

Algorithm Of Generalization Based On The Minimum Quadrates Method

¹ Shaxislam Batirovich Joldasov

² Saida Safibullayevna Beknazarova

^{1,2} Tashkent University of Information Technologies named after Muhammad Al- Khwarizmi, 105, A. Temur, Tashkent, 100142, Uzbekistan

Received: 31 Dec 2025 | Received Revised Version: 16 Jan 2026 | Accepted: 29 Jan 2026 | Published: 18 Feb 2026
Volume 08 Issue 02 2026 | Crossref DOI: 10.37547/tajet/Volume08Issue02-12

Abstract

The article proposes an algorithm based on the least squares method for solving problems of simplifying and generalizing many lines. The proposed approach, unlike classical polyline simplification algorithms, does not require the location of the generated nodes at the points of the initial polyline, which increases the generality of the algorithm and improves the accuracy of geometric approximation. It was shown that the algorithm can be used for flexible simplification of contours at different accuracy levels, reduction of noise effects, and optimization of the number of nodes. The research results confirm that the method can be effectively applied in the fields of cartography, computer graphics, and image vectorization.

Keywords: Least squares method, polyline generalization, contour simplification, geometric approximation, vectorization, computer graphics, digital cartography, algorithm.

© 2026 Shaxislam Batirovich Joldasov, & Saida Safibullayevna Beknazarova. This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). The authors retain copyright and allow others to share, adapt, or redistribute the work with proper attribution.

Cite This Article: Shaxislam Batirovich Joldasov, & Saida Safibullayevna Beknazarova. (2026). Algorithm Of Generalization Based On The Minimum Quadrates Method. The American Journal of Engineering and Technology, 8(2), 126–130. <https://doi.org/10.37547/tajet/Volume08Issue02-12>

1. Introduction

The problem of simplification and generalization of many lines in the field of digital image processing, computer graphics, and cartography is one of the important scientific problems. Representing complex contours in a compact form, reducing the number of nodes, and preserving the basic properties of the geometric shape are necessary in many practical tasks. The need for effective generalization algorithms is especially high when working with vector graphics, maps, and technical drawings.

Currently, various algorithms have been developed to simplify polylines, among the most popular being the

Ramer-Douglas-Peucker algorithm and segmentation-based approaches. However, these algorithms usually require that the generated nodes correspond to the original data points, which does not always provide a geometrically optimal solution[1][2],[3][4][5].

The least squares method is one of the widely used optimization methods in the fields of mathematical statistics and signal processing, allowing the determination of a function or line closest to a given set of points. Applying this approach to the problem of polyline generalization makes it possible to increase geometric accuracy and obtain a smooth result. The article develops a generalization algorithm based on the

least squares method, analyzes its advantages and possibilities of application [5].

The problem of simplifying geometric objects plays an important role in increasing the efficiency of storing and transmitting large volumes of graphical data. By optimally reducing complex contours, it is possible to reduce the volume of information, save computational resources, and increase the speed of visualization. Therefore, the improvement of generalization algorithms is one of the urgent scientific directions [6].

Traditional algorithms are often limited to selecting nodes from the original points, which in some cases leads to the appearance of redundant nodes or an inoptimal approximation of the contour shape. As a result, it becomes difficult to ensure a balance between geometric accuracy and the degree of simplification [7].

The approach based on the least squares method allows us to consider the process of generalization from the point of view of optimization. With the help of this method, it is possible to obtain a smoother and mathematically justified result by minimizing the global errors of the contour points. Therefore, the development and research of a generalization algorithm based on the least squares method is relevant from a scientific and practical point of view. Let there exist a multitude of lines represented by a sequence of points located in a given plane. It is required to construct a simplified polyline that

represents this multitude of lines with a smaller number of nodes, but retains its basic geometric shape [8].

2. Method

The main goal of the problem is to determine the segments of a simplified polyline in such a way that they provide optimal convergence according to the criterion of least squares with respect to the set of initial points. In this case, the resulting nodes are not necessarily located at the points of the original polyline. Also, the algorithm must satisfy the following requirements: minimization of geometric error; reduction of the number of nodes; preservation of the main properties of the form; ensuring computational efficiency.

