A Linear Approach for Spatial Data Integration

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. The suggested method consists of two phases: searching for counterpart boundaries or polylines by triangulation, and rectifying objects without correspondent polylines by a transformation and a shortest path algorithm. At the first phase, middle points of polygon boundaries are used to implement the triangulation. In order to define correspondent boundaries, the polylines of the two datasets which are connected by triangles are compared based on the lengths of lines and the distances between the nodes. At the second phase, vertices of the polylines without counterparts are shifted with respect to the lengths of the shortest distances to the nodes of the polylines with counterpart. The method is effective for pairs of datasets with different degrees of accuracy. Less accurate datasets use precise elements of other datasets for integration and improvement of their accuracy. The resulting data are well integrated with a more accurate map. A review implemented by specialists enables us to say that the results are satisfactory.**

Keywords-Geometry fusion; triangulation; shortest path; topology.

I. INTRODUCTION

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) [2]. 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, editing of objects will be reflected on all maps, which contain them.

The benefits of data integration are demonstrated in this paper by using the 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 total station, 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 space images. 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.

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: the 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 results are discussed in Section VII. The conclusion is presented in Section VIII.

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 [10] and Ontology-based types of attribute matching could be selected. In [13], 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 [11] are based on data fusion algorithms; the aim of the process is to prepare a map, which is a combination of two or more [6]. The merging and fusion of heterogeneous databases has been extensively studied, both spatially [9] and non-spatially [14].

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 [1]. In [12], 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. [5]. A special shape descriptor for defined correspondent objects on raster images was developed by Ma and Longin [17]. Focusing on single shapes does not allow us to apply these algorithms in our task.

In [4], 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. [3]; the applicability of diffusion distances within the Gromov-Hausdorff framework and the presence of topological changes have been explored in this paper.

We have concluded that the mentioned approaches could not be applied to resolve the considered problem. That leads from the fact that the mentioned approaches have been developed for specific conditions. For instance, the 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. 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 mainly presented 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.

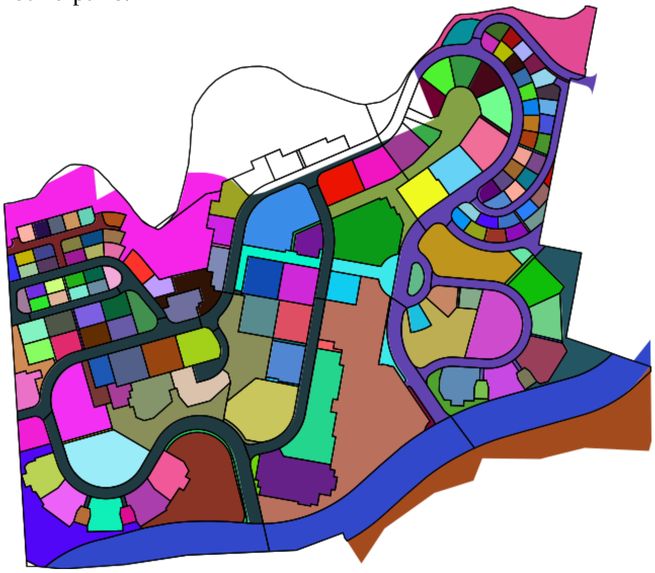

Figure 1. Source data: land-use city planning (colored background) and cadastre (black outline) maps.

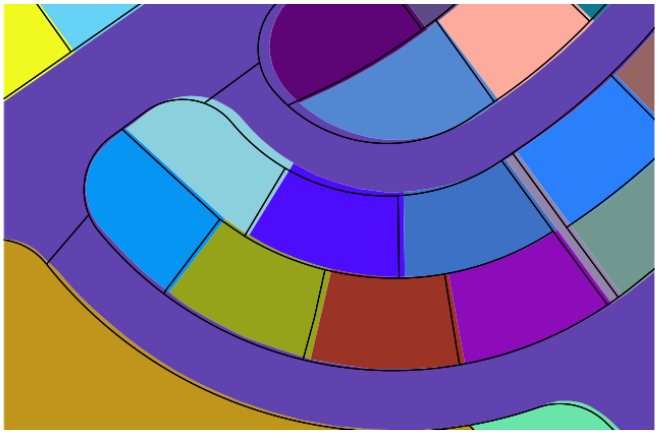

Figure 2. Positional discrepancies of city planning (colored areas) and cadastre (black lines) datasets.

