UDC 517.38

An algorithm for cartographic generalization that preserves global topology

V.V. Alexeev, V.G. Bogaevskaya, M.M. Preobrazhenskaya, A.Yu. Ukhalov, H. Edelsbrunner, O.P. Yakimova [1]

We propose an algorithm for the generalization of cartographic objects that can be used to represent maps on different scales.

1 Introduction

Process of generalization is an important step in map processing. *Generalization* is an adaptation of geographic objects on the map for presentation in accordance with the purpose of the map, scale and features of the territory (see [1]). For a long time generalization was considered to be a subjective process requiring participation of a qualified expert. With development of the computers and automation it became necessary to formalize this process and develop computer algorithmes for performing this job. It is specially important when information have to be used in automatic navigation systems and maps which are available via Internet. It is impossible to store on the server the data for representing maps in all scales which can be demanded by user. It is natural to store only the most detailed representation of the map and scale map on the fly. The chief difficulty in the automatic scaling is the superfluity of information.

For example, the image of area of 1 km^2 in scale 1:1000 occupies 1 m^2 on the map, in scale 1:100 000 – 1 cm^2, in scale 1:1 000 000 — 1 mm^2. It is impossible to represent the area in all these scales with the same amount of details. While decreasing the scale we have to reduce number of objects and eliminate some details. It is required not only because of shortage of space. On the map of the small scale which present a large area, details looses their meaning. It is difficult to understand the map if all the details are preserved. Moreover, transmitting of these unnecessary data via information network consumes valuable resources.

[1] The work was supported by a grant 11.G34.31.0053 from the government of the Russian Federation.

While performing generalization it is important to preserve the relative positions of objects. Simple averaging of the cartographic data may cause serious issues. For example if, diminishing the image in 10 times, we preserve one of 100 pixels out of the source data, it may happen that the road passing along the river will intersect the river though there is no any bridge in this place.

In this work we propose a new algorithm for automated cartographic generalization that preserve topology of the map.

The source data for our algorithm is a set of graphs embedded in the plane, each referred to as a *layer*. Each layer represents a particular cartographic object such as country borders, rivers, roads, etc; see Figure 1 for an example.

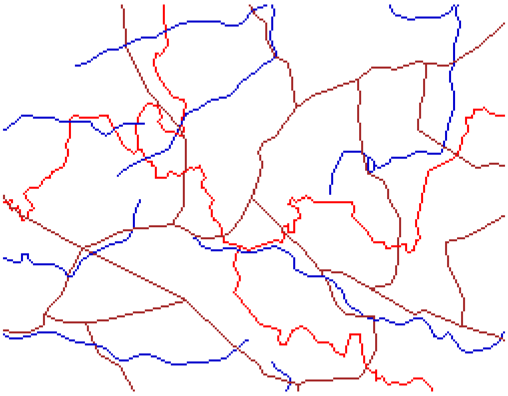

Рис. 1: Sample imput data. Different layers are drawn in different line styles.

Our aim is to simplify the data by decreasing the number of nodes of the graphs while preserving the global topology of the picture, which includes the relative position of objects to each other. The result of such a simplification can be used for representing the original map to given scale.

To generalize the data without changing its topology, we superimpose all layers into one data structure and simplify using the method of edge contraction. Edges are contracted in a sequence that gives priority to small geometric error, and the new vertices are placed to locally minimize the error. To preserve the topology, we check the local neighborhood of the edge and we prohibit the contraction if it would change the topology type of the structure.

2 Principal ideas of the algoriphm

The fundamental operation for simplifying the map is the *contraction* of an edge, AB. It shrinks the edge to a point, X, which replaces both A and B in all edges incident to either endpoint. The position of the new vertex, X, is chosen according to the method described in [2].

