Polygon vector map distortion for increasing the readability of one-to-many flow maps

Laëtitia Viau, Arnaud Sallaberry, Nancy Rodriguez, Jean-François Girres, Pascal Poncelet

ARTICLE TEMPLATE

Polygon vector map distortion for increasing the readability of one-to-many flow maps

Laëtitia Viau[a], Arnaud Sallaberry[a,b], Nancy Rodriguez[a], Jean-François Girres[c,b], and Pascal Poncelet[a]

[a] LIRMM - Université de Montpellier - CNRS, France; [b] AMIS - Université Paul-Valéry Montpellier 3, France; [c] ESPACE-DEV, France.

ARTICLE HISTORY
Compiled March 23, 2023

ABSTRACT

Cartographers have long been interested in the representation of various movements such as migration, commercial exchanges and transportation. There are several techniques for visualizing this information; this paper focuses on flow mapping. A flow map is a type of map that shows a set of movements through line symbols connecting an origin to a destination. Each link is associated with a value that corresponds to the volume of the movement. However, once a certain volume of data is reached, the maps quickly become cluttered and can be difficult to read and understand. Moreover, the values of the movements must be correctly represented so as not to induce biased interpretations.

The objective of this paper is to create flow maps in which the flows have highly variable thicknesses so that the associated value is correctly represented. The technique used to create the flow paths does not create crossings between flows. In order to remove any visual clutter, such as overlaps between flows and geographic features, some areas of the map are distorted. In other words, our method of map distortion adapts the polygon vector base map to the flows, the central information of the visualization, and not the other way around.

KEYWORDS
Geovisualization; one-to-many flow map; map distortion; geospatial network; automated cartography

1. Introduction

Cartographers have long been interested in the representation of various movements such as migration, commercial exchanges and transportation. With the emergence of large volumes of data and globalization, and the corresponding increase in the importance of mobility and exchanges, the visualization of flows remains an issue. A traditional approach from the 1800s involves superimposing links representing the flows onto the maritime areas of a base map representing the geographic features (*i.e.* land), with the thickness of the links representing the magnitude of the flows. Unfortunately, such an approach presents cluttering issues due to overlaps between flows and geographic features. An alternative is to draw thick lines to represent the

CONTACT L. Viau. Email: laetitia.viau@lirmm.fr

flows and to use a sequential color scale to represent the associated quantities, but it is difficult for the human eye to perceive the nuances and compare the values (Bertin 1967). Changing the thickness of the flows is much more effective, since the size channel maps quantitative variables much better than the color channel (Munzner 2014).

The best-known 19th century flow maps are those of Charles Joseph Minard (1781-1870), a French civil engineer (Rendgen 2018). His maps are very easy to interpret and read: each flow has a thickness that corresponds to a value, and no flow overlaps with a geographic feature as a result of base map distortion. Given that the goal of these maps is in no way to measure the surface areas of the geographic features and that the geographic entities remain well recognizable, distortion is a good solution for avoiding cluttering issues while using the thickness of the flows to represent their magnitude. In particular, it makes it possible to have thick flows without any overlap between the geographic information and the flows.

In this paper, we focus on one-to-many flow maps that represent the flow from one origin to several destinations on the maritime areas of a base map representing geographic features, i.e. land (see, for instance, Sun (2019) for previous work on the topic). In such maps, representing the magnitude of the flows with their thickness is more effective than with their color. However, it may cause overlaps between flows and geographic features. Geographic information is then lost as some countries are hidden. For example, a large flow passing over a country can mask a smaller flow going towards that country. The objective of this paper is to provide an automated method for generating one-to-many flow maps inspired by those of Minard, i.e. a method that distorts the base map in order to remove overlaps between flows and geographic features, thus preserving all of the advantages of handmade maps while automating their creation.

The contribution of this paper is threefold:

(1) We propose a new approach to one-to-many flow map visualization based on map distortion;
(2) We propose a method for distorting polygon vector base maps in order to remove overlaps between geographic features and flows;
(3) We propose a new method for generating one-to-many flow maps.

In Sec. 2 below, we present related work on flow maps and map distortion. Then, we introduce our flow map distortion method in Sec. 3. In Sec. 4, we propose a new method for creating one-to-many flow maps. Sec. 5 is dedicated to the discussion of the results, and we conclude in Sec. 6.

2. Related work

The method proposed by this paper relates to the fields of visualizing flows on maps and map distortion. To our knowledge, various methods have been proposed in these two fields, but since Minard, they have never been combined. In this section, we review articles on the creation of flow maps, and then we present the applications of map distortion closest to our method.

2.1. *Flow maps*

Geospatial networks are graphs with nodes that can be associated with geographic locations such as a city, a set of geographic coordinates, or a country. The nodes are

connected by links. Drawing these kinds of networks is relevant to mapping flows. Indeed, flows form a graph in which the nodes are the starting and end points of the flows (and sometimes intermediate locations) and the links are the connecting flows (Jenny *et al.* 2017).

The geospatial network visualization survey by Schöttler *et al.* (2021) raises the question of how to integrate geographic information and network data, as well as other complementary information. The main challenge in visualizing geospatial networks is to present the information as accurately as possible while maintaining readability and clarity. The user must be able to correctly interpret the information at first glance. The survey proposes a design space composed of four dimensions: Geography Representation, Network Representation, Composition, and Interactivity. For the first two dimensions, the modes of representation are placed on a scale from explicit to abstract. The Network Representation dimension is composed of two sub-dimensions: the representation of nodes and of links. In this paper, we focus on the representation of geographic information, nodes and links for flow maps.

Charles Joseph Minard implemented visualizations of geospatial networks in his flow maps (Schöttler *et al.* 2021). These flow maps manage to integrate geographic information and network data, as well as other complementary information, without clutter via link aggregation. With regard to the geographic information, the mode of representation is an intermediate one, between explicit and abstract, because the geographic space is distorted.

The flow maps of Tobler (1987) marked the beginning of automatic flow map creation. These maps have the most explicit representation of the links. All the information is present, but the maps can become very cluttered when the number of flows increases because each flow is represented independently of the others. Moreover, according to Bahoken (2022), the representation of exchanges/movements by a straight line is not satisfactory. It is too approximate and does not correspond to the underlying mode of transport (e.g. maritime transport).

In the following years, some automatic methods that overcome the problem of map cluttering by aggregating the links, like Minard did, were proposed. The majority of these methods rely on an explicit representation of geographic information and nodes as well as an aggregation of links. Indeed, the aggregation of links reduces cluttering while representing the desired information.

To reduce the number of edge crossings, Cui *et al.* (2008) proposed one of the first methods suitable for general graphs. The Geometry-Based Edge Bundling method uses a control mesh to enable the edge-gathering process, which involves forcing all the edges to pass through control points defined on the mesh. However, the results obtained present high curvature variation. Another method of edge bundling for general graphs, by Holten and Van Wijk (2009), avoids this problem. Their force-directed method models edges as flexible springs that attract each other. However, this method is slower than the previous one. Debiasi *et al.* (2014) improve on the method proposed by Holten and Van Wijk (2009). Their novel method for the automatic generation of flow maps describes the motion and force of attraction or repulsion between points through the laws of physics. Gansner *et al.* (2011) also address the problem of curvatures created by the control mesh. They propose a *"multilevel agglomerative edge bundling method based on a principled approach of minimizing ink needed to represent edges, with additional constraints on the curvature of the resulting splines."*

Jenny *et al.* (2016) defined design principles for origin-destination flow maps. The findings of a quantitative analysis and the results from user studies in graph drawing led to design principles such as minimizing the overlap between flows, scaling the flow

width to represent quantity, and many others. On the basis of these design principles, a force-directed layout method for producing origin-destination flow maps in which the flows are quadratic Bezier curves was defined (Jenny *et al.* 2017).

Other methods based on hierarchical clustering or a Steiner tree have been proposed. Phan *et al.* (2005) use hierarchical clustering to define a flow map layout. The hierarchical clustering creates a tree that connects a root to a set of destinations. The method minimizes edge crossings and distorts node positions. The relative positions of the nodes are well preserved. In this case, the geographic representation is not explicit due to the distortion, and labels are added to nodes instead of a base map. Buchin *et al.* (2011) use a type of Steiner tree involving logarithmic spirals. It improves flow tracing by avoiding crossings between flows and geographic entities as much as possible. Another method for creating a one-to-many flow map based on a Steiner tree was proposed by Sun (2019). It involves triangulating the space to create a network and using this triangulation to define a Steiner Tree. A series of automatic mapping operations are then performed to improve the lines representing the flows and add color to them according to their value.

Bahoken (2022) explains the importance of the "route effect", which should be integrated into the representation of the flows in order to provide a qualitative improvement to the map. Without knowing the route of the movements, it is obviously not possible to transcribe the exact path taken in reality, but these movements must be evoked. On a global scale, representing the flows along possible maritime routes leads to the "route effect".

2.2. *Map distortion*

It is important to keep in mind that maps are always distorted, regardless of how the globe is transformed into a flat surface. This transformation is called a map projection. In this article, the term distortion does not refer directly to the map projection itself, but to the modification of a map already projected onto a flat surface.

2.2.1. Cartograms

One of the main uses of map distortion is for cartograms. Initially, the term cartogram was used to refer to different types of maps such as statistical maps and choropleth maps (Tobler 2004). As defined in the paper by Nusrat and Kobourov (2016), the term cartogram now refers to maps that *"combine statistical and geographical information in thematic maps, where areas of geographical regions (e.g., countries, states) are scaled in proportion to some statistic (e.g., population, income)"*. For instance, the diffusion-based method for producing density-equalizing maps of Gastner and Newman (2004) is one of the best-known methods for constructing a cartogram. The technique proposed is based on ideas from elementary physics. Gastner *et al.* (2018) improved this technique by speeding up the process.

2.2.2. Fish-eye

In photography, a fisheye lens is a very wide angle lens that shows nearby objects in great detail while showing objects farther away in less detail. This idea was applied to graphs by Sarkar and Brown (1992). In cartography, a fisheye means having a variable-scale map projection. Fisheye views are very useful for focusing on a particular area containing a lot of information while retaining the most distant elements. The area of

interest, called the focus, is enlarged, but the surroundings, called the context, remain visible. This is referred to as a Focus+Context fisheye. The difficulty lies in providing access to all the detailed information in the focus area while preserving the context.

The Focus+Context fisheye view was proposed as a solution for real-time personal navigation by Harrie *et al.* (2002). A user needs both an overview map and more details around their position. The method uses a scalable map that changes in real time so that the user's position is always in the large-scale part. However, the results deteriorate with increasing magnification intensity. Carpendale *et al.* (2004) present a set of improvements for high magnification by proposing new distortion functions.

