ReBankment: Displacing Embankment Lines From Roads and Rivers With a Least Squares Adjustment

Guillaume Touya, Imran Lokhat

REBANKMENT: DISPLACING EMBANKMENT LINES FROM ROADS AND RIVERS WITH A LEAST SQUARES ADJUSTMENT

A PREPRINT

Guillaume Touya
LASTIG, Univ Gustave Eiffel, IGN, ENSG
Champs-sur-Marne, F-77420
guillaume.touya@ign.fr

Imran Lokhat
LASTIG, Univ Gustave Eiffel, IGN, ENSG
Saint-Mandé, F-94160
imran.lokhat@ign.fr

November 2, 2021

ABSTRACT

While the recent progress on automated generalisation helped National Mapping Agencies to derive topographic maps more and more quickly, there are still practical cartographic issues that require attention. For instance, embankments are represented with line symbols showing the slope of the embankment. This paper proposes an automated algorithm, called ReBankment that displaces the embankment lines from the roads and rivers that overlap the embankment symbol. ReBankment is based on a triangulation to identify neighbourhoods, and on a least squares adjustment to displace and distort the embankment line while preserving its shape. The paper also proposes how to handle complex cases and scaling issues. ReBankment is tested on real data from a 1:25k scale topographic map.

***K*eywords** map generalisation and least squares and displacement and topographic map and embankment

1 Introduction

Topographic maps, particularly at the 1:25k scale, should provide a precise and salient representation of the terrain, as they are used for trekking, surveying or by rescuing services. As a consequence, IGN France, the French National Mapping Agency (NMA), uses shaded relief, contour lines, and height points to portray the terrain. Embankments are a specific terrain form that can be too small to be conveyed through contour lines. But embankments are important features because they can be complex to cross, so they are represented with a specific lines, with additional regular lines on the side of the slope. As we can see on the right side of Figure 1, this symbol demands space on the map. And it is particularly significant because these embankments are often located along roads or rivers, which are also portrayed with wide symbols, as we can see on the left side of Figure 1 where the embankment lines are hidden behind these road/river symbols. This is why we need to displace, and sometimes compress (or shrink) these lines away from roads and rivers.

The current speed of updates at NMAs pushes for map production processes that are as automated as possible, because up-to-date maps have to be published online more frequently.And even when only one scale is targeted, as in our use case, it is important to generalise the map automatically if it is released too or three times a year. So our challenge is to provide a fully automated displacement algorithm, which does not require further interactive corrections. Most displacement algorithms are dedicated to rigid features such as buildings, which are not distorted while being displaced. As a consequence, we need to design a new algorithm able to generate elastic displacements, where the embankment line is sometimes compressed while moved away from the road, to preserve its initial shape. Another challenge is to

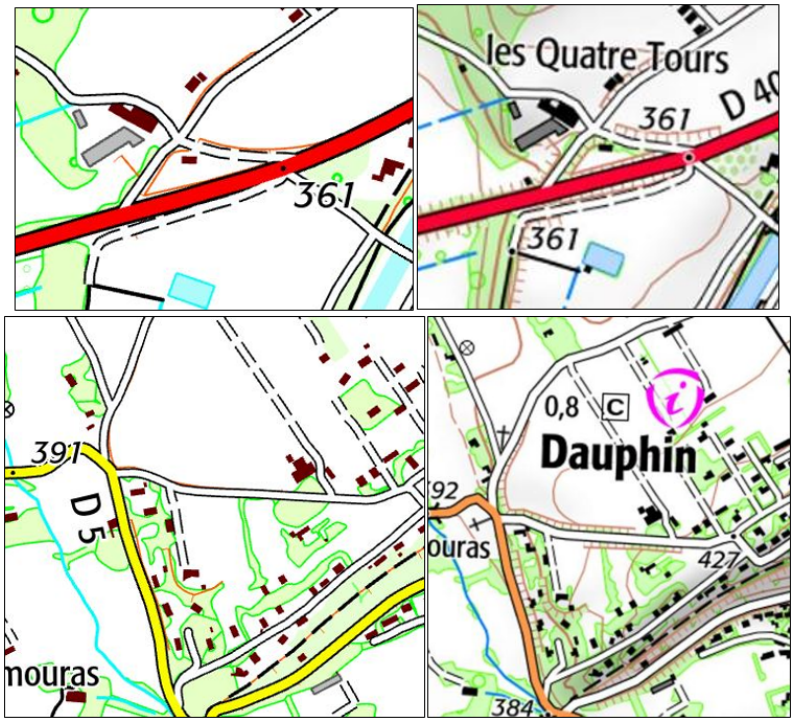

Figure 1: Two extracts of IGN topographic maps, without embankment line displacement on the left, and with semi-automated process on the right.

design a process that could be made quickly available in the new automated workflow to produce topographic maps at IGN. This second challenge has constrained both the design and the implementation of the algorithm. Our proposition is the ReBankment algorithm: a least squares based algorithm to achieve an elastic displacement of the lines. It follows the principles of least squares based generalisation [1, 2], with constraints and pre-processes specifically designed for embankment lines generalisation.

The paper is structured as follows. Section 2 discusses related work on displacement and least squares adjustment in map generalisation. Then, the ReBankment algorithm is presented in Section 3, with the main principles, and the specific processes for complex cases. Results are presented and evaluated in Section 4. Finally, Section 5 draws some conclusions and discusses future research.

