A LINEAR CONFLATION APPROACH FOR THE INTEGRATION OF PHOTOGRAMMETRIC INFORMATION AND GIS DATA

Sagi FILIN, Yerahmiel DOYTSHER
Department of Civil Engineering (Geodesy)
Technion - Israel Institute of Technology, Haifa 32000, Israel
filin.1@osu.edu , doytsher@geodesy.technion.ac.il

Working Group IV/3

KEY WORDS: GIS, Map Conflation, Map Revision

ABSTRACT

Knowledge provided by GIS data can alleviate many problems associated with object recognition from aerial imagery. However, as the scale of the GIS increases (for example a GIS database of topographic maps) positional disagreements between the data sets hamper the efficient utilization of this data. Presented here is a novel approach to the registration of GIS and photogrammetric data, based on local transformations according to linear features existing in both data sets. The results of applying this algorithm show great improvement in GIS accuracy and thus enable better utilization of common feature extraction techniques. The algorithm, being general, can be applied in its entirety or in part to different tasks concerned with integration and evaluation of several vector data sets.

1 INTRODUCTION

Autonomous extraction of geographic information from digital imagery is one of the greatest challenges of modern photogrammetry. However, this is a difficult task due to the complexity that characterizes natural scenes. The underlying problems that make it so complex relate to the amount of data, its variety both in terms of objects and object types, and the complexity of modeling and representing spatial relations between objects. Furthermore, current feature extraction algorithms are still far from being able to provide clean features for use by object recognition algorithms. Consequently, object recognition systems tend to focus on very limited tasks (road extraction, as a typical example, see Baumgartner et al. 1997) in very limited environments (such as high altitude imagery or rural scenes) and even so have a very limited rate of success (Heipke et al. 1997). An alternative approach to overcoming these limitations utilizes existing GIS databases as part of its prior knowledge, thus reducing the uncertainty in the whole procedure. Integrating existing GIS data into the object recognition scheme is reasonable. Conceptually, it represents the existing prior knowledge about the data, thus reducing the uncertainty in annotating a given scene. Practically, it is a useless to detect road objects when the road's outline is already given in another database. Moreover, utilizing existing scene descriptions may help focus on more specific tasks such as map revision and improving the accuracy of the GIS, rather than concentrating on problems posed by the limitations of the feature extraction algorithms.

An important issue that is usually overlooked in this regard is the inherent distortions between the GIS data and the imagery data. This phenomenon might not be as common in data arriving from large-scale mapping (cadastre for example) but it is widespread in medium to small scale mapping (topographic mapping for example), the data which is our concern. Here, cases of objects shifted tens to hundreds of meters away from their expected positions and distorted in respect to their original shape, are common, thus forming a large search space in the image domain for detection of objects. A common solution that is employed is to improve the registration between data sets. This is accomplished by either a global warping transformation or local, rubber-sheeting transformations, that achieve better accuracy. A rubber-sheeting transformation of the GIS features is usually based on using counterpart seed points between both data sets, mostly intersections of road networks, and performing local transformations according to the disagreement between them. Subsequently, after detecting counterpart seed points, they are triangulated to subdivide the plane into local regions, and then Piecewise Linear Homeomorphic (PLH) transformations are applied inside each triangle to transform the objects within. This widely used approach is very limited in truly coping with distortions between sets, since only the seed points are matched and the relative disagreements between the linear features shapes remain unresolved. Alternatively, we may utilize linear features as seed entities for registering the GIS data. This is more adequate as GIS data is linear in nature (road network, streams, etc.), so that relations between data sets are better captured by linear entities. Indeed, results of point-based map conflation show that while seed points are matched, the polylines connecting them are usually not. A line-based transformation employs the following steps: first counterpart line features are detected, then the whole region is partitioned into sub-regions according to the network of counterpart lines. Next, counterpart elements are transformed to their new positions, and all remaining elements within each sub-region are transformed according to the boundary transformation.

The process is composed of three basic stages, the detection of counterpart elements, the establishment of correspondence between the matched entities and transformation of the data set. The first is important for automation of the process. Consider for example a typical topographic map. The number of linear features (or even road intersection) to be matched with counterpart entities is large enough to be impractical for a manual identification process. The matching itself establishes the geometric relations between the two data sets by modeling the distortions and the transformation is the core of the whole process and involves the actual transformation of the data set. Each of these aspects is non-trivial in itself.