The main problems with Focus+Context fisheye views is that the entire area of the map is distorted and the calculation cost is very high. Furthermore, the density of information is too large in the corners of the map. Takahashi (2008) improved this type of fisheye view by introducing the concept of Focus+Glue+Context. A glue area is included between the Focus and Context areas, and distortions in the Focus and Context are completely removed. The Focus area is large-scale in order to display the details of the map, while the Context area is small-scale. The Glue area links the Focus to the Context. Yamamoto *et al.* (2009) proposed a method for generating high-quality maps based on this concept, with Focus+Glue+Context views that can be used in web map services given their generation speed. Similarly, Haunert and Sering (2011) try to improve the fisheye views in cartography by arguing that distortion should preferably take place in areas where the information density is lower. Thus, no predefined mapping function is applied. The method is based on an optimization approach to minimize distortions and respects a set of constraints and objectives.

The objective of our Flow Map Distortion method is very similar to that of the fisheye: to make a set of magnifications in particular areas while keeping the whole map visible.

2.2.3. Cartographic generalization and schematization

Ruas (2008) defines map generalization as *"the process that simplifies the representation of geographical data to produce a map at a certain scale with a defined and readable legend. To be readable at a smaller scale, some objects are removed; others are enlarged, aggregated and displaced one to another, and all objects are simplified."* The result of a cartographic generalization process can be seen as a distorted but more readable map. Over the last few decades, progress in automated cartographic generalization has allowed geographic data on a map to be automatically adapted according to specific constraints (Mackaness *et al.* 2007). For example, specific algorithms were developed to adapt the representation of mountain roads (Mustière 1998) and land use (van Oosterom 1995). Agent-based approaches (Ruas 1998, Gaffuri 2007, Touya and Duchêne 2011) and least-square adjustment methods (Harrie 1999, Sester 2005) were also proposed to resolve conflicts in cartographic generalization.

In this context, cartographic generalization can be seen as a map distortion process since it modifies the initial positions and shapes of geographic features in order to facilitate their visualization on the map.

Finally, map distortion techniques can be illustrated by the application of map schematization approaches (Dykes *et al.* 2010), used to facilitate the visualization of specific features such as complex networks. The automatic generation of metro maps (Stott *et al.* 2011) is a classical illustration of map distortion based on cartographic schematization.

3. Flow map distortion (FMD)

Visualizing flows over maritime areas while using width to represent a quantity generates overlaps between flows and geographic features. We propose a method for removing these overlaps from an existing one-to-many flow map without modifying the width of the flows. Figure 1 illustrates the steps involved: a first distortion, then the drawing of curves, and finally a second distortion that completely removes any overlap.

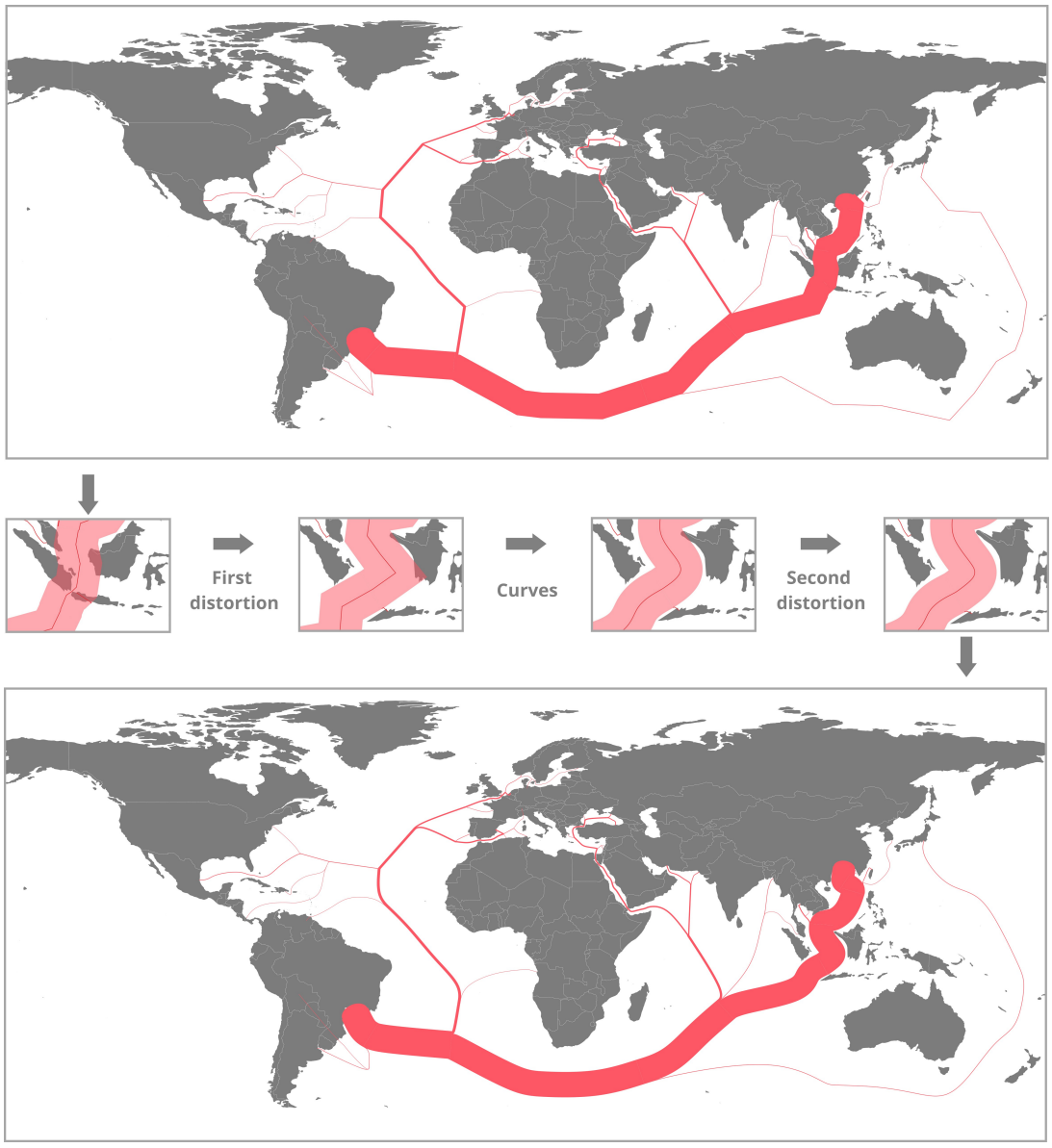


Figure 1. An Overview of the main steps of the FMD method.

3.1. *First distortion*

The method proposed in this article involves carrying out a set of magnifications in the areas where the flows pass in order to leave them enough space that they do not overlap the geographic features. To perform the magnifications, we use the formula proposed by Kadmon and Shlomi (1978). Let us consider the focus point with coordinates (x_f, y_f) and a point P with coordinates (x, y). D is the distance between P and the focus point. (x', y'), the new coordinates of P, are computed as follows:

$$\begin{aligned} x' &= x + \frac{A(x - x_f)}{1 + C \cdot D^2} \\ y' &= y + \frac{A(y - y_f)}{1 + C \cdot D^2} \end{aligned} \tag{1}$$

where A is the level of magnification and C the radial rate of change of this variable. This means that the higher the C, the more the magnification will be localized close to the focus. On the contrary, the lower the value of C, the more the magnification will be spread over the entire map. In our case, the magnification has to be localized, so we set $C = 1$.

Since our method is dedicated to one-to-many flow maps, the flow can be considered as a weighted tree in which the root is the origin, the leaves are the destinations, and the internal nodes are the control points of the original map (on the left in Figure 2). In this case, the weights of the edges are the widths of the flows. To apply magnifications, we first divide the tree into paths such that each internal node has a degree of 2 in the tree (on the right in Figure 2). We then give these paths a weight corresponding to the weight of one of their edges (the weights of all of a path's edges being equal). We call the ordered sets of nodes of each path sequences. Les us consider the set S of the sequences of the tree such that $S = [S_1, S_2, ..., S_s]$. $\forall S_i$ with $1 \leq i \leq s$, w_i is the weight of the sequence.

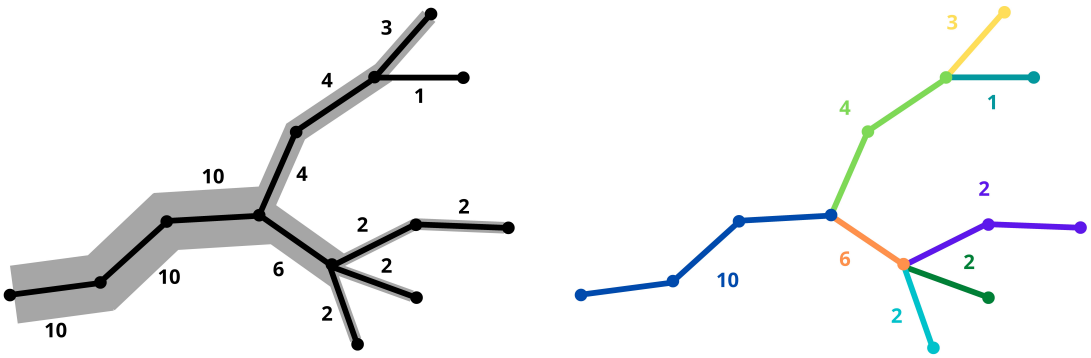


Figure 2. Tree (left) and the sequences of the tree (right). Each color corresponds to a sequence (8 total) and the numbers represent their weight.

The method involves processing this set of sequences in decreasing order of their weight. For each of these sequences, we have a set of nodes. If there are overlaps between the sequence and a geographic feature, then the node closest to a vertex of the base map is sought. This node then becomes a focus point and we apply the formula (see Algorithm 1) to all the vertices forming the shapes corresponding to the geographic features and to all the sequence nodes. This process is repeated until there

is no overlap (see Algorithm 2).

