academicJournals

Vol. 8(12), pp. 498-507, 30 March, 2013
DOI: 10.5897/IJPS2013.3835
ISSN 1992-1950 © 2013 Academic Journals
http://www.academicjournals.org/IJPS

International Journal of Physical Sciences

Full Length Research Paper

Correction of topological errors in geospatial databases

Siejka M., Ślusarski M. and Zygmunt M.*

Department of Land Surveying, University of Agriculture in Krakow, Poland.

Accepted 29 March, 2013

The problems related to geometrical correctness of maps gathered and stored by the geographic information system (GIS) computer systems are still present nowadays. Firstly, digitization of traditional paper maps is still in progress all over the world (the maps are being converted to a vector form and stored in relational databases). Secondly, the existing digital maps are being constantly updated with use of manual and partially automated methods. The following article presents a system, developed in order to detect and correct topological errors of surface objects. The method of operating on virtual objects in computer aided design (CAD) software environment is suggested. The main advantage of this solution is its applicability in verification of topological errors in spatial databases that contain large numbers of objects.

Key words: Cadastre, geographic information system (GIS), land parcel identification system (LPIS), topology, computer aided design (CAD), software, spatial data.

INTRODUCTION

Spatial relations of cadastral objects are nowadays presented in two basic forms: digital and analog. The numerical form is illustrated by digital data. The analog form includes maps or plans composed of lines, points, symbols, and text (Gaździcki, 1995; Klajnšek and Žalik, 2005). In this situation, two basic models of spatial data can be distinguished: raster model and vector model.

The simple vector model has two basic disadvantages. One of them is the redundancy of data in situations when the given point belongs to two or more objects. In such cases, the point's coordinates must be saved in each of these objects. The other significant disadvantage is that the relations between objects can be detected using the methods of analytical geometry with complex and time-consuming calculations. Application of the vector model to real data bases, were a great number of objects are processed, causes significant increase of work time. In order to eliminate the disadvantages of simple vector model, topological vector model should be used. The spatial data in topological vector model consists of data describing topological relations and geometrical data which are the coordinates of points. In topological data there is a redundancy that allows verification. The geometrical data on the other hand, have no redundancy (Gaździcki, 1995; Longley et al., 2005).

Using topological vector models, has become very popular nowadays in various geographic information system (GIS) software systems. An example of topological vector model is georelational model. In this model the geometrical properties of elements and the topological information related to them are stored in computer files. The values of the attributes on the other hand are stored in the tables of the relational database management system (RDBMS) (Nedas et al., 2007).

In all kinds of models we are dealing with the problem of topological inconsistency of objects. The problem is very common and of practical importance. This is why many software packages offer different solutions. It is obvious that the most desirable solution would be elimination of the errors and creation of consistent topology (Hope and Kealy, 2008). Another possible solution is to develop a GIS system that would use

*Corresponding author. E-mail: m.zygmunt@ur.krakow.pl. Tel: +48601182518.

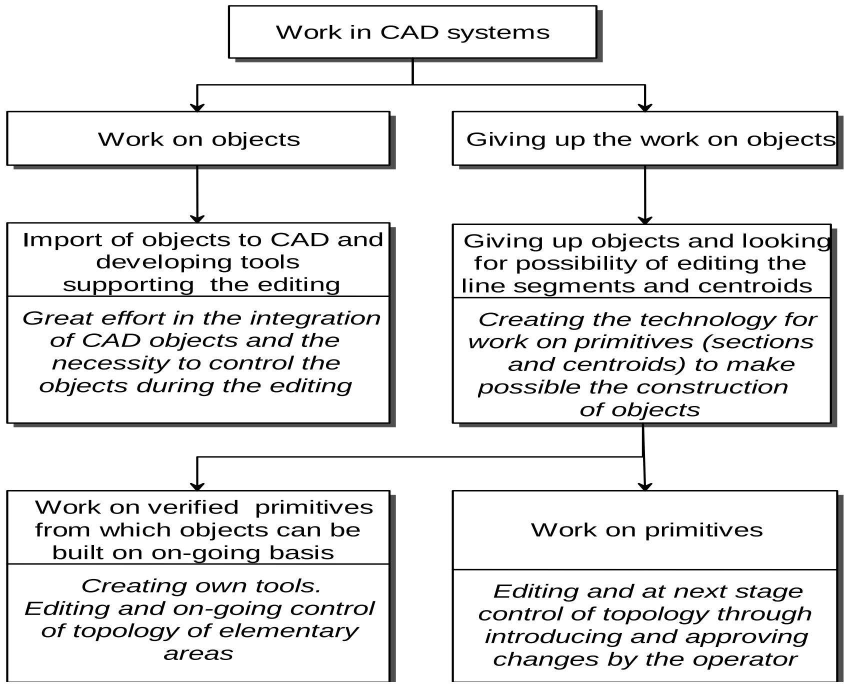


Figure 1. Scheme of alternative technologies of work in CAD.

probabilistic approach to handling uncertain borders (Lagacherie et al., 1996). However, such a solution requires a lot of money and it is time consuming (Klajnšek and Žalik, 2005).

During the processing of large amounts of data for regional or global studies, errors occur (Van Oosterom et al., 2006). It is necessary to eliminate these errors, and the method is dependent mainly on the kind of data processed (Zadravec and Žalik, 2009). In case when demanded positional accuracy of objects is not the most important, we can use highly automated technologies of topological error elimination. An example of such solution was presented in a study describing the methodology of topological error correction in GIS vector data (Maraş et al., 2010). For databases storing official data, such as cadastre, topological error elimination methods cannot decrease quality of the processed data. That is why in this case it is necessary to use a technology that would minimize the number of editing operations and allow to control in detail the correction of topological errors.

In proposed solution of topological errors verification which occurs in official data bases we concentrate on indication of potential errors. Correction of these errors is depends on the operator decision. When the operator was wrong, the error will be detected by the program in the next iteration.

