



ARTICLE



## Force-directed layout of origin-destination flow maps

Bernhard Jenny <sup>a,b,c</sup>, Daniel M. Stephen <sup>c</sup>, Ian Muehlenhaus <sup>d</sup>,  
Brooke E. Marston<sup>c</sup>, Ritesh Sharma <sup>e</sup>, Eugene Zhang<sup>e</sup> and Helen Jenny<sup>c</sup>

<sup>a</sup>Faculty of Information Technology, Monash University, Melbourne, Australia; <sup>b</sup>School of Science – Geospatial Sciences, RMIT University, Melbourne, Australia; <sup>c</sup>College of Earth, Ocean, and Atmospheric Sciences, Oregon State University, Corvallis, OR, USA; <sup>d</sup>Department of Geography, University of Wisconsin Madison, Madison, WI, USA; <sup>e</sup>School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA

### ABSTRACT

This paper introduces a force-directed layout method for creating origin-destination flow maps. Design principles derived from manual cartography and automated graph drawing to increase readability of flow maps and graph layouts are taken into account. The origin-destination flow maps produced with our algorithm show flows with quadratic Bézier curves that reduce flow-on-flow and flow-on-node overlaps, and avoid sharp or irregular bends in flow lines. A survey of expert cartographers found that flow maps created with our automated method are similar in quality to manually produced flow maps.

### ARTICLE HISTORY

Received 7 November 2016  
Accepted 13 March 2017

### KEYWORDS

Origin-destination flow maps; graph drawing; cartographic design principles; map design

## 1. Introduction

Flow maps visualize movement and not only demonstrate which places have been affected by movement but also the type, direction, and volume of movement. They are efficient tools for identifying spatial patterns and answer questions about geographic phenomena. Geographic flow mapping is an underdeveloped subfield of information visualization and geographic information science (Rae 2011). Legible flow maps are labor intensive to create manually and require a high level of cartographic expertise. As map automation and map services become more common methods of making maps, well-designed flow maps are increasingly rare. Despite efforts to develop specialized software (Tobler 1987, Rae 2011, Kim *et al.* 2012, Gerlt 2013) and web map services (Guo 2012), flow map functionality is limited and cumbersome to use in current geographic information systems and web mapping software.

Our computational flow mapping method takes cartographic design principles into account. It creates non-branching origin-destination flow maps where the curvature of the flows can be adjusted freely because the geometry of the flow path is unknown or irrelevant for the visualization.

## 2. Related work

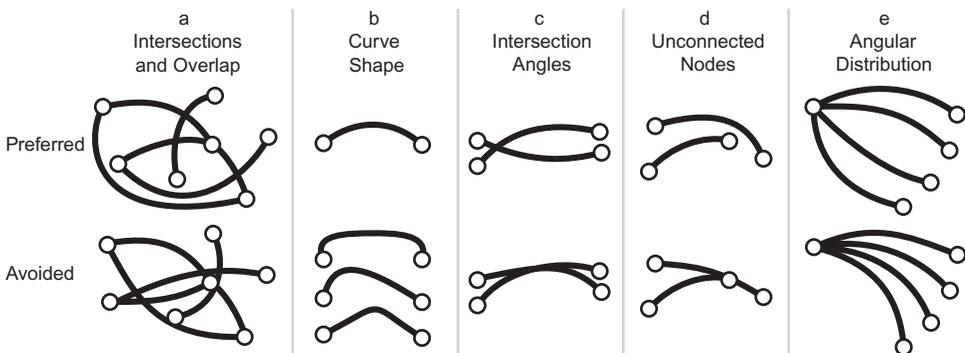
### 2.1. Design principles for flow maps

Cartographic design principles are guidelines that help cartographers to create legible and aesthetically pleasing maps. One of the primary goals for designing origin-destination flow maps is to reduce overlaps between flows because flow-on-flow and flow-on-node overlaps often create ambiguous maps that are difficult to read accurately. Jenny *et al.* (2016) compiled cartographic design principles for origin-destination flow maps from cartographic literature (Imhof 1972, Dent *et al.* 2008, Slocum *et al.* 2009) and graph drawing, a discipline in computer science concerned with generating diagram layouts for graphs.

Graph drawing is relevant to mapping origin-destination flows because flows on maps form a graph; the graph nodes are the starts and ends of flows and the graph edges are the connecting flows. In graph drawing, as in flow mapping, the geometry of edges is adjusted to reduce the number of flow-on-flow and flow-on-node overlaps. Because graph drawing and flow mapping share similar design characteristics, user studies evaluating design principles for graph drawing are relevant to the design of cartographic flow maps. However, principles from graph drawing need to be adapted to flow mapping because unlike graph drawing, the starts and ends of flows in maps are geographically constrained and cannot be freely positioned.

A number of design principles for the design of flow geometry and the arrangement of flow lines were verified by user studies. These user studies show that applying design principles decreases error rates and reading time. Before developing our automated method, we completed a content analysis with 97 manually created flow maps to identify design principles applied by professional cartographers. We also conducted a user study to test additional design options. The results of the user study led us to recommend the following set of design principles for flow maps (Figure 1) (Jenny *et al.* 2016).

The number of flows overlapping should be minimized (Purchase *et al.* 1996, Ware *et al.* 2002, Huang *et al.* 2008). Curving flows can reduce overlaps (Figure 1(a)). Sharp bends should be avoided (Purchase *et al.* 1996), and symmetrically curved flows are preferable to asymmetric flows (Figure 1(b)) (Jenny *et al.* 2016). Acute-angle crossings of



**Figure 1.** Design principles for origin-destination flow maps: preferred (top) and avoided (bottom) arrangements (from Jenny *et al.* 2016).

flows should also be avoided (Figure 1(c)) (Huang *et al.* 2008, 2014). Flows must not pass under unconnected nodes (Figure 1(d)) (Wong *et al.* 2003). Flows should be radially distributed around nodes (Figure 1(e)) (Huang 2007). Small flows are placed on top of large flows (Dent *et al.* 2009).

Design principles for indicating flow quantity and direction are well established in cartography. Quantity is best represented by adjusting the width of flows (Dent *et al.* 2009). Arrowheads are the best indication for direction on flow maps (Jenny *et al.* 2016).

## 2.2. Automated creation of origin-destination flow maps

The first automated methods for creating flow maps used straight bands to connect start and end positions. Tobler (1987) dates the earliest example of a digitally generated flow map to 1959. Kern and Rushton (1969) used thin straight lines to connect origins to destinations. Kadmon (1971) introduced digital flow lines with varying widths to indicate quantities, and Wittick (1976) extended these digital methods to map quantitative flows in networks, and others (Evatt *et al.* 1981, Tobler 1981, 1987) later refined and extended these digital approaches.

Bézier curves (Brandes *et al.* 2000, Guo 2009, Wood *et al.* 2011, Guo and Zhu 2014) and sections of circles (Ho *et al.* 2011) have been used for automated origin-destination flow maps. In these digital applications, the geometric arrangement of flows is neither optimized to minimize the number of intersections among flow lines, nor reduce the number of overlaps between flows.

