Line Simplification Using Self-Organizing Maps

Bin Jiang
Division of Geomatics, Dept. of Technology and Built Environment, University of Gävle, Sweden.

Byron Nakos
School of Rural and Surveying Engineering, National Technical University of Athens, Greece.

ABSTRACT: We develop an approach to line simplification based on shape detection using self-organizing maps (SOM), an artificial neural network algorithm for data clustering and visualization. Based on the coordinates of individual vertices of a line, our approach derives a set of neurons to represent the overall shape of the original line. Through the shape detection, those vertices with maximum similarity to individual neurons are selected for line simplification. We applied the approach to a coastline of an island and compared our simplification results to those using existing algorithms. Our study shows that SOM is robust in detecting the shape of the original lines for simplification purpose.

1 INTRODUCTION

Line simplification is a filtering process that selects those critical points for retaining the overall shape of an original line while eliminates those trivial points. Over the past decades, many algorithms have been developed for line simplification purposes. Among others, Douglas-Peucker (D-P) algorithm (Douglas and Peucker 1973) is one of the most effective algorithms in retaining the shape of the original line, although there are critics from different aspects (e.g. Visvalingham and Whyatt 1990). Dutton (1999, p.36) claims that D-P algorithm is "*really only one global algorithm in use*", since it is able to maintain the overall shape of the original line. Li and Openshaw's algorithm (1992) is another one of such algorithms, and it uses so-called smallest visual object to detect and retain the overall shape of a line based on the so-called natural principle. Wang and Muller (1998) have argued that the most algorithms are geometric solutions rather than cartographic ones, as these algorithms involve only filtering process. Therefore they developed an algorithm based on the detection of individual bends of the original line, and using mathematically designed measures in the course of line generalization. Herein the term of line generalization, different from that of line simplification, involves not only filtering processes but also other processes like combination and exaggeration. However what is common for the two terms is the detection of overall shape of a line, in order to achieve satisfied outcomes.

A sound algorithm for line simplification must be based on sound shape detection, through which a satisfied simplification can be reached. This idea has been widely adopted by cartographic researchers. For instance, shape detection or structure recognition in their own term is the fist step in their automated generalization framework developed by Brassel and Weibel (1988). This paper develops an approach to line simplification based on shape detection using self-organizing map (SOM). SOM is an artificial neural network algorithm (Kohonen, 2001) that is used as a method for shape detection of the original lines based on an unsupervised training process. Through it, we are able to derive a representative line that represents the overall shape of the original line. This representative line can be further used for vertices filtering process in order to select those critical vertices. SOM has been used in many fields such as data classification, pattern recognition, image analysis, and exploratory data analysis (for an overview, see Oja and Kaski 1999). In the domain of GIS and cartography, Openshaw and his colleagues have used SOM approach in spatial data analysis to carry out the classification of census data (Openshaw 1994, Openshaw et al. 1995). SOM has been used for selection of streets from a network – so a kind of model generalization (Jiang and Harrie 2003). In that study, multiple attributes of streets from semantic, geometric and topological perspective are considered for measuring similarity among the streets in order to group them into difference categories. Eventually two types of streets are detectable, i.e. those to be eliminated and those to be selected, for selection purpose. However this paper focuses on graphic generalization, assuming that a prior model generalization is done.

The remainder of this paper is structured as follows. Section 2 presents the basic principle and algorithm of SOM. Section 3 describes how SOM is used for shape detection and line simplification with an illustrative example. Section 4 reports a case study applied to a coastline and a comparison study to other line simplification algorithms. Finally section 5 concludes the paper and points out future work.

2 SELF-ORGANIZING MAP

SOM is a well-developed neural network technique for data clustering and visualization. It can be used for projecting a large data set of a high dimension into a low dimension (usually one or two dimensions) while retaining the initial pattern of data samples. That is, data samples that are close to each other in the input space are also close to each other on the low dimensional space. In this sense, SOM resembles a geographic map concerning the distribution of phenomena, in particular referring to first law of geography: everything is related to everything else, but near things are more related to each other (Tobler 1970). Herewith we provide a brief intuitive introduction to the SOM; readers are encouraged to refer to more complete descriptions in literature (e.g. Kohonen 2001).