METHODOLOGY

The problem of updating digital databases is related to employing proper technologies. In the process of developing optimal technology it is crucial to firstly answer the following questions: whether or not to work on the objects using the present-day GIS software? Or look for another solution? Professional GIS systems, aside from unquestionable advantages, also have disadvantages. The main drawback of GIS systems when it comes to updating large amounts of data, is the low efficiency of editing separate elementary areas (plots, management field) existing in the form of objects. An alternative for GIS systems can be developing a technology based on computer aided design (CAD) software. There are two possible ways of working in CAD. The first one is working on objects. Importing objects to CAD, possibility of creating new objects and tools that support editing the objects. This method is burdened with considerable efforts related to the integration of CAD objects and the need to verify them while editing.

The second way is giving up on working on objects and looking for a possibility to edit segments and centroids. It is connected with developing a technology to work on primitives (segments, centroids) in order to allow building objects. Such a technology can be created in two ways. One of the ways is to work on verified primitives, from which objects can be built. In this case edition and verification of topology of elementary areas is performed on an ongoing basis. The other way is to work only on primitives. In this solution the first step is edition of objects. The next step is verification of topology by entering and confirming adjustments by the operator. A block diagram of alternative technologies of working in CAD is shown on Figure 1.

DETECTION AND ELIMINATION OF TOPOLOGICAL ERRORS

The proposed methodology of work is based on virtual objects – topo on fly (Figure 2). Virtual objects are properly created sets of primitives (segments as representations of borders of surface objects) and descriptions (as identifiers of closed areas). To allow

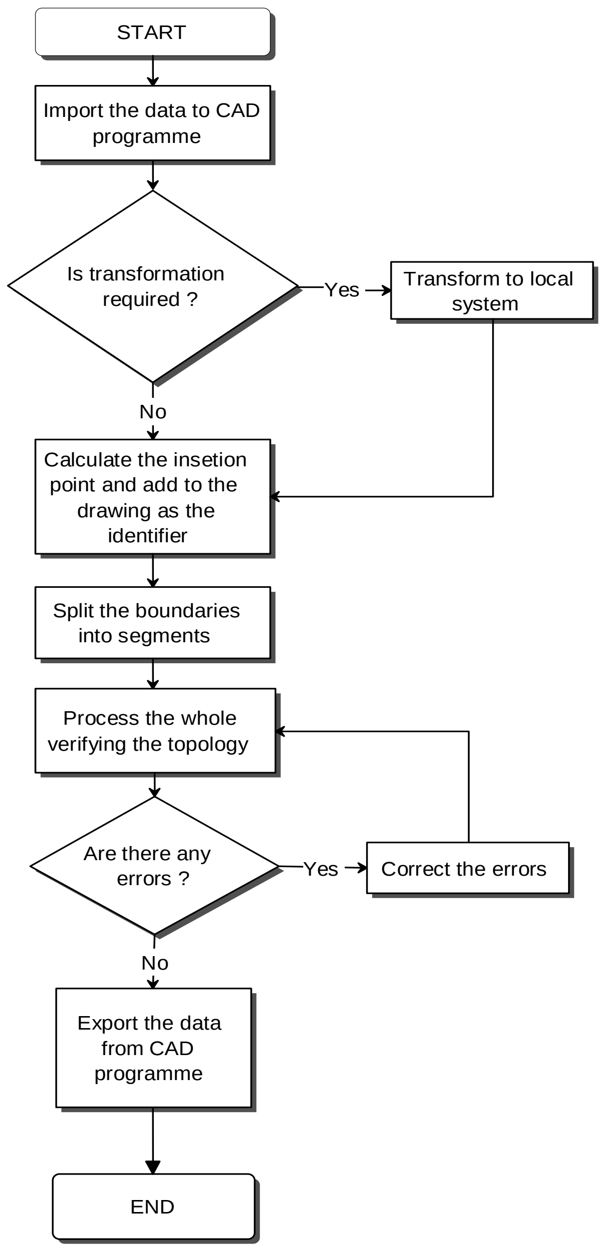


Figure 2. Algorithm of data processing.

proper preparation of vector data for the purpose of working in the described technology (on virtual objects), aside from borders of objects one should also insert an identifier in the form of text. The description must be inside the representation of an object. This implies the necessity to check whether this condition is fulfilled, because calculation of center of gravity of an object does not always give positive result. The problem was solved basing on the algorithm described by O'Rourke (1998). Only the data prepared in such a way can be checked for topological correctness. Verification of topological errors in the presented solution is conducted in the following steps:

(1) Importing the data to CAD software – possibility to read the data that comes from many sources, organizing it and moving to one data format,
(2) Verification and elimination of category I errors – errors in lines of surface objects that make it impossible to construct correct surface objects,
(3) Verification and elimination of category II errors – errors in identifiers of surface objects that make it impossible to unequivocally identify them,
(4) Verification and elimination of category III errors – other errors.
(5) Export of data from CAD software – possibility to save the data in any format of GIS software.

The program was developed in the language of Microsoft Visual Basic for Application (MVBA) for the platform of Bentley System software - PowerMap. MVBA offers algorithms and data structures widening fundamental powers of programming in visual basic. This environment supplies among others, algorithms for data processing, used in range searching, point location, checking of intersections, creating polygons, and also calculation of distances. Launch of the application verifying and eliminating topological errors takes place directly in the Bentley PowerMap work environment, through reading-in and starting the program created by MVBA environment (Figure 2).

