The Cartographic Journal Vol. 43 No. 1 pp. 27–44 March 2006

REFEREED PAPER

Performance Evaluation of Line Simplification Algorithms for Vector Generalization

*Wenzhong Shi and ChuiKwan Cheung**

Advanced Research Center for Spatial Information Technology, Department of Land Surveying and Geo-Informatics, Hong Kong Polytechnic University, Hunghom, Kowloon, Hong Kong.
*Email: lswzshi@polyu.edu.hk; tracy.cheung@alumni.polyu.edu.hk

Many studies of line simplification methods have been developed; however, an evaluation of these methods is still an open issue. This paper aims to evaluate a diversity of automatic line simplification algorithms in terms of positional accuracy and processing time. Past research studies for the performance evaluation were centred on measuring the location difference between a line to be simplified and its simplified version. However, the original line contains positional uncertainty. This paper evaluates performance of the line simplification algorithms using two comprehensive measures of positional accuracy of the simplified line. These two measures include one displacement measure and one shape distortion measure, both of which are able to consider (a) the displacement between the original line and its simplified version, and (b) positional uncertainty of the original line.

BACKGROUND

Vector generalization is indispensable for making a paper map or a digital map. This process enhances the display quality of a map at a smaller scale, analyses data within the map with different degrees of detail, reduces data storage and generates a map from other maps with different degrees of detail (João, 1998). The first step of vector generalization is the selection of a geographical object that will be used to generate the one with less detail. The selected geographical object can be characterized by points, lines, curves or surfaces. Its geometric form determines what generalization type (such as line or polygon generalization) is used. Generalization of a line aims to retain its appearance even when the number of points on the line is reduced, or to enhance the appearance when knowledge of what the line should look like is used. Polygon generalization generally represents an area with fewer points or combines adjacent areas.

Of the research on vector generalization, most work has concentrated on line generalization (João, 1998). One reason is that automated generalization of line features is less intricate, compared with those involving other features (Weibel, 1986). Another reason is that most features found on a typical medium-scale topographic map are lines (Müller, 1991). This paper focuses on line generalization to contribute further to such research. Polygon generalization is not a primary topic of this paper and is discussed elsewhere (DeLucia and Black, 1987; Jones *et al.*, 1995; Cottingham, 1997; Galanda, 2001, 2003; Galanda and Weibel, 2003).

One approach to line generalization is line simplification, which is an automation process in GIS for removing data redundancy in the digital dataset. A diversity of line simplification algorithms has been developed including the shortest distance between consecutive points along the original line (from which its corresponding line with less detail, the simplified line, is compiled) and the angular change, and the positional difference between the original line and its simplified version.

Line simplification introduces positional error in the resultant map and may cause topological errors. Positional error introduced through a line simplification algorithm are normally compared with positional error resulting from the Douglas–Peucker algorithm, a global routine that interactively selects critical points by considering the entire line (White, 1985; McMaster, 1987; Müller, 1987). After comparing shape and displacement measures, as well as processing time for some line simplification algorithms, it was concluded that the Douglas–Peucker algorithm was the most effective to preserve the shape of the line and the most accurate in terms of position. The shape and displacement measures used in these studies assess shape, and positional differences between the original line and its simplified version. Actually, these differences are associated with the line simplification algorithm itself. In line simplification, positional accuracy of the simplified line is potentially affected by positional accuracy of the original line (Cheung and Shi, 2004) and so there is a need to re-evaluate performance of different line simplification algorithms.

DOI: 10.1179/000870406X93490

Topological errors that might be created during line simplification can be line-crossing, coincident lines and collapsed zero-length lines. Lee (2004) proposed a topological error detection routine for locating the three types of topological errors on the simplified line. If the simplified line is with any one of the topological errors, the original line will be simplified again using a reduced tolerance until no more topological errors are found. An alternative approach to topological errors involves using a technique for line displacement with specific topological constraints, which can overcome topology-related conflicts. Techniques for line displacements are mainly of two types (Burghardt, 2005): energy minimization methods such as snakes (Burghardt and Meier, 1997) and beams (Bader, 2001), and least square adjustment (Harrie and Sarjakoski, 2000; Sester, 2000).

The purpose of this paper is to evaluate automatic line simplification algorithms in terms of positional error on the simplified line. The paper is organized into six sections. The first two give an introduction of performance evaluation of line simplification and define the scope of this paper. A set of line simplification algorithms involved in our evaluation then follows. Performance measurement of the line simplification algorithm is then presented followed by an evaluation based on a real data set of different line complexities and results and conclusions.

PROBLEM DEFINITION

In this study, coastlines are simplified by using different line simplification algorithms. Since there is a diversity of line simplification algorithms in automatic generalization, this study has chosen at least one algorithm from each of the five line simplification algorithm categories defined by McMaster (1987), namely independent point algorithms, local processing routines, unconstrained extended local processing routines, constrained extended local processing routines and global routines. Furthermore, the selected line simplification algorithms are compared according to processing time, displacement and shape distortion. Measures of displacement and shape distortion used in this paper are more comprehensive than other existing displacement measures for line simplification due to a consideration of both positional uncertainty of the original line and locational difference between the original line and its simplified version.

LINE SIMPLIFICATION ALGORITHMS

The line simplification algorithms to be evaluated in this paper include the *n*th point routine, the routine of distance between points and the perpendicular distance routine, the Reumann-Witkam routine, the sleeve-fitting polyline simplification algorithm (also called the Zhao–Saalfeld algorithm), the Opheim simplification algorithm, the Lang simplification algorithm, the Douglas–Peucker simplification algorithm and the Visvalingam–Whyatt algorithm. They are point-selection/rejection algorithms with the virtue of not relocating original points.

Independent Point Algorithms

Independent point algorithms are very simple and rapidly determine which points of a line should be retained. They are not under constraint of any mathematical relationship with neighbouring points of the line. Two examples of this category are the *n*th point routine and the routine of random-selection of points. In these two routines, for every fixed number of consecutive points along the line, the *n*th point and one random point among a set of these consecutive points are retained, respectively (Figure 1).

In this paper, the routine of random selection of points will not be compared with other simplification algorithms. This routine will generate a simplified line randomly. It may result in a different version of the simplified line even when the number of consecutive points in each point set is kept unchanged. Comparing a simplified line derived from this algorithm with that derived from any other line simplification algorithm in terms of their positional accuracy could obtain a different conclusion under the same conditions such that (a) the line to be simplified is the same and (b) the values of parameters for the line simplification algorithms are fixed. Therefore, the routine of random-selection of points is not further discussed here.

Local Processing Routines

The second category, local processing routines, regards a relationship between every two or three consecutive original points. Two examples of the relation are (a) the distance between the two consecutive points and (b) the perpendicular distance from a line connecting two points to an intermediate point, which should not be smaller than their individual tolerance bandwidths (a user-supplied minimum distance or angular change). Points within the bandwidth are eliminated, whereas points exceeding the bandwidth are retained. These tolerance routines are called *the routine of distance between points* and *the perpendicular distance routine*, respectively (Figure 2).

Unconstrained Extended Local Processing

The third category is unconstrained extended local processing routines that evaluate relations over sections of the line. An example of this category is *the Reumann-Witkam routine* (Reumann-Witkam, 1974) (Figure 3). In this routine, the line is divided into sections by using a strip. The strip is shifted over the line into the direction of its initial tangent, until the strip hits the line. Points within this strip compose one section of the line. The last point within the strip is the initial point of the remaining part of the line. The strip then has a direction the same as the initial tangent of this remaining part. The whole process is repeated until the strip contains the end point of the line. The initial point of each section of the line and the end point of the line are retained, and those points between them are removed.

The sleeve-fitting polyline simplification algorithm proposed by Zhao and Saalfeld (1997) is another unconstrained extended local processing routine evaluated in this paper. This algorithm is similar to the Reumann-Witkam routine because the original line is divided into sections by using a rectangle (or called the sleeve in Zhao and Saalfeld's model). Figure 4 shows sleeves of a user-defined width.