2.1 *Basic principle*

The SOM training algorithm involves essentially two processes, namely vector quantization and vector projection (Vesanto 1999). Vector quantization is to create a representative set of vectors, so called output vectors from the input vectors. In general, vector quantization reduces the number of vectors. This can be considered as a classification, or clustering, process. The other process, vector projection, aims at projecting output vectors (in d-dimensional space) onto a regular tessellation in lower dimensions (i.e., a SOM), where the regular tessellation consists of an arbitrary number of neurons. In the vector projection each output vector is projected into a neuron where the projection is performed as such, “close” output vectors in d-dimensional space will be projected onto neighbouring neurons in the SOM. This will ensure that the initial pattern of the input data will be present in the neurons.

The two tasks are illustrated in figure 1, where usually the number of input vectors is greater than that of output vectors, i.e. $n \succ k$, and the size of SOM is the same as that of output vectors. It should be emphasized that for an intuitive explanation of the algorithm, we separate it as two tasks, which are actually combined together in SOM without being sense of one after another.

2.2 *The algorithm*

The above two steps, vector quantization and vector projection, constitute the basis of the SOM algorithm. Vector quantization is performed as follows. First the output vectors are initialized randomly or linearly by some values for its variables. Then in the following training step, one sample vector x from the input vectors is randomly chosen and the distance between it and all the output vectors is calculated. The output vector that is closest to the input vector x is called the Best-Matching Unit (BMU), denoted by m_c:

$$\| x - m_c \| = \min_i \{ \| x - m_i \| \} \tag{1}$$

where $\|.\|$ is the distance measure. Second the BMU or winning neuron and other output vectors in its neighbourhood are updated to be closer to x in the input vector space. The update rule for the output vector i is:

$$\begin{aligned} m_i(t+1) &= m_i(t) + a(t)h_{ci}(t)[x(t) - m_i(t)] \qquad && \text{for } i \in N_c(t) \\ m_i(t+1) &= m_i(t) && \text{for } i \notin N_c(t) \end{aligned} \tag{2}$$

where $x(t)$ is a sample vector randomly taken from input vectors, mi(t) is the output vector for any neuron i within the neighbourhood Nc(t), and $a(t)$ and $h_{ci}(t)$ are the learning rate function and neighbourhood kernel function respectively.

Through the training process, all output vectors are projected on to a 1- or 2-dimensional space, where each neuron corresponds to an output vector that is the representative of some input vectors. A 2-dimensional hexagonal map lattice grid is shown in Figure 2 where each hexagonal cell has a uniform neighbourhood.

