Eurographics Conference on Visualization (EuroVis) 2020
M. Gleicher, T. Landesberger von Antburg, and I. Viola
(Guest Editors)

Volume 39 (2020), Number 3

Augmenting Node-Link Diagrams with Topographic Attribute Maps

R. Preiner[1], J. Schmidt[2], K. Krösl[2,3], T. Schreck[1], and G. Mistelbauer[4]

[1]Graz University of Technology, Institute of Computer Graphics and Knowledge Visualization, Austria
[2]VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH, Austria
[3]TU Wien, Institute of Visual Computing & Human-Centered Technology, Austria
[4]Otto-von-Guericke University Magdeburg, Department of Simulation and Graphics, Germany

Abstract

We propose a novel visualization technique for graphs that are attributed with scalar data. In many scenarios, these attributes (e.g., birth date in a family network) provide ambient context information for the graph structure, whose consideration is important for different visual graph analysis tasks. Graph attributes are usually conveyed using different visual representations (e.g., color, size, shape) or by reordering the graph structure according to the attribute domain (e.g., timelines). While visual encodings allow graphs to be arranged in a readable layout, assessing contextual *information such as the relative similarities of attributes across the graph is often cumbersome. In contrast, attribute-based graph reordering serves the comparison task of attributes, but typically strongly impairs the readability of the* structural *information given by the graph's topology. In this work, we augment force-directed node-link diagrams with a continuous ambient representation of the attribute context. This way, we provide a consistent overview of the graph's topological structure as well as its attributes, supporting a wide range of graph-related analysis tasks. We resort to an intuitive height field metaphor, illustrated by a topographic map rendering using contour lines and suitable color maps. Contour lines visually connect nodes of similar attribute values, and depict their relative arrangement within the global context. Moreover, our contextual representation supports visualizing attribute value ranges associated with graph nodes (e.g., lifespans in a family network) as trajectories routed through this height field. We discuss how user interaction with both the structural and the contextual information fosters exploratory graph analysis tasks. The effectiveness and versatility of our technique is confirmed in a user study and case studies from various application domains.*

CCS Concepts

• ***Human-centered computing*** → ***Graph drawings;*** *Social networks; Information visualization;*

1. Introduction

Graphs are used in many application domains as a key data structure to model entities and relationships between them. Possible applications include ancestral graphs modeling family histories, call graphs modeling software systems, or network graphs represeting devices in a computer network. We may depict entities and relationships as nodes and links, and include domain-specific characteristics as attributes thereof. Such *attributed* graphs [FP13] can be visualized in various ways [TKE12], with node-link diagrams emphasizing the topological structure and its links.

As the topology of such graphs can be of arbitrary complexity, their nodes and links are often arranged using force-based layouts [FR91], which are able to depict their structure as clearly as possible. There exist many visual designs to incorporate node and edge attributes in an underlying structural visualization, including glyphs for nodes, and varying line thickness or color for edges. However, comparing relative similarities of attributes remains difficult, since it is often based on pair-wise comparison of node overlays depicting the values. The problem is only partially solved by grouping nodes with similar attributes, e.g., along a timeline in ancestral graph visualizations, as these layouts typically deteriorate the perceivability of the overall topological structure. Moreover, comparing different attributes associated with such graphs requires changing the layout based on the attribute of interest. In such cases, a decoupling of graph layout and attribute visualization is desirable.

In this work, we present *topographic attribute maps (TAMs)*, a novel visualization for attributed graphs that achieves such a decoupling by augmenting node-link diagrams by a visual representation of a contextual interpolating attribute landscape. This way, TAMs support graph layouts providing a comprehensive overview of the topological structure, while at the same time providing a visual context allowing for absolute and relative attribute classifications. Specifically, we present a set of suitable techniques for optimizing the landscape height profile for an appropriate visual embedding of the graph, and employ established techniques from topographic map rendering to give an encompassing visualization of node attributes including scalar values and intervals. We demonstrate the versatility of TAMs on graph examples from different application

domains, and report on a user study evaluating the usability of our method using different geospatial interpolation techniques.

2. Related Work

Graph and network visualization is one of the core fields in Information Visualization, and was extensively studied for decades. Their techniques are found in many applications, such as scientific visualization [WT17], social media analysis [CLY17], traffic data analysis [PXY*05], or financial data analysis [KCA*16].