Algorithm 1 Create a new magnification

Require: P a set of coordinates

function DISTORT(P, $focus$, A, $C = 1$)
  for each $p \in P$ **do**
    $D \leftarrow \sqrt{(x_p - x_{focus})^2 + (y_p - y_{focus})^2}$
    $x'_p \leftarrow x_p + \frac{A \times (x_p - x_{focus})}{1 + C \times D^2}$
    $y'_k \leftarrow y_p + \frac{A \times (y_p - y_{focus})}{1 + C \times D^2}$
  end for
  return P'
end function

Algorithm 2 First distortion

Require: S the set of sequences forming the flow, K the set of points forming the shapes of the geographical features, $\epsilon \in]0, 1[$ a damping factor

Ensure: S the set of sequences with updated positions, K the set of points forming the shapes of the geographical features with updated positions

for each $S_i \in S$ **do**
  $hasOverlap \leftarrow True$
  $r_i \leftarrow \frac{w_i}{2}$
  while $hasOverlap$ **do**
    $distMin \leftarrow \{\min distance(u, k) | \forall u \in S_i, \forall k \in K\}$
    if $distMin > r_i$ **then**
      $hasOverlap \leftarrow False$
    end if
    $focus \leftarrow \{u \in S_i | \min distance(u, k)\ \forall u \in S_i,\ \forall k \in K\}$
    $A_{min} \leftarrow$ solution of equation (6)
    $A \leftarrow A_{min} \times \epsilon$
    $S \leftarrow$ DISTORT($S, focus, A$)
    $K \leftarrow$ DISTORT($K, focus, A$)
  end while
end for

Let D' be the distance between the focus point and the nearest vertex of a geographic feature after transformation and R be the weight of the sequence divided by 2 (*i.e.* the radius of the flow width). To eliminate all overlaps, we must define A_{min}, the minimum value of A so that $D' \geq R$.

$$\begin{aligned} D' \geq R &\Leftrightarrow \sqrt{\left[\left(x + \frac{A_{min}(x - x_f)}{1 + C \cdot D^2}\right) - x_f\right]^2 + \left[\left(y + \frac{A_{min}(y - y_f)}{1 + C \cdot D^2}\right) - y_f\right]^2} \geq R \\ &\Leftrightarrow \left[\left(x + \frac{A_{min}(x - x_f)}{1 + C \cdot D^2}\right) - x_f\right]^2 + \left[\left(y + \frac{A_{min}(y - y_f)}{1 + C \cdot D^2}\right) - y_f\right]^2 \geq R^2 \end{aligned} \tag{2}$$

By isolating the variable A_{min} and expanding the formula we get:

$$\left[\left(x + \frac{A_{min}(x - x_f)}{1 + C \cdot D^2}\right) - x_f\right]^2 = \left(\frac{x - x_f}{1 + C \cdot D^2}\right)^2 A_{min}^2 + \left(\frac{2(x - x_f)^2}{1 + C \cdot D}\right) A_{min} + (x - x_f)^2 \tag{3}$$

$$\left[\left(y + \frac{A_{min}(y - y_f)}{1 + C \cdot D^2}\right) - y_f\right]^2 = \left(\frac{y - y_f}{1 + C \cdot D^2}\right)^2 A_{min}^2 + \left(\frac{2(y - y_f)^2}{1 + C \cdot D}\right) A_{min} + (y - y_f)^2 \tag{4}$$

By adding (3) and (4) we can then obtain a single second-degree polynomial of the form $\alpha A_{min}^2 + \beta A_{min} + \gamma$:

$$\begin{aligned} \alpha &= \left(\frac{x - x_f}{1 + C \cdot D^2}\right)^2 + \left(\frac{y - y_f}{1 + C \cdot D^2}\right)^2 \\ \beta &= 2\frac{(x - x_f)^2 + (y - y_f)^2}{1 + C \cdot D} \\ \gamma &= (x - x_f)^2 + (y - y_f)^2 \end{aligned} \tag{5}$$

All that remains is to solve the equation, the result of which corresponds to the minimum value of A that removes any overlap:

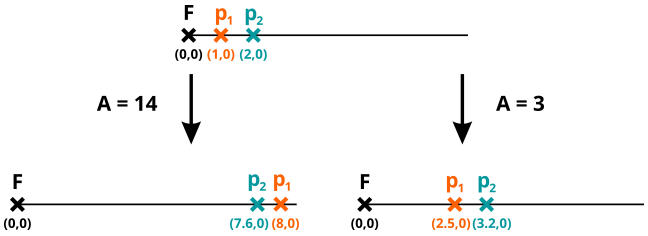
$$\alpha A_{min}^2 + \beta A_{min} + \gamma - R^2 = 0 \tag{6}$$

Initially, very thick sequences can pass through very narrow zones (see Figure 3b). Removing the superpositions with $A = A_{min}$ in a single iteration can create crossings in the shapes of the geographic features (see Figures 3a and 3c). Decreasing the value of A by a damping factor $\epsilon \in]0, 1[$ such that $A = A_{min} \times \epsilon$ moderates the distortion and avoids creating non-simple polygons[1]. To avoid any crossings, we propose carrying out the magnifications recursively (Figure 3d).

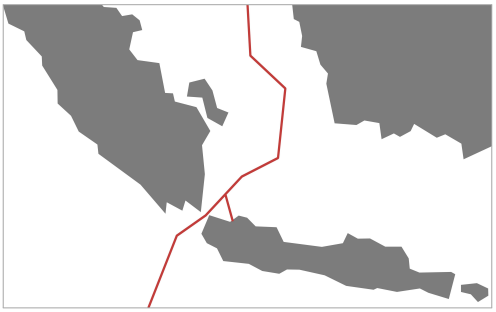
Nodes are not processed in sequential order, and the same node can be processed several times. This distortion step only ends once the overlaps have been completely removed, but it should not be forgotten that the level of magnification is reduced. We found empirically that this prevented the creation of crossings in the geographic features. In addition, the sequential order would always cause the geographic features to move in the same direction as the flow path, which created more distortion than necessary. Another advantage of processing the node closest to a geographic feature at each iteration is that a node is never processed where a distortion is not necessary.

Let l be the number of vertices forming the shape corresponding to a geographic feature, m the size of the tree, and n the number of sequences. The time complexity of checking if there are overlaps is $O(c.l.m)$, where c is the number of iterations. For setting the focus, the time complexity is also $O(c.l.m)$, and for the distortion it is $O(c.m.n + c.l.n)$. Since the number of sequences is less than the size of the tree, $n.l$ is less than $m.n$ and the time complexity of the first distortion is $O(c(l.m+m.n))$. If there are many sequences in the tree T, *i.e.* n tends towards m, the size of the sequences decreases and tends towards 1. The complexity then approaches $O(c(l.m + m^2))$.

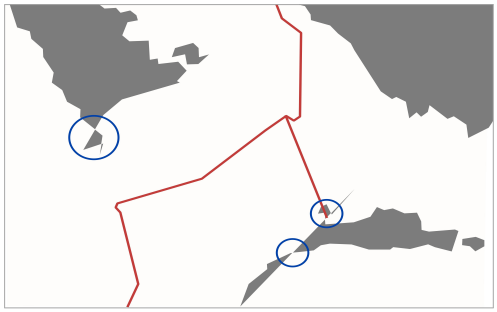
[1]We set $\epsilon = 0.2$. Empirically, this value was the most effective in moderating the distortion and avoiding the creation of non-simple polygons.



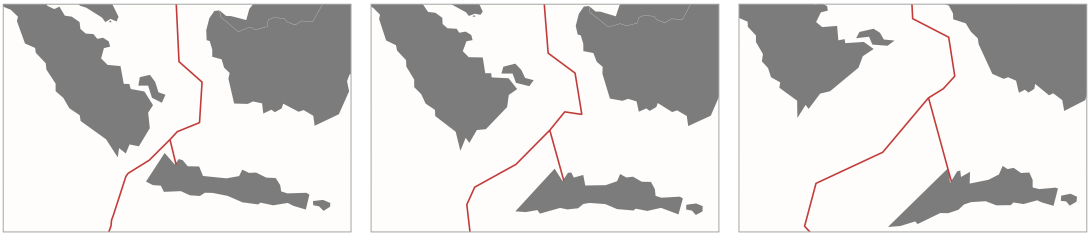
(a) Example of repositioning p_1 and p_2 by distortion with $A = 14$ and $A = 3$, where F is the focus point. In the first case, the value of A is too high: the two points do not keep their positions relative to one another.



(b) Initial geographical features: there is no distortion.



(c) Example of distortion in one iteration with $A = A_{min}$: crossings (circled in blue) are created in the geographic features. The polygons are not therefore simple.



(d) Example of distortion carried out recursively: for each magnification, we set $A = A_{min} \times 0.2$. The polygons are simple and there are no crossings. In this example, not all iterations are illustrated.

Figure 3. Comparison of the results of distortion according to the value of A and the number of iterations.

At the end of this first step, the overlaps have been removed while maintaining recognizable and consistent geographic features. We then move on to creating the flow curves.

3.2. *Curves*

In this second step, we transform each sequence of the tree into a curve. This step is performed just after the first distortion because creating the curves involves changes to the tree. Modifying the tree without having created space could cause crossings between the tree and the geographic features.

For this step, we tested and compared different curves. Our method obtained the

best results with the Catmull-Rom spline and the basis spline (B-spline).

The Catmull-Rom spline uses the specified control points and a parameter α. Its behavior depends significantly on the choice of the value for α. If $\alpha = 0$, some self-intersections may occur, and if $\alpha = 1$, the curve moves too far from the control polygon (Yuksel *et al.* 2009). In our case, $\alpha = 0.5$ seems to be the most appropriate. The main advantage of this curve compared to the B-spline is that it passes through the points of the control polygon, *i*.e. the flow path. This reduces the chances of creating new overlaps since the curve is drawn very close to the original path. However, the B-spline has the advantage of smoothing the path more than Catmull-Rom.

The B-spline is a basis cubic spline straightened with control points. Holten (2006) proposed using it in hierarchical edge bundling to clarify the edges. The spline is straightened according to the β parameter of the curve. $\beta = 1$ is the most suitable for our needs (see Figure 4). We therefore opted for this type of curve, which smooths the paths without superimposing them on the geographic features.

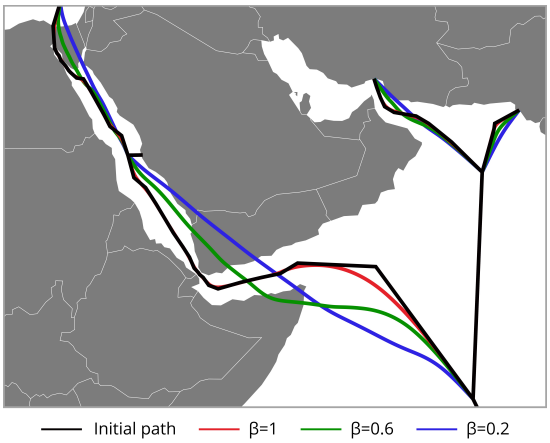


Figure 4. Comparison of the rendering of the B-spline curve with different values of β.

3.3. *Second distortion*

During our tests, we noticed that drawing the curves could create slight overlaps in some cases. A second distortion is necessary to make sure that all overlaps are removed. Using the formula (1) and the equation (6) presented above, a second distortion is carried out (see Algorithm 3).