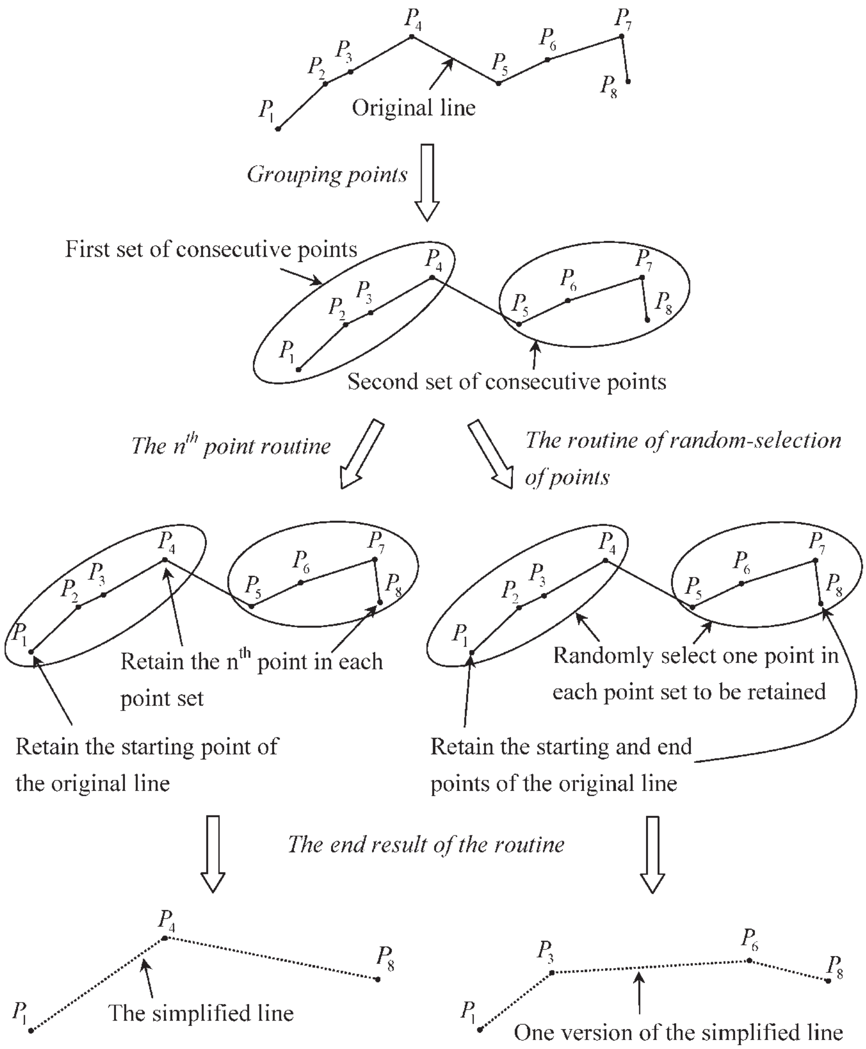

Figure 1. The *n*th point routine and the routine of random-selection of points

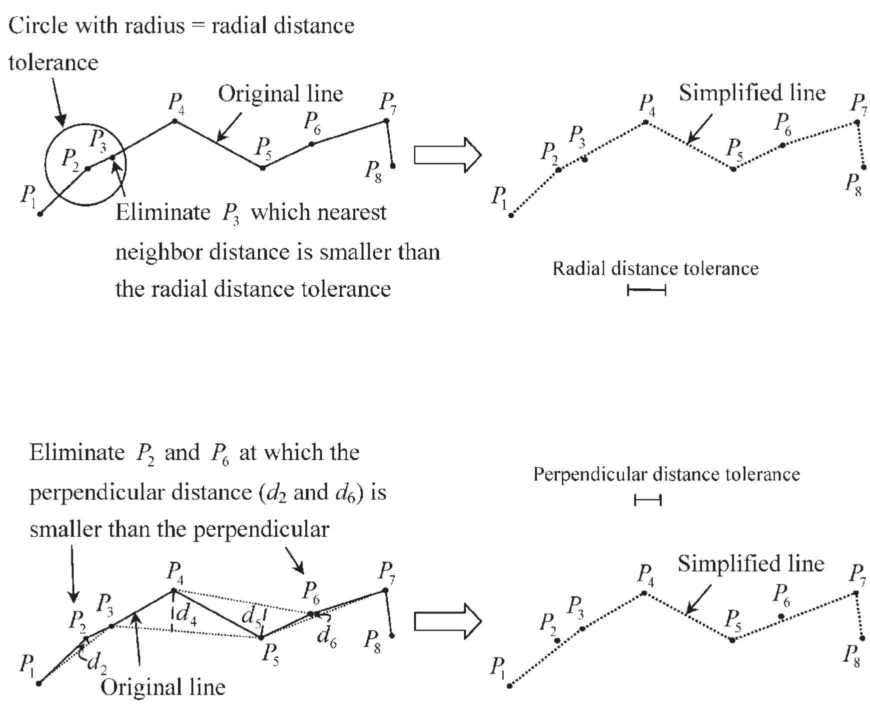

Figure 2. (a) The routine of distance between points and (b) the perpendicular distance routine

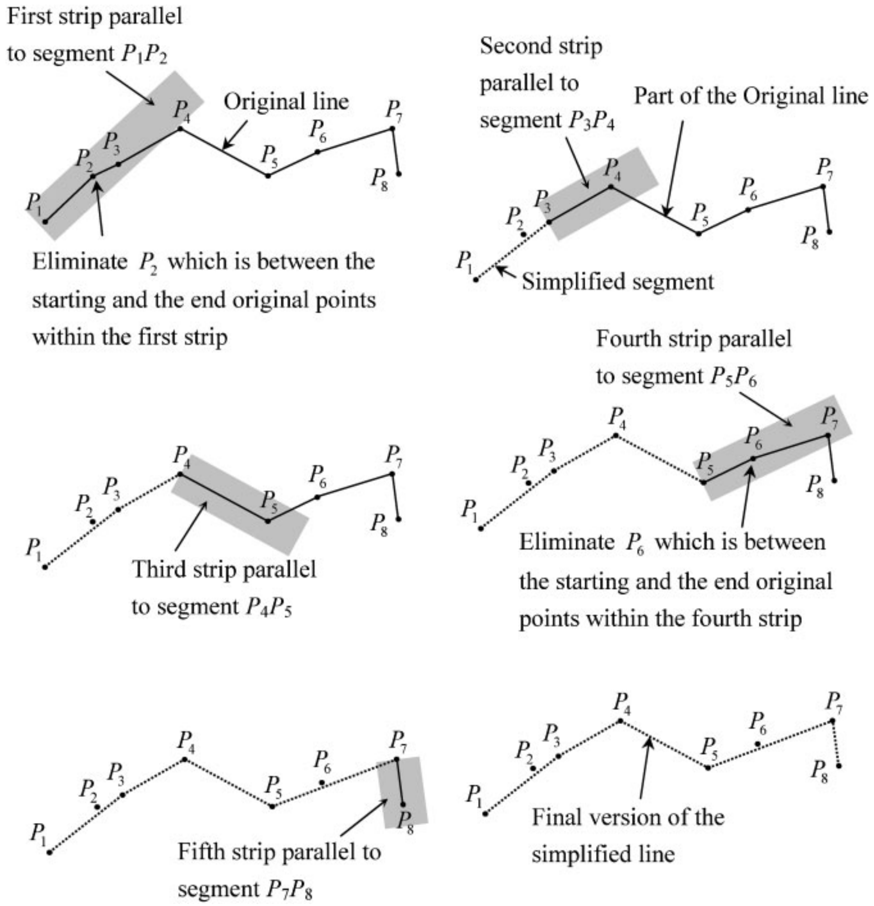

Figure 3. The Reumann–Witkam routine

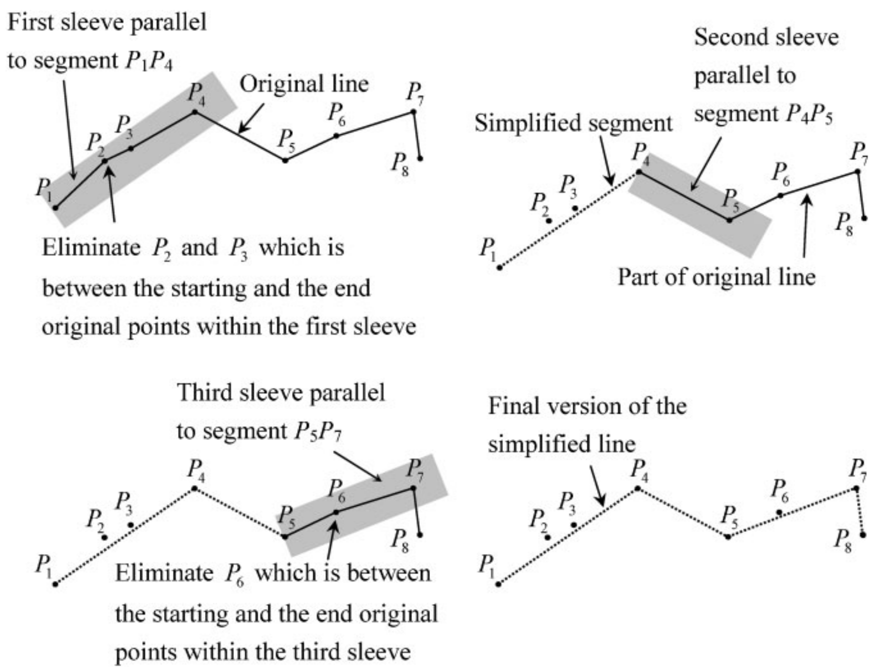

Figure 4. The sleeve-fitting polyline simplification algorithm

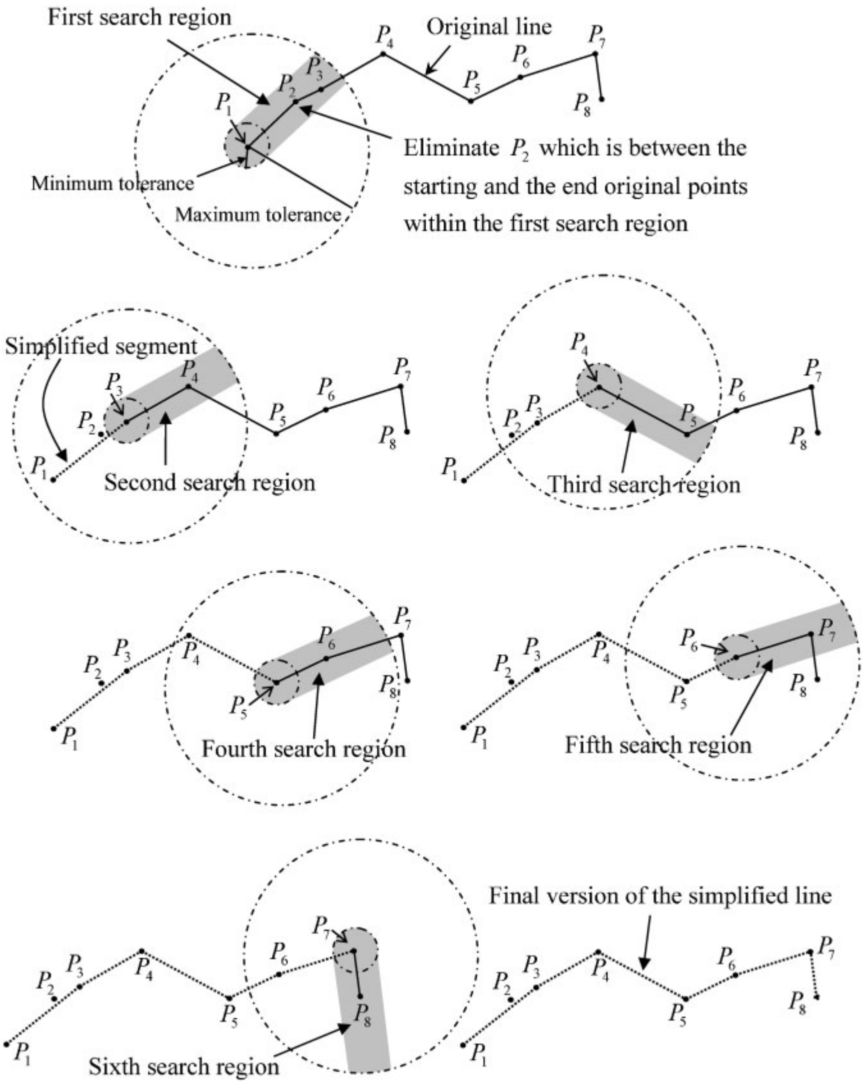

Figure 5. The Opheim simplification algorithm

Each sleeve starts at an original point P_j and contains consecutive original points $\{P_{j+1}, P_{j+1}, \ldots, P_{j+k}\}$, the direction of which is parallel to the line connecting the starting P_j and the last original points P_{j+k} of the sleeve. For the first sleeve, its starting original point is the starting endpoint P_1 of the original line and its last original point is the original point P_s with the largest subindex s such that all original points between P_1 and P_s are inside this sleeve. The second sleeve starts at the last original point P_s of the first sleeve and ends at the original point P_t with the largest subindex t such that the sleeve contains all original points between P_s and P_t. The sleeve is moved over the original line into the direction of the line connecting the starting and the last original points of each sleeve. The centreline of each sleeve is a simplified segment representing those original segments within the sleeve.