2 Related work

As map symbols quickly overlap when scale decreases for map features that are close to each other on the ground, automating the displacement of map features has been one of the first goals of researchers on map generalisation [3]. To displace network lines such as roads, rivers, or railways, the first proposition is to identify the overlap conflicts to derive adapted displacement vectors to the vertices of the lines [4]. Automated displacement of cartographic lines makes a leap forward with the proposition of geometric optimisation techniques such as the Snakes [5], and later the Elastic Beams [6, 7].

But most of the effort made by researchers on automated displacement focused on the displacement of buildings or point symbols inside a block [8], as buildings usually attract more attention in map generalisation research. The first methods were sequential: finding the best building to displace, or the worst conflict to solve, then displacing this building, and then iteratively displacing other buildings until all conflicts are solved [9]. Several algorithms were proposed in recent years following this iterative or sequential approach [10, 11, 12]. There were more global approaches proposed such as the computation of a vector field taking all conflicts and relations to preserve in a block into account [13]. And similarly to line displacement, optimisation approaches were proposed, following the same principles as the Snakes [14], or as the Elastic Beams [15], or using well known meta-heuristic with genetic algorithms [16].

Regarding the generalisation of natural terrain lines, there are very few specific algorithms proposed. Elastic displacements could be achieved using the GAEL model [17, 18]. Previously at IGN, an adaptation of the Elastic Beams [7]

was proposed to displace embankment lines in 1:25k topographic maps, but the algorithm required a significant amount of manual corrections [19].

Least squares based map generalisation has proved to be flexible enough to optimise the elastic displacement of line features such as embankment lines. The principles of least squares based map generalisation have been simultaneously proposed by Lars Harrie and Monika Sester 20 years ago [1, 2], and then improved to provide flexible models able to generalise different types of map features [20, 21, 22]. Then, similar least squares based techniques were adapted to other of geometrical distortions controlled by constraints: data conflation [23, 24, 25], line morphing [26], interactive focus maps [27], or polygon squaring [28]. This past success and the maturity of the approach convinced us to reuse it for our problem of elastic displacement of embankment lines.

3 ReBankment Algorithm

The ReBankment algorithm that reshapes the embankment lines with a least squares adjustment, is presented in this section. The principles of least squares based generalisation are presented in the first subsection. Then, the constraints proposed for embankment line displacement are described in Section 3.2.

3.1 Principles of Least Squares Based Displacement

As explained in the previous section, the principles of Least Squares generalisation were simultaneously proposed by Lars Harrie [1, 20], and Monika Sester [2, 22] twenty years ago. The idea is to translate the contradictory constraints of map generalisation into an overconstrained system of linear equations, and then an optimal solution for this system is found using a least squares adjustment. Equation 1 shows how the constraints of generalisation are translated into a linear function f whose unknowns are the differences between the coordinates (x_i, y_i) of vertex i of a geometry being generalised, before and after generalisation.

$$f(x_1, y1, ..., x_n, y_n) = c_{1,1} \cdot \Delta x_1 + c_{1,2} \cdot \Delta y_1 + ... + c_{n,1} \cdot \Delta x_n + c_{n,2} \cdot \Delta y_n + c \tag{1}$$

where $\Delta x_i = x_i - x_i^0$, x_i is the abscissa of the i^{th} vertex of the generalised line, and x_i^0 is the abscissa of the i^{th} vertex of the initial line.

Some constraints of map generalisation, e.g. constraints on distances, or angles between map features, cannot be translated into linear f functions, and in this case, we simply approximate the function using the first-order Taylor polynomial of the function f:

$$\frac{\partial f}{\partial x_1} \cdot \Delta x_1 + \frac{\partial f}{\partial y_1} \cdot \Delta y_1 + \frac{\partial f}{\partial x_2} \cdot \Delta x_2 + \frac{\partial f}{\partial y_2} \cdot \Delta y_2 + ... + \frac{\partial f}{\partial x_n} \cdot \Delta x_n + \frac{\partial f}{\partial y_n} \cdot \Delta y_n = c - f(\vec{x^0}) \tag{2}$$

where $\vec{x^0}$ is the vector of the initial coordinates of vertices $1..n$. When we merge all the small system of equations from all constraints on all vertices of a map portion, we just have to solve the following equation (with a matrix notation):

$$\vec{A} \cdot \vec{x} = \vec{l} + \vec{v} \tag{3}$$

where $\vec{A}$ is the Jacobian Matrix of the system containing the $c_{i,j}$ values from Equation 1, $\vec{x}$ is the unknown vector, $\vec{l}$ is the vector of constant values c from Equation 1, and $\vec{v}$ is the residual vector that contains the errors for each constraint. Then, the least squares solution of Equation 3 is given in Equation 4:

$$\vec{x} = (\vec{A^T} \cdot \vec{P} \cdot \vec{A})^{-1} \cdot \vec{A^T} \cdot \vec{P} \cdot \vec{l} \tag{4}$$

where $\vec{P}$ is the matrix that contains the relative weights of the constraints. As many constraints are not linear, the first-order Taylor polynomial is not always a good approximation, and this equation does not give an optimal solution. This is why we choose an iterative approach and several iterations of solving Equation 4 are necessary. After each iteration $\vec{x^0}$ becomes the vector of coordinates from last iteration, instead of the vector of the initial coordinates. The algorithm converges either when we reach a maximum number of iterations, or when the norm of $\vec{x}$ is negligible, i.e. the line did not moved so much during the last iteration.

