ISPRS Journal of Photogrammetry and Remote Sensing 78 (2013) 69–84

Contents lists available at SciVerse ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Line segment confidence region-based string matching method for map conflation

Yong Huh [a], Sungchul Yang [b], Chillo Ga [c], Kiyun Yu [c], Wenzhong Shi [a,*]

[a] Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Kowloon, Hong Kong
[b] Spatial Information Research Institute, Korea Cadastral Survey Cooperation, Yeongdeungpo-gu, Seoul, Republic of Korea
[c] Department of Civil & Environmental Engineering, Seoul National University, Gwanak-gu, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 10 July 2012
Received in revised form 23 December 2012
Accepted 20 January 2013
Available online 28 February 2013

Keywords:
Map conflation
Spatial uncertainty
Confidence region of a line segment
String matching
Corresponding point pair

ABSTRACT

In this paper, a method to detect corresponding point pairs between polygon object pairs with a string matching method based on a confidence region model of a line segment is proposed. The optimal point edit sequence to convert the contour of a target object into that of a reference object was found by the string matching method which minimizes its total error cost, and the corresponding point pairs were derived from the edit sequence. Because a significant amount of apparent positional discrepancies between corresponding objects are caused by spatial uncertainty and their confidence region models of line segments are therefore used in the above matching process, the proposed method obtained a high F-measure for finding matching pairs. We applied this method for built-up area polygon objects in a cadastral map and a topographical map. Regardless of their different mapping and representation rules and spatial uncertainties, the proposed method with a confidence level at 0.95 showed a matching result with an F-measure of 0.894.

1. Introduction

An important process in map conflation is object matching, which identifies corresponding object pairs with geometric similarities in terms of distance (Li and Goodchild, 2011; Min et al., 2007), shape (Bel Hadj Ali, 2001) or objects' neighborhood relationship (Kim et al., 2010; Samal et al., 2004) and consequently determines the optimal object pairs with the highest similarities among the candidate pairs (Li and Goodchild, 2011). However, the different mapping and representation rules of surveying agencies and spatial uncertainties in each geospatial dataset lead to inevitable positional discrepancies between corresponding objects. Because the above similarities are easily affected by the discrepancies, a map alignment that detects corresponding point pairs and then transforms a map to adjust and align corresponding objects well is necessary prior to the matching process (Yuan and Tao, 1999).

There have been many studies related to this topic. Beard and Chrisman (1988) proposed the Zipper algorithm to detect corresponding point pairs for edge matching between adjacent geospatial datasets. Gösseln and Sester (2003) detected corresponding point pairs with the iterative closest point (ICP) algorithm for two point sets derived from each contour of the corresponding objects. Masuyama (2006) proposed the integrated apparent difference detection method to determine corresponding point pairs between the boundaries of two tessellation datasets. Butenuth et al. (2007) combined distance and angle similarities to detect corresponding point pairs and aligned geospatial datasets with the dual interval alignment method. Seo and O'Hara (2009) detected corresponding line segment pairs and their corresponding point pairs using the raster-based matching method, which converts line segments into raster data using the segments' existence, length, distance, and orientation, and then the reconstructed vector geometries are used to find corresponding pairs. Huh et al. (2011) proposed a string matching method that searches the optimal point edit sequence to convert the contour of a target object into that of a reference object with a minimum total edit cost; the method uses three types of edit operations, deletion, insertion and substitution, for individual points on the contour of a target object. Given the optimal point edit sequence, corresponding point pairs are determined as pairs on which a substitution edit operation is chosen.

However, these methods can be further improved. Most of the methods detect corresponding point pairs based on a local search strategy that independently identifies pairs one after another (Li and Goodchild, 2011). Given a point in one geospatial dataset, several candidate points in another geospatial dataset are evaluated with a similarity criterion, and a single point with the highest similarity is chosen as the corresponding point. Therefore, when candidate point pairs are evaluated by a local search strategy, as Li and Goodchild (2011) noted, their compatibilities with neighboring

* Corresponding author.
E-mail address: lswzshi@polyu.edu.hk (W. Shi).

0924-2716/$ - see front matter © 2013 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS) Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.isprsjprs.2013.01.006

corresponding pairs are not considered as the contextual matching methods of Kim et al. (2010) or Samal et al. (2004). The iterative matching and evaluating process in the ICP algorithm of Butenuth et al. (2007) and Gösseln and Sester (2003) or the string matching method of Huh et al. (2011) could alleviate the aforementioned problem. However, the performance of the ICP algorithm is sensitive to initial matching pairs and can easily degenerate with the presence of outliers (Chui and Rangarajan, 2003). Additionally, the ICP algorithm needs a transformation model between corresponding points as known a priori. In the case of a map registration or a sensor pose estimation problem, a rigid or an affine transformation model can be assumed to explain auto-correlated positional discrepancies between corresponding points. On the other hand, the positional discrepancies in this study are mainly occurred by a random deformation process rather than a systematic transformation process. Thus a transformation model can lead to an erroneous matching result as shown in Fig. 1. In the figure, there are seven true corresponding point pairs (a_1,b_1), (a_2,b_2), (a_3,b_3), (a_4,b_4), (a_5,b_5), (a_6,b_6) and (a_9,b_8). Even though all the seven pairs are searched, the pairs in the upper left corner dominate a least square estimation process to determine transformation parameters. Thus, iterated correspondence search and map transformation would eventually result five corresponding point pairs in Fig. 1b with one false corresponding pair of (a_8,b_7) and two missed pairs of (a_6,b_6) and (a_9,b_8).

Thus a new matching method considering a deformation model for each line segments and points is necessary. To resolve the above problems, Huh et al. (2011) used a string matching method that used cost functions based on a physical deformation energy model. However, their cost functions could not be generalized and sometimes presented improper results because they were heuristically determined.

In this regard, we modify the string matching method of Huh et al. (2011) to further develop a general method by means of (1) a confidence region model of a line segment (Shi, 1998) and (2) calculating the cost for an edit operation based on the normalized discrepant area (Shi et al., 2003), which measures the change in the confidence region caused by an edit operation on a point. Because every geospatial datasets have their own spatial uncertainties (Gahegan and Ehlers, 2000), the corresponding line segments and their points have inevitable random positional discrepancies. Masuyama (2006) called this type of random discrepancy as apparent discrepancies and distinguished them from substantial discrepancies. Substantial discrepancies are caused by the mapping agencies' different data acquisition time or surveying rules with which the real world entities to be represented are chosen; thus, there is no correspondence between them. We assume that the apparent discrepancies would be adjusted by a substitution edit operation because it moves points of a target object to the positions of their corresponding points of a reference object. However, substantial discrepancies would be removed by deletion and insertion edit operations because the operations delete points of a target object or insert points of a reference object into a target object.

Consequently, we used the uncertainty model of line segments and their endpoints for three purposes. Firstly, the costs for the aforementioned three types of point edit operations are calculated with the confidence regions for each line segment and point. Secondly, criteria to generate a virtual corner point (Huh et al., 2011) are determined from the model. This point is an intersecting point of two straight lines extended from line segments linked to each side of a line segment. By inserting a virtual point, corresponding corners where one is described as a point and the other as a line segment, can have an appropriate corresponding point. Thirdly, a distance threshold for corresponding point pairs is obtained from the model. In the previous study of Huh et al. (2011), these were all independently and heuristically estimated based on an analysis of a training site. However, in this study, they were obtained by introducing a spatial uncertainty model based on the confidence regions of the line segments and points. Therefore, the proposed method could be more generally applicable for various geospatial datasets.

The remainder of the paper is structured as follows. In the next section, the structure of this study is described, and their details of the proposed method are presented. Then, the results of the method are evaluated and discussed in Section 3. Finally, the conclusion is given in Section 4.

2. The proposed method

The logic flow of the proposed line segment confidence region-based string matching method and its evaluation method are presented in Fig. 2. They include the following three steps: (a) data preparation, (b) point matching and (c) result evaluation. The data preparation step is comprised of two processes: calculation of the confidence regions of the line segments based on the spatial uncertainty model and the generation of virtual corner points for the corner areas. Occasionally, corresponding corner areas between two objects are represented as different feature types, such as a point feature in one object and a line feature in the other object. In this case, the virtual corner point obtained by the intersecting point of the two line segments linked to both sides of the corner

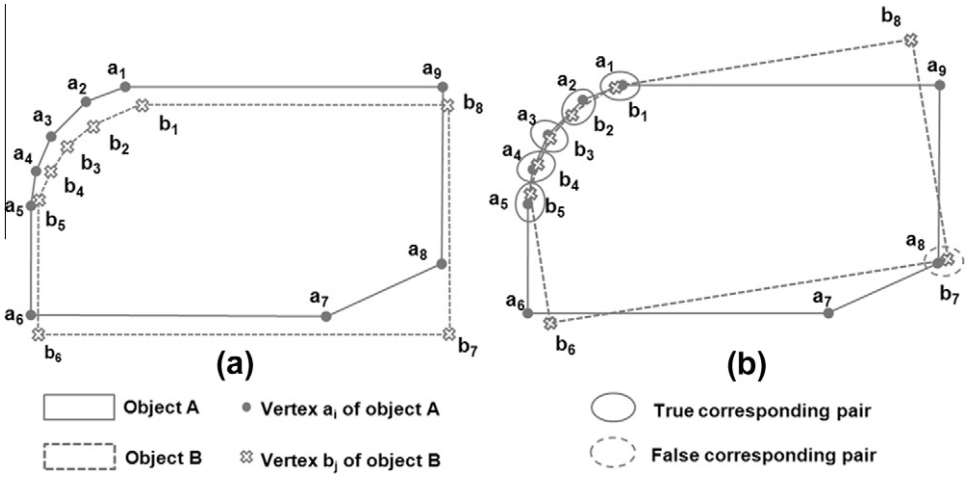


Fig. 1. The problem of ICP-based methods to find corresponding point pairs: (a) original objects *A* and *B*, (b) detected corresponding point pairs and transformation of object *B* with the pairs.

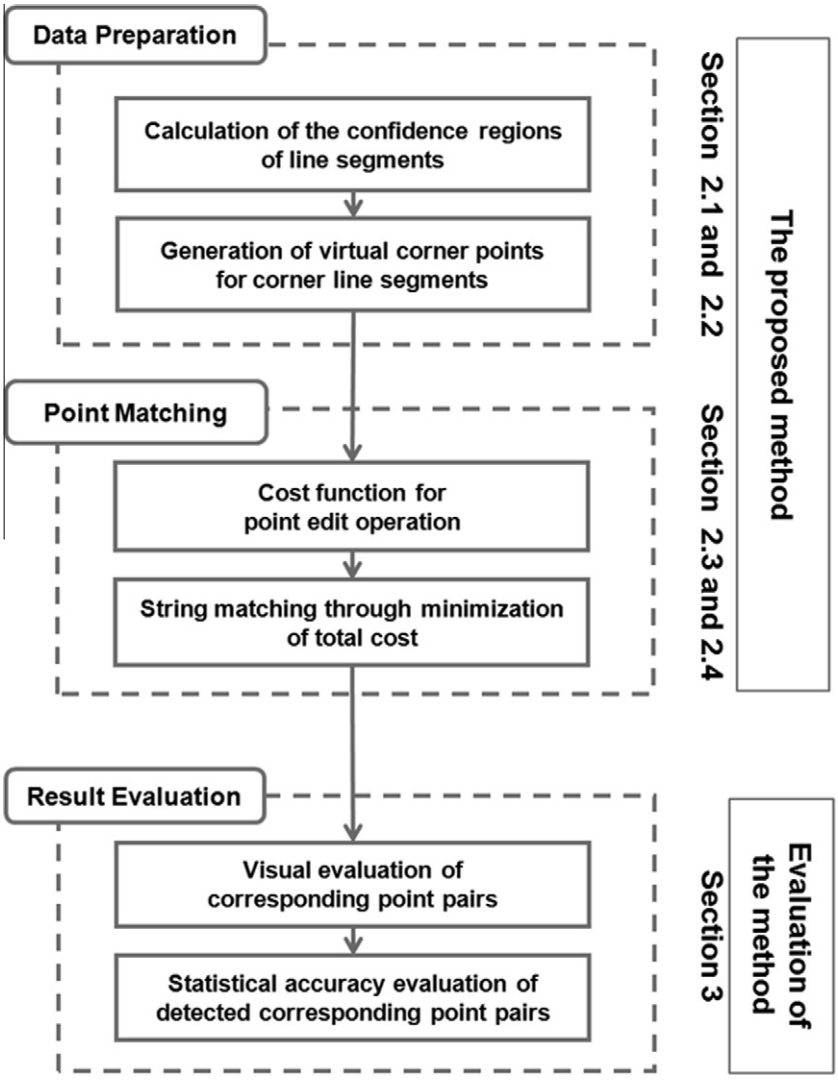


Fig. 2. The logic flow of the proposed method and the evaluation of the method.

line segment could improve the quality of the map alignment (Huh et al., 2011). Details of the step are presented in Sections 2.1 and 2.2.

