

Article

A New Algorithm for Cartographic Simplification of Streams and Lakes Using Deviation Angles and Error Bands

Türkey G ö k g ö z *, Alper Sen, Abdulkadir Memduhoglu and Muslum Hacı

Department of Geomatic Engineering, Faculty of Civil Engineering, Yıldız Technical University, Esenler, Istanbul 34220, Turkey; E-Mails: alpersen@yildiz.edu.tr (A.S.); akadirm@yildiz.edu.tr (A.M.); mhacar@yildiz.edu.tr (M.H.)

* Author to whom correspondence should be addressed; E-Mail: turkaygokgoz@gmail.com; Tel.: +90-212-383-5341; Fax: +90-212-383-5102.

Academic Editor: Wolfgang Kainz

Received: 23 June 2015 / Accepted: 10 October 2015 / Published: 22 October 2015

Abstract: Multi-representation databases (MRDBs) are used in several geographical information system applications for different purposes. MRDBs are mainly obtained through model and cartographic generalizations. Simplification is the essential operator of cartographic generalization, and streams and lakes are essential features in hydrography. In this study, a new algorithm was developed for the simplification of streams and lakes. In this algorithm, deviation angles and error bands are used to determine the characteristic vertices and the planimetric accuracy of the features, respectively. The algorithm was tested using a high-resolution national hydrography dataset of Pomme de Terre, a sub-basin in the USA. To assess the performance of the new algorithm, the Bend Simplify and Douglas-Peucker algorithms, the medium-resolution hydrography dataset of the sub-basin, and Töpfer's radical law were used. For quantitative analysis, the vertex numbers, the lengths, and the sinuosity values were computed. Consequently, it was shown that the new algorithm was able to meet the main requirements (*i.e.*, accuracy, legibility and aesthetics, and storage).

Keywords: cartography; generalization; simplification; algorithm; deviation angle; error band

1. Introduction

The idea that we can abstract and portray geographic information at multiple scales in map form has existed for thousands of years. National Mapping Agencies (NMAs) maintain spatial databases at different levels of detail—Databases that store multiple representations of the same geographic phenomena. Multi-representation databases (MRDBs) can either be created by connecting objects among existing databases or by generating smaller scale representations from a single large scale database via automatic generalization [1]. In many geographical information system (GIS) applications, users need to visualize and inspect data at different scales, which requires different representations to be stored at different levels of detail. The flexibility of MRDBs lies in its ability to derive different types of maps, using models and cartographic generalization methods. In this respect, automated spatial data generalization techniques are as important as data modeling, management, and distribution. Web or mobile mapping applications typically display some thematic foreground data against the spatial reference provided by some background data (e.g., a topographic map or orthoimagery). The background data is usually assumed to be static in content and can, thus, be rendered by a pre-generalized tile service (Open Street Map-based map services, Google Maps, Bing Maps) to provide seamless map interaction. The foreground data, on the other hand, will vary in content depending on the user's request. Thus, cartographic generalization of foreground data has to be achieved in real-time, requiring flexible, on-the-fly generalization algorithms [2].

Nearly all professional and academic cartographers consider the process of cartographic generalization to be one of the most intellectually and technically challenging components of mapmaking. Beginning with the writings of the German cartographer Max Eckert early in the twentieth century, the problems associated with generalization have been carefully articulated by most of the major researchers in cartography. In considering the process of digital cartographic generalization, nearly all applications of the process have as their first step the selection of objects and attributes from the initial database for representation. Once an object or attribute is initially selected, the generalization processes continues by the application of spatial or attribute transformations, respectively. Spatial transformations are those operators that alter the data representation from a geographical or topological perspective. Simplification and smoothing are the first two of ten or more operators. Simplification operators will select the characteristic, critical or shape-describing points to retain, or will reject the redundant points considered to be unnecessary to display the line's character. Smoothing operators produce a line with a more aesthetically pleasing caricature [3]. Detailed explanations and criticisms about line simplification and smoothing algorithms could be obtained from the numerous review papers produced by, for example, White [4], Weibel [5], McMaster [6], Brassel and Weibel [7], Thapa [8], Li [9,10] and Visvalingam and Williamson [11].

As shown in Figure 1, Li [10] classified the critical point detection algorithms into three major groups: corner detection, polygonal approximation, and a hybrid technique which is a combination of the first two.

The two commonly used techniques are sequential methods and iterative methods in polygonal approximation. Most algorithms for the detection of critical points appearing in the cartography/GIS literature belong to the polygonal approximation.

The sequential methods start from a point to find the longest allowable segment, within which all points along the original line can be neglected with a given criterion or tolerance or threshold. The last point on the longest allowable segment is considered as being a critical point.

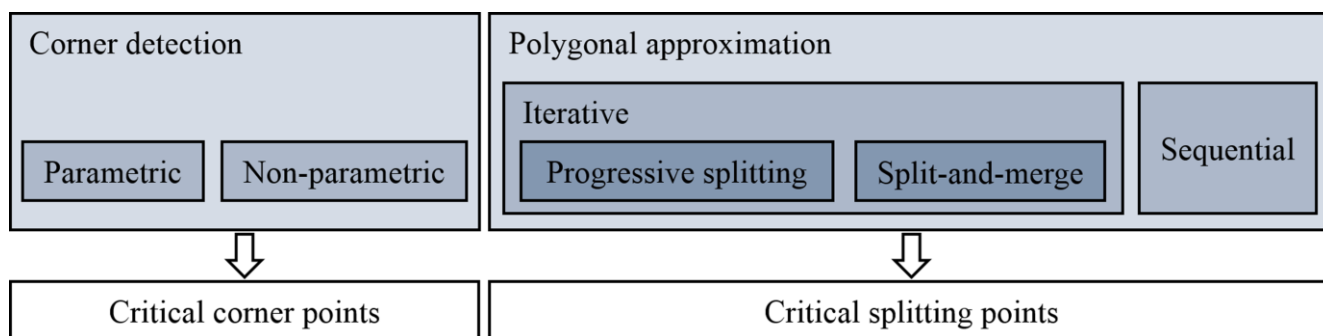


Figure 1. A classification of algorithms for detecting critical points.

There are two commonly used approaches in iterative methods, *i.e.*, progressive splitting and split-and-merge. The progressive splitting methods split the curve segment between two initial points into two parts by a partitioning point between them. This partitioning point is selected as a critical point. In the split-and-merge method, an arbitrary number of points along the line as initial critical points are assigned. For each pair of the critical points, the perpendicular distances of all points to the line joining these two critical points are computed. If any of the perpendicular distances is greater than the criterion, then the point with the largest perpendicular distance is selected as a critical point.

The common stages for corner detection techniques are estimating the curvature for each point on the curve, and locating the points which have local maximum (both positive and negative) curvatures as the corners accordance with the threshold and a process of non-maxima suppression. The detected corners are considered as the critical points on the curve. Corner detection algorithms are classified into two groups, *i.e.*, parametric methods and non-parametric methods. Parametric methods need one or more input parameters such as angle detection, while non-parametric methods design an adaptive approach to determine the region of support for each point on a curve without any input parameter.

Hydrography is the representation of water features such as streams/rivers, lakes/ponds, rapids, reservoirs, swamps/marshes, and dams/weirs on maps [12]. Hydrography is an important element in defining the shape and natural divisions of the Earth’s surface [13]. Cartographers from the Center of Excellence for Geospatial Information Science (CEGIS) of the U.S. Geological Survey (USGS), the University of Colorado Boulder and Pennsylvania State University used the ArcGIS Simplify Line tool, which is based on the Bend Simplify algorithm developed by Wang and Muller [14] to simplify selected (pruned) streams. In this algorithm, the most sensitive attribute affecting bend selection/elimination is size. The size of a bend is defined as the area of the polygon enclosed by the bend and its baseline. The shape of a bend polygon can be described by a compactness index (*cmp*), which, in turn, is defined as the ratio of the area of the polygon over the circle whose circumference length is the same as the length of the circumference of the polygon. For bend selection/elimination, the inverse of the *cmp* is used to adjust the bend size and as the primary criterion for bend selection [14]:

$$adjusted\ size = area \times \left(\frac{0.75}{cmp}\right) \tag{1}$$

In order to carry out bend generalization, the user must set a minimum diameter for a half-circle bend, and this minimum will be used as the tolerance and reference for bend elimination. The cutting action is executed iteratively [14].

