Eurographics Conference on Visualization (EuroVis) 2016
K.-L. Ma, G. Santucci, and J. van Wijk
(Guest Editors)

Volume 35 (2016), Number 3

Composite Flow Maps

D. Cornel[1], A. Konev[1], B. Sadransky[1], Z. Horváth[1,2], A. Brambilla[3], I. Viola[2], J. Waser[1]

[1]VRVis Research Center, Vienna, Austria [2]Vienna University of Technology, Vienna, Austria [3]University of Bergen, Bergen, Norway

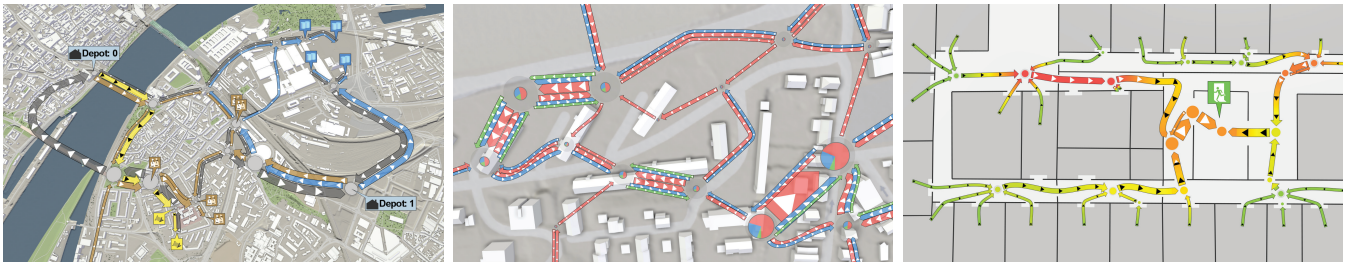

Figure 1: *Quantitative visualization of material movement for several flow components at once. (Left) Different materials in a logistics delivery process. (Middle) Uncertainty in water transportation for a storm water simulation ("at least", "expected", "worst case"). (Right) Flow map for crowd movement in an evacuation scenario, colored by the time of stay.*

Abstract

Flow maps are widely used to provide an overview of geospatial transportation data. Existing solutions lack the support for the interactive exploration of multiple flow components at once. Flow components are given by different materials being transported, different flow directions, or by the need for comparing alternative scenarios. In this paper, we combine flows as individual ribbons in one composite flow map. The presented approach can handle an arbitrary number of sources and sinks. To avoid visual clutter, we simplify our flow maps based on a force-driven algorithm, accounting for restrictions with respect to application semantics. The goal is to preserve important characteristics of the geospatial context. This feature also enables us to highlight relevant spatial information on top of the flow map such as traffic conditions or accessibility. The flow map is computed on the basis of flows between zones. We describe a method for auto-deriving zones from geospatial data according to application requirements. We demonstrate the method in real-world applications, including transportation logistics, evacuation procedures, and water simulation. Our results are evaluated with experts from corresponding fields.

1. Introduction

Flow maps have long been used to illustrate the movement of objects between locations. One prominent application of flow maps is the migration of population between multiple geographic regions. Many fields where users have to explore massive movement data can benefit from using flow map techniques. For logistics, delivery flows between sources and destinations are of interest. Modern architectural planning requires considering possible evacuation scenarios, where flow maps can represent people heading for emergency exits. In flood management, the flows of flood or storm water are important as they may carry potentially dangerous debris. In all these cases, for effective decision support, interactive exploration of multiple flow components at once is needed. The flow components may be given by two opposite transportation directions, alternative scenarios in consideration (e.g., "at least", "worst case", "expected"), different kinds of materials being transported, or people evacuating from specific rooms. At different stages of scenario assessment, the decision maker requires either an overview of general flow trends or a detailed representation of local features. We suggest that such levels of detail should be driven by the application semantics. While uninteresting local flows may be merged to reduce clutter, important details should still be preserved and possibly highlighted using additional overlay visualization.

In this paper, we propose a technique for the automatic generation of flow maps from large movement data. Multiple flow components are combined in one visualization by means of ribbons representing different materials, directions, or flows related to particular origins or destinations. Alternatively, such composite flow maps can display information from different scenarios. Our technique is based on splitting the domain into multiple zones, where the zones

are derived from application semantics and the geospatial context. After this zonation, we compute the flows between adjacent zones. From these, the flow map is generated. The presented technique enables the visualization of composite flows between an arbitrary number of zones in both directions. Additionally, our flow maps support varying levels of detail driven by the geospatial semantics inherent in the application. Irrelevant local features can be generalized, whereas important details are preserved and possibly highlighted.

We demonstrate and evaluate our technique in real-world applications. The logistic application (Figure 1, left) considers planning of delivery routes in an urban area. In the field of flood management (Figure 1, center), we address the surface water movement and interaction with the sewer network in an urban area at the time of a heavy rain (storm water event). The evacuation application considers evacuation scenarios for an office space (Figure 1, right).

We summarize our main contributions as follows:

- Composite flow layout where components are different directions, materials, scenarios, origins, or destinations
- Context-aware and importance-driven levels of detail
- Highlight of problematic areas with overlay visualization of spatial data crucial to the application
- Semantic zonation for flow computation

2. Related Work

A *flow diagram* is a visual representation of the flow of a quantity or the succession of events in a system [Har96]. It consists of nodes connected by arrows, of which width denotes the magnitude of the flow from one node to another. For example, a flow diagram can be used to depict the flow of data through a processing pipeline or the transfer of energy between components of a mechanical system. The data structure underlying a flow diagram is usually a weighted graph. An abstract visualization of a dense graph can be produced by *edge bundling* [Hol06]. Cui et al. [CZQ*08] group edges based on a control mesh. Voronoi diagrams and quadtrees can be combined to generate an adaptive grid for routing edges [LBA10, EHP*11]. Other approaches treat the two-dimensional domain as an image [HET12, HET13, PHT15]. A density-based representation of the graph [DV10, SWVdW*11] is produced via Kernel Density Estimation [Sil86]. The bundling is obtained by iteratively shifting edges towards local maxima of the density map. In a physics-based approach, two edges attract each other if they are sufficiently similar [HVW09].

When the nodes of a flow diagram represent geographical locations, the term *flow map* is often encountered. Already in the 18th century, C.J. Minard adopted hand-drawn flow maps to illustrate, e.g., emigration across continents or Napoleon's Russian Campaign [TGM83]. In these illustrations, Minard carefully chose colors, layouts and compositing rules to depict multiple flows simultaneously.

