A Linear Approach to Improving the Accuracy of City Planning and OpenStreetMap Road Datasets

Alexey Noskov and Yerach Doytsher
Mapping and Geo-Information Engineering
Technion – Israel Institute of Technology
Haifa, Israel
emails: {noskov, doytsher}@technion.ac.il

Abstract—**The developed method allows the user to integrate polygonal or linear datasets. Most existing approaches do not work well in the case of partial equality of polygons or polylines. The suggested method consists of two phases: searching for counterpart boundaries or polylines by a triangulation, and rectifying objects without correspondent polylines by a transformation and a shortest path algorithm. Data covering the Haifa region of Israel have been used for evaluation of the approach. City Planning datasets have been rectified by precise cadastre data. Positional accuracy of the City Planning datasets has been increased significantly. Average distance between segments of the datasets has been decreased in almost five times. Standard deviation has been decreased by thirty-five percent. In addition, more complete road layer of OpenStreetMap covering the city has been rectified by a more precise statutory road layer. Positional accuracy of the rectified layer has been improved significantly. The rectified layer has been utilized to prepare a large-scale map depicting roads with individual widths and statutory buildings. OpenStreetMap rasterization rules have been applied for road widths calculation. The prepared map depicts real-size buildings and roads' widths in scale.**

Keywords-Geometry spatial data integration; triangulation; shortest path; topology; OpenStreetMap; road layers; city planning and cadastral datasets.

I. INTRODUCTION

This paper is an extended version of the work published in [1]. In order to confirm the effectiveness of our algorithm, a rectification of an OpenStreetMap road layer by more accurate statutory road data is considered

We live in the information age. Terabytes of spatial information are available today. Hundreds of sources produce thousands of maps and digital layers every day. We encounter serious problems when trying to use different maps together.

Let us list some popular data producers. Survey companies and agencies prepare accurate topographic maps and plans. Aero and satellite images act as a basis for numerous variations of derivative maps (e.g., thematic and topographic maps). A special niche is reserved for crowd sourcing maps, e.g., OpenStreetMap (OSM) [5]. Significant parts of this sort of map contain data derived from users' devices, mainly GPS devices.

It is very difficult to use all these data together. In many cases, the user decides to draw a map from scratch, despite having existing maps with most of the required elements for the user's map. One of the reasons for this situation is a low degree of integration of existing datasets even when we consider maps containing many identical elements. For instance, soil maps need to be based on topographic maps. Today, soil maps could take basic contours from different sources.

In an ideal situation, spatial datasets use the objects (polylines or polygons) from more accurate datasets. In the real world, many maps are produced by measuring/digitizing objects from satellite images. As a result, despite the fact that most of the objects on different maps are identical, they are presented with small positional discrepancies. The problem is compounded by the fact that different objects in a Geographic Information System (GIS) environment could be depicted by the same geometries (e.g., square or circle). Thus, specific tools and algorithms need to be developed. This makes it difficult to detect identical objects on different maps. The obvious advantage of integrated databases is efficiency of data storing. Equal elements from different maps link to the same object in the storage memory. We do not need to take up extra storage on a disk. Additionally, the editing of objects will be reflected on all those maps which contain them.

The benefits of data integration are demonstrated in this paper by using city planning and cadastral datasets. A cadastral map is a comprehensive register of the real estate boundaries of a country. Cadastral data are produced using quality large-scale surveying with TotalStations, Differential Global Positioning System devices or other surveying systems with a centimeters-level precision. Normally, the precision of maps based on non-survey large-scale data (e.g., satellite images) is lower. City planning data contain proposals for developing urban areas. Most city planning maps are developed by digitizing handmade maps, using images from space. Almost all boundaries have small discrepancies in comparison to cadastral maps. We need to integrate these datasets, where the identical elements in the datasets have to be linked to the same geometries. All the non-identical elements have to be coherent with shared geometries.

In addition, our algorithm was tested on road datasets. Data covering Haifa City (Israel) were used. An OpenStreetMap (OSM) road layer was rectified by a statutory (more accurate) road layer provided by Survey of Israel (SOI). Road data provided by SOI do not contain width attribute. Roads' widths could be obtained from the

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

OSM road layer. In order to integrate an OSM road layer with SOI data (e.g., building and landuse layers), the OSM road layer was rectified using linear approach and SOI road layer.

The approach we suggest enables the user to resolve the described problems. It consists of two main stages: defining correspondent boundaries using triangulation technique, and rectification of the remaining polylines by transformation and the shortest path algorithm. The suggested approach could be applied to polygonal and linear datasets.

This paper is structured as follows: related work is considered in Section II. The initial processing of the source datasets is described in Section III. Section IV focuses on correspondent boundary definition. The problem of resolving line pair conflicts is described in Section V. The shortest path approach for fusion boundaries with and without counterpart is discussed in Section VI. The rectified City Planning data are discussed in Section VII. A review of road datasets is presented in Section VIII. Preprocessing of an OpenStreetMap road dataset is described in Section IX. In Section X, a process of rectifying road layer is presented. Calculating the widths of OpenStreetMap roads is discussed in Section XI. The conclusion is presented in Section XII.

II. Related Work

The main groups of approaches for data matching and data fusion are considered in this section

The wide spread of databases is the reason for developing attribute-based matching methods. Schema-based [18] and Ontology-based types of attribute matching could be selected. In [23], an approach based on both types is presented. Attribute-based matching could be effective when data with sustainable and meaningful structure and content of attribute database is processed.

The map conflation approaches [19] are based on data fusion algorithms; the aim of the process is to prepare a map which is a combination of two or more maps [8]. The merging and fusion of heterogeneous databases has been extensively studied, both spatially [16] and non-spatially [25].

Geometry, size, or area is used in feature-based matching. These allow us to estimate the degree of compatibility of objects. The process is carried out by the structural analysis of a set of objects and analysis of the result, to see whether similar structural analysis of the candidates fits the objects of the other data set [4]. In [22], comparison of objects is based on the analysis of a contour distribution histogram. A polar coordinates approach for calculating the histogram is used. A method based on the Wasserstein distance was published by Schmitzer et al. [20]. A special shape descriptor for defined correspondent objects on raster images was developed by Ma and Longin [13]. Focusing on single shapes does not allow us to apply these algorithms in our task.

In [7], topological and spatial neighborly relations between two datasets, preserved even after running operations such as rotation or scale, were discovered. In relational matching, the comparison of the object is implemented with respect to a neighboring object. We can verify the similarity of two objects by considering neighboring objects. The problem of non-rigid shape recognition is studied by Bronstein et al. [6]; the applicability of diffusion distances within the Gromov-Hausdorff framework and the presence of topological changes have been explored in this paper.

