Automated Edgematching of Features from Adjoining Vector Map Databases

Liviu PORUMB and Oana-Lavinia PORUMB

Abstract—**Topographic maps have proven to be very important since the first utilizations, but their utility has increased significantly with fixed-scale drawing use. The evolution of technologies for maps updating and production was influenced by the development of information technology. Nowadays, geospatial information is mostly used in digital format in order to support human activities and projects. Therefore, it is very important that geographical databases are current and accurate. This paper presents a software application, developed to automate the vector edge matching process, a stage included in the overall process of updating map sheet vector databases used by the Defense Geospatial Information Agency of the Romanian Army (former Military Topographic Directorate). The efficiency of the designed application is based on the power of the algorithm used for determining pairs of corresponding nodes and pairs of possibly corresponding nodes. The main reasons for which the authors developed this application were to reduce the overall vector database update time and to offer an efficient tool for checking the values of attributes for contiguous features, stored in different databases, corresponding to adjacent topographic map sheets.**

Index Terms—**adjoining map sheets, algorithm, edge match, maps update, topology, vector map database.**

I. Introduction

Due to the special impact in all areas of economic and social activities, it is very important that topographic maps represent actual and accurate geographic information, thus avoiding to mislead users due to inconsistencies between terrain data and those provided in various forms: analog maps, raster or vector maps and other various derived products. It is therefore necessary to have the most up-to-date geographical information in order to ensure the premises for the subsequent development in good conditions of the actions and activities that depend on the quality of this information.

For a long time and sometimes to this day, updating topographic maps has been an analogous process that required difficult air campaigns and processing operations that were time-consuming.

The emergence of Geographical Information Systems (GIS) has made possible the storage, analysis, visualization and distribution of geographical information in digital format as well as the modernization of technological process for maps updating and production.

The conversion of analogical geospatial data in digital format and their storage on magnetic media, as well as widespread use of GIS has led to a reduction of working time required to update topographical maps. Also, another advantage of modern maps update technology is the increased efficiency by dramatically reducing financial, human and material resources involved, compared to those used in the analogical process.

Even if the global trend is to update maps automatically, the current technology has its limitations, the operator's intervention being practically impossible to eliminate.

Many research directions were approached in order to partially or fully automate the topographic maps updating process, the main focus being on bringing up to date the geospatial vector databases which store topographic features. The most significant achievements have come in the automated extraction linear objects field of study, using high resolution aerial and satellite images. Among linear features, roads are the most suitable for different automatic extraction algorithm implementation [1]. Other methods used to automatically obtain vector data consist in high resolution images classification, followed by application of various image processing techniques and, eventually, features extraction [2]. However, it is difficult at the moment to fully automatically extract new features from aerial or satellite imagery. It is about automatic recognition of shapes, a field in which algorithms and procedures should be better than existing ones. Some linear elements can be semi-automatically extracted but for areal features, the difficulty lies in delimiting the contours of the various types of topographic elements and especially the types of vegetation: forests, vineyards, orchards, crops, grasslands, etc. [3].

Another type of research is concerned with raster map updating; in that case operators manually identify the changes in each cartographic layer (also in the raster map, composed of all the layers) and at the next step develop a procedure for their automatic incorporation in each map raster layer [4].

The most used method for topographic maps updating is utilization of high-resolution images in order to actualize the vector databases in which features are stored. After this process is done, an edgematch operation with adjoining map sheet databases must be performed. Finally, updated vectors are symbolized and cartographically edited, marginalia information is added and the new map is exported, in raster format for further printing operations.

In all existing GIS software platforms on the market there is no suitable tool for simultaneous edgematching of all adjacent vectors. For example, in ArcGIS there is only the possibility of connecting vectors from adjacent map sheets that belong to a single feature class, a fact which has consequences for maintaining topological relationships between elements of different feature classes [5].

L. Porumb was employed at Defense Geospatial Information Agency of Romanian Army. He is associate lecturer at the Military Technical Academy, Bucharest (e-mail: liviu.porumb@mta.ro).

O.-L. Porumb is scientific researcher at the Military Equipment and Technologies Agency, Bucharest, Romania (e-mail: oana_porumb@yahoo.com).

Digital Object Identifier 10.32754/JMT.2021.1.03

In order to reduce the time required for manual edgematching, the authors designed an application which implements a dedicated algorithm for automated connection of contiguous features from adjacent topographic map sheets, stored in vector map databases.