Constrained Extended Local Processing

Constrained extended local processing routines utilize the criteria in the unconstrained extended local processing routines and additional constraints to define a search region for the original line. Such a search region is used to divide the original line into sections. A first example of extended local processing with constraint was developed by Opheim (1981, 1982). *The Opheim simplification algorithm* defines the search region in a similar manner as the strip of Reumann-Witkam, and with a minimum and maximum distance constraint. The initial search region is shown in Figure 5 (top). Any original points within the minimum tolerance are eliminated; any original points within the search region bounded by the maximum tolerance are also eliminated. The last original point within the search region is retained. It then serves as the starting point of a new search region (Figure 5). The search region is shifted over the original line.

Another example was developed by Lang (1969). The search region of *the Lang simplification algorithm* is defined based on the perpendicular distance from a segment connecting two original points to the original points between them. Each search region is initialized as a region containing a fixed number of consecutive original points. Figure 6 (top), for example, shows the initial search region of four original points. If the distance from the segment,

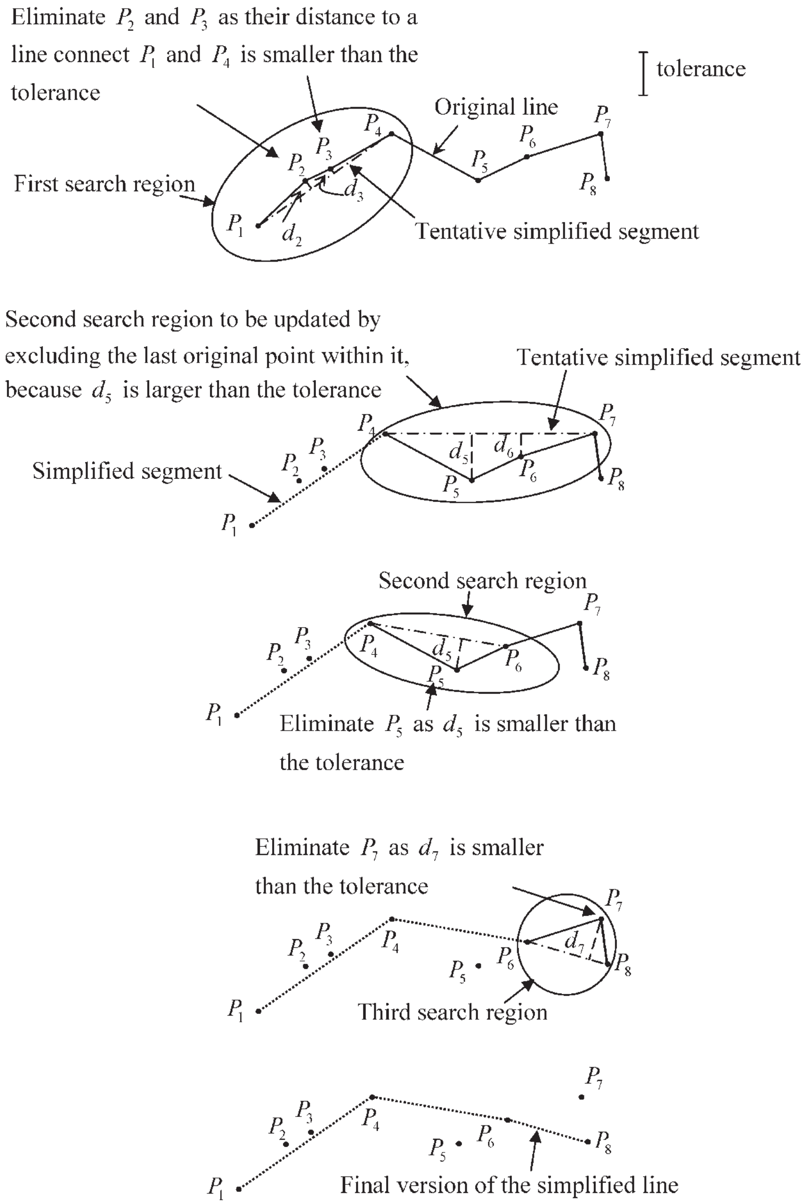

Figure 6. The Lang simplification algorithm

which connects the first and last original points inside the search region, to the individual original points between the segment's endpoints is less than the tolerance, a new search region will be initialized by setting the last original point as the starting point (Figure 6, middle); Otherwise, the search region will be updated by excluding the last original point (Figure 6, bottom). The search region is shifted over the original line.

Global Routines

A global routine considers a line entirely while processing, which is different from the first four categories, which process the line from the beginning to the end and so are sequential in nature. *The Douglas–Peucker algorithm* originally developed by Douglas and Peucker (1973), and modified by Hershberger and Snoeyink (1992) is the most commonly used global simplification algorithm in cartography and GIS. In this algorithm, an initial simplified line is a line connecting the start and end original points of the entire original line (Figure 7, top). For every intermediate original point, its perpendicular distance to the initial simplified line is calculated. The intermediate original point with the maximum perpendicular distance larger than a tolerance is served as one point of the simplified line in the final version. This intermediate original point is then used to partition the initial simplified line into two tentative simplified segments (Figure 7, middle). If the intermediate original point over each tentative simplified segment with the maximum perpendicular distance is larger than the tolerance, this point will be used to further partition the corresponding tentative simplified segment into two. This step is repeated until the perpendicular distance for all intermediate original points is smaller than the tolerance (Figure 7, bottom).

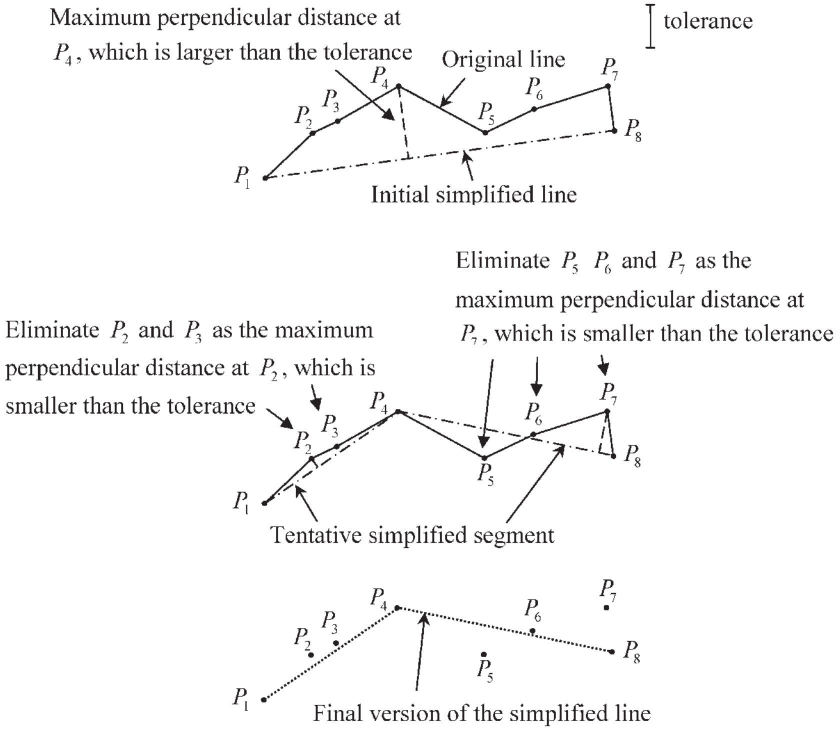

Figure 7. The Douglas–Peucker simplification algorithm

Another example of a global routine is an area-based line generalization proposed by Visvalingam and Whyatt (1993). For each original point between the start and end points of the original line, its effective area is the area of the triangle formed by it and its immediate neighbours (Figure 8). Initially, effective areas for all original points (except the starting and the ending endpoints) are computed and the original point with the minimum effective area is eliminated. At each iteration, the effective areas for the two adjoining points (to the point eliminated) are recomputed and the original point with the minimum effective area is eliminated. This procedure is repeated until the number of retained points is equal to a predefined value (say, 2).

According to Visvalingam and Williamson (1995), it is likely that the Douglas–Peucker algorithm performs better for minor weeding (minimal simplification) and the Visvalingam–Whyatt algorithm performs better for elimination of entire shapes (caricatural generalization) (Weibel, 1997). A variation of the simplified line from the original line is subject to two main factors: the density of original points and the line complexity, where the density of original points refers to the amount of points along the original line per unit length; and the line complexity relates to the original line's shape that can be quantified in terms of convexity, curvature or bending energy, for example. Among the presented algorithms, the two independent point algorithms (including the nth point routine and the routine of random-selection of points) and the routine of distance between points are mainly affected by the density of original points. The Opheim algorithm and the Visvalingam–Whyatt algorithm are affected by both the density of original points and the line complexity. The remaining algorithms introduced in this paper are associated with line complexity, because they consider either the angular change or the distance from one point to the simplified line. It is likely that the degree of line complexity varies with the simplification algorithm. The main factors influencing simplified results for the selected line simplification algorithms are summarized in Table 1.