Graph visualization. In general, graphs can be displayed as *node-link diagrams*. Several algorithms for different graph layouts have been proposed. The arrangement of nodes can be force-directed, constrained, multi-scale, or even layered [vLKS*11]. Multivariate networks comprise attribute data about nodes and edges. An encompassing survey of approaches for visual analysis of multivariate networks is given in [NMSL19]. In their taxonomy, the authors distinguish layout types as node-link based, tabular based, and implicit-tree based, and discuss visual mappings of attribute data to these structures. Node-link based layouts are further distinguished by topology-driven and attribute-driven layouts. In our work we focus on node-link layouts, which are controlled by both topology and attributes in a weighted way. Specifically, we employ the idea of force-directed layouts [FR91].

Graphs with additional, often scalar node values are generally referred to as *attributed graphs* [FP13]. These scalar values can be used to obtain an application-specific node order in the graph layout. Such attribute-ordered layouts commonly include linear layouts, such as pedigree charts [Cum08], TimeNets [KCH10], or SVENs [AB14]; or radial layouts such as Fan-Charts [DR08]. In our approach, we extend the force-directed layout with additional forces, which, for example, take attribute information into account. This not only improves the comparability of nodes with similar attribute values, but also supports the manual adjustment of node positions to achieve custom layouts.

Investigating or exploring graphs requires interaction with its topology and attributes. Interaction techniques can comprise, for instance, reducing or filtering nodes, detecting structural patterns such as circles [VBW17, vLKS*11], incorporating details-on-demand approaches, or embedding network structures with detailed node glyphs. The Caleydo framework [SLK*09] employs a multiple-window approach for exploring graph structures in more detail. We propose focus+context techniques to explore graph nodes based on their underlying attribute values.

Spatialization. Since centuries, geography is concerned with the creation of spatial visualizations, typically aiming to describe physical surroundings, and serving tasks like path planning and spatial analysis [KO10]. Also, Information Visualization to date has made many efforts to visualize non-spatial data as maps for exploration. In fact, one of its core task is to find spatial mappings of non-spatial data. Some approaches [GHK14, BCB03] describe generic frameworks for creating maps from non-spatial data, and compare and discuss applicable computational tools. The first step in these frameworks is map creation: data items are projected (mapped) to an output space, typically a 2D or 3D display space. Dimensionality reduction or embedding methods are useful to this end. The mapped item locations can then be grouped by cluster analysis to form regions in the display space. As a third step, boundaries of groups of data in the map can be computed, visualized, and connected. Once created, data maps can be augmented by overlays to show relationships among map areas by connectors (edges) [GHKV09, GHK10] between clusters or elements. The *Bubble Set* visualization [CPC09] uses marching squares on data points to emphasize set memberships in the data. An application of Bubble Sets in abstract data maps was studied by Gansner et al. [GHK14] and Jianu et al. [JRHT14]. We use contour lines to show the membership to a specific attribute value range. Data maps were applied to many problems, including the analysis of document collections [SBB13, Sku04, WTP*95, Sku02, SdJ05].

Other examples use height maps to emphasize graph properties, such as ambiguity [WSA*16], or visual importance [LLW16]. Lichtenberg and Lawonn [LL18] parameterize background pixels in screen-space projections of tree-like structures depending on the depth value of their projected contour. Van Liere and Leeuw [vLdL03] use splatting as an image-based method to extract a height field from a graph. Wise et al. [WTP*95] create a 3D document landscape by computing a height field indicating area weights. Börner et al. [BCB03] also apply shading to the created 3D landscape. Xu et al. [XChHT07] use the attribute information of graph nodes to compute a smooth 3D landscape that is blended over a graph. Zhang et al. [ZWP17] create a rectangular 3D height field from a rectangular, space-filling partition of the map area. Saket et al. [SSKB14] propose to place graph nodes with similar attributes close together and highlight the relations with colored polygons. Fried and Kobourov [FK14] additionally use heat maps to depict the temporal dimension. Wu and Takatsuka [WT08] employ geodesic self-organizing maps that use node attribute similarities to improve the graph layout and reduce the number of edge crossings. In our work, we provide continuous control over the influence of the attribute values on the force-based layout

Map metaphor. Several previous authors have embedded graphs or networks in map-like environments, ranging from rather abstract 2D map representations [BCB03] to 3D information landscapes for which cartography rendering techniques can be employed. Isopleth maps depict regions of a specific attribute value with lines [Mac82] or extract topological landscapes from scalar valued functions [HW10]. In a work closely related to ours, Gronemann et al. [GJ12] describe a map rendering technique for visualizing a landscape with elevation hints that visually representin cluster properties in a clustered graph. The technique guides edges in the map based on elevation properties, and utilizes common topographic map rendering and shading. In our work, we support arbitrary node attributes to be represented by a contextual attribute landscape. Instead of routing edges around invariant landscape features, we allow the layout-defined edges to shape the underlying height field to achieve an improved visual embedding of the graph.