The algorithm is almost the same as for the first distortion except that the focus points are not created from the nodes of the sequences but from the nodes of the curves of each sequence. Points along the curve are processed in sequential order. Indeed, at this stage, few changes are necessary and one iteration on a point is enough to completely remove any overlay.

Let l be the number of vertices forming the shapes corresponding to the geographic features and m be the number of nodes in all the curves. The time complexity of checking if there are overlaps is $O(l.m)$. For the distortion, it is $O(l.m+m^2)$, and thus the time complexity of the second distortion is also $O(l.m + m^2)$.

At the end of these 3 steps, we obtained flow maps without any overlap. The only initial condition was that no edge of the tree should cross a geographic feature. Various

Algorithm 3 Second distortion

Require: P the set of curves forming the flow, K the set of points forming the shapes of the geographical features

Ensure: P the set of curves with updated positions, K the set of points forming the shapes of the geographical features with updated positions

for each $P_i \in P$ **do**
  for each $p \in P_i$ **do**
    $dist \leftarrow \min distance(p, k), \forall k \in K$
    $r_i \leftarrow \frac{w_i}{2}$
    if $dist < r_i$ **then**
      $focus \leftarrow p$
      $A_{min} \leftarrow$ solution of equation (6)
      $A \leftarrow A_{min}$
      $P \leftarrow \text{DISTORT}(P, focus, A)$
      $K \leftarrow \text{DISTORT}(K, focus, A)$
    end if
  end for
end for

methods can be used to create the flows of a flow map. In the next section, we propose one that limits the overlapping surfaces in order to reduce the number of magnifications that must be carried out during the distortion of the geographic features.

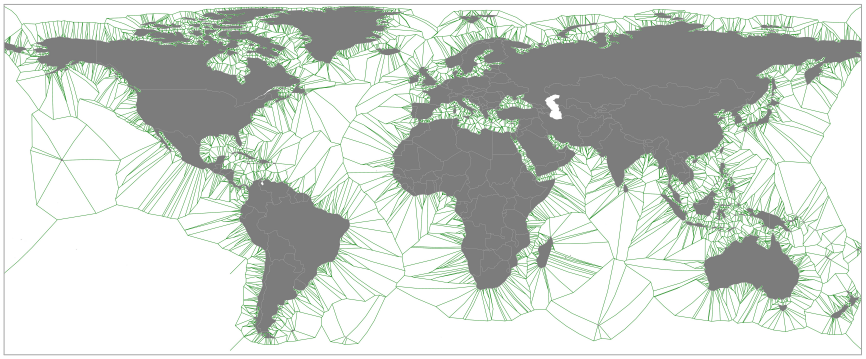
4. Initial one-to-many flow map

In this section, we propose a new method for creating one-to-many flow maps. This method makes it possible to create flows with fluid lines and is the most suitable for applying the FMD method. The flows are drawn along the median axes of the geographic features, which reduces the overlaps between them and the features. Reducing the number of overlaps limits the number of iterations in the FMD process and the distortion of the features.

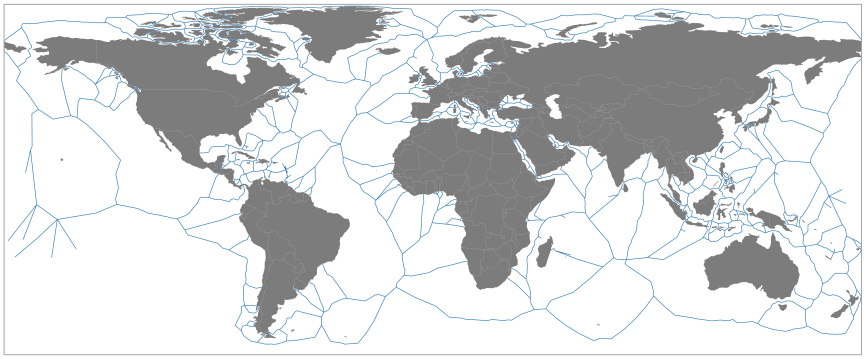
We will now present the steps for creating a one-to-many flow map: (1) the creation of a graph from a skeleton (see Figures 5a and 5b), (2) the definition of a tree representing the flows (see Figure 5c), (3) a simplification of this tree, and (4) the drawing of the flows (see Figure 5d).

4.1. *Initial graph creation*

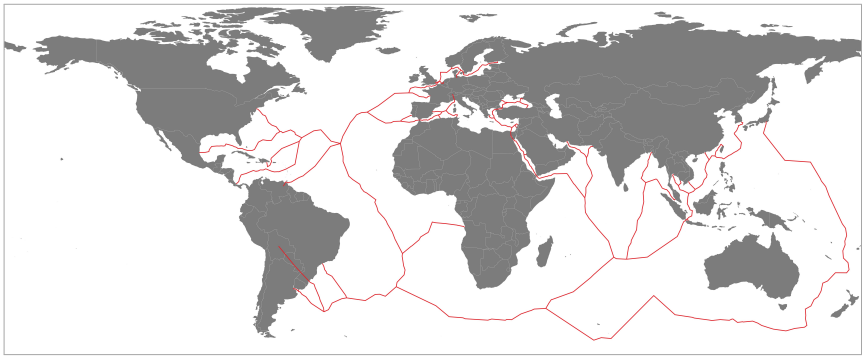
The first step of our method involves creating a graph on top of the base map with edges that lie as far as possible from the geographic features. This graph must contain paths linking the geographic entities of the dataset. In our case, we consider flows between countries, so there must be a path between each pair of countries lying on a coast. This graph will then be used to rout the flows (see next section). Our method is based on a straight skeleton (see Figure 5a). Introduced by Aichholzer *et al.* (1995) and then generalized by Aichholzer and Aurenhammer (1996), straight skeletons can be considered as connected graphs whose nodes include the vertices of the polygons (in our case the vertices forming the coasts of the geographic features) and whose edges lie close to the median axes of the features. Unlike the median axes, which



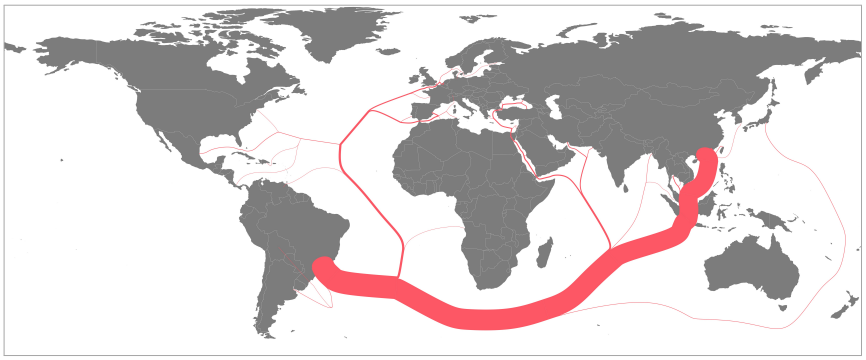
(a) Skeleton



(b) Graph



(c) Tree



(d) Flow map

Figure 5. Creation of a one-to-many flow map.

can contain parabolic curves, straight skeletons are composed exclusively of straight segments. Felkel and Obdrzalek (1998) proposed a method applied to simple polygons and Eppstein and Erickson (1999) a method that builds an improved straight skeleton.

In our implementation, we used a library[2] that gives us the skeleton shown in Figure 5a. We built a straight skeleton out of a polygon with "holes" that correspond to the inner faces of the polygons forming the geographic features (*i.e.* the land).

Since the coastal countries include multiple vertices, there are several paths leading to each country in the skeleton. To simplify the latter, we select one vertex per coastal country (in our implementation, the vertex closest to the biggest port of the country) and iteratively remove non-selected vertices of degree one (and the corresponding edges). This gives us a graph G, shown in Figure 5b.

4.2. *Tree definition*

The second step of our method involves mapping the flows to the graph G in order to create the flow paths. The input flows consist of an origin (the starting country), a set of targets (the destinations), and a quantitative value for each origin/destination pair. From the graph G, we calculate the shortest path between the origin and each destination using Dijktra's algorithm. For these calculations, there are 3 possibilities for defining the weight w of the edges:

- $w = 1$; there is no weighting.
- $w = d$, with d the distance between the two nodes. Long edges are penalized.
- $w = d + (p \times (180 - a))$; the smaller the value a of the angle (in degrees) with the previous edge, the higher the weight w. Small angles are thus penalized.

The results presented are based on the last definition, with $p = 5$. We will discuss this parameter in Section 5.

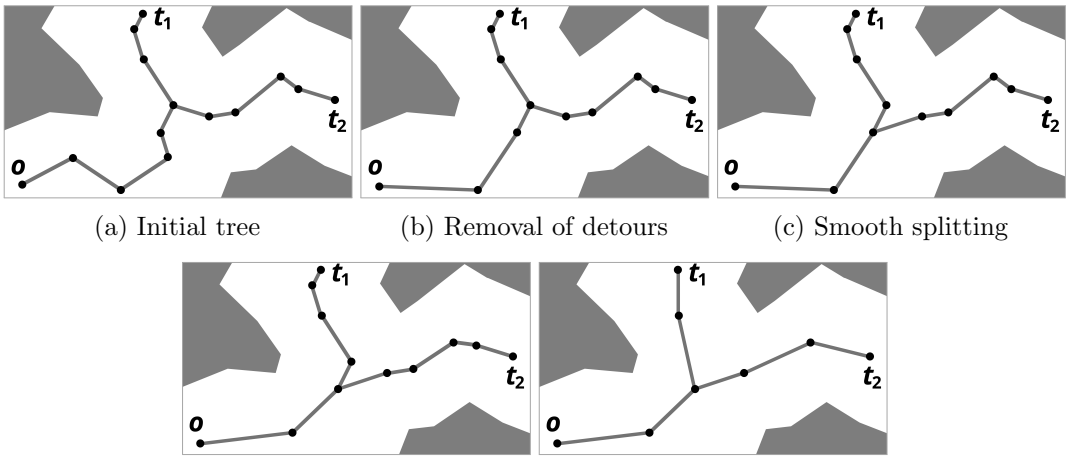
We then create a graph $T = (V, E)$ via the union of the shortest paths (see Figure 5c). T is connected, acyclic and edge directed; it is thus a directed tree in which the root represents the origin of the flows and the leaves represent the targets of the flows.

4.3. *Simplification of the tree*

When we visualize the tree computed in the previous step, we can observe many unnecessary bends along the paths (see Figure 6a). This step consists in improving the appearance of the flows by smoothing the paths defined by the tree. There is a very wide variety of solutions for simplifying the tree. What is essential is to not create any crossings between the nodes and the geographic features. We propose making a series of successive improvements (see Figure 6). We will describe each of the steps for achieving a better appearance of the flows in detail (see the final result in Figure 6e).