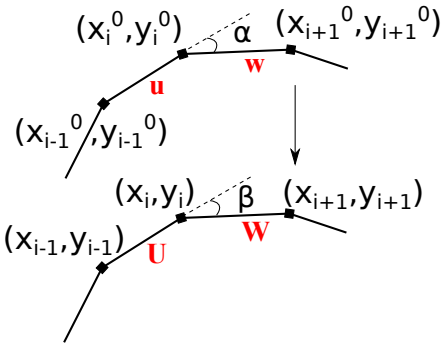

Figure 2: A polyline to displace on top, and the displaced version below. The curvature constraint forces angles α and β to stay equal, as well the norms of vectors $\vec{u}$ and $\vec{U}$.

3.2 Constraints for Embankment Lines

There are four constraints that are introduced to displace embankment lines while preserving their shape and their relative positions:

- (C_1) position preservation constraint,
- (C_2) curvature preservation constraint,
- (C_3) symbol overlap constraint,
- (C_4) relative position constraint [29] between two embankment lines.

These four constraints, and their translation into linear equations following equation 1, or into f functions linearized following equation 2, are presented in the next four sub-sections.

3.2.1 Position Preservation Constraint

Constraint (C_1) is classical in map generalisation: generalisation transformations should not move map features far from their initial position, which represent their exact position on the ground. This constraint is translated into a system of linear equations as previously proposed by [1] and [2]: the displacement of all vertices should be equal to zero (Equation 5).

$$\{ \Delta\ x_i = 0 \Delta y_i = 0 \tag{5}$$

3.2.2 Curvature Preservation Constraint

Constraint (C_2) intends to preserve the curvy or straight shapes of embankment lines. It derives from a similar constraints previously proposed by [1]. This constraint applies to two consecutive segments of polylines, and constrains both segment length and the angle between the segments (Figure 2). We use the vector product of vectors $\vec{u}$ and $\vec{v}$ from Figure 2 (and their counterpart in the displaced polyline) to constrain the angle (Equation 6). These two equations are not linear, so they are linearized using the first-order Taylor polynomial.

$$\{ \parallel \vec{u} \wedge \vec{w} \parallel = \parallel \vec{u} \parallel \cdot \parallel \vec{v} \parallel \cdot \sin\alpha \parallel \vec{U} \wedge \vec{W} \parallel = \parallel \vec{U} \parallel \cdot \parallel \vec{W} \parallel \cdot \sin\beta \tag{6}$$

Equation 7 shows how the segment length is constrained to stay the same during the displacement. Once again, the equation is not linear and is linearized using the first-order Taylor polynomial.

$$\parallel w \parallel = \parallel W \parallel \Leftrightarrow \sqrt{(x_{i+1} - x_i)^2 - (y_{i+1} - y_i)^2} - \sqrt{(x_{i+1}^0 - x_i^0)^2 - (y_{i+1}^0 - y_i^0)^2} = 0 \tag{7}$$

3.2.3 Symbol Overlap Constraint

With constraint (C_3), we want the embankment lines to be distant from the road or river centreline r_k of a value $distmin_k = width_k + separation$ where $width_k$ is equal to half the symbol width of the road or river, and $separation$

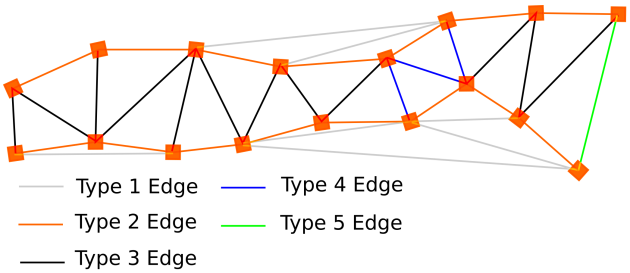

Figure 3: The constrained triangulation used to constrain the relative position of several embankment lines: type 1 edges connect two nodes of the same embankment, so are ignored; type 2 edges are the segments of the embankments, so are ignored too; type 3 edges connect two lines with a distance above minimum threshold but below influence threshold; type 4 edges connect two lines with a distance below the minimum threshold (the segment should be enlarged); type 5 edges connect two lines but with a distance above the influence threshold (the segment is not constrained).

is a minimum separation threshold for legibility (usually set to 0.1 mm on the map). As the symbol width varies according to road or river importance, this $distmin_k$ values varies from one road or river to another.

$$dist(v_i, r_k) = max(distmin_k - dist(v_i^0, r_k), 0) \tag{8}$$

where v_i is i$^{\text{th}}$ vertex of an embankment line to be displaced, and v_i^0 is the initial position of i$^{\text{th}}$ vertex. This equation should be computed for each vertex and for each road, which would generate a very large number of unnecessary equations. This is why we limit this constraint to the roads that lie nearby an embankment line (see Section 3.3 for more details). With the computation of a distance between a point and a line, this constraint is clearly not linear, and once again, it is linearized using the first-order Taylor polynomial.