Verification and elimination of Category I errors

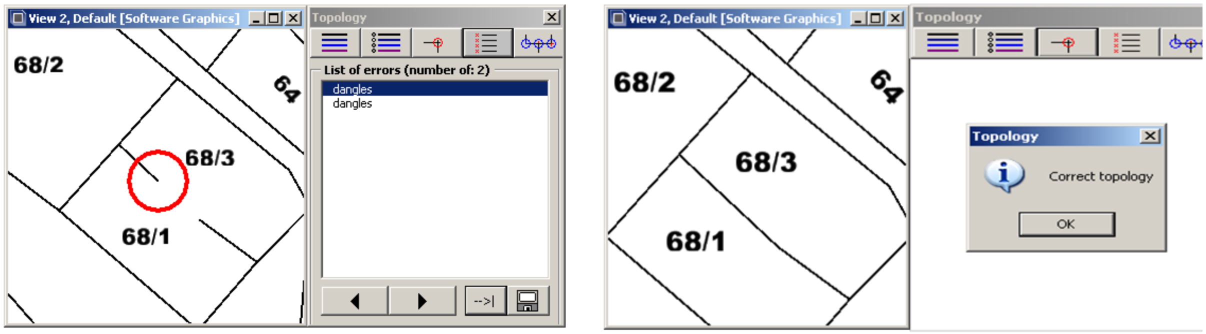
The group of surface object line errors, that make it impossible to build correct objects, includes (Figure 3 to 5):

(a) Dangle errors,
(b) Intersection of borders of surface objects in points which do not belong to the borders,
(c) Near element error – special case.

The most commonly occuring category I error is dangle. It is a segment whose one end is not connected with any node. Such errors make it impossible to automatically generate outlines of surface objects. This error occurs when the line has not been drawn all the way to the node, or after an existing node has been removed. Algorithm checking this kind of error tests if for the beginning point (B) of checked linear object, exists at least one node overlaping investigated object. In the same way the end of the segment (E) is checked. This task is executed by the following algorithm:

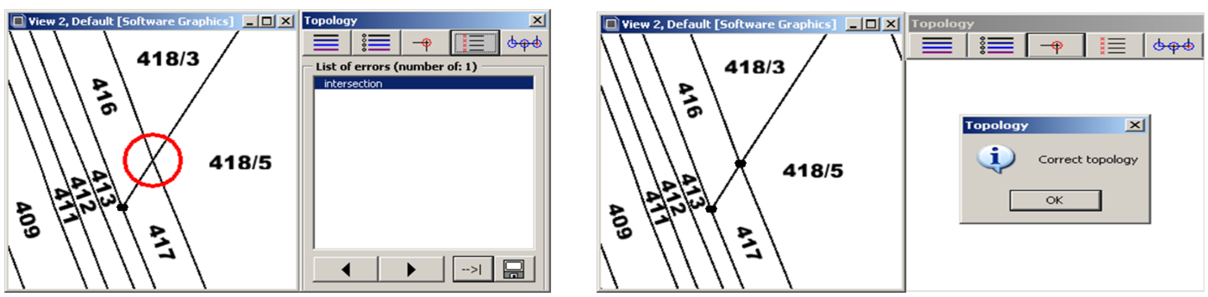
```
For i=1 to n; n – all linear objects
  Is_dangleB=true
  Is_dangleE=true
  For j=1 to m; m – all object assigned to be checked
    If distance|B[i]-B[j]|=0 or distance|B[i]-E[j]|=0 then
       Is_dangleB=False
   End If
   If distance|E[i]-B[j]|=0 or distance|E[i]-E[j]|=0 then
      Is_dangleE=False
    End If
    If Is_dangleB=False and Is_dangleE=False then exit for
  Next j
  If Is_dangleB=true then add B[i] to the list of errors
  If Is_dangleE=true then add E[i] to the list of errors
Next i
```

Elimination of this error boils down to drawing the line to an existing node or adding a node in the place of intersection with another line. Intersection error occurs when one line divides another line into two separate parts in a place where a node does not exist. Algorithm checking this kind of error verifies if two segments intersect. If the answer is yes, it calculates the node, and adds it to the list of errors.



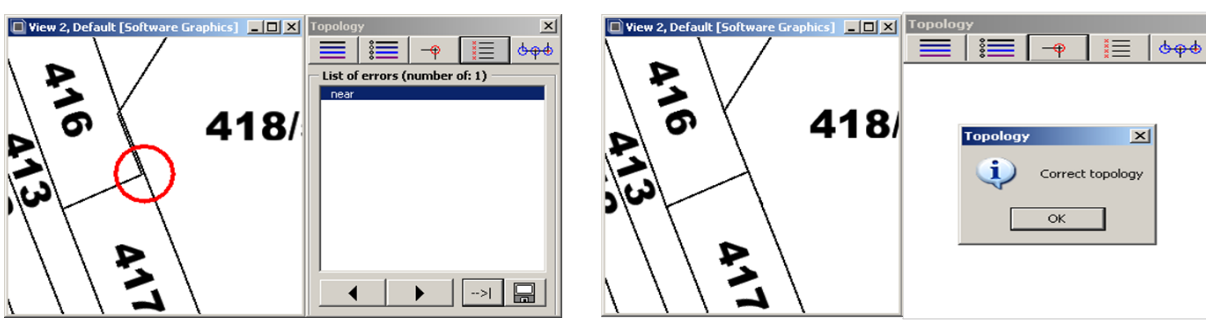
(a) (b)

Figure 3. Indication of dangle error, a) before correction, b) after correction.