First, we remove the nodes that form detours (from Figure 6a to Figure 6b). Let us consider four successive nodes v_0, v_1, v_2, v_3 and three edges (v_0, v_1), (v_1, v_2) and (v_2, v_3) (see Figure 7a). We compute the polar coordinates of the nodes v_1, v_2 and v_3 from the node that precedes each of them on the interval $[0, 2\pi[$. For example, for v_2 we will use v_1. Let us note the polar coordinates as (r_1, θ_1), (r_2, θ_2) and (r_3, θ_3) and consider 4 intervals: $[0, \frac{\pi}{2}[$, $[\frac{\pi}{2}, \pi[$, $[\pi, \frac{3\pi}{2}[$ and $[\frac{3\pi}{2}, 2\pi[$. If θ_1 and θ_3 are in the same interval and θ_2 in a different one, we remove the node v_1 (which gives us the edge

[2]The Computational Geometry Algorithms Library (CGAL): `https://www.cgal.org` (accessed: 2022-09)

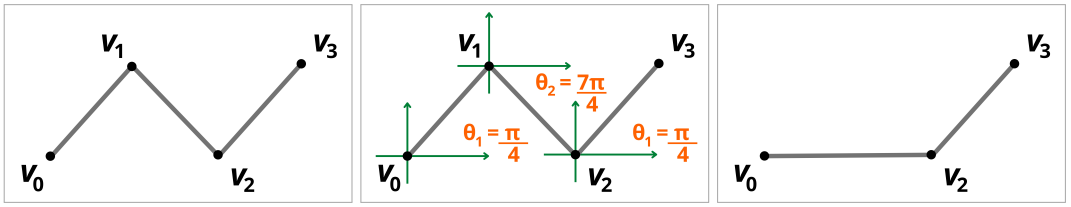


(a) Initial tree (b) Removal of detours (c) Smooth splitting

(d) Repositioning of nodes (e) Removal of the nodes that are not significant

Figure 6. Simplification steps. o is the root of the tree (the origin of the flows) and t_1 and t_2 are the leaves (the targets of the flows).

(v_0, v_2)). This is the case in Figure 7b : θ_1 and θ_3 are in the interval $[0, \frac{3\pi}{2}[$ while θ_2 is in the interval $[\frac{3\pi}{2}, 2\pi[$. Thus, v_1 is removed (see Figure 7c).



(a) Initial nodes and edges. There are four successives nodes : v_0, v_1, v_2, v_3, and three edges: (v_0, v_1), (v_1, v_2) and (v_2, v_3).

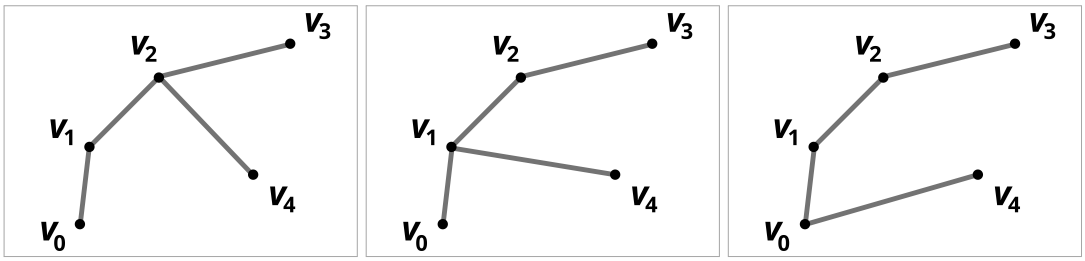
(b) Definition of the angular coordinate of the nodes v_1, v_2 and v_3 based on the preceding node in the interval $[0, 2\pi[$. $\theta_1 \in [0, \frac{\pi}{2}[$, $\theta_2 \in [\frac{3\pi}{2}, 2\pi[$ and $\theta_3 \in [0, \frac{\pi}{2}[$.

(c) The node v_1 is removed because θ_1 and θ_3 are in the same interval and θ_2 in a different one.

Figure 7. Procedure for removing detours.

The second improvement consists in smoothing the paths when the flows split, *i.e.* when the nodes have a degree greater than 2. The idea is to produce a smooth path by avoiding bends forming an angle of less than $\frac{\pi}{2}$ (from Figure 6b to Figure 6c). Let us consider a split composed of four nodes v_1, v_2, v_3, v_4 and three edges (v_1, v_2), (v_2, v_3) and (v_2, v_4) (see Figure 8a). If the polyline made by the path v_1, v_2, v_4 forms an angle of less than $\frac{\pi}{2}$ (this is the case in the Figure 8a), we remove the edge (v_2, v_4) and add the edge (v_1, v_4) (see Figure 8b). If the angle produced by the parent v_0 of v_1, v_1, and v_4 is less than $\frac{\pi}{2}$, we delete (v_1, v_4) and add an edge between v_0 and v_4 (see Figure 8c). We repeat this process as long as there is a parent node and until the angle is greater than $\frac{\pi}{2}$.

To eliminate the effects of "half-turns" on the tree, we reposition the 2-degree nodes for which the angle formed is below a threshold (from Figure 6c to Figure 6d). Let us consider three successive nodes v_1, v_2, and v_3 and two edges (v_1, v_2) and (v_2, v_3). If the



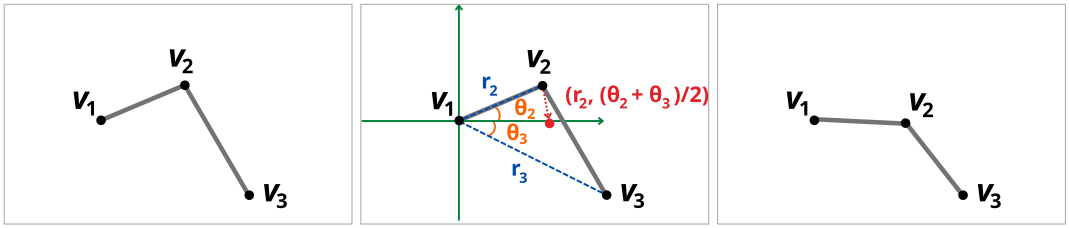
(a) Initial flow paths : the path v_1, v_2, v_4 forms an angle of less than $\frac{\pi}{2}$.

(b) The edge (v_2, v_4) is removed and replaced by (v_1, v_4). The new path v_0, v_1, v_4 forms an angle of less than $\frac{\pi}{2}$.

(c) The edge (v_1, v_4) is removed and replaced by (v_0, v_4).

Figure 8. Path smoothing procedure when flows split.

polyline made by the path v_1, v_2, v_3 forms an angle of less than 0.6π (see Figure 9a), we reposition the node v_2. For this, we first compute the polar coordinates of v_2 and v_3 from v_1 on the interval $]-\pi, \pi]$. The polar coordinates of v_2 and v_3 are respectively noted as (r_2, θ_2) and (r_3, θ_3). Then, we redefine the coordinates of v_2 as $(r_2, \frac{\theta_2+\theta_3}{2})$ (see Figure 9b). All that remains is to convert these polar coordinates into cartesian coordinates. The node v_2 is then repositioned (see Figure 9c).



(a) Initial position of nodes v_1, v_2, v_3. The path v_1, v_2, v_3 forms an angle of less than 0.6π: the node v_2 will be repositioned.

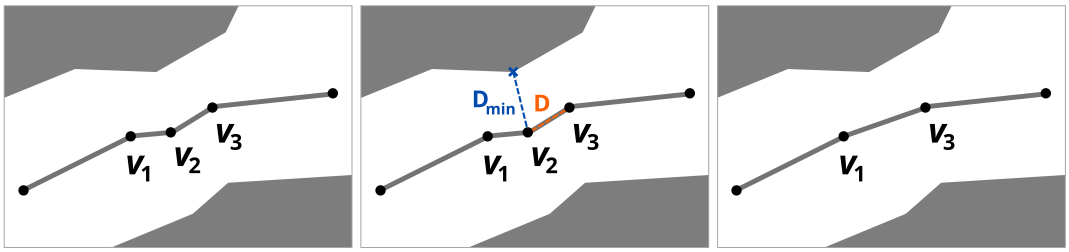
(b) Definition of the polar coordinates of v_2 and v_3 on the basis of v_1 on the interval $]-\pi, \pi]$. The coordinates are noted as (r_2, θ_2) and (r_3, θ_3). The new position of the node v_2 is shown in red.

(c) New path formed by the nodes v_1, v_2 and v_3.

Figure 9. Procedure for eliminating "half-turns" effects.

Finally, we delete the 2-degree nodes that are not significant (from Figure 6d to Figure 6e)[3]. Let us consider three successive nodes v_1, v_2, and v_3 and two edges (v_1, v_2) and (v_2, v_3) (see Figure 10a). D_1 is the distance between v_1 and v_2, D_2 the distance between v_2 and v_3, and D_{min} the distance between v_2 and the nearest vertex of a geographic polygon. Let us note the larger of D_1 and D_2 as D (see Figure 10b). If $D < D_{min}$, we remove the node v_2 (see Figure 10c).

[3]When using the FMD method, we recommend performing this last step after the first distortion, just before drawing the curves, because these nodes, which are not significant in terms of appearance, can be significant for the first distortion step. On the other hand, if the objective is just to create a one-to-many flow map without distortion, then this last step can take place in the order described before drawing the curves and thicknesses of the flows like in Figure 5d.



(a) Initial path: three successive nodes v_1, v_2 and v_3 are processed.

(b) D, shown in orange, is the larger edge length out of (v_1, v_2) and (v_2, v_3). D_{min}, in blue, is the distance between v_2 and the nearest vertex of a geographic polygon. Since $D < D_{min}$, v_2 is removed.

(c) Final path without the node v_2, which was not significant.

Figure 10. Procedure for removing nodes that are not significant.

4.4. *Drawing the flows*

After the simplification step, the flows can be drawn using the tree. We draw each path, the nodes of which have a degree of at most 2, with a B-spline and add a thickness corresponding to the sum of the quantities of the flows that pass through it. The result is shown in Figure 5d.

In this section, we have provided a new solution for creating a one-to-many flow map. Using this method is not mandatory but highly recommended. The flow paths follow the median axes between the geographic features, which minimizes the number and size of overlapping surfaces and therefore the magnifications to be carried out. This method can be used alone, without FMD, just to create a one-to-many flow map.

5. Discussion

In this section, we discuss the results obtained on three different datasets. First, we present the characteristics of the datasets and the computation times. Then, we analyze the maps created by the one-to-many flow map method and the FMD method and the impact of the parameter p. Finally, we report expert feedback on the strengths and limitations of our method.