For each vertex of the graph we calculate the error matrix as follows. Let N be a number of edges which are incident to the vertex A. Each edge determines the line on the plane. Let $a_i x + b_i y + c_i = 0$, where $a_i^2 + b_i^2 = 1$ $(i = 1, \ldots, N)$ – the equations of these lines. Let's find the sum of the distances from each line to the point (x, y)

$$F(x, y) = \sum_{i=1}^{N} (a_i x + b_i y + c_i)^2 = q_{11}x^2 + 2q_{12}xy + q_{22}y^2 + 2q_1 x + 2q_2 y + q_0.$$

This quadratic form is called *error* of the vertex A. Matrix of this quadratic form

$$Q_A = \begin{pmatrix} q_{11} & q_{12} & q_1 \\ q_{12} & q_{22} & q_2 \\ q_1 & q_2 & q_0 \end{pmatrix}$$

is called an *error matrix* of the vertex A. We will associate the edge AB with the matrix $Q_{AB} = Q_A + Q_B$, where Q_A, Q_B – error matrices for vertices A и B correspondingly.

When we contract the edge AB the position of the new vertex $X = (x, y, 1)^T$ will be determined as a solution of the following minimization problem

$$XQ_{AB}X^T \to \min .$$

This minimal value we will call *cost* of contraction of the edge AB. Problem of minimizing of the quadratic form is equivalent to solving the system of two linear equations. If solution of this system is not determined, the new vertex will coincide with one of vertices A or B – the one which gives the minimal value of the quadratic form $XQ_{AB}X^T$. The cost of contraction in this case will be equal the corresponding value of this quadratic form.

The contraction of an edge can change the topology of the map, in which case we prevent the contraction. More precisely, we restrict ourselves to *topology preserving* edge contractions that translate into isotopies of the map. In other words, there is a homotopy of the plane to itself that starts with the situation before and ends with the situation after the contraction and whose restriction to any moment of time is a homeomorphism. General conditions that recognize topology preserving edge contractions are given in [3, 4, 5]. For the case of an embedded graph, these conditions are straightforward: the contraction of an edge AB preserves the topology if and only if (a) at least one of the two endpoints belongs to exactly two edges (see Figure 2) and (b) vertices A and B have no common neighbours.

3 The Algoriphm

We describe our algorithm as a sequence of five steps. Most important is Step 3, which constructs the simplification by repeated edge contraction.

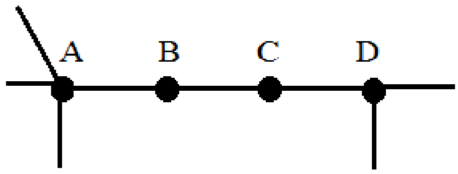

Рис. 2: Condition on edge contraction. All three edges, AB, BC, CD can be contracted, but after contracting two, the remaining edge has endpoints of degree 4 on the left and 3 on the right. Its contraction is does therefore no longer preserve the topology.

1. **Superimposing.** We superimpose all the layers into one data structure. This allows us to simplify all layers simultaneously, thus preserving its global topology and not just the topology of each individual layer. We remember for each edge the original layer it belongs to.

2. **Sorting.** For each edge, we calculate the *cost* as the geometric error that results from contracting this edge; see [2]. We store all edges in a priority queue ordered by cost so that edges can be contracted in the order of increasing cost.

3. **Simplifying.** We simplify the structure by contracting edges in the order of increasing cost. We stop when the number of vertices drops below a specified percentage of original number of vertices.

4. **Smoothing.** We finally smooth the resulting polylines with B-splines for improved appearance.

5. **Decomposing.** The simplified structure can now be decomposed into the individual layers, each a simplified version of the corresponding original layer.

We illustrate the algorithm in Figure 3, which shows an original layer and its simplified version. Note that both vertices of degree 3 are preserved.

4 Computational experiments

We have implemented the algorithm and use the software to compare its performance with our own modification of the *Li and Openshaw Algorithm*. The original version of that algorithm described in [6] works only for polygonal lines, which are characterized by having only vertices of degree 2 and of degree 1. We have adapted it so simplify graphs in which vertices of degree 3 or higher are viewed as endpoints of polygonal lines.