The Bend Simplify algorithm requires a user-defined threshold value. The result of the Bend Simplify algorithm depends on the threshold value (*i.e.*, the user) directly. There is no unique result for a source dataset. The threshold value is determined via trial and error. Thus, evaluation of each result takes a long time. Furthermore, there is no object or criterion related to accuracy assessment in the Bend Simplify algorithm. Accordingly, some (parts) of the streams or lakes cannot meet the accuracy requirement.

Wang and Muller (1998) preserved the overall structure with line bends, which are mathematically defined according to size, shape and context. They claimed that no bends are detected by the well-known Douglas-Peucker algorithm [15], which measures the perpendicular distances of each intervening point as unique criteria for point selection. They defined a bend as the part of a line that contains a number of subsequent vertices, with the inflection angles on all vertices included in the bend being either positive or negative and the inflection of the bend's two end vertices being opposite signs.

Shahriari and Tao [16] proposed user-defined positional accuracy instead of supplying a tolerance value in the Douglas-Peucker algorithm for simplifying stream lines with different complexities. Thus, they aimed to specify tolerance values for each stream feature automatically.

Sinha *et al.* [17] developed a multi-criteria line simplification method to automate the simplification of linear hydrographic features based on spatial, topological, and hydrogeological constraints. Three criteria for the multi-criteria line simplification were the Douglas-Peucker distance, a change in topographic slope, and the log inverse distance to the nearest well. They used the multi-criteria score, which determines the probability of a point's selection.

Gary *et al.* [18] simplified streams in the 1:100,000 scale National Hydrography Dataset (NHD) to create a national 1:1,000,000-scale hydrography dataset using the Bend Simplify algorithm with a tolerance value of 500 meters. Similarly, waterbodies were simplified using this algorithm with a tolerance value of 500 meters.

Brewer *et al.* [19] simplified stream lines, artificial paths and secondary (unnamed) artificial paths using the Bend Simplify algorithm. The tolerance value 75 meters was used for both stream lines and secondary artificial paths, while 100 meters was used for artificial paths. Thus, they produced 1:50,000 scale maps from 1:24,000 scale maps for flat humid, flat dry, hilly humid, and mountainous humid regimes. Artificial paths are the centerlines of water areas include wide stream channels with two banks drawn. Secondary artificial paths belong to stream channels of which names are not known.

Wilmer and Brewer [20] simplified pruned streams in 1:5000 scale data using the Bend Simplify algorithm with 24-, 100-, and 2000-metre tolerance values for 1:24,000, 1:100,000, and 1:2,000,000 scales, respectively. Additionally, they derived the streams in 1:100,000 and 1:2,000,000 scales from 1:24,000-scale data using 100- and 2000-metre tolerance values, respectively.

Stanislawski and Bittenfield [21] studied a dry mountainous sub-basin watershed of the NHD that was generalized from a 1:24,000 scale, which is appropriate for cartographic mapping at scales between about 1:50,000 and 1:200,000. They also used the Bend Simplify algorithm to simplify polygon boundaries and stream lines (flowlines) with tolerance values of 200 and 150 meters, respectively.

Similarly, Stanislawski *et al.* [22] used the Bend Simplify algorithm to simplify streams in the NHD sub-basins compiled from 1:24,000-scale source materials. The geometric characteristics of features of

flat, hilly, and mountainous topographies of humid regions before and after using five simplification tolerance values (15, 25, 50, 100, and 200 meters), which are referenced in the baseline of the bends, are measured and evaluated using several metrics.

In this paper, a new algorithm, which is called hereafter Segment Simplify, is proposed for the simplification of streams and lakes. The Segment Simplify algorithm is based on the algorithm developed by Gökğöz [23] for contour simplification, which is called hereafter Contour Simplify. In both algorithms, it is aimed to obtain accurate and smooth features (*i.e.*, contours, streams, and lakes) to be depicted with a minimum number of characteristic points at the target scale. According to the Contour Simplify algorithm, simplification of a contour is to consecutively select and connect sufficient numbers of characteristic points in turn, beginning from the most characteristic point in addition to the first and the last contour point until falling inside the area of its error band. Characteristic points are ordered in accordance with the deviation angles. The contour point at which the value of the deviation angle is maximum becomes the most characteristic point of the contour. Boundary points of an error band are the end points of the steepest slope lines at both sides of each contour point. Length of each steepest slope line is limited to the mean square planimetric error at each contour point. Since contours are always represented as smooth curves on maps, a simplified contour is smoothed by laying a cubic spline onto the simplified contour points. The Segment Simplify algorithm mainly differs from the Contour Simplify algorithm in two aspects. Firstly, the most characteristic points of streams and lakes are positioned at their sharp bends, and they do not generally yield smooth depictions. On the other hand, there is no need to represent streams and lakes as smooth as contours. Therefore, in the Segment Simplify algorithm, some of the most characteristic points are eliminated, and cubic splines are not used for smoothing. Secondly, the steepest slope lines could not be used in construction of error bands in the Segment Simplify algorithm. Error bands are constructed as fixed width buffers by using the planimetric error of streams and lakes reported by National Map Agency. According to Li's classification mentioned above, the Segment Simplify algorithm could be considered as a parametric corner detection algorithm.

Segment Simplify algorithm is tested using high-resolution NHD data at a 1:24,000 scale of a sub-basin, Pomme de Terre, to obtain the corresponding medium-resolution data at a 1:100,000 scale. The results are compared to the corresponding medium-resolution NHD data and the results obtained using both Bend Simplify and Douglas-Peucker algorithms and several metrics.

2. Methodology

The line features to be represented at the target scale (1:100,000) had been obtained in a previous study [24]. This stage corresponds to the yellow part of the flowchart in Figure 2. Therefore, the selection/elimination process was performed on only the polygon features in the dataset, considering the USGS NHD standards. This stage corresponds to the blue part of the flowchart in Figure 2. In this stage, the polygon features with geometries that must be changed to lines were determined. The eliminated polygons that caused the network to be disconnected were also determined. As a result, both the polygons and the artificial paths to be simplified were obtained at the end of this stage. The simplification stage corresponds to the green part of the flowchart in Figure 2. In order to keep the connections between features after simplification, all selected line and polygon features were broken at connections. Consequently, all selected line and polygon features were changed to the line features composed of two

or more points without any connections, except their end points. They are hereafter called segments. Buffer polygons were created around the segments as their error bands in accordance with the United States National Map Accuracy Standards. All the segments were simplified by Segment Simplify algorithm. Finally, the target dataset was obtained and the simplified segments were merged according to their attributes, including stream types and levels. Each stage is explained in detail below. With the aim of comparison, all the segments were simplified by the Bend Simplify and Douglas-Peucker algorithms, as well.