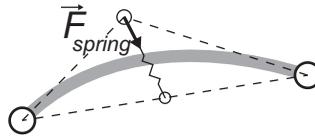
A few authors explore methods for the automated creation of branching flow maps (Phan *et al.* 2005, Verbeek *et al.* 2011, Nocaj and Brandes 2013, Debiasi *et al.* 2014a, 2014b). Complementary approaches for flow clustering (Zhu and Guo 2014) and edge bundling are also explored. Zhou *et al.* (2013) present an overview of edge bundling applied to flow visualization.

In graph drawing, curved edges are used for node-link diagrams (Riche *et al.* 2012, Xu *et al.* 2012). The reasoning behind using curved lines in graph drawing can also be applied to flow mapping: better use of canvas space and reduced visual clutter because fewer edges overlap or intersect.

## 3. Force-directed layout method

We develop a force-directed layout method to automate the creation of non-branching origin-destination flow maps. Flows are modeled with quadratic Bézier curves that use one control point, which is placed off the line. Quadratic Bézier curves are an appropriate choice for origin-destination flow maps because they cannot have loops, are never S-shaped, and are included in common graphics libraries and exchange formats. If necessary, they can be converted to cubic curves for editing in vector graphics software.

The control point of each quadratic Bézier curve is attached to a spring (Figure 2). The opposing end of the spring is attached to the midpoint of the line between the start and end points of the flow. Other flows exert repulsing forces onto the control point of the quadratic Bézier curve. Inspired by methods for force-directed graph drawing (Brandes and Wagner 2000, Kobourov 2012, Fink 2013), an iterative process computes the equilibrium state between the retracting forces of the springs and the repulsing forces



**Figure 2.** Quadratic Bézier curve modeling a flow with a spring pulling the control point toward the midpoint between start and end points.

of other flows. We develop a series of refinements to this algorithm (discussed below). For example, spring stiffness is adjusted for peripheral flows, excessively asymmetric flows are avoided, flows are moved away from overlapping nodes, and control points are constrained to a bounding rectangle to avoid excessive curvature. Our method is controlled by a set of parameters. We provide default parameter values that can be adjusted using a graphical user interface.

The pseudo-code below provides an outline of the method. On each iteration, for each flow  $f$ , five forces are computed:  $F_{flows}$  is the total repulsing force exerted by all other flow curves;  $F_{nodes}$  is the total repulsing force exerted by all other flow nodes;  $F_{antiTorsion}$  is a force countering asymmetric distortion of the flow;  $F_{spring}$  is the force exerted by the spring on the Bézier control point; and  $F_{angRes}$  is a force improving the angular resolution of flows around nodes. Each of the five forces is scaled by an individual, user-defined weight. An additional weight  $w$  is applied to the first four forces. This weight linearly decreases from 1 to 0 to decrease the energy in the system toward the end of the iterative computations. The stabilizing weight for  $F_{angRes}$  is  $w - w^2$ . We use this weight instead of  $w$  to reduce the influence of the angular adjustments during the initial phase of computations. We use the following default values for the five weights:  $w_{flows} = 1$ ;  $w_{nodes} = 0.5$ ;  $w_{antiTorsion} = 0.8$ ;  $w_{spring} = 1$ ; and  $w_{angRes} = 3.75$ . The default is 100 iterations for small flow data sets. For complex or large flow maps, a larger number of iterations is necessary for the layout to converge to a stable equilibrium.

$F_{total}$  is the sum of the five weighted forces and is applied to the Bézier control point of  $f$  to compute a new translated control point position. The translated control points are constrained to stay within a rectangle aligned with the start-to-end line of the flow. They are also constrained to stay within a rectangle aligned with the canvas space. After the control points of all flows have been translated, intersecting flows and flows that overlap unconnected nodes or arrowheads are moved.

**Input:** Flows in map  $M$

**Output:** An improved layout of  $M$

$j = 0$

**for**  $i$  in  $0 \dots \#iterations - 1$  **do**

// forces exerting on Bézier control points

$w = 1 - i/\#iterations$

**for each** flow  $f$  in  $M$  **do**

$F_{flows} \leftarrow$  force of flows in  $M$  against  $f$  (Section 3.1.1)

$F_{nodes} \leftarrow$  force of nodes in  $M$  against  $f$  (Section 3.1.2)

```

FantiTorsion ← anti-torsion force of f (Section 3.1.3)
Fspring ← spring force of f (Section 3.1.4)
FangRes ← angular resolution force of f (Section 3.1.5)
Ftotal = W · (Wflows · Fflows + Wnodes · Fnodes + WantiTorsion · FantiTorsion +
         Wspring · Fspring) + (W - W2) · WangRes · FangRes
pf ← copy control point of f and translate by Ftotal
Constrain pf to rectangle aligned with f (Section 3.2.1)
Constrain pf to canvas rectangle (Section 3.2.1)
end for
for each flow f in M do
    Assign control point pf to f
end for

// reducing flow intersections (Section 3.2.2)
P ← pairs of intersecting flows connected to a shared node
for each pair p in P do
    Move control points of both flows of p
end for

// moving flows off nodes and arrowheads (Section 3.2.3)
if (i > 10% of #iterations and j ≤ 0) then
    N ← flows overlapping nodes and arrowheads
    j ← number of iterations until next flow is moved off nodes
    n ← number of flows to move off nodes
    for each flow f in N do
        if (geometry for f without overlap exists) then
            Move control point pf of f
            break for loop if n flows have been moved
        end if
    end for
else
    j = j - 1
end if
end for

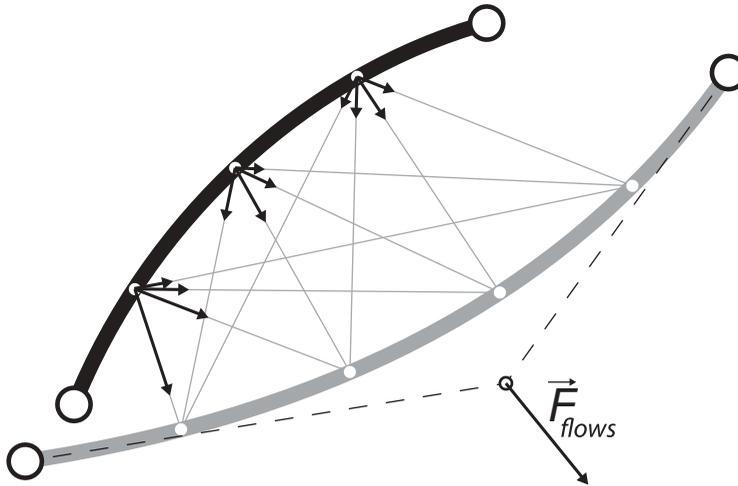
```

### 3.1. Forces exerting on Bézier control points

#### 3.1.1. Curving flows with flows-against-flow forces

Flows on the map exert repulsing forces against the control points of all other flows, which curve the flows. The purpose of the flows-against-flow force is to spread flows apart to reduce overlap and avoid intersections between flows.