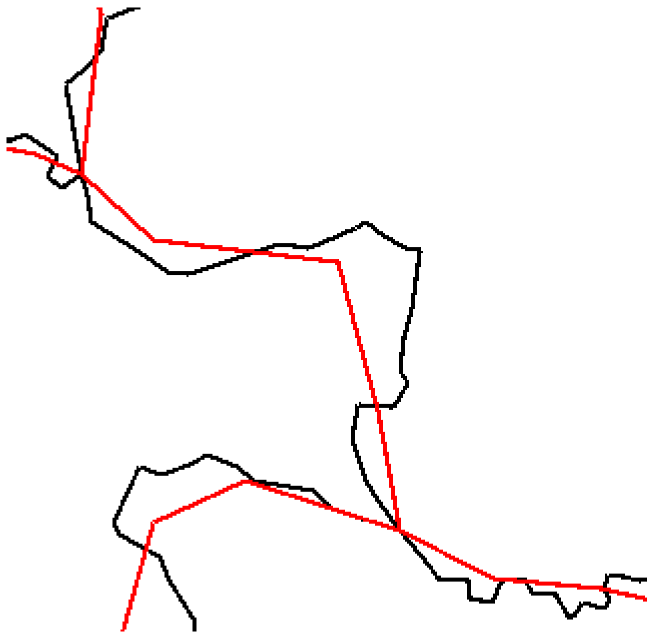

Рис. 3: A small segment of the data simplified by our algorithm.

To measure the error of approximation, we introduce a notion of distance between two embedded graphs. Letting G_i be a graph, we write V_i and n_i for the set and the number of vertices, and we write E_i for the set of edges. Letting $d(X, e)$ be Euclidean distance of the point X from the edge e, we write $d(X, E_i) = \min_{e \in E_i} d(X, e)$. Then we compute

$$D_{1,2} \quad = \quad \max\left\{ \frac{\sum_{Y \in V_2} d(Y, E_1)}{n_2}, \frac{\sum_{X \in V_1} d(X, E_2)}{n_1} \right\}. \tag{1}$$

This notion incorporates elements of the Hausdorff distance and of the Wasserstein distance between sets in the plane.

We compare the performance of our Edge Contraction Algorithm with that of the Whirlpool Algorithm on a set of nine datasets. The results of the comparison are shown in Table 1 and Figure 4.

Our experiments give evidence to the claim that the algorithm proposed in this paper performs better than the Li – Openshaw Algorithm.

Алгоритм Ли и Опеншоу.

The Li – Openshaw Algorithm (the Raster-vector algorithm) [7] woks as follows.

1. The line is being superimposed with the regular grid. The size of the grid step is equal to the minimal visible element of the map in the new scale.