(a) (b)

Figure 4. Indication of intersection error, a) before correction, b) after correction



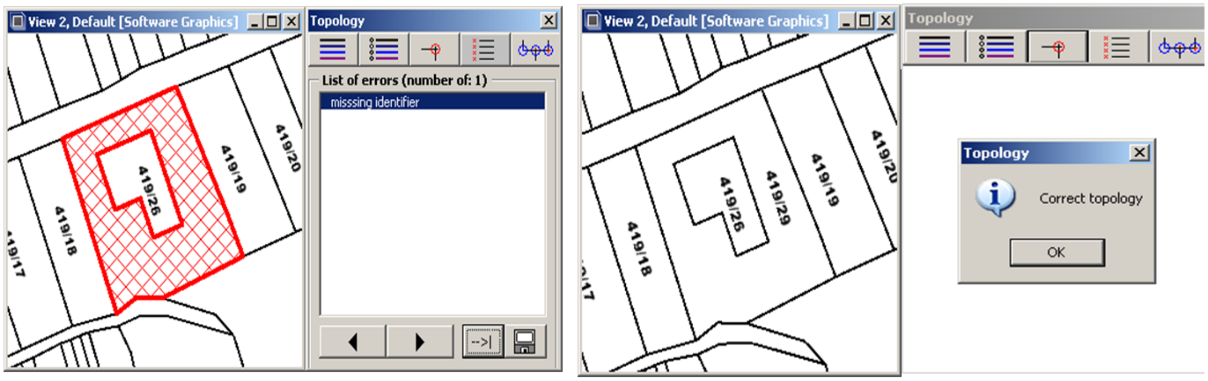
(a) (b)

Figure 5. Indication of near element error – special case, a) before correction, b) after correction.

The checking algorithm makes use of intersections verification function implemented in MVBA environment.

```
For i=1 to n; n – all linear objects
  For j=1 to m; m – all objects assigned to be checked
    If Is_intersection [i]-[j] = True  then
      add an intersection point to the list of errors
    End If
  Next j
Next i
```

This kind of error can be automatically corrected by designating and assuming the point of intersection. However, one should remember that in case of official databases the processed information is a legal document, there should not be any automatic changes without authorization. According to the assumed classification of errors, near element belongs to category I only if the border lines partially overlap (Figure 5). Algorithm checking this kind of error verifies multicollinearity of points. If checked points B[j], E[j] lie on the line segment [i], and are not its nodes – appropriate information is added to the list of errors.



(a) (b)

Figure 6. Missing surface object identifier, (a) before correction, (b) after correction.

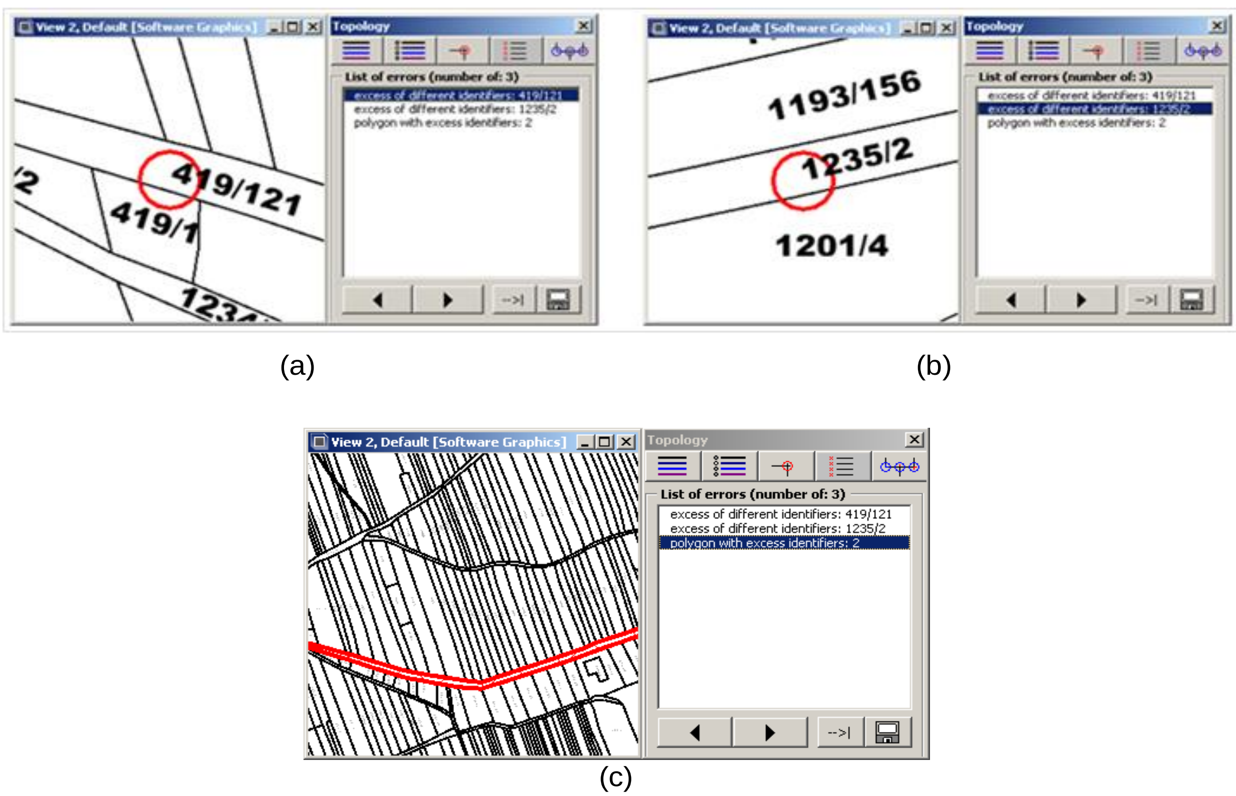


Figure 7. Surface object described by more than one identifier – before correction; a) identifier 419/121, b) identifier 1235/2, c) closed surface object described by two different identifiers.

```
For i=1 to n; n – all linear objects
  For j=1 to m; m – all objects assigned to be checked
    If distance perpendicular between B[j] and Line [i] < ε then
      add B[j] to the list of errors; ε=0.000001
    End If
    If distance perpendicular between E[j] and Line [i] < ε then
      add E[j] to the list of errors; ε=0.000001
    End If
  Next j
Next i
```

This type of error can be automatically corrected.