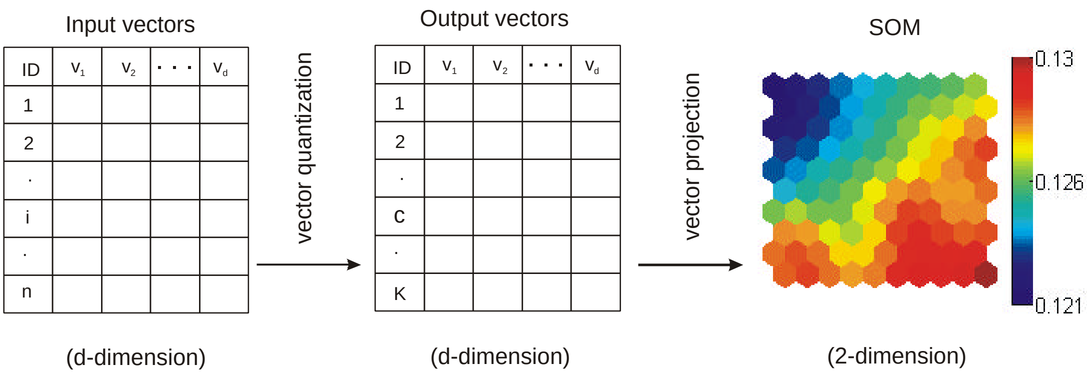

Figure 1: Illustration of SOM principle

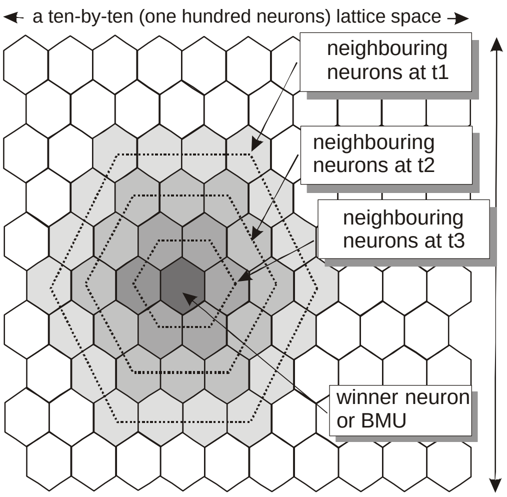

Figure 2: The characteristics of a 10x10 SOM (t1<t2<t3 with $h_{ci}(t)$ in equation 3)

3 LINE SIMPLIFICATION USING SOM

To illustrate how SOM can be used for line simplification, we adopt a line from McMaster (1989) that involves 40 vertices (Figure 3). We consider the coordinates of individual vertices for training process. Thus the input vectors consist of x y coordinate of individual vertices (Table 1), noting that two ending vertices are excluded from the input vectors, as they will be kept in any simplification. Assume that we intend to retain 14 vertices, so we decide 14 output vectors or neurons, which are initialized randomly and are imposed on the top of input vectors. After a training process, s SOL that consists of 14 neurons (Table 2) is created to represent the original shape of line. We could remark that four bends are represented with the SOL, i.e. the black line in Figure 4.

For simplification purpose, we further compute the distances between individual vertices and their best matching neuron; only those vertices with minimum distances are retained in the end. Figure 4 illustrates the simplification procedure. In the figure, blue line is original one with 40 vertices; red line is simplified line with 14 vertices; black line is the representative line with 14 neurons that are labelled with numbers; green lines show the distances between individual vertices and their best matching neurons. From the figure, one can note that those vertices with minimum distances (or maximum similarity) are indeed selected; the same vertices have also been highlighted in Table 1. Note that Table 1, Table 2 and the black line in Figure 4 correspond to input vectors, output vectors and SOM respectively in Figure 1.

This simple and straightforward example illustrates the basic procedure for line simplification. The result of the line simplification is interesting, for example, the simplified line does represent the overall shape of the original line, and the 5 bends are retained in the end.