Several previous work analyzed the general usability of map metaphors for different applications. Tory et al. [TSW*07] conducted a study on the effectiveness of 3D map metaphors for scatterplot visualizations, and concluded that maps do not add

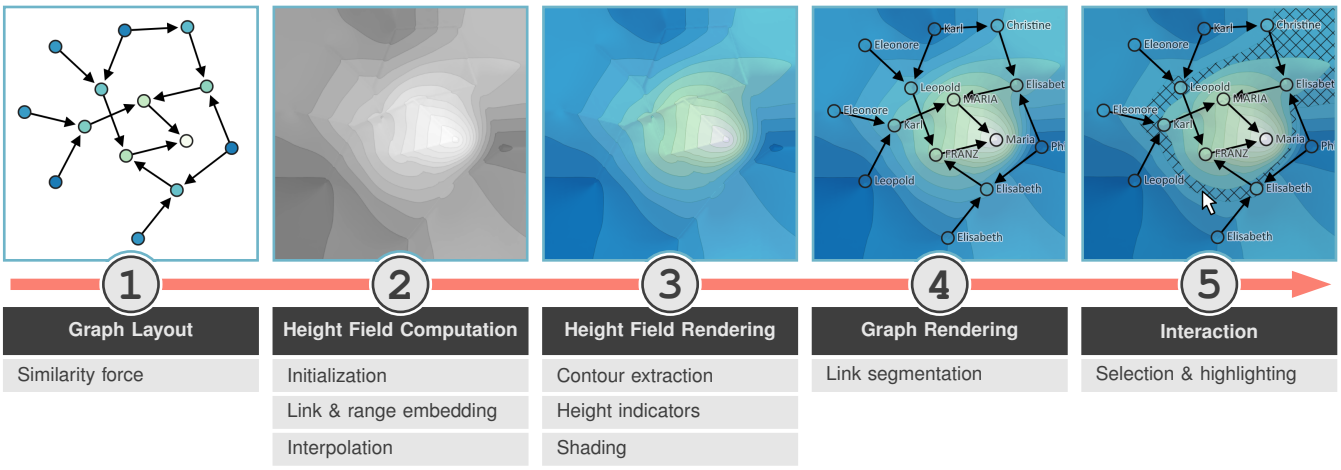


Figure 1: *TAM rendering workflow. After computing the layout for a given graph, a height field is defined by interpolating fixed attribute values at node positions and along edges of embedded links and ranges. The resulting height field is rendered with contour lines, indicators and shading to create a terrain-like impression. The graph is then rendered with node labels and segmented links resolving edge-crossings. Finally, the user can explore the data by selecting specific nodes, highlighting attribute ranges and zooming.*

additional useful information to scatterplots. On the other hand, Saket et al. proved in several studies that map-based visualizations are both more memorable [SSKB15] and more enjoyable [SSK16] than pure node-link diagrams. We report on a user study and different case studies that show that for attributed graphs, landscape metaphors can be useful to convey the distribution of node attributes, but also their development along edges.

3. Topographic Attribute Maps

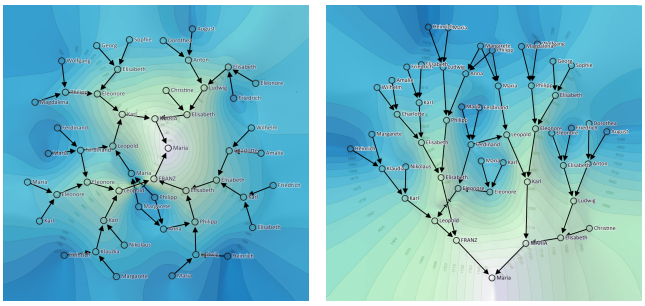
In this section we will discuss the individual steps of the TAM computation workflow, as outlined in Fig. 1. The input is an attributed graph $G(N,A,L)$ with $N = \{n_i\}$ being a set of nodes, $A = \{a_i\}$ denoting its associated set of attribute values, and L a set of links connecting nodes $n_i, n_j \in N$. Depending on the data, attribute values of A can represent any quantitative context, e.g. age or income of persons in a social graph, temporal data such as date of events in a causal graph, or execution run times of procedures in a call graph.