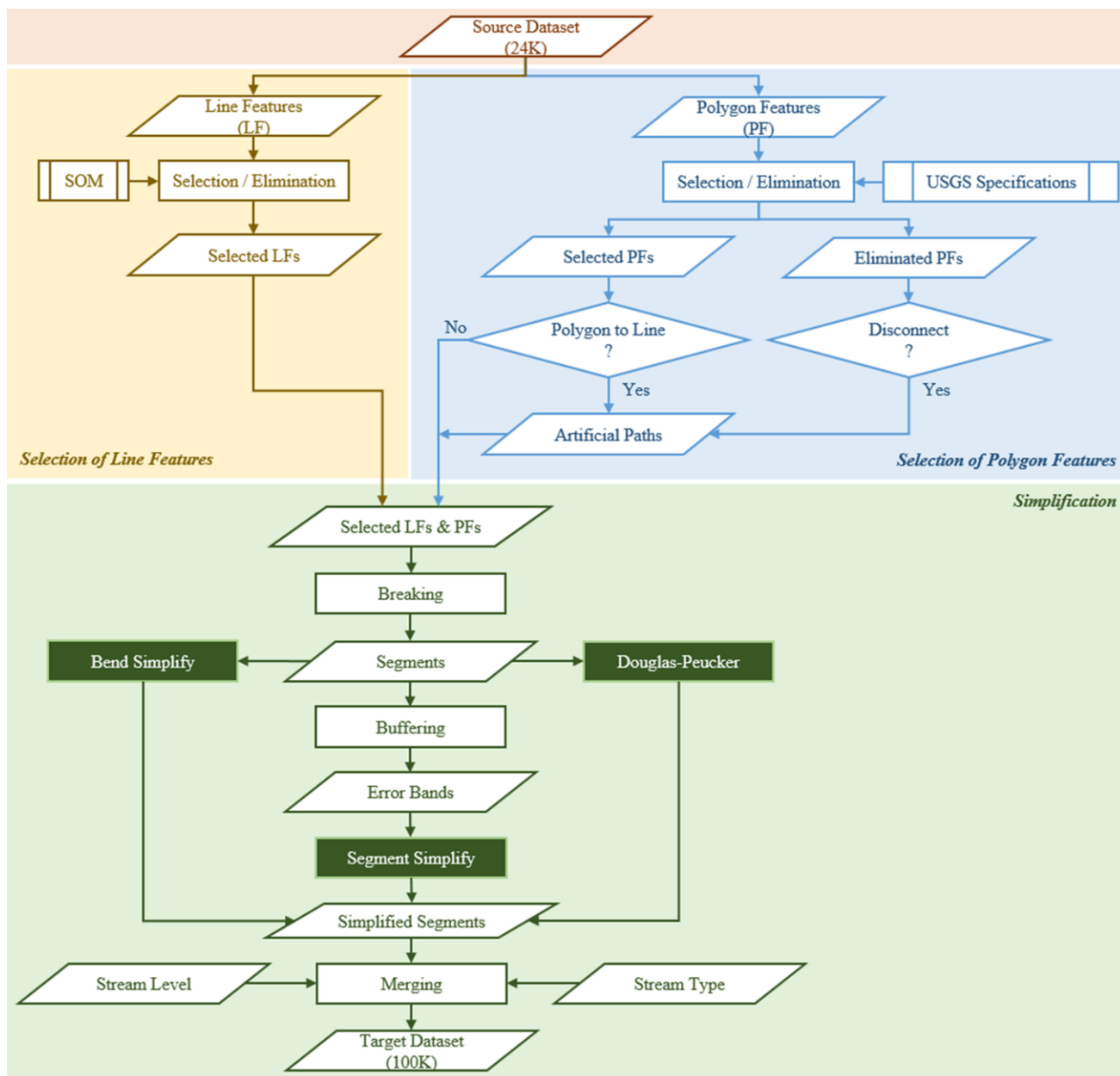


Figure 2. Flowchart of the study.

The processes of breaking, buffering, and merging were carried out by ArcGIS tools. Segment Simplify algorithm was performed automatically in AutoCAD using a program written by AutoLISP. Bend Simplify and Douglas-Peucker algorithms were performed by ArcGIS tools.

2.1. Data Source

Data used in this study, a high-resolution NHD (NHDH1029), and the corresponding medium-resolution NHD (NHDM1029) of a sub-basin Pomme de Terre (PT), were downloaded in June 2011. PT features a dendritic pattern. Terrain conditions affect the geometric characteristics of linear features for several feature types. In this context, PT is in a humid climate region. PT is in a hilly location on the Ozark Plateau in the interior highlands [25]. The data contain polygon and line features such as lakes/ponds, streams/rivers, and artificial paths. The source dataset is shown in Figure 3.

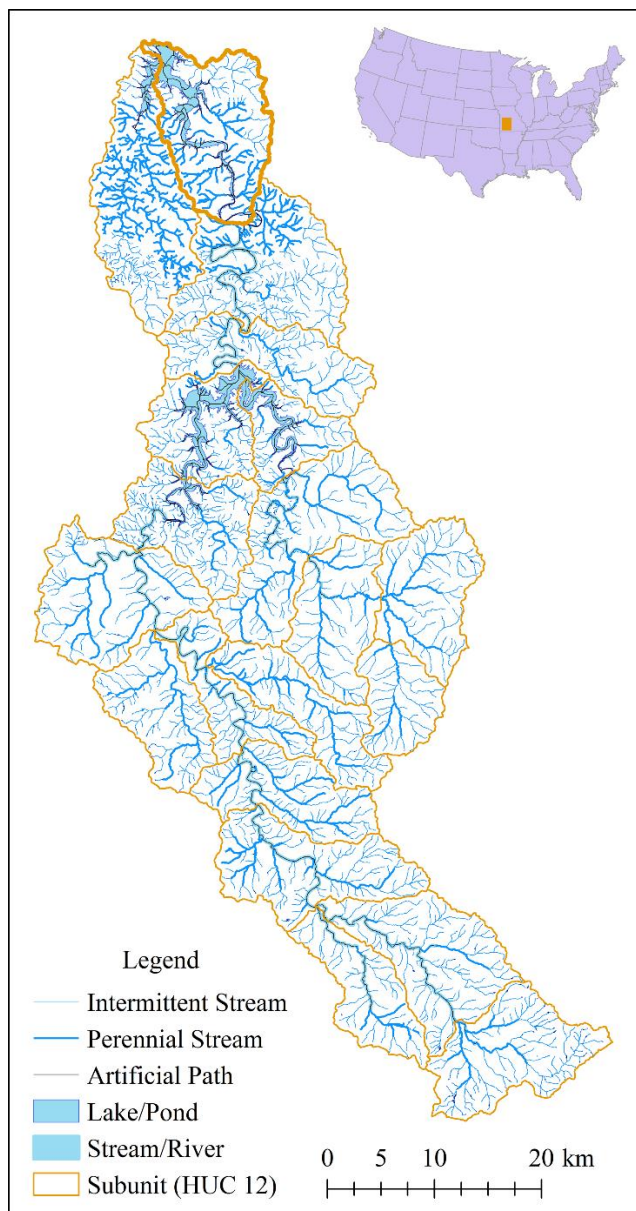


Figure 3. The source dataset of the study.

2.2. Selection/Elimination of Line Features

We selected the line features in accordance with the method proposed by Sen *et al.* [24]. In this method, self-organizing maps (SOM), an unsupervised method of artificial neural networks is used.

The most suitable attributes of the stream objects (*i.e.*, length, sinuosity, degree, betweenness, closeness, stream level, and type) were used as input variables. The attributes were weighted using Pearson's chi-square independence test. Töpfer's radical law [26] was used to determine how many features should be selected. An incremental approach was used to determine which clusters should be selected from the SOM.

2.3. Selection/Elimination of Polygon Features

We eliminated the polygon features according to the standards for NHD [27]. There are two main parameters used for representation and selection: the length and the width, which are shown as l and w in Table 1, respectively.

Table 1. Representation and selection rules of National Hydrography Dataset standards for the features shown in the study area.

Fcode	Name	Representation and Selection Rules
46006	Stream/River	If $l < 1600.2$ m then eliminate. If $w < 63.5$ m then represent as a line. If $63.5 \leq w < 101.6$ m and $l < 6705.6$ m then represent as a line.
39004	Lake/Pond	$w \leq 152.4$ m then eliminate.
34305	Dam/Weir	If $l < 304.8$ m then eliminate.

To determine the length and width of a feature that may be a regular or irregular polygon, excluding linear elongated stream-like polygons, oriented or non-oriented best fitting rectangles were used, respectively (Figure 4). An oriented best fitting rectangle is the smallest area enclosing a polygon, and a non-oriented best fitting rectangle is the envelope of a polygon with edges that are parallel to X and Y axes. We used the ArcGIS Minimum Bounding Geometry tool to obtain the best fitting rectangles.

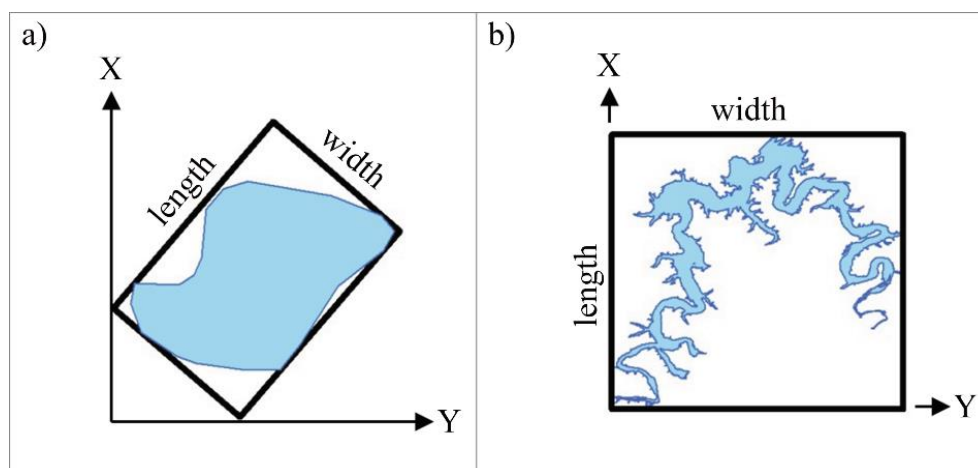


Figure 4. Best fitting rectangles. (a) Oriented for a regular polygon. (b) Non-oriented for an irregular polygon.