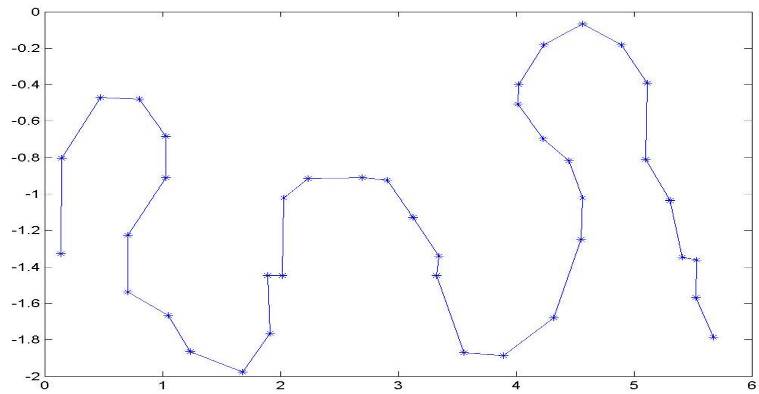

Figure 3: McMaster line with 40 vertices

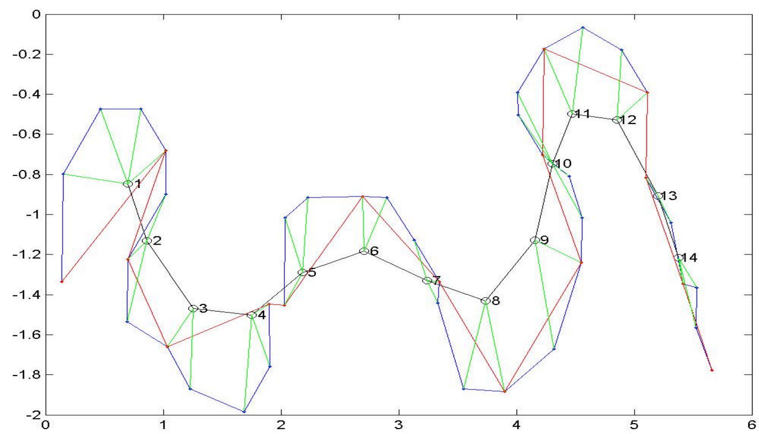

Figure 4: Illustration of line simplification using SOM

Table 1: Input vectors (or 40 vertices of original line)

ID	X	Y	ID	X	Y	ID	X	Y	ID	X	Y
1	0.152	-0.798	11	1.909	-1.760	21	3.549	-1.872	31	4.563	-0.067
2	0.470	-0.474	12	1.898	-1.447	22	3.897	-1.883	32	4.894	-0.179
3	0.807	-0.474	13	2.027	-1.453	23	4.322	-1.671	33	5.112	-0.391
4	1.025	-0.680	14	2.033	-1.016	24	4.552	-1.241	34	5.100	-0.816
5	1.025	-0.898	15	2.228	-0.916	25	4.557	-1.016	35	5.312	-1.040
6	0.700	-1.223	16	2.694	-0.910	26	4.445	-0.810	36	5.419	-1.347
7	0.694	-1.535	17	2.900	-0.916	27	4.221	-0.704	37	5.531	-1.364
8	1.037	-1.659	18	3.130	-1.129	28	4.015	-0.503	38	5.525	-1.565
9	1.231	-1.872	19	3.343	-1.335	29	4.009	-0.391			
10	1.685	-1.984	20	3.331	-1.441	30	4.233	-0.173			

Table 2: Output vectors (or 14 neurons of the representative line)

ID	X	Y	ID	X	Y
1	0.701	-0.846	8	3.740	-1.431
2	0.861	-1.132	9	4.158	-1.127
3	1.258	-1.468	10	4.307	-0.746
4	1.754	-1.503	11	4.476	-0.497
5	2.181	-1.290	12	4.856	-0.527
6	2.709	-1.184	13	5.206	-0.908
7	3.240	-1.330	14	5.379	-1.216

4 CASE STUDY AND COMPARISON RESULTS

In order to further demonstrate the validation of the approach, we carry out a case study applied to the coastline of Peristera Island, a coastline characterized by a high degree of complexity. The coastline was digitized from a paper map of scale 1:50K with an average step of 15 meters (figure 5). The raw data were cleaned up from duplicate vertices, spikes, or switchbacks and from redundant co-linear vertices, after a 'weeding' process as it has been suggested by Jenks (1981) in order to produce the original line. The original line involves totally 2415 vertices. We conducted a scale-driven simplification based on six tasks with respect to the following seven map scales: 1:100K, 1:250K, 1:500K, 1:1M, 1:2M, and 1:5M. The number of vertices in each map scale is decided by "Principles of Selection" (Töpfer and Pillewizer 1966), which is expressed as an equation,

$$n_d = n_s \sqrt{(M_s / M_d)^x}$$

where n_d is the number of objects at a derived map, n_f is the number of object in original source map, M_s is the scale denominator of the source map, and M_d is the scale denominator of the derived map, and the exponent x is a variable for different spatial objects, usually for linear objects $x=2$ (Töpfer and Pillewizer 1966, p. 12). Thus by applying the "Principles of Selection" the number of retained vertices is linearly depended by the ratio of scale denominators of source over derived map. The desired reduction of vertices was achieved by selecting appropriate tolerances with trial and error method. In Table 3 the nominal map scales, the reduction percentages and the number of retained vertices for the three sets of derived lines for all tasks is given.

