# Modelling Error Propagation for Spatial Consistency

**Min Deng[1, 2], Xiaoyong Chen[1], Wenbao Liu[3], Michiro Kusanagi[1], Huynh N. Phien[1]**

[1]Space Technology Applications and Research Program, Asian Institute of Technology,
PO BOX 4, Klong Luang, Pathumthani 12120, Thailand

[2]Department of Urban and Resource Science, Nanjing University,
22 Road Hankou, Nanjing, 210093, P.R. China

[3]Department of Geosciences, Shandong University of Science and Technology,
223 Street Daizong, Tai'an, 271019, P.R. China

## Abstract:

Currently, there are numerous effective models for managing error propagation in data manipulation and analysis. But the method is still lack in handling spatial consistency. This paper focuses on the algorithms and models of error propagation for spatial consistency, investigating the basic types of spatial data inconsistency and the procedures for correcting them. Since the correction operations often involve changing the position of points of one or more spatial objects, we propose a generalized algorithm (GA) and a related model of error propagation for point groups snapping within a fuzzy tolerance, which are built upon the least square method. Simplified algorithms and models were derived for some special cases with different statistical characteristics. A simple example is provided to demonstrate the potential application of the proposed method.

## 1 Introduction

Data quality in spatial databases has been widely recognized as one of the most critical issues in GIS applications (Veregin, 1999). It is a key dimension of spatial data suitability assessment, including accuracy, precision, uncertainty, compatibility, consistency, completeness, availability and timeliness recorded in the lineage data (Gong and Mu, 2000). In the last two decades, the topic of data quality has attracted much attention in the GIS community. Efforts have been systematically devoted to investigate various aspects in spatial databases (Goodchild and Gopal, 1989; Heuvelink, 1993; Guptill and Morrison, 1995; Leung and Yan, 1998; Shi and Liu, 2000). Among the various problems raised, the issue of spatial consistency has received some special attentions over years (Egenhofer et al., 1994; Gong and Mu, 2000; Servigne et al., 2000), especially for the issue of topological consistency, which is crucial for computer-based spatial analysis, graphical display and the reliability of results from spatial operations, queries and analysis.

In a spatial database, data are often organized by different map layers. Each layer can be obtained from different source which covers the same geographic area with different levels of detail meeting some proper needs of end users. For a single-valued map, spatial inconsistencies often occur from digitalization, e.g. overshoots or undershoots. Consequently, such inconsistencies will further propagate as an overlay operation is done for different map layers. In practical applications, even though each individual map layer may be consistent, it is often unavoidable that inconsistencies will occur in an overlaid map due to different representations for the same geographic objects using different geometric elements, i.e. point, line, and area. In order to integrate the information across different map layers, it is important for certain aspects of the geometry of the objects to get captured consistently throughout the input or overlaid map layers.

In recent years, several efforts have been done on dealing with spatial inconsistency. Kufoniyi et al. (1993) investigated editing topologically-modelled single-valued vector maps in a relational database environment to ensure that the data integrity is preserved, and detailed procedures are formulated in performing editing operations. Servigne et al. (2000) presented a general methodology for spatial consistency, which is based on error survey and classifications. Three

Journal of Geospatial Engineering, Vol. 5, No.2 (December 2003), pp.51-60.

kinds of errors are identified which lead to three kinds of consistency, namely, structural, geometric and topo-semantic consistency. Moreover, some special checking and correcting procedures are available for each of them, but it often involves changing the position of points of spatial objects by operations such as node snapping. In particular, different snapping commands, like 'MOVE' and 'ADJUST', will generate different results for the same group of snapped points. In addition, the accuracy information is unavailable after the editing operations in existing commercial GIS packages. Therefore, a quantitative technique is suggested for assessing the best location for grapping and the accuracy of the resultant objects after rectification. Otherwise, it is difficult to assess errors in GIS products and their use suitability may not be fully qualified.

The remainder of this paper is structured as follows: section 2 discusses the different types of inconsistencies existing at the input and overlaid map, and some procedures for correction are given. Section 3 presents a generalized algorithm for snapping operation and the related error propagation model using the least square method. In section 4, a simple example is provided to demonstrate their potential application, and to make some comparisons with different handling approaches. This paper ends with some conclusions in section 5.

## 2 Approaches to Spatial Consistency Processing

### 2.1 Spatial Inconsistency and Related Approaches

Servigne et al. (2000) classified spatial inconsistencies into three types: structural, geometric, and topological inconsistency. Structural inconsistency comes from the data structures which cannot be implemented to correct a certain data model. In this paper we just consider geometric and topological inconsistencies.