Table 1. Main factors that influence different simplified results for different line simplification algorithms

Line simplification algorithm	Density of original points	Line complexity
*n*th point routine	*	
Routine of distance between points	*	
Perpendicular distance routine		*
Reumann–Witkam routine		*
Sleeve-fitting polyline simplification		*
Opheim simplification algorithm	*	*
Lang simplification algorithm		*
Douglas–Peucker algorithm		*
Visvalingam–Whyatt algorithm	*	*

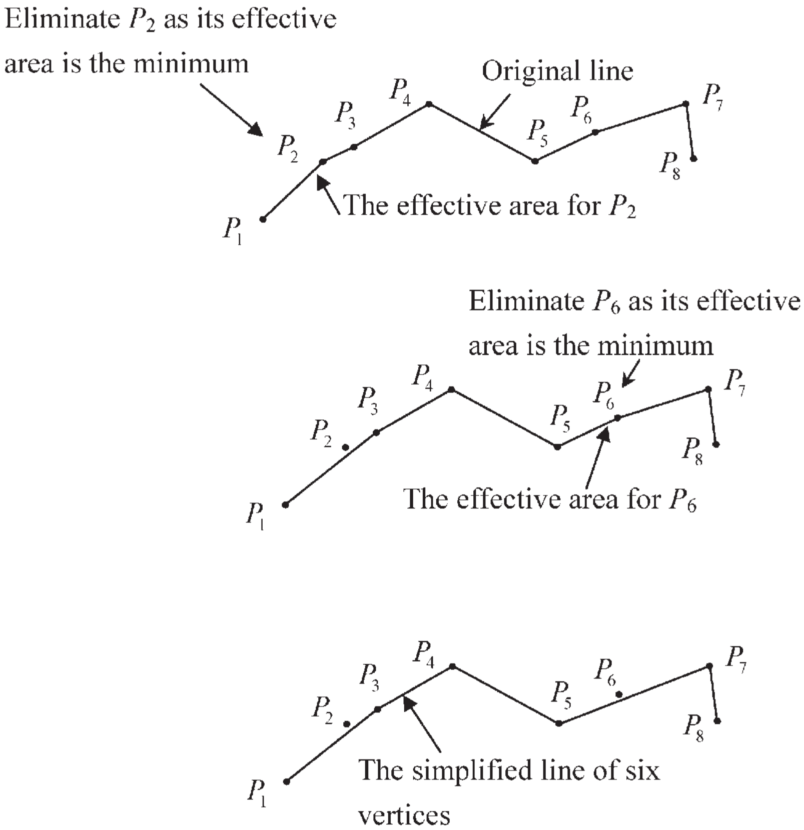

Figure 8. The Visvalingam–Whyatt algorithm

UNCERTAINTY MEASURES FOR LINE SIMPLIFICATION

Positional uncertainty in line simplification comes from positional uncertainty of the original line propagated through line simplification, and generalization error resulting from a displacement between the simplified line and the original line. In this paper, the positional uncertainty in line simplification is assessed with two measures. The first measure is distance-based, which assesses a maximum deviation between the locations of the original line and its simplified version.

In addition to the displacement between the original line and its simplified version, the cartographer is concerned with shape distortion because, based on the principle of cartographic generalization, the simplified or generalized version of the line should be similar to the original line in shape as much as possible. A shape measure can be derived in order to compare the simplified line with the original line in terms of shape (Cheung and Shi, in press). This shape measure also takes the positional uncertainty of the original line and the generalization error into consideration. Both the distance measure and shape measure are briefly introduced below.

A Maximum Distance Measure

The maximum distance measure derives an uncertainty description for the original line and the maximum distortion for the location difference between the original line and its simplified version. The uncertainty description for the original line used in this model was derived from the error ellipse model that is used to depict positional uncertainty of a point in surveying.

Figure 9(a) shows the position of each original point on the digital map at which an error ellipse is centred. The error ellipse represents a region in which the true location

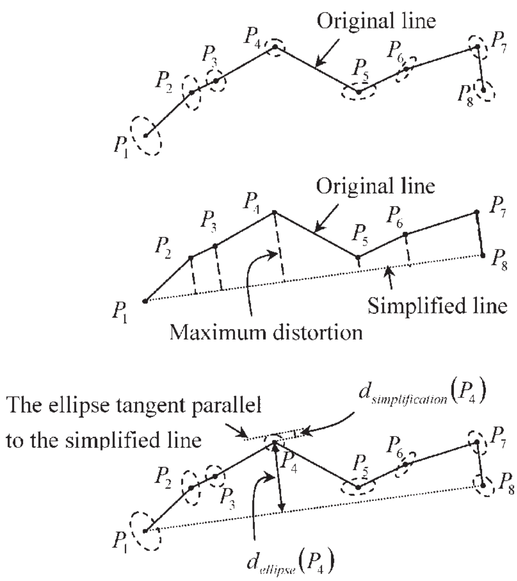

Figure 9. (a) Error ellipses about original points, (b) the maximum distortion of the simplified line and (c) the distance of an original point to the simplified line that equates the sum of $d_{simplification}(P)$ and $d_{ellipse}(P)$

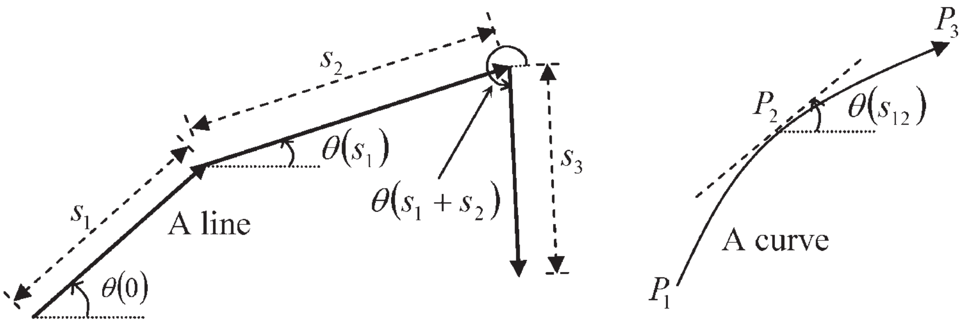

Figure 10. A straight line with different orientations at its vertices in (a), and a curve of the total arc length s_{13}, along which orientation is varied, where s_{12} is the arc length or the length of arc P_1P_2 along the curve in (b)

of the original point falls with a specific confidence. The size and orientation of each error ellipse is not necessarily the same, as this varies with the horizontal and vertical uncertainties in the point and their covariance. However, the error ellipse in Figure 9 will be an equal-sized circle if the overall positional accuracy of the digital map is given (instead of the individual point's positional accuracy).

The error induced by line simplification (or the location difference between the original line and its simplified version) is measured by the maximum distortion from the maximum individual perpendicular distance of the original points to the simplified line as shown in Figure 9(b).

To consider both the positional uncertainty of the original line and the generalization error simultaneously, a location difference between an original point with uncertainty and its closest position on the simplified line is computed, which is the sum of two distances:

- the greatest perpendicular distance of any point within the error ellipse to the original point [denoted by $d_{simplification}$ (P), where P represents an original point];
- the shortest distance of the original point to the simplified line [denoted by $d_{ellipse}$ (P)] [Figure 9(c)].

The maximum of the location difference for all original points with uncertainty is a measure of uncertainty in line simplification:

$$d_{max\_dist} = \max[d_{simplification}(Pi) + d_{ellipse}(Pi)] \quad (1)$$

In the case where the original line is considered to be without positional uncertainty, the maximum location difference between the original line and its simplified version is given by:

$$D_{mean\_dist} = \max[d_{simplification}(Pi)] \quad (2)$$

A Shape Dissimilarity Measure

Error in line simplification can be measured based on displacement measurements and linear attribute measurements (McMaster, 1986). Displacement measurements assess the gap between the original line and its simplified version. One example of this type is the maximum distance measure that has already been introduced. Linear attribute measurements create a similarity or dissimilarity function that measures the degree of similarity or dissimilarity between the original line and its simplified version, based on a geometric characteristic (such as the angle of inclination).

The angle of inclination is the (horizontal) angle between a straight line and the x-axis over the unit length of the line or the angle between a tangent to a curve and the x-axis over the unit length of the curve (the arctangent of the tangent's slope) (Figure 10). Its difference between the original line, A, and its simplified version, B, can be expressed as the following dissimilarity function:

$$d_{mean\_\theta}(A,B) = \int_{s=0}^{1} AngDiff\left(\tilde{\theta}_A(s), \tilde{\theta}_B(s)\right) ds \quad (3)$$

where $\tilde{\theta}_A(s)$ and $\tilde{\theta}_B(s)$ are functions for the normalized inclination angle of the line A and the line B, respectively, given that the length of these two lines is scaled to 1, and $AngDiff(\alpha,\beta)$ for α,β (0°, 360°) is the minimum angle difference between α and β (Figure 11). This dissimilarity function is applicable where the original line does not contain positional uncertainty.

When the original line contains positional uncertainty, the dissimilarity function is defined as

$$d_{\max\_\theta}(A,B) = \int_{s=0}^{1} \max_i\left\{AngDiff\left(\tilde{\theta}_{A(i)}(s), \tilde{\theta}_B(s)\right)\right\} ds \quad (4)$$

where $\tilde{\theta}_{A(i)}(s)$ refers to the function of the normalized inclination angle of line A_i given that the original line A on the digital map is scaled to 1, A_i is a line each of which vertices is centred inside the error ellipse at the corresponding original point on the digital map, and $\tilde{\theta}_B(s)$ is a function of the normalized inclination angle of the line B. Figure 12 shows the inclination angle of the simplified line B is fixed, $\tilde{\theta}_B(0)$; for each segment P_jP_{j+1} of the original line A, its inclination angle is between:

$$\left[\tilde{\theta}_{\min\_A}(s_j), \tilde{\theta}_{\max\_A}(s_j)\right]$$

The maximum difference between the inclination angle for P_jP_{j+1} and its simplified representation is computed based upon a comparison between $\tilde{\theta}_B(0)$ and $\left[\tilde{\theta}_{\min}(s_j), \tilde{\theta}_{\max}(s_j)\right]$, as shown in Figure 12(c), and is denoted by $\Delta\tilde{\theta}(s_j)$. Therefore, the dissimilarity function for the line simplification given in

