating the position of both end points on the line in UPS0 and to checking whether both points are in the same UPS1 field. The lines that are not sorted in either of the UPS1 fields, hence, those whose both end points are not located in the same UPS1 field remain to be tested with each test line (or points, according to the test type) (Fig. 5).

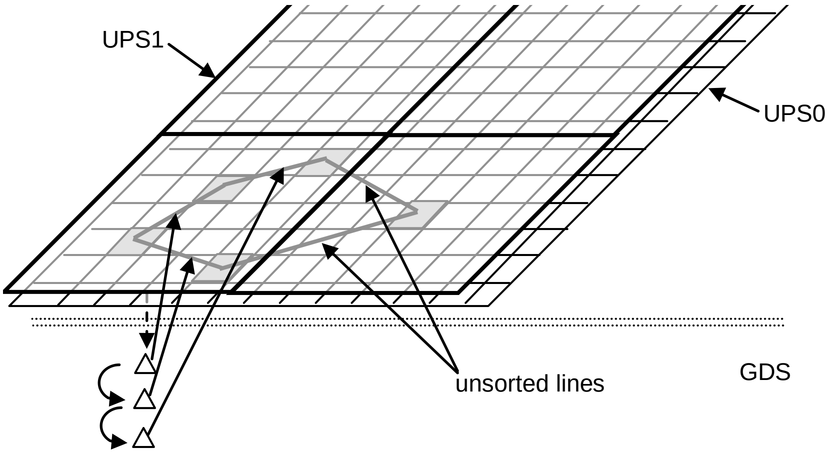

Fig. 5. Relationship of UPS levels

When using that kind of approach to spatial indexing, it is important to choose the size of the field of both UPS levels properly. Due to the nature of cadastral data, very low level of grouping of points can be expected at smaller distances. Not to lose a positive effect, it is also not desirable to create the division with to small field (smaller dimensions of individual fields result in larger total number of fields).

Determination of UPS1 field size is more complex. The field with to small dimensions will cause a large quantity of lines that are not classified and that need to be always tested. The field with to large dimensions will result in classifying large quantities of lines into individual field, which causes the loss of efficiency because the effect of filtering is reduced.

4.3 Performance Testing

In order to perform the index efficiency testing, it is necessary to define the sizes of fields on both levels. Therefore, UPS0 field size of 2 meters (2x2 meters) has been chosen. The reduction of field size could result in even less points being classified into individual field, but it would bring no significant effect, because the number of tests for each new point from the total number of points in the involved area has been reduced anyway to as much as it can be expected in the area of 2x2 meters. For the size of UPS1 field the value of double average length of all edges in the cadastral municipality has been chosen empirically.

For the purpose of testing two transactions have been prepared. One transaction is average size that can be expected in everyday operation of the system, and the other one is actually the creation of an average cadastral municipality (about 4000 parcels). (Fig. 6). Furthermore, the information about 12211 new points and 33816 half-edges (.i.e. as much polygon vertices) is more important than the number polygons (parcels) themselves.

Fig. 6. The test transactions

On both transactions there has been the testing executed without using any indexing level, and as expected, because of a very large data set, the time for the big transaction to be executed was not acceptable (almost 9 hours) (table 1). After that, the testing of both transactions has been made, first by using only UPS0 (indexed points) and then also UPS1 (indexed lines).

Table 1.

Transaction	No indexing	UPS0 used	UPS0+UPS1 used
Big	08:46:50	00:01:50	00:01:21
Small	00:00:03	00:00:02	00:00:02

The results of the test show that with smaller transactions to be most frequently tested there is no significant profit in using UPS1. It has been expected because the quantity of data to be tested is small, hence, much slower permanent storage (HDD) subsystem more significantly influences performance. Nevertheless, the increase of performance is obvious in a transfer from the situation without index to the usage of UPS0, i.e. to the indexed points.

With large transaction the differences are much more significant. Using UPS0 the time of testing drops from unacceptable 9 hours to not more than 2 minutes, i.e. about half a minute less when using UPS1. This is logical, too, because the usage of index has made it possible to avoid unnecessary distance calculation for a large quantity of data pairs.