In database, the data model is used to give a representation of the real world. Often, this representation must be simple and should hold the important properties of real objects. For example, a polygon must be closed, otherwise a non-closed one is not consistent geometrically. Likewise, a line must not be self-intersected, otherwise geometrically inconsistent. Apparently geometric inconsistency comes from the geometric part of objects (the shape and position). Most geometric inconsistencies can be reduced to the problem concerning the object points. Therefore all possible operations to handle points must be clearly defined. The following enumerates five basic operations that can be applied to points:

a) adding a new point;
b) deleting a point;
c) merging two points;
d) projecting a point on a segment;
e) modifying the coordinates of an existing point.

Correspondingly, the correction can be divided into three parts:

a) computation of the best location of the point;
b) projection;
c) deletion of the useless points.

A topological inconsistency is defined as a forbidden topological relation between two objects. It is related to the meaning of the real objects represented in the database and to the topological relations associated with other objects. Thus, the way correcting such inconsistency will change the topological relation between those objects. It can be performed through the following changes:

a) objects modification including moving and reshaping the objects;
b) deleting one object; and
c) object splitting (creating a new object).

### 2.2 Basic Procedures for Inconsistency Correcting

For an individual map layer, the geometric data consists of a set of line segments and polylines. The errors in the observations may cause violations on geometric relations among them. Figure 1 shows some types of inconsistent data marked by dashed circles. Apparently, it is difficult to get the desired topological relations without processing these inconsistent data. Take a look at them:

a) Two line segments intersect at exactly one point (case a). This is the most common situation and is handled by the following rules: first the vertices of both involved line segments $l_i$ and $l_j$ are stored, then $l_i$ and $l_j$ are erased. A new vertex is created at the intersection point. By using the stored vertices and the new vertex, four new line segments are generated.

Before checking the other types of inconsistencies, we define first what a connected degree means. For any point $p_i$ in a line, its connected degree can be determined by two steps: first, draw a circle of radius ε (a small number) around point $p_i$, then count the number of intersection points of the circle with the line, i.e. the connected degree of the point $p_i$, noted as $Dn(p_i)$. For each vertex, its connected degree can be regarded as the number of line segments ending in it.

b) One of the endpoints in line segment $l_i$, $p_i$, is very close (within fuzzy tolerance) to another line segment $l_j$, see cases d and e in Figure 1. This situation can be handled by the following rule: the endpoint $p_i$ in line segment $l_i$ splits line segment $l_j$. The original line segment $l_i$ is eliminated and two new line segments are generated. They share the same endpoint.

c) A group of nodes or vertices with $Dn(p_i) \geq 1$ within the fuzzy tolerance can be joined into a common node, see cases b and c.

There is a special case for (c). Consider a group of vertices with $Dn(p_i) = 2$, which is closer than the given fuzzy tolerance, it is necessary to determine whether these vertices are distinct or in the same position (case f). In this case, the user's intervention may be needed.

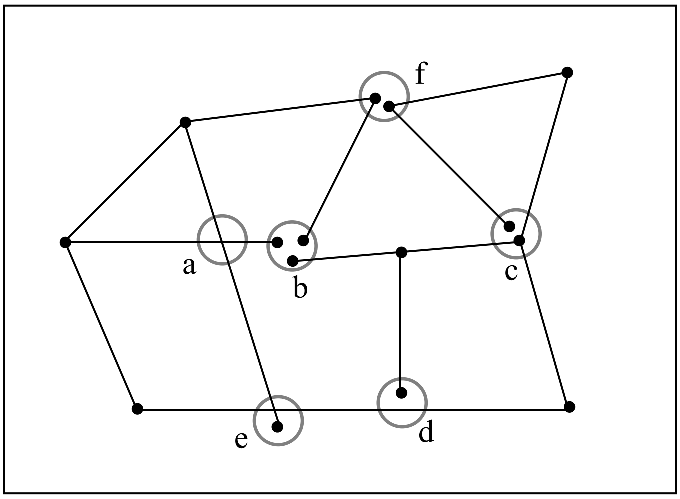


**Figure 1. An illustration of inconsistent geometric data**

In the following discussion, the issue of inconsistent lines will be considered.

d) For two line segments which may be in the same location, there are three kinds of inconsistent relations due to positional errors or uncertainties (see Figure 2). Hence, we can define such two segments that fall inside the buffer of each other within a given fuzzy tolerance to be completely overlapped. Under this circumstance, a simple solution is to remove one of two line segments together with none (case a), one (case b) or two vertices (case c). For case b and c, a generalized approach is described as follows: first capture the inconsistent point pairs of both involved line segments, then create and store the new points (nodes or vertices); afterwards erase the original segments; then generate a new segment.