The flow between two locations can be easily depicted by a straight arrow [Tob87]. But occlusions of arrows increase as the number of locations grows. Edge bundling methods cannot be directly employed for generating flow maps, since quantities are neither accurately represented nor conserved. Phan et al. [PXY*05] address this by applying hierarchical clustering to locations to group arrows and avoid overlaps. Buchin et al. [BSV11b, BSV11a] optimize a set of quality requirements such as avoiding edge crossing and minimizing the length of arrow segments. The diagram can include multiple tree structures, each showing the flow from one source to several destinations. Guo [Guo09] combines flow maps with other techniques to visualize multivariate data. Andrienko et al. [AA11, AABW12] subdivide the spatial domain according to a Voronoi diagram derived from input data. Straight arrows are used to depict the flow between adjacent cells. Multiple flows within the same flow map are considered, but no dedicated strategy is adopted for avoiding clutter and overlaps. A recent survey [AAB*13] provides more detail on the visual analysis of movement data. Debiasi et al. [DSD14] present a supervised, force-based technique for generating flow maps from one root node to several leaf nodes. Flows can be aggregated while avoiding overlaps of edges with other nodes and edges. We adopt this work as a starting point for our flow map layout algorithm.

3. Flow Map Generation Pipeline

In this section, we give a brief overview of the flow map generation pipeline. The table in Figure 2 summarizes the steps. As a prerequisite, we assume that the input datasets are given. These are the actual material movement data as well as the corresponding geospatial datasets, available from public GIS servers or the competent authorities.

In the first step (Figure 2a), we derive context-aware zones within the domain of interest from the input datasets and the semantic information contained in them. Depending on the application, zones can be one- or two-dimensional (Figure 2a, rows 1 and 2, respectively). We then compute *border flows* between all adjacent zones (Figure 2b). Border flows are the aggregated material quantities transported from one zone to another over the common interface in the considered time range. Two zones are considered adjacent if their geometric representations share an endpoint in the one-dimensional case or a polygonal chain in the two-dimensional case. As a result, zones and their connectivity are represented by an abstract zone graph with vertices corresponding to zones and edges carrying the zone adjacency information (Figure 2c). The border flows are attributed to these edges and comprise the components for each relevant material in both directions separately. Finally, the spatial embedding of the zone graph is computed (Figure 2d), where the zone graph vertices are mapped to spatial locations. The graph is then simplified according to the context-based level of detail by iteratively merging the vertices and aggregating the corresponding flow values. The flow component ribbons are generated according to the border flow values and are spatially arranged to avoid overlaps. Additional overlay visualization may be added to highlight important information on top of the ribbons.

For a concrete example of the flow map generation, the logistics application is considered (Figure 2, first row). First, a zone representation of the street network is created by splitting up the street lines at all intersections (a1). Border flows between connected street zones are then determined by counting the number of trucks moving from one street to the next one (b1). The movement data are time-dependent, so movements between streets are accu-

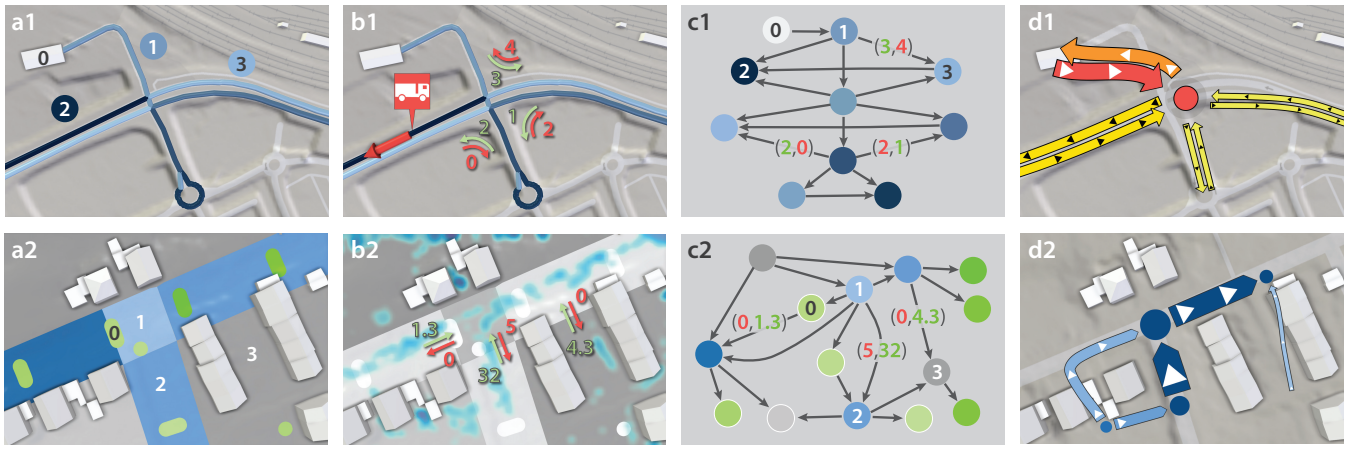

Figure 2: *Flow map generation pipeline for 1D (top row) and 2D (bottom row) flow data. (a) Semantic-based zonation. (b) Bidirectional border flows between example zones (red, green) are derived from simulation data. (c) Abstract, directed acyclic zone graph. (d) Resulting flow map visualization.*

mulated over all time steps up to the time step of interest. From the zones and border flows, a directed acyclic graph is derived (c1). Zones are represented as vertices and are connected to adjacent street zones with weighted edges, where the weights are the border flows. For example, a border flow of (3, 4) between vertices 1 and 3 means that three trucks moved from street 1 to street 3, while four trucks moved from street 3 to street 1. The abstract zone graph (c1) serves as the common representation of the flow in all applications, from which the geospatial visualization (d1) is derived.

4. Data Preparation

This section gives a more detailed description of the steps *a-c* of Figure 2. These steps are dedicated to data preprocessing before the actual flow map creation. Step *d* is thoroughly described in a later section.

4.1. Semantic-Based Zonation

With all input datasets available, the automatic zonation is performed (Figure 2a). Multiple connected zones are derived in the domain in order to compute border flows between them. In contrast to other related efforts [AA11], we do not use a tessellation of the domain based on the analysis of the material movement data, but rather derive meaningful zones from the geospatial context. We argue that arbitrary zonation might hide important information from the user while exposing unimportant details. For example, for planning sewer gullies, it is crucial to understand how much water is consumed or emitted by each gully, whereas water flows between different parts of the same street are irrelevant. For indoor evacuation planning, the important transitions are made from rooms through doors to corridors, with doors being potential bottlenecks. These two examples suggest to tailor the zonation to particular application cases, e.g., to distinguish street zones from sewer zones for the first example, and room zones from door zones and corridor zones for the second one.