The point matching step is comprised of two processes: determination of the cost functions for point edit operations of deletion, insertion and substitution and a string matching through the minimization of the total cost, also referred to as an edit distance, required to convert the contour of a target object into that of a reference object by means of editing the points of the target object. Details of this step are presented in Sections 2.3 and 2.4.

The result evaluation step assesses the result of the proposed method. We applied the proposed method to a cadastral map and a topographical map and then evaluated the result with a visual and statistical analysis. In the visual analysis, the detected corresponding point pairs according to several confidence levels and boundaries of two objects from the maps are overlapped and compared. In the statistical evaluation, we compared the corresponding point pairs detected by the proposed method and those manually detected. Details of the step are presented in Section 3.

2.1. Calculation of the confidence regions of line segments

Shi (1998) provides a statistical approach for modeling the positional error of geometric features with a confidence region model. A confidence region of a line segment is a band within which the true position of the line segment lies with a probability larger than a confidence level. The region was derived using the error of a line segment's two endpoints. Assuming a line segment $s_{0,1}$ with two end points p_0 and p_1, then P_0 and P_1 are estimators of the true positions of endpoint p_0 and p_1, respectively. The positional errors of the two points are modeled as the following equation (Meidow et al., 2009; Shi, 1998):

$$P_0 = \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_{0,X} \\ \mu_{0,Y} \end{bmatrix}, \begin{bmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{bmatrix} \right),$$
$$P_1 = \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_{1,X} \\ \mu_{1,Y} \end{bmatrix} \begin{bmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{bmatrix} \right) \quad (1)$$

where $\mu_{i,X}$ and $\mu_{i,Y}$ are the mean x- and y-direction position of point p_i, σ_X^2 and σ_Y^2 are their variances, and σ_{XY} is the covariance. A line segment is determined by a set of arbitrary points p_r on the line segment between p_0 and p_1 as shown in Fig. 3.

By applying the error propagation law to the above definition and assuming independent and equal variances and covariances, the positional error of a line segment can be represented by the following equation (Shi, 1998):

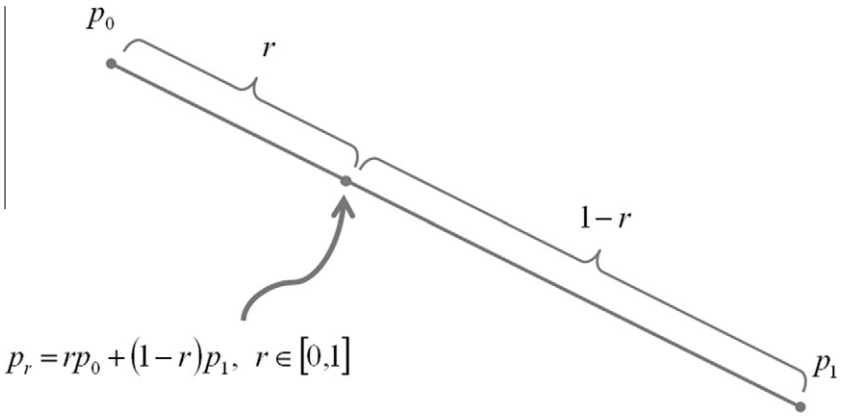


Fig. 3. An arbitrary points p_r on the line segment between p_0 and p_1.

$$P_r = \begin{bmatrix} X_r \\ Y_r \end{bmatrix} \sim N\left(\begin{bmatrix} (1-r)\mu_{0,X} - r\mu_{1,X} \\ (1-r)\mu_{0,Y} - r\mu_{1,Y} \end{bmatrix},\ ((1-r)^2 + r^2) \begin{bmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{bmatrix} \right), \quad r \in [0,1] \tag{2}$$

where p_r is an estimator of position of point p_r, X_r and Y_r are estimators of x- and y-direction positions of p_r, respectively. With the above error model, the confidence region $J_{0,1}$ of a line segment $s_{0,1}$ is defined as a region that all points p_r on the line segment are contained within the region with a probability larger than a confidence level γ, as described by the following equation:

$$\text{Prob}(p_r \in J_{0,1} \quad \text{for} \quad \text{all} \quad r \in [0,1]) > \gamma \tag{3}$$

where Prob( ) is a probability function. This region $J_{0,1}$ is the union of the confidence regions J_r of each point p_r that satisfy Eq. (4), where the parameter k is set as $\chi^2_{2;(1_\gamma)/2}$ (Shi, 1998).

$$\begin{aligned} X_r - [k((1-r)^2 + r^2)]^{1/2}\sigma_X \leqslant x_r \leqslant X_r + [k((1-r)^2 + r^2)]^{1/2}\sigma_X \\ Y_r - [k((1-r)^2 + r^2)]^{1/2}\sigma_Y \leqslant y_r \leqslant Y_r + [k((1-r)^2 + r^2)]^{1/2}\sigma_Y \end{aligned} \tag{4}$$

To practically calculate the confidence region $J_{0,1}$, the original confidence region could be simplified as the union of confidence regions of J_r as shown in Fig. 4. Thus confidence regions of p_r with a positional quality of the geospatial dataset and confidence level γ are necessary. Among these, the confidence level could be set by a user and the other could be estimated from the analysis of a training site. Details of this process are presented in Section 3.

2.2. Generation of virtual corner points for corner line segments

Sometimes, substantially corresponding areas between two objects are represented by different geometries, such as the areas within the circles of Fig. 5. While one corner of a map is represented as a line segment, its corresponding corner in the other map is represented as a point. In this case, a matching method chooses one of the following: matching of the corner point with either of the end points of the corner line segment or no matching for that area. If any of the matching pairs were used for map alignment, their neighboring line segments would be distorted. However, if no pair was used for that area, the alignment of the neighboring line segments would not be performed well because of the fewer corresponding point pairs.

This problem can be resolved by introducing the virtual corner point, which is the intersecting point of straight lines derived from the two line segments linked to both sides of the corner line segment (Huh et al., 2011). By inserting these points, sufficient candidate corresponding point pairs can be considered in the proposed matching process. However, there is a need to define a geometric condition to generate and insert these points into an object. In this

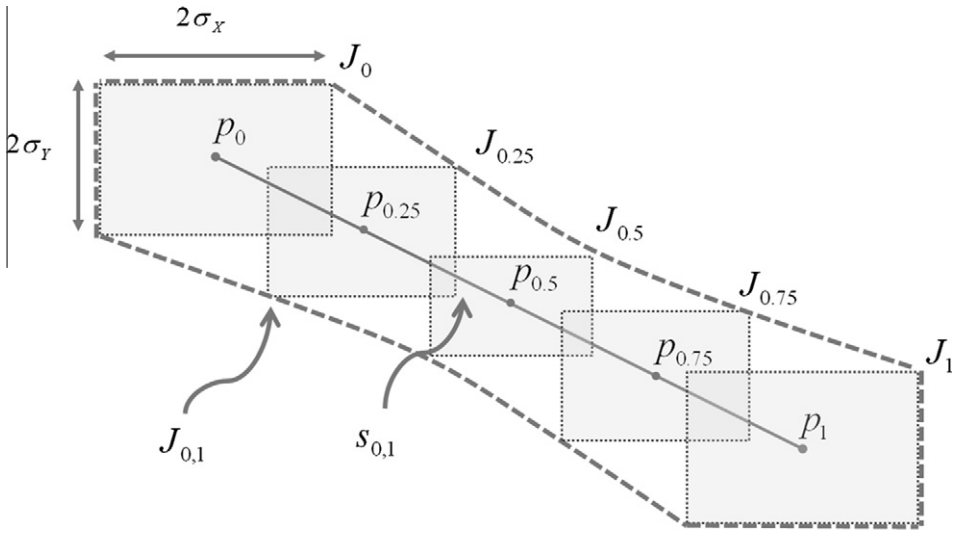


Fig. 4. Simplified confidence region $J_{0,1}$ of a line segment $s_{0,1}$.

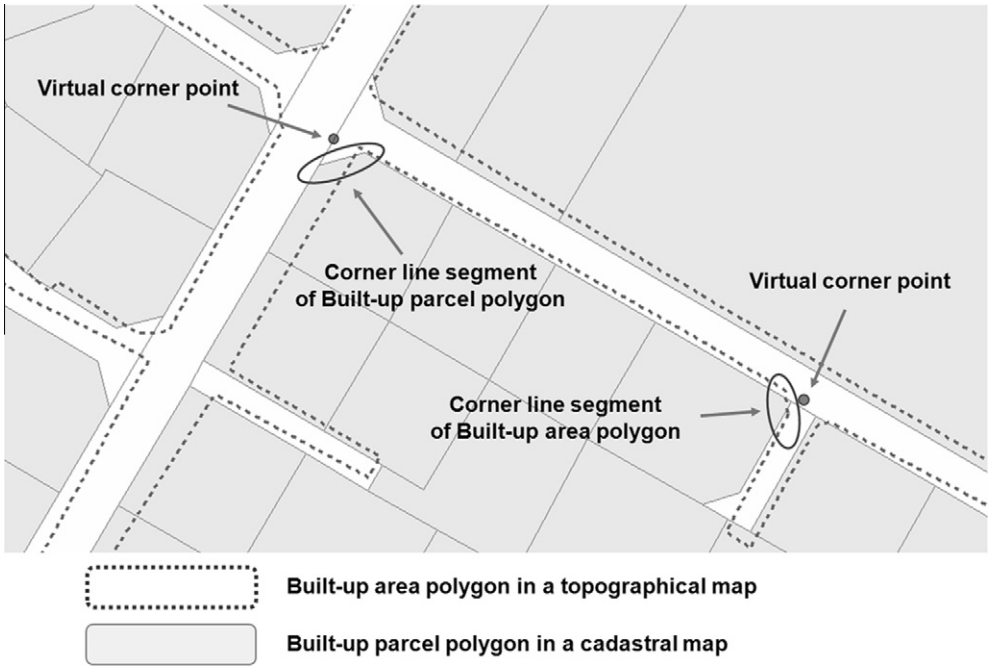


Fig. 5. Comparison of corresponding objects in a topographical map and a cadastral map.