5.1. *Datasets*

The first dataset concerns tea exports from China (CHN), the second soybean exports from the United States (USA), and the third vanilla exports from Madagascar (MDG). The data is available in FAOSTAT [4]. For each input dataset, the number of destinations (*i.e.* countries in these examples) is reported in Table 1. This table also shows the minimum, maximum, mean, median, and standard deviation of the values associated with the flows.

[4] `https://www.fao.org/faostat/en/#data/TM` (accessed: 2022-11)

Dataset	Nb of dest.	Min	Max	Median	Mean	Std
CHN	115	1	74260	280	3189.57	8134.21
USA	74	1	22575972	23183.5	707951.15	2717691.15
MDG	23	1	625	9	64.35	145.24

Table 1. Input dataset characteristics

5.2. *Computation time*

The initial flow maps for each dataset were created as presented in Section 4. For this, we initially had 6996 vertices forming the polygons of the geographic features. The skeleton created from these polygons had 14002 vertices, 42006 edges and 6890 faces. The derived graph had 3697 nodes and 3808 edges. The creation of the skeleton and of the graph took about 5 minutes (deleting unnecessary nodes and edges). Regardless of the dataset, the time to create the initial flow map from the graph and the data source was about 2 minutes. All processing was done on a laptop with a 2.3 GHz quad-core CPU and 32GB memory. The process was implemented in Python, except for the creation of the skeleton, which was done using the C++ library mentioned in Section 4.

Then, we executed the FMD method; the times[5] are indicated in Table 2. We used the same computer but the implementation was carried out in JavaScript.

Dataset	# of nodes	FMD (time)
CHN	1243	51 seconds
USA	1108	36 seconds
MDG	619	18 seconds

Table 2. Tree characteristics and computation times of the FMD method

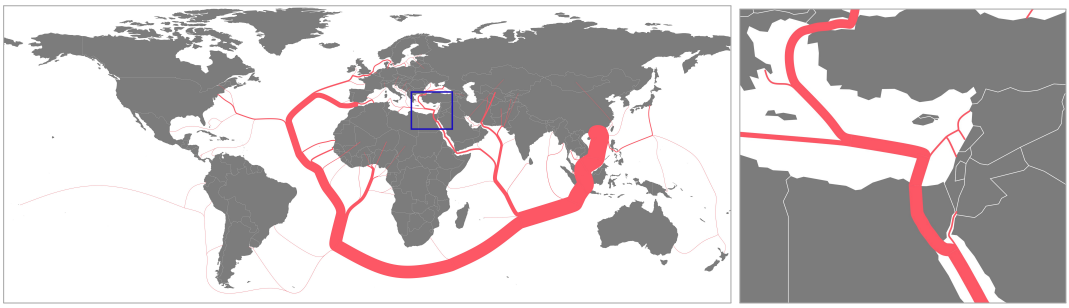
5.3. *Results*

Figures 11, 12, and 13 show the results obtained for each of the 3 datasets with and without distortion. The results without distortion were obtained by the method proposed in Section 4. The results with distortion were obtained using the FMD method of Section 3 on the results without distortion.

In some areas of the initial map (*i.e.* the map without distortion), a thick flow can hide small flows and geographic features such as islands. For example, Figure 14 compares the Mediterranean area with and without distortion. Without the distortion, the island of Crete, south of Greece, is not visible at all, while Sicily and Sardinia are partly hidden. With the distortion, the three islands are all fully visible and well recognizable. Likewise, several smaller flows to various countries in North Africa, such as Tunisia and Algeria, are barely or not at all visible. After the distortion, we can distinguish these flows much better.

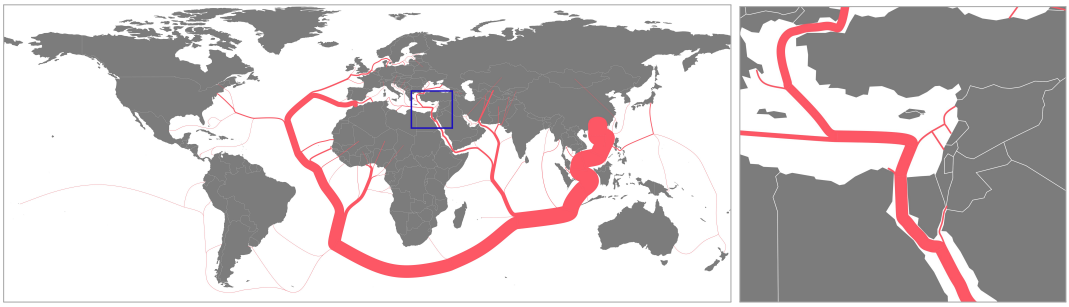
The risk with these distortions is that the geographic entities may no longer be identifiable. In our examples, despite the more or less significant distortions, it is still

[5]The times indicated are the average of 10 different runs.



(a) Entire flow map without distortion

(b) Zoom on an area of the flow map without distortion



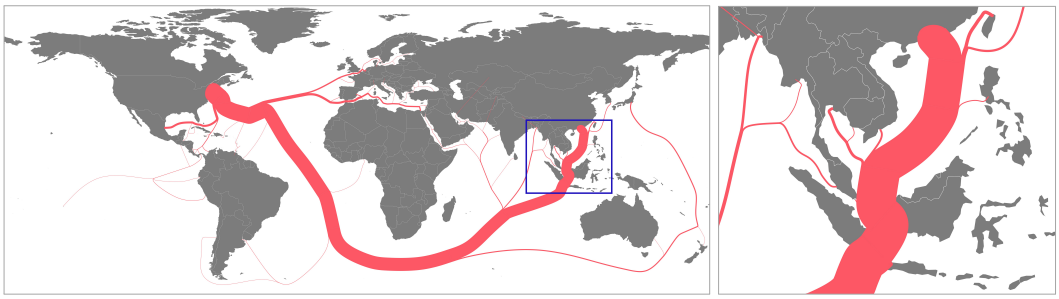
(c) Entire flow map with distortion

(d) Zoom on an area of the flow map with distortion

Figure 11. Results of CHN dataset.

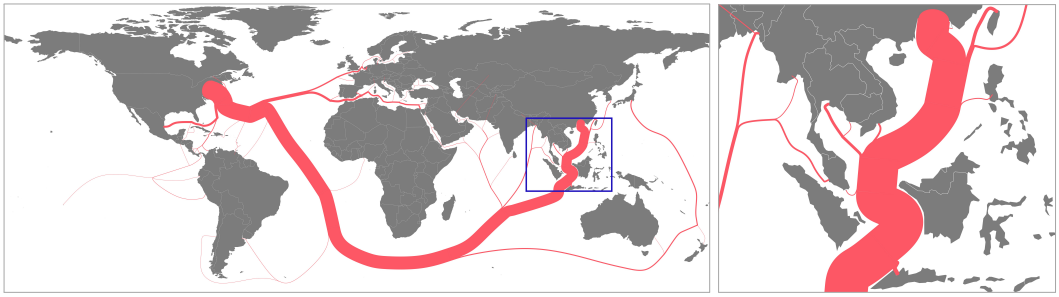
possible to recognize the geographic entities quickly. Figure 15 illustrates an example for Indonesia that compares the geographic features with and without distortion: they keep their positions relative to each other, the deformations are localized, and the features remain identifiable. We performed a small experiment with 5 participants. It consisted in matching several countries labeled on a map before distortion to the map after distortion (without the flow drawing). Two areas were studied: the Mediterranean Sea (as in Figure 14) and the area of Figure 15. Participants had no specific knowledge of mapping and all obtained 100% correct answers.

To obtain all the maps presented above, we used the parameter $p = 5$ (except in Figure 14). As a reminder, this parameter corresponds to the penalty that we apply to the weight of the edges of the graph when computing the tree (see Section 4.2). With $p = 0.5$ and $p = 5$, the flows do not necessarily follow the shortest path in terms of Euclidean distances, but favor the longest edges and then the widest spaces (see Figure 16). This limits the distortions to be performed and makes the computation time of FMD faster. As we can see in Table 3, when p increases, the number of relocated vertices decreases, and the total distance, *i.e.* the sum of the distances between the initial and the final positions of the relocated vertices, is much smaller. Whatever the value of p, the average distance travelled by the relocated vertices remains in the same range. Therefore, the total of the distances between the initial and final vertices depends mainly on the number of vertices relocated and thus decreases when we have a penalty (*i.e.* $p > 0$). As a result, we recommend defining the tree via a straight skeleton as presented in Section 4 and penalizing the weight of the edges when computing the shortest paths in order to avoid areas that are too narrow (Section 4.2).



(a) Entire flow map without distortion

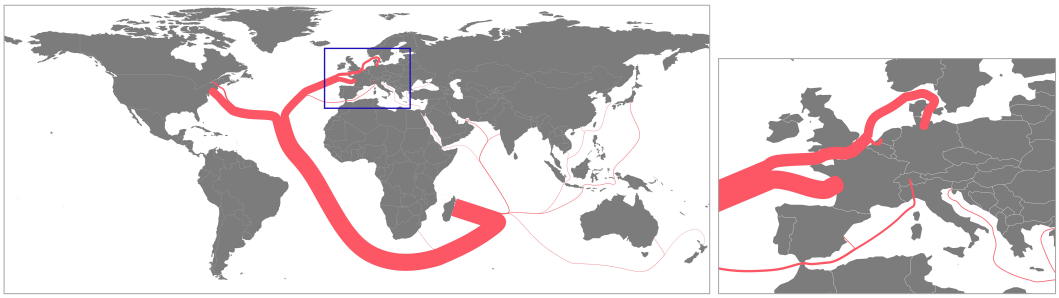
(b) Zoom on an area of the flow map without distortion



(c) Entire flow map with distortion

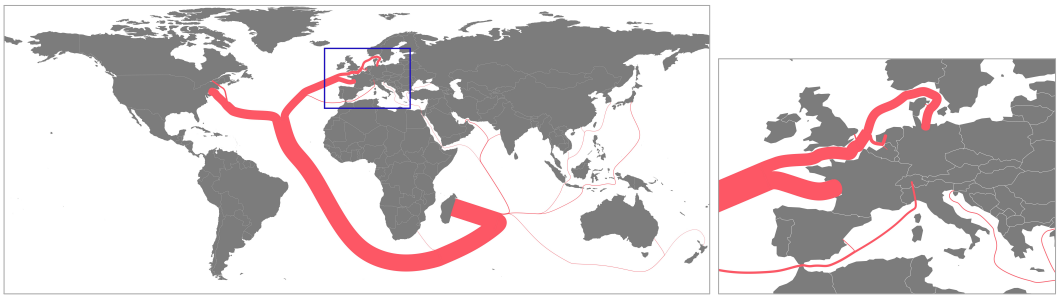
(d) Zoom on an area of the flow map with distortion

Figure 12. Results of USA dataset.