An extension of (d) is to consider the situation of inconsistent polylines. It can be classified into two cases, as shown in Figure 3. Both of them need user's intervention for consistency correction

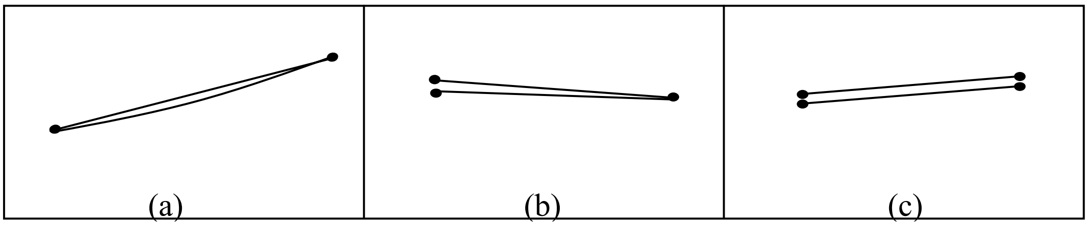


**Figure 2. Three cases of inconsistent line segments**

.

e) Two polylines $l_i$ and $l_j$ completely overlapped within a given fuzzy tolerance, and each vertex of $l_i$ can find a corresponding vertex to match in $l_j$. In case a, the inconsistent point pairs within the fuzzy tolerance are marked with dashed circles. In this case, if $l_i$ and $l_j$ are determined to represent the same geographic object, the possible approaches used for consistency processing are similar to those described in (d).

f) Two polylines $l_i$ and $l_j$ completely overlap within a given fuzzy tolerance but, each vertex of $l_i$ does not always find a comparable vertex in $l_j$ within the fuzzy tolerance (case b). In practical applications, this case often occurs in an overlaid map. Those comparable polylines within the fuzzy tolerance may be of different number of line segments and input points. In order to make the number of segments of the two comparing lines equal, and to ensure their composite points comparable, object normalization is proposed here. This is performed by projecting all the input points of each polyline unto their corresponding one, thus dividing the original polylines into the same number of line segments and points. It should be mentioned that the projection of the vertex will be cancelled if the projected point falls within the fuzzy tolerance from the existing points, such unnecessary segmentation can be avoided. Further processing can apply the methods described in (d).

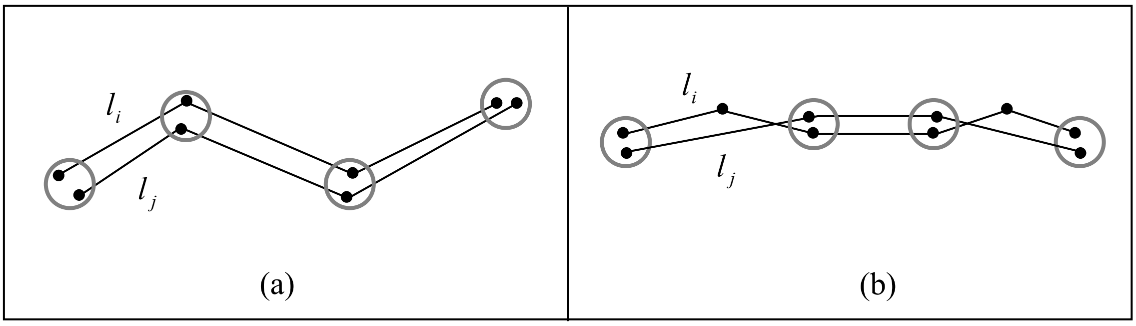


**Figure 3. Two possible cases for inconsistent polylines**

The critical issue in the above investigations is to locate the created points. In the following section, a generalized method is presented for computing the coordinates of the new points, and a related model for accuracy assessment is provided.

# 3 Algorithm and Related Error Propagation Models

## 3.1 Generalized Algorithm and Error Models

*Let* $\boldsymbol{z}_i = (x_i, y_i)^{\mathrm{T}}$ *be the* $i$*-th point from a group of points within a specified fuzzy tolerance, with the covariance matrix of* $\boldsymbol{\Gamma}_{ii} = \begin{bmatrix} \sigma_{x_i}^2 & \sigma_{x_i y_i} \\ \sigma_{y_i x_i} & \sigma_{y_i}^2 \end{bmatrix}$. *If it satisfies* $\sigma_{x_i y_i} = \sigma_{y_i x_i} \neq 0, (i = 1, 2, \cdots, n)$ *, then* $z_i$ *is an auto-correlation vector. Further, let* $z^* = (x_1, y_1, x_2, y_2, \cdots, x_n, y_n)^{\mathrm{T}}$ *, then the covariance matrix of the* $n$*-point snapping group is expressed as:*