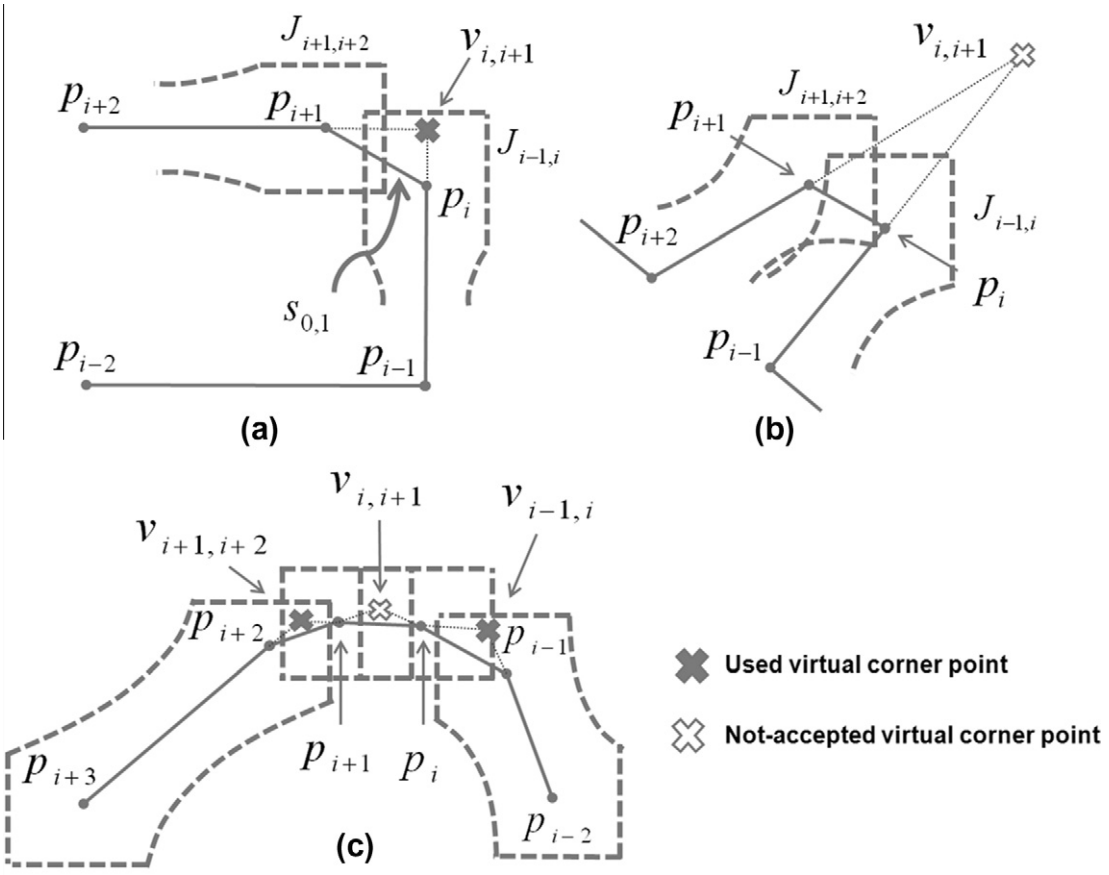


Fig. 6. Three cases for virtual corner point generation: (a) apparently different representation of a corner area, where the virtual corner point $v_{i,i+1}$ of corner line segment $s_{i,i+1}$ exist in one of the confidence regions of the two linked line segments $J_{i-1,i}$ and $J_{i+1,i+2}$, (b) substantially different representation of a corner area, (c) an additional constraint for virtual corner point in a curve area.

study, we determined whether the virtual corner point of a corner line segment to be generated or not based on the confidence regions of the line segments linked to both sides of the corner line segment, as shown in Fig. 6. A confidence region is defined as an

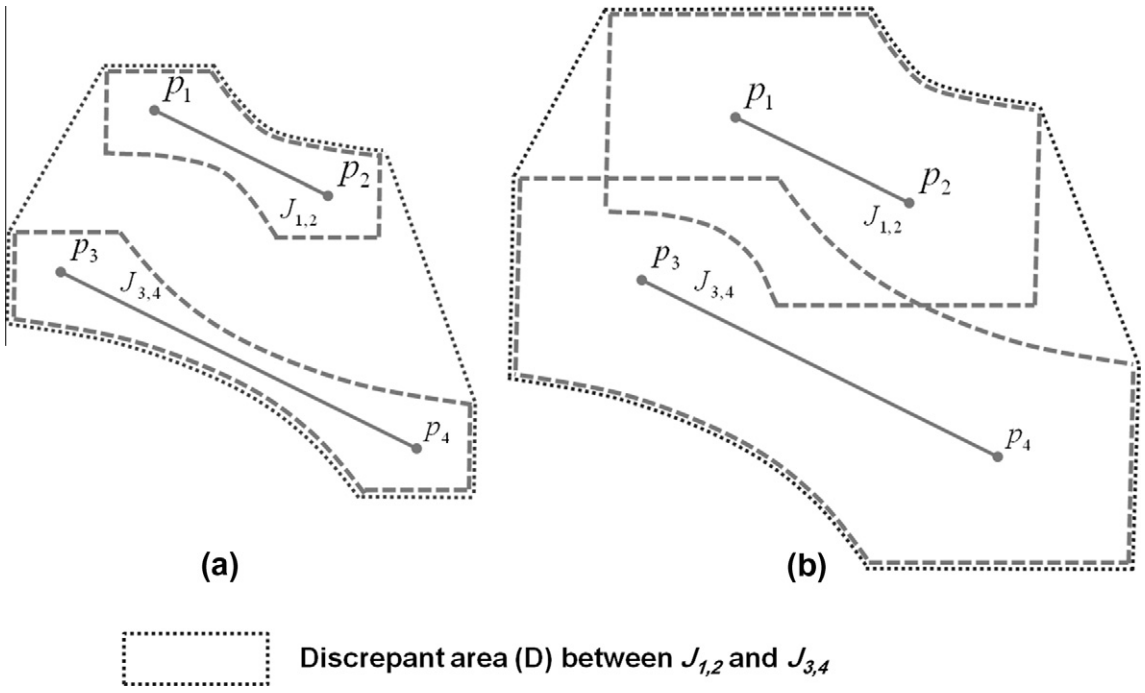


Fig. 7. Discrepant area between $J_{1,2}$ and $J_{3,4}$ with a small confidence level (a) and a large confidence level (b).

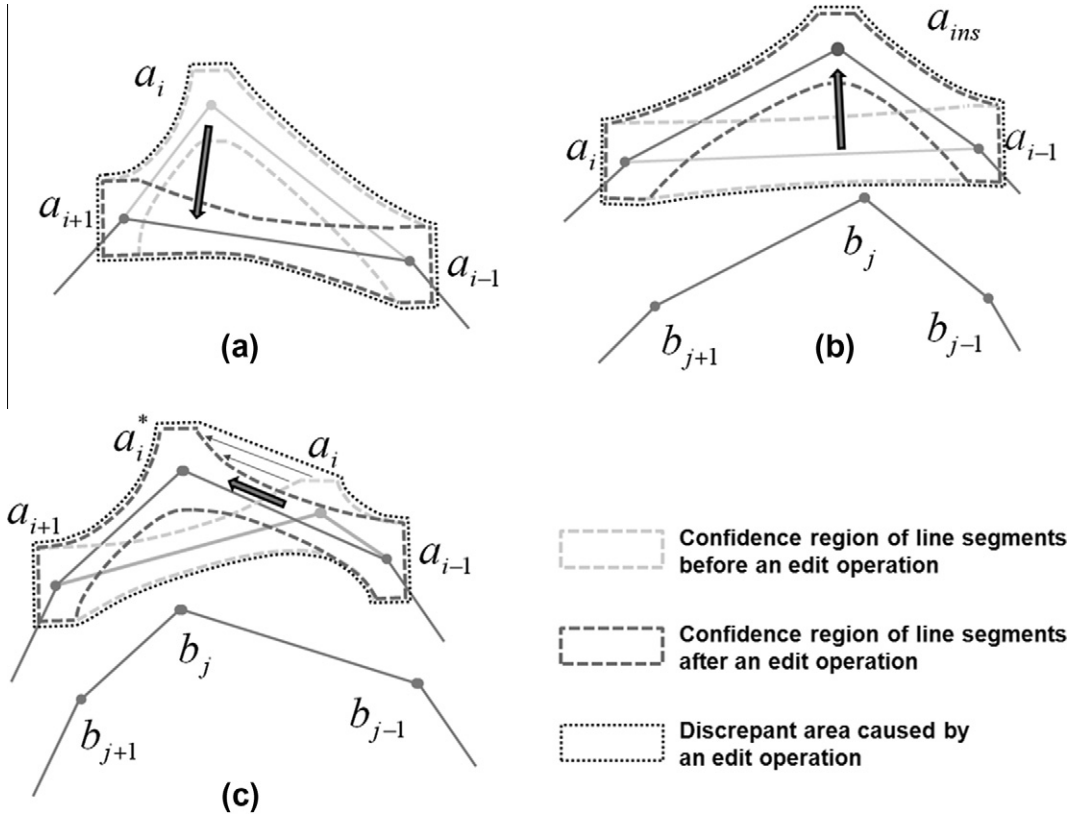


Fig. 8. Three types of point edit operations and their discrepant areas: (a) deletion, (b) insertion and (c) substitution.

area within which the true position exists with a probability larger than a confidence level (Shi, 1998). Therefore, if the virtual corner point $v_{i,i+1}$ of corner line segment $s_{i,i+1}$ exists in either of the confidence regions of the two linked line segments $J_{i-1,i}$ and $J_{i+1,i+2}$, the virtual corner point and corresponding corner line segment could be assumed to be apparently different representations of the corner area, as shown in Fig. 6a. However, if the virtual corner point does not exist in either of the confidence regions of the two linked line segments, they could be assumed to be substantially different representations of the corner area, as shown in Fig. 6b. Finally, we consider an additional constraint for a curved corner area, as shown in Fig. 6c, where virtual corner points would be successively

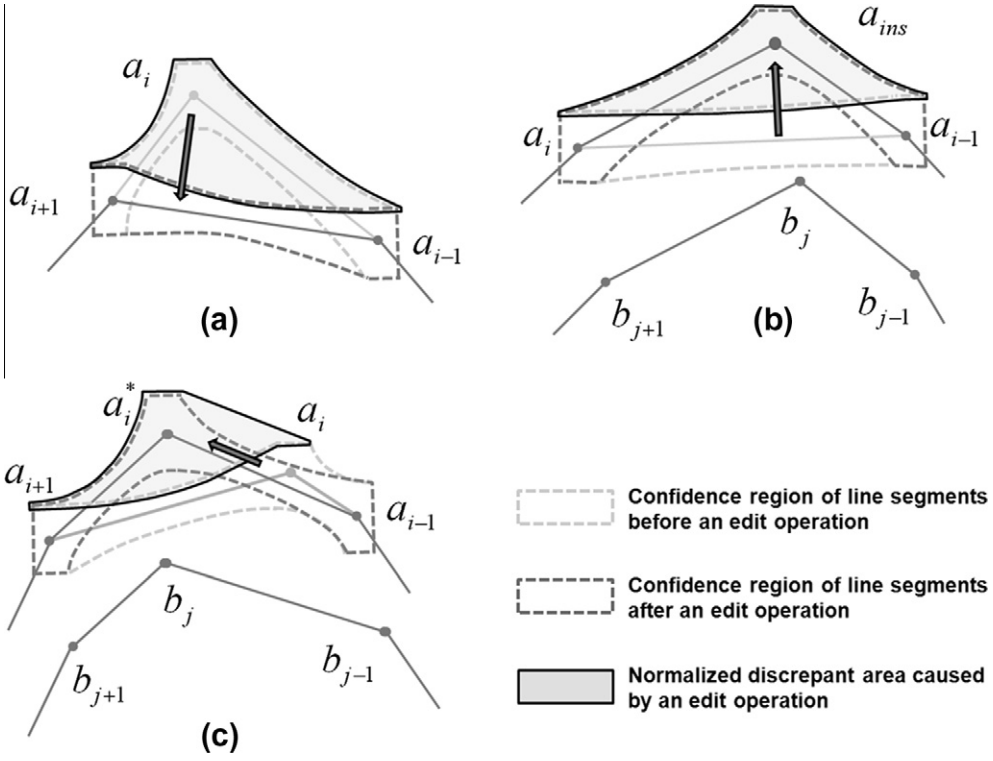


Fig. 9. Three types of point edit operations and their normalized discrepant areas: (a) deletion, (b) insertion and (c) substitution.