Depending on the application, the required zonation can be in one or two dimensions. In case of overland logistic deliveries, it is reasonable to assume that materials or goods are transported through streets only. Therefore, it is sufficient to decompose the street network into one-dimensional zones, where each zone is a line representing a street (Figure 2, a2). Implementation-wise, such zonation can be obtained from a set of lines describing the street network in the domain of interest or, as an optimization, only the subset corresponding to the streets utilized for actual deliveries. The lines are split at their intersection points and duplicates are removed. In the resulting set of unique, connected line segments, each element constitutes a one-dimensional zone (Figure 2, a1).

Crowd or water can move freely across the surface. In such cases, two-dimensional zones are required that fully cover the domain (Figure 2, a1), taking into account shape data describing the domain. For large-scale urban scenarios, these can be, e.g., street shape lines, land use polygons, or sewer locations. In the water-related example, three types of zones are needed (Figure 2, a2). These are street zones (blue), sewer zones (green) and block zones (grey). The zonation algorithm starts by splitting all street lines at their intersection points and removing duplicates. For each of the resulting unique, connected street line segments, a spatial buffer with an empirically determined radius of 9.9 m is created around it. Similarly, circular buffers with a radius of 2 m are created for each point describing a sewer gully location. The outlines of all buffers are split at their intersection points. Each zone in the domain is marked out by a unique set of connected segments of spatial buffer lines. Identifying these can be reduced to the problem of finding all faces of a planar graph. We interpret each endpoint of each segment as a graph vertex. For any two vertices of the graph, there is an edge connecting them if and only if there exists a line segment with the two corresponding endpoints. Since all segment intersections are already eliminated, the resulting graph is guaranteed to be planar. The planar faces of this graph exactly correspond to the desired zones. For the planar faces computation, *The Boost Graph Library* [Boo] is used. From the faces, the actual geospa-

tial zones can be easily reconstructed and labeled as street zones (inside street line buffers), sewer zones (inside the sewer buffers) and block zones (the other zones). For indoor applications such as evacuation, wall lines and door locations are needed, as well as semantic annotations for rooms and other areas. The actual zonation process is similar to the one described above.

4.2. Zone Graph and Border Flows

The actual flow map is generated from border flows, which are computed using the zonation and the material movement data. Border flows are the material quantities transported between each two adjacent zones through their common interface in the considered time frame. Since the transported materials can be either continuous or discrete, we distinguish two cases. In case of a continuous medium (e.g., water), the quantity is represented by the volumetric flow rate of the transported medium. For water movement, this value is equivalent to the hydrological discharge between the zones (Figure 2, b2), which we compute numerically. At each time step, the value is multiplied with the time step size to calculate the actual material volume transported in that time step. For each zone-to-zone interface, such quantities are accumulated over all time steps to obtain the overall material quantities. In case of movement of discrete entities (e.g., people or material containers), the accumulated quantity is simply the number of entities transported (Figure 2, b1). Note that the border flows include components for every material (or scenario) under consideration in both directions separately.

For the computed zonation, a *zone graph* describing the connectivity is created. It is a directed acyclic graph where each vertex corresponds to a zone. For each pair of vertices in this graph, there exists a directed edge connecting them if and only if the two corresponding zones are adjacent. In the computed zone graph, every edge needs to be attributed with the corresponding pair of border flow values (Figure 2c). The first value indicates the material quantity transported along the directed edge, while the second value indicates the material quantity transported in the opposite direction. From this follows that the direction of an edge in the graph is independent from the flow direction. The directed and acyclic properties of the zone graph merely simplify its traversal, but the flow directions follow from the border flow values.

5. Spatial Embedding of the Zone Graph

In the previous section, we described a way to generically express material movement in the domain of interest via border flows between adjacent zones. The zone connectivity is represented as an abstract zone graph with border flow values attributed to the edges of this graph. To obtain the geospatial flow map (Figure 2d), vertices and edges of the graph have to be embedded into the spatial domain. This is a multi-step process shown in Figure 3. The spatial embedding of the abstract zone graph relies on *zone representations*, which are geometric characterizations of the zones. If a zone originates from a line, e.g., a street zone, then the representation is the actual street line from which the zone was initially derived. Otherwise, the zone representation is a point set, which may contain one point for smaller zones or several points for large or complex zones.

5.1. Geospatial Graph Creation

For each vertex of the abstract zone graph (Figure 3a), there exists a corresponding spatial zone representation, which is either a line or a point set (Figure 3b). The zone representations are initially disconnected and independent from each other. They need to be connected according to the connectivity information of the zone graph (Figure 3c). Two zones represented by adjacent lines can simply be joined at their endpoints. If one or both of the zone representations are point sets, the two closest points of them should be connected. The resulting set of connected lines is resampled with an application-specific step size (logistics: 100 m, water: 10 m, evacuation: 7 m) to unify the density of line points in preparation for a subsequent vertex merging step. Each point of every resampled line is interpreted as a vertex of a new *geospatial graph* (Figure 3d). Connections between the points are interpreted as edges, of which each inherits the flow values from the corresponding edge of the zone graph. The resulting geospatial graph (Figure 3d) has generally more vertices and edges than the original abstract graph as all spatial points of the zone representations are now individual graph vertices. If all components of the flow value assigned to an edge are zero, the edge is removed. Potentially disconnected graphs resulting from this removal do not require any special treatment.

5.2. Force-Driven Semantic Levels of Detail

The geospatial graph created in the previous step has the highest possible level of detail, and the input flow data is replicated multiple times. A semantic-driven simplification is applied to the geospatial graph so that distinct flows can be identified more easily. This is done by using our iterative force-driven merge algorithm (Figure 3e) inspired by the work of Debiasi et al. [DSD14]. We introduce attraction and repulsion forces between the geospatial graph vertices. At each iteration, we move the vertices according to these forces. Vertices within a given radius r are then merged to a new vertex at their centroid. The algorithm terminates when the forces between the vertices reach an equilibrium or if a specified maximum number of iterations is exceeded.

A main difference of our algorithm to the one by Debiasi et al. is that we do not assume a distinct flow direction from a single root vertex to multiple leaf vertices. Instead, flow can occur in both directions between any adjacent zones, so there is no distinction between source and sink vertices. Debiasi et al. also rely on a parent-child relationship between vertices for the calculation of forces, therefore implicitly ranking their importance. With bidirectional flows, such hierarchies are not given. Instead, we only use a flag for important vertices that should never be moved or merged, such as material depots in a logistics application.