$$\boldsymbol{\Gamma}_{z^*z^*} = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1y_1} & \sigma_{x_1x_2} & \sigma_{x_1y_2} & \cdots & \sigma_{x_1x_n} & \sigma_{x_1y_n} \\ \sigma_{y_1x_1} & \sigma_{y_1}^2 & \sigma_{y_1x_2} & \sigma_{y_1y_2} & \cdots & \sigma_{y_1x_n} & \sigma_{y_1y_n} \\ \sigma_{x_2x_1} & \sigma_{x_2y_1} & \sigma_{x_2}^2 & \sigma_{x_2y_2} & \cdots & \sigma_{x_2x_n} & \sigma_{x_2y_n} \\ \sigma_{y_2x_1} & \sigma_{y_2y_1} & \sigma_{y_2x_2} & \sigma_{y_2}^2 & \cdots & \sigma_{y_2x_n} & \sigma_{y_2y_n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \sigma_{x_nx_1} & \sigma_{x_ny_1} & \sigma_{x_nx_2} & \sigma_{x_ny_2} & \cdots & \sigma_{x_n}^2 & \sigma_{x_ny_n} \\ \sigma_{y_nx_1} & \sigma_{y_ny_1} & \sigma_{y_nx_2} & \sigma_{y_ny_2} & \cdots & \sigma_{y_nx_n} & \sigma_{y_n}^2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Gamma}_{11} & \boldsymbol{\Gamma}_{12} & \cdots & \boldsymbol{\Gamma}_{1n} \\ \boldsymbol{\Gamma}_{21} & \boldsymbol{\Gamma}_{22} & \cdots & \boldsymbol{\Gamma}_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{\Gamma}_{n1} & \boldsymbol{\Gamma}_{n2} & \cdots & \boldsymbol{\Gamma}_{nn} \end{bmatrix} \qquad (1)$$

where $\boldsymbol{\Gamma}_{ij} = \begin{bmatrix} \sigma_{x_ix_j} & \sigma_{x_iy_j} \\ \sigma_{y_ix_j} & \sigma_{y_iy_j} \end{bmatrix}$. If $\boldsymbol{\Gamma}_{ij} \neq 0\,(i \neq j)$ , then $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ $(i \neq j)$ are a pair of cross-correlation vectors. If all of $\boldsymbol{\Gamma}_{ij}\,(1 \le i \neq j \le n)$ satisfy $\boldsymbol{\Gamma}_{ij} \neq 0$ , then $\boldsymbol{z}^*$ is called a full cross-correlation vector; otherwise, if there exist at least one $\boldsymbol{\Gamma}_{ij}\,(i \neq j)$, satisfying $\boldsymbol{\Gamma}_{ij} \neq 0$ , then $\boldsymbol{z}^*$ is called a partial cross-correlation vector. Let $\boldsymbol{\Psi}$ be the inverse matrix of $\boldsymbol{\Gamma}_{z^*z^*}$ , i.e.

$$\boldsymbol{\Psi} = \boldsymbol{\Gamma}_{z^*z^*}^{-1} = \begin{bmatrix} \boldsymbol{\Phi}_{11} & \boldsymbol{\Phi}_{12} & \cdots & \boldsymbol{\Phi}_{1n} \\ \boldsymbol{\Phi}_{21} & \boldsymbol{\Phi}_{22} & \cdots & \boldsymbol{\Phi}_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{\Phi}_{n1} & \boldsymbol{\Phi}_{n2} & \cdots & \boldsymbol{\Phi}_{nn} \end{bmatrix}$$

where $\boldsymbol{\Phi}_{ij}$ is a $2 \times 2$ partitioned matrix. Assuming that $\boldsymbol{z} = (x, y)^{\mathrm{T}}$ is the optimally estimated vector of the coordinates of the new point created by snapping operation, with covariance matrix of $\boldsymbol{\Gamma} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}$. Since the algorithm of node snapping of a group of points within fuzzy tolerance is essentially an adjustment algorithm of direct observations on the basis of the least square principle in geodesy (Mikhail, 1976), we can derive the following generalized formula for estimating the coordinates of the new point

$$\boldsymbol{z} = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{\Phi}_{ij}\right)^{-1} \sum_{i=1}^{n}\sum_{j=1}^{n} (\boldsymbol{\Phi}_{ij} \boldsymbol{z}_j) \qquad (2)$$

Moreover, we can represent the universal error propagation model of coordinates through the snapping operation from $\boldsymbol{z}^*$ to $\boldsymbol{z}$

$$\boldsymbol{\Gamma} = (\sum_{i=1}^{n}\sum_{j=1}^{n}\boldsymbol{\Phi}_{ij})^{-1} \tag{3}$$

Apparently, equations (2) and (3) can be used for multiple data sources with varying resolutions.