We calculate the force  $F_{flows}$  exerted on the control point of flow *f* by all other flows on the map by first locating evenly spaced points along all flows (Figure 3). Points along

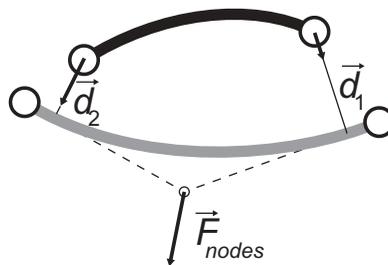


**Figure 3.** Flow-against-flow forces are the distance-weighted sum of forces between points on flows. The forces of only one flow exerting on a second flow are shown.

all other flows emit a repelling force against each point of  $f$ . The repelling forces are interpolated for each point of  $f$  using Shepard's (1968) inverse distance weighting. The resulting force for point  $p$  on  $f$  is:  $F_p = \sum d_i \cdot w_i / \sum w_i$ , with  $w_i = 1/|d_i|^\alpha$ , where  $|d_i|$  is the distance between  $p$  and another point not on  $f$ , and  $\alpha$  is a parameter. The final force  $F_{flows}$  is the sum of the forces exerted onto all points of  $f$  divided by the number of points  $n$  on  $f$ :  $F_{flows} = \sum F_p / n$ .  $F_{flows}$  is applied on the control point of  $f$ . The default value for parameter  $\alpha$  is 4.

### 3.1.2. Curving flows with nodes-against-flow forces

Nodes exert a repelling force that moves flows apart to prevent flows from touching unconnected nodes. To compute the repelling force  $F_{nodes}$  for flow  $f$ , for each node  $n_i$  not connected to  $f$ , we compute the vector  $d_i$  between node  $n_i$  and the closest point on  $f$  (Figure 4).  $F_{nodes}$  is computed with inverse distance weighting:  $F_{nodes} = \sum d_i \cdot w_i / \sum w_i$ , with  $w_i = 1/|d_i|^\beta$ , where  $|d_i|$  is the length of  $d_i$ , and  $\beta$  is a parameter.  $F_{nodes}$  is applied to the control point of  $f$ . The default value for parameter  $\beta$  is four. Figure 5 shows the effect of the nodes-against-flow forces.



**Figure 4.** Nodes-against-flow forces by the nodes of the black flow on the gray flow.

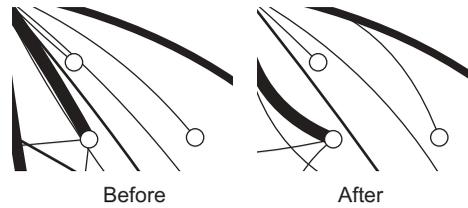


Figure 5. Before (left) and after (right) applying nodes-against-flow forces.

### 3.1.3. Anti-torsion to counter asymmetric flows

The force  $F_{antiTorsion}$  reduces Bézier curve asymmetry by pushing the control point toward the perpendicular bisector of the line between the start and end points (Figure 6). The length of  $F_{antiTorsion}$  equals the distance between the control point and the perpendicular bisector, which is computed with  $\gamma$ . Figure 7 shows the effect of the anti-torsion forces.

### 3.1.4. Spring force to reduce curvature

The spring force  $F_{spring}$  of each flow pulls the control point of the curve toward the base point, the point halfway along a line between the start and end points of the flow (Figure 2). This results in a straight flow if no external force is exerted. The purpose of the spring force is to oppose external forces that are causing the flow to curve, thereby preventing the flow from curving too much and creating equilibrium with external forces.

The spring force  $F_{spring}$  is computed with Hooke's law:  $F_{spring} = k \cdot L$ , where  $k$  is the spring constant and  $L$  is the length of the spring. The spring constant  $k$  varies with the length of the flow. The spring constant for a flow with zero length is defined by parameter  $k_{short}$ .

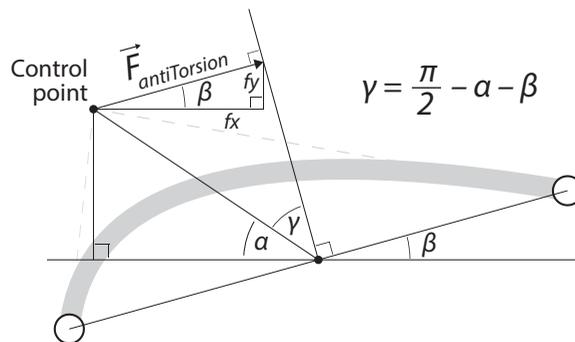


Figure 6. Calculation of the anti-torsion force pulling the Bézier control point toward the perpendicular bisector of the start–end line.

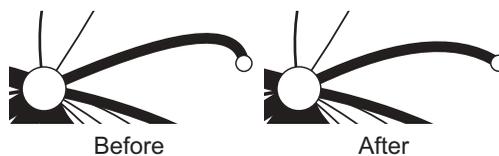


Figure 7. Before (left) and after (right) applying anti-torsion forces.

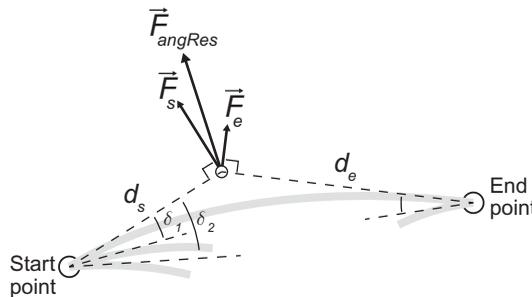
Parameter  $k_{long}$  is the spring constant for the longest flow on the map. The spring constant for flow  $f$  is linearly interpolated with  $k = (k_{long} - k_{short}) \cdot (B/B_{max}) + k_{short}$ , where  $B$  is the distance between the start and end points of flow  $f$ , and  $B_{max}$  is the longest  $B$  of all flows in the map.

Peripheral flows are made straighter by increasing their spring constant. This counters the tendency of peripheral flows to curve in an asymmetric way, which is caused by the unilateral distribution of external forces for peripheral flows. The peripherality of flow  $f$  is computed by dividing the Bézier curve into straight-line segments. For each line segment, we compute the force  $F_s$  that all other flows exert with the method for computing flow-against-flow forces outlined in 3.1.1. We sum the force vectors,  $\sum F_s$ , and the length of the force vectors,  $\sum |F_s|$ . For a flow surrounded on both sides by other flows, the ratio  $\sum F_s / \sum |F_s|$  is close to 0 because the opposing forces  $\sum F_s$  exerted by surrounding flows tend to compensate each other. For a peripheral flow, the ratio will be close to 1. The spring constant  $k$  is multiplied by  $\sum F_s / \sum |F_s| \cdot C_p + 1$  to increase the spring constant for peripheral flows. Parameter  $C_p$  adjusts the amount of correction for peripheral flows. We use the following default values for the three parameters:  $k_{short} = 0.5$ ;  $k_{long} = 0.05$ ; and  $C_p = 2.5$ .