Table 3: Tasks of line simplification

Task	Nominal map scale	Reduction percentage	Number of retained vertices D-P	BEND	SHAPE
1	1:100K	50%	1208	1208	1208
2	1:250K	20%	483	484	483
3	1:500K	10%	243	247	242
4	1:1M	5%	122	130	121
5	1:2M	2.5%	61	62	61
6	1:5M	1%	24	22	24

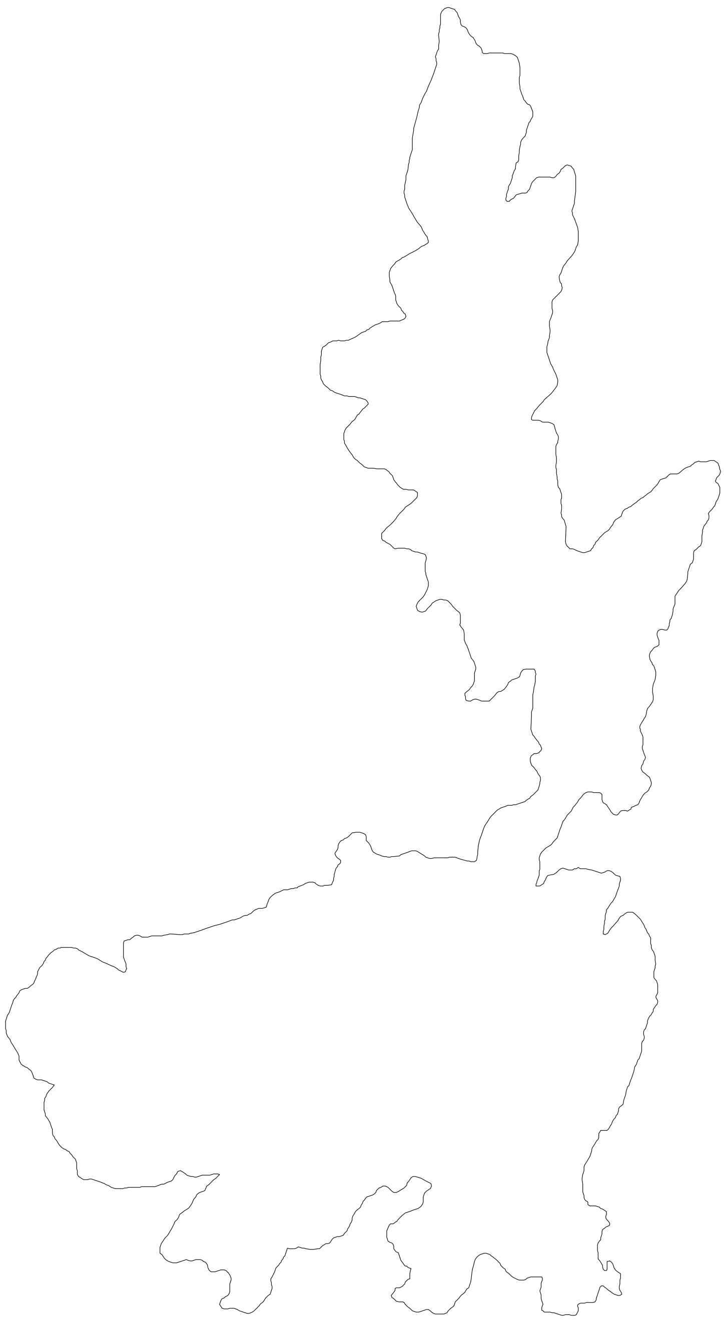

Figure 5: The original line (Peristera Island coastline digitized from a map of scale 1:50K)

We also conducted a comparison of our approach of line simplification (SHAPE) with two well-known line simplification algorithms, the D-P algorithm (Douglas and Peucker 1973) and bendsimplify (BEND) algorithm (Wang and Müller 1998). As general comment, our approach of line simplification when compared visually with the two other algorithms is found that it keeps both the overall shape and a certain level of details of the line in a balanced way. Assuming that, BEND algorithm is a line simplification algorithm that preserves the shape of the line with a cartographically satisfactory way by retaining certain bends of the line (Wang and Müller 1998), and that D-P algorithm is a line simplification algorithm that produces a result sensitive to high frequencies -see for example the criticism stated by Visvalingam and Whyatt (1990)- our approach of line simplification runs in the center line between them.