### 3.2 Algorithm and Error model for Pure Auto-correlation

In practice, it is often assumed that the points in the snapping group from $n$ data layers are independent of each other, or that the cross correlation between them can be ignored due to the lack of the information on their correlations, i.e., $\boldsymbol{\Gamma}_{ij} = 0\ (1 \le i \ne j \le n)$. Further, if $\exists\ i$, it satisfies $\sigma_{x_i y_i} = \sigma_{y_i x_i} \ne 0$, then $\boldsymbol{z}^*$ is called a pure auto-correlation vector. At the same time, equation (1) becomes a partitioned diagonal matrix, i.e. $\boldsymbol{\Gamma}_{z^*z^*} = diag\{\boldsymbol{\Gamma}_{11}\boldsymbol{\Gamma}_{22}\cdots\boldsymbol{\Gamma}_{nn}\}$, while $\boldsymbol{\Psi} = \boldsymbol{\Gamma}_{z^*z^*}^{-1} = diag\{\boldsymbol{\Gamma}_{11}^{-1}\ \boldsymbol{\Gamma}_{22}^{-1}\cdots\boldsymbol{\Gamma}_{nn}^{-1}\}$. As a result, equations (2) and (3) are reduced to

$$\boldsymbol{z} = (\sum_{i=1}^{n}\boldsymbol{\Gamma}_{ii}^{-1})^{-1}\sum_{i=1}^{n}(\boldsymbol{\Gamma}_{ii}^{-1}\boldsymbol{z}_i) \tag{4}$$

$$\boldsymbol{\Gamma} = (\sum_{i=1}^{n}\boldsymbol{\Gamma}_{ii}^{-1})^{-1} \tag{5}$$

Under this circumstance, a special case exists where the points in the snapping group are from data sources of equal accuracy. Therefore we have: $\boldsymbol{\Gamma}_{ii} \equiv \boldsymbol{\Gamma}_{00} = \begin{bmatrix} \sigma_{x_0}^2 & \sigma_{x_0 y_0} \\ \sigma_{y_0 x_0} & \sigma_{y_0}^2 \end{bmatrix}, (i = 1,2,\cdots,n)$. One example is topological cleaning of digitised data from one map source. Furthermore, equation (4) can be represented as:

$$z = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n} x_i \\ \frac{1}{n}\sum_{i=1}^{n} y_i \end{pmatrix} \tag{6}$$

Accordingly, equation (5) can be represented as:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \sigma_x{}^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y{}^2 \end{bmatrix} = \begin{bmatrix} \sigma_{x_0}{}^2 / n & \sigma_{x_0 y_0} / n \\ \sigma_{x_0 y_0} / n & \sigma_{y_0}{}^2 / n \end{bmatrix} \tag{7}$$

As shown in equations (6) and (7), when the accuracy of the points in the snapping group is equal, the estimated coordinates (x, y) of the created point, are independent of the accuracy information of all points in the snapping group, and can be calculated separately by equation (6); their covariances, $\sigma_x^2$ and $\sigma_y^2$, are with the correlation between $x_i$ and $y_i$ of the snapping points. Here only $\sigma_{xy}$, which is the auto-covariance between $x$ and $y$ of the new point, is dependent on $\sigma_{x_0 y_0}$. It is pointed out that equation (6) is also the expression adopted by 'ADJUST' algorithm.

### 3.3 Practical Algorithm and Error Models

In practice, it is common that there is no cross-correlation among the snapping points and the accuracy of the coordinate components, and are equal with each other, that is,

$\boldsymbol{\Gamma}_{ij} = 0 \ (1 \le i \ne j \le n)$ and $\sigma_{x_i}^2 = \sigma_{y_i}^2 = \sigma_i^2, (i = 1,2,\cdots,n)$. By definition of correlation coefficient, i.e., $\rho_i = \sigma_{x_i y_i} / \sigma_{x_i}\sigma_{y_i}$, we have

$$\boldsymbol{\Gamma}_{ii} = \begin{bmatrix} \sigma_{x_i}^2 & \sigma_{x_i y_i} \\ \sigma_{y_i x_i} & \sigma_{y_i}^2 \end{bmatrix} = \sigma_i^2 \begin{bmatrix} 1 & \rho_i \\ \rho_i & 1 \end{bmatrix} \tag{8}$$