In [2], spatial data integration is considered as a process of unifying layers in a unique database to provide a unified environment for processing, modeling, and visualization. Three main aspects are considered: spatial reference of the data, projection of the data, and format of the data. A geo-spatial data integration method for three-dimensional subsurface stratification is proposed by Kim et al. in [10]. In [26], integration of remotely sensed data is considered. A proposed framework can provide an effective solution for distributed storage, data format conversion and interoperability for satellite remote sensing big data. Foster and Mayfield consider geospatial data integration in the context of defense and security. Integration of global land cover datasets was considered in [24]. Kipf and Kemper extended high-performance main-memory database systems with temporal and geospatial processing capabilities to tackle emerging mobility workloads in [11]. Integration of geospatial data regarding crimes is discussed in [17].

We concluded that the approaches mentioned could not be applied to resolve the considered problem. This derives from the fact that the mentioned approaches have been developed for specific conditions. For instance, feature-based matching is effective for detecting separate outstanding objects; attribute-based matching is effective for definite and well-designed databases. Thus, a new approach should be developed.

III. City Planning Data preparation

Spatial data sets covering a part of Yokne'am (a town in the northern part of Israel) have been used. They are depicted in Figure 1. Land-use city planning and cadastre polygons are displayed as color areas and as black boundaries, correspondingly. As can be seen in the figure, in most cases the boundaries of two datasets are the same. Some boundaries are presented in the first dataset and are not presented in the second. The white background of the cadastre polygons means that this area is not covered by the city planning dataset. It is presented mainly in the upper part of the figure. The case where black cadastre boundaries cross an area with a similar background color means that these boundaries are not presented in the city planning datasets.

The city planning data have sensitive positional irregular discrepancies. Because of the small scale, they cannot be observed in Figure 1; hence, the problem is illustrated in Figure 2. The figure shows that the problem could not be resolved by transformation only, and that a more sophisticated technique is required. The figure leads us to an approach based on defining corresponding objects and further modification of the remaining objects with respect to found pairs.

In the previous approach [15], we defined correspondences between polygons. We encountered two

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

problems. Because whole polygons are processed, it is difficult to precisely define the points connecting polygons with and without counterparts. Considering a polygon as a separate object does not allow us to unambiguously detect polygons' shared nodes. As a result, in some cases, it is difficult to correctly eliminate gaps between objects. Using centroids in the polygon triangulation approach is the reason for the second problem. For non-compact polygons, even small changes in the polygon's boundary lead to significant changes in the centroid position. It could negatively impact the results.

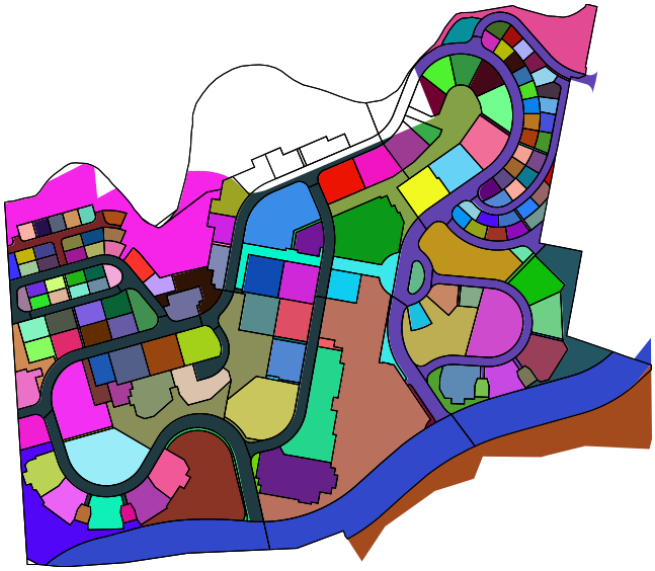

Figure 1. Source data: land-use city planning (colored background) and cadastre (black outline) maps.

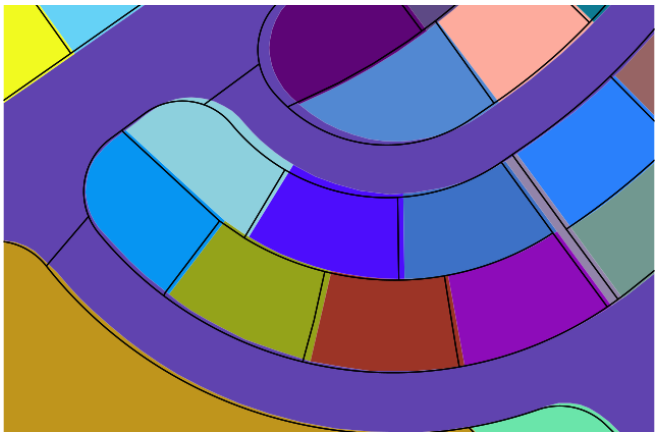

Figure 2. Positional discrepancies of city planning (colored areas) and cadastre (black lines) datasets.

In this paper, we propose a technique which is based on defining line pairs by triangulation. In most cases, spatial data are found in non-topological data format (e.g., ESRI's Shape Files, GeoJSON, MapInfo Tab Files). This means that the boundaries of neighboring objects are repeated for each polygon. This fact leads us to the possibility of modifying the boundary of neighbor polygons independently. In the most cases, it is a source of many difficulties; e.g., small gaps between boundaries or the necessity of repeating the same action for each polygon separately. Because of the problems mentioned we use topological data format provided by GRASS GIS 7 [12]. The source shape files have been converted to this format. A sample part of the city planning dataset found in a topological format is presented in Figure 3. Polygon data comprise 3 types of elements: boundary, node, and centroids. Nodes separate boundary polylines. Each group of closed boundaries could be considered as an area. The polygons' centroids link the polygons to certain rows in an attribute table by category numbers. Each raw in the attribute table starts with a "cat" field, which could be connected to a centroid with a given "cat" value.

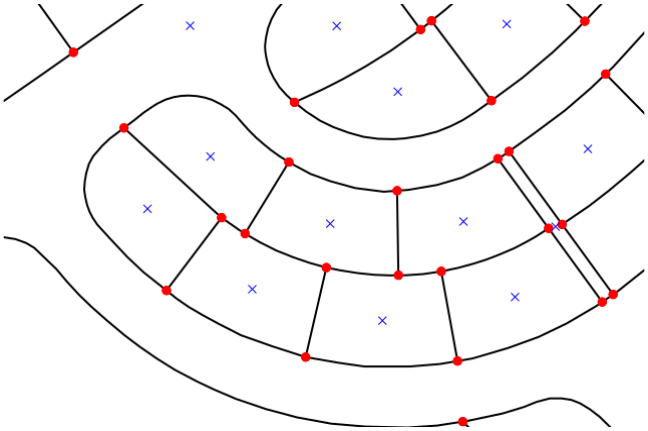

Figure 3. A sample of the city planning dataset residing in GRASS GIS's topologycal format. Nodes – red circles, centroids – blue croses, and boundaries – black lines.