In our method, an attraction force $\vec{F}_a(v)$ pulls each vertex v from the set of all vertices V towards all other vertices, weighted by their flow magnitudes:

$$\vec{F}_a(v) = \sum_{v_i \in V} \left(\frac{i(v_i) + o(v_i)}{i(v_i) + o(v_i) + i(v) + o(v)} \cdot \frac{\vec{p}(v_i) - \vec{p}(v)}{\|\vec{p}(v_i) - \vec{p}(v)\|^2} \right),$$

where $i(v)$ and $o(v)$ denote the summed (over all incident edges and all flow components) inflow and outflow magnitudes of v, respectively, and $\vec{p}(v)$ is the position of v. A stress force $\vec{F}_s(v)$ moves

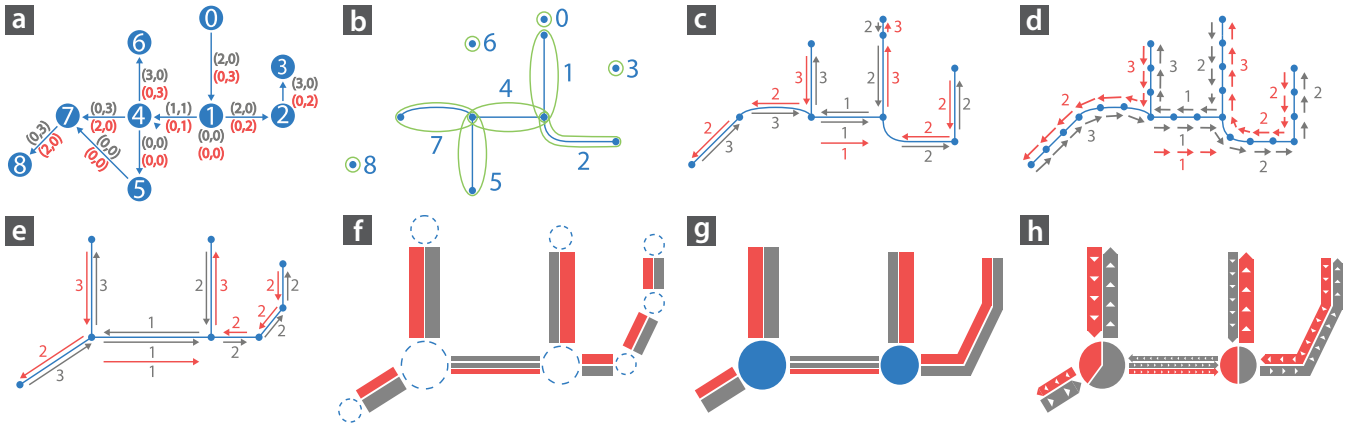

Figure 3: *Overview of spatial embedding. (a) Input zone graph with two flow components (red, grey). (b) Input zone representations. (c) Connection of relevant zone representations. (d) Resampling of lines and creation of geospatial graph. (e) Iterative vertex merging. (f) Layout of individual flow lines. (g) Removal of some connection circles. (h) Visual styles and rendering.*

each vertex v towards a weighted centroid of the set V_N of vertices adjacent to v:

$$\vec{F}_s(v) = \sum_{v_i \in V_N} \left(\frac{f(v,v_i) + f(v_i,v)}{i(v) + o(v)} \cdot (\vec{p}(v_i) - \vec{p}(v)) \right).$$

Here, $f(v,v_i)$ denotes the summed magnitude of all flow components directed from v to v_i. Additionally, if the set of important vertices $V_I \subseteq V$ is not empty, we apply a repulsion force $\vec{F}_r(v)$ to all other vertices $v \in V \setminus V_I$ to segregate the important vertices from all other vertices in the flow map:

$$\vec{F}_r(v) = \sum_{v_i \in V_I} \frac{\vec{p}(v) - \vec{p}(v_i)}{\|\vec{p}(v) - \vec{p}(v_i)\|^2}$$

All these forces are applied to each vertex $v \in V \setminus V_I$ via a displacement vector $\vec{D}(v)$ added to the position of v:

$$\vec{D}(v) = k_a \cdot \vec{F}_a(v) + k_s \cdot \vec{F}_s(v) + k_r \cdot \vec{F}_r(v),$$

where $k_a, k_s, k_r \in [0,1]$ are weights for the forces to control the convergence of the algorithm. After displacement, all vertices closer to each other than a radius r are merged into a new vertex placed at the centroid of their locations.

The second major difference to the work by Debiasi et al. is that neither the force weights k_a, k_s, k_r nor the merge radius r are constant. Instead, only a range is defined for these parameters, but the actual values within the range are location-dependent. The ranges have been found empirically for the different applications (logistics: $r \in [70,1400]$ m, $k_a \in [0.1,1]$, $k_s \in [0,0.05]$, $k_r \in [0,1]$, water: $r \in [1,100]$ m, $k_a = k_s = k_r = 0$, evacuation: $r = 10$ m, $k_a = k_s = k_r = 0$). The values at point $\vec{p}$ in the spatial domain are determined by blending the maximum and minimum of the corresponding ranges with a blend weight, which is the level of detail $\lambda(\vec{p}) \in [0,1]$. A higher $\lambda(\vec{p})$ leads to smaller forces and to less merging of vertices. This preserves details of the zone representations (e.g., of street lines) and thus leads to a stronger correspondence of the resulting flow map with the geospatial context. A lower $\lambda(\vec{p})$ leads to a more generalized flow map in which border flows close together are merged.

We choose $\lambda(\vec{p}) = \max(\lambda_s(\vec{p}), \lambda_v(\vec{p}))$, where $\lambda_s(\vec{p})$ and $\lambda_v(\vec{p})$ are the semantic and view-dependent levels of detail, respectively. Namely, $\lambda_s(\vec{p})$ assigns an importance value in $[0,1]$ to each point $\vec{p}$ in the spatial domain. We use the semantic level of detail to reduce the visual complexity in context regions while emphasizing focus regions like the neighborhood of important infrastructure. A view-dependent level of detail $\lambda_v(\vec{p})$ allows for the interactive control of the granularity of the flow map. It is dependent on the distance between the view point and $\vec{p}$, which is normalized to $[0,1]$ using an application-specific maximum distance (logistics: 4000 m, water: 400 m, not used in evacuation). Therefore, an overview flow map is automatically refined as the user zooms in on an area of interest, while keeping the density of flow lines displayed on the screen approximately constant. The use of both semantic and view-dependent level of detail is demonstrated in the accompanying video [com].