In this paper we are presenting a novel algorithm that was developed for this task with examples demonstrating its use. The counterpart linear entities that were utilized as the core for this transformation are part of a road network that was previously extracted from photogrammetry. By applying the algorithm we have managed to significantly reduce the disagreement between the GIS data and the rectified images, which results in a dramatic narrowing of the search space for corresponding objects for further applications. The whole algorithm or parts of it can be applied in other GIS applications such as data fusion, map generalization, change detection and other algorithms involving integration of data sets. Let us first present the algorithm.

2 THE LINE-BASED MAP CONFLATION ALGORITHM

The overall algorithm is composed of four parts. First the counterpart elements are detected, then matching is established by the counterpart objects. The third part of the algorithm is concerned with subdivision of the plane into closed parts in which the local transformation takes place. The final part involves applying the transformations to correct the distortions. We first address the detection of counterpart objects.

2.1 Detection of counterpart elements

Counterpart linear features are expected to be polylines lying a relatively short distance from one another and expected to have similar characteristics. The characteristics of interest are shape similarity, cumulative distance and similarity between emanating nodes at both end points, the first two attributes being geometric in nature while the third one is topological. The main hurdle with using these criteria is that the correspondence between linear features is not always 1:1, meaning that one polyline can be represented by a set of polylines in the counterpart set. In such a case the above characteristics are not much use unless they are incorporated into a more general algorithm. Figure 1 depicts two typical scenarios: in the first (Figure 1.a), the correspondence is between one polyline to three polylines thus forming a 1:N (one to many) relationship, the second scenario (Figure 1.b) presents correspondence between three and four polylines thus forming a N:M (many to many) relationship. In practice the relations can be between more segments in both sets and multiple candidate paths are not an unlikely possibility.

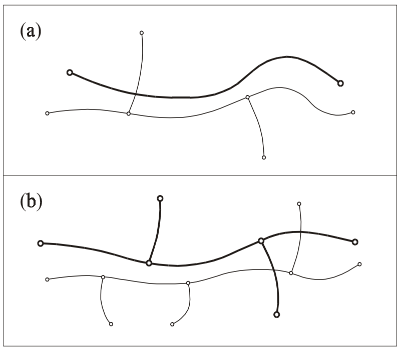

Figure 1. Possible correspondence between linear features

One possible approach to handling this is to extend the relations between one polyline in both data sets into multiple polylines (N:M relations) or sub-polylines (see Walter and Fritsch 1999, and Gabay and Doytsher 1995), however this complicates the modeling. An alternative solution that avoids handling such or even more complex cases is to reduce the problem to the detection of counterpart nodes (i.e. intersection of polylines), which are easier to manipulate and can be further extended to detect counterpart polylines. Working with nodes rather than edges is advantageous as it reduces the dimensionality of the objects concerned from 1D to 0D, hence reducing the number of attributes as well as their complexity. Consequently, problems as that above are suppressed and detection of counterpart candidates becomes easier (there are pathological cases were a node in one data set is a degenerate representation of a broken intersection, but these cases are easy to track down and deal with). The algorithm for detecting counterpart nodes works as follows: first candidate nodes are detected according to a proximity criterion, then candidates are evaluated according to their attributes - the number of emanating nodes and their orientations. Due to differences in the sources from which the data were compiled, the acceptance criteria for candidate counterparts are relaxed so that some disagreement is permitted. Candidate counterparts are then validated (and either accepted or eliminated) by introducing a topological validation procedure we call a "round trip walk". This procedure enables verifying the correctness of a candidate, disqualifying a candidate and also proposing a candidate based on topological criteria. The final stage in the algorithm is constructing the paths between nodes. Paths are constructed using Best First Search (BFS), a graph search algorithm that is useful for finding short paths. Candidate paths are then validated and fixed according to a geometric criterion that evaluates the average distance between the two paths (after normalizing them). For paths failing to fulfill this criterion the algorithm searches, where needed, for better candidate edges that minimize the distance. In complicated cases where the correspondence is very loose, the procedure may still associate some wrong candidates; therefore some human intervention may be required. An assessment procedure was developed as part of the algorithm, for evaluating the level of uncertainty in matching between counterparts. This is a function of the agreement between the original sources (due to the reasons outlined above) rather than the performance of the algorithm. Counterparts with high level of uncertainty are recommended to be checked by a user and be corrected if needed. Applying an editing mechanism that also re-computes the matching of related nodes to the one that was corrected, has shown that once one or a few nodes are corrected many other wrong node associations are automatically corrected as well. Therefore human intervention, even in difficult cases, is very limited. The result of the overall procedure is a set of counterpart paths that form a set of counterpart edges in a graph structure, when grouped together.

