Geometric Simplification of Administrative Borders With Mixture of Irregular and Orthogonal Segments

Timofey Samsonov
Lomonosov Moscow State University
Faculty of Geography
Leninskiye Gory 1
Moscow, Russia
tsamsonov@geogr.msu.ru

Olga Yakimova
Demidov Yaroslavl State University
Faculty of Mathematics
Soyuznaya str. 144
Yaroslavl, Russia
polya@uniyar.ac.ru

Abstract

Line generalization is one of the essential data processing operations in GIS and cartography. Many point reduction, line simplification and generalization algorithms have been developed for this purpose so far. Several specialized algorithms can be found that allow simplification of buildings — mainly to preserve their rectangular shape and reproduce it in general form. In this article we present a more common methodology for simplification of lines consisting of both natural (irregular) and artificial (orthogonal) segments. The core of the presented methodology is an algorithm for detection of right angle sequences. After the line is subdivided into irregular and orthogonal parts, their simplification is made separately by different approaches. The methodology is assesed on the example from Russian admninistrative units.

Keywords: line generalization; geometric simplification; orthogonality; algorithms

1 Introduction

Interest to line generalization lasts for nearly half a century. Li (2007) differentiates between point reduction, smoothing and scale-driven generalization of lines. All generalization algorithms can be formally described in terms of objectives, operations and workflows.

The objectives for line generalization are usually formulated in terms of conditions and constraints. Conditions, at their point, can be classified into *global* and *local* ones.

Examples of global conditions are: Hausdorff distance (Haunert and Wolff, 2010), subsequence of vertices (McMaster, 1987), the desired number of edges/vertices (Buchin et al., 2011) or spatial resolution (Li, Openshaw, 1992). Local conditions can be formulated in terms of: perpendicular distance (Douglas and Peucker, 1973), effective area (Visvalingam, Whyatt, 1993), turning angle or curvature (Rosenfeld and Johnston, 1973), edge length or ratio (Teh and Chin, 1989), bend area (Wang and Müller, 1998) and so on.

Sometimes additional constraints are applied such as area preservation (Buchin et al., 2011) and shape preservation (Wang and Müller, 1998). Forcing all edges to follow the set of given directions, is the core constraint in line schematization wokflows (Moulemans et al., 2010). One of the most important generalization constraints is the preservation of topology (de Berg et al., 1998). All conditions and constraints must be accompanied by functions that evaluate them while performing generalization operations.

The second component of line generalization algorithms are operations that are applied to the line. These in turn can be classified into preprocessing and processing operations. Preprocessing operations are usually dealt with enrichment of our knowledge about lines. For example, Wang and Müller (1998) algorithm is based on line segmentation into bends which are the parts of line with constant sign of the turning angle. Buchin et al (2011) use arc and orientation detection and then wall squaring prior to polygon simplification to generalize buildings correctly. These are the typical examples of preprocessing.

Processing operations comprise the atomic core of generalization algorithm and can be a kind of: point removal (Douglas and Peucker, 1973), replacement of a line segment by a representative point in a grid cell (Li, Openshaw, 1992), edge contraction (Dey et al., 1999), edge move (Buchin et al., 2011), creating a short-cut (Haunert and Wolff, 2010) and so on.

Finally, the evaluation of the conditions and performing operations must arranged into workflows, which are usually sequential or iterative (Li, 2007), but can also be implemented by means of optimization (Sester, 2005;