3.2.4 Relative Position Constraint Between Two Embankment Lines

Constraint (C_4) seeks to preserve the relative position between two embankments lines, when one or more are displaced because of (C_3). It is based on a constrained Delaunay triangulation between all the vertices of the embankment lines, and the segments of embankments are constrained to be edges of the triangulation. Then, a distance constraint following Equation 7 is added on some edges of the triangulation with constant values depending on the type of edge (Figure 3). The five types of edges in this constrained triangulation are the following:

- *Type 1 Edge*: the edges that connect two nodes from the same embankment line. There is no need to constrain such edges, so they are ignored.
- *Type 2 Edge*: the edges that are segments of one embankment line. These segments are already constrained by the curvature constraint, so there is no need to double this effect, and the edges are ignored.
- *Type 3 Edge*: the edges that connect two different embankment lines with a length above the minimum distance threshold, but below the influence threshold. These edges are constrained to preserve this distance.
- *Type 4 Edge*: the edges that connect two different embankment lines with a length below the minimum distance threshold. These edges are constrainted to be enlarged to the minimum threshold.
- *Type 5 Edge*: the edges that connect two different embankment lines with a length above the influence threshold. It means that the lines are too distant from each other to care if they get a little closer, and these edges are not constrained.

Similarly to (C_3), constraint (C_4) can generate heavy computations if the constrained triangulation contains all the embankment lines of the dataset. This is why the triangulation is computed on dedicated areas (Section 3.3 for more details).

3.2.5 Weight Setting

As shown in Equation 4, a weight matrix is used to solve the system of equations, to deal with the relative importance of each constraint. These weights are usually complex to set up [21]. Table 1 shows the weights that we empirically defined to obtain our best results. We carried out a sensitivity analysis, making small variations of each weight, to determine the weight values that provide the best overall results. The fact that (C_3) has the highest weight shows that

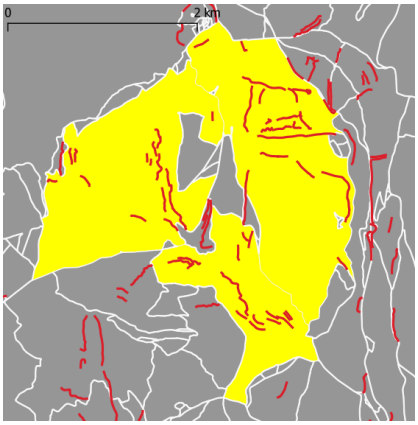

Figure 4: Three of the largest meshes in our test area, showing proving that computing the adjustment on meshes enables the scaling of the process.

displacement remains the most important goal of the algorithm, and we prefer a displaced but slightly distorted line, to a preserved shape but with insufficient displacement.

constraint	(C_1)	(C_2)	(C_3)	(C_4)
weight	8	8	200	15

Table 1: Optimal weights for the four constraints to displace embankment lines.

3.3 Scaling the Adjustment

The least squares based generalisation techniques require the inversion of large matrices (see Equation 4), so the number of features that can be included at once is limited [25]. More generally, map generalisation is computationnally intense, and it is usual to partition the dataset, and avoid processing everything at once [30]. In this case, roads and rivers form natural boundaries to partition the dataset, as embankment lines are not supposed to cross such features. So, we first build a mesh from the roads and rivers, which are integrated in a single planar graph.

Figure 4 shows three of the largest faces of the road network (in yellow) in our test dataset (see Section 4). Although the number of embankment lines remains small in each mesh, the number of roads and rivers can be very high, and even too high to process these large meshes quickly. So, to generate smaller partition, we build and merge the spatial neighbourhood of the embankment lines, as proposed previously to partition data for map generalisation [31]: (1) we create a buffer area around each embankment line, with a width $3 \cdot distmin_k$; (2) we merge the intersecting buffer areas; (3) we select the roads and rivers contained in each merged area to obtain our partition unit.

3.4 Complex Cases

Working on real datasets with countrywide production ambitions implies additional difficulties. In this case, testing ReBankment on a large dataset highlighted four different complex cases that we had to deal with:

1. connected embankment lines;
2. embankment lines touching roads (or rivers);
3. cases without solution because of a lack of space;
4. errors in input data (embankments crossing each other, or crossing a road)

In the experimented dataset, complex cases with one of these four types occurred for 5% of the embankment lines. Figure 5 shows several instances of connected embankment lines, or embankments represented with several connected lines (complex case (1)). In this case, rather than processing lines independently, which can cause disconnections, we prefer adding a pre-process that merges this lines into one.

The complex case (2) is typical of the abstraction process of the surveyors that captured these embankment lines: embankments that lie close to a road are extended to the road centreline, and embankments that go along a road very

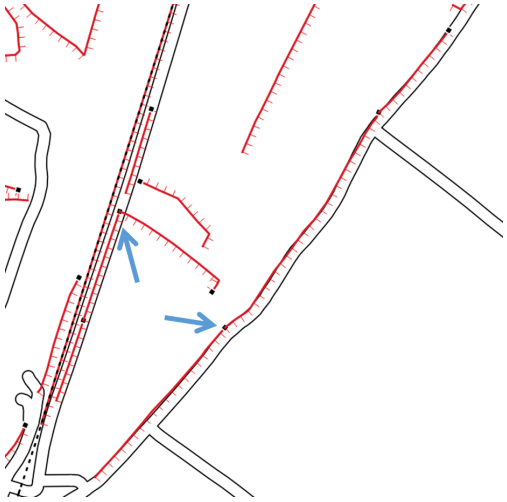

Figure 5: Some embankment lines are connected in the dataset, and must be merged first to avoid their separation during the adjustment process.