### 3.1.5. Angular resolution of flows around nodes

The purpose of the  $F_{angRes}$  force is to increase the angular resolution of flows sharing the same node to avoid overlaps between flows close to nodes. Our method for quadratic Bézier curves is inspired by the work of Brandes and Wagner (2000) and Finkel and Tamassia (2005).

$F_{angRes}$  pushes the control point of a flow in the direction that increases the angular resolution at a node. For each flow, the angular differences  $\delta_i$  to all other flows connected to the same nodes are computed (Figure 8).  $\delta_i$  is computed by calculating the angular difference between the lines connecting the start or end point with the Bézier control points. For the start point, the angular differences  $\delta_i$  are converted to a force:  $F_s = d_s \cdot \sum \text{sign}(\delta_i) e^{-K\delta_i^2}$ , where  $d_s$  is the distance between the start point and the control point,  $\delta_i$  are the angular differences at the start point, and  $K$  is a parameter. Similar computations using  $d_e$ , the distance between the end point and the control point, are carried out, resulting in  $F_e$ . We sum the two forces and clamp the length of the resulting force to  $\min(d_s, d_e)/C$ , where  $C$  is a parameter. The clamping avoids excessive corrections. The default values we use for the parameters are  $K = 4$  and  $C = 4$ .



**Figure 8.** Angular resolution is increased by pushing the control point away from flows connected to the same nodes.

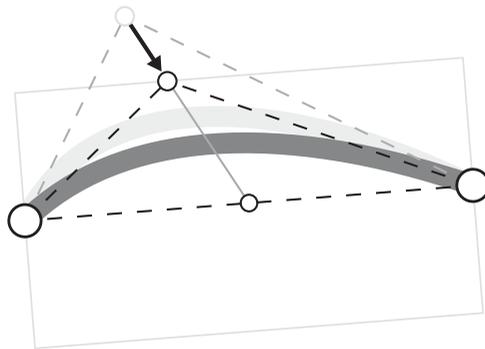
### 3.2. Layout constraints

The forces described above are applied to all flows in the map. We apply additional constraints to the geometry of flows to adjust strongly asymmetric or curved flows, reduce the number of intersections, and move overlapping flows away from nodes to improve legibility.

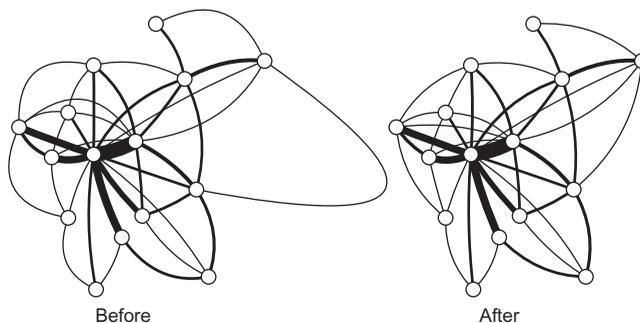
#### 3.2.1. Constraining curvature and asymmetry

To limit the maximum curvature of a flow and prevent excessive asymmetric curving, the Bézier control point is constrained to the inside of a rectangle oriented with the line connecting the start and end points of the curve (Figure 9). The length of the perpendicular sides of the rectangle is a percentage of the distance between the start and end points. The default value for this parameter is 50%. If the control point is outside the rectangle, the control point is moved to the intersection of a line connecting the control point to the midpoint between the start and end points and the edge of the rectangle. Figure 10 shows the effect of the rectangle constraints.

The control points of all Bézier curves are also constrained to the inside of a rectangle around all flows. This prevents flows around the edges of the map from curving outward excessively and constrains the map layout to a specified area. If the forces exerted on a



**Figure 9.** The control point is constrained to the inside of the rectangle. The arrow indicates the displacement applied to the control point.

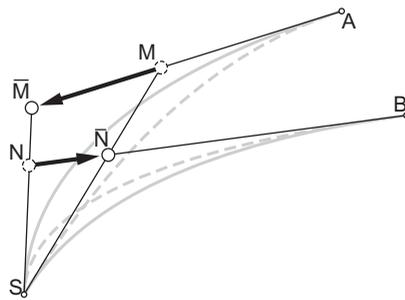


**Figure 10.** Before (left) and after (right) constraining control points to rectangles aligned with the start–end line of flows.

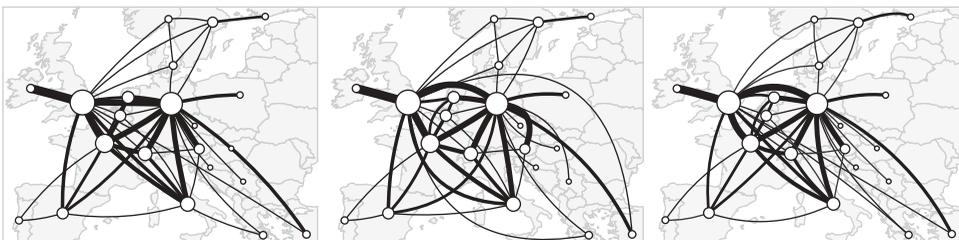
flow push the control point outside the canvas, the same method for rectangles around flows is applied. The canvas size can be increased if desired. The default canvas size is twice the length and width of the bounding box around all flows.

### 3.2.2. Reducing flow intersections

Acute-angle intersections often occur for flows connected to a shared node. For example, in [Figure 11](#), two intersecting flows represented as dashed gray lines are both connected to the shared node  $S$ . We identify pairs of intersecting flows connected to a shared node, then adjust the position of the Bézier control points  $M$  and  $N$  of the two intersecting flows. Control point  $M$  is moved along a line connecting the control point  $M$  and the non-shared node  $A$  ([Figure 11](#)) to position  $\bar{M}$ . The new position is the intersection of the line through  $A$  and  $M$  with the line through  $S$  and  $N$ . Control point  $N$  is moved to  $\bar{N}$ , which is the intersection of the line through  $N$  and  $B$  and the line through  $S$  and  $M$ . These adjustments to the Bézier control points are applied at each iteration after the forces described in [Section 3.1](#) have been applied to the control points of all flows (see the pseudo-code at the beginning of [Section 3](#)). The control points are constrained to lay within the limiting rectangles described in [Section 3.2.1](#). [Figure 12](#) (center and right) shows the effect of moving control points to avoid intersections. In the center of [Figure 12](#), the following flow pairs intersect: from Germany to Austria and Turkey, from Great Britain to Greece and Italy, from Great Britain to Germany and Turkey.



**Figure 11.** Displacement of Bézier control points of intersecting flows connected to the same node  $S$ . Control points  $M$  and  $N$  are moved to  $\bar{M}$  and  $\bar{N}$ , respectively. Gray dashed lines indicate the initial intersecting flows defined by  $M$  and  $N$ ; solid lines indicate the amended flows defined by  $\bar{M}$  and  $\bar{N}$ .