At the same time, the application represents a tool for verifying the values of the attributes of the corresponding elements in the adjacent map sheets.

II. Vector Standards

For military applications, the majority of standards related to geographic information are developed by the United States Department of Defense. Among them, the most important standard for large geographic databases in vector format is Vector Product Format (VPF) [6]. The VPF data model may be considered to be layered into four structural levels, one of these being Vector Map (VMap). Depending on the scale of the source data, VMap products are classified into four categories, as follows:

- VMap level 0, which contains features based on 1:1,000,000 scale Operational Navigation Charts [7];
- VMap level 1, which is based on 1:250,000 scale Joint Operations Graphics [8];
- VMap level 2, which is based on 1:50,000 scale Topographic Line Maps (TLM50K), its architecture being described by the MIL-PRF-89033 standard [9];
- UVMap, which is based on 1:25,000-scale (and larger) City Graphics [10].

VMap2 databases are designed as high-resolution geographic data sources for both military and civilian GIS applications. Classified into ten thematic layers (Boundaries, DataQuality, Elevation, Hydrography, Industry, Physiography, Population, Transportation, Utilities, Vegetation), a VMap2 database contains a total of 154 feature classes. Also, each database includes a reference library that stores generalized data for a broader area of interest.

The main characteristics of VMap2 databases are:

- implementation of object-oriented vector data model;
- geographical elements represented as objects with properties, behavior and relationships;
- object types include: geographic elements (location), networks (objects with integration relations and connectivity with neighbors), topologically related elements and inscriptions;
- geographic data structured in a hierarchy of objects, which are stored in feature classes, object classes and datasets;
- feature classes in a dataset have the same coordinate system;
- data model implements the main properties of objects: polymorphism, encapsulation and inheritance
- allowing the users to define the behavior of the object using software tools such as: domains, validation rules and subtypes;
- facilitating the production of standard topographic maps TLM50K, used by NATO countries.

The Defense Geospatial Information Agency (DGA) uses VMap2 standard to store topographic information extracted from maps produced before 1989, as well as updated geospatial information, extracted from orthorectified images acquired starting with 2010.

On ESRI platforms, VMap2 data can be stored either in personal or file geodatabases. As DGA uses ArcGIS software, all VMap2 vectors are stored in personal geodatabases. A typical database used to process VMap2 data consists of two datasets, a standard one, named *VMap2TLM*, which contains classes of elements in the same coordinate system (UTM zone 34 or zone 35), and a newly added one, named *Cadre_UTM*, which uses the same coordinate system as the first dataset. This second dataset contains only one polygon feature class that contains a single polygon representing the edges of correspondent TLM50K map sheet. To exemplify, if a VMap2 database store vectors for 4090-II map sheet, by convention, the polygon feature class from the second dataset has the same name as the map sheet, which means 4090-II.

From all the point, line and polygon feature classes included in *VMap2TLM* dataset, only the linear and areal elements are of interest. For further processing, in this dataset there must be created the topology based on a set of 519 rules, such as: ***BuiltupA*** *Must Not Overlap with* ***LakeresA***, ***WatcrsA*** *Must Not Overlap with* ***TreesA***, etc.

In order to ensure the following of interoperability requirements within NATO and with its partners, a new standardization agreement in geospatial domain, named NATO Geospatial Information Framework (NGIF), was issued in 2018 [11]. The purpose of this standard is to specify a common geospatial information architecture that is to be used for the generation and exchange of standardized geospatial products and services. Geospatial products implementing new technologies, methods and processes, according with NGIF standard, become less time-consuming and consequently less expensive.

Today, even VMap2 databases are still used in many military command and control systems, they are transformed in NGIF format, to be utilized by DGA for maps updating and army geospatial support. At an equivalent 1:50,000 scale, the vector data is stored in LTDS (Local Topographic Dataset) databases, classified in twenty-three thematic groups which can contain up to 276 topographic features.

III. Vector Map Updating Process

The common updating process of topographic maps in vector format consists of performing geometrical and attributes editing in cartographic databases (CDB) in order to properly reflect the reality on the ground. The generic workflow of updating vector maps is shown in Fig. 1.