We can conclude from the first two figures, that most of the counterpart polygon boundaries of the datasets are located close to each other and present the same objects. It is efficient to define a measure for detecting the fact that two objects certainly could not be defined as counterparts. In other words, we can use it as a filter. A maximal distance parameter could fulfill this role.

In addition, it is quite popular to use buffers for detecting the fact that two objects certainly could not be defined as counterparts. For instance, in [27] the authors have applied a buffer with a certain buffer size, where all objects outside the buffer could not be considered as counterparts. We have found that a segmentation technique could be more sensitive and flexible in this context. Segmentation means dividing polygon boundaries (or any other sort of polyline) into equidistant segments. Point delimiters are used to calculate distances between the considered datasets. An example of segmentation is depicted in Figure 4.

Maximal distance (D_{max}) is calculated as follows. For each point in the first dataset, a distance to the closest point belonging to the second dataset is assigned. Then we apply a loop from the first to the last percentile (from the percentile with maximal number and minimal distance to that with minimal number and maximal distance) on a list of 100 percentiles of the calculated distances. D_{max} equals percentile i if the standard deviation of distances between percentiles i-

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

1 and i is more then 1. D_{max} is used mainly to filter considered objects. In our case, the distances between the nearest equidistant points of the cadastre and the city planning data sets' boundaries are in an interval from 0 to 92.7 meters. The boundaries of the percentiles number (i.e., i decrement) 6, 5, 4, and 3 are 2.09, 4.97, 7.88, and 17.75 meters, correspondingly. Standard deviations for distances in intervals between percentiles 6-5, 5-4, and 4-3 are as follows: 0.78, 0.89, 1.46, and 2.77. Hence, D_{max} equals 7.88, because 7.88 belongs to percentile number 4 (the first with a standard deviation of more than 1). Objects residing further than D_{max} are excluded from the processing. For Yokne'am datasets, D_{max} equals 7.9 meters. A 2-meter distance between nearest points has been assigned for our test.

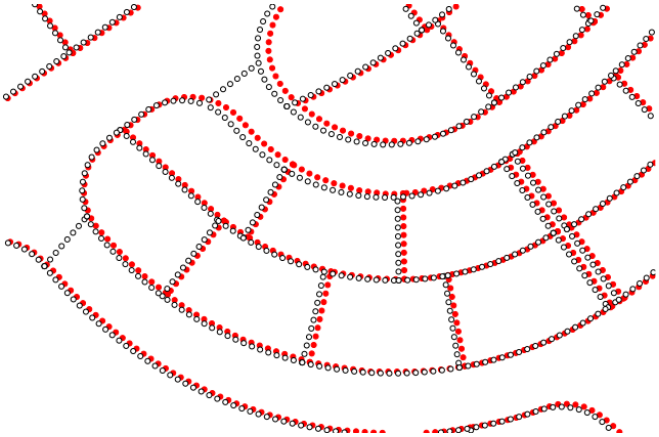

Figure 4. Point delimiter of equidistant segments. City planning – red, cadastre - black points.

IV. DEFINING CORRESPONDING LINES OF DATASETS BY TRIANGULATION

In this section, the main process is described. It is based on identifying correspondent triples of polygon boundaries of the considered datasets. Delaunay triangulation enables us to easily connect points by triangles. We use it to divide boundaries into triples. Figure 5 illustrates the triangulation process. The triangulation is based on the middle points of boundaries' polylines. In the figure, the boundaries' middle points are depicted as gray circles; the boundaries are colored lines; and the triangulation layer is presented as a colored background.

Now, we have grouped middle points into triples boundaries of cadastre and city planning datasets. The next step is searching for correspondent triple candidates, and it is implemented as follows.

First, the lengths of all boundary polylines are calculated. Sorted lengths of correspondent boundaries are stored into "A", "B" and "C" fields of attribute table for each triple. "A" stores the shortest length; "C" stores the longest. Then, we compare all possible pairs of triples.

To reduce the number of comparisons we consider only the nearest triples. These are defined by comparing the coordinates of the start and end nodes of their boundaries. For further consideration, all start and end nodes of the second triple boundaries have to be inside the extent of the first triple's nodes (defined by an enlarged buffer). Buffer size is equal to the square root of the median polygon area. In our case it is 32 meters. The areas of both datasets are sorted into one list to find a median value.

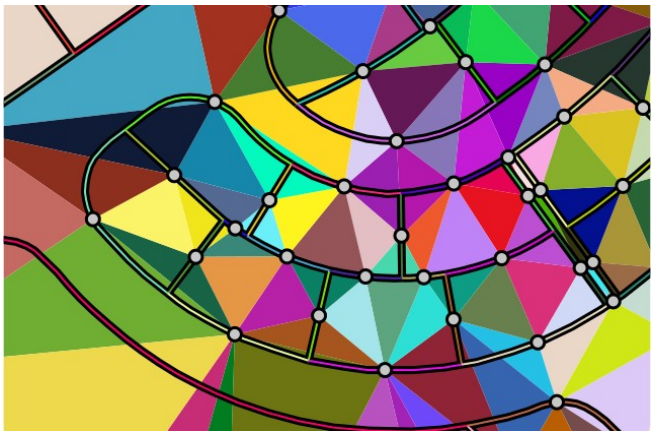

Figure 5. The triangulation of boundaries' middle points of a city planning dataset.

In the next step, we compare boundary lengths. As mentioned above, ordered lengths are stored in an attribute table ("A", "B" and "C" fields). Triple pairs are added into a list for further processing if a correspondent length (A-A, B-B, or C-C) resident in the second triple is within an interval of between 80% to 120% of a length resident in the first triple, and are considered as triple pair candidates. This two-step initial filter by extents and lengths comparison is illustrated in Figure 6. In the figure, blue lines are city planning boundaries; black lines are cadastre boundaries; grey and green triangles are candidate cadastre boundaries obtained by an extent (red rectangle) and by length comparisons, correspondingly. Candidates are defined for a triple of city planning boundaries marked by a red triangle.

At this point, we have a few candidates. In order to define the "winner" candidate, we calculate distances between nodes of the correspondent boundaries. We need to determine pair boundaries belonging to a considered triple candidate. The brute force process is implemented; all possible combinations are considered. The most acceptable combination is one with a minimal sum of distances between correspondent points. The brute force process is not time sensitive, because it is implemented only for a few filtered candidates. A candidate is marked as a triple pair if the maximal distance between correspondent nodes is less than D_{max}, as defined in Section III.

In this section, correspondent boundaries have been defined. The candidate triples have been filtered by extent and lengths comparison, then line pairs have been defined by distances between nodes.

V. RESOLVING LINE PAIRS' CONFLICTS

In this section, we describe the process of searching for wrongly defined boundary pairs and resolving these situations.