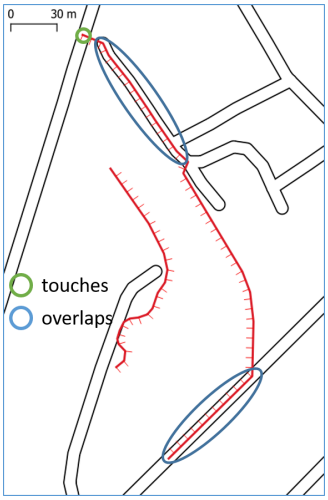

Figure 6: Examples of *overlap* and *touch* spatial relations between embankment lines and roads.

closely are also snapped to the road centreline geometry (Figure 6). This is a complex case because the symbol overlap constraint does not handle in the equations. These two cases are identified by qualifying the topology relations of each embankment line with the nearby roads (and rivers). The embankment lines with a *touch* or *overlap* relation [32] are then pre-processed to moved them a little from the roads. Lines are then disjoint. Figure 7 shows how lines are moved a little: a translation vector is computed using the orientation of the first segment of the line, and a norm of 1 meter to keep the translation minimal. For *overlap* relations, the general orientation of the overlapping part is computed and the direction of the translation is perpendicular to this orientation.

The complex case (3) is classical in map generalisation, as road and river symbol are wide, leaving sometimes no blank space to draw the symbols of additional features (Figure 8). We use an approach similar as the one proposed in CartACom [29] to compute free space around each embankment line. A buffer is computed around the embankment line, and the polygons created by the symbols of roads and embankments are removed from the buffer, leaving only the available blank space. When this blank space is too small, as in Figure 8, the ReBankment algorithm is not triggered, the embankment line is simply eliminated.

Finally, the complex case (4) corresponds to errors in the way embankments are captured as vector lines. We identified two types of errors in our test dataset: embankment lines that cross a road or a river (Figure 9), and embankment lines that intersect each other (Figure 10). In both cases, the situation cannot exist on the terrain, and these cases are not correctly handled by the symbol overlap and relative position constraints. In the case of embankment lines crossing a

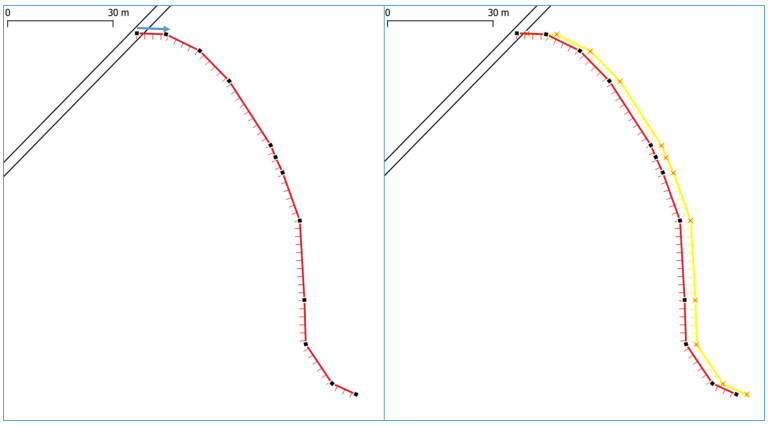

Figure 7: Example of an embankment line touching a road. On the left, the blue arrow shows the translation direction following the direction of the first segment of the line. The right image shows the small translation used to detach the embankment line from the road.

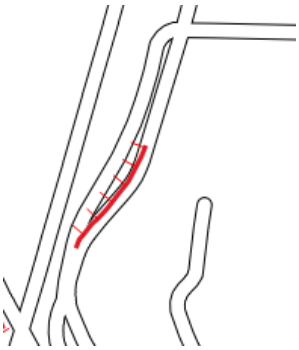

Figure 8: Some cases are too constrained to enable a valid displacement of the embankment line without displacing the roads.

road (Figure 9), we cut the line in two at intersections with roads, and then process the cut lines as instances of the *touch* case (2).

In the case of embankment lines intersecting each other (Figure 10), we select one of the intersecting line, and move it from the others with the same method as the one presented in Figure 7. The other lines are then merged using the pre-process presented for complex case (1).

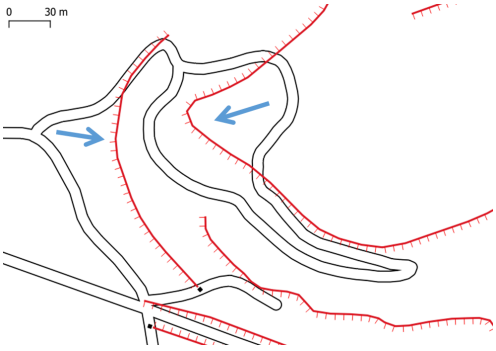

Figure 9: Two examples of embankment lines crossing a road.