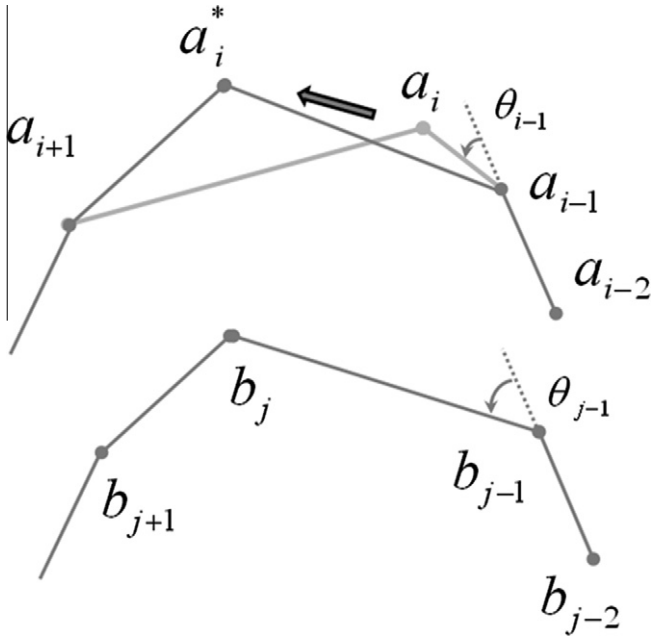


Fig. 10. Change of exterior angle at a_{i-1} from θ_{i-1} to θ_{j-1}.

generated. Virtual corner points could improve the performance of the map alignment by means of properly aligning two line segments linked to both sides of the corner line segments. However, these successive short line segments in a curved corner area could be just different representations of corner points, not meaningful line segments for which corresponding line segments are to be found. In this case, the segments would generate redundant virtual corner points, which could give unreasonable results. To prevent these points, we add a constraint that both the proceeding and following line segments are long enough so that the confidence regions of their endpoints do not overlap each other. The aforementioned conditions for a virtual corner point can be described by the following equation:

$$\left\{ v_{i,i+1} \middle| \begin{array}{c} (v_{i,i+1} \cap union(J_{i-1,i}, J_{i+1,i+2}) \neq \phi) \quad \text{and} \\ (J_{i-1} \cap J_i \neq \phi \quad \text{and} \quad J_{i+1} \cap J_{i+2} \neq \phi) \end{array} \right\} \tag{5}$$

2.3. Cost determination of the point edit operation

Because spatial uncertainties arise in data acquisition and representation processes, the positions of corresponding spatial objects do not overlap with the true position of the real world entity that the objects represent. Under this circumstance, a proper error model is necessary to measure the cost of a point edit operation in terms of the change in confidence regions. There are several indicators that measure errors between two regions, such as commission error, omission error, discrepant area and normalized discrepant area (Shi et al., 2003).

Among them, the normalized discrepant area measure was chosen to calculate the cost of point edit operations. Because the positional error of each point between two regions is measured by the discrepancy between them, the sum of the all positional errors of all the point pairs between the regions denote the error between the regions as a whole (Shi et al., 2003). Therefore, the discrepant area (D) is defined as the union of the discrepancies of all point pairs between the regions, as shown in Fig. 7. However, the area measures an error between two identical line segments as the area of their confidence regions, not zero even thought there is no difference between the line segments. To address this problem, the normalized discrepant area (ND) is proposed, as described by the following equation (Shi et al., 2003):

$$\mathrm{ND}(J_{1,2}, J_{3,4}) = \mathrm{Area}(D) - \min\{\mathrm{Area}(J_{1,2}), \mathrm{Area}(J_{3,4})\} \tag{6}$$

```
T(0,0) = 0                // initialization
ED(0,0) = 0
for i = 1 to n
    T(i,0) = -1
    ED(i,0) = ED(i-1,0) + c_del(a_i → Φ)
end
for j = 1 to m
    T(0,j) = +1
    ED(0,j) = ED(0,j-1) + c_ins(Φ → b_j)
end
for i = 1 to n
    for j = 1 to m
        T_1 = -1          // deletion
        ED_1 = ED(i-1, j) + c_del(a_i → Φ)
        T_2 = +1          // insertion
        ED_2 = ED(i, j-1) + c_ins(Φ → b_j)
        T_3 = 0           // substitution
        if ( J_ai ∩ J_ai*) = null   // distance constraint
            ED_3 = inf.
        else
            ED_3 = ED(i-1,j-1) + c_sub(a_i → b_j)
        end
        k = argmin(ED_1, ED_2, ED_3)  // cost minimization
        T(i,j) = T_k
        ED(i,j) = ED_k
    end
end
```

Fig. 11. Pseudo code of the proposed method to find corresponding point pairs.

The following cost functions to convert the contour of a target object into that of a reference object by means of editing the points of the target object are measured by the normalized discrepant area. When a point edit operation changes the position of confidence regions of the involved line segments, the normalized discrepant area between confidence regions before and after the operation is used for the cost. Details of each operation and their costs are as follow.

2.3.1. Deletion edit operation

A deletion edit operation $o(a_i \rightarrow \phi)$ removes a point a_i from a target object A as shown in Fig. 8a, so that the two line segments, $s_{i-1,i}$ and $s_{i,i+1}$ become a single line segment $s_{i-1,i+1}$. Therefore, a cost of a deletion edit operation is calculated as a normalized discrepant area between the union of $J_{i-1,i}$ and $J_{i,i+1}$, and $J_{i-1,i+1}$, as described by Fig. 9a and the following equation:

$$c_{del}(a_i \rightarrow \phi) = \mathrm{ND}(Union(J_{i-1,i}, J_{i,i+1}), J_{i-1,i+1}) \tag{7}$$

2.3.2. Insertion edit operation

An insertion edit operation $o(\phi \rightarrow b_j)$ inserts point b_j of a reference object B into its relative position on a target object A shown in Fig. 8b, so that a single line segment, $s_{i-1,i}$, is split into two line segments, $s_{i-1,ins}$ and $s_{ins,i}$. Essentially, our spatial uncertainty model assumes random errors, and therefore, the original position of b_j should be used for that of the inserted point. However, even though corresponding point pairs over the entire experimental coverage have random positional errors, the local corresponding point pairs in a single corresponding object pair could have auto-correlated positional errors. To address this problem, the position of an inserted point a_{ins} is obtained by the relative position of b_j based on the latest corresponding point pair, if two points of a_{i-1} and b_{j-1} are a corresponding point pair, they become a_{LS} and b_{LS}, respectively.

$$a_{ins} = a_{LS} + (b_j - b_{LS}) \tag{8}$$

where a_{LS} and b_{LS} are a point pair on which the latest substitution operation is chosen before an involved edit operation. Then, the cost of an insertion point operation is calculated as a normalized discrepant area between a union of $J_{i-1,ins}$ and $J_{ins,i}$, and $J_{i-1,i}$, as described by Fig. 9b and the following equation:

$$c_{ins}(\phi \rightarrow b_j) = \mathrm{ND}(J_{i-1,i}, Union(J_{i-1,ins}, J_{ins,i})) \tag{9}$$

2.3.3. Substitution edit operation

A substitution edit operation $o(a_i \rightarrow b_j)$ moves point a_i in a target object A to the relative position a_i^* of b_j in a reference object B shown in Fig. 8c, so that the shape of a polyline with two line segments, $s_{i-1,i}$ and $s_{i,i+1}$, is converted into that with two line segments s_{i-1,i^*} and $s_{i^*,i+1}$. Here, the position of a_i^* is similarly obtained as a_{ins} with Eq. (8). This substitution operation moves a polyline with two line segments, $s_{i-1,i}$ and $s_{i,i+1}$ at the same time, thus its discrepant area and normalized discrepant area are measured as shown in Figs. 8c and 9c.

Different from the previous edit operations, an angle change penalty term is considered when the cost of a substitution edit operation is calculated as Eq. (10). Given (a_{i-1}, b_{j-1}) and (a_{i-2}, b_{j-2}) are the latest corresponding point pairs, the angle change of line segments between $s_{i-1,i}$ and s_{i-1,i^*} is measured as the difference between exterior angles at a_{i-1} and b_{j-1} as shown in Fig. 10. If there is no two previous corresponding point pairs such as a substitution edit operation involved with the first or second vertex of any contours, $c_{sub}(a_i \rightarrow b_j)$ is calculated only with $\mathrm{ND}(J_{i-1,i,i+1} J_{i-1,i^*,i+1})$.

$$c_{sub}(a_i \rightarrow b_j) = \mathrm{ND}(J_{i-1,i,i+1}, J_{i-1,i^*,i+1}) \cdot \exp(\lambda|\theta_{i-1} - \theta_{j-1}|) \tag{10}$$

where $J_{i-1,i,i+1}$ denotes an union of $J_{i-1,i}$ and $J_{i,i+1}$. $J_{i-1,i^*,i+1}$ denotes an union of J_{i-1,i^*} and $J_{i^*,i+1}$. λ denotes a coefficient for angle change and is set to 0.1. θ_{i-1} and θ_{j-1} denote exterior angles in radian at vertex a_{i-1} and b_{j-1}, respectively.

In this study, the confidence region of a spatial object is a region within which the object's true position lies with a probability larger than a confidence level. Therefore, the confidence regions of corresponding points should intersect each other. To impose this constraint, Eq. (10) is modified as the following equation:

Table 1
Specifications of the experimental datasets.

	Cadastral map	Topographical map
Mapping institute	Korea cadastral survey Corp.	National geographic information institute
Coordinate system	Bessel TM	GRS-80
Scale	1:800	1:1000
Corresponding class	Built-up area	
Test site	Central urban area of Suwon	
Coverage	1.5 km by 1.5 km	

Table 2
Error estimation from corresponding point pairs at a training site.