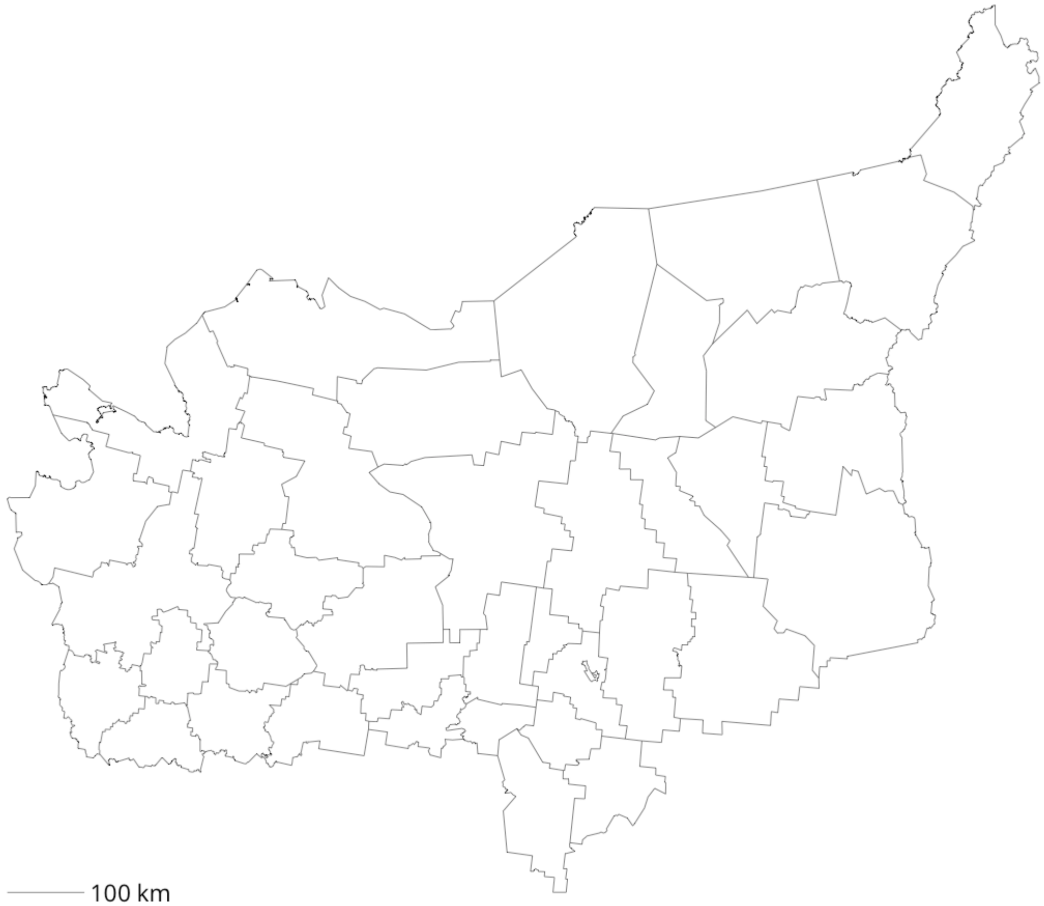

Figure 1: Example of administrative borders with mixture of irregular and orthogonal segments (Akhangelsk and Komi regions in Northern European Russia).

Haunert and Wolff, 2010).

Any line generalization algorithm is a mixture of the elements described above. While many algorithms are developed for the common case of the line or polygon, some of them have the specific combination of elements developed for the particular kind of the source data. These include buildings (Damen et al., 2008; Bayer, 2010; Buchin et al., 2011) which tend to have artificially sharp or curved angles, coastlines and contours (Wang and Muller, 1993; Ai, 2010) that commonly have hierarchical structure of bends.

These propositions are global (appliable for the whole line) for the mentioned kinds of objects. Thus, wall squaring can be applied to all objects in buildings layer without the lost of meaning. At the same time, there are line datasets that contain both regular (squared and arc-like) and irregular (with no shape pattern) segments. One common example is administrative borders of states and counties in USA that may follow parallel and meridian directions in one part (being straight or squared) and follow rivers and mountain ridges in another (being natural-looked). Another example from Russian administrative borders of such kind are presented in Figure 1. In this case, no one algorithm from those mentioned above would be effective globally.

In this paper we present the approach to generalization of such kind of lines. It performs line segmentation into squared and non-squared parts and applies the appropriate generalization algorithms to achieve the geographically meant result. Our preprocessing and processing workflows are expounded in the methodological part of the paper. Finally experimental results on Russian administrative units are presented and discussed.

2 Methodology

We developed an approach to generalization of lines consisting of both irregular and orthogonal segment patterns. The approach consists generally of two steps: preprocessing that derives right angle sequences and actually processing that is performed separately for orthogonal and irregular line parts.

The intuitive term *right angle* should be formalized to be applied in a generalization workflow, because it includes not only the value of angle, but also some characteristics of line edges that comprise the angle. To achieve this we put forward the following requirements:

- The angle should be composed by two line edges which are long enough to be perceived. I.e. angle should be visible to the map user. Thus, we need to set the length tolerance S. Each side of the angle must be longer or equal to S. The value of S depends on the resolution of the source data, thus it is convinient to set it in millimeters on screen. Then the number of long segments will depend on the scale of visualization that is set at the preprocessing stage,
- The angle should be approximately equal to $90°$. Some corners on the map are perceived as right or almost right, although in fact they are slightly more or less than $90°$. The strict equality is ineffective since borderlines are often do not follow the accurate right angle sequence. Digitizing errors and precision will also affect the angle values. Thus we also need some angle tolerance $\hat{\alpha}$ that allows us to consider the angle as right if it's value is in the $90° \pm \hat{\alpha}$ neighbourhood.