According to the method developed in this study, the width of a linear elongated stream-like polygon was determined as follows. The bisectors at the artificial path vertices, except the end points, were drawn as one centimeter lines on both sides. After that, they were extended to the edges of the polygon (Figure 5). The

width of a linear elongated stream-like polygon was computed by dividing the total length of all the lines by the number of vertices at which the bisectors were drawn. This process was managed automatically using a script written by AutoLISP.

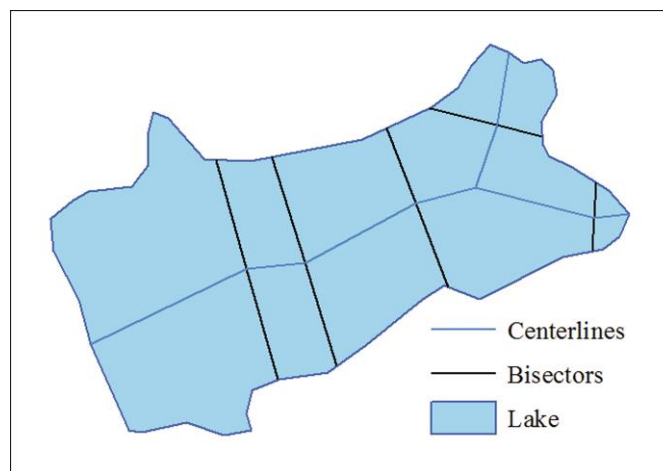


Figure 5. Bisectors at the artificial path vertices of a linear elongated stream-like polygon.

2.4. Simplification

The Segment Simplify algorithm was developed in this study with the following requirements. Firstly, the simplified segments should be inside the error bands (accuracy requirement). Secondly, the simplified segments should be smooth enough (legibility and aesthetic requirement). Thirdly, the simplified segments should be depicted with the minimum number of points (storage requirement). Here, the segments refer to all of the features available at the end of the breaking stage.

The first requirement is related to the planimetric accuracy of the segments. Since the topographical terrain is completely irregular, without any mathematical conformity, the geographic positions of segments on the maps are “certain” but not “true”. In other words, segments always have planimetric errors. Therefore, the true geographic position of a segment is anywhere inside a particular area of the map. These areas are assumed to be the error bands of the segments. In this study, buffer polygons are created around the segments to serve as error bands. Buffer distance is determined in accordance with the United States National Map Accuracy Standards. It is reported that 90% of well-defined streams are within 167 feet (0.02 inches at map scale) of their true geographic position at a 1:100,000 scale [28]. As a result, the buffer distance is 50.80 meters.

The second and third requirements are related to the characteristic points of the segments. Since these points are positioned at the sharp bends of a segment, they do not generally yield smooth depictions of the segment. Therefore, they should not be given priority when selecting points to obtain legible and aesthetic segments. On the other hand, storing redundant points is not a desirable option. Therefore, the minimum number of characteristic points should be selected in order to meet the storage requirement. It is clear that this is a complex problem and that the characteristic points of a segment should be defined first.

The points with maximum curvatures can be assumed to be the characteristic points of the segments. According to this assumption, the characteristic points of segments are directly determined, but it is difficult to use the curvatures at segment points. Various methods can be used, such as cubic spline or parabolic blending, but each yields a different result. Therefore, different values for each curvature at a

segment point could be obtained. Moreover, characteristic points and their order, which is determined by the curvatures, could be different each time. For these reasons, in this algorithm, deviation angles are computed rather than the curvatures at segment points. The angle between two consecutive parts of a segment is assumed to be the deviation angle at a segment point (Figure 6).

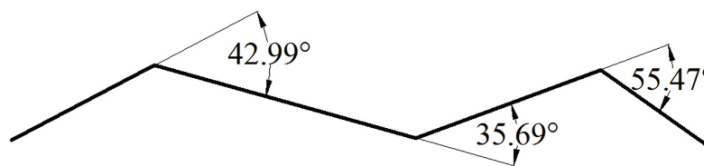


Figure 6. Deviation angles at segment points.

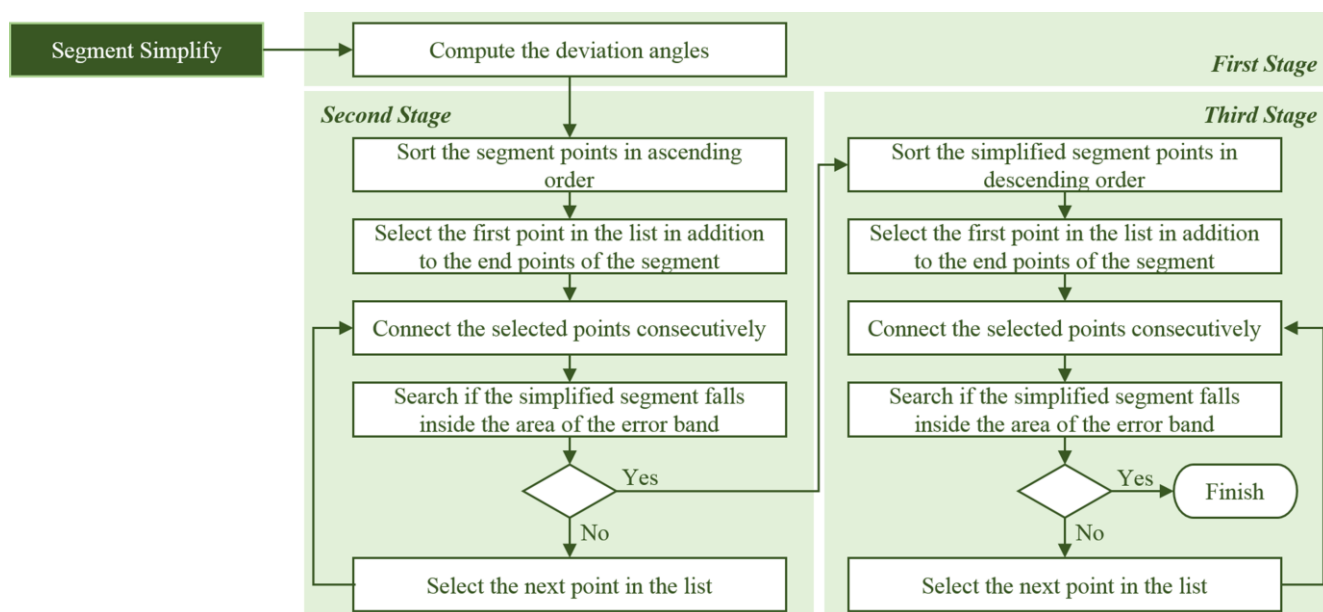


Figure 7. Flowchart of the Segment Simplify algorithm.

As shown in the flowchart of the algorithm (Figure 7), a segment is simplified in three successive stages. In the first stage, deviation angles of a segment are computed. The second stage is focused on the first two requirements. Therefore, the second stage begins with the sorting of segment points in ascending order with respect to the deviation angles. In addition to the end points of the segment, the first point in the list, the least characteristic point of the segment, is selected. The selected points are connected consecutively and, thus, a simplified segment is obtained. To determine whether the simplified segment falls inside the error band, it is necessary to discover whether there is any intersection between the simplified segment and the boundary of the error band. If there is an intersection, the simplified segment is not completely inside the error band. In this case, the next point in the list, the next least characteristic point of the segment, is selected and included in the set of the selected points. After this, one returns to the connection step. These steps are repeated until the simplified segment completely falls inside the area of the error band. The simplified segment obtained in this way (*i.e.*, the output of the second stage) is the input of the third stage, along with the error band. The third stage focuses on the first and third requirements. Therefore, the third stage begins with the sorting of the simplified segment points in descending order of deviation angles. This is only the difference between the second and the

third stage. The remaining steps of the third stage are executed in the same manner as the second stage using a new list of points. Consequently, a simplified segment meeting all the requirements mentioned above is obtained.

In Figure 8a, an original segment (red) composed of 92 points and its error band (grey) is shown. After the second and the third stages of the algorithm, the number of points is reduced to 53 and 29, respectively. In other words, 42.4% of the points of the original segment were eliminated in the second stage, and 45.3% of the points of the simplified segment obtained at the end of the second stage were eliminated in the third stage. In total, 68.5% of the points of the original segment were eliminated by the algorithm. The results of the second and third stages are shown in Figure 8b–d.