2.2 Matching of counterpart objects

Creating the graph of counterpart features is followed by computing a transformation between counterpart edges. In the point-based case this part is trivial as the translation of a node towards its counterpart is the translation (*dx, dy*) from the original to the new position. With lines, this becomes more complex. The main hurdle is the discrete representation of polylines. If polylines could have been easily described analytically, a transformation could have been given in an analytical form as well. However, when polylines are described by a set of vertices, analytical transformation is less likely to exist. This in turn requires a more precise definition of the transformation characteristics. Here we define them as the complete matching of the counterpart polylines, so that the transformed polyline will coalesce with its counterpart (this was not accomplished in the point-based matching), and maintaining the original shape of the polyline so that the original shape will not be distorted by the transformation. The second property relates to the view that the polyline transformation should be the basis for the transformation of all other elements within the bounded region. Maintaining the polyline shape prevents distortions when related elements are transformed.

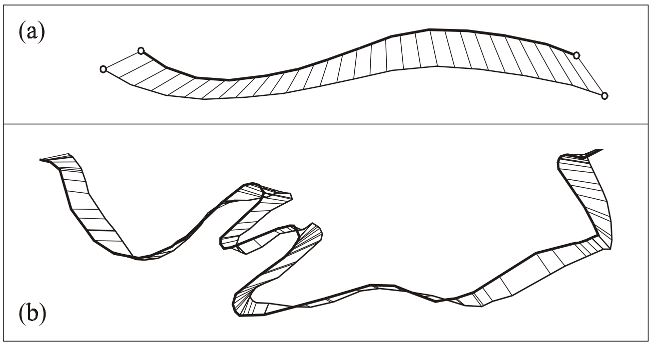

Figure 2. Projection by using normalized cumulative distances

The transformation is based on polyline projection, or in other words, mapping between polylines. Since polylines are described by vertices, the projection is seen as a translation of the vertices ($L_1(x_i, y_i) \mapsto L_2(x_i^{'}, y_i^{'})$) from their original position to their expected position on the counterpart polyline. Several experiments with different projections have shown that the best projection is achieved by using the normalized cumulative distance as the measure for the expected position of a given vertex. Figure 2 illustrates this concept. Having both polylines coincide, entails projecting vertices from both polylines towards each other (Filin and Doytsher 1999) - $L_1(x_i, y_i) \mapsto L_2(x_i^{'}, y_i^{'})$ *and* $L_2(x_j, y_j) \mapsto L_1(x_j^{'}, y_j^{'})$ - so the transformation becomes direction independent.

The algorithm was further improved to cope with some distortions in the projection that could occur even though the shapes are similar. The distortions have mostly to do with scaling - some similar segments along the polyline may be a bit shorter or a bit longer then their counterparts, thus shifting all projected vertices and causing distortions (see Figure 3.a). Such

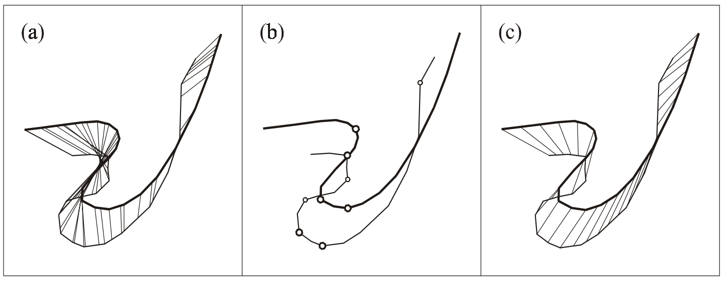

Figure 3. Projection of polylines with/without "breakpoints"

distortions are removed if the projections are performed between similar segments in the polylines instead of the whole polylines. As breakpoints are the features that partition the polylines into segments it is necessary to first identify them on both polylines (as depicted in Figure 3.b), match them, and then apply the projections on the segments between them, Figure 3.c illustrates the modified projection algorithm.