Correction and elimination of Category II errors

The group of surface object identifier errors, that make it impossible to unequivocally identify objects, includes (Figures 6 to 9):

(a) Missing surface object identifier,

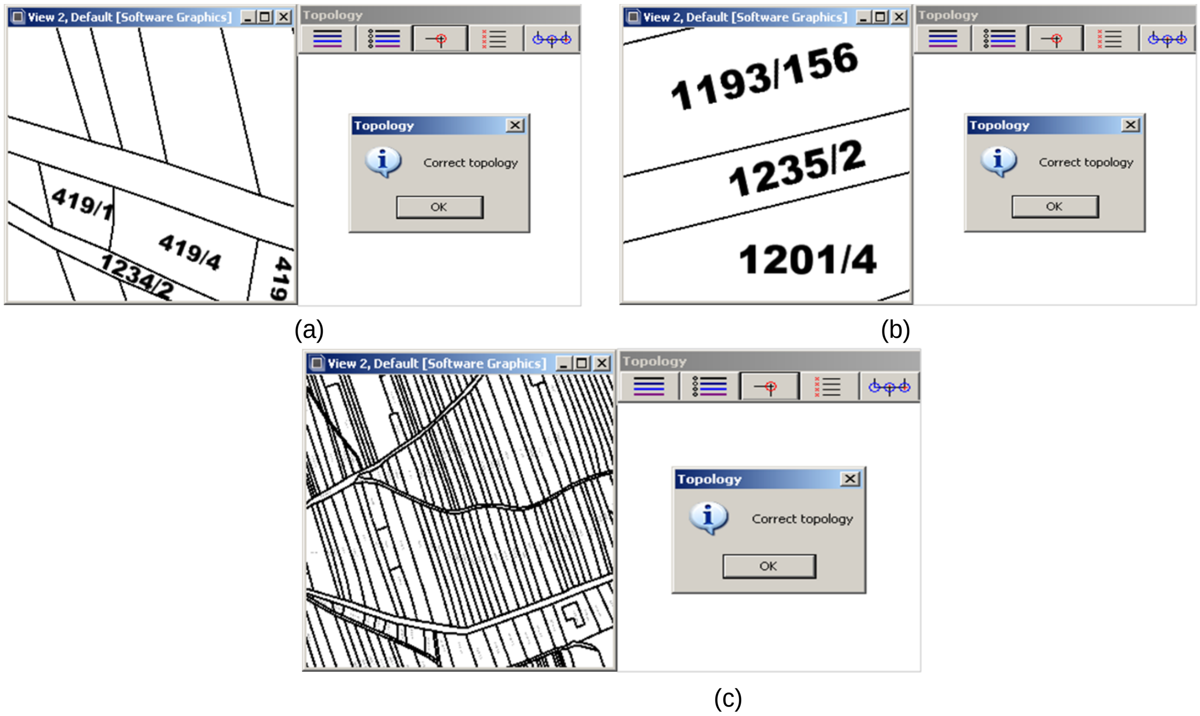


Figure 8. Surface object described by more than one identifier – after correction; a) elimination of identifier 419/121, b) leaving a correct identifier 1235/2, c) investigated polygon, after correction does not prove any error.

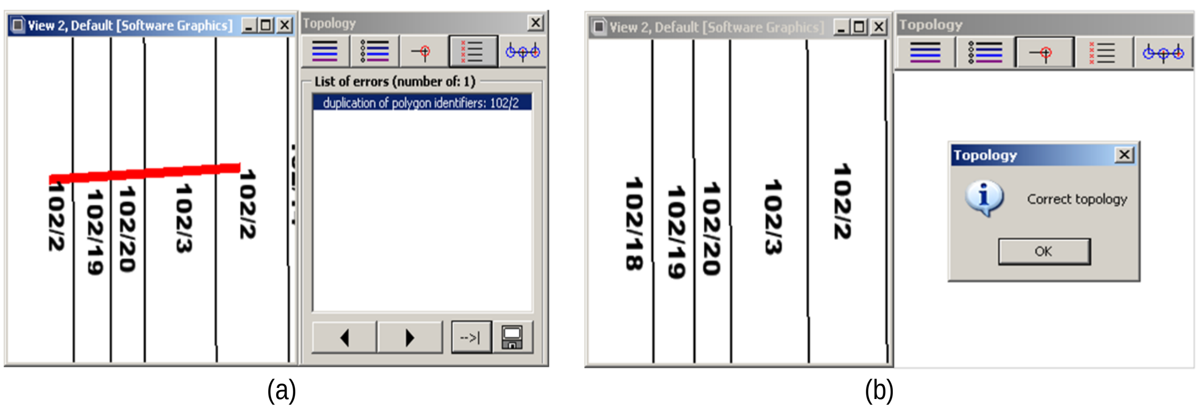


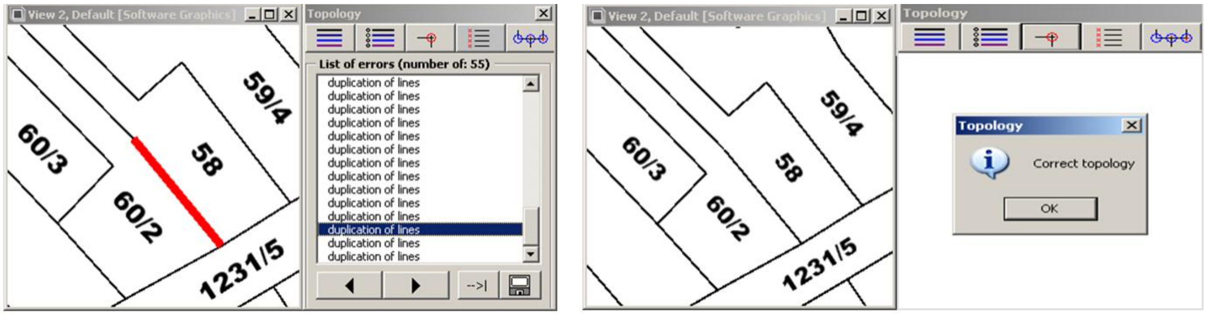
Figure 9. Surface object identifier duplication , a) before correction, b) after correction.