Because the density of line points may vary and also there can be a noise in point coordinates we need to remove extra points and thus clear the line for the effective search of the long edges. This operation can be performed using any point-reduction (Li, 2007) algorithm with small linear tolerance (about 0.5 on screen).

Considering these issues, the preprocessing stage consists of three steps:

1. Extraction of long straight edges
2. Detection of right angles and right angle sequences
3. Squaring of right angle sequences

The source data for preprocessing is the line string given by vertices – two dimensional points representing some linear geographic object (borderline of the state, river, contour of the continent etc.).

2.1 Preprocessing

2.1.1 Straight edge extraction

To extract long straight edges we applied a simple sequential point-reduction algorithm based on perpendicular distance d (Lang, 1969). The idea is to apply it with small linear tolerance that will almost have no impact on the shape of the line, but will remove extra points. The algorithm works as follows:

1. Let the first point of the polyline be a start point p_i.
2. Skip p_{i+1} and let p_{i+2} be an end point p_j.
3. Take the straight line l through p_i and p_j.
4. Compute the perpendicular distance d_k between the line l and each point p_k between p_i and p_j.

   If all d_k less than d then $j = j + 1$. Goto 3.

   Else delete all points between i and $j - 1$. Let $i = j - 1$ and goto 2 if we have not reached the end of the polyline.

Finally, all the intermediate points are deleted. Using the small value of d we are able to remove small and invisible details from the line. Then the edges which are longer than S are considered to be long straight edges.

2.1.2 Detection of right angles and right angle sequences

The angles are calculated between the two adjacent line edges which both have length more than S. Let p_i be the i-th vertex of the line and β_i be its angle value. We denote by e_i the edge between p_i and p_{i+1}, and l_i is the length of e_i.

Let $s_i = min(l_{i-1}, l_i)$ be the length of shortest angle side, $\alpha_i = |\beta_i - \frac{\pi}{2}|$ is deviation of the angle from right and $\hat{\alpha}$ is the maximum allowable value of α_i for the angle with $s_i = S$. We call $\hat{\alpha}$ *angle tolerance*. If $\alpha_i \leq \hat{\alpha}$ and $s_i = S$ then the angle is marked as right.

Additionally, $\hat{\alpha}$ must be corrected for the cases when $s_i > S$. This is up to two reasons. First, the human eye is capable of estimating angles more accurately when angle sides are longer. The next reason is that squaring the angles with long edges will lead to significant shifts of the edge endpoints, which will eventually lead to large distortion of the shape and probably some topological errors. Therefore, the longer is the shortest side of the angle, the smaller should the value of $\hat{\alpha}$.

We call this the *corrected angle tolerance* $\hat{\alpha}_i$ and calculate it using the the following equation:

$$\hat{\alpha}_i = \hat{\alpha}\left(1 - \frac{s_i - S}{ks_i}\right), \tag{1}$$

where $k \geq 1$ is inhibition parameter. The greater is k, the slower angle tolerance decreases in proportion to the increase of the shortest angle side. This transformation is illustrated in Figure 2 with $S = 2$ mm and $\hat{\alpha} = 10°$. When $k = 1.0$, then the proportion is inverse linear: if edge increases 5 times, then angle tolerance decreases 5 times (10°to 2°). But for $k = 2.0$ the angle tolerance decrease will be just 1.67 times (10°to 6°).