Usually, current aerial images are in digital format, but, in very rare cases they are in analogical format. In this situation, analogical images will be transformed into digital format by scanning them at a very good resolution, minimum 1200 dpi. All source data, consisting of recent digital aerial or satellite images, cartographic databases in VMap2 format and digital terrain model, must have the same coordinate system. Then, using a high-resolution digital terrain model, source images are transformed from central projection to orthogonal projection, in other words they are orthorectified.

The detection of changes on the ground is done semi-automatically or visually by operators, overlapping existing vector information with rectified images. At present, it's difficult to fully automate the change detection step,

 Digital Object Identifier 10.32754/JMT.2021.1.03

modified features being automatically extracted if older and recent images are acquired by similar sensors, in same seasons, good atmospheric conditions and with equivalent spatial resolutions. In case of comparing multi-temporal images, collected with different sensors, radiometric adjustments and resampling must be made firstly.

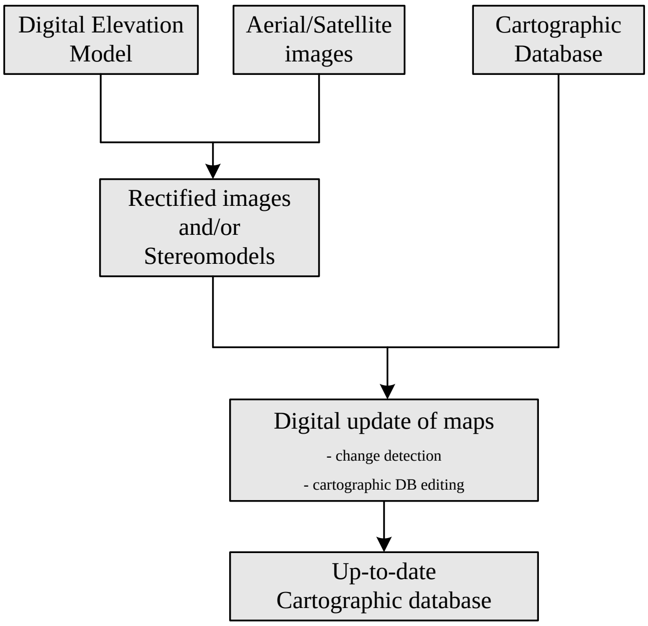

Figure 1. Digital maps update flow

In the next step, the vector database is updated by reshaping modified elements, deleting disappeared elements and adding new vectors representing features that appeared on the ground. The attribute values of changed or new elements are updated as well. If stereograms are available, then a digital terrain model can be obtained in the area of interest, so changes in relief can also be detected. Numerical terrain models can be obtained automatically either using automated correlation of overlapping images or using points with known 3D coordinates.

Eventually, an overall quality control is performed in order to ensure an accurate and consistent updated database for upcoming uses.

Even most details about modified features could be determined at the office, there are some characteristics of the topographic elements, especially descriptive (destination of new buildings, height of trees, height of antennas, types of forests or height of bridges, etc.), that need to be clarified on the basis of field trips and on-site information or by making them available by third institutions: various ministries, town halls, telephone operators, utilities and infrastructure companies as well as other firms.

Thus, the field checking stage is not completely eliminated and a field verification project shall be drawn up specifying the coordinates of a point at the center of the element or area which needs clarification.

After field stage execution and introduction of the data determined at this stage in the database, the quality control is performed, using visual checking and software tools. Next, the connection of contiguous or corresponding linear and polygon features from adjoining map sheets has to be done.

After executing edgematch operation for all edges of a TLM50K map sheet perimeter and quality control, the vector data from VMap2 database are symbolized and cartographic editing is performed. Eventually, standardized marginal information, inner and outer frame of the map is created [12].

In the last step, the digital map previously realized is exported in color raster files, which are further used by plate processors and modern printing systems.

IV. Presentation of Application for Automated Edgematching of Adjacent Map Sheets

In order to automatically connect corresponding vectors from adjoining TLM50K map sheets, the authors developed an application, named ***Edgematch***. To obtain the executable file, which is of dynamic linking library type, Visual Basic.NET object-oriented programming language and ArcObjects library were used.

The application runs on computers that have Microsoft Windows operating system and ArcGIS 10.2.2 software installed.

The basic concept used in this software is topology, which allows users to modify simultaneously the geometry of features from a dataset without affecting topological relationships between them.