**Figure 12.** Initial layout (left), layout after moving flows off nodes (center), and layout after reducing intersections (right). The map at the center is Map 6 for the expert study. (Map after Telegeography Inc. (2000))

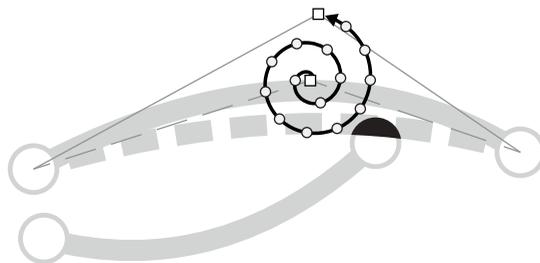
After applying the method described in this section, these flow pairs do not intersect (right of Figure 12).

### 3.2.3. Moving flows off nodes and arrowheads

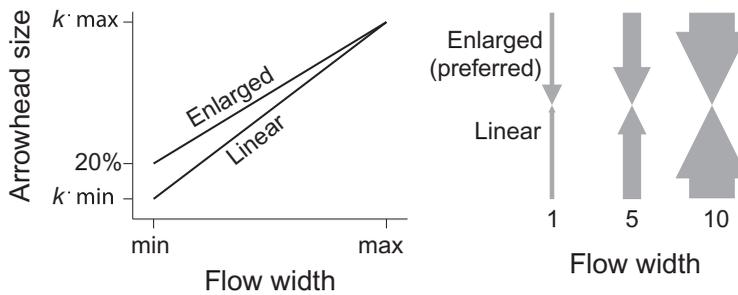
The nodes-against-flow forces described in Section 3.1.2 do not guarantee that maps will not have flows overlapping nodes or arrowheads of other flows. In dense areas, it is still possible for flows to overlap nodes and arrowheads they are not connected to, causing significant ambiguity. To avoid this, flows are moved to create a minimum distance between flows and obstacles (unconnected nodes or arrowheads). The default parameter value for the minimum distance of flows from obstacles is four pixels. Figure 12 (left and center) shows the effect of moving flows off nodes. In the map on the left of Figure 12, flows between Germany and Turkey, Germany and Croatia, Great Britain and Germany, Great Britain and Greece, and France and the Netherlands intersect nodes. After applying the method described in this section, these flows do not intersect any nodes (center of Figure 12). The pseudo-code at the beginning of Section 3 is an overview of the algorithm; details are included below.

If a flow overlaps an obstacle, the control point of the Bézier curve is moved along an Archimedean spiral centered on the initial location of the control point until the curve is a minimum distance from all obstacles (Figure 13). Once a flow is moved away from obstacles, its control point location cannot be changed in subsequent iterations. This prevents the control point from moving back to the same location and the flow from overlapping the same obstacle. Because a control point can move a relatively large distance, additional conflicts with neighboring flows can result. Ideally, each movement away from an obstacle is followed by a few iterations without similar movements in order to give neighboring flows the opportunity to stabilize their geometry. Similarly, no flows are moved away from obstacles during the first 10% of iterations to let the flows find a roughly stable geometry. When iteration  $i$  after 10% of all iterations is reached, the following algorithm is applied.

All flows overlapping obstacles are identified and stored in set  $N$ . We compute two values from the number of overlapping flows. The first value is the number of iterations before the next flow is moved off obstacles:  $j = (\#iterations - i) / (|N| + 1) / 2$ , where  $|N|$  is the number of flows in  $N$ , and the division by 2 is heuristic to increase the number of



**Figure 13.** The dashed flow overlaps an unconnected node. The overlap area is illustrated in black. Its control point (square symbol) is moved along an Archimedean spiral until there is a minimum distance between the flow and all obstacles. Sampling points along the spiral are marked with circles.



**Figure 14.** The size of arrowheads is enlarged for thin flows to increase readability.

iterations without flows overlapping obstacles. The second value is the number of flows to be moved in the current iteration:  $n = \text{ceil}(|N|/(\#iterations - i - 1))$ . The value of  $n$  is usually 1, unless there are more flows overlapping obstacles than remaining iterations. An attempt is then made to move  $n$  flows away from all obstacles. Once  $n$  flows are moved away from obstacles, the next force iteration is started. This algorithm is repeated after  $j$  iterations.

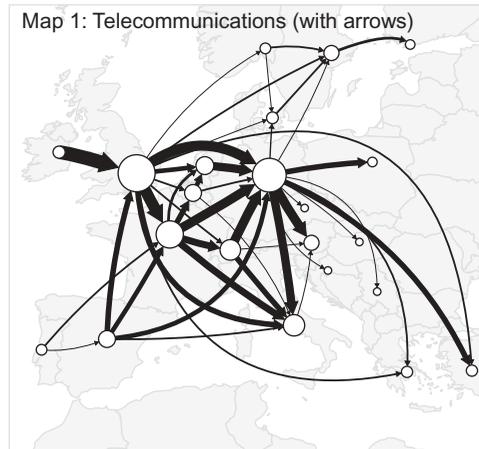
To find a control point location that does not result in an overlap with any obstacle, we create and test candidate positions for the control point along an Archimedean spiral (Figure 13). The spiral is centered on the initial control point location. The windings of an Archimedean spiral are separated by a constant distance, which we set to the minimum distance parameter with a default value of four pixels. The candidate positions are separated by the same minimum distance parameter along the spiral. Candidate points outside of the constraining rectangle described in Section 3.2.1 are not considered. We sequentially place the Bézier control point on each candidate position, moving along the spiral, until a position for the control point is found that results in a curve with a minimum distance from all obstacles. If, after checking all possible positions, no position is found where the flow is not the minimum distance from other points, the flow remains in its original position.

### 3.3. Arrowheads

The lengths and widths of arrowheads are scaled to the width of the flows through linear interpolation, but the smallest arrowheads are increased for better visibility (Figure 14). The default enlargement factor for the thinnest flow line is 20%. The default length and width parameters of arrowheads are 1.6 times the flow width.

## 4. Expert evaluation

To evaluate our force-directed layout method for origin-destination flow maps, we created six maps (Figure 12 center, and Figures 15–19) with our method and invited professional cartographers to provide feedback. We conducted this study with experts in flow mapping instead of general user subjects to more easily and quickly identify design issues and to simplify the study setup. Our study was not designed to assess the effectiveness of flow map design principles. Instead, our study uses the experts to critique the results of our automated flow mapping method. The goals of the survey



**Figure 15.** Map 1 for the expert evaluation (after Telegeography Inc. (2000)). This is a variant of Map 6 (Figure 12 center) with arrowheads.



**Figure 16.** Map 2 for the expert evaluation (after Board *et al.* (1970)).

were to (1) collect qualitative feedback on the curvature, distribution, and intersection of flows, and the design of arrowheads and (2) learn whether additional design aspect should be added to the automated method.