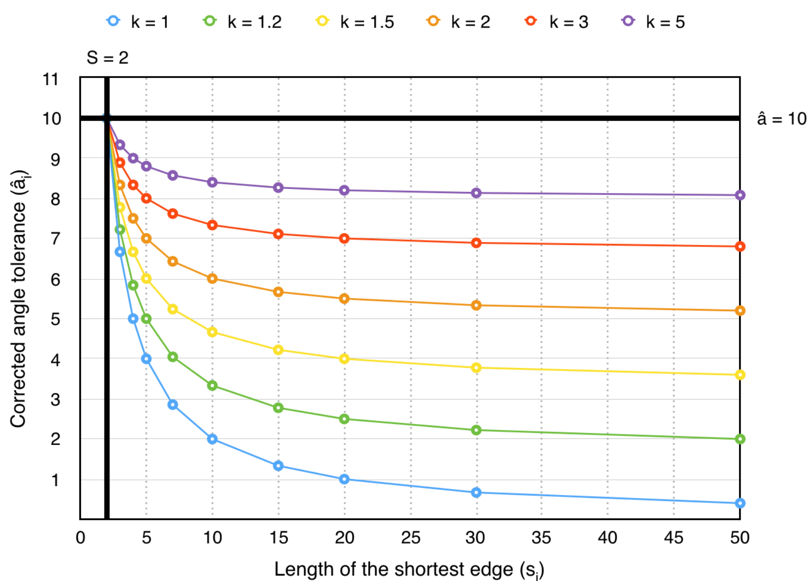

Figure 2: Relationships between corrected angle tolerance and the length of the shortest angle side

Inclusion of such a parameter allows us to consider the angles with relatively long sides for squaring and convey some experiments for selection of optimal value of k which will be described later.

2.1.3 Squaring of right angle sequences

It is a well established rule in cartographic generalization to make the relations stronger in smaller scales. Thus fuzzy relations become strict and almost perpendicular lines must be made prependicular. So, before further simplification we need to make the angles in right angle sequences be exactly right.

The squaring is performed in a following manner for each sequence:

1. Compute the total lengths of odd and even edges. The largest sum corresponds to the more extended direction or the main orientation of the sequence.

2. For each edge in main orientation compute the convergence angle γ_i between the edge and the Y axis.

3. Obtain the weighted average of γ_i as the descriptor of main orientation:

$$\gamma_{av} = \frac{\sum_i l_i \gamma_i}{\sum_i l_i},$$

4. Rotate each edge in the main direction around its middle point so that its convergence is strictly equal to γ_{av}.

5. Reconstruct each secondary edge by building the line going through its middle point and perpendicular to adjacent main direction segments.

After the preprocessing each point gets one of the four types:

1. Ortho — right angle point.
2. Two straights — the angle is formed by two long straight edges which do not intersect at right angle.
3. One straight – only one adjacent edge is long straight while the other is shorter than S.
4. No straights — both adjacent edges are shorter than S.

The following three figures illustrate the preprocessing step and influence of parameters S, $\hat{\alpha}$ and k. This example is constructed by the two polylines adjacent at the south-eastern point. Figure 3 shows the influence of angle tolerance on the detection of right angles. Obviously, the larger is tolerance, the more is the number of detected right angles.

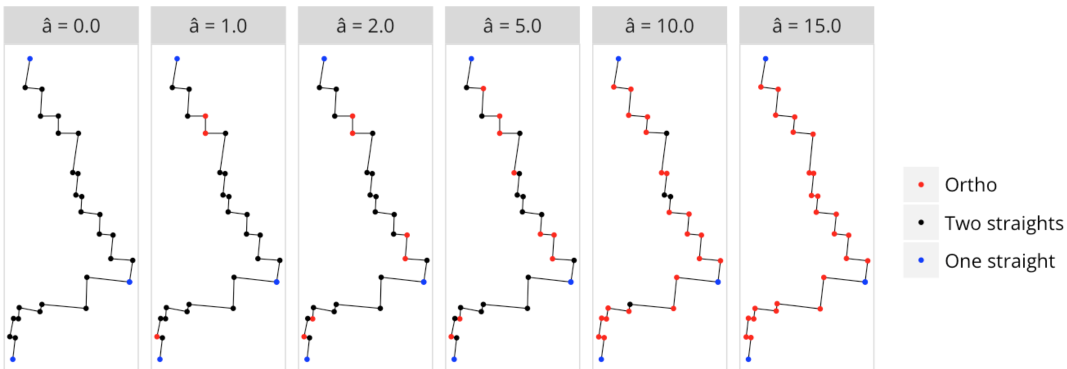

Figure 3: Squaring of detected right angle sequences with different angle tolerances ($k = 3$, $S = 2$)

Figure 4 shows the influence of length tolerance S on the detection of right angles. Here, smaller lengths does not mean larger nummber of right angles. This can be learned by comparing the facets with $S = 1.0$ mm and $S = 2.0$ mm on Figure 4. The angle at black point for $S = 1.0$ is not considered to be right because of the corrected angle tolerance for that angle that depends on the length of shorter edge.