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

First of all, in many cases line pairs are repeated in neighboring triples. The participation of a line in different pairs is marked as a problem. It is quite obvious that a boundary from the first dataset could have only one counterpart boundary in the second dataset. In order to resolve conflicts, we compare the number of times they participate in triples. For instance, we have two line pairs A1-B1 and A1-B2. If A1-B1 pair is encountered in 2 triples and A1-B2 in 1, then the combination A1-B2 is eliminated and A1-B1 remains. If both are encountered simultaneously, both candidates are eliminated.

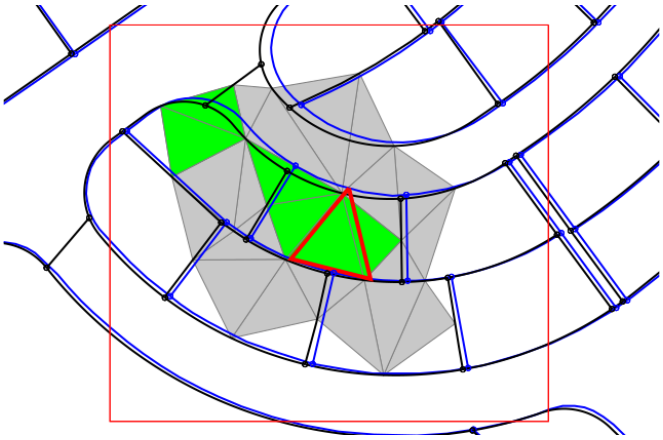

Figure 6. Filtering possible triple pairs.

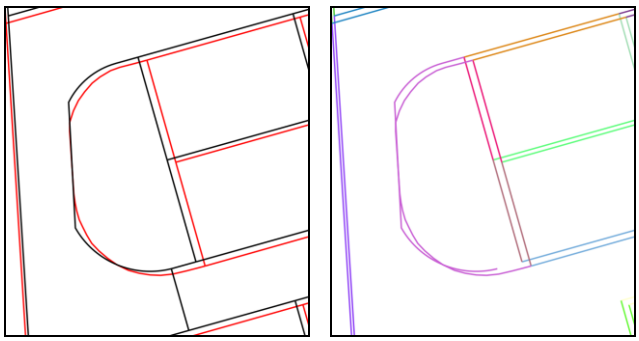

Figure 7. An example of a line pair found incorrectly. Left – original boundaries of the city planning (red lines) and cadastre (black lines) datasets. Right – detected linepairs.

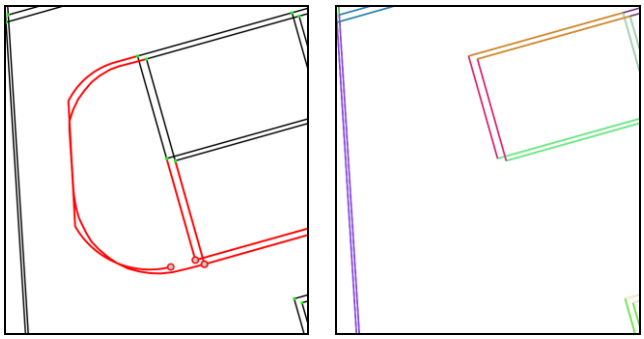

Figure 8. Detecting incorrect pairs. Left – incorrect nodes and line pairs are marked in red. Right – final line pairs.

Additionally, we need to consider the situation illustrated in Figure 7. The curved purple line pair is detected incorrectly. This line is composed of two lines in the cadastre dataset, because of the line, which is connected to the bottom part. The connected line does not exist in the city planning dataset.

These types of errors could be detected by analyzing the line junctions. Each node is identified by a set of ids of lines connected to the node. The required conditions for the remaining line pairs are as follows. First, node values (a set of ids of lines) have to be unique. Second, each node has to have a node of equal value, and vise versa. If one of the conditions is false, all lines connecting with the incorrect node are eliminated on both datasets. The process is illustrated in Figure 8.

VI. A SHORTEST PATH APPROACH FOR BOUNDARIES FUSION

At this point, we have the pairs of corresponding boundaries. As mentioned in Section I, cadastre datasets are produced using quality large-scale data. They are more accurate than city planning datasets. Hence, replacing the city planning boundaries with their cadastre counterparts will significantly improve the accuracy of the resulting map. This was done in the previous step. In this section, we consider how to integrate boundaries without counterparts with pair boundaries. This is implemented in two steps.

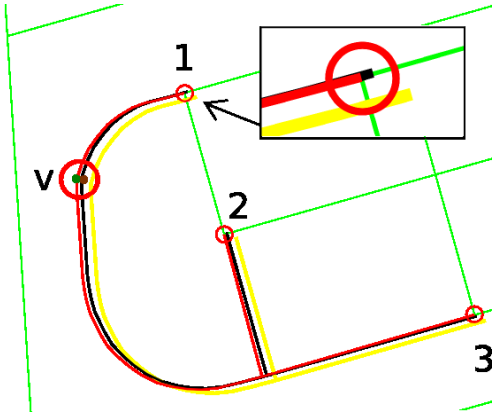

Figure 9. A vertex moved with respect to the shortest paths to bridge nodes.

In the first step we use coordinates of correspondent pair nodes as Ground Control Points for second-order affine transformation. We transform the boundaries without counterpart to make them closer to the cadastre dataset. We shall henceforth call it “transformed boundaries or dataset”.

The transformed boundaries still have gaps between them and the remaining boundaries. A shortest path approach has been developed to integrate both types of boundaries.

The idea of the approach is quite simple. Each vertex (including nodes) of the transformed boundaries is processed. We calculate the shortest path from a vertex to each bridge node. Bridge nodes connect a nest (group of lines joined without gaps) of transformed boundaries to

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

boundaries with counterparts. In figure 9, the described elements are presented.

The figure explains the algorithm. Green lines are cadastre counterparts. Black lines are transformed city planning boundaries without pairs. They still have small gaps with cadastre counterparts. Red lines are the result of applying the shortest path approach to each vertex. Vertex v is the considered vertex and 1, 2, and 3 are the bridge nodes. Bridge nodes of a transformed dataset differ from the other nodes by having a counterpart node in the cadastre pair boundaries. Thus, we can precisely say how to move bridge nodes in order to locate them exactly on the node of cadastre boundaries with pairs. It is not correct to only move a bridge node; we need to move other vertices too.