Let us consider again the results when changing from UPS0 to UPS1 with both transactions. For smaller transaction at smaller involved area the lines will not be shorter, but they will fall into a few fields of UPS1 only, or the most of them will remain unclassified, which causes a small to none performance gain. Large involved area of large transaction profits on the other hand from UPS1 because a lot of point/line pairs need not be tested. Furthermore, natively implemented mechanism of Oracle SDBMS (validate_geometry_with_context) that is extremely efficient has been used in the implementation to test the simplicity of polygon. If this was not available, the benefit of using UPS1 would be higher even with smaller transactions because its usage would provide the elimination of some of computationally very expensive line-crossing tests.

5. CONCLUSIONS AND FURTHER RESEARCH

This paper has illustrated that it is possible to implement the system for fully automatic testing and executing the transactions in planar partition with explicitly stored topology in the form of half-edges. The tests are based on provable theoretic assumptions, and the usage of innovative approach to in-memory spatial indexing makes it possible to test for correctness the changes that include very large data sets in acceptable time.

Within the scope of this research the effect of index field size change on performance has not been systematically inspected, but there was an empirical conclusion derived from concrete testing that a good ratio of performance between large and small transactions can be achieved by selecting the size of the field for indexing linear data that is twice as large as the average length of all edges in the planar partition. Further research can be aimed to finding exact values for optimal size of indexing fields for both point and linear data.

REFERENCES

Baars, M., Stoter, J., Oosterom, P. van, Verbree, E. (2004): Rule-Based or Explicit Storage of Topology Structure: a Comparison Case Study, 7th AGILE conference on Geographic Information Science April 29th - May 1st, Heraklion, Greece.

Baumgart, B. (1972): Winged-edge polyhedron representation, Technical report CS-320, Stanford University, Stanford.

CGAL (2003): CGAL user manual, CGAL Consortium, 2003.

Gröger, G., Plümer, L. (1997): Provably Correct and Complete Transaction Rules for GIS, In: Laurini, R. / Bergougnoux, P.: Proceedings of the 5th International Workshop on Advances in Geographic Information Systems (ACM-GIS'97). Las Vegas, Nevada 1997, pp. 40 - 43.

Hoel, E., Menon, S., Morehouse, S. (2003): Building a Robust Relational Implementation of Topology, Advances in Spatial and Temporal Databases, Proceedings of the 8th International Symposium on Spatial and Temporal Databases. SSTD 2003, Santorini Island, Greece July 2003, Springer-Verlag Lecture Notes in Computer Sciences 2750.

ISO (2004a): ISO 19125-2:2004 Geographic information -- Simple feature access -- Part 1: Common architecture, ISO

ISO (2004b): ISO 19125-2:2004 Geographic information -- Simple feature access -- Part 2: SQL option, ISO.

Kettner , L. (1998): Designing a Data Structure for Polyhedral Surfaces, Proc. of the 14th ACM Symp. on Computational Geometry, str. 146-154, Minneapolis.

Milenkovic, V (1988): Verifiable Implementations of Geometric Algorithms Using Finite Precision Arithmetic, Artificial Intelligence , vol. 37, str. 377-401.

Molenaar, M. (1998): An Introduction to the Theory of Spatial Object modelling for GIS, Taylor & Francis: London.

Morrison, J. C. (1991): Distance from a point to a line, In: Arvo J, editor, Graphics GEMS II, Academic Press, New York.

Oosterom, P. van (1997): Maintaining consistent topology including historical data in a large spatial database, Auto-Carto 13. pp 327-336.

Oosterom, P. van, Stoter, J., Quak, W., Zlatanova, S. (2002): The balance between geometry and topology, Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling, (Richardson i Oosterom urednici), Springer-Verlag, Berlin.

Paeth, A.W. (1990): A fast 2D point-on-line test, In: Glassner A.S., editor, Graphics GEMS, Academic Press, New York.