In the previous approach [8], we defined correspondences between polygons. We encountered two 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.

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 [16]. 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. Centroids link polygon to certain raw in attribute table by a category number. Each raw in the attribute table starts with a "cat" field, which could be connected to a centroid with a given "cat" value.

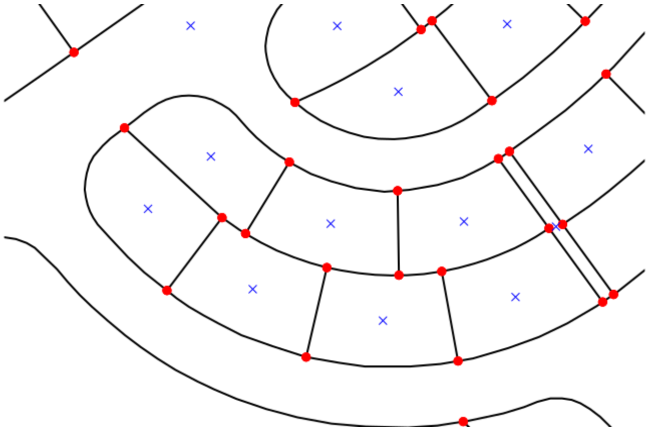

Figure 3. A sample of the city planning dataset residing in GRASS GIS's topologycal format. Nodes – red circles, centroids – blue croses, and boundaries – black lines.

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. Maximal distance parameter could fulfill this role.

In addition, it is quite popular to use buffers for detecting this fact. For instance, in [15] 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.

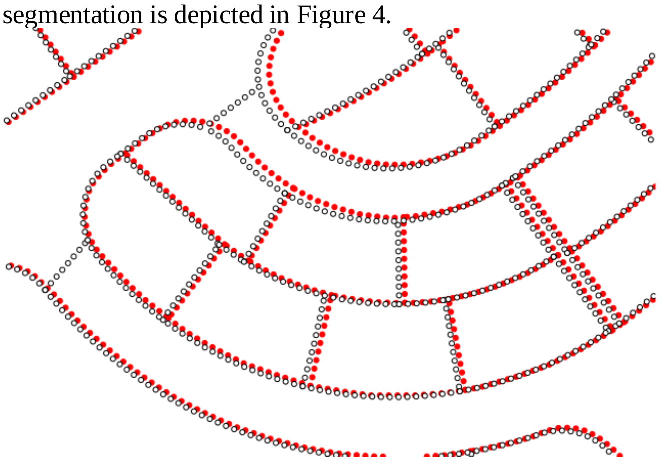

Figure 4. Point delimiter of equidistant segments. City planning – red, cadastre - black points.

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 the list of 100 percentiles of the calculated distances. D_{max} equals percentile i if the standard deviation of distances between percentiles i-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.

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 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.

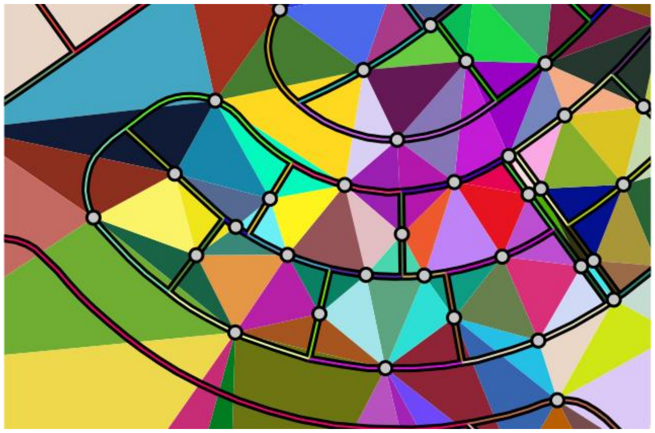

Figure 5. The triangulation of boundaries' middle points of a city planning dataset.

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 a combination 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.