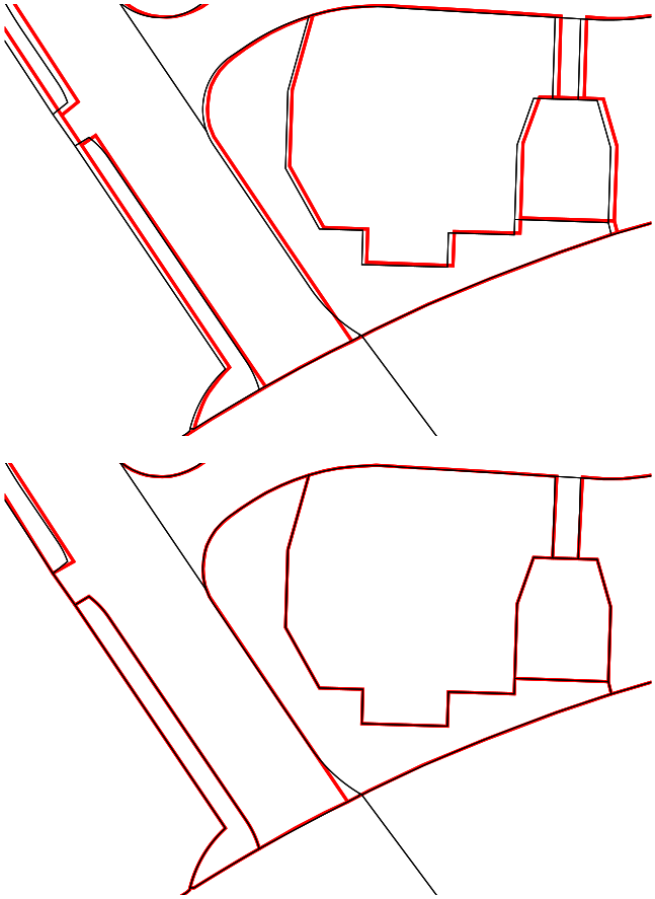

Figure 10. Zoomed-in extent 1. Boundaries of original (upper) and result (lower) datasets: city planning – red, cadastre - black.

To define new coordinates we use shortest paths. Three nodes are impacted for the vertex "v". Thus, three shortest paths are calculated: v-1, v-2, and v-3. v-2 and v-3 are partially overlapped paths. We need to note an important condition. If a path touches more then 1 bridge node, the path is eliminated from further consideration. Only paths intersected by one bridge node are considered. The new coordinates of a vertex are calculated as follows.

$$c_2 = c_1 + \sum_{0}^{n} (c_{oi} - c_{ti}) \cdot (1 - l_i / l_{sum}) \quad (1)$$

In (1), c denotes x or y coordinate; c_1 is the source coordinate; c_2 is the target. n is number of bridge nodes, i is index of the current bridge node. c_o and c_t are x or y coordinates of pair bridge nodes resident in cadastre counterpart and transformed (without pair) city planning boundaries, correspondingly. l_i is the length of the shortest path to be considered as a bridge node. l_{sum} is the sum of lengths of the shortest paths to bridge nodes from the vertex.

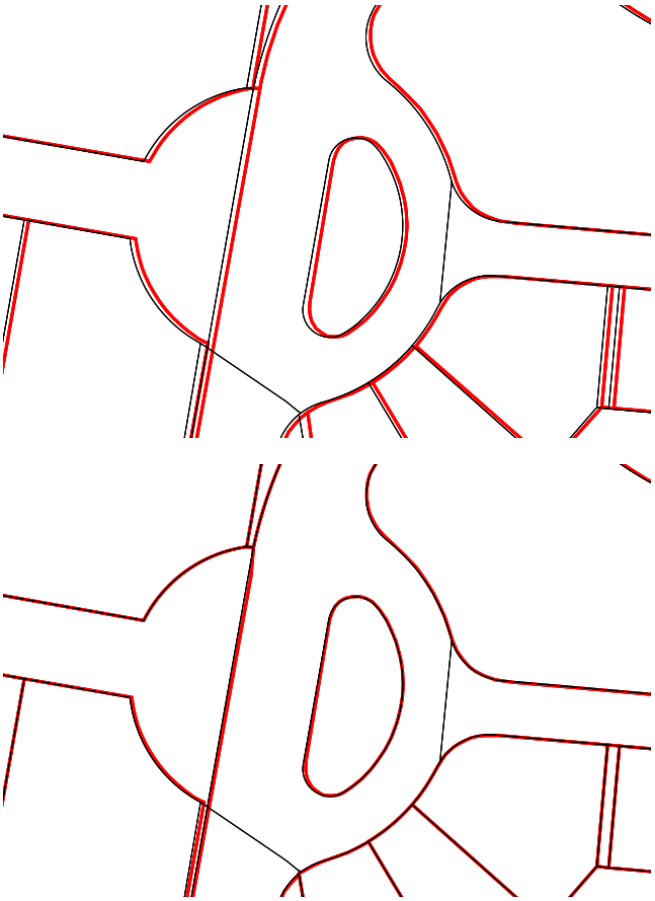

Figure 11. Zoomed-in extent 2. Bountaries of original (upper) and result (lower) datasets: city planning – red, cadastre - black.

Let us consider an example of calculating new coordinates by the shortest path method. We have 3 paths from vertex v to bridge nodes 1, 2 and 3. The paths' lengths are 19.8, 66.8, and 76.3. c_o - c_t values are (x y) -0.39 -0.14, -0.34 -0.24, and -0.23 0.16. For such parameters we need to add -0.67 -0.18 to the x y coordinates of the vertex.

VII. RESULTING CITY PLANNING DATASET

In order to acquire a final result, cadastre pairs of the boundaries are merged with the rectified boundaries without counterparts. Since pair boundaries have the same id and the rectified boundaries of the city planning dataset without cadastre pairs inherit the original ids, the correspondences between original and final polygons could be established by

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

comparing ids of boundaries comprising a polygon. It is derived from the fact that each polygon could be identified by a unique set of ids of boundaries.

TABLE I. AVERAGE DISTANCES AND STANDARD DEVIATIONS

Parameter	Dataset compared with cadastral layer	
	Original city planning	*Result city planning*
Average distance, m	1.15	0.24
Standard deviation, m	0.64	0.41

The result datasets are presented in Figure 10 and Figure 11. We can conclude that most boundaries have been taken from the cadastral dataset; others have been rectified to connect boundaries without corresponding pairs and boundaries with pairs. The result looks satisfactory; the final map is holistic and does not contain significant deficiencies. A review implemented by specialists enables us to state that the results are satisfactory.