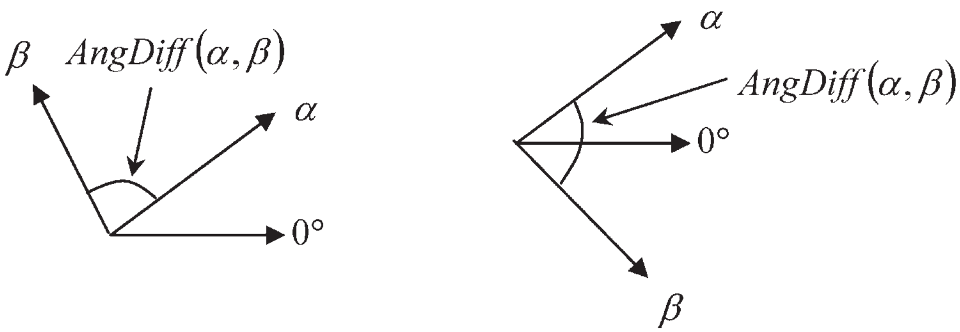

Figure 11. The minimum angle difference between α and β

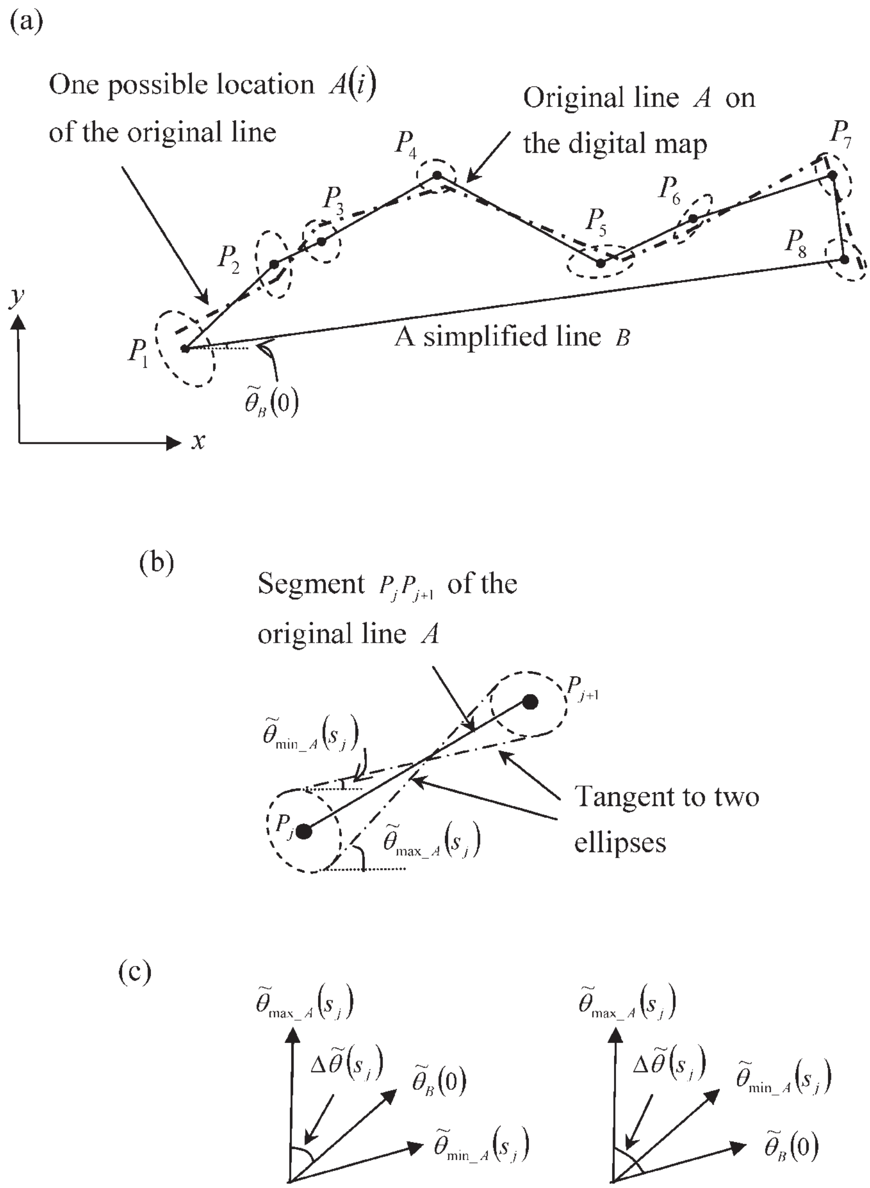

Figure 12. (a) One possible location A_i of the original line A and the location of the simplified line B with the inclination angle $\tilde{\theta}_B(0)$. (b) The inclination angle of segment P_jP_{j+1} of the original line A, which is between $[\tilde{\theta}_{\min\_A}(s_j), \tilde{\theta}_{\max A}(s_j)]$. (c) The difference between the inclination angles of $\tilde{\theta}_B(0)$ and $[\tilde{\theta}_{\min\_A}(s_j), \tilde{\theta}_{\max A}(s_j)]$

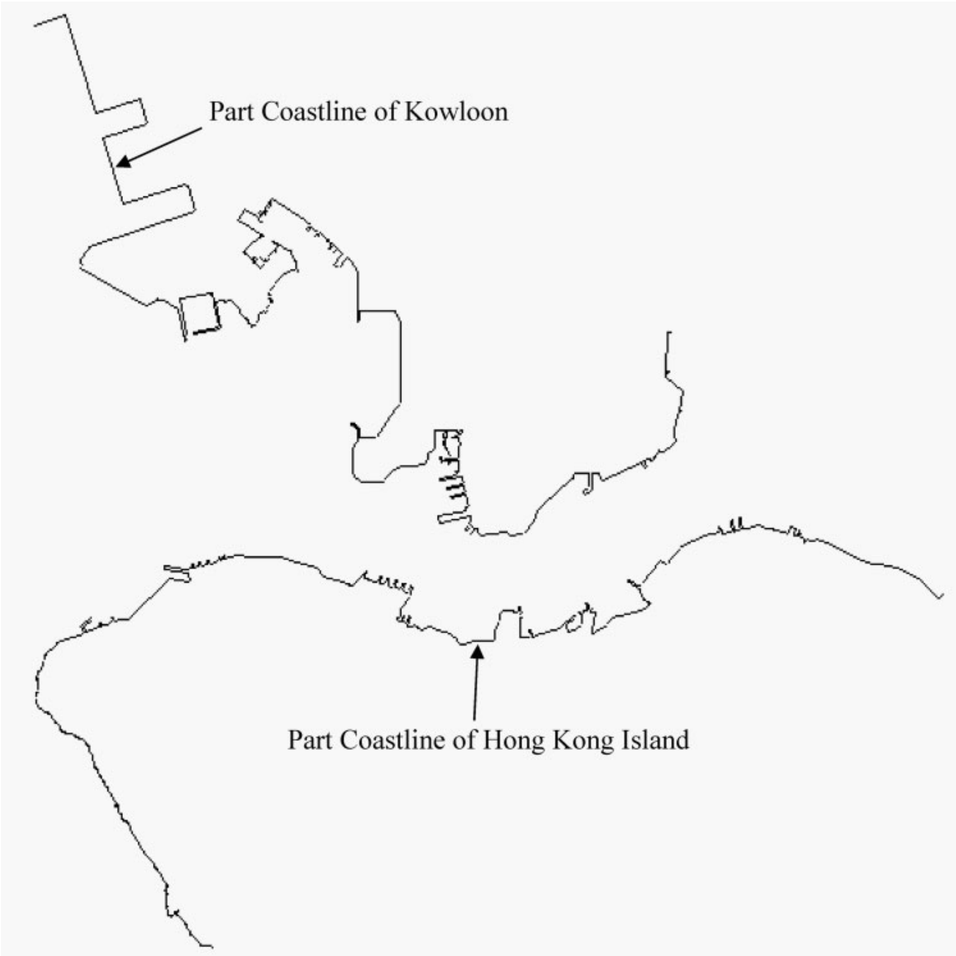

Figure 13. Part coastlines of Hong Kong Island and Kowloon at a scale of 1:80,000, which was originated from a digital coastline (1:5000) by zooming out in ArcMap 8 ×

Figure 12 can be modified as:

$$d_\theta(A,B) = \sum_{j=0}^{7} \Delta\widetilde{\theta}(s_j)\cdot(s_{j+1} - s_j) \qquad (5)$$

where

$$s_j = \frac{\text{the length of the line with vertices } P_0, P_1, \cdots, P_j}{\text{the length of the line with vertices } P_0, P_1, \cdots, P_8}$$

The detail of this dissimilarity measure is presented in Cheung and Shi (2005, in press).

TEST DATA SET FOR EVALUATING THE LINE SIMPLIFICATION ALGORITHMS

Part coastlines of Hong Kong Island and of Kowloon (both in Hong Kong) were selected as original lines to be simplified. These original lines contain natural and artificial features, which cover lines of low and high intensity directional changes characteristic. Figure 13 shows the selected coastlines of 1500 points each. The coastline is one layer in a 1:5000 dataset (B5000) that is derived from the digital geographical database B1000 of the Land Information Centre (LIC), Surveying and Mapping Office, Lands Department, Hong Kong Special Administrative Region. In general, accuracy of features in the B5000 dataset is same as that in the B1000 dataset, except for some generalized features [Land Information Centre LIC, 2003]. The accuracy of features in the B1000 varies from about 0.015 to 1.500 m under the assumption that errors are uncorrelated and normally distributed (LIC, 2002). Since the data supplier has not stated accuracy of individual features in the B5000 dataset in hand, we assume that horizontal and vertical positional uncertainties in each vertex of the coastlines are identical to 1.500 m, which is the value in the B1000 dataset in the worse case for the sake of convenience.

EVALUATION OF THE LINE SIMPLIFICATION ALGORITHMS

This section compares different versions of the simplified line generated by testing line simplification algorithms based on a visual comparison, the maximum distance measure (d_{max_dist} computed from Equation 1), the shape dissimilarity measure [d_{max_θ} computed from Equation (4)], and the processing time (the time the computer takes to complete a prescribed line simplification algorithm). Such a comparison will make it possible for a cartographer or a data user to evaluate effectiveness and efficiency of the line simplification algorithms. The effectiveness is interpreted by the two uncertainty measures, while efficiency is indicated by the processing time.

It should be noted that another important factor in the efficiency evaluation is the compression ratio; the ratio of the number of the retained original points to the total number of the original points. The compression ratio, however, will not be considered in this paper because different line simplification algorithms use different criteria to remove or retain original points. For example, the *n*th point routine retains every *n*th point, the routine of distance between points considers a distance criterion, and the Reumann–Witkam routine uses a directional criterion. The original points to be retained are subject to parameter(s) of a prescribed line simplification algorithm relative to its own criterion, which could have different meanings in various line simplification algorithms. This implies that comparing simplified lines generated by different line simplification algorithms with the same value of the parameter(s) is unvalued. In this study, for each comparison of a simplified lines' set, the compression ratio remains unchanged. In other words, the number of the retained original points is fixed in the performance evaluation.

The simplified lines generated by testing line simplification algorithms are first compared visually. Figures 14 and 15 show the simplified lines of 500 points, which are simplified versions of coastlines of Hong Kong Island and Kowloon, respectively, which are roughly ordered from a higher similarity (top, left) to a lower similarity (bottom, right) relative to the original line. It is found that the Visvalingam–Whyatt (VW) algorithm, the Douglas–Peucker (DP) algorithm, the Zhao–Saalfeld (ZS) algorithm, the Reumann–Witkam (RW) algorithm, the routine of distance between points (DBP), the Opheim (OP) algorithm and the Lang (LA) algorithm provide a similar detailed representation of the shape of the original line, as compared with the other line simplification algorithms, because their simplified line is more similar to the original line. The perpendicular distance (PD) routine provides a general representation of the shape of the original line, while the *n*th point (NTH) routine only provides a rough representation of the shape of the original line. In this preliminary analysis, therefore, the VW, DP, ZS, RW, DBP, OP and LA simplified lines of 500 points each have less shape distortion and higher accuracy in terms of visual shape comparison.