As can be seen, the modified procedure performs better in capturing the idea of polyline projection, as it at first associates counterpart breakpoints and only then associates the intermediate segments and the vertices along them. With the identification of counterpart elements between both sets and with establishing a transformation between them, the problem of merging common features is solved. Now it is only left to transform the related features to the unified database, thus solving the problem of maintaining topological relations and if one data sets is more accurate than another, then transforming the related objects to their new location can be expected to increase their accuracy.

2.3 Planar subdivision of the data

As local transformations that also coalesce counterpart features are in concern (according to the criteria introduced above), a natural approach would be to subdivide the data set according to the counterpart features. Having a network structure of the linear features makes this choice even more natural by providing a natural subdivision of the data into adjacent regions according to the minimal set of polygons formed by all the edges. Indeed, it is not mandatory for linear features to form a network structure, in such a case, the plane can be subdivided into regions using common algorithms, (for example, De Berg et al. 1997). Subdividing the space into regions extends the topological structure of the data from a network topology into cell topology by introducing "faces"- 2D objects defined by the edges (1D objects) enclosing them. The cell structure can based on identifying faces by their enclosing edges. This requires forming a convention regarding the position of a face in respect to the defining edge, for example, the right-hand side of an edge defines the inner part of a face. Since each edge participates in defining two faces - one on the left and one on the right, they are duplicated (virtually) so that each undirected edge turns into two directed edges, and each participates in defining only one face. The face construction algorithm then scans the edge list, and finds, for the current unchecked directed edge in the list, the next directed edge that forms the maximum clockwise angle in respect to the current directed edge. This process is repeated until the original directed edge is reached again. The criterion for terminating the algorithm is when no edges are left in the edge list. The process is comprehensive, but generates some undesirable phenomena such as connecting separate regions or including segments that contribute nothing to the overall face definition. These phenomena are mostly concerned with free edges (for example access roads leaving a main roads) or linking edges (such as roads connecting a major road network to a closed network of roads in a residential area). Figure 4 demonstrates such cases, and following the algorithm we can see that the inner structure will be part of the left face even though it is inside the right face. To eliminate such situations the algorithm is refined to identify and eliminate free edges and linking edges. The first situation is easier to perform as free edges are characterized by having at least one free end node. These edges are separated from the edge list for the cell topology. This process should be iterative as the removal of one edge might turn another one free. Links are more difficult to detect as their end points are connected to at least two edges. To track them we use the fact that in the definition of cell boundaries they will appear twice in the definition of the same face.

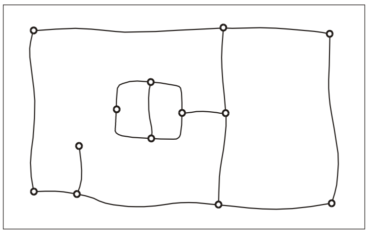
Figure 4. Free and linking edges while defining closed regions

Elimination of link edges can create inner faces inside a face (holes). Holes appear twice in the face list, one defining the outer boundary and the other the inner boundary. Viewing Figure 5 and keeping in mind the idea of local transformations shows that holes play an important role in the rubber-sheeting transformation. Naturally we would like the rubber-sheeting transformation of objects inside region P_1 to be performed according to its boundary but not in respect to the external boundary of polygon P_3. By the same token we would like the objects in P_3 to be transformed according to the external face

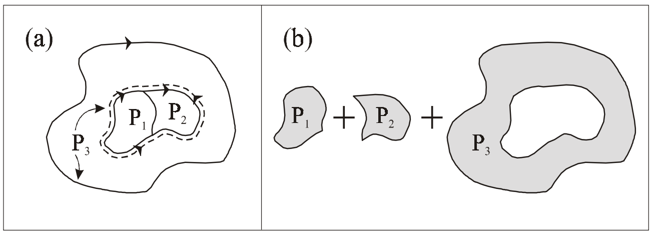

Figure 5. Inner faces (holes) inside a face

but also in respect to the internal hole. Thus the modified planar subdivision procedure creates a list of three object types, external faces (defining the inner part of a face) external faces (holes) and a separated list of counterpart free or linked edges. Next, the last two types are associated with the first. Since holes or edges cannot belong to more then one face it is sufficient to determine whether or not one of their vertices is inside a face (by applying any point in polygon algorithm). Note that although holes can appear inside another hole, they should be associated only with the inner most one. Having the overall plane subdivided (equivalent to triangulating the plane in the point-base conflation) enables performing the local rubber-sheeting transformations.