$\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\} J(\vec{x}, \vec{a}) = \sum |\vec{a} \times (\vec{x} - \vec{x}_1)|^2 J \vec{x} \vec{a} \vec{a} = (x_a - y_a)$ Let's look for a convergent line for a set of points on a plane using the least squares method. This algorithm is used as an auxiliary procedure for image generalization. Let a set of points be given on the plane. Some line l is also given. As is known about this line, the point $x \beta$ lies on it and is parallel to the unit length vector $a \beta$. The sum of the square distances from the points to the line can be written as, where " \times " is the product of the vector. The search for a convergent line using the least squares method consists of minimizing with respect to a and β . In addition, we will consider the form by components. Expression (1) can be written explicitly [9].

$$J(\vec{x}, \vec{a}) = \sum |\vec{a} \times (\vec{x} - \vec{x}_1)|^2 = \sum_{i=1}^n (x_a(y - y_i) - y_a(x - x_i))^2 = \sum_{i=1}^n (x_a^2(y - y_i)^2 - 2x_a y_a(x - x_i)(y - y_i) + y_a^2(x - x_i)^2) \tag{2}$$

At the minimum point, the gradient of the function must be equal to zero (a necessary condition for the extremum). Therefore, we can write:

$$\frac{\partial J}{\partial x} = 2ny_a^2x - 2y_a^2 \sum_{i=1}^n x_i - 2nx_a y_a y + 2x_a y_a \sum_{i=1}^n y_i = 0. \tag{3}$$

Let's add an additional symbol:

$$x^* = \frac{1}{n} \sum_{i=1}^n x_i$$

$$y^* = \frac{1}{n} \sum_{i=1}^n y_i \tag{4}$$

Replacing these symbols with (3), making simple transformations, we obtain

$$(x - x^*)y_a - (y - y^*)x_a = 0, \tag{5}$$

$x^* y^* \vec{a} x_a \vec{a} - \vec{a}$ from which it follows that the point lies on the convergent line (\cdot). Since we know the point lying exactly on the convergent line, it remains only to find the optimal value. Since it is known that the length of this vector is one, it is sufficient to find only. It should be noted that at least one of the two vectors and has a non-negative y-component [10].

Replacing the obtained results with (2), we write the final formula of the problem:

$$J(x_a) = \sum_{i=1}^n (x_a^2(y^* - y_i)^2 - 2x_a \sqrt{1 - x_a^2}(x^* - x_i)(y^* - y_i) + (1 - x_a^2)(x^* - x_i)^2) \rightarrow min. \tag{6}$$

Let's make the transformations:

$$\begin{aligned}
 b &= \sum_{i=1}^n (y^* - y_i)^2 - \sum_{i=1}^n (x^* - x_i)^2, \\
 c &= 2 \sum_{i=1}^n (x^* - x_i)(y^* - y_i), \\
 d &= \sum_{i=1}^n (x^* - x_i)^2.
 \end{aligned}
 \tag{7}$$

Using these transformations, we write the required minimum condition:

$$J'_{x_a} = 2bx_a + c \frac{1-2x_a^2}{\sqrt{1-2x_a^2}} = 0.
 \tag{8}$$

$x_a \in [-1, 0, 1]$ This equation is easily reduced to a biquadratic equation, which can be solved by certain methods. After solving it, you need to select the minimum value of when each root is equal to (8).

The complexity of solving this problem is equal to the complexity of calculating x^* , y^* , b, c and d , i.e., $O(n)$. An example of a predictive line is shown in Figure 1.