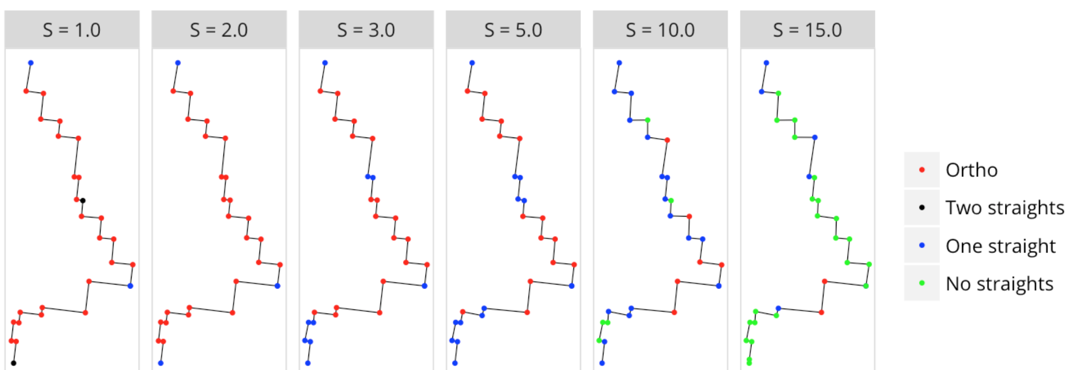

Figure 4: Squaring of detected right angle sequences with different step tolerances ($k = 3$, $\hat{\alpha} = 15°$)

Finally, Figure 5 illustrates the influence of ihibition parameter k on the detection of right angles. Here, larger inhibitions expectedly lead to the increase in the number of detected right angles.

An optimal parameters for this example are $\hat{\alpha} = 15.0°$, $S = 2.0mm$ and $k = 3.0$. They will be used later in results section for the generalization of the whole dataset.

2.2 Simplification of regular line parts

Geometric simplification of regular line parts which are right angle sequences, can be theoretically performed by various approaches. This slot in technological scheme can be fulfilled by methods such as edge-move (Buchin et al., 2011), shortcut (Haunert and Wolff, 2010) or some other technique adopted from building generalization domain.

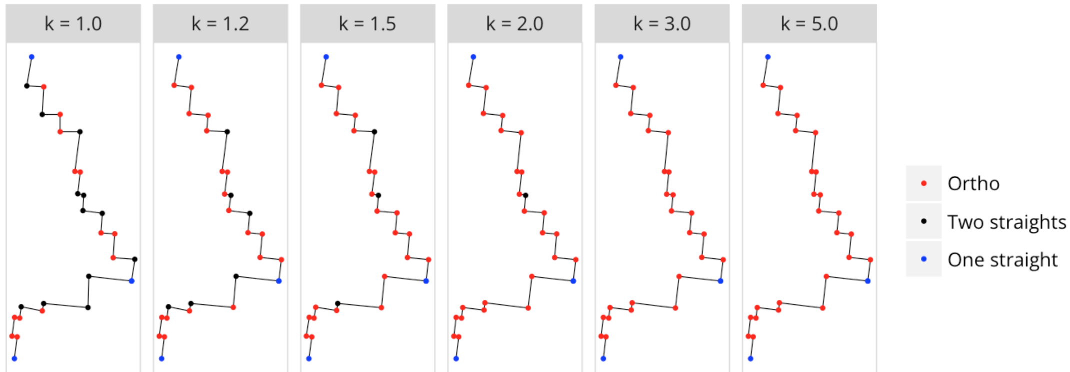

Figure 5: Squaring of detected right angle sequences with different inhibition values ($\hat{\alpha} = 15°$, $S = 2$)

We based the simplification of regular parts on edge contraction procedure. First, user should set the resulting scale which is t times smaller than the source scale. Then edges shorter than S in the resulting scale must be contracted. The same result can be achieved if we fix the preprocessing scale and enlarge S by multiplying it on t for simplification, but this makes the methodology less clear.