Before launching the application, two preliminary operations must be performed on each VMap2 database used to automatically connect containing features:

- clip the vectors from *VMap2TLM* dataset using as clip element a polygon describing the perimeter (inner frame) of the corresponding TLM50K map sheet. This polygon is stored in the feature class associated with each map sheet, in *Cadre_TLM* dataset;
- build *VMap2TLM* dataset topology, based on a set of 519 default rules.

Topology building is a very important operation, thus resulting topological elements of VMap2TLM datasets. New created nodes, edges and polygons will be further used to find corresponding adjacent features and to geometrically connect them.

The launch of the application leads to the appearance of its interface (Fig. 2), and the following operations must be completed:

- selection of TLM50K map sheet to edgematch;
- choosing of edge(s) for matching corresponding features;
- choosing of geometrical type of nodes connection: mediation, move elements from the neighboring map sheet or move elements from the chosen map sheet;
- selection of buffer distance along the chosen edge(s) for edgematch, in order to select nodes for testing;
- choosing of search distance for determining corresponding nodes in adjacent map sheet database;
- check the option to connect corresponding nodes which have different attribute values;
- choosing of directory where VMap2 databases are located, in personal geodatabase format.

The distance of the corresponding node search could be set by users within the range of 1–50 m and the buffer distance, needed to select nodes from adjacent maps sheets, varies within the range of 1–20 m.

The application allows users to choose the modality to

Digital Object Identifier 10.32754/JMT.2021.1.03

geometrically connect corresponding elements from neighboring map sheets databases, from following three methods: nodes from the adjacent map sheet are moved (Fig. 3a); nodes from both map sheets are moved at half distance between them (Fig. 3b), and nodes from selected map sheet are moved (Fig. 3c).

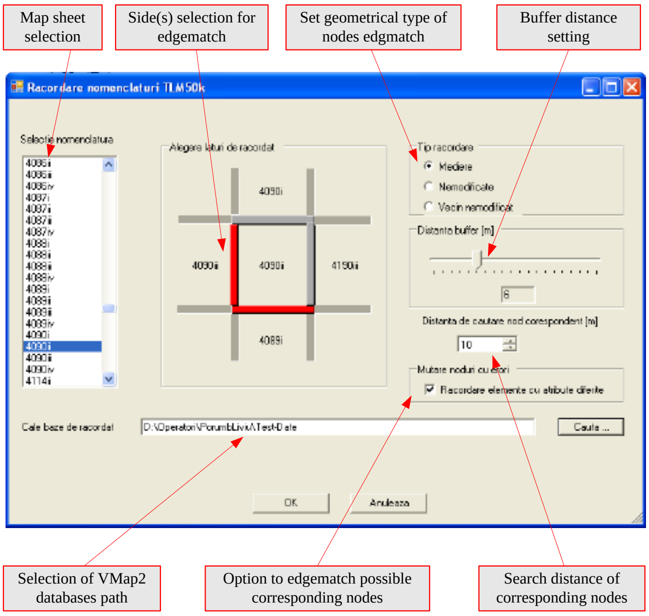

Figure 2. Edgematch.dll application interface

Execution phase begins with finding out the corresponding nodes (nodes in which corresponding features have same parent classes and the same values for descriptive attributes, according to Table I) and possible corresponding nodes (nodes in which corresponding features have same parent classes but values of descriptive attributes are different) for one selected edge. Next, the nodes which have correspondents in the neighboring map sheet are connected geometrically by moving them in the new positions according to selected edgematch method and the nodes without correspondent, as well as their attribute values, are saved in a ESRI shapefile. If more sides are selected, the process of determining and connecting nodes is repeated for every one, in selection order.