Plümer, L., Gröger, G. (1996): Nested Maps - a Formal, Provably Correct Object Model for Spatial Aggregates In: Shekhar, Shashi / Bergougnoux, Patrick (Hg.): Proc. of the fourth ACM Workshop on Advances in Geographic Information Systems (ACM-GIS'96). Rockville, Maryland 1996, Str. 76-83.

Theobald, D. (2001): Topology revisited: representing spatial relations, International journal of geographical information science, 2001, vol. 15, no. 8, 689-705.

Weikum, G., Vossen, G. (2002): Transactional information systems, Theory, Algorithms, and Practice of Concurrency Control and Recovery, Morgan Kaufmann Publishers, San Francisco.

Zalik, B. (1999): A topology construction from line drawings using a uniform plane subdivision technique, Computer-Aided Design, Volume 31, Number 5, 30 April 1999 , pp. 335-348(14).

BIOGRAPHICAL NOTES

Hrvoje Matijević is currently head of Department for Geoinformatics in Geofoto L.L.C. Prior to this Hrvoje has worked at the Faculty of Geodesy in Zagreb. He received M.Sc. in 2004 with thesis "Cadastral Data Modeling" and in 2006 he defended his Ph.D. thesis "Modeling Changes in Cadastre" at same university. His main research interests were and still are GIS and Spatial DBMS technology in service of spatial data theory and practice.

Zvonko Biljecki is CEO and President of company Geofoto L.L.C. since 1998 when he returned from Switzerland after 9 years of working in company Geofoto Lugano as technical director. After few years of Ph.D. research on Institute of Photogrammetry and Remote sensing I.P.F. at Vienna University of Technology in 2007 he defended his Ph.D. thesis "Concept and implementation of Croatian Topographic Information System". His main research interest are GI, Laser scanning and automated feature extraction. Currently he follows a postdoctoral study at Faculty for Mathematic and Geoinformation.

Stipica Pavicic graduated in 1998 on Geodetic faculty in Zagreb. Same year he starts to work on faculty as assistant in Department for Geomatics where he starts postgraduate scientific study (M.Sc.) and finishes it in 2003. In 2003 he starts to work in Croatian Geodetic Institute as professional collaborator in Department for Geoinformation Systems and Databases. From the year of 2004 he is working at the Department for Geoinformatics in Geofoto L.L.C. As a team leader and the team member he participates in the number of GI projects. He is also Ph.D. student at the Faculty of Geodesy in Zagreb.

Miodrag Roić is a professor at Spatial Information Management/Cadastre at the Faculty of Geodesy in Zagreb and a national delegate to FIG Commission 3. Currently he is Vice Dean of the Faculty . He leads and participates actively in numerous projects at international and national level. In 1996 he received his Ph.D. from the Vienna University of Technology. He was an Editor-in-Chief of "Geodetski list", an internationally recognized Croatian scientific geodetic magazine.

CONTACTS

Dr. sc. Hrvoje Matijević
Geofoto L.L.C.
Buzinski prilaz 28
10010 Zagreb
CROATIA
tel. + 385 1 66 80 111
fax. + 385 1 66 80 182
Email: hrvoje.matijevic@geofoto.hr
Web site: www.geofoto.hr

Dr.Techn.sc. Zvonko Biljecki
Geofoto L.L.C.
Buzinski prilaz 28
10010 Zagreb
CROATIA
tel. + 385 1 66 80 111
fax. + 385 1 66 80 182
Email: zvonko.biljecki@geofoto.hr
Web site: www.geofoto.hr

Mr.Sc. Stipica Pavicic
Geofoto L.L.C.
Buzinski prilaz 28
10010 Zagreb
CROATIA
tel. + 385 1 66 80 111
fax. + 385 1 66 80 112
Email: stipica.pavicic@geofoto.hr
Web site: www.geofoto.hr

Prof. dr. sc. Miodrag Roić
UNIVERSITY OF ZAGREB - FACULTY OF GEODESY
Institute of Applied Geodesy, Chair of Spatial Information Management, Kačićeva 26 Zagreb
HR-10000
CROATIA
Tel: +385 1 4639229
Fax +385 1 4828081
Email: mroic@geof.hr
Web site: http://www.geof.hr/~mroic