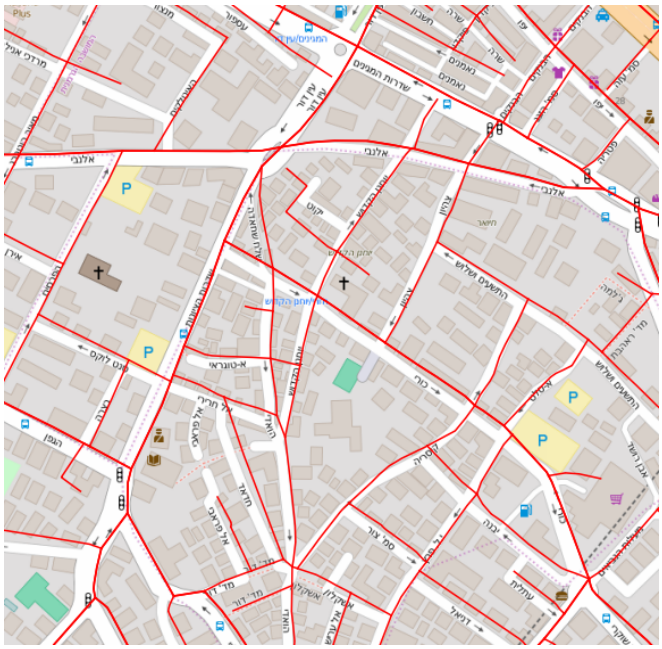

Figure 12. OpenStreetMap (background image) and SOI roads (red lines).

In order to estimate the results quantitatively, we use distances between the closest equidistant points of the cadastral and the city planning data sets' boundaries. The distances have been calculated between original city planning and cadastral datasets, as well as between the result and cadastral datasets. Only distances less than D_{max} have been taken into account. In Table I, average distances and standard deviations are presented.

According to the table, the average distance has been reduced five times over; standard deviation has been reduced by a factor of three. We can conclude from the table that the accuracy of the original dataset has been significantly improved.

Figure 13. Positional discrepancies of OSM (green) and SOI (red) road networks.

VIII. ROAD DATASETS REVIEW

A linear approach was successfully applied to City Planning polygonal data. In order to confirm the quality of our algorithms, another dataset was used. As mentioned at

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

the beginning of this paper, this approach could be applied to linear data as well.

We encountered a significant problem using geodata covering Haifa City area: an absence of widths in roads' attributes. In order to use a road network with known individual widths of roads, OSM road layer was rectified according to statutory SOI road maps. Unfortunately, OSM road layer covering Haifa does not contain width attributes. Road widths were obtained from the rules of road type rasterization.

The source datasets were retrieved from different sources in ESRI shape file format. The data were preprocessed with GDAL/OGR command line tools and converted to GRASS GIS 7 topological geodatabase.

Haifa datasets were provided by Survey of Israel (SOI). The data contain ESRI shape files: contour lines (line layer), roads (line layer), and buildings (polygon layer). These data are proprietary. In addition, the OpenStreetMap (OSM) road data covering Haifa were processed in this work. The data (actual shape files) were downloaded from Geofabrik web site.

SOI roads consist of only two types of roads (main and regular). Attributes allowing us to calculate individual road widths (even approximately) are not provided. Thus, another source of road dataset was found. We decided to use OpenStreetMap (OSM) data. They are freely available and the quality is fairly high. We could use the rules of rasterization vector OSM elements to raster tiles to define the width of individual road types. In Figure 12, OpenStreetMap and SOI roads are depicted.

In Figure 13, two extents of overlaid OSM and SOI maps are depicted. From the figure, we could note irregular positional discrepancies. OSM roads should be rectified. From the figure, one could conclude that, in many cases, OSM roads intersect SOI buildings and it is impossible to correctly set building-quarter correspondences. In this work, we will evaluate the quality of rectified data by intersections with building layers. If buildings are located correctly to the right or to the left of a road, the road is considered to be correctly rectified.

IX. PREPROCESSING OF OSM ROAD DATASET

Figure 14 reflects two significant problems. First, OSM roads contain many more road types (including paths, pedestrian ways, steps, etc.) than SOI. Second, circular intersections (roundabouts) are not presented in the SOI layer. Regular intersections are used instead. The preprocessing stage of data integrations consists of two steps: removing minor road types and eliminating circular intersections.

Excessive OSM roads were removed by the following SQL request: "type NOT IN ('construction', 'cycle way', 'footway', 'path', 'pedestrian', 'platform', 'proposed', 'steps', 'track', 'bridleway', 'rest area') AND type NOT LIKE '%link%' AND tunnel=0". This SQL request eliminated all road lines contained in 'type' string column word 'link', roads with 'tunnel' attributes equaling '1', and minor road types: 'construction', 'cycle way', 'footway', 'path', 'pedestrian', 'platform', 'proposed', 'steps', 'track', 'bridleway', and 'rest area'.

Figure 14. OSM (red) and SOI (green) roads.

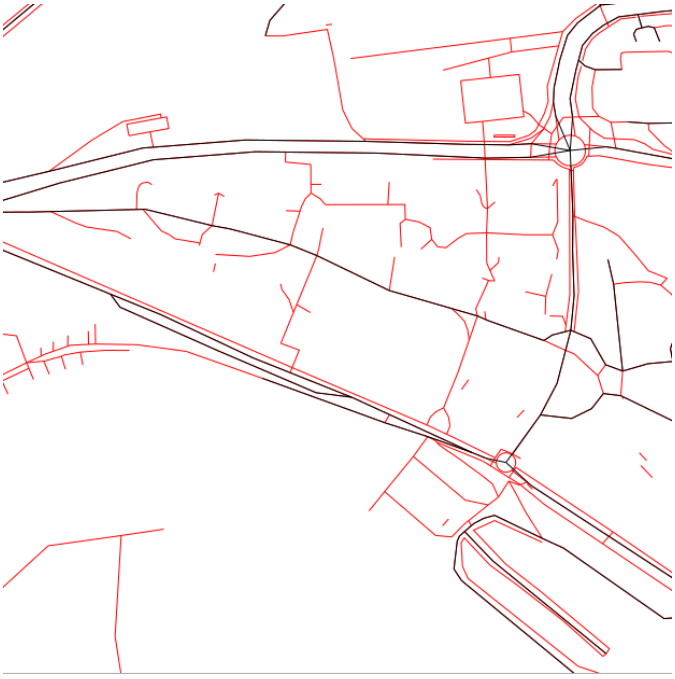

Figure 15. Cleaned OSM roads: black –result roads, red –removed.

Next, we need to calculate the compactness of polygons formed by closed road segments using the following equation.

$$compactness = perimeter / (2 \cdot \sqrt{\pi \cdot area}) \quad (2)$$

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

In the next step, segments constructing polygons with compactness < 1.01 (i.e., circular intersections) are removed. In the final step, surrounding polylines nodes are snapped to the centroids of polygons formed by removed polylines.

Figure 16. Carmel Center, Haifa. Original data – upper, rectified data – lower. Brown – SOI buildings, black – SOI roads, red – OSM roads.

In Figure 15, the OSM cleaning results are presented. Minor roads have been removed. The circular intersections have been replaced by regular intersections.

Now, the OSM roads dataset is more suitable for integration with an SOI road layer. Roads datasets are quite complex for processing (many intersections of polylines, complex topology, different types of roads, dissimilarity of counterpart objects, etc.).

Figure 17. Carmel Center, Haifa. Original data – upper, rectified data – lower. Brown – SOI buildings, black – SOI roads, red – OSM roads.