	σ_X	σ_Y
Measured σ_X and σ_Y between the maps	0.814	0.738
Estimated σ_X and σ_Y of each map	0.576	0.522

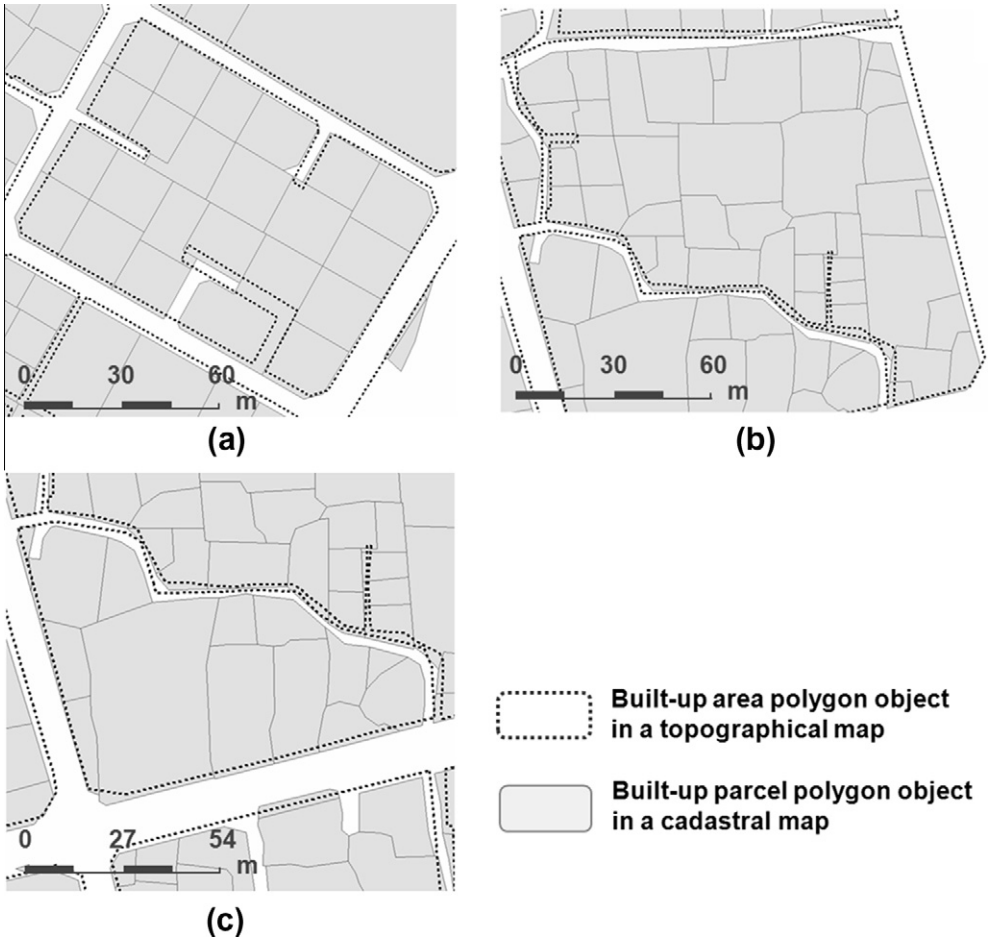


Fig. 12. Three corresponding polygon object pairs between a topographical map and a cadastral map.

$$c_{sub}(a_i \rightarrow b_j) = \begin{cases} \mathrm{ND}(J_{i-1,i,i+1}, J_{i-1,i^*,i+1}) \cdot \exp(\lambda|\theta_{i-1} - \theta_{j-1}|) & \text{if } J_{ai} \cap J_{ai^*} \neq \text{null} \\ \inf & \text{otherwise} \end{cases} \quad (11)$$

where exp and inf denote an exponential function and an infinite value, respectively.

2.4. String matching through minimization of total cost

There have been various studies that use a string matching technique for shape recognition (Chen et al., 1998) and shape matching (Kaygin and Bulut, 2002). Their methods represent the boundary contour of an object with a string, where each attributed characteristic corresponds to points on the contour. Let $A = [a_1, a_2, \ldots, a_n]$ and $B = [b_1, b_2, \ldots, b_m]$ denote two point sequences of objects to be matched. An edit sequence is defined as a sequence of ordered edit operations $O = [o_1, o_2, \ldots, o_N]$, where o_i is one of the three types of edit operations. If c_i is the cost of o_i, then the edit distance between A and B is defined as the minimum total cost of operation sequence that converts A into B, as described by the following equation:

$$\mathrm{ED}(A, B) = \min\left\{\sum_O c_i | \mathrm{O \text{ is an edit sequence converting A to B}}\right\} \quad (12)$$

The optimization of the edit sequence to attain the minimum edit distance can be solved by a dynamic programming technique based on the following property of the following equation (Bunke and Buhler, 1993):

$$\mathrm{ED}(A_{\langle 1,i\rangle}, B_{\langle 1,j\rangle}) = \min\left\{\begin{array}{l} ED(A_{\langle 1,i-1\rangle}, B_{\langle 1,j\rangle}) + c_{del}(a_i \rightarrow \phi) \\ ED(A_{\langle 1,i\rangle}, B_{\langle 1,j-1\rangle}) + c_{ins}(\phi \rightarrow b_j) \\ ED(A_{\langle 1,i-1\rangle}, B_{\langle 1,j-1\rangle}) + c_{sub}(a_i \rightarrow b_j) \end{array}\right\} \quad (13)$$

where $A_{\langle 1,i\rangle} = [a_1, a_2, \ldots, a_i]$ and $B_{\langle 1,j\rangle} = [b_1, b_2, \ldots, b_j]$ denote two partial point sequences from a_1 to a_i, and from b_1 to b_j, respectively. However, the point pairs for the substitution edit operation cannot be obtained from the minimum edit distance itself. It is necessary to maintain the sequences of the previously chosen edit operations. Therefore, an edit operation matrix T, where $T(i,j)$ records the latest chosen edit operation of $\mathrm{ED}(A_{\langle 1,i\rangle}, B_{\langle 1,j\rangle})$ is also constructed. With this matrix, the optimal edit sequence for $\mathrm{ED}(A_{\langle 1,i\rangle}, B_{\langle 1,j\rangle})$ could be obtained by a backtracking analysis from $T(i,j)$. This analysis checks the previously chosen edit operation among the deletion from $T(i-1,j)$, the insertion from $T(i,j-1)$, and substitution from $T(i-1,j-1)$ and repeats these steps until $T(0,0)$ is reached (Kruskal 1983). A final sequence is also obtained by the backtracking analysis from $T(n,m)$. These two matrices could be obtained by the following pseudo code shown in Fig. 11.

Before we apply the above method, two practical problems should be addressed. Firstly, when we calculate the cost of an insertion or substitution edit operation, sometimes there is no previously determined corresponding point pair for a_{LS} and b_{LS} in Eq. (8). To address this problem, we set the origin of coordinates (0,0) and (0,0) as an initial corresponding point pair. Secondly, when we apply the above dynamic programming technique to two polygon objects, a problem occurs related to the starting points of the objects. Generally, the starting point of the contour of a polygon object is arbitrary. Given $A = [a_1, a_2, \ldots, a_n]$ and $B = [b_1, b_2, \ldots, b_m]$, a_1 can correspond to a_k and then a_2 to b_{k+1},

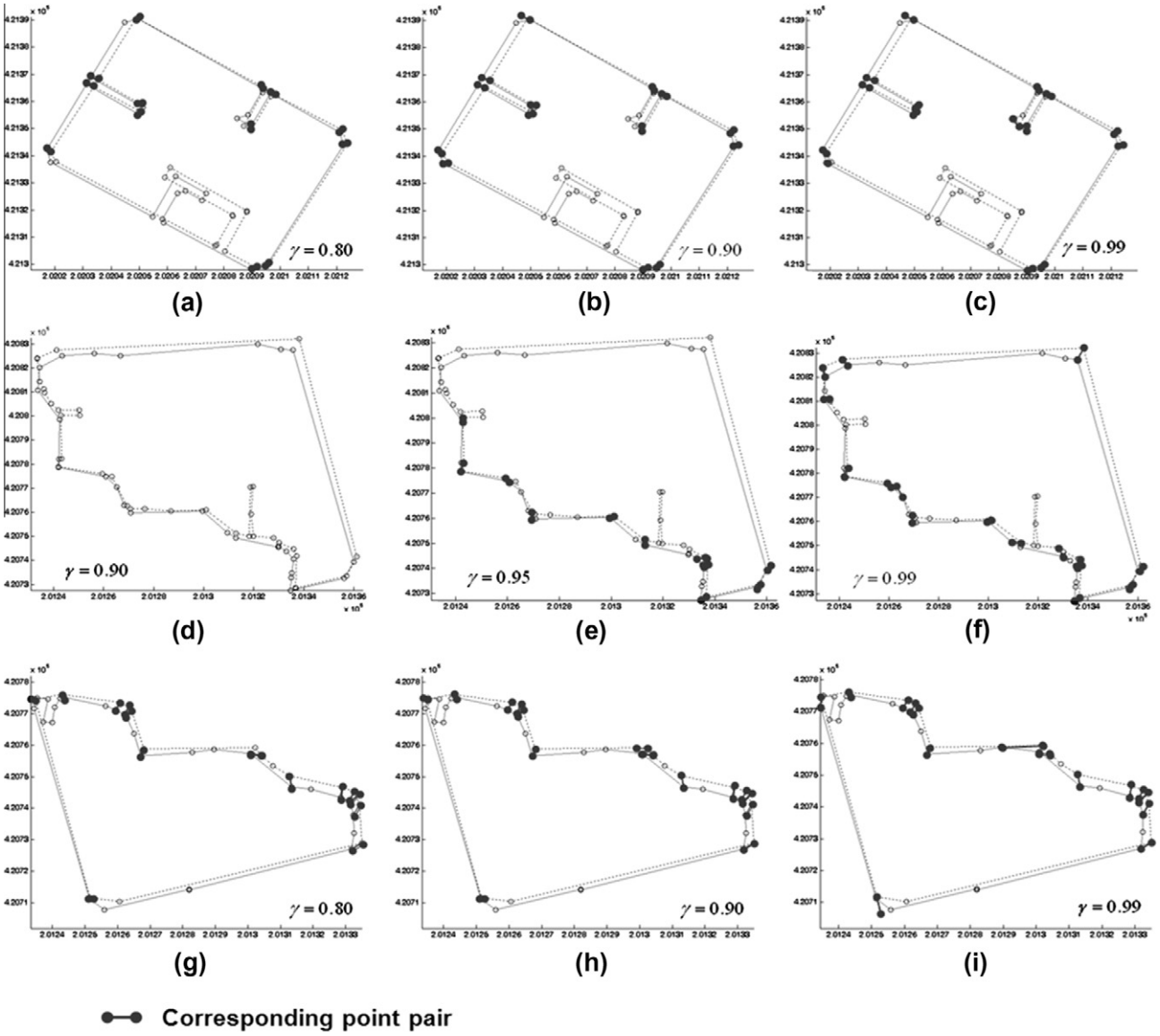


Fig. 13. Comparison of detected corresponding point pairs of Fig. 12 according to confidence levels at 0.80, 0.90 and 0.99 for Fig. 12a and c and 0.90, 0.95 and 0.99 for Fig. 12b.

sequentially. In this case, the proposed method should be applied for $A = [a_1, a_2, \ldots, a_n]$ and $B' = [b_k, b_{k+1}, \ldots, b_m, b_1, \ldots, b_{k-1}]$. This problem can be resolved by fixing a starting point of one object and then shifting the starting point one by one in the string of the other object (Bunke and Buhler, 1993). Among all cost matrices according to the shifting, the matrix that has the minimum final cost value at $ED(n,m)$ is chosen and its operation matrix T is used to determine corresponding point pairs.

3. Experiment and result

3.1. Preparation with spatial uncertainty between two geospatial datasets

We applied the proposed method to two heterogeneous geospatial datasets, the cadastral map of Korea land information system (KLIS) and the national base topographical map of the National Geographic Information Institute. The experiment area chosen was the central urban area of Suwon. Details of the datasets are presented in Table 1.

Before applying the proposed method, a corresponding class pair from which the corresponding object pairs are searched should be determined. In this study, we chose the built-up area for the corresponding class. Though there are several classes that correspond to each other between the two maps, many of them are related to transportation classes, such as roads, or hydrology classes, such as rivers. In general, these objects constitute network structures, and thus, it is not suitable to derive object pairs from which corresponding contours are obtained (Huh et al., 2011). Therefore, we alternatively use objects in a built-up area class that are contained by transportation or hydrology objects because built-up area objects are generally isolated each other and it is easy to find corresponding objects between geospatial datasets (Timpf, 1998).