5.3. Layout of Composite Flow Lines

The result of the above algorithm is a geospatial graph simplified according to the required level of detail. To visualize this graph, for each edge, lines have to be created for all flow components in both directions. These lines should be visually connected at their end vertices without overlaps. In contrast to existing uni-directional flow map algorithms, we cannot simply join the individual lines and visualize junctions as arborizations without giving a false impression of a principal flow direction. Arborizations would also introduce unavoidable overlaps increasing exponentially with the number of flow components. We solve this by using connection circles at places where multiple adjacent lines connect.

At each vertex position, a circle is created that needs to be large enough to fit the widths of all incident edges on its circumference. Let E_c be the set of all edges to be connected to a circle c. Each edge $e \in E_c$ is essentially a group of individual lines for each separate

flow component and direction. A user-defined transfer function τ is used to compute the width of a line l from its flow magnitude m_l. Since the width w_e of the entire edge is the sum of the widths of all lines $l \in L_e$ belonging to it, the required radius of the circle c should be:

$$r_c = \frac{1}{2\pi} \cdot \sum_{e \in E_c} w_e = \frac{1}{2\pi} \cdot \sum_{e \in E_c} \sum_{l \in L_e} \tau(m_l)$$

In practice, this is not sufficient. If the incoming directions of all edges are approximately the same, half of the edges have to be connected to the part of the circle facing away from them, leading to unnecessary stretching. Doubling r_c solves this problem, but usually leads to unnecessarily large circles. Empirically, we found that scaling r_c by a value between 1.3 to 1.5 gives sufficiently good results.

Let e be an edge with a normalized direction $\vec{d} = \left(d_x, d_y\right)^T$ that needs to be connected to the connection circles c_a and c_b at its start and end, respectively. Then $\alpha = atan2(d_y, d_x)$ is the polar angle of a point on the circumference of c_a to which e should be connected. The polar angle for connecting e to c_b is $\beta = atan2(-d_y, -d_x)$. These angles lead to the shortest line between the two corresponding connection circles. However, this is not generally optimal. As all edges now have their widths, using only direct connections can lead to overlaps for edges with similar directions. To avoid overlaps, a relaxation step is applied to each circle which changes the connection angles of all incident edges to fit them on the circumference. Since the circumference is greater than or equal to the sum of widths of all edges, an arrangement without overlaps is guaranteed to exist. The optimal arrangement is free of overlaps and has the smallest summed difference of new polar angles $\hat{\alpha}$, $\hat{\beta}$ to the original polar angles α, β, i.e., it has the shortest connections possible without overlaps. Solving this optimization problem with the time constraints of an interactive visualization is very challenging. We approximate the solution by generating different connection arrangements and choosing the one with the smallest sum of connection lengths.

The new start point of edge e on the circumference of the connection circle c_a is determined by adding the radius-vector $\vec{v}_a = r_a \cdot (\cos(\hat{\alpha}), \sin(\hat{\alpha}))^T$ to the center of c_a. The respective radius-vector for connection circle c_b of the end point is given by $\vec{v}_b = r_b \cdot \left(\cos(\hat{\beta}), \sin(\hat{\beta})\right)^T$. However, these connection points have been calculated for the edge as a whole, which is now split up into individual lines for the flow components. These lines should be offset relative to the connection point according to their widths. The offsets are computed along the edge normal so that all lines belonging to the same edge remain parallel (Figure 3f). There are different options to group and order the individual lines. For our applications, we found it most useful to group them by flow component or by direction. The normalized radius-vectors $\vec{v}_a$ and $\vec{v}_b$ are used as connection tangents for a Hermite spline approximation of the individual lines. This allows the lines from edges that have been offset significantly in the relaxation step to connect to the circles more smoothly.

The connection circles are only necessary for vertices with three or more incident edges. The complexity of the flow map can be reduced by removing unnecessary circles (Figure 3g). If a vertex has only one incident edge, all individual lines are now extended until the middle of the removed circle. If a vertex has two incident edges and the flow values of both edges are equal, the corresponding individual lines of both edges can simply be connected.

5.4. Rendering

At this point, we have a collection of smooth lines corresponding to individual flow components. For rendering a line, we generate a triangular mesh ribbon from the points of this line. As stated above, the width of the ribbon for a line l is given by the mapped flow magnitude $\tau(m_l)$. An arrow tip is added at the end of each ribbon to visually express the direction of the flow. The arrow tip can either be a simple narrowing of the line width to a single point or an extruded arrow head (Figure 3h). The latter method is used for lines with a screen-space width below a specified threshold.

Coloring is applied to the ribbon, which can either be a solid color distinctive for individual flow components (Figure 1, left) or some value mapped to a color with a user-defined transfer function. A natural choice for this value is the flow magnitude m_l corresponding to the line l (Figure 2d). It is also possible to add new geospatial information to the flow map as an overlay. For example, in applications involving a traffic network, visualizing the traffic conditions or accessibility of roads with color can help to highlight problematic areas. In Figure 1, right, a density measure of the evacuating crowd is shown with the color overlay. For such applications, we evaluate the mapped property for each vertex of the original geospatial graph before applying the iterative vertex merging and then keep track of the values while merging. This allows us to visualize aggregates such as the minimum, average, and maximum road accessibility of each vertex of the merged geospatial graph. On top of the flow ribbons, glyphs are added. We limit ourselves to repeating directional arrows to indicate the flow direction. They are either black or white, depending on which one gives a better contrast to the color of the arrow (Figure 2d). Depending on the application, more complex glyphs can be used.