(b) Too many identifiers within one surface object,
(c) Duplication of surface object identifiers.

Missing surface object identifier (for example plot number, name of management field) is one of the most common topological errors. Such an error is especially difficult to detect in case of long objects (roads, rivers, etc.). According to general assumptions, identifier should be located within the outline of an object. The control algorithm of topology indicates the missing identifier error by thickening the edges and hatching. Checking algorithm verifies if identifiers belong to each polygon. If it can assign one identifier – the check is correct. If no identifiers are assigned, appropriate information is added to the list of errors.

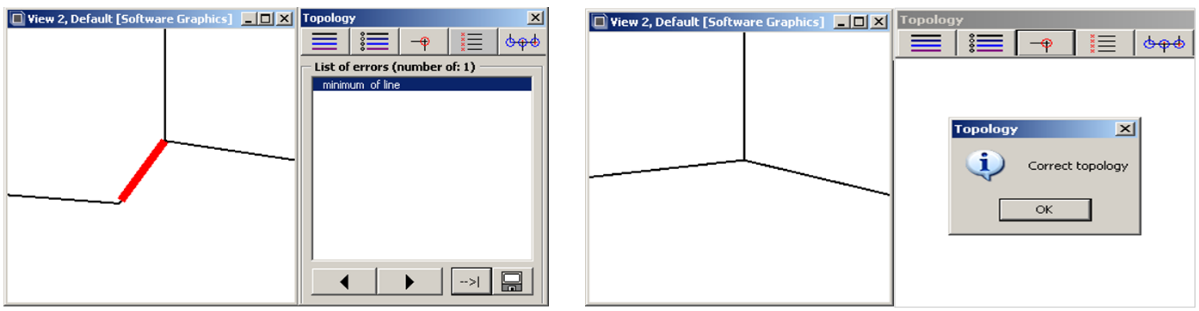
```
For i=1 to n; n – all polygons
  Is_in_polgon=False
  For j=1 to m; m – all identifiers selected for checking
    If Is_identifier [j] inside polygon [i] then Is_in_polygon=True
  Next j
  If Is_in_polygon=False then add polygon [i] to the list of errors
Next i
```

The error is eliminated by inserting the missing identifier. Inside a surface object there can be only one identifier. If the software detects more identifiers, it indicates the error by thickening the edges. Fixing this error boils down to deleting the excess descriptive elements. This kind of error can be eliminated by a



(a) (b)

Figure 10. Indication of "duplication" error a) before correction b) after correction.



(a) (b)

Figure 11. Indication of “minimum” error a) before correction b) after correction.

topology control algorithm only if the identifiers are identical. Checking algorithm verifies if identifiers belong to each polygon on the investigated area. If it can assign one identifier to one polygon – the check is correct. For more than one identifier assigned to one polygon appropriate information is added to the list of errors.

```
For i=1 to n; n – all polygons
  Is_in_polygon=0
  For j=1 to m; m – all identifiers selected for checking
    If Is_identifier [j] inside polygon [i] then
      Is_in_polygon= Is_in_polygon+1
    End If
  Next j
  If Is_in_polygon>1 then add polygon [i] to the list of errors
Next i
```

Identifiers of polygons in the framework of one elaboration must ensure unequivocality of identification. Error of duplication of polygons consists in assigning the same identifier to more than one polygon. Detection of the error is performed by identifiers sorting and comparison of neighboring elements of the table. In the case of existing repetitions appropriate information is added to the list of errors.

```
Sort n
For i=1 to n-1; n – number of all identifiers
   If n[i]=n[i+1]  then add n[i] to the list of errors
Next i
```

In case of official data that contain legal information, the correction must be supervised.

Verification and elimination of Category III errors

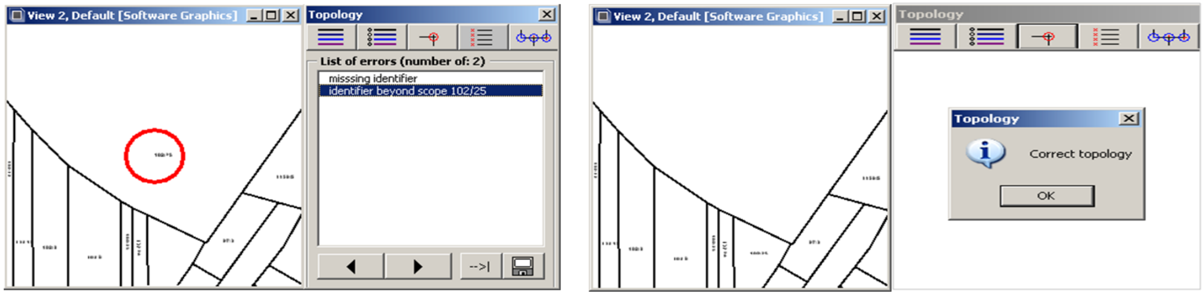
Category III errors are the errors, whose correction improves the database structure (Figure 10 to 15):

(a) “Duplication” error,
(b) “Minimum” error,
(c) Presence of an identifier outside of elaborated area,
(d) Area below the minimum,
(e) Near element error,
(f) Identifier beyond dictionary.

"Duplication" error occurs when there are lines drawn repeatedly basing on the same pair of points. Both topology control algorithms and applied software can ignore such errors. Omitting this error is a necessary condition while working on closed objects, such as SHP. Algorithm looking for errors sorts lines according to coordinates (x,y) of the beginning and end. Neighboring elements of table containing lines are compared.