We first chose 100 corresponding point pairs at a training site and measured the variances between the maps in the x- and y-coordinates. Assuming the errors in the maps are independent and identical, we divided them by $\sqrt{2}$ and estimated the variances of each map as shown in Table 2. Now the processes of generation of a virtual corner point, calculation of a point edit operation cost and distance constraint for a substitution edit operation can use

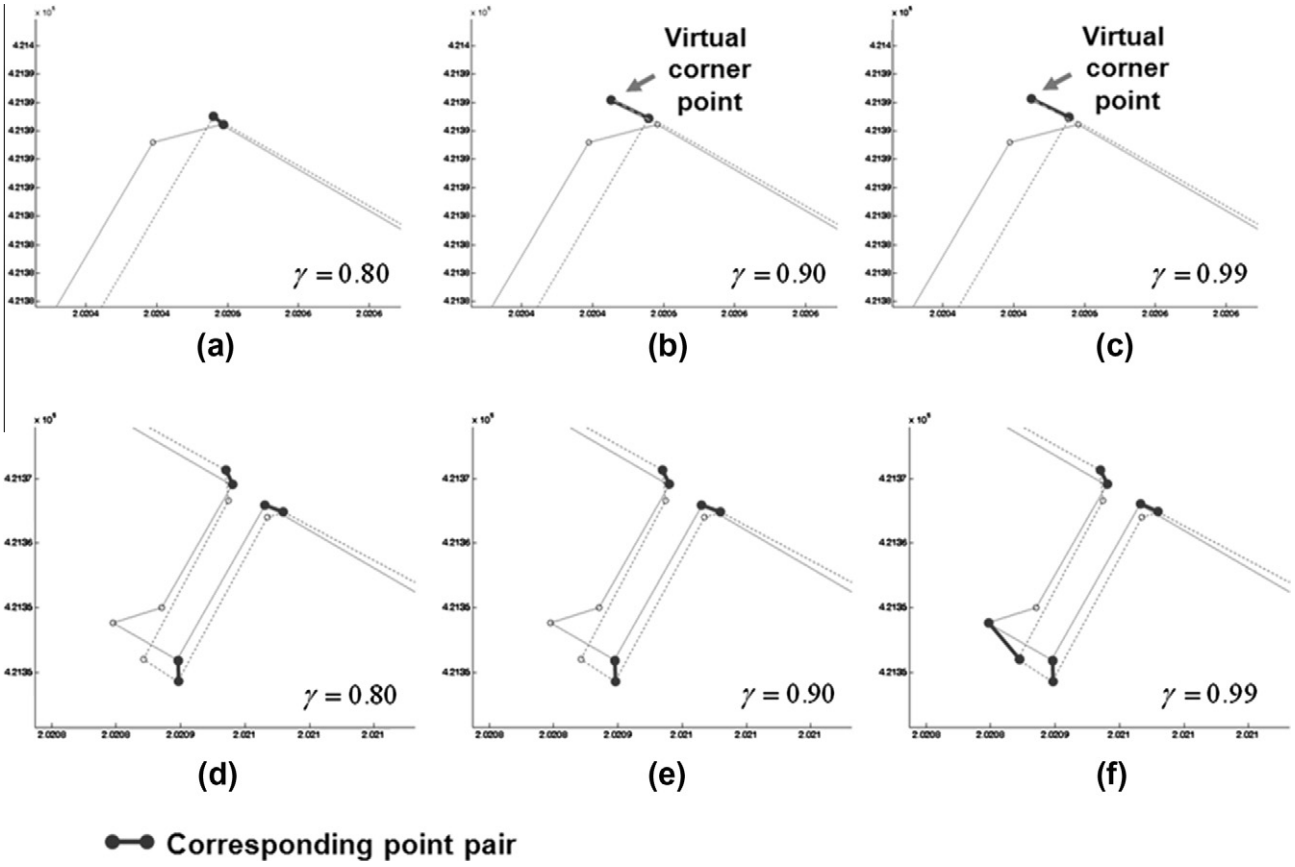


Fig. 14. Comparison of detected corresponding point pairs of Fig. 12a according to confidence levels at 0.80, 0.90 and 0.99.

the above error estimations. However, it is unclear what confidence level would show an optimal matching result. Therefore, we chose candidate levels, 0.80, 0.85, 0.90, 0.95 and 0.99, then evaluated the levels with the experimental dataset.

3.2. Result and discussion

The proposed method was evaluated with 100 built-up area polygon object pairs at the test site. We chose three object pairs in Fig. 12 and compared their detected corresponding point pairs according to three confidence levels as shown in Fig. 13.

Fig. 14 compares the detected corresponding point pairs of Fig. 12a around the upper left corner area (Fig. 14a–c) and a dead-end street (Fig. 14d–f). The corner area is represented by an edge in the cadastral map and a point in the topographical map. When the confidence level was set at 0.8, the corner edge did not generate a virtual corner point. Thus the point in the cadastral map was matched to the closer end point of the corner edge as shown in Fig. 14a. However, considering the purpose to align contours of corresponding objects, it needs to find a point pair which properly aligns the line segments linked to the corner areas of each map. This was achieved by inserting a virtual corner point. As shown in Fig. 14b and c, the virtual corner points improved alignment performance especially for corresponding corner areas where one is represented by a point and the other by an edge. Around the dead-end street in Fig. 14d–f, the results of the entrance corners were same regardless of the confidence levels. However, the results of the dead-end corner edge were different. Confidence levels at 0.80 and 0.90 found only one point pair and the level at 0.99 found two point pairs. In fact, it is unclear which result is a true one because the left corner of the dead-end corner has different shapes. Therefore, the above results indicate that; as a confidence level of the proposed method increases, corresponding points whose positions and the geometries of neighboring line segments are different begin to become a corresponding point pair.

The above property is also shown in Fig. 15 that confidence levels at 0.80, 0.85 and 0.90 found no corresponding point pairs. After a confidence level at 0.95, the proposed method began to find corresponding point pairs. This means that the optimal confidence level for an object pairs should be chosen according to the geometric differences of involved objects. As magnitudes of discrepancies between corresponding points are large and directions of them are irregular as shown in Fig. 15, a sufficient large confidence level should be selected.

However, a large confidence level can present more false corresponding point pairs as shown in Fig. 16. In the Fig. 16a–c, a larger confidence level found more corresponding point pairs even though some of these pairs deformed much of the geometries of original line segments especially in Fig. 16c. This is because detailed geometries of line segments' confidence regions are smoothed as a confidence level increases as shown in Fig. 7. Thus different local-scale geometries are underestimated, and thus the proposed method chooses more substitution edit operations to deform the contour of a target object in that of a reference object. This property can distort not only involved a point pair but also its neighboring pairs. The proposed method with confidence levels at 0.80 and 0.90 determined the virtual corner point and b_1 as a corresponding point pair instead of (a_1,b_1) or (a_2,b_1) as shown in Fig. 16d and e. This is because the neighboring corresponding point pair (a_0,b_0) worked as a_{LS} and b_{LS} in Eq. (8). According to Eq. (8), the previously determined corresponding point pair affects its following edit operations by adjusting the position of an inserted or substituted point. Therefore, the neighboring corresponding point pairs in Fig. 16d–f moved the position of the cadastral map to

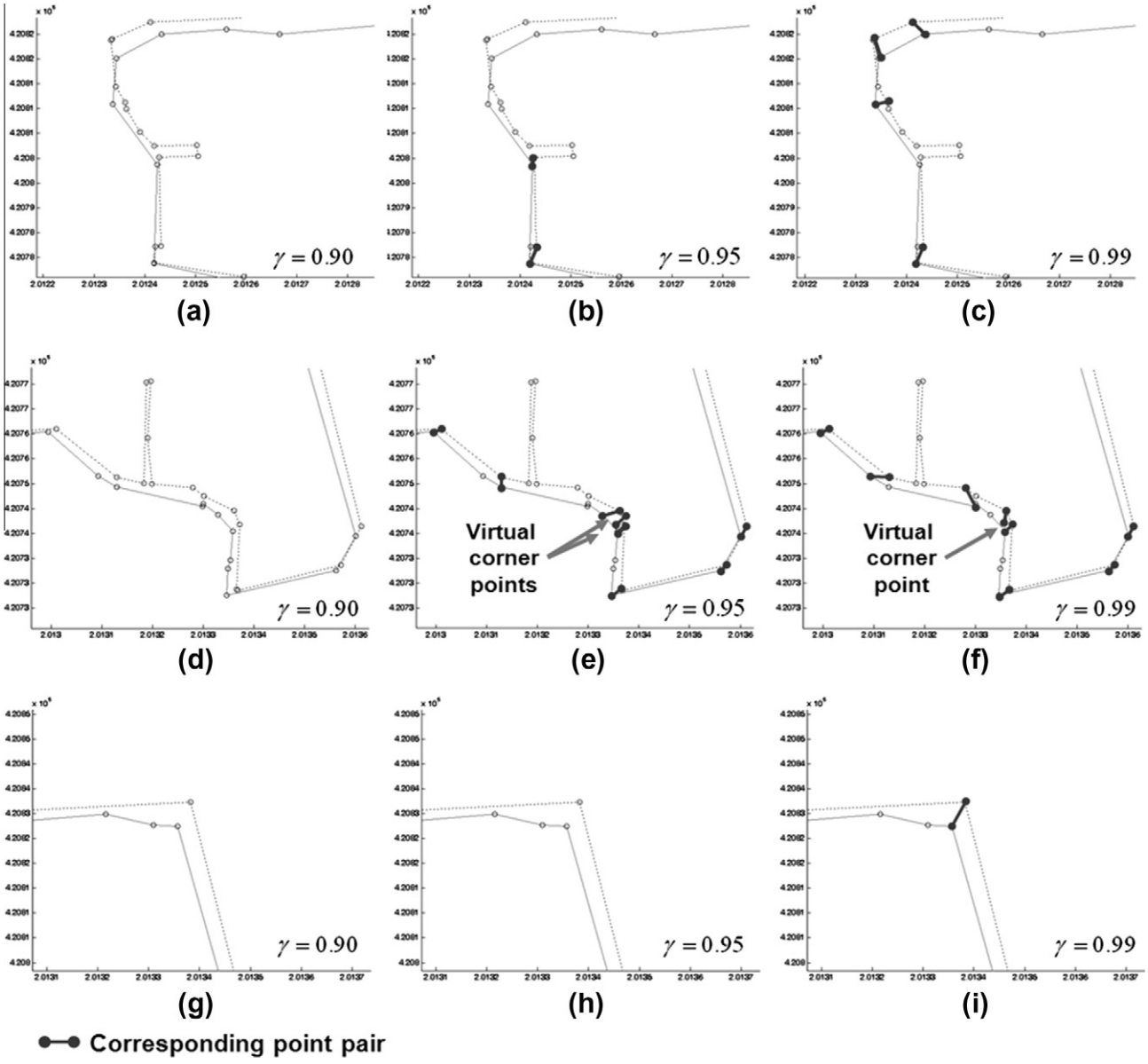


Fig. 15. Comparison of detected corresponding point pairs of Fig. 12b according to confidence levels at 0.90, 0.95 and 0.99.

slightly upper right position when the involved edit costs were calculated. Meanwhile, in case of confidence level at 0.99, a pair of (a_1, b_1) became a corresponding point pair because b_2 chose the virtual corner point in Fig. 16i as its corresponding point instead of a_3 in Fig. 16g and h.

To analyze the effect of a confidence level to determine an optimal edit sequence, one simple alignment of two polylines in Fig. 17 was examined whether a substitution edit operation of (a_i, b_j) is chosen or not according to different confidence levels. For simplicity, (a_{i-2}, b_{j-2}) and (a_{i-1}, b_{j-1}) are assumed to be corresponding point pairs. Thus it only needs to compare two edit sequences; substitution of a_i and a_i^* with the cost of $c_{sub}(a_i \to b_j)$ and inserting a_{ins} then deleting a_i with the cost of $c_{ins}(\phi \to b_j) + c_{del}(a_i \to \phi)$. With the error estimation in Table 2, Fig. 18 shows that the ratio of $c_{sub}(a_i \to b_j)$ to $c_{ins}(\phi \to b_j) + c_{del}(a_i \to \phi)$ decreases as a confidence level increases. This means that a large confidence level measures relatively a less cost to a substitution edit operation comparing to an alternative edit sequence to align the two polylines in Fig. 17, thus more corresponding point pairs can be found even though geometric discrepancies are very large. This also means that a large confidence level would lead to erroneous deformation of individual line segments with false corresponding point pairs. Thus it needs to find an optimal confidence level with a statistical evaluation.