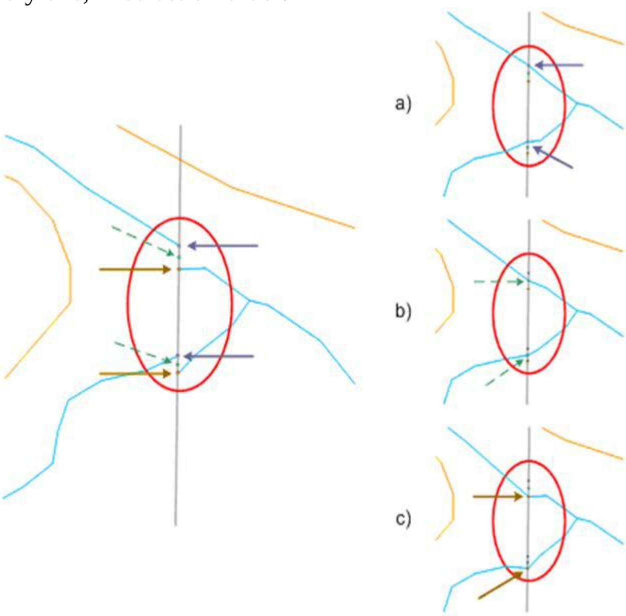

Figure 3. Geometrical modalities to edgematch corresponding nodes

In order to find corresponding elements from adjacent vector map sheet databases, the algorithm, designed by authors, compares the descriptive attributes values of sixteen feature classes (Table I).

TABLE I. FEATURE CLASSES AND ATTRIBUTES WHOSE VALUES ARE AUTOMATICALLY CHECKED

No.	Feature classes	Geometry type	Attributes values that are checked and their description
1	BluffL	Line	**PFH** - predominant height in decimeters; **SGC** - the value of the slope as a percentage.
2	BuiltupA	Polygon	**BAC** - classification of inhabited area according to the density of buildings; **EXS** - state of existence of inhabited area; **NAM** - name of inhabited area; **USS** - inhabited area type.
3	ContourL	Line	**ZV2** - altitude value; **HQC** - level curve type; **MCC** - modeled surface type.
4	PolbndA	Polygon	**ACC** - representation accuracy; **USS** - type of administrative unit; **NAM** - name of administrative unit.
5	PolbndL	Line	**USS** - delimitation type between two administrative units; **NM3** – name for the first administrative unit; **NM4** – name for the second administrative unit; **ACC** - representation accuracy; **BST** - administrative line status.
6	RoadL	Line	**RST** - road surface type; **RTT** - road type; **TUC** - transport category; **WTC** - practicability depending on weather conditions; **LTN** - total number of bands; **LOC** - road location; **EXS** - road existence status; **ACC** - precision representation; **MED** - median road materialization; **NAM** - name; **RTN** - official road name; **USE** - road use; **WD1** - road width (decimeters).
7	RailrdL	Line	**FCO** - element configuration; **RRC** - railway type; **RRA** - food type; **ACC** - precision representation; **EXS** - state of existence of the railway; **LOC** - element location; **GAW** - gauge width; **LTN** - total number of bands; **NAM** - name; **RGC** - classification by track gauge.
8	LakeresA	Polygon	**HYC** - lake type; **NAM** - lake name; **EXS** - element existence state; **SCC** - water characteristic; **ZV2** - altitude value.
9	WatrcrsL	Line	**HYC** - river type; **NAM** - river name; **EXS** - element existence state; **TID** - tidal fluctuations category; **WID** - width value.
10	WatrcrsA	Polygon	**HYC** - river type; **NAM** - river name; **EXS** - element existence state; **TID** – tidal fluctuations category; **WID** – width value.
11	TreesA	Polygon	**DMT** – forest density in percent; **EXS** - element existence state; **NAM** - name; **PHT** - predominant height; **VEG** - vegetation characteristics.
12	TrackL	Line	**WTC** - practicability depending on weather conditions.
13	TrailL	Line	**WTC** - practicability depending on weather conditions.
14	TeleL	Line	**EXS** - element existence state.
15	PowerLine	Line	**ACC** - precision representation; **TST** - power line support type.
16	CropA	Polygon	**FTC** - farm type; **VEG** - vegetation type.

The algorithm used to determine corresponding nodes for one selected matching edge, designed by authors, consists of the following steps:

a) determination of the buffer zone around the edgematching side, based on distance chosen by operator;

b) creation of a collection of objects, called *colNodes_Int*, that stores all the nodes, from selected map sheet database, which are located in buffer zone. If there are no elements in created collection, the algorithm stops;

c) creation of another collection of objects, called *colNodes_Ext*, in which there are stored the nodes from the neighboring map sheet database that are located in the buffer zone. If there are no elements in the created collection, the algorithm stops;

d) the first item from *colNodes_Int* collection is selected. At subsequent iterations, the next object in the collection is selected, until the last item is reached. The selected node at this step will be mentioned forwards as **selected_node**. If all nodes from this collection were analyzed, then algorithm jumps to step (h);