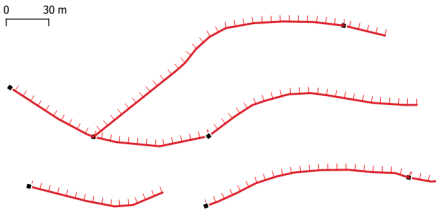

Figure 10: Example of embankment line crossing (or touching) another embankment line, which cannot reflect an existing morphology of the terrain.

4 Implementation and Results

4.1 Use Case and Implementation

Dataset: 55,000 m^2 in the Alps, in France, extracted from the 1:25k base topographic map produced by IGN France. 4,258 embankment lines. Mostly a rural area, with a few towns.

The ReBankment algorithm was first implemented in, the open map generalisation research platform, on top of the least squares mechanisms already available in the platform [33]. Then, to make the algorithm usable in the production process of IGN France, based on ArcGIS Pro and PostGIS, the algorithm was ported in Python with the open source libraries Numpy[1] and Shapely[2].

4.2 Results

Processing all the lines contained in our test dataset took us approximately 45 minutes.

Some results are presented in Figure 11. We can see that the lines are correctly moved from roads, while preserving the shape of the embankment line, even when it requires compressing a bend, such as the bottom embankment line in the top right corner of Figure 11. The bottom right image of Figure 11 shows that ReBankment is also capable of solving proximity conflicts between embankment lines that are too close to each other.

4.3 Evaluation

Our generalisation is specified by four constraints, and we basically evaluate the algorithms using monitors for these constraints [34, 35], i.e. checking for each map feature if all the constraints are satisfied.

To verify if (C_2) is satisfied, we need to measure the similarity between the embankment lines before and after generalisation. In this case, usual line distances do not apply: they would measure displacement more than distortions. So, to compare the shape similarity only, we first translate the initial line, using a vector from the middle of the initial line, to the middle of the generalised line. Then, we compute the following line similarity measures:

- Hausdorff distance (m_1);
- discrete Fréchet distance [36] (m_2);
- dynamic time-warping similarity [37] (m_3);
- average distance [38] (m_4).

We use these four measures because they may highlight different kinds of dissimilarities. (m_1) and (m_2) measure maximum gaps between both lines, but (m_2) is more sensitive to length differences. (m_4) measures the average separation between the lines, while (m_3) measures the similarity in the distribution of vertices in the lines. All four values are expressed in meters. The results of this evaluation are presented in Table 2. These results show that the lines are significantly distorted, and not only translated, which was expected as distortion often gives better results in areas constrained by road symbols. However, the distortion remain limited for most features, with a significant amount (a third) of features having a negligible distortion, and only 10 to 15 % of the features suffering from large distortions.

The measures used for (C_2) also apply for (C_1) when there is no translation prior to the similarity measure. The results show that the lines are often displaced, the constraint did not prevent displacement, it only limited it. The median

[1] https://numpy.org/

[2] https://pypi.org/project/Shapely/

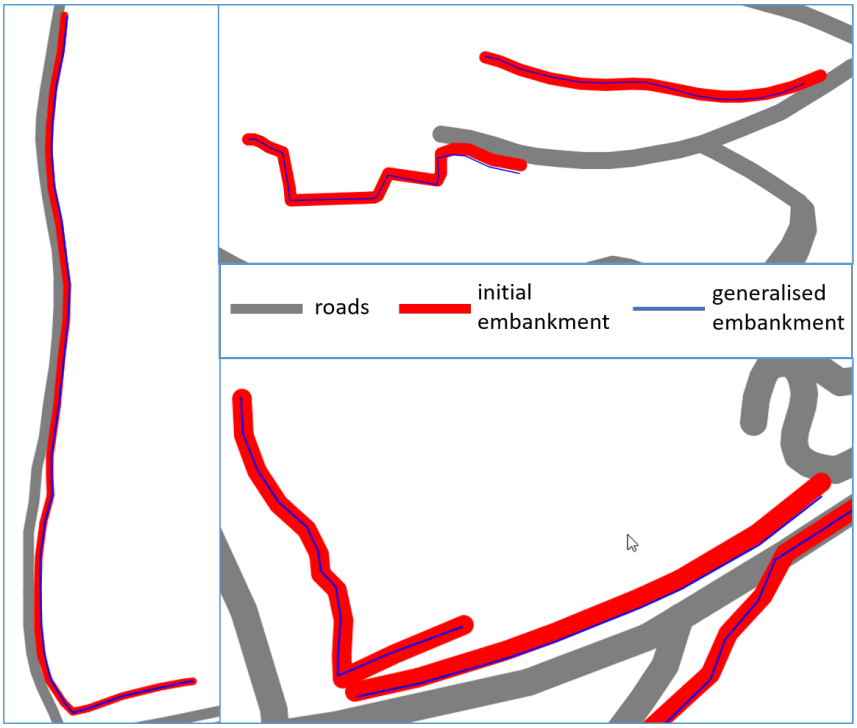

Figure 11: Three examples of displacement obtained with ReBankment. In this case, the symbols are not the ones of the map for the sake of simplicity (the initial embankment lines are in wide red, the generalised embankment lines are in blue, and the roads in grey).

measure	(m_1)	(m_2)	(m_3)	(m_4)
mean	31	35	370	22
median	9	9	39	14
% <5	33	32	15	23
% >50	14	15	42	10

Table 2: Line similarity measure after translation of the initial line.

value of the average distance, for instance, is 13.9 m. Only 2% of the lines are strongly displaced, but even this strong displacement (around 100 m for the average distance value) is acceptable to avoid symbol overlap.

