

# A NEW LINE SIMPLIFICATION METHOD

Ehsan Mohammadi <sup>a,\*</sup>

<sup>a</sup>Zanjan University, Surveying Faculty, Zanjan, Iran – mohammadihsan@yahoo.com

THE 5<sup>TH</sup> INTERNATIONAL SYMPOSIUM ON MOBILE MAPPING TECHNOLOGY MMT'07

**KEY WORDS:** polyline, simplification, cartography, algorithm, optimization, small scale

## ABSTRACT:

Line simplification is one of the generalization methods, used when converting large scale maps to smaller ones. This method is applied on linear features and boundaries of polygons. Most algorithms work on the vertices of the linear features. These algorithms accept some vertex points to remain on their position and ignore the others. So, all of the vertex points have two states. They are either part of the final simplified polyline or not. The difference between algorithms is in the method of accepting and ignoring vertex points. This paper introduces a new method in which all of the points accompany in making the final simplified polyline. The result in this method is unique and even for polygons it gives a unique result. It is also faster than other methods and is flexible enough to let the vertex points of original polyline have weights in shaping the final result.

## 1. EXISTING ALGORITHMS

Need for multi-scale maps in mobile mapping and web-GIS applications imposes developers to use different generalization operations, including polyline-simplification methods. A major objective when producing a multi-scale map is the representation of linear features and boundaries of polygons, in smaller scales, using less possible vertices. Vertices in linear features act as the main skeleton of the feature. The more vertices in a feature, the smoother the feature will be illustrated. But in small scale maps, all vertices are not essentially useable, because most of them can not be seen in such scales. So, it is useless to attempt to show all vertices in a small scale map. Besides, the memory when representing a polyline in a small scale digital map should not be occupied unnecessarily with points that have no effect in showing the polylines. So, some points in a linear feature have to be removed. But which ones?

There are different line-simplification algorithms available, which are used in different applications. These algorithms may be divided into a number of classes:

1- Independent point routines: these algorithms are very simple in nature and do not take into account the mathematical relationship of neighboring coordinate points. An example of this is the nth point routine where every nth coordinate pair (e.g. 2<sup>nd</sup>, 4<sup>th</sup>, 6<sup>th</sup> etc.) is retained. These routines are very computationally efficient, but they are unacceptable for accurate mapping applications.

2- Localized processing routines: these algorithms utilize the characteristics of the immediate neighboring coordinate points in deciding whether to retain coordinate pairs. An example of these algorithms uses the minimum

Euclidean distance between two coordinate pairs. If the distance between two coordinate pairs is less than a predefined minimum, one pair is rejected.

3- Extended local processing routines: these algorithms search beyond the immediate neighboring coordinate pairs and evaluate sections of the polyline. The algorithms are often constrained by the number of coordinates and a distance parameter (tolerance). An example algorithm of this type was first described by Lang (1969) and was reported by Douglas and Peucker (1973) as producing an acceptable result. Another algorithm, reported by Jenks (1989), calculates the perpendicular distance from a polyline connecting two coordinate pairs to an intermediate coordinate pair.

4- Global routines: the previous algorithms process a polyline piece by piece from beginning to end. The fourth class of algorithms describe a different, holistic approach to simplification. A global routine considers the polyline in its complete shape while processing. The only global simplification algorithm commonly used in cartography was developed by Douglas and Peucker (Poiker) in 1973. This algorithm is both mathematically and perceptually superior. It produces the best results in terms of both vector and area displacement. Many cartographers consider it to be the most accurate simplification algorithm available, while others think it is too slow and costly in terms of computer processing time. [George Taylor]

All of the mentioned algorithms choose some of the original points and leave the others. The difference between the algorithms is in the method of choosing points. In all of above

algorithms each point has one of two states: It is either part of the simplified polyline or not.

## 2. NEW ALGORITHM

The new method introduced in this paper is not looking at the simplification as a method of choosing some points and leaving the rest. It actually tends to use all of the points that shape a polyline. In other words, all of the original points in a linear feature have a share in the result of simplification.