2.4 The Rubber-Sheeting Transformation

The transformation of objects inside each face is based on the transformations between the vertices of the bounding polylines, just as it was in the matching of counterpart polylines. However, not all vertices participate in the transformation of each object. This happens when some of the reference vertices are occluded by other segments of the polygon or by other reference objects. Occlusions between or within reference elements can block the influence of regions on the transformation of a given point (Doytsher and Gelbman, 1995). To demonstrate this, consider for example a point in Figure 6, located under one of the two internal structures. As the upper part of the bounding polygon is hidden, it is reasonable to assume that this part will not affect the translation of that point. The influence of the upper part will be due to the internal structure. Self-occluded areas are also possible, for example with non-convex bounding polygons and certainly with holes, for which one of their sides is always hidden. Regions of influence define the reference vertices affecting the transformation of a given point, therefore their actual effect is in defining which vertices participate in the transformation and which do not.

As the influence regions are a function of both self-occlusions and occlusions between reference objects, they should be computed separately and then superimposed to form a composite influence region. The suggested algorithm for their computation has a linear running time, and is based on scanning the polygon vertices in a clockwise order (the outcome of the face construction procedure) starting from a reference vertex that is visible from the object point. The visibility or invisibility of the consecutive points along the polygon is based on a relative topological analysis of their locations in regard to the object point. The algorithm also holds for the computation of the influence regions for holes, while in this case the order of the vertices should be counterclockwise (i.e., the original order of vertices for holes). For computing influence regions for polylines the same algorithm holds as well, simply duplicating the polylines in an inverse order and treating them as closed holes. The superposition of influence regions checks the relations between objects in terms of occluding parts. This is best captured by polar coordinate notation, i.e. distance from the given object point and angular coordinates, so that for overlapping objects, the one nearest to the point will be the influential one. Figure 6 illustrates the output of the algorithm for a non-convex polygonal boundary and an internal hole. As can be seen, parts of the outer polygon do not influence the point due to self-occlusion, in addition, another occluded part is a result of the superposition with the internal hole. For this region, the influential displacements are due to the visible part of the hole. This combination would have most likely been a natural human interpretation of the influencing components. The computational complexity of the algorithm (assuming regular cases) is of linear order (O(n)) for the computation of the self-occluding parts (which is the best that can be achieved) and linear as well (O(m)), for the mutual occlusions, where 'm' is the number of elements ('m' is usually small).

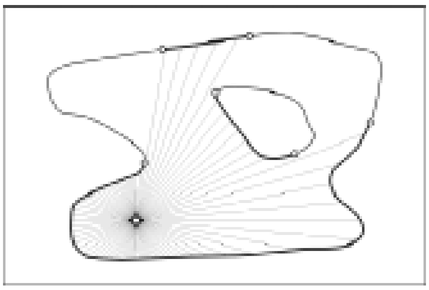

Figure 6. Influence regions

The transformation of a given point inside a boundary is performed by computing the displacements of the object vertices according to the influencing reference vertices (computed previously). The transformation that was used for this purpose is based on the weighted-average of displacements of all related points. A measure for the weight that was found to be adequate is the reciprocal of the square distance between the point in question and a given reference vertex.

3 EXPERIMENTAL RESULTS

The algorithm was applied to several data sets each composed of a vector compilation of a 1:50,000 scale map and from data (road network) digitized from respective orthophoto maps. The difference between the two sources was great in almost every aspect. First the amount of detail in the 1:50,000 maps was far greater than in the other set, then the period of time between the compilation of the both sets generated major changes in content. The average positional difference between the data sets varied between 70m and more then one hundred meters (depending on the data set that was used), the maximal distance we encountered in several data sets was in the order of 300m. Figure 7 illustrates some of the aspects mentioned above.