Thus equation (4) becomes

$$\boldsymbol{z} = \left( \sum_{i=1}^{n} \frac{1}{\sigma_i^2 (1-\rho_i^2)} \begin{bmatrix} 1 & -\rho_i \\ -\rho_i & 1 \end{bmatrix} \right)^{-1} \sum_{i=1}^{n} \frac{1}{\sigma_i^2 (1-\rho_i^2)} \begin{pmatrix} x_i - \rho_i y_i \\ y_i - \rho_i x_i \end{pmatrix} \tag{9}$$

and equation (5) can be reduced to this error model:

$$\boldsymbol{\Gamma} = \left( \sum_{i=1}^{n} \frac{1}{\sigma_i^2 (1-\rho_i^2)} \begin{bmatrix} 1 & -\rho_i \\ -\rho_i & 1 \end{bmatrix} \right)^{-1} \tag{10}$$

It has been shown from equations (9) and (10) that in the case of a snapping point group with unequal accuracy, both the estimated coordinates ($x$, $y$) and their covariances $\sigma_x^2$ and $\sigma_y^2$, are dependent on the auto-correlation coefficient, $\rho_i$, of each point in the groping group, such that reliable estimates of their coordinates and covariances can be only obtained by applying simultaneously equations (2) and (3). In the following discussion, some special cases will be investigated with certain assumptions:

**Assumption a)** $\rho_i \equiv \rho_0, (i = 1, 2, \cdots, n)$

This assumption means that the auto-correlation of each point in the snapping group can be completely identical. In this case, equation (9) can be reduced to

$$\begin{cases} x = \sum_{i=1}^{n} (\sigma_i^{-2} x_i) / \sum_{i=1}^{n} \sigma_i^{-2} \\ y = \sum_{i=1}^{n} (\sigma_i^{-2} y_i) / \sum_{i=1}^{n} \sigma_i^{-2} \end{cases} \tag{11}$$

and equation (10) can be simplified into

$$\begin{cases} \sigma_x = \sigma_y = \pm 1 / \sqrt{\sum_{i=1}^{n} \sigma_i^{-2}} \\ \sigma_{xy} = \sigma_{yx} = \rho_0 / \sum_{i=1}^{n} \sigma_i^{-2} \end{cases} \tag{12}$$

It can be seen from equations (11) and (12) that both the estimated coordinates and their covariances are independent of the auto-correlation coefficient, $\rho_0$, of each point in the snapping group. However, the auto-covariance, $\sigma_{xy}$, of the new coordinates is dependent on $\rho_0$.

**Assumption b)** $\rho_i \equiv 0$, $(i = 1,2,\cdots,n)$

This assumption shows that there is no auto-correlation for each point in the snapping group. In this case, equation (11) remains unchanged, while equation (12) can be reduced to

$$\begin{cases} \sigma_x = \sigma_y = \pm 1 / \sqrt{\sum_{i=1}^{n} \sigma_i^{-2}} \\ \sigma_{xy} = \sigma_{yx} = 0 \end{cases} \tag{13}$$

**Assumption c)** $\rho_i \equiv 0$ and $\sigma_i^2 \equiv \sigma_0^2$, $(i = 1, 2, \cdots, n)$

This assumption means that there is no auto-correlation for each point in the snapping group, and all points are of equal accuracy. In this case, equation (11) can be reduced to equation (6), while equation (13) will be reduced to

$$\begin{cases} \sigma_x = \sigma_y = \sigma_0 / \sqrt{n} \\ \sigma_{xy} = \sigma_{yx} = 0 \end{cases} \tag{14}$$

Here, equation (14) can also be derived directly from equation (7) based on the given assumption above.

## 4 Numerical Example

Figure 4 shows that Poly#1 and Poly#2 are two adjacent polygons from two independent data layers. The accuracy of coordinates in this example is designed based on a land use map at a scale of 1:24,000 (Hord, 1976). In Table 1, only the coordinates and accuracy information of inconsistent neighboring boundaries are listed, where the identification number is encoded after object normalization (see in Figure 5).

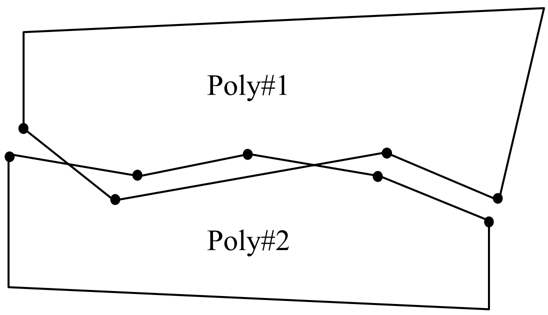


**Figure 4. Schematics of inconsistent data**

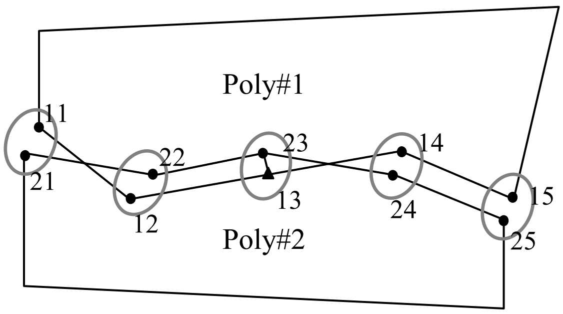


**Figure 5. Illustration of normalization for inconsistent lines**

In order to make a comparison on the approach presented in this paper, three different snapping algorithms are taken to deal with the inconsistent lines in Figure 4. These available methods are 'MOVE', 'ADJUST', and General Algorithm (abbreviated as 'GA').