X. Applying a Linear Approach for OSM Road Dataset and Evaluating Results

In order to improve the positional accuracy of an OSM road network and integrate it with SOI road data, the developed linear fapproach was applied. OSM road data

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

were rectified. Figure 16 and Figure 17 demonstrate the results of the integration.

In order to estimate the quality of a rectified road layer, building and road datasets were converted to raster layers (resolution 1 meter). The pixels touching vector objects got value "1"; the reminded pixels got value "0". The road raster layers were overlaid with building raster layers. In raster algebra terms [21], "AND" or "&&" condition was applied and pixels with value "1" on two overlaid dataset were selected. The following table contains numbers of pixels with value "1" from different raster maps.

TABLE II. STATISTICS OF RECTIFICATION RESULTS ("1"-VALUED PIXEL NUMBER OR SQUARE METERS).

Source data (square meters or number of pixels)			
SOI Buildings	6,367,165	OSM Buildings	6,167,209
SOI Roads	411,189	OSM Roads	639,855
Overlay			
OSM roads && OSM buildings	1,371	OSM roads && SOI buildings	39,589
SOI roads && SOI buildings	291	Rect. OSM roads && SOI buildings	3,067

According to the table, the number of intersections of OSM roads and SOI buildings was significantly reduced, from 39,589 to 3,067. We can conclude from this that the rectification results are satisfactory. The rectified OSM road network can be used in further research.

XI. CALCULATING WIDTHS OF OSM ROADS

Now, we need to calculate the widths of roads. OSM data covering the Haifa area do not have width attributes, but the rasterization rules of OSM vector data allows us to get road widths in pixels. The OSM wiki web page [28] provides an equation for estimating pixel size in meters for any zoom level:

$$Ps_z = \frac{C \cdot \cos y}{2^{z+8}} \quad (3)$$

Where, Ps is pixel size, z is zoom level, C is the (equatorial) circumference of the Earth; y is the latitude of the position. According to the equation, tile pixel size of zoom level 15 and 16 could be defined as follows:

$$Ps_{16} = \frac{40075696 \cos 32.795}{2^{16+8}} \approx 2 \quad (4)$$

$$Ps_{15} = \frac{40075696 \cdot \cos 32.795}{2^{15+8}} \approx 4 \quad (5)$$

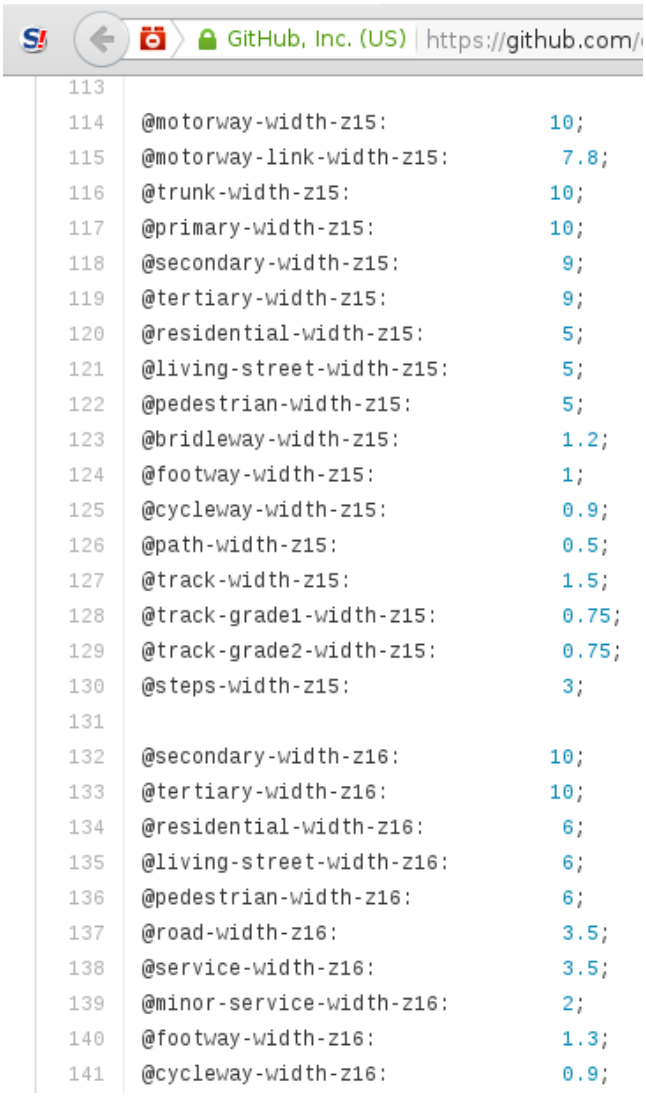

Figure 18. CartoCSS web site, roads.mss files actual lines. Line widths in pixels.

The widths of roads are equal to Ps multiplied by width in pixels. Width in pixels could be derived from CartoCSS open project (road.mss file [29]). The screenshot of the actual lines of the style sheet is presented in Figure 18.

The calculated road types' widths are presented in Table III.

In Figure 19, a map of OSM roads is depicted. The map is colored randomly by road type. Each road has a width according to Table III.

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

XII. CONCLUSION AND FUTURE WORK

An approach for improving linear and polygonal spatial datasets is presented. Land-use city planning dataset locations have been corrected according to the cadastral dataset.

TABLE III. ROAD TYPES' WIDTHS.

Type	Pixels	Zoom level (pixel size)	Width (meters)
Living Street	6	16 (2)	12
Motorway	10	15 (4)	40
Primary	10	15 (4)	40
Residential	6	16 (2)	12
Road	3.5	16 (2)	7
Secondary	10	16 (2)	20
Service	3.5	16 (2)	7
Tertiary	10	16 (2)	20
Unclassified	6	16 (2)	12

The outline of the approach is as follows. The conventional polygon data have been converted to topological data format. Boundaries have been split into equidistant segments to calculate D_{max}. Then, correspondent boundaries have been defined using triangulation technique. Rectification of the remaining polylines by transformation and the shortest path algorithm has been implemented.

The developed algorithm has been tested on a city planning polygonal dataset and an OpenStreetMap road linear layer. In both cases, the resulting datasets are satisfactory. The resulting data quality has been evaluated by two different approaches: the first approach is based on equidistant points' statistics, while the second approach is based on defining intersections of building and road layers.

In the future, we need to test the approach with more datasets and different parameters, to compare it with other approaches. In order to improve the presented approach by also defining correspondences between parts of boundaries (not only whole boundaries), we would like to combine this approach with the segmentation-based algorithm published in [14]. This will allow us to apply the method to other types of datasets.

ACKNOWLEDGEMENT

This research was supported by the Survey of Israel as a part of Project 2019317. The authors would like to thank the Survey of Israel for providing the financial support and data for the purpose of this research.