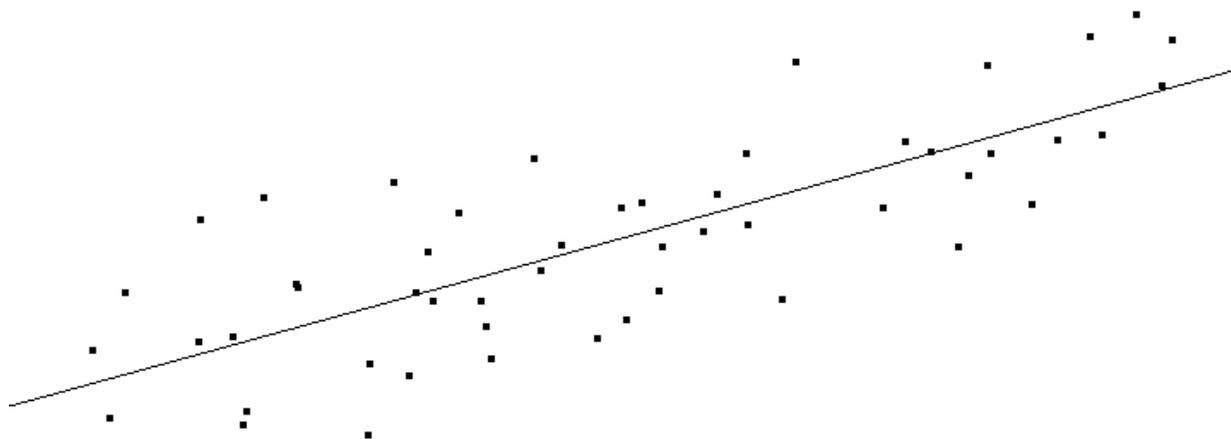


Figure 1. Approaching a straight line.

Algorithm for polyline generalization using the least squares method.

A generalization algorithm based on solving the problem from the previous section is described below. This algorithm is similar to the one described in [4], which also belongs to the "Greedy" and heuristic class. To generalize the polyline, you need to perform the following actions.

1. P_1, P_2, \dots, P_n polyline is given to the input. The current peak is indicated by 1.
2. When $i \neq n$, perform the following subparagraphs.
 - 2.1. $j \in \{P_i, P_{i+1}, \dots, P_j\}$ $\epsilon_i = 1$ $P_1 j = n$ $P_n i = 1$ $j = n$ The maximum, is found in such a way that the approximate line taken by the LSM for the set of points, is separated from each of these points by a distance not greater than (control parameter). If, then it is required to pass through the line, similarly, if, then it is required to pass through the line. Therefore, if and, then there exists only one possible line and optimization is not performed.
 - 2.2. i, j The found string is "remembered," the variable is assigned a value.
3. Construction of the obtained broken line. In the cycle, all pairs of consecutively converging lines are checked.
 - 3.1. $l_1 \{P_i, P_{i+1}, \dots, P_j\}$ $l_2 \{P_j, P_{j+1}, \dots, P_k\}$ Let the first row be close to the set of points, and the next row to the set. It will be known whether there is a point of intersection of these two lines.

3.2. $P_j \varepsilon P_{j1} P_{j2}$ If a point of intersection exists and it is separated from and not exceeding, then it is added to the resulting polyline; otherwise, two points are added: from perpendicular to and from perpendicular to.

4. $P_1 P_n$ The starting point and the endpoint are added to the resulting polyline. End of the algorithm.

$jO(N^2)O(N \log N)$ The complexity properties of this algorithm are similar to those of the algorithm [4]. If we search for the global maximum in step (2.1), then the complexity will be. If we sacrifice our perseverance and use binary search, we get hard work and a slightly worse result.

The difference between this algorithm and those described in section (2.3) is that the nodes of the resulting polyline do not necessarily lie on the original polyline. In this sense, the described algorithm is more general. The results of this algorithm are shown in Figure 2.