e) the nodes from *colNodes_Ext* collection which are in the search distance range of the **selected_node** are determined. Further, this nodes selection group will be referred to as **tested_nodes**;

f) from **tested_nodes,** the closest node to **selected_node** group is chosen for testing. If no node is found within search distance range, then the algorithm resumes the execution from step (d);

g) for both **selected_node** and the node obtained at step (f) the attribute values and names of parent feature classes are compared. If all analyzed values are found to be equal, then these two matching nodes are saved in another collection of objects, called *colEdgematch_Nodes*, and after removing these nodes from *colNodes_Int* and *colNodes_Ext* collections, the algorithm returns at step (d). If there is no match, the tested node is excluded from **tested_nodes** group and the algorithm resumes from step (f). *Phase I* of processing is considered to be completed when all items from *colNodes_Int* were selected for matching analysis;

h) steps (d)-(g) are repeated, only this time the processing order of remaining nodes in initial collections is reversed, meaning that nodes are selected successively from *colNodes_Ext* collection, looking for corresponding nodes in *colNodes_Int* collection (*phase II*);

i) if the user checked the option to connect possible corresponding nodes from unmatched ones in *colNodes_Int* and *colNodes_Ext* collections, then steps (d)-(g) are repeated (phase III), the difference from first phase consisting in checking only the corresponding feature classes names (not attribute values);

j) the algorithm repeats step (h), firstly treating the remaining nodes in *colNodes_Ext* collection;

k) finally, unmatched nodes and possible corresponding nodes from the two collections, along with attribute values of theirs “parent” feature classes, are saved in an ESRI shapefile, used further by operator to analyze and manually correct the errors.

V. Practical Results

After implementation of proposed algorithm for automated vector matching, the application was initially used to connect features of two VMap2 databases, corresponding to 4090-II and 4090-III TLM50K adjacent map sheets.

The input data used are the following:

- selected map sheet: *4090-II*;
- side(s) used for edgematch: *West* (neighboring map sheet - 4090 III);
- method of matching features: *mediation*;
- buffer distance used to obtain nodes for testing, from both adjacent map sheets: *6 m*;
- search distance for corresponding nodes: *10 m*;
- connect nodes with different attribute values: *enabled*.

The program ran on a workstation with 3 GHz processor frequency and 2 GB RAM and its execution has been completed in 25 seconds, and a visual extract of matching results are shown in Fig. 4.

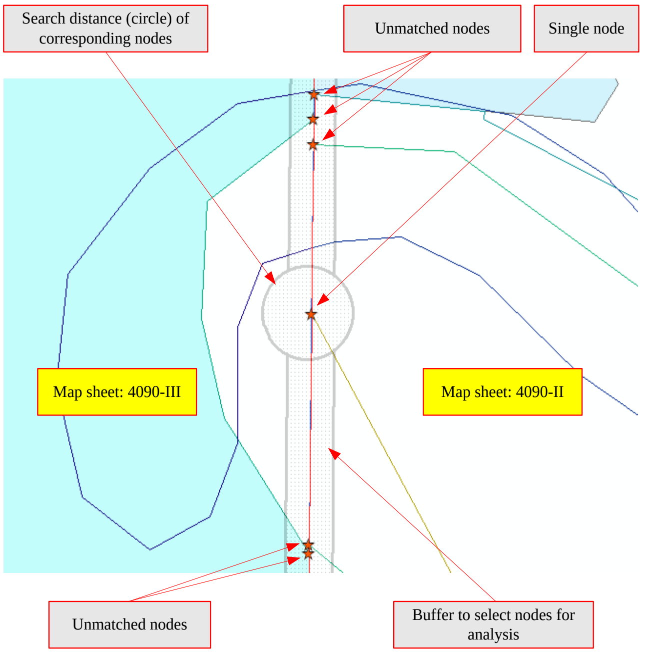

Figure 4. Exemplification of results after running application

Intermediate and final results, consisting of the numbers of corresponding and possible corresponding nodes determined in different phases of execution, are presented in Table II.