Figure 6: The simplified lines after applying the three algorithms for all tasks

We claim that those who want to simplify lines in a way of both preserving the overall shape of the line and a certain degree of details they may find our approach more appropriate.

5 DISCUSSION AND CONCLUSION

This paper explored a new approach to line simplification based on shape detection using one-dimensional SOM or self-organizing line in our term. Our study has shown that it is an effective approach to line simplification based on a comparison with two other algorithms. First the approach considers both global and local context in the training process, thus the outcome is comparable to that of D-P algorithm. What appears to be advantage for the approach is that it is flexible to decide radius changes from global to local. Second the approach can carry out scale-driven simplification, i.e. the level of simplification is not specified by a tolerance but by the number of retained vertices. Despite these advantages, it is not without problems for the approach. It is still considered to be a filtering approach rather than cartographic approach to line simplification, in particular when compared to BEND algorithm. For instance, for the task 5 with the case study, the "neck" part of the island is less than threshold that human beings can perceive.

REFERENCES

Brassel K., Weibel R. (1988). A review and conceptual framework of automated map generalization. International Journal Geographical Information Systems, Vol. 2, no. 3 pp. 229-244

Douglas, D.H. and Peucker, T.K. (1973). Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature, *The Canadian Cartographer*, 10(2): 112-122.

Dutton, G. (1999). Scale, Sinuosity and Point Selection in Digital Line Generalization, *Cartography and Geographic Information Science*, 26(1):33-53.

Jiang, B. and Harrie, L. (2003). Cartographic Selection Using Self-Organizing Maps, ICA Generalisation Workshop Working paper, available at
http://www.geo.unizh.ch/ICA/docs/paris2003/papers03.html

Kohonen, T. (2001). *Self-Organizing Maps* (third edition), Springer, Berlin, Heidelberg, New York.

Li, Z. and Openshaw, S. (1992). Algorithm for Automated Line Generalization based on a Natural Principle of Objective Generalization, *International Journal of Geographic Information Systems*, Vol. 6, No. 5, pp. 373 – 389.

McMaster, R.B. (1989). The Integration of Simplification and Smoothing Algorithms in Line Generalization. *Cartographica*, 26(1): 101-121.

Oja, E. and Kaski, S. (editors. 1999). *Kohonen Maps*, Elsevier.

Openshaw S., Blake M. and Wymer C. (1995), Using Neurocomputing Methods to Classify Britain's Residential Areas, (html paper), available at http://www.geog.leeds.ac.uk/papers/95-1/

Openshaw, S. (1994), Neuroclassification of spatial data, in Hewitson, B. C. and Crane, R. G. (eds), *Neural Nets: Applications in Geography*, Dordrecht: Kluwer Academic Publishers, 53-70.

Tobler, W.R. (1970). A Computer Movie Simulating Urban Growth in Detroit Region, *Economic Geography*, 46, pp. 234-240.

Töpfer, F. and Pillewizer, W. (1966). The Principles of Selection, *The Cartographic Journal*, 3(1):10-16.

Vesanto, J. (1999). SOM-Based Data Visualization Methods, *Intelligent Data Analysis*, Elsevier Science, 3(2): 111-126.

Vesanto J., Himberg J., Alhoniemi E. and Parhankangas J. (2000), *SOM Toolbox for Matlab 5*, Report A57, Helsinki University of Technology, Libella Oy, Espoo.

Visvaligam, M., and Whyatt, J.D. (1990). The Douglas-Peucker algorithm for line simplification: Re-evaluation through visualization, *Computer Graphics Forum*, 9(3):213-228.

Wang, Z., and Müller, J.C. (1998). Line Generalization Based on Analysis of Shape Characteristics, *Cartography and Geographical Information Systems*, 25(1): 3-15.