Next, a series of numerical comparisons based on the mean distance d_{mean_dist} and maximum distance d_{max_dist} measures, the mean dissimilarity d_{mean_θ} and maximum dissimilarity d_{max_θ} measures, as well as the processing time is performed. Values of the four measures and the processing time for the testing line simplification algorithms are shown in Figures 16–18. In these figures, simplified lines generated by the individual testing line simplification algorithms are of 500, 750, 1000 and 1250 points. The only exception is the *n*th point routine, where the simplified lines contain 500 and 750 points, due to the fact that the parameter of this routine must be a positive integer.

As Figure 16 illustrates, the mean distance appears to be identical to the maximum distance for all of the line simplification algorithms considered in this case study. This result means that the displacement between the original line and its simplified version dominates the uncertainty of the simplified line in terms of distance. In other words, the uncertainty of the original line does not significantly affect the uncertainty of the simplified line. However, Figure 17 shows that the mean dissimilarity measure is significantly smaller than the maximum dissimilarity measure. The uncertainty of the original line cannot, therefore, be ignored in assessing the uncertainty of the simplified line in terms of shape. Both displacement and shape distortion for a simplified line should be considered in line simplification to ensure that the simplified line retains the information of the original line as much as possible. The uncertainty of the original line should be taken into account in assessing positional uncertainty of a simplified line.

The mean (or maximum) distance shown in Figure 16 has an inverse relation with the number of vertices of the simplified line. This is due to the fact that the gap between the original line and its simplified version usually enlarges as fewer points are retained to compose the line in the simplified version. The only exception is the Opheim's algorithm: the mean (or maximum) distance at the number of vertices of the simplified line of 500 and 750 are equal. The reason is that the mean distance measure adopted here is the maximum of perpendicular distances of individual original points to the simplified line [the maximum value of $d_{ellipse}(P_i) + d_{simplification}(P_i)$ as shown in Figure 9]; and the original point with the maximum perpendicular distance for the simplified lines of 500 and 750 is the same.

The shape distortion between the original line and its simplified version may not have an inverse relation with the number of vertices of the simplified line for some line simplification algorithms. One example of this situation is the routine of distance between points as shown in Figure 17. Such a situation demonstrates that a simplified line of more vertices may not result in a better shape description relative to the original line. Figure 19 shows an example of an original line and two versions of its simplified line, in which the shape distortion for the simplified line with two vertices is less than that for the simplified line with three vertices.

We now turn our attention to the two uncertainty measures in the 'certain' case in which the original line does not contain positional uncertainty. For the mean distance shown in Figure 16, its value for the Douglas–Peucker algorithm is the minimum; this implies that this algorithm is ranked best in terms of displacement among the testing line simplification algorithm in the 'certain' case. In general, the Lang algorithm, the Reumann–Witkam algorithm, the Zhao–Saalfeld algorithm and the Visvalingam–Whyatt algorithm perform less well.

Figure 17 gives the result of the shape-based approach in the 'certain' case. The most acceptable outcome is given by the Douglas–Peucker algorithm with a minimum mean dissimilarity as compared with the other testing line simplification algorithms. The Visvalingam–Whyatt

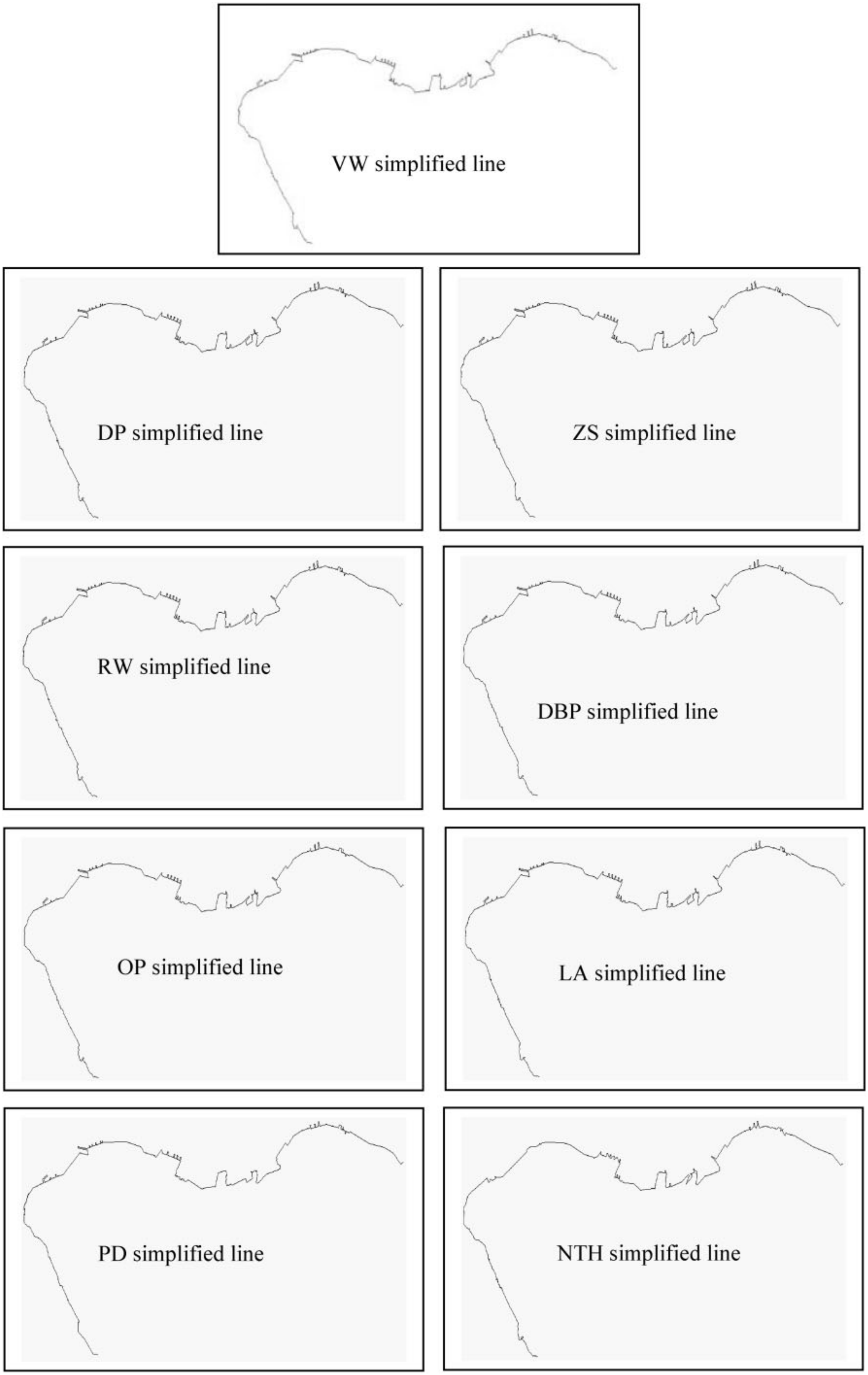

Figure 14. A set of 500-point simplified lines of the coastline of Hong Kong island generated by: (a) the Visvalingam–Whyatt algorithm; (b) the Douglas–Peucker algorithm; (c) the Zhao–Saalfeld algorithm; (d) the Reumann–Witkam algorithm; (e) the routine of distance between points; (f) the Opheim algorithm; (g) the Lang algorithm; (h) the perpendicular distance routine; and (i) the *n*th point routine

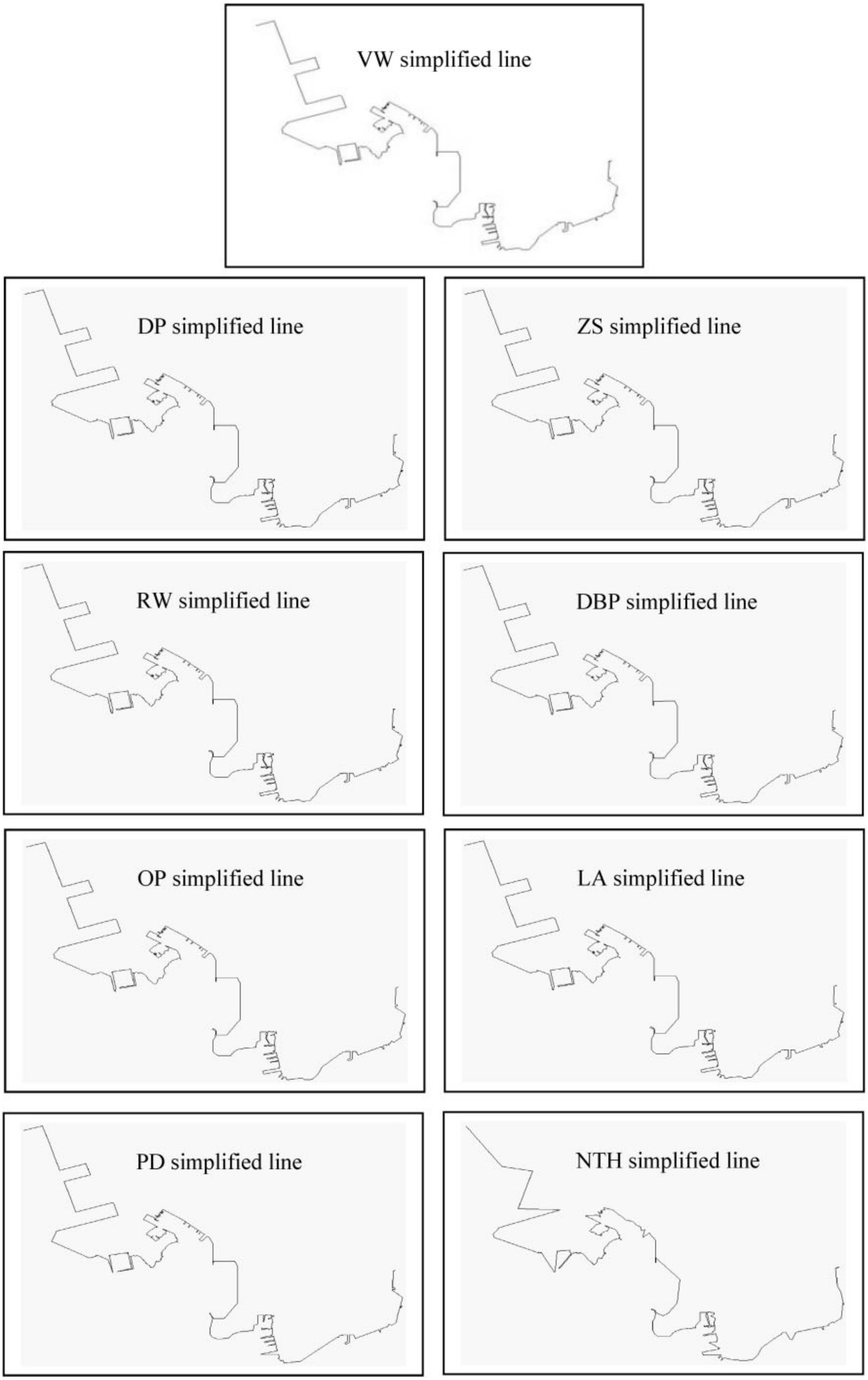

Figure 15. A set of 500-point simplified lines of the coastline of Kowloon generated by: (a) the Visvalingam–Whyatt algorithm; (b) the Douglas–Peucker algorithm; (c) the Zhao–Saalfeld algorithm; (d) the Reumann–Witkam algorithm; (e) the routine of distance between points; (f) the Opheim algorithm; (g) the Lang algorithm; (h) the perpendicular distance routine; and (i) the *n*th point routine