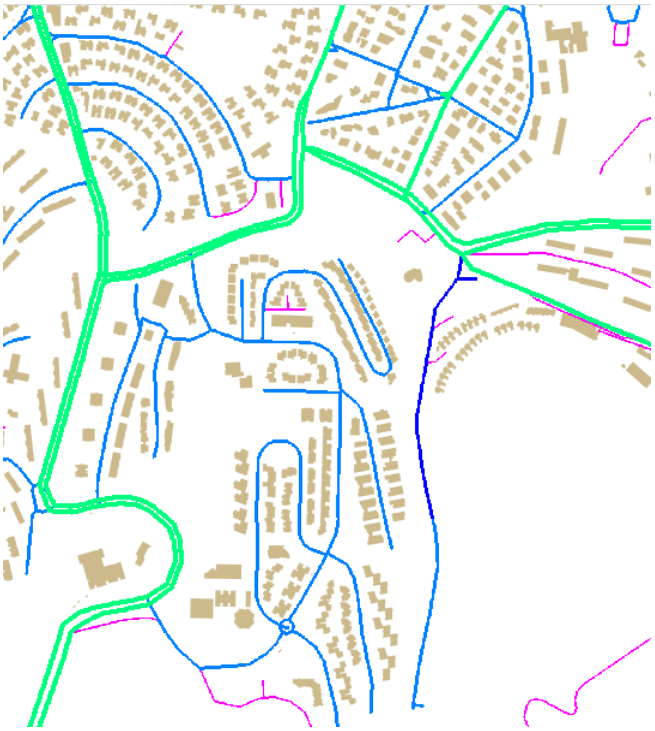

Figure 19. Map of roads with individual widths, randomly colored by road type.

REFERENCES

[1] A. Noskov and Y. Doytsher, "A Linear Approach for Spatial Data Integration," GEOProcessing 2016, Venice, Italy, 2016, pp. 93-99.

[2] R. Abdalla, "Geospatial Data Integration", Introduction to Geospatial Information and Communication Technology (GeoICT), Springer International Publishing, 2016, pp. 105-124.

[3] A. Arozarena, G. Villa, N. Valcárcel, and B. Pérez, "Integration of Remotely Sensed Data Into Geospatial Reference Information Databases. Un-Ggim National Approach," ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 721-5, 2016.

[4] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(4), 2002, pp. 509—522.

[5] J. Bennett, "OpenStreetMap - Be your own cartographer," ISBN: 978-1-84719-750-4, Packt Publishing, 2011.

[6] A. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro, "A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching," International Journal of Computer Vision, vol. 89(2-3), 2010, pp. 266-286.

[7] X. Chen, "Spatial relation between uncertain sets," International archives of Photogrammetry and remote sensing, vol. 31(B3), Vienna, 1996, pp. 105-110.

[8] S. Filin and Y. Doytsher, "The detection of corresponding objects in a linear-based map conflation," Surveying and land information systems, vol. 60(2), 2000, pp. 117-127.

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

[9] D. Foster and C. Mayfield, “Geospatial Resource Integration in Support of Homeland Defense and Security”, International Journal of Applied Geospatial Research (IJAGR), vol. 7(4):53-63, 2016.

[10] H. Kim and C. Chung, “Geo-spatial data integration for subsurface stratification of dam site with outlier analyses,” Environmental Earth Sciences, vol. 75(2), 2016

[11] A. Kipf and A. Kemper, “An Integration Platform for Temporal Geospatial Data. Digital Mobility Platforms and Ecosystems,” 2016.

[12] M. Landa, “GRASS GIS 7.0: Interoperability improvements,” GIS Ostrava, Jan. 2013, pp.21-23.

[13] T. Ma and J. Longin, "From partial shape matching through local deformation to robust global shape similarity for object detection," Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. IEEE, 2011, pp. 1441-1448.

[14] A. Noskov and Y. Doytsher, “A Segmentation-based Approach for Improving the Accuracy of Polygon Data,” GEOProcessing 2015, Portugal, 2015, pp. 69-74.

[15] A. Noskov and Y. Doytsher, “Triangulation and Segmentation-based Approach for Improving the Accuracy of Polygon Data,” International Journal on Advances in Software, vol. 9 (1-2), 2016.

[16] C. Parent and S. Spaccapietra, “Database integration: the key to data interoperability,” Advances in Object-Oriented Data Modeling, M. P. Papazoglou, S. Spaccapietra, Z. Tari (Eds.), The MIT Press, 2000.

[17] K. Piętak, J. Dajda, M. Wysokiński, M. Idzik, and L. Leśniak, “Geospatial Data Integration for Criminal Analysis”, Man–Machine Interactions 4, Springer International Publishing, 2016, pp. 461-471.

[18] E. Rahm and P. Bernstein, “A survey of approaches to automatic schema matching,” The International Journal on Very Large Data Bases (VLDB), vol. 10(4), 2001, pp. 334–350.

[19] A. Saalfeld, “Conflation-automated map compilation,” International Journal of Geographical Information Science (IJGIS), vol. 2 (3), 1988, pp. 217–228.

[20] B. Schmitzer and C. Schnorr, "Object segmentation by shape matching with Wasserstein modes," Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer Berlin Heidelberg, 2013.

[21] M. Shapiro and J. Westervelt, “r. mapcalc: An algebra for GIS and image processing”, Construction Engineering Research Lab (ARMY), Champaign IL; 1994.

[22] X. Shu and X. Wu, “A novel contour descriptor for 2D shape matching and its application to image retrieval,” Image and vision Computing, vol. 29.4, 2011, pp. 286-294.

[23] P. Shvaiko and J. Euzenat, “A survey of schema-based matching approaches,” Journal on Data Semantics IV, Springer Berlin Heidelberg, 2005, pp. 146-171.

[24] N. Tsendbazar, S. Bruin, S. Fritz, and M. Herold, “Spatial Accuracy Assessment and Integration of Global Land Cover Datasets”, Remote Sensing, vol. 7(12):15804-21, 2015.

[25] G. Wiederhold, “Mediation to deal with heterogeneous data sources,” Interoperating Geographic Information System, 1999, pp. 1–16.

[26] R. Xie, Y. Liu, X. Li, L. Yu, “A Framework of Satellite Observation Data Integration System,” International Conference on Mechatronics, Electronic, Industrial and Control Engineering (MEIC 2015), 2015.

[27] S. Zheng and J. Zheng, "Assessing the completeness and positional accuracy of OpenStreetMap in China," Thematic Cartography for the Society, Springer International Publishing, 2014, pp. 171-189

[28] http://wiki.openstreetmap.org/wiki/Zoom_levels [accessed 15.06.2017]

[29] https://github.com/gravitystorm/openstreetmap-carto/blob/master/roads.mss [accessed 15.06.2017]

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org