The basis of the new method is on the best-fitting polyline algorithm. Each linear feature is made up of a series of points with (x, y) coordinates in a 2D Cartesian coordinate system. (Figure 1.)

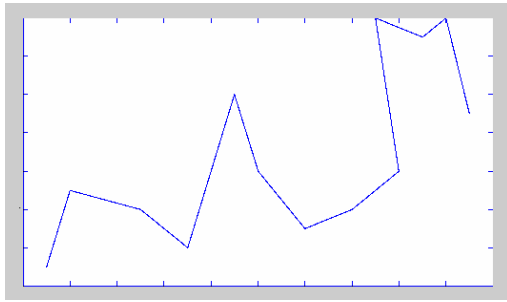


Figure 1. A polyline is made up of a series of points

The steps of simplifying in this method are as follow:

1- A rotation is applied to all points, in order to equalize y coordinate of the first and the last point in the polyline. In polygons rotation is in a way that the greatest diagonal of the polygon becomes parallel to x axis.

2- A line is fitted to the vertices of the original linear feature. The equation of this line is:

$$y'=ax+b. \quad (1)$$

But no line passes exactly through a series of points that are not aligned. In order to have the best-fitting line the values of "a" and "b" in the above equation must be determined in a way that:

$$\Sigma(y_i-y_i')^2 \rightarrow \min \quad (2)$$

in which:

i is the index of points shaping the polyline,  
 $y_i$  is the y coordinate of each point shaping the polyline, and  
 $y_i'$  is the y coordinate of each point on the best fitting line.

In order to solve the above equation the two next equations must be solved:

$$\begin{aligned} \delta\Sigma/\delta a &= 0 \\ \delta\Sigma/\delta b &= 0 \end{aligned} \quad (3)$$

Two above equations give the values for two unknown parameters of the best-fitting line: "a" and "b". The achieved line passes among all points, but none of the points may pose exactly on the line. Some of them may fall on the right of the line and others on the left.

3- The difference between original y coordinate of each point and the correspondent  $y'$  coordinate of each point on the best-fitting line is calculated:

$$d_i = y_i - y_i' \quad (4)$$

4- Each difference ( $d_i$ ) is compared with a predefined tolerance (t). In this method, like some other methods, a known tolerance must be introduced to the algorithm. So:

- 1- if  $d_i > t$  the point remains in its original position to be acted upon in the next step.
- 2- else, if  $d_i \leq t$  the  $y_i$  of the point is replace with  $y_i'$ .

4- The consecutive points which have not moved from their original place and remain on same sides are chosen and put in new groups. It may result up to n group in which n is the number of original points.

5- If the number of points in each group is two or less, the points will remain in their original places. Else, another best-fitting polyline will be calculated for each group of points and the steps of 2, 3, 4 and 5 will be accomplished, till all points either remain in their original places or move on a best-fitting line which is determined from the coordinates of those points.

6- In the last step all the points are rotated back, in opposite direction from the rotation in the first step.

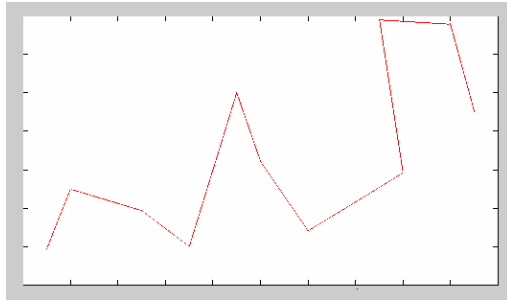


Figure 2. Result of simplification

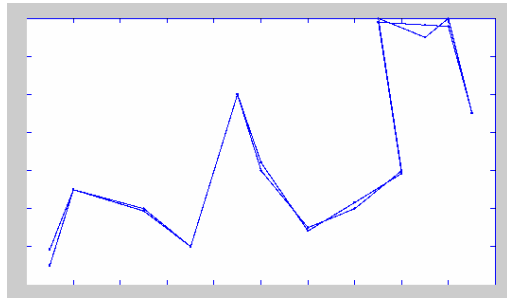


Figure 3. An overlay of original polyline and simplified polyline

The flowchart of the algorithm is shown in figure 4.