To verify if (C_3), the most important constraint, is preserved, we searched for the number of embankment lines that are still too close to a road or river after the displacement. 80 lines out of the 3,000 are still too close to a road or a river after processing the complete dataset (2.5%). We tried to restart the displacement of these lines with a much larger limit of iterations, and with this large number of iterations, the displacement is good, and the lines are not too close to the roads anymore. We kept the maximum number of iterations small (100) to limit processing time, and because most of the cases do not require more iterations to achieve a good result. But it appears that some cases require a larger maximum number of iterations, and a two-pass process could be developed to improve ReBankment: (1) process all lines with 100 as a maximum number of iterations; (2) process the few remaining lines with 500 for the maximum number of iterations.

Regarding (C_4), we found only one case of close embankment lines that are too close to each other after the displacement. After re-processing the lines with more iterations, the problem is solved, so it is once again a problem caused by our number of iterations kept rather small to limit processing time.

5 Conclusion and Future Work

To conclude, this paper propose the ReBankment algorithm to displace embankment lines from roads and rivers, while preserving their shape. It is dedicated to the generalisation of 1:25k topographic maps. Although it is based on the

classical principles of a geometrical least squares adjustment introduced in map generalisation by Harrie and Sester, the contribution of this research is the specialization of these principles for this problem of elastic displacement. The paper also proposes different solutions to deal with the complex cases occuring with our use case data.

To go further, we think that the iterative nature of the algorithm can be beneficial to adapt it to a multi-scale displacement for multi-scale target maps, instead of just the 1:25k topographic map targeted in this paper. We also plan to compare our algorithm to the Elastic Beams that were previously used at IGN to displace embankment lines. We do not have access to an implementation of this algorithm right now, but a port of the algorithm in our open source platform CartAGen [33] is scheduled. Finally, we plan to adapt the ReBankment algorithm to other problems of line displacement, e.g. cycleways along roads, or ditches [39].

References

[1] Lars E. Harrie. The Constraint Method for Solving Spatial Conflicts in Cartographic Generalization. *Cartography and Geographic Information Science*, 26(1):55–69, January 1999.

[2] Monika Sester. Generalization Based on Least Squares Adjustment. *International Archives of Photogammetry and Remote Sensing*, XXXIII(B4), 2000.

[3] Kurt E. Brassel and Robert Weibel. A review and conceptual framework of automated map generalization. *International Journal of Geographical Information Systems*, 2(3):229–244, January 1988.

[4] Bradford G. Nickerson. Automated Cartographic Generalization for Linear Features. *Cartographica*, 25(3):15–66, 1988.

[5] Dirk Burghardt and S. Meier. Cartographic Displacement using the Snakes Concept. In W. Foerstner and L. Pluemer, editors, *Semantic Modelling for the Acquisition of Topographic Information from Images and Maps*. Birkhauser, Basel, Swiss, 1997.

[6] Mats Bader and Mathieu Barrault. Improving Snakes for Linear Feature Displacement in Cartographic Generalization. In *Proceedings of the 5th International Conference on GeoComputation*, Greenwhich, UK, 2000.

[7] Matthias Bader, Mathieu Barrault, and Robert Weibel. Building displacement over a ductile truss. *International Journal of Geographical Information Science*, 19(8):915–936, 2005.

[8] William A. Mackaness. An Algorithm for Conflict Identification and Feature Displacement in Automated Map Generalization. *Cartography and Geographic Information Systems*, 21(4):219–232, January 1994. Publisher: Taylor & Francis _eprint: https://doi.org/10.1559/152304094782540646.

[9] Anne Ruas. A method for building displacement in automated map generalisation. *International Journal of Geographical Information Science*, 12(8):789–803, 1998.

[10] Janne Kovanen and L. Tiina Sarjakoski. Sequential displacement and grouping of point symbols in a mobile context. *Journal of Location Based Services*, 7(2):79–97, June 2013. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/17489725.2013.764024.

[11] Z. Wei, J. He, L. Wang, Y. Wang, and Q. Guo. A Collaborative Displacement Approach for Spatial Conflicts in Urban Building Map Generalization. *IEEE Access*, 6:26918–26929, 2018.

[12] Haipeng Liu, Ling Zhang, Yi Long, and Yi Zheng. Real-Time Displacement of Point Symbols Based on Spatial Distribution Characteristics. *ISPRS International Journal of Geo-Information*, 8(10):426, October 2019.

[13] Tinghua Ai, Xiang Zhang, Qi Zhou, and Min Yang. A vector field model to handle the displacement of multiple conflicts in building generalization. *International Journal of Geographical Information Science*, 29(8):1310–1331, August 2015.

[14] Peter Hojholt. Solving Space Conflicts in Map Generalization: Using a Finite Element Method. *Cartography and Geographic Information Science*, pages 65–74, January 2000.

[15] Yuangang Liu, Qingsheng Guo, Yageng Sun, and Xiaoya Ma. A Combined Approach to Cartographic Displacement for Buildings Based on Skeleton and Improved Elastic Beam Algorithm. *PLOS ONE*, 9(12):e113953+, December 2014.