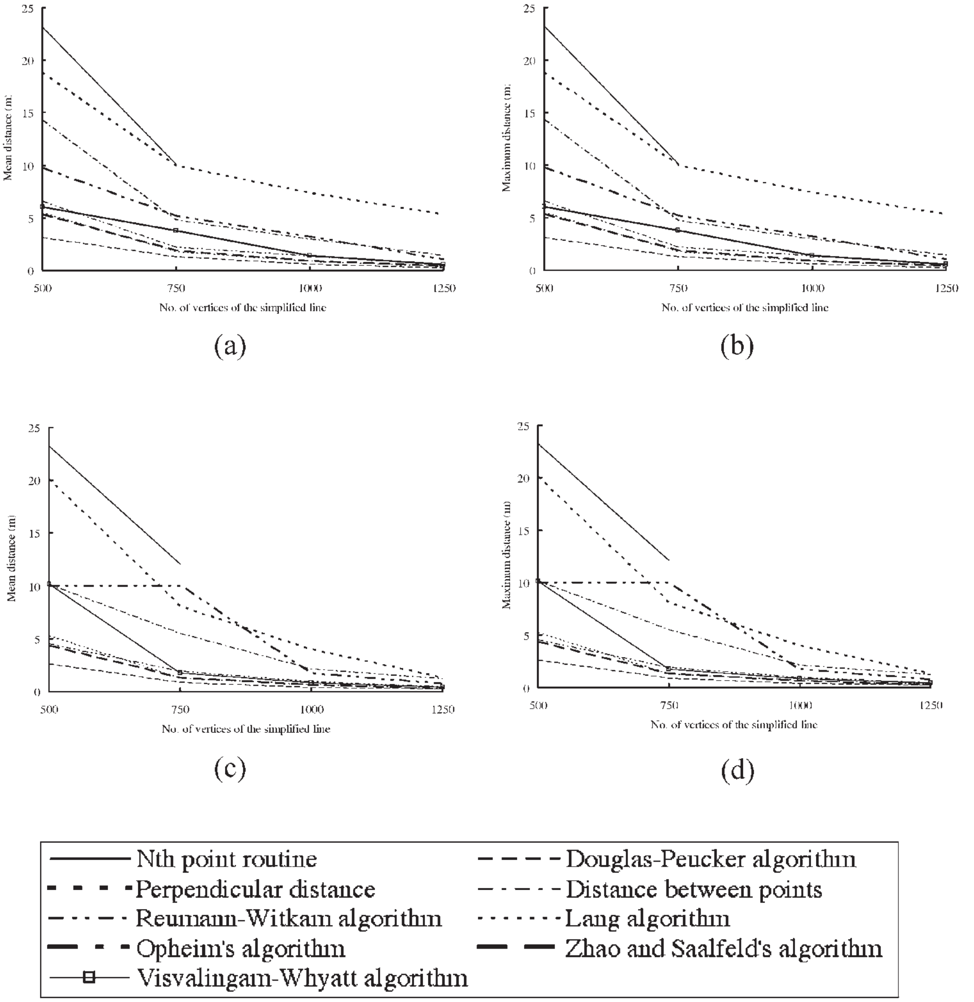

Figure 16. The mean distances and the maximum distances for Hong Kong Island (a,b) and for Kowloon (c,d)

algorithm, the Zhao–Saalfeld algorithm, the Reumann–Witkam, the Lang algorithm and the perpendicular distance routine ranks lower. This result is similar to the outcome obtained when using the mean distance measure. Both the mean distance measure and the mean dissimilarity measure suggest that the Douglas–Peucker algorithm provides a higher accurate simplified line.

Where the original line contains positional uncertainty, the question remains as to whether the Douglas–Peucker algorithm also results in a better performance? As Figure 16 illustrates, the maximum distance measure shows that that the Douglas-Peucker algorithm is the most accurate line simplification algorithm and Figure 17 shows the relationship between the number of vertices of the simplified line and the maximum dissimilarity in the uncertain case. The Douglas–Peucker algorithm has the minimum value of the maximum dissimilarity and so it has the best performance in terms of shape distortion.

A summary of the positional accuracy of the simplified line generated by different line simplification algorithms is given in Table 2. First, the Douglas–Peucker algorithm provides a simplified line with less displacement and less shape distortion. This algorithm, however, is relatively time-consuming (Figure 18). The Douglas–Peucker algorithm may not be the best choice if a large amount of data is to be simplified rapidly, such as the process of zooming out on the digital map. An alternative in this instance may be the Lang or the Zhao–Saalfeld algorithm, which have positional accuracy lower than the Douglas–Peucker algorithm, but higher than the other testing line simplification algorithms (except the Reumann–Witkam algorithm); and they require significantly less processing time than the Douglas–Peucker and the Reumann–Witkam algorithm as shown in Figure 18.

CONCLUDING REMARKS

Automatic line generalization has been developed over four decades. A diversity of automatic algorithms have been derived, based on a distance between two points, a perpendicular distance of a point to a line, an angular change between points and others. This study compares the performance of line simplification algorithms in terms of positional accuracy and processing time.

Positional accuracy of the simplified line was measured by a displacement measure and a shape distortion measure. The displacement measure was used to quantify the gap

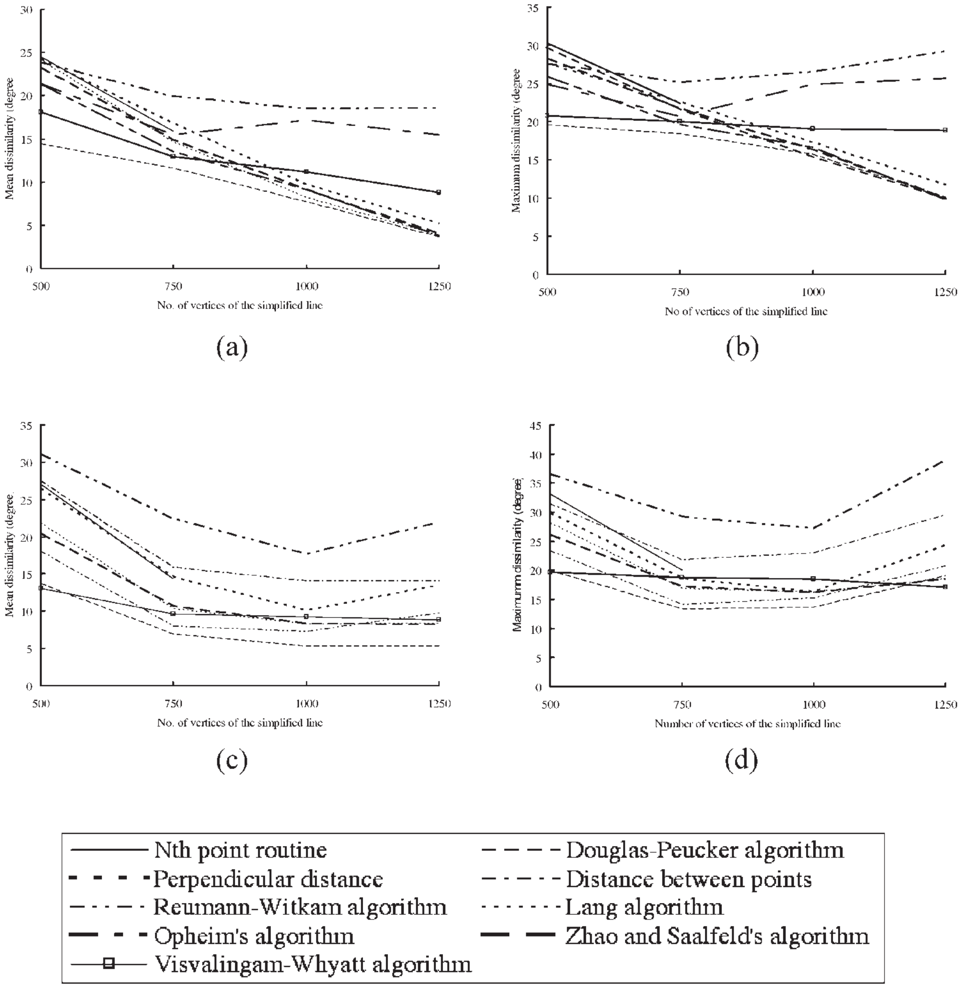

Figure 17. The mean dissimilarities and the maximum dissimilarities for Hong Kong Island (a,b) and for Kowloon (c,d)

between the original line with uncertainty and its simplified version. The shape distortion measure was used to assess the shape dissimilarity between the original line with uncertainty and its simplified version. These two measures are different from the positional accuracy measure used in other research in performance evaluation, which only considered the gap or shape dissimilarity between the original line without uncertainty and its simplified version.

The case study illustrated that the Douglas–Peucker algorithm produced the most accurate generalization; a finding similar to previous research. However, the performance is hindered by it being time-consuming. In order to

Table 2. A summary of the visual difference, the displacement, the shape distortion and the computation time for the simplified result obtained from different line simplification algorithms. More stars in each field represent a better performance in terms of the corresponding assessment. Note that the 'certain' case refers to the case in which the original line is assumed to be without positional error, while the uncertain case refers to the case in which positional uncertainty of the original line is considered.