(a) Entire flow map without distortion

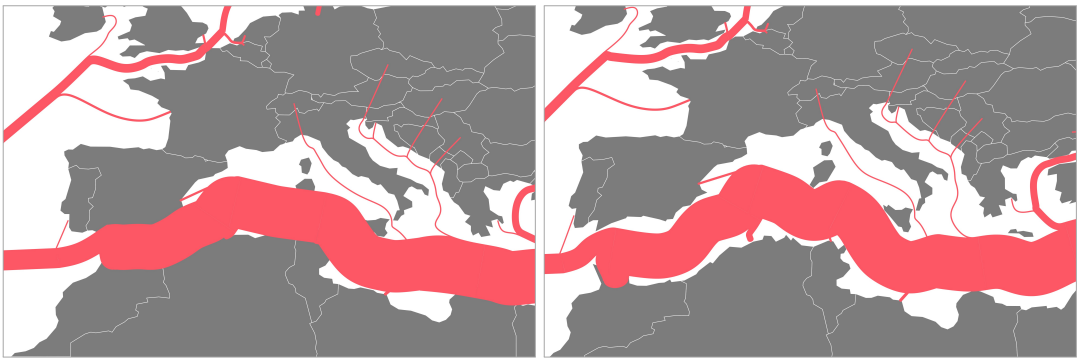
(b) Zoom on an area of the flow map without distortion



(c) Entire flow map with distortion

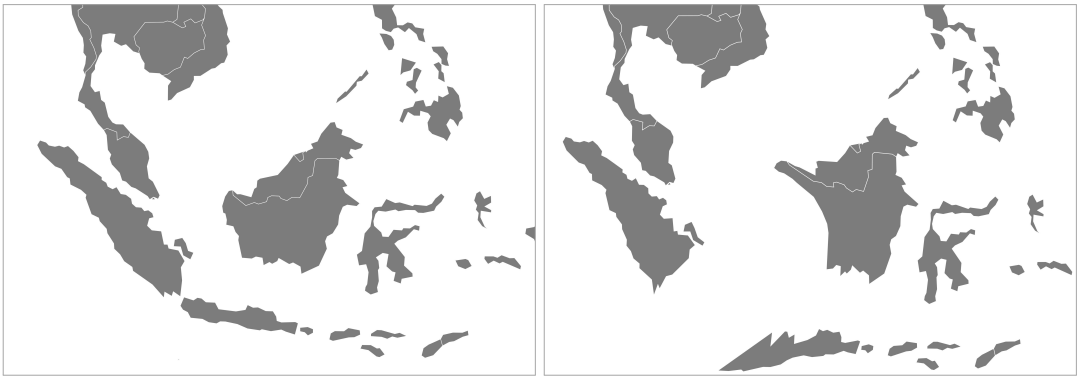
(d) Zoom on an area of the flow map with distortion

Figure 13. Results of MDG dataset.



(a) Without distortion (b) With distortion

Figure 14. Example of the benefits of the distortion (from CHN with $p = 0$ instead of $p = 5$ as in Figure 11)



(a) Without distortion (b) With distortion

Figure 15. Example of distortion (from USA as in Figure 12)

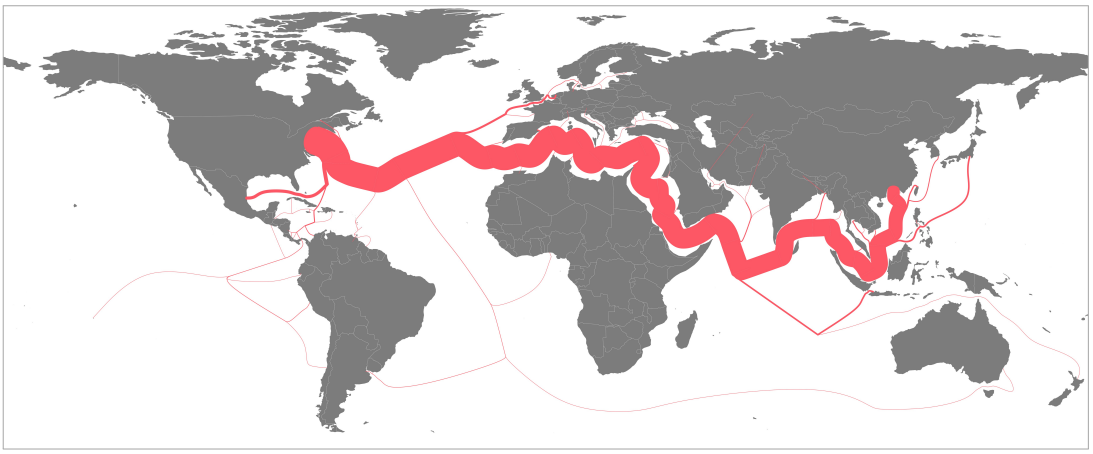
Dataset	**# of relocated vertices**			**Total distance**			**Average distance**		
	$p = 0$	$p = 0.5$	$p = 5$	$p = 0$	$p = 0.5$	$p = 5$	$p = 0$	$p = 0.5$	$p = 5$
CHN	2811	1242	1265	19093	8899	9157	6.79	7.16	7.24
USA	4576	219	219	38536	1453	1453	8.42	6.64	6.64
MDG	794	46	46	4425	280	280	5.57	6.06	6.06

Table 3. Modification of the positions of the vertices of the polygons with the FMD method.

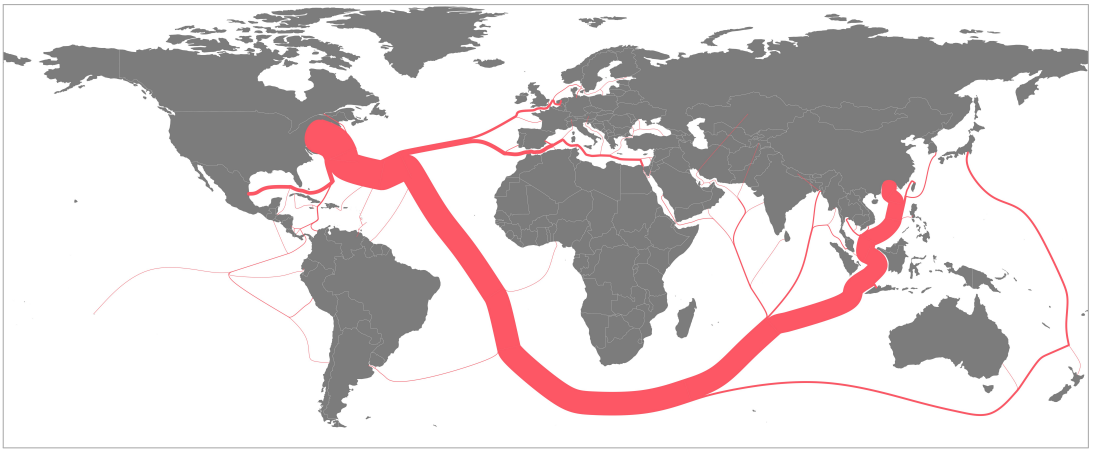
5.4. *Expert feedback*

In this section, we report the feedback of an expert in cartography who was not involved in the development of our method. We solicited his expertise and provided him with the previous sections of this article and an online application containing all the datasets presented above and the possibility to run our method on them step by step. We then asked him to write a report on the strengths and limitations of our approach:

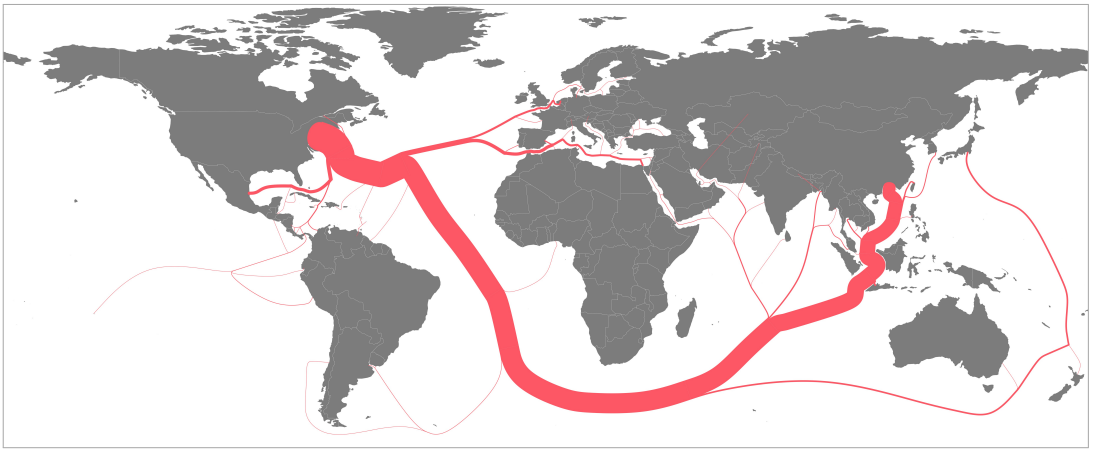
> The FMD method makes it possible to automatically create Minard-style flow maps



(a) $p = 0$



(b) $p = 0.5$



(c) $p = 5$

Figure 16. Results with different p values for USA

from a one-to-many origin-destination matrix. The main difficulty of the method lies in maintaining a balance between the realism of map distortions and the visibility of flows and bordering geographic features. The results illustrate the effectiveness and limitations

of this method, at the global and the local level.

At the global level, the flow maps presented in Figures 11, 12 and 13 make it possible to clearly visualize the main destinations from an origin, as well as the quantities associated with each flow. The overall objective of the flow map is therefore achieved: the reader is able to (1) clearly prioritize the destinations corresponding to the flows from an origin, and (2) associate a quantity with each flow. Graphically, the use of splines offers comfort in terms of the legibility of major maritime routes. When dividing a flow towards several destinations, the orientation of the curvatures in the direction of the flow guarantees consistency in the dynamic reading of the map. On the other hand, the general realism of the proposed routes can be discussed. Indeed, the skeletonization method builds roads far away from the coasts and major shipping lanes. The reader's perception of the actual organization of flows may therefore be impacted. Nevertheless, keeping flows away from the coasts offers an advantage, since it limits the overlapping conflicts between flows and geographic features, and avoids the consequent distortions of geometries.