[16] Yageng Sun, Qingsheng Guo, Yuangang Liu, Xiaoya Ma, and Jie Weng. An Immune Genetic Algorithm to Buildings Displacement in Cartographic Generalization. *Transactions in GIS*, 20(4):585–612, August 2016.

[17] J. Gaffuri. Field deformation in an agent-based generalisation model: the GAEL model. In Florian Probst and Carsten Kessler, editors, *GI-days 2007 - young researches forum*, volume 30 of *IFGI prints*, pages 1–24. September 2007.

[18] Julien Gaffuri. Three reuse example of a generic deformation model in map generalisation. In *24th International Cartographic Conference*. International Cartographic Association, November 2009. event-place: Santiago, Chile.

[19] François Lecordix, Jean-Marc Le Gallic, and Loïc Gondol. Development of a new generalisation flow line for topographic maps. In *Proceedings of 11th ICA Workshop on Generalisation and Multiple Representation*, 2007.

[20] Lars E. Harrie and Tapani Sarjakoski. Simultaneous Graphic Generalization of Vector Data Sets. *Geoinformatica*, 6(3):233–261, 2002.

[21] Lars E. Harrie. Weight-Setting and Quality Assessment in Simultaneous Graphic Generalization. *Cartographic Journal, The*, 40(3):221–233, December 2003.

[22] Monika Sester. Optimization approaches for generalization and data abstraction. *International Journal of Geographical Information Science*, 19(8):871–897, 2005.

[23] Lothar Gruendig, Frank Gielsdorf, and Bernd Aschoff. Merging Different Data Sets Based on Matching and Adjustment Techniques. In *Strategic Integration of Surveying Services*, Hong-Kong, 2007.

[24] S. Dalyot, Tobias Dahinden, M. J. Schulze, J. Boljen, and Monika Sester. Geometrical Adjustment towards the Alignment of Vector Databases. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-4:13–18, 2012.

[25] Guillaume Touya, Adeline Coupé, Jérémie Jollec, Olivier Dorie, and Frank Fuchs. Conflation Optimized by Least Squares to Maintain Geographic Shapes. *ISPRS International Journal of Geo-Information*, 2(3):621–644, July 2013.

[26] Dongliang Peng, Jan-Henrik Haunert, Alexander Wolff, and Christophe Hurter. Morphing Polylines Based on Least-Squares Adjustment. In *Proceedings of 16th ICA Workshop on Generalisation and Multiple Representation*, Dresden, Germany, August 2013.

[27] Thomas C. van Dijk and Jan-Henrik Haunert. Interactive focus maps using least-squares optimization. *International Journal of Geographical Information Science*, pages 1–24, March 2014.

[28] Imran Lokhat and Guillaume Touya. Enhancing building footprints with squaring operations. *Journal of Spatial Information Science*, 13:33–60, December 2016.

[29] Cécile Duchêne, Anne Ruas, and Christophe Cambier. The CartACom model: transforming cartographic features into communicating agents for cartographic generalisation. *International Journal of Geographical Information Science*, 26(9):1533–1562, April 2012.

[30] Guillaume Touya, Justin Berli, Imran Lokhat, and Nicolas Regnauld. Experiments to Distribute and Parallelize Map Generalization Processes. *The Cartographic Journal*, 54(4):322–332, October 2017.

[31] Guillaume Touya. Relevant Space Partitioning for Collaborative Generalisation. In *Proceedings of 12th ICA Workshop on Generalisation and Multiple Representation*, Zurich, Switzerland, 2010.

[32] Max J. Egenhofer, Eliseo Clementini, and Paolino Di Felice. Topological relations between regions with holes. *International Journal of Geographical Information Science*, 8(2):129–142, March 1994.

[33] Guillaume Touya, Imran Lokhat, and Cécile Duchêne. CartAGen: an Open Source Research Platform for Map Generalization. *Proceedings of the ICA*, 2:1–9, July 2019.

[34] Guillaume Touya and Cécile Duchêne. CollaGen: Collaboration between automatic cartographic Generalisation Processes. In Anne Ruas, editor, *Advances in Cartography and GIScience*, volume 1 of *Lecture Notes in Geoinformation and Cartography*, pages 541–558. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[35] Guillaume Touya. Social Welfare to Assess the Global Legibility of a Generalized Map. In Ningchuan Xiao, Mei-Po Kwan, Michael F. Goodchild, and Shashi Shekhar, editors, *Geographic Information Science 7th International Conference, GIScience 2012*, volume 7478 of *Lecture Notes in Computer Science*, pages 198–211. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2012.

[36] Thomas Devogele. Mesure d'exactitude et processus de fusion à l'aide de la distance de Fréchet discrète. *Revue Internationale de Géomatique*, 10(3):359–381, 2000.

[37] C. S. Myers and L. R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected-word recognition. *The Bell System Technical Journal*, 60(7):1389–1409, 1981.

[38] Robert B. McMaster. A Statistical Analysis of Mathematical Measures for Linear Simplification. *The American Cartographer*, 13(2):103–116, January 1986.

[39] Sandro Savino, Massimo Rumor, and Matteo Zanon. Pattern Recognition and Typification of Ditches. In Anne Ruas, editor, *Advances in Cartography and GIScience*, volume 1 of *Lecture Notes in Geoinformation and Cartography*, pages 425–437. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.