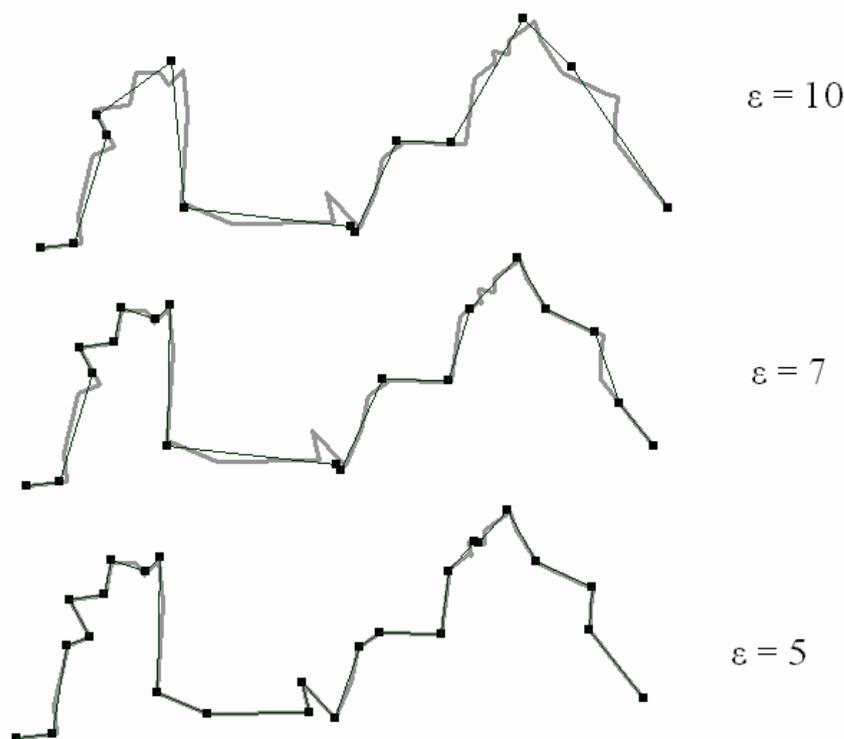


Figure 2. Generalized refracted lines for different values of ε .

The figure shows the results of the generalization algorithm based on the least squares method for various accuracy parameters ($\varepsilon = 10, \varepsilon = 7, \varepsilon = 5$). From the results, it can be seen that as the value of ε increases, the polyline becomes simpler, i.e., the number of nodes decreases, and the contour is represented by larger segments while maintaining the general shape. In the case of $\varepsilon = 10$, maximum simplification is observed, in which some local curvatures may disappear, but the global geometry of the object is preserved. At the value of $\varepsilon = 7$, a balance between accuracy and simplification is ensured, and the main features of the shape are sufficiently expressed. In the case of $\varepsilon = 5$, the result closest to the original contour is obtained, but the number of nodes is greater.

An important advantage of this algorithm is that the resulting polyline nodes do not necessarily lie at the points of the original polyline. This provides a more general approach compared to classical simplification algorithms and allows for the mathematically optimal approximation of the contour shape. As a result, the impact of noise is reduced, the curves appear smoother, and geometric accuracy is improved.

Thus, the results of the drawing confirm that the generalization algorithm based on the least squares method allows for flexible simplification of the contours, control of the number of nodes, and preservation of the basic geometry of the shape. This shows that the

algorithm can be effectively applied in cartography, computer graphics, and vectorization.

The article proposes an algorithm based on the least squares method for generalizing multiple lines. The proposed approach, unlike classical simplification algorithms, does not require the placement of generated nodes at starting points, which increases the generality of the algorithm and improves geometric accuracy. It was shown that the algorithm can be used to simplify the contours to varying degrees, reduce the impact of noise, and optimize the number of nodes.

The research results confirmed the flexibility and effectiveness of the method and showed its applicability in such areas as computer graphics, cartography, image vectorization, and object contour modeling. In the future, it is advisable to develop the algorithm in the direction of selecting adaptive parameters and generalization for three-dimensional objects.

References

1. Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8 (6), 679-698.
2. Douglas, D., & Peucker, T. (1973). Algorithms for reducing the number of points required to represent a digitized line. *The Canadian Cartographer*, 10 (2), 112-122.
3. Unser, M. (1999). Splines: Perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16 (6), 22-38.
4. Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.
5. Suzuki, S., & Abe, K. (1985). Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*.
6. Achanta, R., et al. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (11), 2274-2282.
7. Freeman, H. (1974). Computer processing of line drawing images. *ACM Computing Surveys*, 6 (1), 57-97.
8. Xu, C., & Prince, J. (1998). Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7 (3), 359-369.
9. Beknazarova, S., Abdullayeva, O., Abdullayev, S., Abdullayev, Z. Image processing: Face recognition by neural networks/E3S Web of Conferences 587
10. Beknazarova, S., Abdullayev, S., Abdullayeva, O., Abdullayev, Z. Machine Learning Method for Predicting Human Movements/AIP Conference Proceedings 3244 (1)