To statistically evaluate the performance of the proposed method, we compared the detected pairs by the proposed method according to confidence levels with those manually detected. We used three types of measures: commission error, omission error and the F-measure from Eq. (14) (Euzenat and Shvaiki, 2007). As shown in Table 3, the result using a confidence level at 0.95 presented the highest F-measure. All of the confidence levels, omission errors are larger than commission errors. This is because we added a distance threshold and the penalty term of angle change as Eq. (11).

$$\text{F-Measure} = \frac{(1 - E_C) \times (1 - E_O)}{0.5 \times (1 - E_C) + 0.5 \times (1 - E_O)} \tag{14}$$

where E_C denotes a commission error and E_O denotes an omission error.

We also compared the results of this study and those of Huh et al. (2011) to evaluate the improvement of the proposed method.

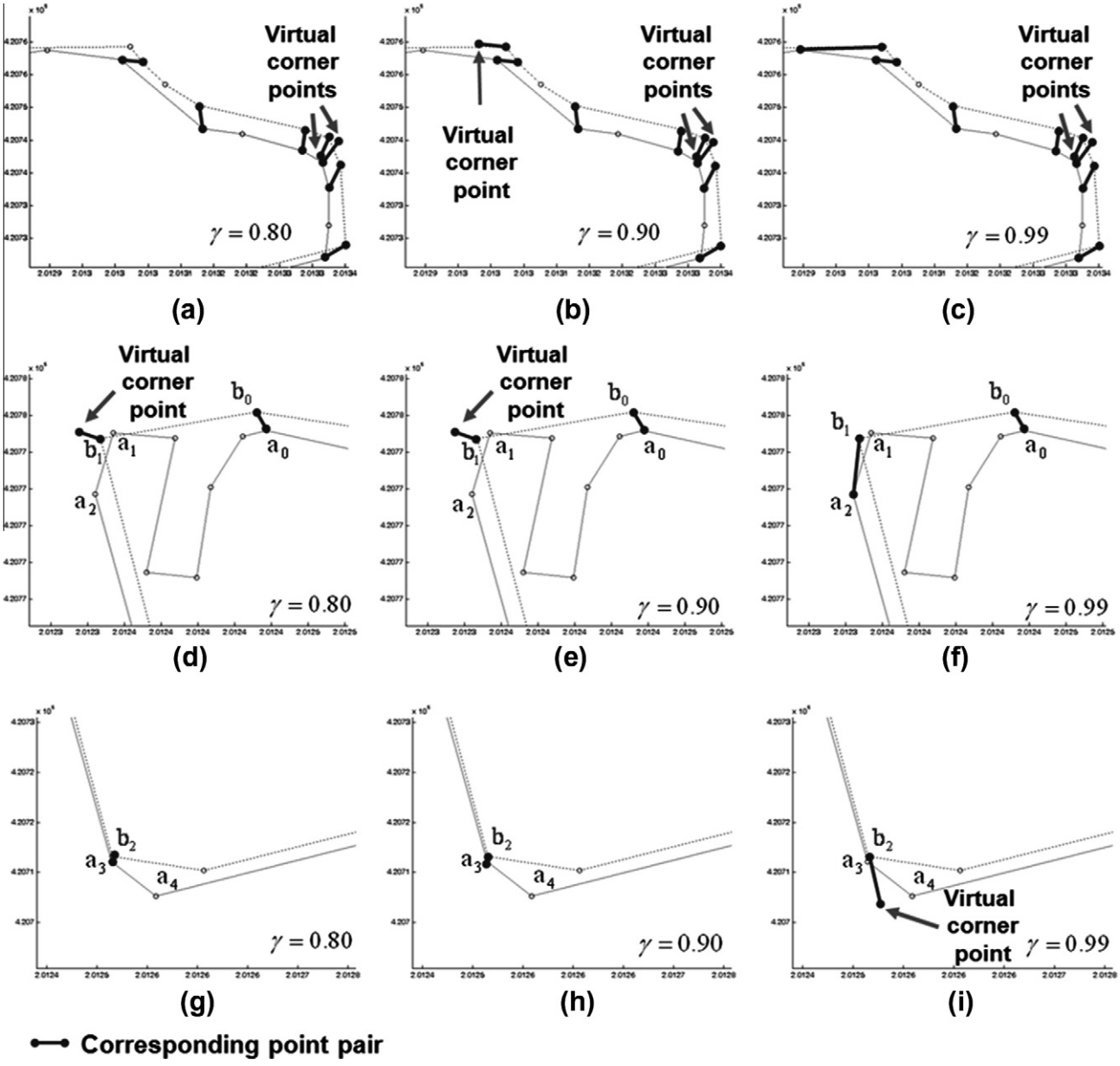


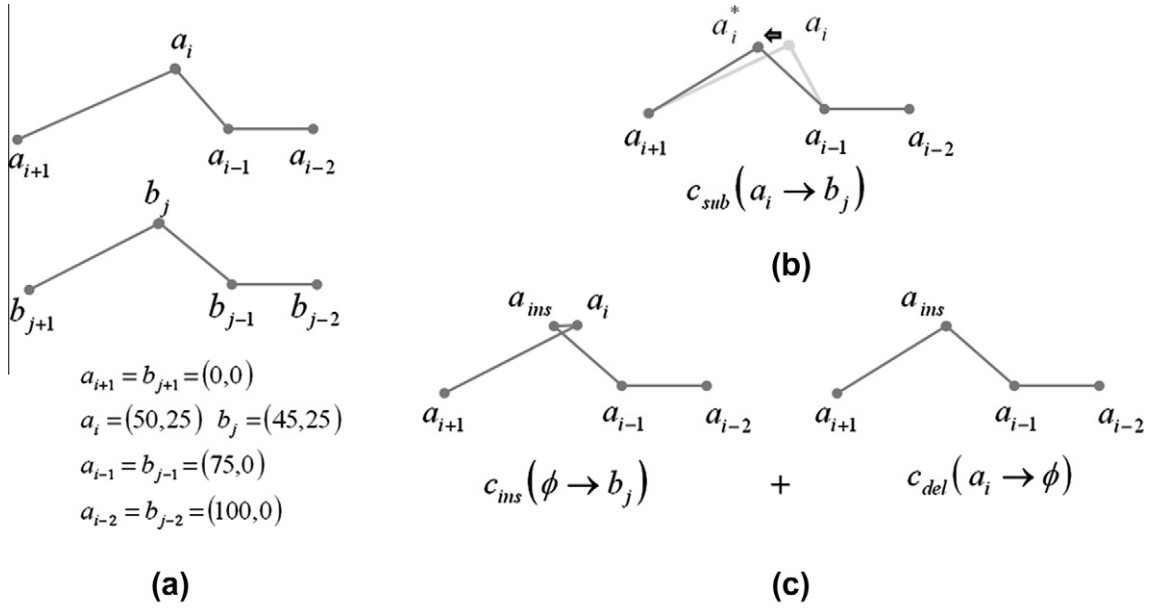
Fig. 16. Comparison of detected corresponding point pairs of Fig. 12(c) according to confidence levels at 0.80, 0.90 and 0.99.

The processes of the two methods are similar in that they have two steps: generation of virtual corner point and detection of corresponding point pairs with a string matching method. In fact, the first step is a pre-process to improve the performance of the matching step by means of approximating shapes of two polygon objects. Therefore, we only compared matching results caused by different edit operation cost functions in the second step. Given object pairs with a simple shape, both methods presented nearly the same results, whereas in the case with complicated boundaries composed of short line segments, different results were obtained. The previous method detected a smaller number of corresponding point pairs. This difference is caused by the characteristics of the cost functions. The previous cost functions of Huh et al. (2011) are based on a physical deformation energy model, which assumes line segments as elastic materials and combines stretching and bending energy of the segments, as described by Eq. (15). Each energy is obtained from the changes in the length (Δl) and angle($\Delta\theta$) of the line segments and combined with a weight coefficient $\alpha = 0.98$. Thus a larger change of line segment's length or interior angle between linked line segments to align contours causes a larger cost. Consequently, the methods of this study and Huh et al. (2011) have similar ideas to measure the cost of an edit operation; as an operation changes more geometries of involved line segments, its cost becomes larger. However, the method in this study uses difference between confidence regions of involved line segments to measures changes of geometries, meanwhile the previous method of Huh et al. (2011) used direct changes of geometries as the following equation:

$$c = \alpha(\Delta l/P)^2 + (1-\alpha)(\Delta\theta)^2 \quad (15)$$

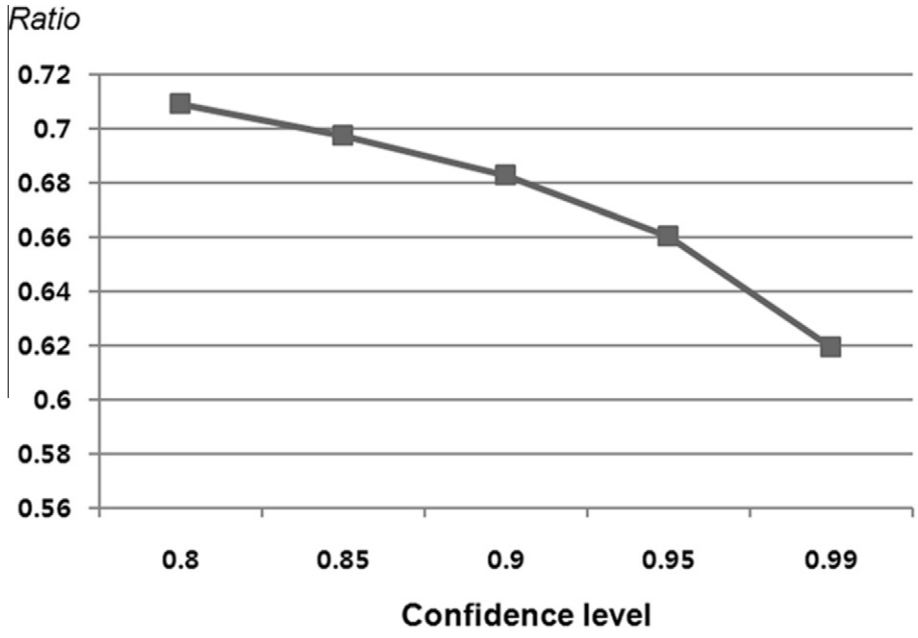
where Δl and $\Delta\theta$ are the changes in the length and angle, respectively, caused by a point edit operation. P is the perimeter of a reference object. Here, the angle change occurs in a limited interval, from 0 to 2π; however, the length change has no such interval. To resolve this problem, the length change is normalized by the perimeter of a reference object.

However, the above weight coefficient was heuristically determined and could present unstable matching results according to



$$ratio = \frac{c_{sub}(a_i \to b_j)}{c_{ins}(\phi \to b_j) + c_{del}(a_i \to \phi)}$$

Fig. 17. Alignment of two polylines with two possible edit operation sequence: (a) two polylines to be aligned, (b) substitution of a_i and b_j, and (c) insertion of b_j then deletion of a_i.



$$ratio = \frac{c_{sub}(a_i \to b_j)}{c_{ins}(\phi \to b_j) + c_{del}(a_i \to \phi)}$$

Fig. 18. Effect of confidence level on cost ratio between two possible edit operation sequences in Fig. 17; σ_X and σ_Y of Table 2 are used for error estimation.

the size and shape of involved object pairs. Therefore, in the case of short line segments with complicated shape in a large object, angle differences primarily dominate overall edit operation costs and matching results as shown in Fig. 19. In the figure, the previous method of Huh et al. (2011) finds smaller number of corresponding point pairs whose discrepancies have similar directions. Conse-

Table 3
Statistical evaluation of the proposed method according to the five confidence levels and comparison with the previously used method of Huh et al. (2011).