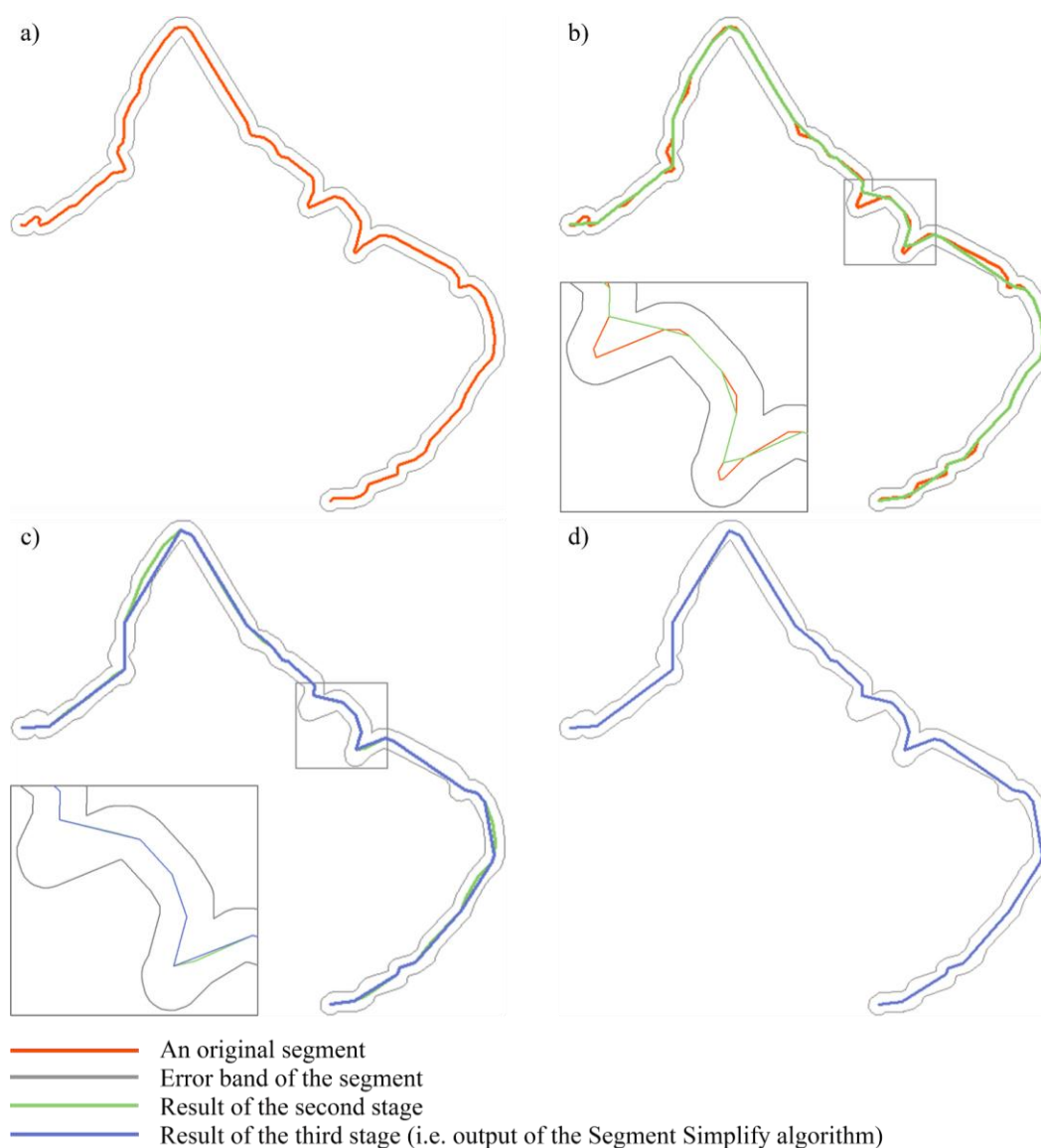


Figure 8. The stages of the Segment Simplify algorithm. (a) An original segment and its error band. (b) The result of the second stage. (c) The result of the third stage. (d) The output of the algorithm.

3. Results and Discussion

This methodology was used to perform a computational experience which is mainly illustrated in Figure 9. The network in Figure 9a was obtained using the Bend Simplify algorithm, setting 100 and 200 meters as tolerance values for the streams and lakes, respectively. The network in Figure 9b was also obtained with the Bend Simplify algorithm, but the tolerance values for the streams and the lakes were set as 150 and 200 meters, respectively. The tolerance values for the Bend Simplify algorithm are suggested by Wilmer and Brewer [20], Stanislawski and Battenfield [21] and Stanislawski *et al.* [22]. The network in Figure 9c was obtained with the Douglas-Peucker algorithm, setting the tolerance value as 25.4 meters, which is the half of the buffer distance, *i.e.*, the error band, used in the Segment Simplify algorithm. The network obtained by the Segment Simplify algorithm is shown in Figure 9d. The USGS dataset at 100K scale is shown in Figure 10.

The numbers of the features in the networks at 24K and 100K scales, in addition to the values computed by Töpfer's formulas, are given in Figure 11a. Since all the networks obtained in this study consist of the same features, the numbers of the features are represented by the "Study" bar. All the streams depicted as polygon features in the USGS network at 24K scale were changed to the line features at the stage of selection/elimination in accordance with USGS specifications. All the streams are already depicted as line features in the USGS network at 100K scale. As a result, there is no conflict between the networks from the view of the feature type of the streams. However, the segment numbers of streams in the networks (*i.e.*, the USGS network at 100K scale and the networks obtained in this study) are not the same. The segment number of streams in the USGS network at 100K scale is less than in this study. Whereas the cartographer selected 20% of the segments in the USGS network at 24K scale, we selected 22% of them. The reason for this is that different methods were used for selection/elimination. Unlike the USGS, an unsupervised method of artificial neural networks (*i.e.*, SOM) was used for the selection/elimination of the streams in this study. The segment number of streams computed by Töpfer's formula is larger than the segment numbers of streams for both networks at 100K scale. Töpfer's formula used for the streams is:

$$n_f = n_a \sqrt{(M_a/M_f)^2} \quad (2)$$

where n_f is the number of features at the target scale, n_a is the number of features at the source scale, M_a is the scale denominator of the source map, and M_f is the scale denominator of the target map.

There is a considerable difference between the lake numbers in both networks at 100K scale. While the cartographer selected 8% of the lakes in the USGS network at 24K scale, we only selected 1% of them. The numbers of lakes in both networks at 100K scale are considerably smaller than the value computed by Töpfer's formula below:

$$n_f = n_a \sqrt{(M_a/M_f)^3} \quad (3)$$

The results of the quantitative analysis for assessing the performance of the Segment Simplify algorithm are given in Figure 11b–d. While the vertex number of the streams in the network obtained by the Bend Simplify algorithm using 100 meters as the threshold value is larger than that of the USGS network at 100K scale, the vertex number of the streams in the network obtained by the Bend Simplify

algorithm using 150 meters as the threshold value is smaller than that of the USGS network at 100K scale (Figure 11b). Moreover, the Douglas-Peucker algorithm retained the least number of stream vertices. The vertex number of the streams in the network obtained by the Segment Simplify algorithm is between the values of the Bend Simplify and Douglas-Peucker algorithms. According to the value computed by Töpfer's formula below, all the results are acceptable. However, in terms of the accuracy assessment, the results obtained by both the Segment Simplify and Douglas-Peucker algorithms are acceptable. All the simplified streams obtained by the Segment Simplify and Douglas-Peucker algorithms are completely inside the error bands, whereas some of the simplified streams obtained by the cartographer or the Bend Simplify algorithm are not (Figure 12).

$$n_f = n_a \sqrt{(M_a/M_f)} \quad (4)$$

The lakes were simplified by the Bend Simplify algorithm using 200 meters as the threshold value [20]. As seen in Figure 11b, the vertex number of the lakes simplified by the Douglas-Peucker algorithm is smaller than that obtained by the USGS cartographer, the Bend Simplify and Segment Simplify algorithms. When simplified by the Segment Simplify algorithm, the vertex number of the lakes is smaller than that of the USGS cartographer but larger than that using the Bend Simplify and Douglas-Peucker algorithms. The vertex numbers of the lakes in all the networks are smaller than the value computed by Töpfer's Equation (4). As a result, all the networks are satisfactory in terms of the vertex number of the simplified lakes. However, the results (*i.e.*, the simplified lakes) obtained by the Segment Simplify and Douglas-Peucker algorithms are acceptable from the standpoint of accuracy assessment. All the simplified lakes obtained by the Segment Simplify and Douglas-Peucker algorithms are completely inside the error bands, whereas some of the simplified lakes obtained by the cartographer or the Bend Simplify algorithm are not.