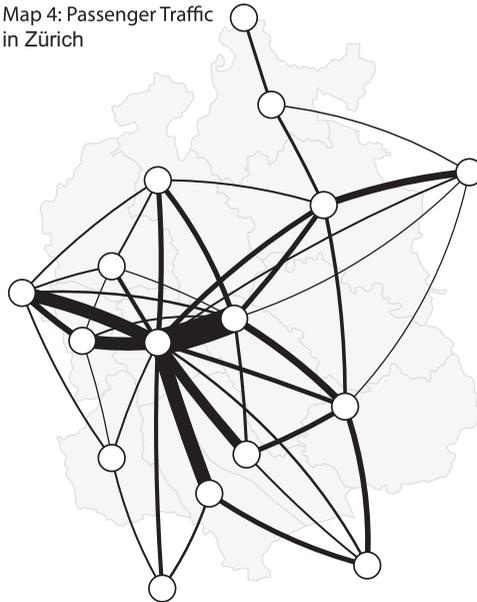
The maps, modeled after existing origin-destination flow maps, contained between 27 and 56 flows and 10 and 32 nodes. Only one of the six maps indicated flow direction (Figure 15) using arrowheads of varying size. We assumed that a single map with arrowheads would be sufficient to judge the size and design of arrowheads. All maps had a title, but did not include legends, toponyms, details about the represented data, or additional information because we were not seeking feedback on the design of these

Map 3: International Investments



**Figure 17.** Map 3 for the expert evaluation (after Roxburgh *et al.* (2009, p. 18)).

Map 4: Passenger Traffic  
in Zürich

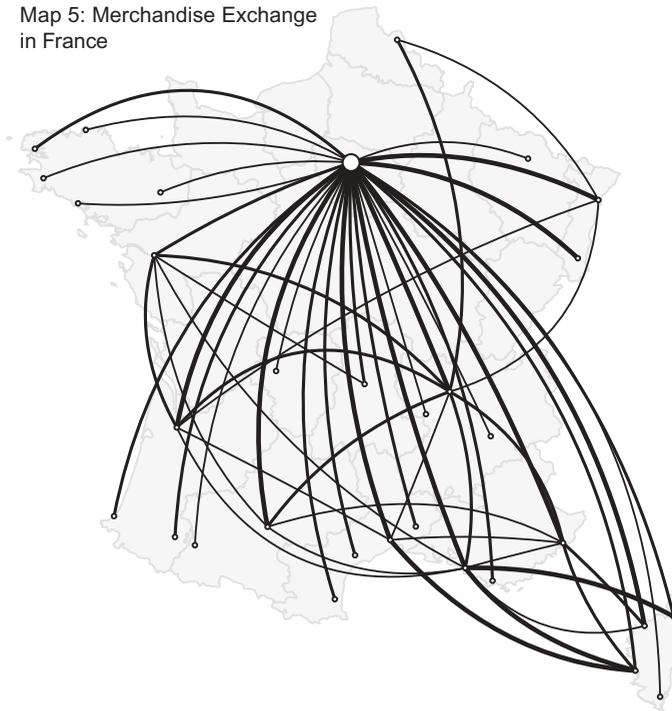


**Figure 18.** Map 4 for the expert evaluation (after Baudirektion Kanton Zürich (2003, p. 10)).

elements. Parameters for the automated method were adjusted to arrange flows in a way we found aesthetically pleasing.

All components of the force-directed layout method described in the previous section were used to create the six maps with two exceptions. First, the number of intersections

Map 5: Merchandise Exchange in France



**Figure 19.** Map 5 for the expert evaluation (after Atlas de France (2000, p. 73)).

between flows connected to shared nodes was not reduced (Section 3.2.2). This resulted in a few acute-angle intersections. In Figure 15, for example, the flows from Germany to Greece and from Germany to Serbia intersect at an acute angle. Second, while we treated nodes as obstacles that flows should not cross, we did not treat arrowheads as obstacles (Section 3.2.3). This resulted in a few arrowheads partially covered by other flows. For example, in Figure 15, the arrowhead for the flow from Spain to France is partially covered by another flow. We added these two components to the force-directed layout method after the expert evaluation to address comments received from survey participants.

We solicited comments on features that the automated method handled well and that could be improved. The following question was asked: 'We are looking for feedback on aspects that the method handles well and aspects that need to be improved. Please comment on:

- Curvature of flows
- Distribution of flows around points
- Intersections and overlaps among flows
- Design of arrows
- Any other design-related aspects'

We asked experts not to compare our maps with the original maps, because we did not want experts to identify differences between automated and original maps. Instead

we wanted them to evaluate automated maps based on their individual expertise and preference. We also asked experts to ignore the lack of legends and other map elements.

We received feedback from nine experienced cartographers who produce flow maps for different media outlets, teach courses in thematic mapping, or have published cartographic textbooks discussing flow map design (for names and affiliations, see the Acknowledgments section).

**General comments:** All cartographers commented positively on the design of maps created with our method. Six cartographers provided very positive general feedback and were surprised by the impressive quality of the maps: ‘Overall (and especially for an automated process) excellent’, ‘generally, the aspects you’ve asked for feedback on seem to work well for the examples’, ‘great work’, ‘I don’t really see any issues’, ‘overall I think the maps are clear’, and ‘I really don’t have anything negative to say. I like the overall design of the maps and the manner in which the flows are depicted’.

**Curvature:** Five cartographers provided feedback on the level and type of curvature, and comments were encouraging: ‘In terms of the line curvature [...], I think [the maps] are excellent’, ‘you’ve got the [...] curvature well scaled’, ‘I appreciate that the lines are curved’, ‘good; with clear arcs that are generally easy to follow’, and ‘curvature in general looks good’. Three cartographers pointed out that straight flows should be avoided by curving all flows for aesthetic reasons. Cartographer 1: ‘Some [curved lines] appear “stiffer” than others. In Map 3 [Figure 17], for example, the flow from Australia to Europe is nearly straight, while several other flows are much more curved. Sometimes this appears to be a matter of necessity, but other times it looks like there’s flexibility in how curved the lines could be, and there’s a bit of disharmony. It’s not severe, but it’s something that I noticed’. Cartographer 2: ‘I wonder if all lines should be curved as there is sometimes a visual dissonance between straight lines and curves when there is no intended significance behind this difference in form’. Cartographer 3: ‘Because of (carto) graphic reasons, there should be no straight line in these maps. Even “direct connections” without obstacles in between should be slightly bowed’. One cartographer recommended more variety in curvature: ‘Curvature [...] is more consistent than I would expect in a nicely design[ed] flow arrow map; [maps] would look better with more variety in the radius of curves and with more curves that are based on asymmetrical radii’. One cartographer recommended avoiding short flows with strong curvature: ‘There are a couple of awkward short paths with high curvature in Map 6 [Figure 12 center] that tend to coalesce with other lines but it’s a minor issue’.

**Angular resolution:** Three cartographers commented that the angular resolution of flows around nodes could be improved for some nodes. Cartographer 1: ‘I probably wouldn’t have paid much mind to the way the flows are distributed around the points, if you hadn’t called attention to it. But now that you mention it, they could sometimes be a little more even. On Map 3 [Figure 17], for example, there are a series of lines coming off of North America that could be spaced a bit better’. Cartographer 2: ‘[I try to avoid] lines twisting around each other (e.g. on map 3 [Figure 17] the lines from USA to Central Asia and East Asia intersect each other over Europe)’. Cartographer 3: ‘Need better radial separation around origin and destination points’.