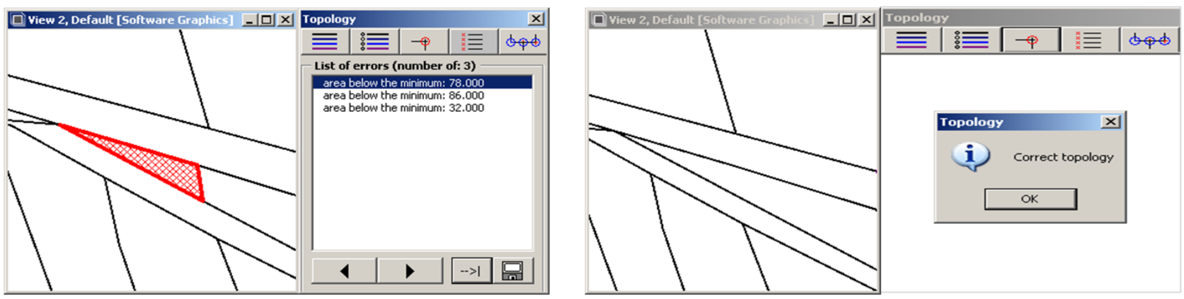
```
Sort lines
For i=1 to n-1
   If Bx[i]=Bx[i+1] and By[i]=By[i+1] and Ex[i]=Ex[i+1] and
     Ey[i]=Ey[i+1]  then add line[i] to the list of errors
   End If
Next i
```

According to the authors, redundancy of boundary lines is the cause of errors in the export of information to other software. This error is eliminated by removing excess elements, which can be done automatically.



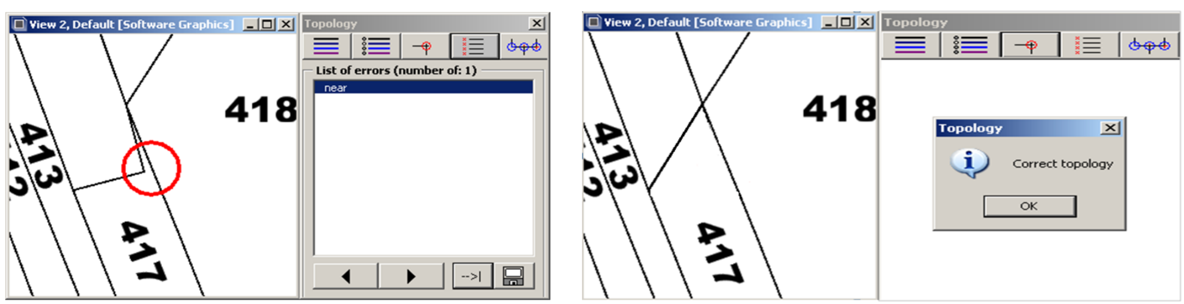
(a) (b)

Figure 12. Identifier outside of elaborated area a) before correction, b) after correction.



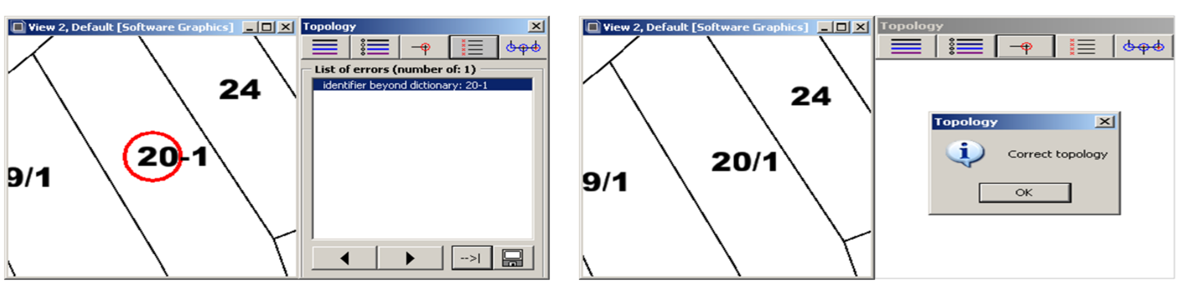
(a) (b)

Figure 13. Area below the minimum, a) before correction, b) after correction.



(a) (b)

Figure 14. Indication of near element error, a) before correction, b) after correction.



(a) (b)

Figure 15. Identifier beyond dictionary, a) before correction, b) after correction

"Minimum" type lines are not visible on a map and it may seem that they do not have influence on the quality of working with a digital map. However, elements such as lines of zero length are "littering" the map, making it difficult to precisely connect elements or verify the coordinates. In case when there are many such elements, they significantly influence the effectiveness of software that processes the information from the digital maps. Such zero length lines can be deleted automatically. Another minimum type error are lines longer than zero, but shorter than the minimum length defined by the user. Deleting such elements is not necessary, but it influences the quality of the map drawing and the speed of information processing. Algorithm looking for errors calculates the length of each line and compares it with given criterion.

```
For i=1 to n; n – number of all lines
   If distance[i]< min and distance[i]<>0 then
     add line[i] to the list of errors (min)
   End If
   If distance[i] = 0 then add line[i] to the list of errors (zero)
Next i
```

This error can be eliminated by deleting the minimum object as long as topological correctness is sustained. Topology control algorithm indicates the presence of surface object identifiers outside of described area. Most commonly the source of such errors are corrections of the course of borders, narrow elements located on the outskirts of the described area. In this situation the correction can be done in two ways. One way is to automatically delete the identifier outside of the described area and then filling the blank area with a proper identifier. The other way is to move the identifier inside the edited surface object.