The total length of the simplified streams obtained by the Bend Simplify algorithm using 100 meters as the threshold value is closer to that of the USGS streams at 100K scale, whereas the total length of the simplified lakes obtained by the Douglas-Peucker algorithm is closer to that of the USGS streams at 100K scale (Figure 11c). Additionally, the total length of the simplified lakes obtained by the Segment Simplify algorithm is between the values of the Bend Simplify and Douglas-Peucker algorithms.

As seen in Figure 11d, the sinuosity value of the simplified segments obtained by the Segment Simplify algorithm is closer to that of the USGS segments at 24K scale, though there is no considerable difference between any of them. As a result, the Segment Simplify algorithm also produced the most satisfactory result in regard to the sinuosity criterion; it maintained the character of the original segments.

As a summary, the Douglas-Peucker algorithm retained the least numbers of stream and lake vertices. Furthermore, the simplified streams obtained by the Segment Simplify algorithm were the minimum in total length, whereas the simplified lakes obtained by the Bend Simplify algorithm were the minimum in total length. The vertex number of the simplified streams obtained by the Bend Simplify algorithm using 150 meters as the threshold value is closer to that of the USGS streams at 100K scale, whereas the vertex number of the simplified lakes obtained by the Segment Simplify algorithm is closer to that of the USGS streams at 100K scale. The values computed by Töpfer's formulas for the vertex numbers of the streams and lakes at 100K scale deviate widely from all the related values in this study. Therefore, the values computed by Töpfer's formulas are not helpful in the quantitative analysis.

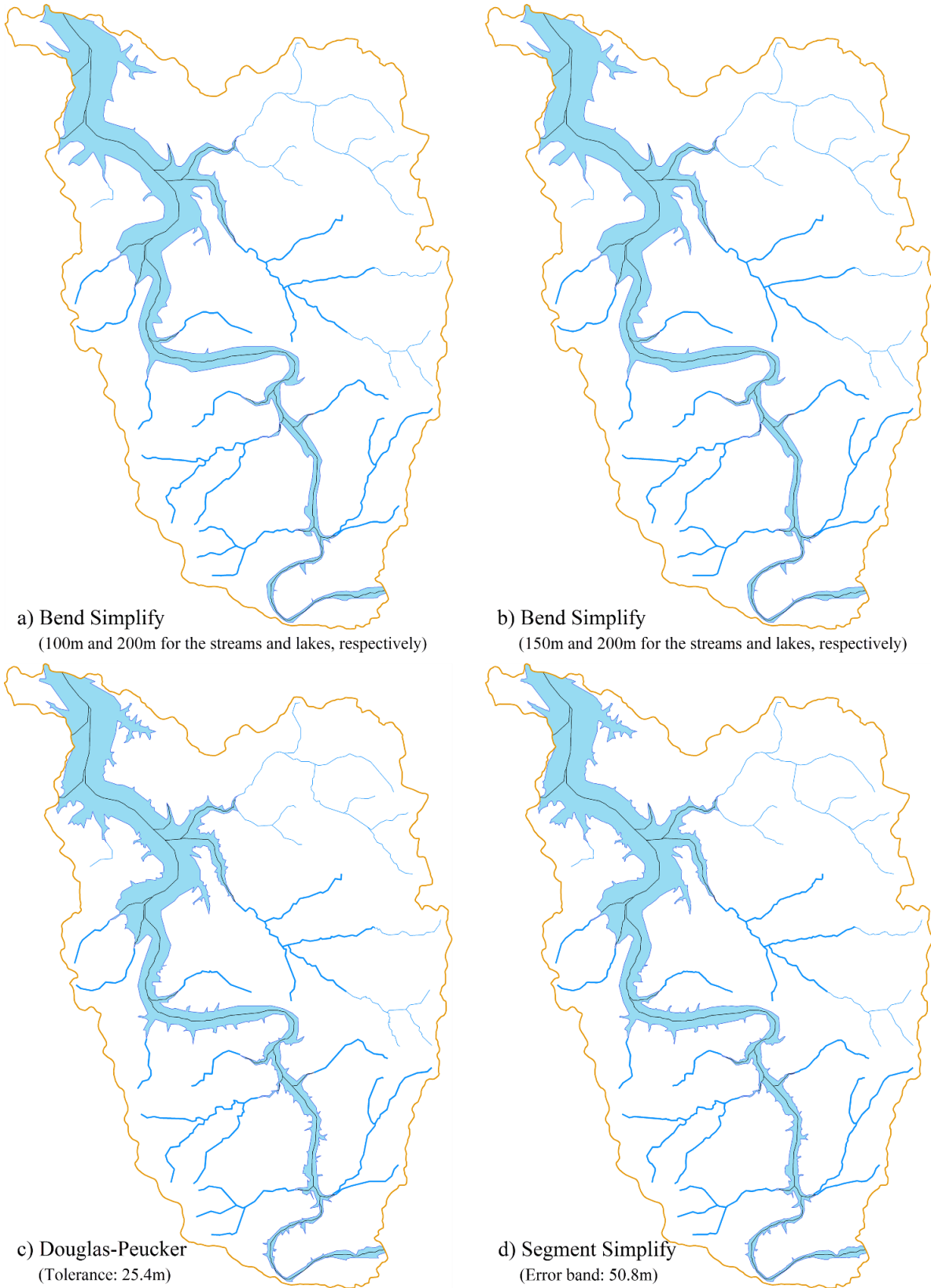


Figure 9. The outputs of the algorithms. (a) Bend Simplify (Tolerances: 100m and 200m for the streams and lakes, respectively). (b) Bend Simplify (Tolerances: 150m and 200m for the streams and lakes, respectively). (c) Douglas-Peucker (Tolerance: 25.4m). (d) Segment Simplify (Error band: 50.8m).

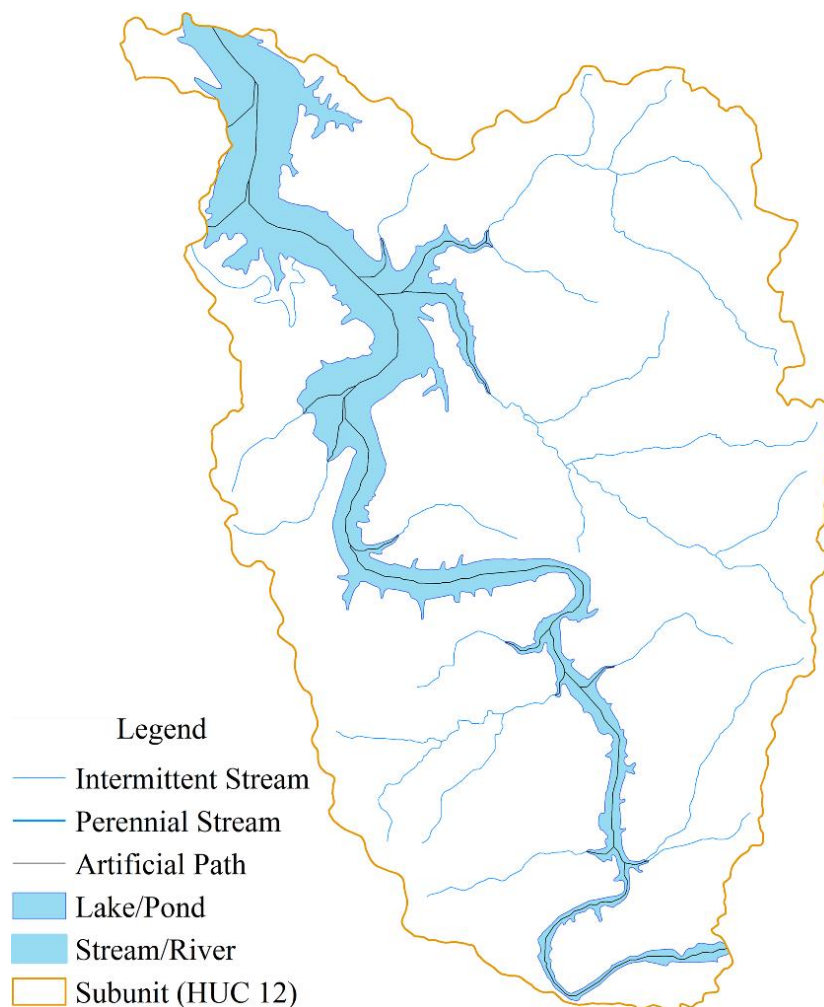


Figure 10. The USGS dataset at 100K scale.