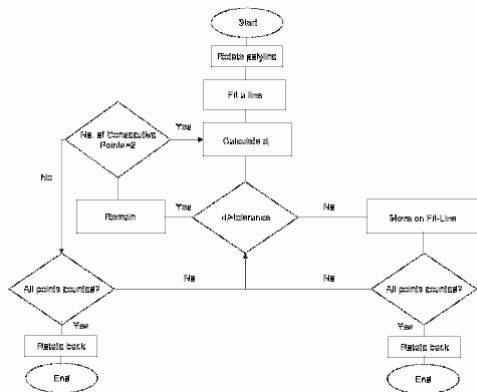


Figure 4. Flowchart of the algorithm

### 3. WEIGHED POINTS

Points enjoy a great benefit in this method. Each point can have a weight, which means how much a point can be effective in presenting the final simplified line. Contrary to previous methods in which all points just could have two states in shaping a line, the new method allows all the points to have a changeable ratio in shaping the final line.

To implement the weight for each point, beside the coordinate of points, a value should be introduced to the algorithm as the weight of the point. So each point is presented with three measures: (x, y, w).

The weight of points affects the slope of best fitting line and its distance from the points. In order to

apply the weight of points in the result of best-fitting line the equation (2) will be changed as follow:

$$\sum w_i * (y_i - y_i')^2 \rightarrow \min \quad (5)$$

in which:  $w_i$  is the weight of point number  $i$ .

Other steps have no change and should be followed as mentioned before.

Figure 5. compares two lines which fit three points. One of the lines considers the second point as having a weight of 5 times more than the others, and the other line considers same weight for all points.

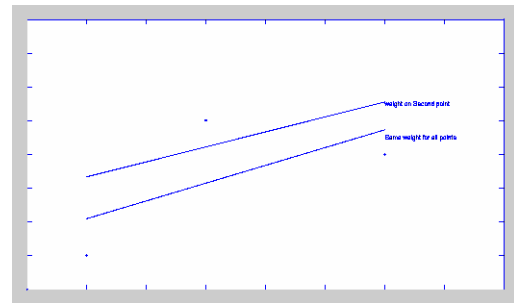


Figure 5. Comparing best fitting lines, for points with weight and points without weight

### 4. ADVANTAGES OF THE NEW METHOD

The introduced new method has some advantages over other methods. It looks to the polyline as a whole object and follows the trend of polyline for simplifying it. This method also acts upon the trend of consecutive points. So, all of the curvatures and offsets in the main polyline will influence in the final shape.

This method is also faster than other methods, especially when the number of points increases or the trend of the main polyline does not change suddenly.

The ability of applying weights to the vertex points of the polyline is another advantage of this new method. This ability gives the opportunity to consider points as less important or more important than the other ones. This advantage can be used in other algorithms in which points may differ from each other in importance.

Also, the result of simplification in this method is unique. It means that only one polyline is produced after simplification. Contrary to some other methods that the result may change if the simplification starts from other end of the polyline, or in polygon simplification that the beginning or the end of the polygon is not known.

## **5. CONCLUSION**

In this paper a new method for simplification is introduced. There is no doubt that each method may have superiors or inferiors to others. The main attempt in this paper was to look on simplification as a fuzzy process not as an absolute process, and letting each vertex have different influence in the final result.

The applications for this method may be developed in the future. The method may go under many changes. It may also be revised to suit some applications or software implementations. It is suggested that researchers test the new method. Going toward this method will explicit its advantages and disadvantages better in different applications.

## **6. REFERENCES**

Dr. George Taylor, "Line Simplification Algorithms"

## **7. ACKNOWLEDGEMENTS**

My best appreciation is dedicated to my colleagues in Zanjan University. Their help and support encouraged me to go toward this research.