To decide how the contraction should be made, a set of possible configurations has been derived. These are classified into Z-like, U-like, endpoint and short configurations in Figure 6. In case of Z- and U-like configurations the contracted edge e_i is replaced by the edge prependicular to it and connected to the e_{i-2} and e_{i+2} or their extensions. Thus, e_{i-1} and e_{i+1} are removed and the line structure is reindexed. Endpoint and short configurations are the special cases. Contraction of the first edge of the orthogonal part is made by shortcut between p_1 and e_3 thus replacing e_1 and e_2 by a single edge. The similar approach is applied in case of the last point. Finally, all 3-point configurations and 4-point with contracted middle edge are replaced by a straight edge connecting endpoints.

The topology is preserved in a simplistic manner: if edge contraction leads to topological error (intersection with other edges), then this contraction is prohibited. We plan to implement a wider set of possible contractions in future. That will allow us to apply alternative strategies in case of topological errors istead of the simple prohibition.

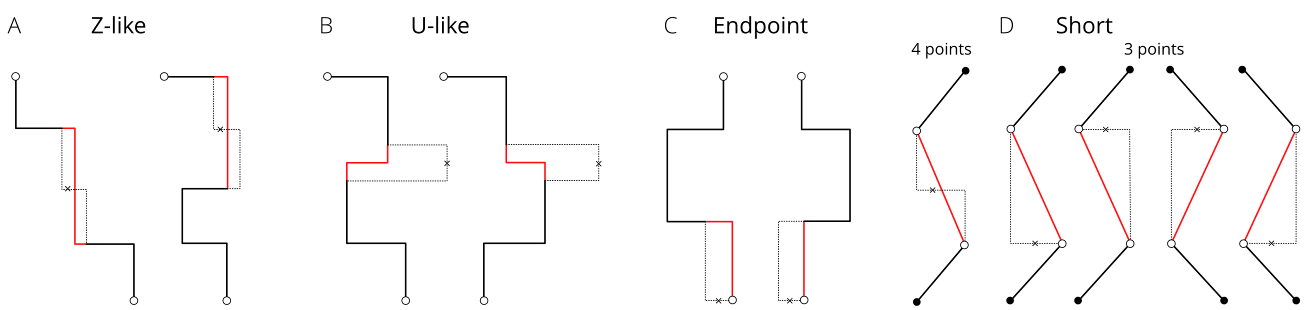

Figure 6: Edge contraction strategies for various configurations of orthogonal line segments. Endpoints are symbolized with hollow dots, old configuration is a dotted line, contracted edge is marked by X sign and the new configuration is depicted in red

The applied simplification algorithm has three parameters:

- resulting scale factor in meters per mm on screen;
- the generalization tolerance d in mm on screen, which is used to parameterize Douglas-Peucker or Li-Openshaw algorithm for simplification of non-orthogonal (irregular) parts of the line.
- the minimum length of the line segment S in mm on screen.

An example simplification of a small line part is presented in Figure 7.

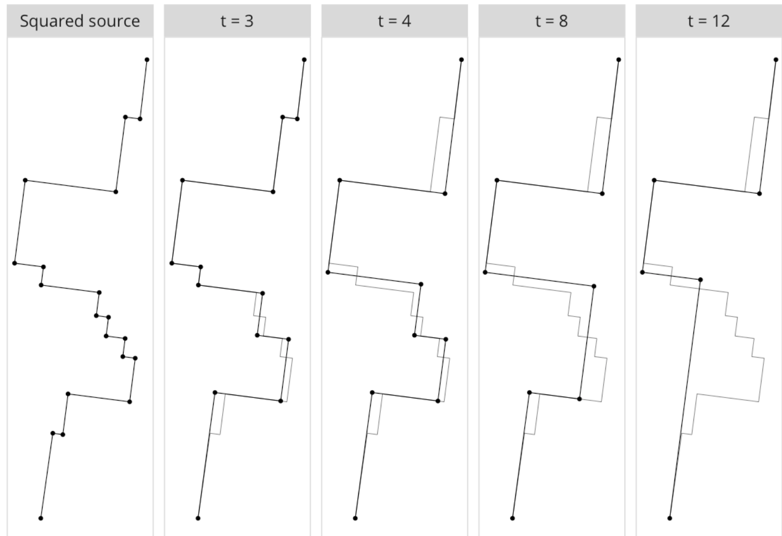

Figure 7: Example simplification of orthogonal line part. t parameter corresponds to decrease in scale.