During the check-up the algorithm verifies assignment of particular identifiers to the polygons. If there are left on the list identifiers which were not assigned to a polygon, they form set of identifiers which are outside the range of investigated polygons.

```
For i=1 to n; n – number of all identifiers
   If identifier[i].use=False then
      add identifier[i] to the list of errors
   End If
Next i
```

The error of minimal area applies to objects whose area is below the defined criterion. In case presented on Figure 13, 100 m^2. The criterion set for the minimal area depends on the scale of the map and it can be a result of an incorrect vectorization. Algorithm processes all ranges comparing their area with the given criterion. When ranges of area smaller than given criterion are detected, appropriate information is added to the list of errors.

```
For i=1 to n; n – number of all polygons
   If area(polygon[i]) < criterion then
      add polygon[i] to the list of errors
   End If
Next i
```

Correction of this error depends on the user's decision, especially in case of official databases that include legal information. Elements lying or running very close to each other or unnaturally acute angles between the lines are a very special kind of errors that result from the process of vectorization of analog maps. For every line drawn on the map the verification procedure searches the buffer area defined by the user. Algorithm checking this error calculates the distance of a point to a line. If the distance of checked points B[j], E[j] to a line [i] is smaller from given criterion and the points are not the nodes of the line [i] appropriate information is added to the list of errors.

```
For i=1 to n; n – all linear objects
  For j=1 to m; m – all linear objects assigned to chech-up
    If distance perpendicular between B[j] and Line [i] < min then
      add B[j] to the list of errors
    End If
    If distance perpendicular between E[j] and Line [i] < min then
      add E[j] to the list of errors
    End If
  Next j
Next i
```

User's decision is necessary to fix this error. Using an identifier other than the ones defined in thematic dictionary is indicated as an error. Checking algorithm verifies in the used identifier on the list of permitted values. In the case of lack of information about error is added.

```
For i=1 to n; n – all identifiers
  Is_identifier_permitted=False
  For j=1 to m; m – all permitted identifiers
    If identifier[i]=identifier_permitted[j] then
       Is_identifier_permitted=True
      Exit For
    End if
  Next j
  If Is_identifier_permitted=False then
     add identifier[i] to the list of errors
  End If
Next i
```

Correction of such an error requires user's decision and an analysis of the source materials.

RESULTS

The presented system of topological correctness verification of surface objects was employed in modernization of cadastre in selected regions of Poland. The next application was verification of topological correctness of plots and management areas as a part of land parcel identification system (LPIS). LPIS is a geoinformation system supporting the management of the common agricultural policy in the European Union. As a part of LPIS over 1.5 million parcels in southern Poland were verified. At present in Poland in the works connected with managing LPIS commonly the solutions based on the GIS software are used. It is in the first place connected with the format of data in which spatial data are written.

Editorial effectivity of polygons in GIS systems and objects based on vector model in CAD systems, shows clear supremacy of CAD systems. Specific character of data bases in LPIS system needs processing of large numbers of data in short period of time. Hence, execution of a task consisting in updating LPIS data base including 1.5 million parcels, was possible to accomplish in required short time, thanks to application of suggested method.

DISCUSSION

Topological correctness of objects is a problem that requires solving in all areas related to creating spatial databases. This applies both to GIS and cadastre (Gröger and Plümer, 2011; Laurini and Milleret-Raffort, 1994). The presented system of verification of topological correction of surface objects can be freely used in databases containing a large number of objects. This includes both GIS databases and official databases such as cadastre or LPIS. The system was developed as an application for the products of Bentley Systems family and not as independent software. Thanks to this, it can use all the tools in the base program.

An additional advantage of the proposed system is the possibility to read the data originating from many sources and saved in different formats, such as extensible markup language/geography markup language (XML/GML), shapefile (SHP), and comma separated values (CSV). The elaborated technology does not perform the check-up of done graphical operations all the time, but only on demand of the operator. This situation may cause creation of errors as a result of operator actions, which will be detected after successive activation of the application. Further works should be aimed on improvements of the program. Particular attention must be drawn on making possible topological correctness check-up immediately during editorial works of polygons.

Conclusion

In official databases such as cadastre or LPIS, where the data is treated as legal document, the possibilities of fixing the errors automatically are very limited. The future elaborations should include further research and tests of the methodology of geometrical correctness automatization.

REFERENCES

Gröger G, Plümer L (2011). Topology of surfaces modeling bridges and tunnels in 3D-GIS. Comput. Environ. Urban Syst. 35:208-216.

Gaździcki J (1995). Cadastral systems. PPKW. Warsaw. P. 123.

Hope S, Kealy A (2008). Using topological relationships to inform a data integration process. Trans. GIS. 12(2):267-283.

Klajnšek G, Žalik B (2005). Merging polygons with uncertain boundaries. Comput. Geosci. 31:353-359.

Lagacherie P, Andrieux P, Bouzigues R (1996). Fuzziness and uncertainty of soil boundaries: from reality to coding in GIS. In: Burrough, P.A., Frank, A.U. (Eds.). Geographic Objects with Indeterminate Boundaries. GISDATA2. Taylor and Francis. London. pp. 275–286.

Laurini R, Milleret-Raffort F (1994). Topological reorganization of inconsistent geographical databases: A step towards their certification. Comput. Graphics 18(6):803-813.

Longley PA, Goodchild MF, Maquire DJ, Rhind DW (2005). Geographic Information Systems and Science. John Wiley & Sons Ltd. Chichester. P. 519.

Maraş SS, Maraş HH, Aktuğ B, Maraş EE, Yildiz F (2010). Topological error correction of GIS vector data. Int. J. Phys. Sci. 5(5):476-483.

Nedas KA, Egenhofer MJ, Wilmsen D (2007). Metric details of topological line-line relations. Int. J. Geogr. Infor. Sci. 21(1):21-48.

O`Rourke J (1998). Computational Geometry in C. Cambridge University Press. P. 392.

Van Oosterom P, Lemmen Ch, Ingvarsson T, van der Molen P, Ploeger H, Quak W, Stoter J, Zevenbergen J (2006). The core cadastral domain model. Comput. Environ. Urban Syst. 30:627-660.

Zadravec M, Žalik B (2009). A geometric and topological system for supporting agricultural subsidies in Slovenia. Comput. Elect. Agric. 69:92-99.