**Overlaps and intersections:** Comments on the number of overlaps and intersections were positive overall: ‘avoidance [is] excellent’, and ‘for an automated method, I am impressed with its ability to minimize overlaps and intersections’. Five cartographers

identified intersections and overlaps they would improve. Cartographer 1: 'On Map 1, for example, the line from Athens to London is partly covered by the line from London to Paris. There are still a few situations like that'. Cartographer 2: 'Graphically, there is a problem around Nice [in Map 5, [Figure 19](#)]: lines are overlapping, and they are changing "curvature direction"'. Cartographer 3: 'Some potential for improvement here. Where thinner and thicker lines overlap it can be difficult to see the paths of the thinner lines. [...] I would also try to minimize all overlaps – sometimes the flow lines can take a slightly longer path to avoid crossing each other'. Cartographer 4: '[Map 6 ([Figure 12](#) center)] seems to have a couple of links which follow sibling links in parallel for some distance, for example, Germany–Austria and Germany–Turkey. I think it would be nice if a one-pixel line could be drawn to separate these parallel bands'. Cartographer 5: 'Lines twisting around each other (e.g. on Map 3 [[Figure 17](#)] the lines from USA to Central Asia and East Asia intersect each other over Europe)'. One cartographer suggested distributing intersections ("junctions") more evenly: 'some of the junctions could be clearer – perhaps either get lines to cross exactly at the same zone or space out the junctions more'.

**Arrowheads:** Overall, the cartographers liked the arrowhead design. Two cartographers criticized overlaps between arrowheads and flows, and one suggested tapered lines: 'Arrow design (map 1 [[Figure 15](#)]) generally OK, but you get a funky effect in Sweden and Italy where lots of them collide. I might look at a simple taper to a point, but that not communicate direction as well'. One cartographer commented positively on the size of arrowheads, which were scaled non-linearly, and two recommended increasing the size of large arrowheads. One cartographer suggested adding empty space between nodes and arrowheads: 'The arrows seem to come a little closer to the circles than I'd recommend. Instead of touching them exactly, [you] might want to push them back a short distance'.

**Other suggestions:** Two cartographers expressed interest in seeing maps with denser and more complex flow patterns. Two cartographers suggested using visual methods such as transparency and breaking lines to clarify intersections. One cartographer suggested varying color and transparency: 'Many of the problems of overlapping or coalescing could be dealt with by simply using color or changing the transparency of the line symbols so they visually disentangle a little'. One cartographer suggested the use of branching lines. Two cartographers noted it was unclear on some maps whether flows were between cities or countries because point symbols typically communicate point locations: 'For true point-to-point data (flights in particular), the whole thing works well. The circles do really communicate as single points, so where they are meant to indicate whole countries or regions, the net result is less clear'.

In **summary**, the nine professional cartographers generally liked the curvature of flows. The automated method could be extended to avoid straight flows by curving all flows, a suggestion offered by a few cartographers; however, there is no indication from user studies that curving all flows would increase readability. Cartographers suggested reducing the number of intersections, which we addressed after the survey by developing the method described in [Section 3.2.2](#) for reducing intersections of flows connected to a shared node. This addition reduced the number of acute-angle intersections considerably ([Figure 12](#) center and right). Another post-survey addition was treating arrowheads as obstacles that should not be crossed by flows ([Section 3.2.3](#)). This addressed comments by two cartographers.

## 5. Conclusion

Information in flow maps is often dense and visual clutter is difficult to resolve because nodes can only be moved within small geographical limits. We introduce a force-directed method for creating non-branching origin-destination flow maps that take cartographic design principles into account. Our method can be applied to a variety of small- and medium-sized flow data sets. Computation times are modest. For example, the creation of an origin-destination flow map with up to 200 flows and 40 nodes requires 3.5 s with our single-threaded Java implementation using a 2.3 MHz Intel Core i7 CPU. It is likely that limiting the inverse distance weighting to local neighborhoods and using a spatial index for spatial queries could accelerate the algorithm. Our method is fast enough to implement in web maps for automated on-the-fly flow mapping in web browsers, as demonstrated by Stephen and Jenny (submitted).

Feedback from nine professional cartographers is very positive or positive. Most cartographers identified various design aspects that could be improved. Three cartographers suggested extending the automated method to avoid straight flows and curve all flows. Improvements to the angular distribution of flows around nodes and improvements to overlaps and intersections are also suggested.

Our method could be extended to handle more complex flow map data. For example, our method does not handle branching or merging flows. Our method currently requires trial and error to find a parameter combination that will result in a map where design principles are applied appropriately. Further research could integrate aesthetic metrics to automate the selection of suitable parameters.

Our research focused on the optimization of the geometric layout of non-branching origin-destination flows. While the resulting layouts are satisfactory for flow data sets of small and medium size, it is often impossible to find a layout without intersecting or overlapping flows for large, dense, or complex flow maps. The readability of flow maps can be further improved by applying other design options, such as varying visual variables (color, transparency), animating flow lines, or exploring flows with interactive tools.

## Acknowledgments

The authors thank the anonymous reviewers for their comments, and Nat Case (Incase LLC), Ken Field (Esri Inc.), Daniel Huffman (University of Wisconsin Madison), Alex Kent (Canterbury Christ Church University), Charles Preppernau (National Geographic), René Sieber (ETH Zurich), Terry A. Slocum (University of Kansas), Alex Tait (International Mapping, National Geographic), Hans van der Maarel (Red Geographics) for providing feedback for the expert evaluation.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Funding

This work was supported by grant 1438417 from the US National Science Foundation NSF.

## ORCID

Bernhard Jenny  <http://orcid.org/0000-0001-6101-6100>  
 Daniel M. Stephen  <http://orcid.org/0000-0001-9106-5130>  
 Ian Muehlenhaus  <http://orcid.org/0000-0001-7016-1238>  
 Ritesh Sharma  <http://orcid.org/0000-0003-1160-3918>