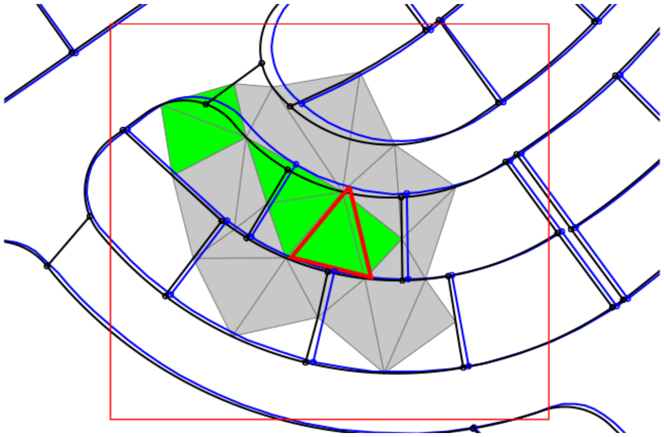

Figure 6. Filtering possible triple pairs.

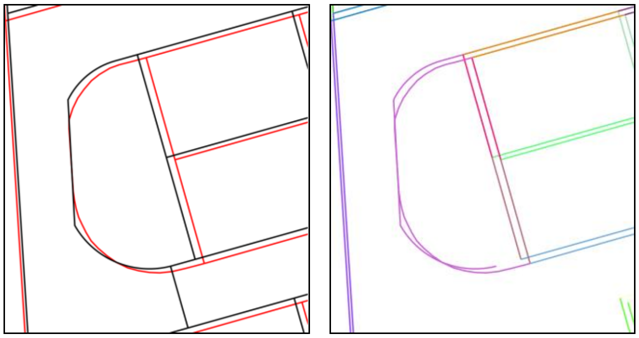

Figure 7. An example of an incorrectly found line pair. Left – original boundaries of the city planning (red lines) and cadastre (black lines) datasets. Right – detected linepairs.

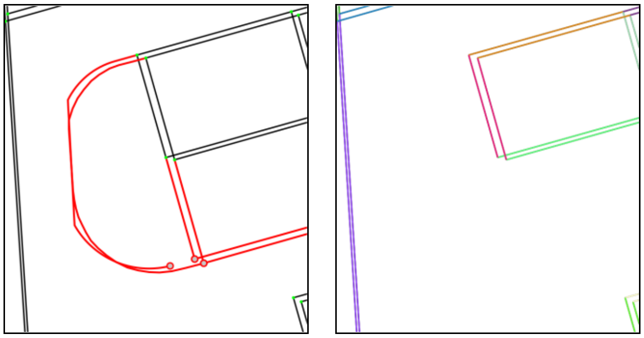

Figure 8. Detecting incorrect pairs. Left – incorrect nodes and line pairs are marked in red. Rigth – final line pairs.

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.

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.

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 (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.

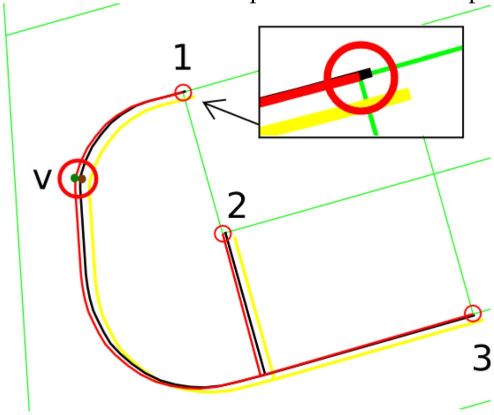

Figure 9. A vertex moved with respect to the shortest paths to bridge nodes.

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 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.

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.

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

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.

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. Results

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 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.