3 Results and discussion

The above methodology was implemented in a computer program using C++ language. We selected Komi and Arkhangelsk regions to assess the effectiveness of methodology proposed. One can distinguish the three types of line patterns looking at Figure 1:

1. irregular lines, following natural borders (rivers, ridges) or not related to any of them but constructed artificially without regard to any selected azimuth. These occur mostly along the outer border of the area and in the western (Arkhangelsk) part of the selected area

2. long straight lines of arbitrary direction. These occur mostly in the northern part of the region and are so long that will hardly be a subject of generalization

3. ladder-like lines composed of short and middle-length straight edges. These edges are perpendicular to each other and comprise a sequence of of Z- and U-shape steps.

We performed the preprocessing in the source scale (1:1 000 000, 1000 m in 1 mm on the screen) with varying values of d, $\hat{\alpha}$, S and k parameters. Here we will present the results obtained with $d = 0.5$ mm, $\hat{\alpha} = 15°$, $S = 2$ mm, $k = 3.0$ which proved to be effective particularly on this dataset.

After preprocessing the simplification was done with reduction of scale in $t = 2$, 4, 6, 8, 10, 12, 14 and 16 times with $S = 2$ mm and $d = 1.0$ mm parameters. That is, line segments in orthogonal part of the line are contracted if they visible length is smaller than 2 millimeters in a resulting scale. And d is used to parameterize the grid resolution of Li-Openshaw algorithm which is applied in other parts of the line.

The results are presented in Figure 8. For shape comparison we applied Li-Openshaw (1992) and Douglas-Peucker (1973) algorithms globally without line preprocessing and the results are presented in Figures 9 and 10. Generalized Hausdorff distances were calculated in each case for precision assessment and represented in graphic form in Figure 11. This is the average of all Hausdorff distances calculated for each line and its generalized result.

Looking at Figure 8 one can note that the proposed approach emphasizes the orthogonal nature of the corresponding line parts. This can be considered a kind of a typification process applied to the orthogonal line bends. Small steps are aggregated into larger ones which results in an orthogonal pattern reproduced for smaller scale visualization. This algorithm gives intermediate precision of the result, according to Figure 11. Li-Openshaw algorithm is applied in other parts of the line dataset which helps to keep the smooth shape of the natural lines.

Li-Openshaw (Figure 9) algorithm works very well in irregular (natural) parts of the line, giving the line that both smoothly and precisely fits into she shape of the original one. However it fails to represent the shape pattern of the orthogonal line part, making it spineless and round-cornered. This also concerns non-right angles at which long straight segments are intersected. Li-Openshaw algorithm gives the best precision of the result, according to Figure 11.

Figure 8: Line generalization with proposed approach. Non-orthogonal parts are generalized with Li-Openshaw (1992) algorithm

In the opposite, Douglas-Peucker (Figure 10) algorithm works better in orthogonal part of the line and helps to keep its edgy and angular nature. However this edgy approach is spread over the whole dataset making natural line parts also artificially looked. Additionally, Douglas-Peucker algorithm does not preserve right and near-right angles which leads to distortion of the original shape. Douglas-Peucker algorithm gives the worst precision of the result, according to Figure 11 and demonstrates linear dependency between the Generalized Hausdorff distance and the reduction of scale.

Based on the analysis of these fugures we derive the conclusion that the best solution can be obtained by the combination of the three approaches. First, orthogonal parts of the line can be extracted and generalized by specialized algorithm like that is presented in this paper. Next, the remaining part of the line should be subdivided in two sub-parts. The first sub-part should include the sequences of the long straight lines that do not intersect at right angles. These can be generalized by Douglas-Peucker algorithm to keep their edgy nature. Finally, all the remaining line segments can be generalized by Li-Openshaw algorithm to keep their smooth and natural appearance.

Acknowledgements

This research was supported by Russian Basic Research Foundation (RFBR) grant 15-07-06789 A.

References

1. Ai, T. & Li, J., 2010. A DEM generalization by minor valley branch detection and grid filling. ISPRS Journal of Photogrammetry and Remote Sensing, 65(2), pp.198–207.

2. Bayer, T., 2009. Automated Building Simplification Using a Recursive Approach. In Cartography in Central and Eastern Europe. Lecture Notes in Geoinformation and Cartography. Berlin, Heidelberg: Springer, pp. 121–146.

3. Buchin, K., Meulemans, W. & Speckmann, B., 2011. A New Method for Subdivision Simplification with Applications to Urban-area Generalization. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York, NY, USA: ACM, pp. 261–270.