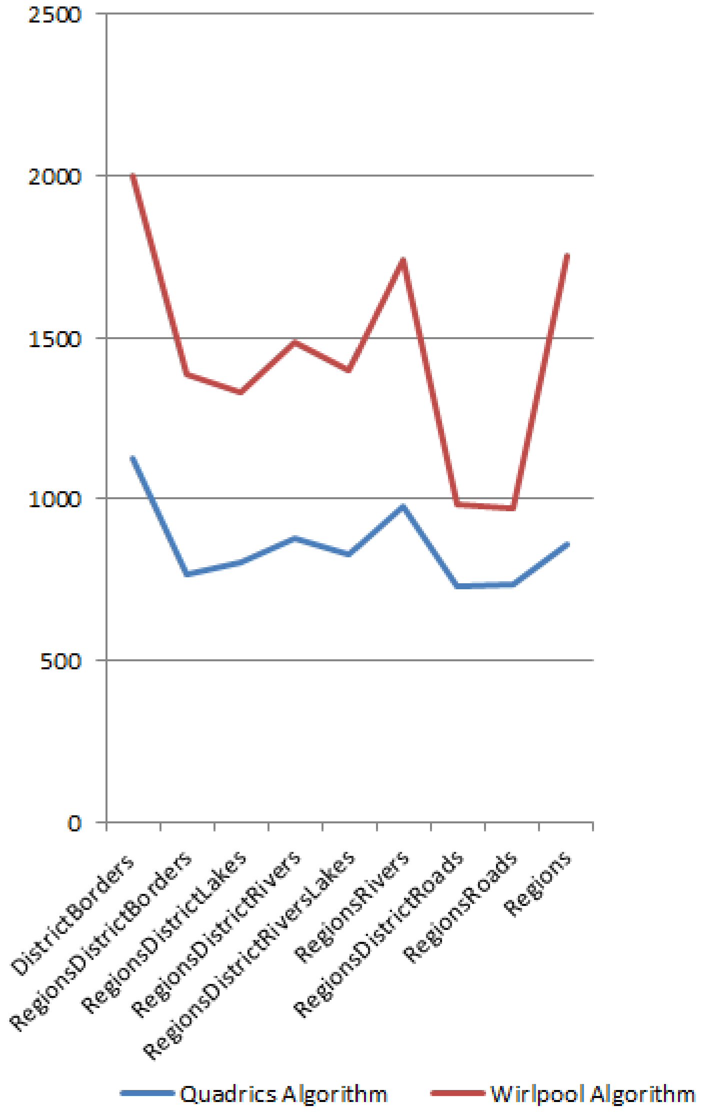

Рис. 4: Graphs illustrating the numerical results in Table 1 which compare the Edge Contraction Algorithm with our modification of the Whirlpool Algorithm.

Таблица 1: Сравнение алгоритмов стягивания ребер и вихревого алгоритма. В таблице приводятся рассояния между исходным и упрощенным графом после того, как число вершин снизилось до некоторой заданной величины.

Object	Contraction	Whirlpool
DistrictBorders	1126	2001
RegionsDistrictBorders	766	1389
RegionsDistrictLakes	804	1332
RegionsDistrictRivers	880	1485
RegionsDistrictRiversLakes	829	1401
RegionsRivers	980	1739
RegionsDistrictRoads	732	986
RegionsRoads	734	971
Regions	859	1751

2. Points of the first and last intersection of the line with the cell border are being marked.

3. The middle of the line connecting these two poits is being used for representing the line in this cell.

5 Acknowledgements

We would like to offer our special thanks to students of the Department of Mathematics of Demidov Yaroslavl State University A.A. Gorohov and V.N. Knyazev for participation in developing the program and assistance in preparation of test data.

Список литературы

[1] A. M. Berlyant, O. R. Musin and T. V. Sobchuk. Cartographic generalization and fractal theory. Moscow, 1998.

[2] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. *SIGGRAPH '97, Proc. 24th Ann. Conf. Comput. Graphics, 209–216.*

[3] T. K. Dey, H. Edelsbrunner, S. Guha and D. V. Nekhayev. Topology preserving edge contraction. *Publ. Inst. Math. (Beograd) (N.S.) 66 (1999), 23–45.*

[4] F. Vivodtzev, G.-P. Bonneau and P. Le Texier. Topology preserving simplification of 2D non-manifold meshes with embedded structures. *The Visual Computer 21 (2005), 679–688.*

[5] D. M. Thomas, V. Natarajan and G.-P. Bonneau. Link conditions for simplifying meshes with embedded structures. *IEEE Trans. Vis. Comput. Graphics 17 (2011), 1007–1019.*

[6] E. Fritsch. Use of Whirlpool algorithm for ADBS data generalization. *ADBS meeting, 1999.*

[7] Li, Z., Openshaw, S. Algorithms for automated line generalization based on a natural principle of objective generalization.International Journal of Geographical Information Systems, 6(5) (1992), 373-389.