	Commission error	Omission error	F-measure
$\gamma = 0.80$	0.096	0.221	0.837
$\gamma = 0.85$	0.093	0.196	0.852
$\gamma = 0.90$	0.068	0.155	0.886
$\gamma = 0.95$	0.077	0.134	0.894
$\gamma = 0.99$	0.110	0.170	0.856
Huh et al. (2011)	0.109	0.178	0.855

quently, the proposed method in this study can find more stable and sufficient corresponding point pairs between complicated contours where corresponding point pairs cannot be well explained by a transformation model.

4. Conclusion

In this paper, a method to detect corresponding point pairs between polygon object pairs with a string matching method based on a confidence region model of a line segment was proposed. We assumed that apparent discrepancies of corresponding point pairs would be aligned by a substitution edit operation and substantial discrepancies by a deletion or insertion operation with which to convert the contour of a target object into that of a reference object. We applied this method for built-up area polygon objects in a cadastral map and a topographical map. Regardless of their different mapping rules and spatial uncertainties of the two maps, the proposed method stably found corresponding point pairs using an F-measure of 0.894 when the confidence level was 0.95.

Unlike to previous the ICP based methods, the proposed method do not need a transformation model to explain discrepancies between corresponding point pairs. It finds the pairs by searching the optimal edit sequence with the minimum total edit cost for deforming the contour of a target object into that of a reference object. Therefore no assumption of a transformation to align contours is required. This can improve matching performance to find corresponding point pairs whose discrepancies cannot be explained by a transformation model. Moreover, the costs of the above point edit

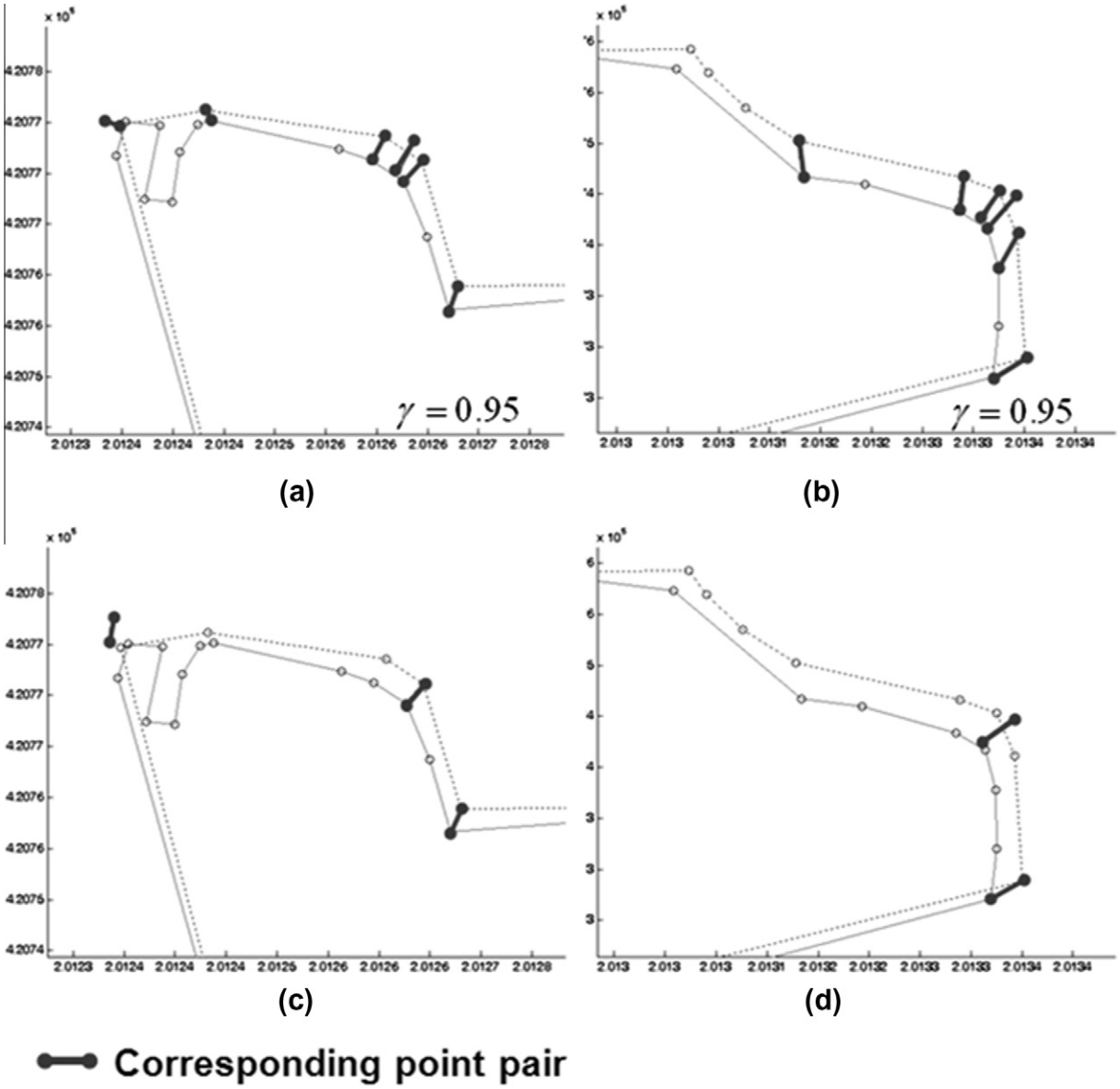


Fig. 19. Comparison of detection results of the proposed method in this study with a confidence level at 0.95 (a and b) and those of Huh et al.(2011) (c and d).

operations and several thresholds were derived from a unified spatial uncertainty model of the involved datasets. Therefore, the proposed method could be more generally applicable for various geospatial datasets and its matching result can be changed according to a confidence level. With a high confidence level, more corresponding point pairs can be found because a high confidence level measures relatively a less cost to a substitution edit operation comparing to an alternative edit operation sequence with deletion and insertion operations to align contours. However, this means that a high confidence level can present false corresponding point pairs which erroneously align contours. Moreover, these false ones concatenately affect their neighboring detections because of the optimization process of the dynamic programming in a string matching method. Thus it needs to find an optimal confidence level considering irregularities of discrepancies within corresponding point pairs of an object pair; a higher confidence level when shape differences between the pairs are large or their shapes are complicated.

Even though the spatial uncertainty of a geospatial dataset is an important factor that causes different geometric representations and positional discrepancies between corresponding object pairs, the previous methods do not fully consider this property. In this regard, our proposed method uses the property in the matching process and can obtain more accurate performances, which is the originality and contribution of this research.

The proposed method can be further applied for various data integration problems which need to find corresponding point pairs. For example, image to image, image to vector and TIN to image or vector integrations can be supported by the proposed method once corresponding polygon pairs are searched with an object matching method. However, according to segmentation methods to raw datasets, there can occur a complicated many-to-many object matching problem. Since the result of the proposed method is affected by the object matching result from which contours to be aligned are determined, it is necessary to develop an effective many-to-many object matching method. Then the proposed method can be applied to diverse data integration issues and would address their discrepancy problems.

Acknowledgements

The work presented in this paper is supported by The Hong Kong Polytechnic University (Project No.: 1-ZV4F, G-U753, G-YK75, G-YJ75, G-YJ75) and Grants from the Ministry of Land, Transport and Maritime Affair, Rep. of Korea (Project No.:11 High-Tech Urban G10.

References

Beard, M., Chrisman, N., 1988. Zipper: a localized approach to edge matching. The American Cartographer 15 (2), 163–172.

Bel Hadj Ali, A., 2001. Positional and shape quality of areal entities in geographic databases: quality information aggregation versus measures classification. In: Proc. ECSQARU'01 Workshop, Toulouse, 19–21 September (on CDROM).

Bunke, H., Buhler, U., 1993. Applications of approximate string matching to 2D shape recognition. Pattern Recognition 26 (12), 1797–1812.

Butenuth, M., Gösseln, G., Tiedge, M., Heipke, C., Lipeck, U., Sester, M., 2007. Integration of heterogeneous gespatial data in a federated database. ISPRS Journal of Photogrammetry and Remote Sensing 62 (5), 328–346.

Chen, S.W., Tung, S.T., Fang, C.Y., 1998. Extended attributed string matching for shape recognition. Computer Vision and Image Understanding 70 (1), 36–50.

Chui, H., Rangarajan, A., 2003. A new point matching algorithm for non-rigid registration. Computer Vision and Image Understanding 89 (2–3), 114–141.

Euzenat, J., Shvaiki, P., 2007. Ontology Matching. Springer, Berlin.

Gahegan, M., Ehlers, M., 2000. A framework for the modelling of uncertainty between remote sensing and geographic information systems. ISPRS Journal of Photogrammetry and Remote Sensing 55 (3), 176–188.

Gösseln, G., Sester, M., 2003. Semantic and geometric integration of geoscientific data sets with ATKIS-applied to geo-objects from geology and soil science. In: Proc. ISPRS Commission IV Joint Workshop 'Challenges in Geospatial Analysis, Integration and Visualization II', Stuttgart, Germany, 8–10, September (on CDROM).

Huh, Y., Yu, K.Y., Heo, J., 2011. Detecting conjugate-point pairs for map alignment between two polygon datasets. Computers, Environment and Urban Systems 35 (3), 250–262.

Kaygin, S., Bulut, M.M., 2002. Shape recognition using attributed string matching with polygon vertices as the primitives. Pattern Recognition Letters 23 (1–3), 287–294.

Kim, J.O., Yu, K.Y., Heo, J., Lee, W.H., 2010. A new method for matching objects in two different geospatial datasets based on the geographic context. Computers and Geosciences 36 (9), 1115–1122.

Kruskal, J.B., 1983. An overview of sequence comparison: time warps, string edits, and macromolecules. SIAM Review 25 (2), 201–237.

Li, L., Goodchild, M.F., 2011. An optimisation model for linear feature matching in geographical data conflation. International Journal of Image and Data Fusion 2 (4), 309–328.

Masuyama, A., 2006. Methods for detecting apparent differences between spatial tessellations at different time points. International Journal of Geographical Information Science 20 (6), 633–648.

Meidow, J., Beder, C., Förstner, W., 2009. Reasoning with uncertain points, straight lines, and straight line segments in 2D. ISPRS Journal of Photogrammetry and Remote Sensing 64 (2), 125–139.

Min, D., Zhilin, L., Xiaoyong, C., 2007. Extended Hausdorff distance for spatial objects in GIS. International Journal of Geographical Information Science 21 (4), 459–475.

Samal, A., Seth, S., Cueto, K., 2004. A feature-based approach to conflation of geospatial source. International Journal of Geographical Information Science 18 (5), 459–489.

Seo, S., O'Hara, C.G., 2009. Quality assessment of linear data. International Journal of Geographical Information Science 23 (12), 1503–1525.

Shi, W.Z., 1998. A generic statistical approach for modeling errors of geometric features in GIS. International Journal of Geographical Information Science 12 (2), 131–143.

Shi, W., Cheung, C.K., Zhu, C., 2003. Modelling error propagation in vector based buffer analysis. International Journal of Geographical Information Science 17 (3), 251–271.

Timpf, S., 1998. Hierarchical Structures in Map Series. Thesis (PhD). Technical University Vienna.

Yuan, S., Tao, C., 1999. Development of conflation components. In: Proc. Geoinformatics'99 Conference, Ann Arbor, USA, 19–21 June, pp. 1–13.