Since the Douglas-Peucker algorithm selects the farthest points in each case, the intervening spaces between two sides of some bends approach coalescence at the target scale. In other words, since the intervening spaces between two sides of some bends are less than 0.25 mm, which is the graphic limit for the intervening space between two lines at the target scale [29], the Douglas-Peucker algorithm produces some sharp bends which are illegible at the target scale. For example, the bend with the intervening space of 0.13 mm, which is less than the graphic limit, in Figure 13a was presented at the target scale by the Douglas-Peucker algorithm (Figure 13c). As a result, the Douglas-Peucker algorithm caused the legibility problem in some bends at the target scale in this study. Accordingly, the Douglas-Peucker algorithm did not maintain the aesthetic quality of those features. These were the main weaknesses of the Douglas-Peucker algorithm. However, the Segment Simplify algorithm does not select the farthest points and, thus, it did not cause such problems in this study.

The Bend Simplify algorithm produced the simplified features which were smoother than that of the Douglas-Peucker and the Segment Simplify algorithms by removing some bends which were already legible at the target scale. For example, the bend with the intervening space of 1.11 mm, which is larger than the graphic limit, in Figure 13a was not presented at the target scale by the Bend Simplify algorithm (Figure 13b). On the other hand, the Bend Simplify algorithm retained many more vertices. In this

context, it seems that some of the vertices retained by the Bend Simplify algorithm could be less characteristic, critical or shape-describing.

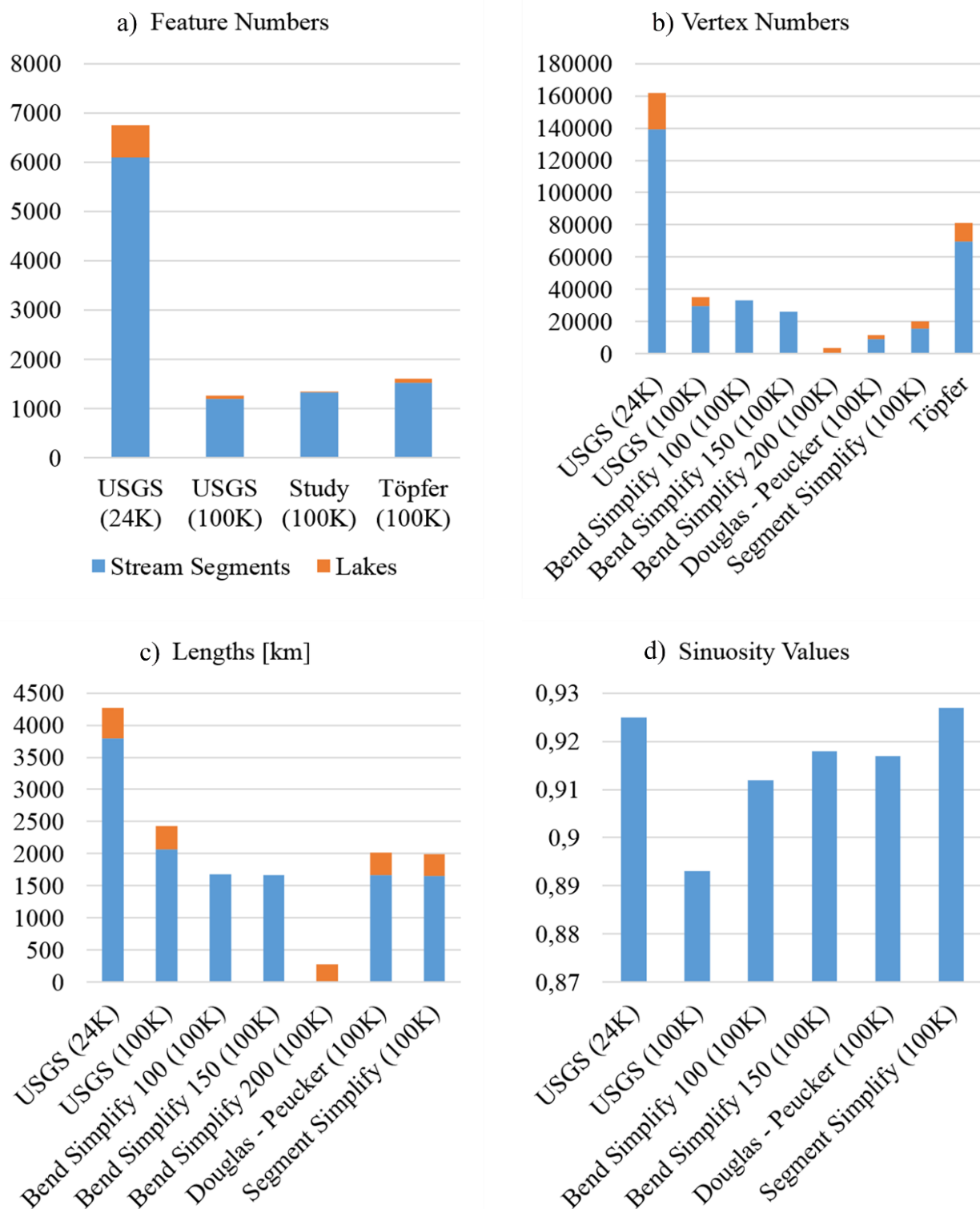


Figure 11. The statistics of the experiment. (a) Feature numbers. (b) Vertex numbers. (c) Lengths (km). (d) Sinuosity Values.

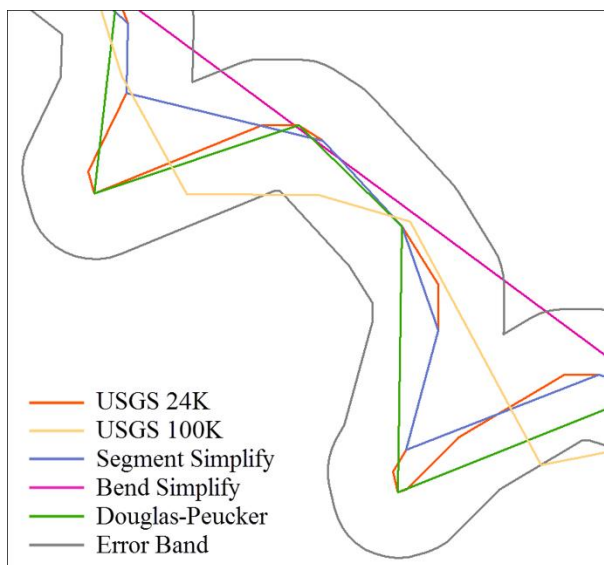


Figure 12. The originals and the outputs of the algorithms.

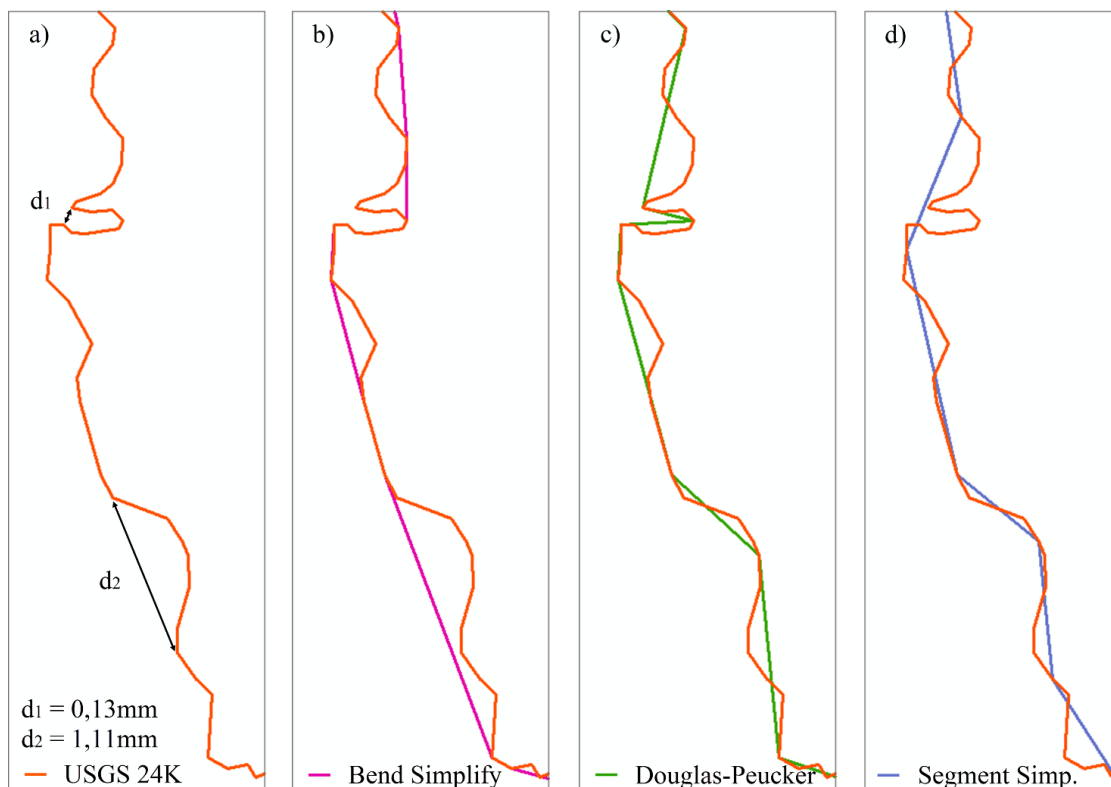


Figure 13. Behaviors of the algorithms at several bends. (a) An original stream. (b) Bend Simplify (Tolerance: 150m). (c) Douglas-Peucker (Tolerance: 25.4m). (d) Segment Simplify (Error band: 50.8m).