Line simplification Algorithm	Visual difference	Displacement in the certain' case	Displacement in the uncertain case	Shape distortion in the 'certain' case	Shape distortion in the uncertain case	Reciprocal of the computation time
Douglas–Peucker	***	***	***	***	***	*
Zhao–Saalfield	***	**	**	**	*	***
Reumann–Witkam	***	**	**	**	**	*
Visvalingham–Whyatt	***	**	**	**	*	*
Distance between points	***	*	*	*	*	***
Opheim	***	*	*	*	*	*
Lang	***	**	**	**	*	**
Perpendicular distance	**	*	*	*	*	***
*n*th point	*	*	*	*	*	***

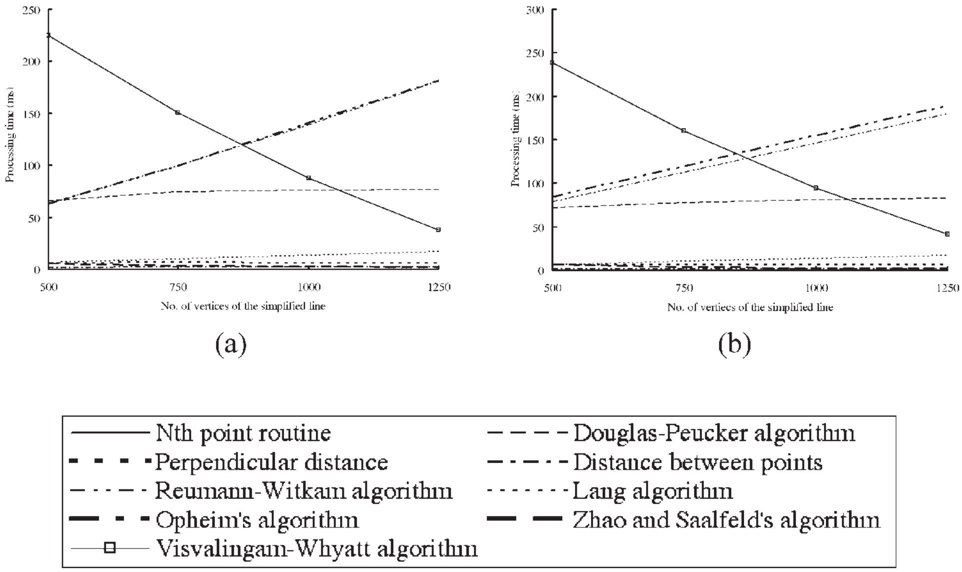

Figure 18. The processing time for Hong Kong Island (a) and Kowloon (b)

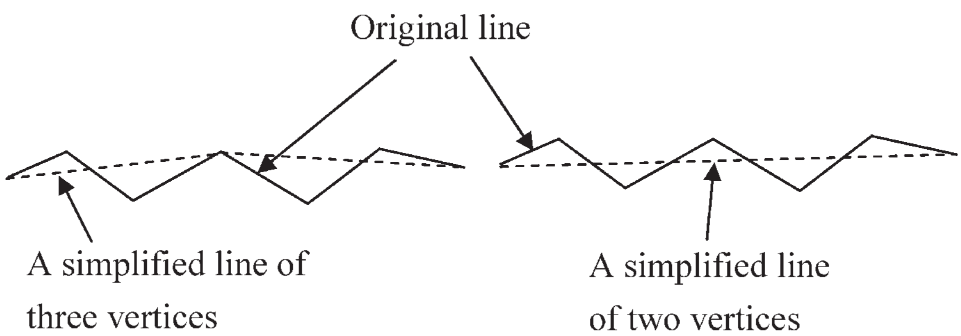

Figure 19. A simplified line of three vertices (left) that does not describe the original line better than a simplified line of two vertices (right)

provide a GIS user with an on-line line simplification process, the Lang algorithm and the Zhao–Saalfeld algorithm would be of greater practical benefit. Both could rapidly produce a simplified line with positional accuracy higher than the other line simplification algorithms.

ACKNOWLEDGEMENTS

The work described in this paper was substantially supported by The Hong Kong Polytechnic University (Project No. G-YX27). We would like to thank the anonymous referees for valuable comments on this paper. We also thank the Survey and Mapping Office, the Government of the Hong Kong Special Administrative Region, for providing the digital map.

REFERENCES

Bader, M. (2001). 'Energy Minimization Methods for Feature Displacement in Map Generalization'. PhD thesis, University of Zurich.

Burghardt, D. (2005). 'Controlled line smoothing by snakes', **GeoInformatica**, 9(3), 237–252.

Burghardt, D. and Meier, S. (1997). 'Cartographic displacement using the snakes concept', in **Semantic Modeling for the Acquisition of Topographic Information from Images and Maps**, 59–71, ed. by Foerstner, W. and Pluemer, L., Birkhaeuser-Verlag, Basel.

Cheung, C. K. and Shi, W. Z. (2004). 'Estimation of the positional uncertainty in line simplification in GIS', **The Cartographic Journal**, 41(1), 37–45.

Cheung, C. K. and Shi, W. Z. (in press). 'Positional error modeling for line simplification based on automatic shape similarity analysis in GIS', **Computers & Geosciences**.

Cottingham, S. (1997). 'Generalization of Categorical Maps: A Phenomenological Approach to Soil Mapping'. PhD dissertation, University of Edinburgh.

DeLucia, A. A. and Black, R. T. (1987). 'A comprehensive approach to automatic feature generalization', in **Proceedings of the 13th Conference of the International Cartographic Association**, Morelia, vol. 4, pp. 173–191.

Douglas, D. and Peucker, T. (1973). 'Algorithms for the reduction of the number of points required to represent a digitized line or its caricature', **Canadian Cartographer**, 10, 112–122.

Galanda, M. (2001). 'Optimization techniques for polygon generalization', in **ICA-Workshop on Progress in Automated Map Generalization**, Beijing. Available at: http://www.geo.unizh.ch/ICA/docs/beijing2001/papers01.html

Galanda, M. (2003). 'Modeling constraints for polygon generalization', in **Proceedings of the 5th ICA Workshop on Progress in Automated Map Generalization**, Paris.

Galanda, M. and Weibel, R. (2003). 'Using an energy minimization technique for polygon generalization', **Cartography and Geographic Information Science**, 30(3), 263–279.

Harrie, L. and Sarjakoski, T. (2000). 'Generalization of vector data sets by simultaneous least squares adjustment', **International Archives**

of Photogrammetry and Remote Sensing, Amsterdam, XXXIII(A4), 340–347.

Hershberger, J. and Snoeyink, J. (1992). 'Speeding up the Douglas–Peucker line-simplification algorithm', in **Proceedings of the Fifth International Symposium on Spatial Data Handling**, 1, pp. 134–143.

João, E. M. (1998). **Causes and Consequences of Map Generalization**, Taylor & Francis, London.

Jones, C. B., Bundy, L. and Ware, J. M. (1995). 'Map generalization with a triangulated data structure', **Cartography and Geographic Information Systems**, 22(4), 317–331.

Land Information Centre (2002). Surveying and Mapping Office, Lands Department, the Government of the Hong Kong Special Administrative Region. 1:1000 Basic Mapping System – Metadata, 9 December 2002, 10 November 2004. Available at: http://www.info.gov.hk/landsd/mapping/eng/lic/metadata/b1000.html.

Land Information Centre (2003). Surveying and Mapping Office, Lands Department, the Government of the Hong Kong Special Administrative Region. B5000 Mapping System – Metadata, 14 August 2003, 10 November 2004. Available at: http://www.info.gov.hk/landsd/mapping/eng/lic/metadata/b5000.html.

Lang, T. (1969) 'Rules for robot draughtsmen', **Geographical Magazine**, 42, 50–51.

Lee, D. (2004). 'Geographic and cartographic contexts in generalization', in **Proceedings of ICA Workshop on Generalization and Multiple Representation**, Leicester.

McMaster, R. B. (1986). 'A statistical analysis of mathematical measures for linear simplification', **The American Cartographer**, 13(2), 103–117.

McMaster, R. B. (1987). 'The geometric properties of numerical generalization', **Geographical Analysis**, 19(4), 330–346.

Müller, J. C. (1987). 'Fractal and automatic line generalization', **The Cartographic Journal**, 24, 27–34.

Müller, J. C. (1991). 'Generalization of spatial databases', in **Geographical Information Systems: Principles and Applications**, ed. by Maguire, D., Goodchild, M. and Rhind, D., 1, pp. 457–475, Longman, Harlow.

Opheim, H. (1981). 'Smoothing a digitized curve by data reduction methods', in **Eurographics '81**, ed. by Encarnacao, J. L., pp. 127–135, North-Holland Publishing Company, Amsterdam.

Opheim, H. (1982). 'Fast data reduction of a digitized curve', **Geo-Processing**, 2, 33–40.

Reumann, K. and Witkam, A. P. M. (1974). 'Optimizing curve segmentation in computer graphics', in **Proceedings of International Computing Symposium**, pp. 467–472, North-Holland Publishing Company, Amsterdam.

Sester, M. (2000). 'Generalization based on least squares adjustment', **International Archives of Photogrammetry and Remote Sensing, Amsterdam**, XXXII(B4), 931–938.

Visvalingam, M. and Whyatt, J. D. (1993). 'Line generalization by repeated elimination of points', **The Cartographic Journal**, 30, 46–51.

Visvalingam, M. and Williamson, P. J. (1995). 'Simplification and generalization of large scale data for roads: a comparison of two filtering algorithms', **Cartography and Geographic Information Systems**, 22(4), 264–275.

Weibel, R. (1986). 'Automated cartographic generalization', in **A Selected Bibliography on Spatial Data Handling: Data Structures, Generalization, and Three-dimensional Mapping**, ed. by Sieber, R. and Brassel, K., 6, pp. 20–35, Geo-Processing Series, Department of Geography, University of Zürich.

Weibel, R. (1997). 'Generalization of spatial data: principles and selected algorithms', in **Algorithmic Foundations of Geographic Information Systems**, ed. by van Kreveld, M., Nievergelt, J., Roos, T. and Widmayer, P., pp. 99–152, Springer-Verlag, Berlin.

White, E. (1985). 'Assessment of line generalization algorithms using characteristic points', **The American Cartographer**, 12, 17–28.

Zhao, Z. and Saalfeld, A. (1997). 'Linear-time sleeve-fitting polyline', in **Autocarto 13, ACSM/ASPRS'97 Technical Papers**, Seattle, Washington, April 1997, 5, pp. 214–223.