4. Damen, J.; van Kreveld, M.; Spaan, B. High Quality Building Generalization by Extending the Morphological Operators. In Proceedings of the ICA Workshop on Generalization, Montpellier, France, 20–21 June 2008.

5. de Berg, M., van Kreveld, M. & Schirra, S., 1998. Topologically Correct Subdivision Simplification Using the Bandwidth Criterion. Cartography and Geographic Information Science, 25(4), pp.243–257.

6. Dey, T.K., Edelsbrunner H, Guha S., Nekhayev D.V., 1999. Topology preserving edge contraction. Publ. Inst. Math. Beograd N.S., 66, pp.23–45.

7. Douglas, D.H. & Peucker, T.K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: The International Journal for Geographic Information and Geovisualization, 10(2), pp.112–122.

8. Haunert, J.-H. & Wolff, A., 2010. Optimal and topologically safe simplification of building footprints. In D. Agrawal et al., eds. GIS. ACM, pp. 192–201.

9. Lang, T., 1969. Rules for robot draughtsmen, Geographic Magazine, 42(1), 50–51.

10. Li, Z. Algorithmic Foundation of Multi-Scale Spatial Representation. CRC Press, 2006.

11. Li, Z. & Openshaw, S., 1992. Algorithms for automated line generalization based on a natural principle of objective generalization. International Journal of Geographical Information Systems, 6(5), pp.373–389.

12. McMaster, R. B., 1987. Automated line generalisation, Cartographica, 24(2), pp. 74–111.

13. Meulemans, W., van Renssen, A. & Speckmann, B., 2010. Area-Preserving Subdivision Schematization. In Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 160–174.

14. Rosenfeld, A. & Johnston, E., 1973. Angle Detection on Digital Curves. IEEE Transactions on Computers, C-22(9), pp.875–878.

15. Sester, M., 2005. Optimization approaches for generalization and data abstraction. International Journal of Geographical Information Science, 19(8-9), pp.871–897.

16. Teh, C.-H. & Chin, R. T., 1989. On the detection of dominant points on digital curves, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(8), 859–872.

17. Visvalingam, M. & Whyatt, J.D., 1993. Line generalisation by repeated elimination of points. The Cartographic Journal, 30(1), pp.46–51.

18. Wang, A., & Müller, J.-C., 1993. Complex coastline generalization. Cartography and Geographic Information Systems, Vol. 20, No. 2, pp. 96-106.

19. Wang, Z. & Müller, J.-C., 1998. Line Generalization Based on Analysis of Shape Characteristics. Cartography and Geographic Information Science, 25(1), pp.3–15.

Figure 9: Line generalization with Li-Openshaw (1992) algorithm

Figure 10: Line generalization with Douglas-Peucker (1973) algorithm

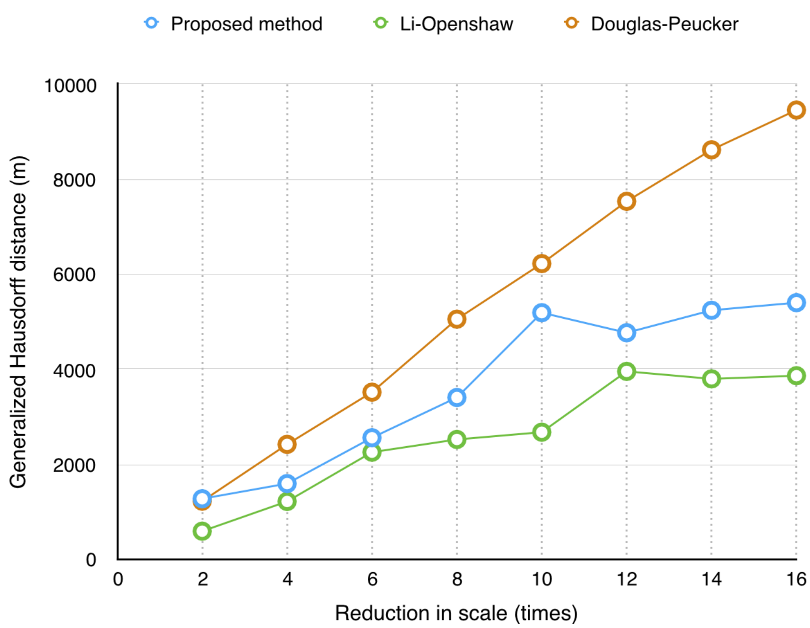

Figure 11: Generalized Hausdorff distances for various algorithms