Table II. Achieved Results after Edgematch.dll Execution, for Western Side of 4090 II Map Sheet

	Initially	Corresponding nodes determination		Possible corresponding nodes determination	
		I	II	III	IV
Number of nodes in *colNodes_Int* collection	187	50	45	29	27
Number of nodes in *colNodes_Ext* collection	190	53	48	32	30
Number of corresponding nodes	0	**137**	**142**	**158**	**160**

VI. CONCLUSIONS

The presented application completely automates the stage of connecting features from adjacent map sheets stored in VMap2 format and allows vectors from the selected map sheet to be edgematched with those from one or all four neighboring map sheet databases. Adjoining corresponding features are detected and connected further by moving corresponding and possibly corresponding topological nodes.

At the same time, the software program checks descriptive attribute values for sixteen feature classes, which contain common topographic elements met on national territory. The other feature classes have few or no descriptive attributes, or appear less frequently and, consequently, only their names are checked.

The application can be easily extended to compare descriptive attribute values for all feature classes from full VMap2 database structure. Errors that occur after automated connection, consisting in nodes without correspondent or nodes with different attribute values for parent feature classes, are manually corrected by operators.

After the testing phase, the following conclusions have resulted regarding the software application:

- ensures the certainty of moving the corresponding nodes and possible corresponding nodes exactly on the chosen matching side(s) for selected TLM50 map sheet;
- contributes to the efficiency of the overall VMap2 database updating process by considerably reducing the edgematching time. Typically, depending on the number of corresponding elements to be connected along one common side of two adjacent map sheets, an operator can perform edgematch operation interactively in minimum 1 hour and maximum 40 hours. The realized software reduces the time needed for matching corresponding elements to maximum 1 minute per side, including initial data setting;
- provides an ESRI shapefile containing remaining unconnected nodes in order to be used by operator to clarify occurred errors.

The realized software could be used in the overall quality control stage for updated TLM50k maps sheets in VMap2 format, taking into account that it checks only attribute values of the elements from Table I.

The presented algorithm is very robust and it ensures database consistency and accuracy, mainly due to the use of topological relationships between feature classes, which can be used to correct possible overlapping errors, but also to the use of topological nodes when reshaping areal or linear vectors.

In order to extend the use of the application, future work would take into consideration the algorithm improvement by adding the possibilities to automatically match vectors stored in VMap level 1 and LTDS databases.

ACKNOWLEDGMENT

The authors gratefully thank (to) the reviewers for their support and valuable comments, which helped to improve this article.

REFERENCES

[1] J. B. Mena, "*State of the art on automatic road extraction for GIS update: a novel classification*," Pattern Recognition Letters 24, pp. 3037-3058, Elsevier B.V., 2003.

[2] T. Ohlhof, T. Emge, W. Reinhardt, K. Leukert, C. Heipke, K. Pakzad, "Generation and update of VMap data using satellite and airborne imagery", International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, Part B4, Amsterdam, 2000.

[3] L. Porumb, "Military maps updating using photogrammetric methods," Ph.D. dissertation, Military Technical Academy, Bucharest, Romania, 2010.

[4] D. Pantazis, H. Karathanassis, D. Stathakis, L. Yingcheng, T. Guo, K. Santimpantakis, "The update of middle scale topographic maps using highresolution satellite images: a semi-automatic method proposal. Results, critics, conclusions and perspectives from a bilateral research project", in 8th AGILE Conference on "Geographic Information Science", Estoril, Portugal, 2005. doi:10.13140/2.1.4724.3205

[5] ESRI**,** ArcGIS 10.2.2 Desktop Help, 2014.

[6] MIL-STD-2407, "Vector Product Format", U.S. Army, 28 June 1996.

[7] MIL-PRF-89039, "Performance specification for Vector Map Level 0", U.S. Army, 09 February 1995.

[8] MIL-PRF-89032, "Performance specification for Vector Map Level 1", U.S. Army, 01 June 1995.

[9] MIL-PRF-89033, "Performance specification for Vector Map Level 2", draft, U.S. Army, 15 April 1996.

[10] MIL-PRF-89035A, Performance specification for Urban Vector Map, U.S. Army, 01 August 2002.

[11] STANAG 2592, "NATO Geospatial Information Framework (NGIF) - AGeoP-11" Edition A, NATO standardization agreement, 22 October 2018.

[12] MIL-T-89301A, "Military specifications for 1:50,000 scale topographic maps", U.S. Army, 28 February 1995.

 Digital Object Identifier 10.32754/JMT.2021.1.03