Finally, circles are rendered at the spatial positions of vertices with three or more incident edges. The radius r_c^m of each rendered circle c should directly correspond to the flow magnitude through the corresponding vertex. It is chosen as

$$r_c^m = \tau\left(\sum_{l \in L_c} m_l\right),$$

where L_c is the set of all lines meeting at this circle, m_l is the magnitude of the flow component represented by the line l, and τ is the mapping introduced above. In general, r_c^m is not equal to the radius r_c of the connection circle, so the connection circle of radius r_c does not properly represent the flow through the corresponding vertex. It is rather a circle to which the lines connect. We show both circles at the spatial position of the vertex, but give the (larger) connection circle just enough opacity to be barely visible (Figure 2, d1). This provides a hint to the user to which vertex an edge connects. The opaque inner circles allow for an accurate comparison of flow magnitudes by size. To these circles, the same coloring as to the ribbons can be applied. Alternatively, a pie chart can be displayed which shows the ratios of different flow components through this

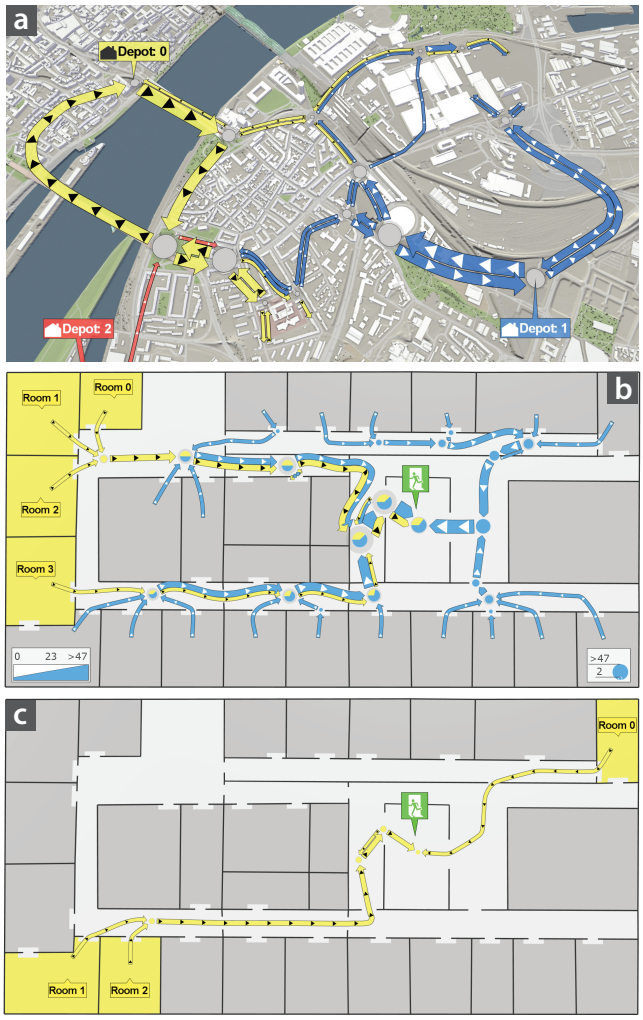

Figure 4: *Flow map composition according to different material origins. (a) Depots. (b) User-defined rooms (yellow) combined with all other rooms (blue). (c) Interactive decomposition according to rooms.*

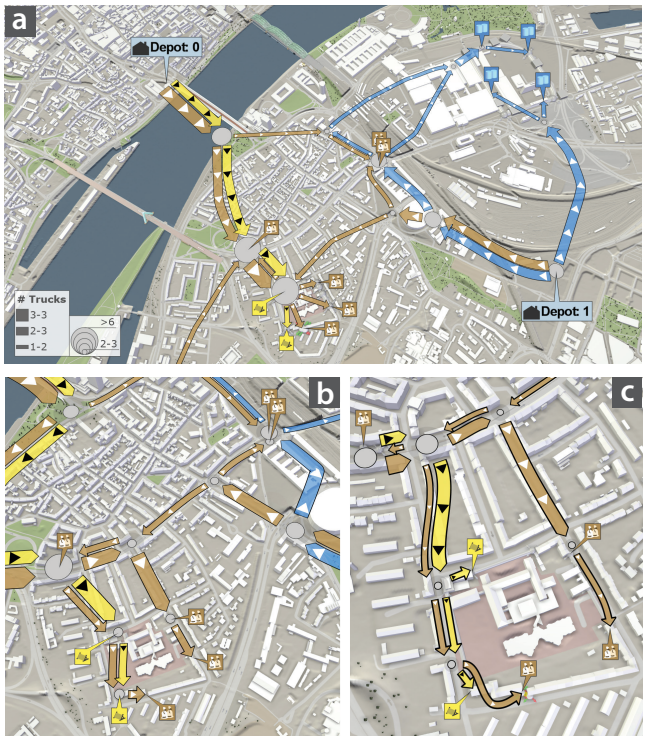

Figure 5: *Zoom-dependent level of detail (LoD). (a) Low. (b) Medium. (c) High.*

vertex (Figure 1, center). These pie charts can also be assigned a predefined maximum capacity to visualize the utilization of important infrastructure, e.g., a material depot in a logistics application.

6. Results and Evaluation

We implemented our technique in the Visdom framework [vis] and demonstrate it in three different applications. The first application is concerned with transportation logistics for constructing flood protection barriers. To plan the routes for reliable material deliveries, an overview of multiple time-dependent processes is needed. The actual delivery data was simulated with a heuristic algorithm [WKS*14] based on routing data automatically obtained from *Google Directions*. The second application deals with the planning of sewer inlets based on data provided by a public GIS service [Ope], the Flood Protection Center of Cologne, Germany, and a shallow-water simulation [HWPa*15]. The sewers must be able to consume water without overflows even under extreme weather events. To effectively place new inlets, the expert needs an understanding of the local flow behavior, including the water exchange of the surface with the sewer network. In the third application, the evacuation from an office space is modeled, where multiple rooms are connected to an emergency exit via shared corridors. Planning of the doors placement and populating rooms requires considering possible evacuation activities. People should be able to quickly leave the area without getting stuck or jammed. The crowd movement is modeled by our in-house simulator. All applications feature basic navigation for view-dependent levels of detail, interactive flow decomposition, and animation of the flow maps over time. For a demonstration of these interactive features and additional results, we refer to the accompanying video [com].