4. Conclusions

We have presented a new algorithm, called Segment Simplify, for the simplification of streams and lakes. The error bands of streams or lakes are the objects used for the accuracy assessment in the Segment Simplify algorithm. The Segment Simplify algorithm does not allow a simplified stream or lake to go out of its error band. Therefore, all simplified streams or lakes obtained by the Segment Simplify algorithm

are always inside the error bands, and there is no doubt as to their accuracy. In the Segment Simplify algorithm, the most characteristic ones of the least characteristic points of a stream or lake are selected to an extent. In other words, the most characteristic points of a stream or lake are eliminated to an extent. Therefore, the bends of a simplified stream or lake are not as sharp or smooth as that of the other algorithms. In this context, the Segment Simplify algorithm produces legible and aesthetic features. The Segment Simplify algorithm eliminates a considerable amount of points. The Segment Simplify algorithm runs automatically, does not have any user-defined parameters, and produces a unique result for a source dataset.

As a future study, the Segment Simplify algorithm may be performed on other data which belongs to, for example, an arid climate region with a trellis pattern. Furthermore, a study should be designed to determine the width of error bands to be used by the Segment Simplify algorithm when using geographical coordinates instead of projected coordinates at a certain scale.

Acknowledgments

Authors would like to thank United States Geological Survey for the datasets, and Barbara Battenfield (University of Colorado at Boulder) and Larry Stanislawski (Center of Excellence for Geospatial Information Science) for their support. Authors are also grateful to the anonymous reviewers whose comments helped to improve the paper.

Author Contributions

Türkey Gökğöz and Alper Sen conceived the study, wrote the paper and interpreted the results. Abdulkadir Memduhoglu and Muslum Hacı designed the study, performed the experiments and analysed the data.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Chaudhry, O.; Mackaness, W.A. Automatic identification of urban settlement boundaries for multiple representation databases. *Comput. Environ. Urban Syst.* **2008**, *32*, 95–109.
2. Bereuter, P.; Weibel, R. Algorithms for on-the-fly generalization of point data using quadtrees. In Proceedings of AutoCarto 2012, Columbus, OH, USA, 16–18 September 2012.
3. McMaster, R.B.; Shea, K.S. *Generalization in Digital Cartography*; Association of American Geographers: Washington, DC, USA, 1992.
4. White, E.R. Assessment of line-generalization algorithms using characteristic points. *Am. Cartogr.* **1985**, *12*, 17–27.
5. Weibel, R. Automated cartographic generalization. In *A Selected Bibliography on Spatial Data Handling: Data Structures, Generalization and Three-Dimensional Mapping*; Sieber, R., Brassel, K.E., Eds.; Geographisches Institut der Universität Zürich: Zürich, Switzerland, 1986; pp. 20–35.
6. McMaster, R.B. Automated line generalization. *Cartographica* **1987**, *24*, 74–111.

7. Brassel, K.E.; Weibel, R. A review and conceptual framework of automated map generalization. *Int. J. Geogr. Inf. Syst.* **1988**, *2*, 229–244.
8. Thapa, K. A review of critical points detection and line generalization algorithms. *Surv. Mapp.* **1988**, *48*, 185–205.
9. Li, Z. An algorithm for compressing digital contour data. *Cartogr. J.* **1988**, *25*, 143–146.
10. Li, Z. An examination of algorithms for the detection of critical point on digital cartographic lines. *Cartogr. J.* **1995**, *32*, 121–125.
11. Visvalingam, M.; Williamson, P.J. Simplification and generalization of large scale data for roads: A comparison of two filtering algorithms. *Cartogr. Geogr. Inf. Sci.* **1995**, *22*, 264–275.
12. Raisz, E. *Principles of Cartography*; McGraw-Hill: Tokyo, Japan, 1962.
13. Loon, J.C. Cartographic Generalization of Digital Terrain Models. Ph.D. Thesis, The Ohio State University, Columbus, OH, USA, 1978.
14. Wang, Z.; Muller, J.C. Line generalization based on analysis of shape characteristics. *Cartogr. Geogr. Inf. Sci.* **1998**, *25*, 3–15.
15. Douglas, D.H.; Peucker T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Can. Cartogr.* **1973**, *10*, 112–122.
16. Shahriari, N.; Tao, C.V. Minimizing positional errors in line simplification using adaptive tolerance values. In *Advances in Spatial Data Handling*; Richardson, D.E., Oosterom, P.V., Eds.; Springer: Berlin, Germany, 2002; pp. 153–166.
17. Sinha, G.; Silavisesrith, W.; Craig, J.R.; Flewelling, D.M. Multicriteria Line Simplification (MCLS) for AEM groundwater modeling. In Proceedings of the 6th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Science, Portland, ME, USA, 28 June–1 July 2004.
18. Gary, R.H.; Wilson, Z.D.; Archuleta, C.M.; Thompson, F.E.; Vrabel, J. *Production of a National 1:1,000,000-Scale Hydrography Dataset for the United States—Feature Selection, Simplification, and Refinement*; U.S. Geological Survey: Reston, VA, USA, 2009; p. 22.
19. Brewer, C.A.; Buttenfield, B.P.; Usery, E.L. Evaluating generalizations of hydrography in differing terrains for the national map of the United States. In Proceedings of the 24th International Cartographic Conference (ICC 2009), Santiago, Chile, 15–21 November 2009.
20. Wilmer, J.M.; Brewer, C.A. Application of the radical law in generalization of national hydrography data for multiscale mapping. In Proceedings of the Special Joint Symposium of ISPRS Technical Commission IV & AutoCarto in Conjunction with ASPRS/CaGIS 2010 Fall Specialty Conference, Orlando, FL, USA, 15–19 November 2010.
21. Stanislawski, L.V.; Buttenfield, B.P. Hydrographic feature generalization in dry mountainous terrain. In Proceedings of the Special Joint Symposium of ISPRS Technical Commission IV & AutoCarto in Conjunction with ASPRS/CaGIS 2010 Fall Specialty Conference, Orlando, FL, USA, 15–19 November 2010.
22. Stanislawski, L.V.; Raposo, P.; Howard, M.; Buttenfield, B.P. Automated metric assessment of line simplification in humid landscapes. In Proceedings of the AutoCarto 2012, Columbus, OH, USA, 16–18 September 2012.
23. Gökğöz, T. Generalization of contours using deviation angles and error bands. *Cartogr. J.* **2005**, *42*, 145–156.

24. Sen, A.; Gokgoz, T.; Sester, M. Model generalization of two different drainage patterns by self-organizing maps. *Cartogr. Geogr. Inf. Sci.* **2014**, *41*, 151–165.
25. Stanislawski, L.V.; Finn, M.; Barnes, M.; Usery, E.L. Assessment of a rapid approach for estimating catchment areas for surface drainage lines. In Proceedings of 2007 ACSM-IPLSA-MSPS, St. Louis, MO, USA, 9–12 March 2007.
26. Töpfer, F.; Pillewiser, W. The principles of selection. *Cartogr. J.* **1966**, *3*, 10–16.
27. United States Geological Survey (USGS). The National Hydrography Dataset: Concepts and Contents. Available online: http://nhd.usgs.gov/chapter1/chp1_data_users_guide.pdf (accessed on 17 June 2015).
28. United States Geological Survey (USGS). The National Hydrography Dataset: Frequently Asked Questions. Available online: http://nhd.usgs.gov/nhd_faq.html#q108 (accessed on 17 June 2015).
29. Rytz, A.; Bantel, E.; Hoinkes, C.; Merkle, G.; Schelling, G. *Kartographische Generalisierung: Topographische Karten*; Schweizerische Gesellschaft für Kartographie: Bern, Switzerland, 1975. (In German)

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).