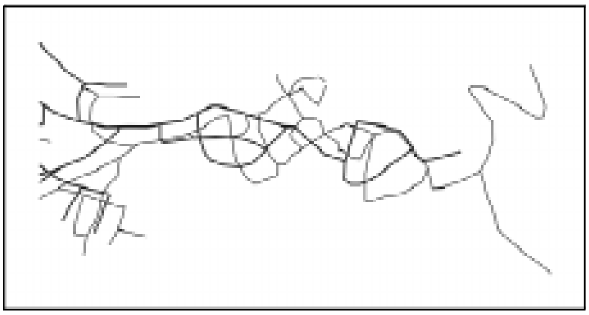

Figure 7. An example for differences between data sets

The difference has shifted our concern in evaluating the algorithm to two aspects, the improvement in accuracy achieved by applying the algorithm and the level of automation achieved in the counterpart detection algorithm. The second concern had mostly to do with assessing the fitness of the counterpart detection algorithm and the level of success in automation of this process. Let us first address the matching procedure. The number of nodes (the matching was based on detection of counterpart nodes between the data sets) received from the GIS data was usually in the upper thousands while the order of nodes from the photogrammetric based GIS was usually half of that. The amount of counterparts that were detected was usually 50% or more of the number of nodes in the photogrammetric data. The percentage of correct counterparts detected by the algorithm was higher than 90% (the lowest percentage was 92.5% and the highest 96.4%). However our major concern was alerting about the presence of suspicious counterparts. An evaluation algorithm we developed for reliability of counterparts enabled visiting and correcting suspicious counterparts. The amount of nodes that were needed to be visited was mostly on the order of 10% (of which ~ 40% were wrongly classified). These nodes tended to cluster around areas with poor correspondence and usually by correcting few, many others were corrected as well as a follow-up by the application.

To evaluate the improvement in positional accuracy we evaluated the improvement in the accuracy of roads that did not take part in the transformation (mostly secondary roads). These secondary roads were digitized from the orthophoto maps and served as a “comparison” sample set. The coordinates of these features (as digitized from the 1:50,000 topographic maps) were corrected according to the presented algorithm, and the corrected coordinates were checked against the "comparison” sample set. It was found the improvement in accuracy that was on the average on the order of 75%, so that the average distance between the objects original location and their true location was reduced, for example, from 44m to 9m on average, 113m to 30m and so on, whereas the MSE values were reduced similarly from 49m to 11m, 114m to 34m and so on. Evaluation of other GIS objects such as position of creeks and others objects, have shown similar results.

4 SUMMARY

Utilizing GIS data to improve object recognition tasks has not received the attention it deserves, most likely due to its not being part of the vision paradigm. However, since it exists and since the main task of object recognition from aerial imagery is the creating and updating of GIS databases, it is only reasonable to use the data already in existence. In this paper, we have addressed a preliminary task that is usually overlooked, concerning registration of one data set to another. Indeed this task was preformed in the past by different methods, for example using global transformations or using point based local transformations. However, issues such as having counterpart features coalesce or improvement in accuracy of related objects were not of much concern. The current algorithm addresses both these issues and performs this task in a more natural way in respect to the data in question.

The algorithm described here addresses the complete process beginning with the detection of counterpart points, continuing with the matching of counterpart lines and concluding with the planar subdivision and the transformation of related objects. The results indicate a vast improvement in accuracy and indicate that the subsequent tasks will become easier to perform. As mentioned, the algorithm is general and can be applied for many other tasks, among which we may count change detection between vector data sets, integration in general of two or more vector data sets and so forth.

REFERENCES

Baumgartner A., Steger C., Mayer H., Eckstein W., 1997. Semantic Objects and Context for Finding Roads. Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision III, Editors: D.M. McKeown, Jr., J.C. McGlone, O. Jamet, Proceedings of SPIE, Orlando, Florida (USA), Vol. 3072, S. 98-109

De Berg M., Kraveld M., Overmars M., Schwartzkopf O., 1997. Computational Geometry, Springer.

Doytsher Y., Gelbman E., 1995. A Rubber Sheeting Algorithm for Cadastral Maps. Journal of Surveying Engineering – ASCE, 121(4): 155-162.

Filin S., Doytsher Y., 1999, Linear Approach to Map Conflation: Matching of Polylines. Surveying and Land Information Systems, Vol. 59(2), pp. 107-114.

Gabay Y., Doytsher Y., 1995. Automatic Feature Correction in Merging of Line Maps. Proceedings of the 1995 ACSM-ASPRS Annual Convention, Charlotte, North Carolina (USA), 2:404-410.

Heipke C., Mayer H., Wiedemann C., Jamet O., 1997. Evaluation of Automatic Road Extraction. International Archives of Photogrammetry and Remote Sensing, International Society for Photogrammetry and Remote Sensing, Vol. 32 (3-2W3).

Walter V., Fritsch D., 1999. Matching Spatial Data Sets: a Statistical Approach. International Journal of Geographical Information Science, 13(5), pp. 445-473.