The 'MOVE' algorithm snaps the points of lower accuracy to the point of high accuracy. In this case, the points of Poly#1 in Figure 5 are captured to the corresponding points of Poly#2, and the accuracy of the new points are equal to the points of Poly#2. The results are listed in Table 2. For the 'ADJUST' algorithm, the coordinates of the new points are taken as the average values of the snapped points. In this example, they can be simply represented as:

$$\begin{cases} x_{1k}^* = \frac{1}{2}(x_{1k} + x_{2k}) \\ y_{1k}^* = \frac{1}{2}(y_{1k} + y_{2k}) \end{cases}, \quad (k=1,2,\cdots,5) \tag{15}$$

Further, there has

$$\begin{cases} \sigma_{x_{1k}^*} = \dfrac{1}{2}\sqrt{\sigma_{x_{1k}}^2 + \sigma_{x_{2k}}^2} \\ \sigma_{y_{1k}^*} = \dfrac{1}{2}\sqrt{\sigma_{y_{1k}}^2 + \sigma_{y_{2k}}^2} \end{cases}, \quad (k=1,2,\cdots,5) \qquad (16)$$

**Table 1. Location and accuracy data of Poly#1 and Poly#2 after pre-processing**

| Poly# | No. | $x/m$ | $y/m$ | $\sigma_x/m$ | $\sigma_y/m$ | $\rho_{xy}$ |
|---|---|---|---|---|---|---|
| 1 | 11 | 768657.60 | 2935308.40 | 20.00 | 20.00 | 0 |
| | 12 | 769083.56 | 2935029.68 | 20.00 | 20.00 | 0 |
| | 13 | 769552.58 | 2935114.66 | 14.46 | 14.46 | 0 |
| | 14 | 770125.82 | 2935218.52 | 20.00 | 20.00 | 0 |
| | 15 | 770637.93 | 2935011.80 | 20.00 | 20.00 | 0 |
| 2 | 21 | 768651.84 | 2935294.78 | 10.00 | 10.00 | 0 |
| | 22 | 769098.25 | 2935038.13 | 10.00 | 10.00 | 0 |
| | 23 | 769550.22 | 2935127.73 | 10.00 | 10.00 | 0 |
| | 24 | 770118.16 | 2935207.89 | 10.00 | 10.00 | 0 |
| | 25 | 770634.65 | 2935002.48 | 10.00 | 10.00 | 0 |

**Table 2. Location and accuracy data after consistency correcting**

| Operation | No. | $x/m$ | $y/m$ | $\sigma_x/m$ | $\sigma_y/m$ | $\rho_{xy}$ |
|---|---|---|---|---|---|---|
| *MOVE* | 11* | 768651.84 | 2935294.78 | 10.00 | 10.00 | 0 |
| | 12* | 769098.25 | 2935038.13 | 10.00 | 10.00 | 0 |
| | 13* | 769550.22 | 2935127.73 | 10.00 | 10.00 | 0 |
| | 14* | 770118.16 | 2935207.89 | 10.00 | 10.00 | 0 |
| | 15* | 770634.65 | 2935002.48 | 10.00 | 10.00 | 0 |
| *ADJUST* | 11* | 768654.72 | 2935301.59 | 11.18 | 11.18 | 0 |
| | 12* | 769090.91 | 2935033.91 | 11.18 | 11.18 | 0 |
| | 13* | 769551.40 | 2935121.40 | 8.79 | 8.79 | 0 |
| | 14* | 770121.99 | 2935213.20 | 11.18 | 11.18 | 0 |
| | 15* | 770636.29 | 2935007.14 | 11.18 | 11.18 | 0 |
| *GA* | 11* | 768652.99 | 2935297.50 | 8.94 | 8.94 | 0 |
| | 12* | 769095.31 | 2935036.44 | 8.94 | 8.94 | 0 |
| | 13* | 769550.69 | 2935125.12 | 8.22 | 8.22 | 0 |
| | 14* | 770119.69 | 2935210.02 | 8.94 | 8.94 | 0 |
| | 15* | 770635.30 | 2935004.34 | 8.94 | 8.94 | 0 |