## References

- Atlas de France, 2000. *Transports et énergie*. Montpellier: RECLUS.
- Baudirektion Kanton Zürich, 2003. *Raumbewachung Kanton Zürich: Verkehrsentwicklung*, vol. 23. Zürich: ARV, Amt für Raumordnung und Vermessung.
- Board, C., Davies, R.J., and Fair, T., 1970. The structure of the South African space economy: an integrated approach. *Regional Studies*, 4 (3), 367–392.
- Brandes, U., Shubina, G., and Tamassia, R., Improving angular resolution in visualizations of geographic networks. *Data Visualization 2000: Joint Eurographics and IEEE TCVG Symposium on Visualization*, 2000 Amsterdam, NL, 23–32.
- Brandes, U. and Wagner, D., 2000. Using graph layout to visualize train interconnection data. *Journal of Graph Algorithms and Applications*, 4 (3), 135–155. doi:10.7155/jgaa.00028
- Debiasi, A., Simões, B., and De Amicis, R., 2014a. Force directed flow map layout. *Information Visualization Theory and Applications (IVAPP), 2014 International Conference on*. 170–177.
- Debiasi, A., Simões, B., and De Amicis, R., Supervised force directed algorithm for the generation of flow maps. *WSCG 2014 22nd International Conference on Computer Graphics, Visualization and Computer Vision 2014b* Plzen, Czech Republic.
- Dent, B., Hodler, T., and Torguson, J., 2009. *Cartography: thematic map design*. New York: McGraw-Hill Education.
- Evatt, B., et al., FLOWMAP: an interactive graphic mapping program for displaying patient origin-destination patterns in space and time. *Fifth Annual Symposium on Computer Application in Medical Care* November 1–4 1981 Washington, D.C., 1072–1077.
- Fink, M. 2013. Drawing metro maps using Bézier curves. In: W. Didimo, et al. eds. *Graph drawing: 20th international symposium, GD 2012, Redmond, WA, USA, 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 463–474.
- Finkel, B. and Tamassia, R., 2005. Curvilinear graph drawing using the force-directed method. In: J. Pach, ed. *Graph drawing: 12th international symposium, GD 2004, New York, NY, USA, 2004*. Berlin, Heidelberg: Springer, 448–453.
- Gerlt, B., 2013. *Distributive flow maps – more raster, more faster* [online]. Available from: <https://blogs.esri.com/esri/apl/2013/08/26/flow-map-version-2/>.
- Guo, D., 2009. Flow mapping and multivariate visualization of large spatial interaction data. *IEEE Transactions on Visualization and Computer Graphics*, 15 (6), 1041–1048. doi:10.1109/TVCG.2009.143
- Guo, D., et al., WMS-based flow mapping services. *Services (SERVICES), 2012 IEEE Eighth World Congress on, 2012*, 234–241.
- Guo, D. and Zhu, X., 2014. Origin-destination flow data smoothing and mapping. *IEEE Transactions on Visualization and Computer Graphics*, 20 (12), 2043–2052. doi:10.1109/TVCG.2014.2346271
- Ho, Q., et al., 2011. Implementation of a flow map demonstrator for analyzing commuting and migration flow statistics data. *Procedia - Social and Behavioral Sciences*, 21, 157–166. doi:10.1016/j.sbspro.2011.07.029
- Huang, W., Using eye tracking to investigate graph layout effects. *APVIS '07 6th International Asia-Pacific Symposium on Visualization, 2007* Sydney, 97–100.
- Huang, W., Eades, P., and Hong, S.-H., 2014. Larger crossing angles make graphs easier to read. *Journal of Visual Languages & Computing*, 25 (4), 452–465. doi:10.1016/j.jvlc.2014.03.001
- Huang, W., Hong, S.-H., and Eades, P., Effects of crossing angles. *Visualization Symposium, PacificVIS '08. IEEE Pacific, 2008*, 41–46.

- Imhof, E., 1972. *Thematische Kartographie*. Berlin: Walter de Gruyter.
- Jenny, B., et al., 2016. Design principles for origin-destination flow maps. *Cartography and Geographic Information Science*, 1–15. doi:10.1080/15230406.2016.1262280
- Kadmon, N., 1971. KOMPLOTT “Do-it-yourself” computer cartography. *The Cartographic Journal*, 8 (2), 139–144. doi:10.1179/caj.1971.8.2.139
- Kern, R. and Rushton, G., 1969. MAPIT: A computer program for production of flow maps, dot maps and graduated symbol maps. *The Cartographic Journal*, 6 (2), 131–137.
- Kim, K., et al., 2012. Developing a flow mapping module in a GIS environment. *The Cartographic Journal*, 49 (2), 164–175.
- Kobourov, S.G., 2012. Spring embedders and force directed graph drawing algorithms. arXiv.org, 1201.3011. [online].
- Nocaj, A. and Brandes, U., 2013. Stub bundling and confluent spirals for geographic networks. *Lecture Notes in Computer Science*, 8242, 388–399.
- Phan, D., et al., 2005. Flow map layout. *INFOVIS '05 Proceedings of the 2005 IEEE Symposium on Information Visualization*. 219–224.
- Purchase, H.C., Cohen, R.F., and James, M., 1996. Validating graph drawing aesthetics. In *Graph Drawing, GD 1995*, vol. 1027. Berlin, Heidelberg: Springer, 435–446.
- Rae, A., 2011. Flow-data analysis with geographical information systems: a visual approach. *Environment and Planning B: Planning and Design*, 38 (5), 776–794.
- Riche, N.H., et al., Exploring the design space of interactive link curvature in network diagrams. *AVI '12 International Working Conference on Advanced Visual Interfaces*, 2012, 506–513.
- Roxburgh, C., et al., 2009. *Global capital markets: entering a new era*. McKinsey Global Institute.
- Shepard, D., A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 23rd ACM National Conference*, 1968, 517–524.
- Slocum, T.A., et al., 2009. *Thematic cartography and geovisualization*. Upper Saddle River, NJ: Pearson/Prentice-Hall.
- Stephen, D.M. and Jenny, B. submitted. Automated layout of origin-destination flow maps: U.S. county-to-county migration 2009–2013. *Journal of Maps*.
- Telegeography Inc., 2000. *Telecommunications [map]*. Telegeography, Inc.
- Tobler, W.R., 1981. A model of geographical movement. *Geographical Analysis*, 13 (1), 1–20.
- Tobler, W.R., 1987. Experiments in migration mapping by computer. *The American Cartographer*, 14 (2), 155–163. doi:10.1559/152304087783875273
- Verbeek, K., Buchin, K., and Speckmann, B., 2011. Flow map layout via spiral trees. *IEEE Transactions on Visualization and Computer Graphics*, 17 (12), 2536–2544.
- Ware, C., et al., 2002. Cognitive measurements of graph aesthetics. *Information Visualization*, 1 (2), 103–110.
- Wittick, R.I., 1976. A computer system for mapping and analyzing transportation networks. *Southeastern Geographer*, 16 (1), 74–81.
- Wong, N., Carpendale, S., and Greenberg, S., EdgeLens: an interactive method for managing edge congestion in graphs. *Ninth Annual IEEE Conference on Information Visualization, INFOVIS'03, 2003* Seattle, Washington, 51–58.
- Wood, J., Slingsby, A., and Dykes, J., 2011. Visualizing the dynamics of London’s bicycle-hire scheme. *Cartographica*, 46 (4), 239–251. doi:10.3138/carto.46.4.239
- Xu, K., et al., 2012. A user study on curved edges in graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18 (12), 2449–2456. doi:10.1109/TVCG.2012.189
- Zhou, H., et al., 2013. Edge bundling in information visualization. *Tsinghua Science and Technology*, 18 (2), 145–156.
- Zhu, X. and Guo, D., 2014. Mapping large spatial flow data with hierarchical clustering. *Transactions in GIS*, 18 (3), 421–435. doi:10.1111/tgis.2014.18.issue-3