At the local level, the proposed method makes it possible to avoid overlaps between large flows (materialized by a significant width) and geographic features located near the coasts (which constitute important landmarks for the map reader). The example in the Mediterranean Sea in Figure 14 clearly illustrates the interest of the method in order to visualize islands masked by the symbolized flows. On the other hand, the realism of specific distortions (associated with the proximity of large flows to small geographic features) can be discussed. For example, unrealistic distortions are generated on the islands of Java and Borneo in Figure 12d. Here, the map reader can lose important reference points on the base map, due to the excessive modification of the original shape of geographic features.

In this context, the integration of constraints to preserve the initial shape of geographic features could constitute an interesting perspective in order to maintain a compromise between visibility and realism. Indeed, such constraints would generate distortions that minimize the visual impact on small geographic features, which can be seen as important landmarks for the map reader.

6. Conclusion and future work

In this paper, we have presented a method, FMD, for removing the overlaps between the flows and the geographic features in a one-to-many flow map. We have also presented a method for generating one-to-many flow maps with few overlaps that can be employed before running FMD. Through several examples, the discussion shows that our method reveals hidden objects while maintaining the recognizability of the geographic features. The results meet the main objective, which was to automatically create distorted maps inspired by the hand-drawn maps of Charles Joseph Minard in the 19th century.

In future work, we plan to apply FMD to other kinds of base maps, such as cities. In this case, the geographic features could be the buildings and the empty areas through which the flows must pass could be the streets. Our method could be useful for better visualizing traffic.

The future work will also concern the processing of many-to-many flow maps, which will certainly be the most complex enhancement. Indeed, this will surely lead to an increase in computation time since the number of sequences will be higher. Moreover, several questions arise, in particular for the definition of the tree: should the sequences be grouped together or treated independently depending on the starting point? How should the overlapping of flows be managed?

Data and codes availability statement

The data and codes that allowed us to obtain the results presented in this paper are available in figshare at the permanent link `https://doi.org/10.6084/m9.figshare.21501999`.

Funding

This study was partially funded by EU grant 874850 MOOD and is catalogued as MOOD 065. The contents of this publication are the sole responsibility of the authors and not necessarily reflect the views of the European Commission.

This work has also been supported by the STIC-AmSud program, code: 21-STIC-06.

Notes on contributors

Laëtitia Viau is a PhD student at the Université de Montpellier, France. She is a member of the data mining and visualization research team (ADVANSE) at the Laboratory of Computer Science, Robotics and Microelectronics of Montpellier (LIRMM). Her research interests include geospatial data visualization and visual analytics.

Arnaud Sallaberry is an Assistant Professor at the Université Paul Valéry Montpellier 3, France. He is the head of the applied mathematics and computer science research team (AMIS). He is also a member of the data mining and visualization research team (ADVANSE) at the Laboratory of Computer Science, Robotics and Microelectronics of Montpellier (LIRMM). His research interests include information visualization and visual analytics. He received his PhD in computer science from the Université de Bordeaux, France.

Nancy Rodriguez is an Assistant Professor at the Université de Montpellier, France. She is a member of the data mining and visualization research team (ADVANSE) at the Laboratory of Computer Science, Robotics and Microelectronics of Montpellier (LIRMM). Her research interests include virtual and augmented/mixed reality, interaction, accessibility and visualization. She received her PhD in Computer Science from the Université de Toulouse, France.

Jean-François Girres is an Assistant Professor at the Université Paul Valéry Montpellier 3, France. He is a member of the applied mathematics and computer science research team (AMIS) and is also a member of the ESPACE-DEV research unit. His research interests include spatial data quality, spatial analysis and geovisualization. He received his PhD in Geographic Information Science from the Université Paris-Est, France.

Pascal Poncelet is a Full Professor at the Université de Montpellier, France, and head of the data mining and visualization research team (ADVANSE) at the Laboratory of Computer Science, Robotics and Microelectronics of Montpellier (LIRMM). His research interests include advanced data analysis techniques for emerging appli-

cations, data mining, and visual analytics. He received his PhD in Computer Science from the Université de Nice-Sophia Antipolis, France.

References

Aichholzer, O. and Aurenhammer, F., 1996. Straight skeletons for general polygonal figures in the plane. *In*: *Proceedings of the International Computing and Combinatorics Conference (COCOON)*. Springer, 117–126.

Aichholzer, O., *et al.*, 1995. A novel type of skeleton for polygons. *In*: *Journal of universal computer science*. Springer, 752–761.

Bahoken, F., 2022. Représenter la mondialisation par des flux, le rôle de la distance cartographique perçue. *Mappemonde. Revue trimestrielle sur l'image géographique et les formes du territoire*, (133).

Bertin, J., 1967. *Sémiologie graphique, les diagrammes, les réseaux, les cartes.* Paris: Gauthiers-Villars.

Buchin, K., Speckmann, B., and Verbeek, K., 2011. Flow map layout via spiral trees. *IEEE Transactions on Visualization and Computer Graphics*, 17 (12), 2536–2544.

Carpendale, S., Light, J., and Pattison, E., 2004. Achieving higher magnification in context. *In*: *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. 71–80.

Cui, W., *et al.*, 2008. Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14 (6), 1277–1284.

Debiasi, A., Simões, B., and De Amicis, R., 2014. Force directed flow map layout. *In*: *Proceedings of the International Conference on Information Visualization Theory and Applications (IVAPP)*. 170–177.

Dykes, J., Müller-Hannemann, M., and Wolff, A., eds., 2010. *Schematization in Cartography, Visualization, and Computational Geometry, 14.11. - 19.11.2010*, Dagstuhl Seminar Proceedings, vol. 10461. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany.

Eppstein, D. and Erickson, J., 1999. Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions. *Discrete & Computational Geometry*, 22 (4), 569–592.

Felkel, P. and Obdrzalek, S., 1998. Straight skeleton implementation. *In*: *Proceedings of the Spring Conference on Computer Graphics (SCCG)*. 210–218.

Gaffuri, J., 2007. Field deformation in an agent-based generalisation model: The gael model. *In*: *Proceedings of the Geographic Information Days (GI-Days), Young Researchers Forum*. IFGI Prints num. 30, 1–24.

Gansner, E.R., *et al.*, 2011. Multilevel agglomerative edge bundling for visualizing large graphs. *In*: *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*. 187–194.

Gastner, M.T. and Newman, M.E.J., 2004. Diffusion-based method for producing density-equalizing maps. *Proceedings of the National Academy of Sciences (PNAS)*, 101 (20), 7499–7504.

Gastner, M.T., Seguy, V., and More, P., 2018. Fast flow-based algorithm for creating density-equalizing map projections. *Proceedings of the National Academy of Sciences (PNAS)*, 115 (10), E2156–E2164.

Harrie, L.E., 1999. The constraint method for solving spatial conflicts in cartographic generalization. *Cartography and Geographic Information Science*, 26 (1), 55–69.

Harrie, L.E., Sarjakoski, T., and Lehto, L., 2002. A variable-scale map for small-display cartography. *Proceedings of the Symposium on GeoSpatial Theory, Processing, and Applications*, 8–12.

Haunert, J.H. and Sering, L., 2011. Drawing road networks with focus regions. *IEEE Transactions on Visualization and Computer Graphics*, 17 (12), 2555–2562.

Holten, D., 2006. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12 (5), 741–748.

Holten, D. and Van Wijk, J.J., 2009. Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28 (3), 983–990.

Jenny, B., *et al.*, 2016. Design principles for origin-destination flow maps. *Cartography and Geographic Information Science*, 45 (1), 62–75.

Jenny, B., *et al.*, 2017. Force-directed layout of origin-destination flow maps. *International Journal of Geographical Information Science*, 31 (8), 1521–1540.

Kadmon, N. and Shlomi, E., 1978. A polyfocal projection for statistical surfaces. *The Cartographic Journal*, 15 (1), 36–41.

Mackaness, W., Ruas, A., and Sarjakoski, L., 2007. *Generalisation of geographic information: Cartographic modelling and applications*. Elsevier.

Munzner, T., 2014. *Visualization analysis and design*. A.K. Peters visualization series. A K Peters.

Mustière, S., 1998. GALBE: Adaptive Generalisation. The need for an Adaptive Process for Automated Generalisation, an Example on Roads. *In*: *Proceedings of the Geographic Information Systems PlaNet (GIS'PlaNet)*.

Nusrat, S. and Kobourov, S., 2016. The state of the art in cartograms. *Computer Graphics Forum*, 35 (3), 619–642.

Phan, D., *et al.*, 2005. Flow map layout. *In*: *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*. 219–224.

Rendgen, S., 2018. *The Minard system: The complete statistical graphics of Charles-Joseph Minard*. Princeton Architectural Press.

Ruas, A., 1998. A method vor building displacement in automated map generalisation. *International Journal of Geographical Information Science*, 12 (8), 789–803.

Ruas, A., 2008. *Map generalization*. Springer US, 631–632.

Sarkar, M. and Brown, M.H., 1992. Graphical fisheye views of graphs. *In*: *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. 83–91.

Schöttler, S., *et al.*, 2021. Visualizing and interacting with geospatial networks: A survey and design space. *Computer Graphics Forum*, 40 (6), 5–33.

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

Stott, J.M., *et al.*, 2011. Automatic metro map layout using multicriteria optimization. *IEEE Transactions on Visualization and Computer Graphics*, 17 (1), 101–114.

Sun, S., 2019. A spatial one-to-many flow layout algorithm using triangulation, approximate steiner trees, and path smoothing. *Cartography and Geographic Information Science*, 46 (3), 243–259.

Takahashi, N., 2008. *An elastic map system with cognitive map-based operations*. Springer Berlin Heidelberg, 73–87.

Tobler, W.R., 1987. Experiments in migration mapping by computer. *The American Cartographer*, 14 (2), 155–163.

Tobler, W.R., 2004. Thirty five years of computer cartograms. *Annals of the Association of American Geographers*, 94 (1), 58–73.

Touya, G. and Duchêne, C., 2011. Collagen: Collaboration between automatic cartographic generalisation processes. *In*: A. Ruas, ed. *Advances in cartography and giscience*. Springer Berlin Heidelberg, 541–558.

van Oosterom, P., 1995. The GAP-tree, an approach to "on-the-fly' map generalization of an area partitioning. *In*: J.C. Müller, J.P. Lagrange and R. Weibel, eds. *Gis and generalization: Methodology and practise*. 120–132.

Yamamoto, D., Ozeki, S., and Takahashi, N., 2009. Focus+glue+context: An improved fisheye approach for web map services. *In*: *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS)*. 101–110.

Yuksel, C., Schaefer, S., and Keyser, J., 2009. On the parameterization of catmull-rom curves. *In*: *Proceedings of the SIAM/ACM Joint Conference on Geometric and Physical Modeling (GD/SPM)*. 47–53.