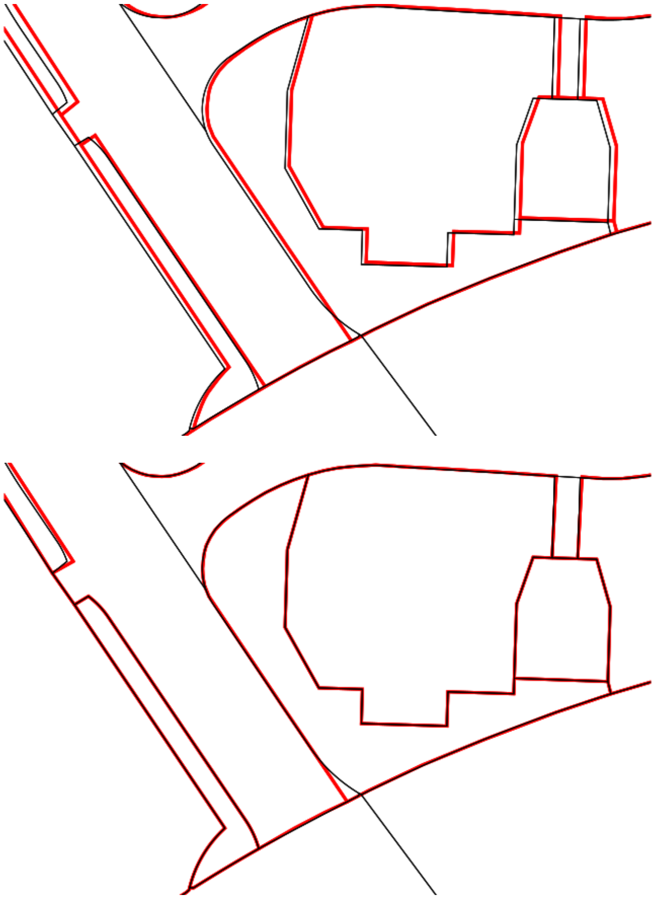

Figure 10. Zoomed-in extent 1. Bountaries of original (upper) and result (lower) datasets: city planning – red, cadastre - black.

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.

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, 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.

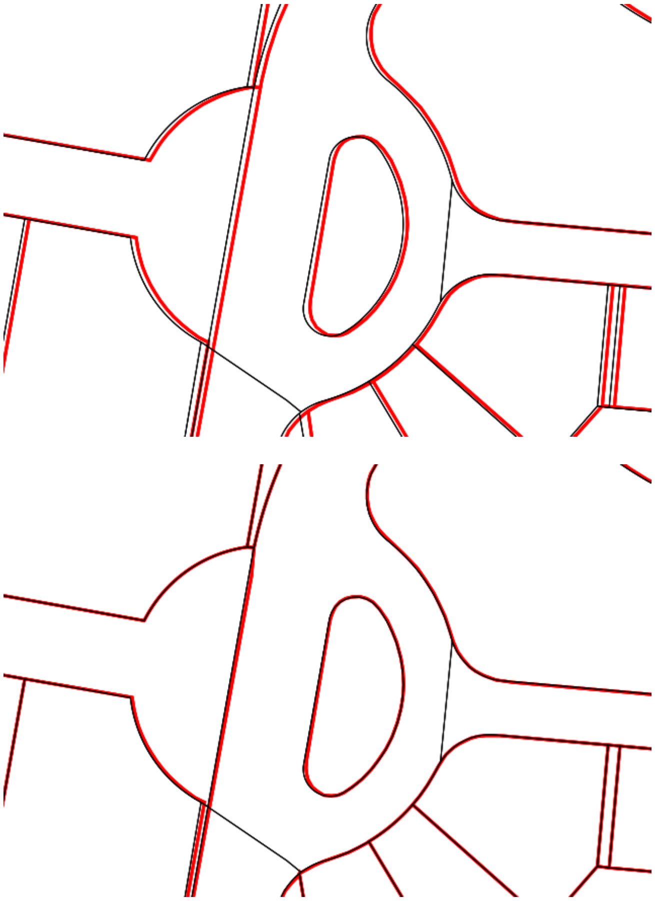

Figure 11. Zoomed-in extent 2. Bountaries of original (upper) and result (lower) datasets: city planning – red, cadastre - black.

According to the table, the average distance has been reduced by five times; 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.

To implement the approach, we used Python 2.7 programming language, GRASS GIS 7.1, and Debian GNU/Linux 8 operating system.

VIII. 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.

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.

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 [7]. This will allow us to apply the method to other types of datasets. For instance, OSM datasets are usually complete, updated, and relatively non-accurate. In order to produce updated and precise layers, it could be useful to integrate OSM data with an accurate dataset.

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.

References

[1] 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.

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

[3] 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.

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

[5] 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.

[6] 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.

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

[8] 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, accepted, in progress.

[9] 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.

[10] 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.

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

[12] 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.

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

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

[15] 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

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

[17] 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.