According to above equations, we can compute all of the snapped point pairs, and the results are listed in Table 2. While in this example general algorithm can be reduced to equations (12) and (13), having:

$$\begin{cases} x_{1k}^* = (\sigma_{x_{k1}}^{-2} + \sigma_{x_{2k}}^{-2})^{-1}(\sigma_{x_{1k}}^{-2} x_{1k} + \sigma_{x_{2k}}^{-2} x_{2k}) \\ y_{1k}^* = (\sigma_{y_{1k}}^{-2} + \sigma_{y_{2k}}^{-2})^{-1}(\sigma_{y_{1k}}^{-2} y_{1k} + \sigma_{y_{2k}}^{-2} y_{2k}) \end{cases}, \quad (k=1,2,\cdots,5) \quad (17)$$

and,

$$\begin{cases} \sigma_{x^*_{1k}} = \pm(\sigma^{-2}_{x_{1k}} + \sigma^{-2}_{x_{2k}})^{-1} \\ \sigma_{y^*_{1k}} = \pm(\sigma^{-2}_{y_{1k}} + \sigma^{-2}_{y_{2k}})^{-1} \end{cases}, \ (k=1,2,\cdots,5) \qquad (18)$$

The results are also given in Table 2. The results obtained by the 'MOVE', 'ADJUST' and 'GA' algorithms are completely different. For the 'MOVE' method, only location and accuracy of the data with better quality are considered, ignoring related information of other data sources. The 'ADJUST' method considers location and accuracy of all data sources equally. The 'GA' method may be regarded as a kind of a weighted average value by taking into account the accuracies of all data sources.

## 5 Conclusion

In this paper, the basic approaches to spatial data consistency processing are investigated, which to a certain degree extends the existing snapping functions used in the commercial GIS packages. In particular a generalized algorithm is proposed for the calculation of the best location of new point created by snapping a point group within a fuzzy tolerance. The related error propagation model is provided for accuracy assessment, which is often lack in existing GIS packages.

The algorithm based on equation (6) for the ADJUST command in ArcInfo is only a special case of the new method based on equations (2) and (3) presented in this paper. The ADJUST command neither deals with the node snapping of a group of points with non-equal accuracy nor provides accuracy information of snapping results with equal accuracy. In addition, the new method can be applied to a general circumstance, whether with equal or non-equal accuracy, and whether dependent or independent snapping point group.

## References

Egenhofer, M.J., Clementini, E. and Di Felice, P., 1994. Evaluating inconsistencies among multiple representations. *6th Proceedings of the International Symposium on Spatial Data Handling*, Edinburgh, Scotland, 901-920.

Gong, P. and Mu, L., 2000. Error detection through consistency checking. *Geographic Information Science*, **6**(2): 188-193.

Goodchild, M.F. and Gopal, S. 1989. *Accuracy of Spatial Databases*. Tayloy and Francis, London.

Guptill, S.C. and Morrison, J.L., 1995. *Elements of Spatial Data Quality*. Elsevier Science, New York, 1-12.

Harvey, F., 1994. Defining unmovable nodes/segments as part of vector overlay, in: *Advances in GIS Research*. Taylor and Francis, London, 159-176.

Hord, R., 1976. Land-use map accuracy criteria. *Photogrammetric Engineering and Remote Sensing*, **42**(5): 671-677.

Heuvelink, G.B.M., 1993. *Error propagation in quantitative spatial modelling and its application in GIS*. PhD thesis, Netherlands.

Jones, C.B., 1997. *Geographical Information Systems and Computer Cartography*. Longman, Harlow, 121-131.

Kufoniyi, O., Molenaar, M. and Bouloucos, T. 1993. Topological editing of relationally structured single-valued vector maps. *ITC Journal*, No.4, 332-339.

Leung Y. and J.P. Yan, 1998. A locational error model for spatial features. *International Journal of Geographical Information Sciences*, **12**(6): 607-620.

Mikhail, E.M., 1976. Observations and Least Squares. A Dun-Donnelley Publisher, New York.

Servigne, S. Ubeda, T. and Puricelli, A. 2000. A methodology for spatial consistency improvement of geographic databases. *Geoinformatica*, **4**(1): 7-34

Shi, W.Z. and Liu W.B. 2000. A stochastic process-based model for positional error of line segments in GIS. *International Journal of Geographic Information Sciences*, **14** (1), 51-66.

Veregin, H., 1999. Data quality parameters. In: Longley P A. et al. (ed.), *Geographical Information System*, 2nd Eds. Vol 1, John Wiley & Sons, New York, 177~189.