The evaluation consists of two phases conducted before and after the implementation. The involved committee includes a logistics expert; an expert for sewer networks; two consulting engineers for integrated catastrophe management; an expert for hydrology; and an expert for crowd modeling. During a preliminary evaluation, the experts gave feedback on mockups of the planned results and provided us with hints on actual problems in their fields for which composite flow maps could be helpful. For the evaluation of the final results, the experts were shown composite flow maps for the different applications and were asked if they found them expressive, helpful, and aesthetic. In general, the experts concur that our composite flow maps are an effective and aesthetic way to present

Figure 6: *Flow maps for storm water. (a) Overview. Circles indicate the sewer effects (red=outflow, green=inflow). (b) Glyphs show details about the sewer-surface coupling. (c) Increased local LoD nearby important buildings.*

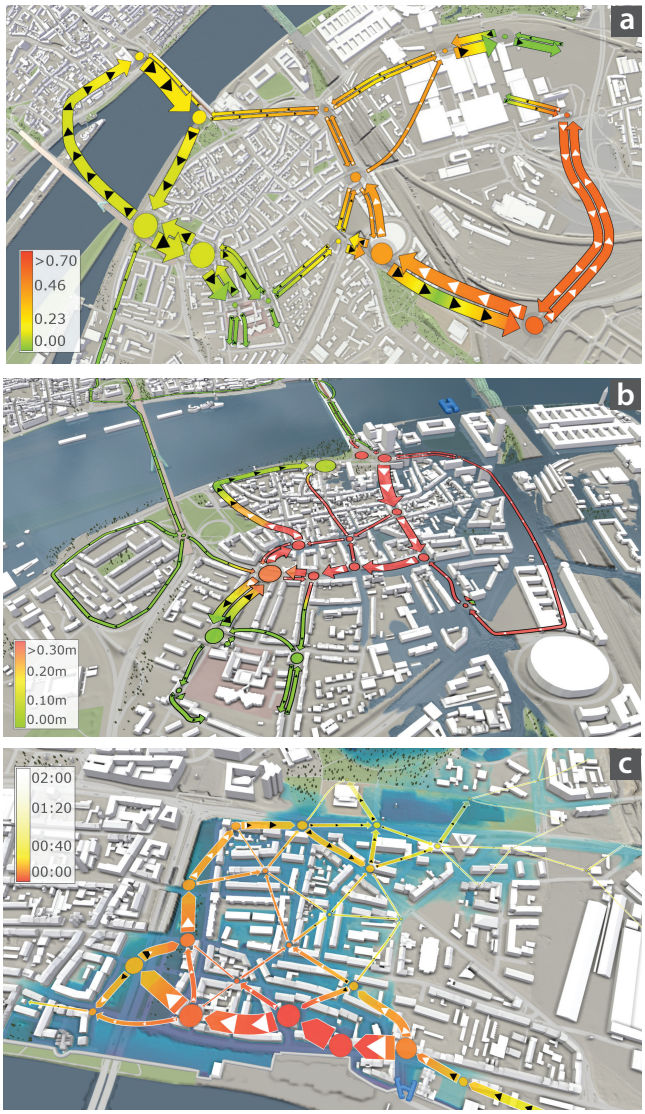

Figure 7: *Overlay visualization of important information. (a) Monitored traffic conditions aggregated over a time span. (b) Road inaccessibility due to water level. (c) Wave arrival times for a flood wall breach [hh:mm].*

different material flows, including quantities, in the geospatial context. In particular, they provide a comprehensible way to get a quick overview of time-dependent flow data for logistics and evacuation applications. We received mixed feedback regarding the storm water application. Visualizing major flows of storm water is a fairly new yet important task for flood management. According to the domain experts, our method is a step in the right direction but needs further improvements.

Composite flow maps in the logistics application consist of separate flow components for the deliveries of different materials (Figure 1, left) and the deliveries from different depots (Figure 4a). For the evacuation, individual rooms can be selected to separate the crowd flow originating from these rooms (Figure 4b) from the rest of the flow. In both applications, the flow maps can be adjusted interactively to visualize only specific flow components (Figure 4c). This was appreciated by the domain experts as a helpful feature for planning. The flow uncertainty visualization over multiple different scenarios (see Figure 1, middle), where the components show the “at least”, “worst case” and “expected” scenarios, was seen as useful only for very specific tasks such as debris transport by flood water.

The zoom-dependent levels of detail were seen as a helpful feature in both the logistics (Figure 5) and the storm water (Figure 6) applications. In Figure 5a, an overview flow map for material de-

liveries is shown. As the user zooms into the hospital area, the flow map is gradually refined (Figure 5b,c). In Figure 6a, an overview flow map for storm water is supplemented with circular glyphs for the sewers that consume (green) or emit (red) significant water amounts. This was welcomed by all domain experts. However, for this case, more generalized flow maps were considered misleading, because the visible relation to the actual zones is lost. This might improve with an enhanced zonation algorithm. On the highest level of detail, we display an arrow glyph representation of the sewer inflows and outflows using the same visual encoding as for the flow map (Figure 6b). This was considered a good visualization of overflows for planning tasks. In Figure 6c, we use a semantic level of detail based on the proximity to pharmacies for keeping the flow representation in the surrounding area more accurate. This was well received by the experts.

We visualize additional geospatial information on top of the flow map if necessary. In the logistics application, these are traffic conditions obtained from a public service [Ope] (Figure 7a) or the inundation of streets (Figure 7b). This highlight of potential transportation bottlenecks based on dynamic data was highly rated for planning applications and characterized as superior to current systems. For evacuation, we display the occupancy of rooms over time (Figure 1, right) to highlight critical sections on the evacuation routes. In Figure 7c, we visualize the wave arrival times in a breach scenario as an overlay. In total, overlay visualizations for all three applications have been highly appreciated. However, our approach to always use connection circles in the flow map layout has received mixed feedback due to possible misinterpretation. Especially for the evacuation scenarios, it is desirable to use arborizations where possible. Displaying pie charts on top of connection circles (Figure 1, center and Figure 4) was considered useful for determining the workload at a particular junction and for easy comparison of the amounts of different flow components. Finally, animating the flow map over time was considered especially helpful for the logistics and evacuation application, as it allows the user to investigate the chronology of events and shows how critical regions emerge in the evacuation scenario.

In summary, our main contribution – the composite flow map – has been highly rated by all six experts, the interactive decomposition of flows was considered useful by five experts. The use of composite flow maps was seen as an improvement for their work in two out of three applications. Levels of detail used in the logistics and water application were appreciated by four out of five experts. All six experts concurred that the overlay visualization of additional data on top of the flow maps was very helpful for common tasks in their fields.

7. Conclusions and Future Work

In this paper, we propose a simplification and visualization technique for material movement data. It fills the void between conventional flow maps for rooted trees and edge-bundling for arbitrary multigraphs. With composite flow maps, multiple flows can be shown in a single visualization. The visualization can be simplified with interactive and locally varying levels of detail while preserving the representative magnitudes of the generalized flows. Using the layout algorithm, visual representations of flow components are arranged in the geospatial domain without overlaps.

Our technique relies on a unified representation of flow data by means of a weighted zone graph which only requires a flow of material in one or two dimensions between zones. The required zonation of the spatial domain can range from a regular subdivision to complex semantic structures. This allows for the generic treatment of flows for different applications and data modalities. We demonstrate the wide applicability of our technique with three different types of flows. Other possible applications include the visualization of CFD data (e.g., wind, gas flows), particle-based simulations, and also non-spatial transportation data such as network traffic.

The evaluation partners from various professional fields emphasize the benefits of our technique for planning in their applications. There is a demand for techniques allowing the user to investigate and directly compare not only entire flows, but also parts of the flows isolated in separate flow components. This is especially true for applications in the fields of logistics and evacuation planning. For the storm water application, conclusive information could not always be drawn from our visualization. The domain experts attribute it to the diffuseness of the original movement data.

Reliable extraction and visualization of principal flows in storm water data remains a challenging task. One possible solution lies in incorporating the data itself into the semantic-based zonation process to align zones with existing flow trends. Another direction of future work is the use of composite flow maps to visualize the uncertain transport of sediments, debris, and pollutants in water. A need for these applications is expressed by the experts. One more goal is the use of arborizations at junctions in the layout algorithm. A hybrid approach using arborizations in unidirectional segments of the flow can further reduce the visual complexity.

8. Acknowledgments

This work was supported by grants from the Vienna Science and Technology Fund (WWTF): ICT12-009 (Scenario Pool) and the Austrian Science Fund (FWF): W1219-N22. The competence center VRVis is funded by BMVIT, BMWFW, and the Vienna Business Agency within the scope of COMET – Competence Centers for Excellent Technologies (Project-Nr. 843272). The program COMET is managed by FFG. We thank the mobility department of the AIT, riocom, and the Stadtentwässerungsbetriebe Köln, AöR.

References

[AA11] ANDRIENKO N., ANDRIENKO G.: Spatial generalization and aggregation of massive movement data. *IEEE Transactions on Visualization and Computer Graphics 17*, 2 (2011), 205–219. 2, 3

[AAB*13] ANDRIENKO G., ANDRIENKO N., BAK P., KEIM D., WROBEL S.: *Visual analytics of movement*. Springer Science & Business Media, 2013. 2

[AABW12] ANDRIENKO G., ANDRIENKO N., BURCH M., WEISKOPF D.: Visual analytics methodology for eye movement studies. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2889–2898. 2

[Boo] The Boost Graph Library. http://www.boost.org/doc/libs/release/libs/graph/ (last visited on April, 11th 2016). 3

[BSV11a] BUCHIN K., SPECKMANN B., VERBEEK K.: Angle-restricted steiner arborescences for flow map layout. In *Algorithms and Computation*. Springer, 2011, pp. 250–259. 2

[BSV11b] BUCHIN K., SPECKMANN B., VERBEEK K.: Flow map layout via spiral trees. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2536–2544. 2

[com] Video: Composite Flow Maps. http://visdom.at/media/videos/mp4/composite_flow_maps.mp4 (last visited on April, 11th 2016. 5, 7

[CZQ*08] CUI W., ZHOU H., QU H., WONG P. C., LI X.: Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1277–1284. 2

[DSD14] DEBIASI A., SIMÕES B., DE AMICIS R.: Supervised force directed algorithm for the generation of flow maps. In *22nd International Conference on Computer Graphics, Visualization and Computer Vision* (2014). 2, 4

[DV10] DEMŠAR U., VIRRANTAUS K.: Space–time density of trajectories: exploring spatio-temporal patterns in movement data. *International Journal of Geographical Information Science 24*, 10 (2010), 1527–1542. 2

[EHP*11] ERSOY O., HURTER C., PAULOVICH F. V., CANTAREIRO G., TELEA A.: Skeleton-based edge bundling for graph visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2364–2373. 2

[Guo09] GUO D.: Flow mapping and multivariate visualization of large spatial interaction data. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1041–1048. 2

[Har96] HARRIS R. L.: *(Information graphics: A comprehensive illustrated reference)*. Oxford University Press, 1996. 2

[HET12] HURTER C., ERSOY O., TELEA A.: Graph bundling by kernel density estimation. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 865–874. 2

[HET13] HURTER C., ERSOY O., TELEA A.: Smooth bundling of large streaming and sequence graphs. In *Visualization Symposium (PacificVis), 2013 IEEE Pacific* (2013), IEEE, pp. 41–48. 2

[Hol06] HOLTEN D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 741–748. 2

[HVW09] HOLTEN D., VAN WIJK J. J.: Force-directed edge bundling for graph visualization. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 983–990. 2

[HWPa*15] HORVÁTH Z., WASER J., PERDIGÃO R. A. P., KONEV A., BLÖSCHL G.: A two-dimensional numerical scheme of dry/wet fronts for the saint-venant system of shallow water equations. *International Journal for Numerical Methods in Fluids 77*, 3 (2015), 159–182. 7

[LBA10] LAMBERT A., BOURQUI R., AUBER D.: Winding roads: Routing edges into bundles. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 853–862. 2

[Ope] Offene Daten Köln. http://www.offenedaten-koeln.de/ (last visited on April, 11th 2016). 7, 9

[PHT15] PEYSAKHOVICH V., HURTER C., TELEA A.: Attribute-driven edge bundling for general graphs with applications in trail analysis. In *Visualization Symposium (PacificVis), 2015 IEEE Pacific* (2015), pp. 39–46. 2

[PXY*05] PHAN D., XIAO L., YEH R., HANRAHAN P., WINOGRAD T.: Flow map layout. In *IEEE Information Visualization (InfoVis)* (2005), pp. 219–224. 2

[Sil86] SILVERMAN B. W.: *Density estimation for statistics and data analysis*, vol. 26. CRC press, 1986. 2

[SWVdW*11] SCHEEPENS R., WILLEMS N., VAN DE WETERING H., ANDRIENKO G., ANDRIENKO N., VAN WIJK J. J.: Composite density maps for multivariate trajectories. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2518–2527. 2

[TGM83] TUFTE E. R., GRAVES-MORRIS P.: *The visual display of quantitative information*, vol. 2. Graphics press Cheshire, CT, 1983. 2

[Tob87] TOBLER W. R.: Experiments in migration mapping by computer. *The American Cartographer 14*, 2 (1987), 155–163. 2

[vis] Visdom - An integrated visualization system. http://visdom.at (last visited on April, 11th 2016). 7

[WKS*14] WASER J., KONEV A., SADRANSKY B., HORVÁTH Z., RIBIČIĆ H., CARNECKY R., KLUDING P., SCHINDLER B.: Many Plans: Multidimensional Ensembles for Visual Decision Support in Flood Management. *Computer Graphics Forum 33*, 3 (2014), 281–290. 7