The first step is to compute an appropriate layout of the graph that strives for respecting the topological structure of G while at the same time favoring the comparability of attributes between neighboring nodes. Based on this layout, an ambient scalar field is computed that interpolates the attributes at node positions using particular spatial interpolation techniques. This step also accounts for specific visual optimizations to provide an improved visual embedding of the graph, its links and optional attributes ranges, into the height field. In the next step, the resulting height field is rendered as a terrain using standard cartographic mapping techniques, employing iso-contouring, height indicators, and additional shading to enhance the terrain perception. Finally, the embedded graph is composited on top of the rendered height field. To this end, crossings of graph links are visually resolved using a tunnel metaphor and associated with cartographic glyphs based on the given height field. The resulting visualization can be interactively explored using common interaction means.

3.1. Graph Layout

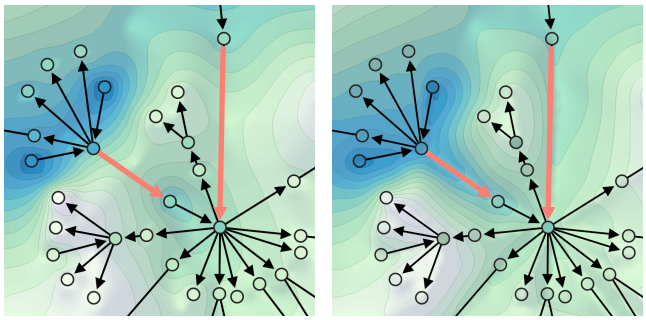
Computing a suitable graph layout requires to find an embedding $E : n_i \mapsto \mathbf{x}_i \in \mathbb{R}^2$, that defines a set $X = \{\mathbf{x}_i\}$ of node positions on the plane. In general, finding such an embedding is a non-trivial task, as large graphs often exhibit a complex topological structure that potentially result in undesirably overlapping nodes and links. Force-directed layout strategies minimize such overlaps [FR91]. These methods perform a particle simulation on the graph's nodes that incorporate repulsive forces striving for expansion, while constraining linked nodes with spring-like forces to maintain their connectivity. This typically leads to layouts that depict the topological structure of a graph in an understandable and intuitive way, an important aspect for many topological analysis tasks. Classic force-directed layouts, however, do not include node attributes in the embedding and can therefore distribute nodes with similar attribute values far across the image domain, as shown in Fig. 2a. Since we aim at providing a comprehensible overview of both the structural *and* the attribute domain of G, we seek for a compromise between purely structural layouts and attribute-aware mappings.

Similarity forces. To achieve this compromise, we offer additional attribute-based forces to be optionally incorporated in the force-layout computation. These forces aim at maintaining a spatial distance $d_S(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\|$ between node locations that is proportional to their attribute difference. This is realized by introducing Hookean forces of 'virtual' springs between nodes n_i and n_j, with spring rest lengths r_{ij} defined by $r_{ij} = |a_i - a_j|/\kappa$. Here, κ is a user-defined global factor, that defines the intended slope of the height field between two points of different attributes. The introduction of such additional similarity forces can, as expected, limit the force system and affect its ability to converge towards a topologically meaningful layout. We thus need to balance the influence of this similarity force on nodes with the influence of the standard link force that maintains the graph link distances. This can be achieved by normalizing the *force strength* parameters, which



(a) (b)

Figure 2: *(a) Force-directed graph layout of an ancestral graph. (b) Adding similarity forces results in a more attribute-ordered layout, without too much disrupting the overall structure.*



(a) Node embedding (b) Node + link embedding

Figure 3: *(a) A height field defined by node values alone leads to graph links running along non-monotonic height profiles. (b) Setting height field constraints along the links consistently embeds graph branches as a whole.*

weight the force vectors during the simulation. We found a meaningful balance between both forces to be best achieved by scaling the similarity force strength exerted on a node n_i by the ratio $v_i/|S_i|$, where v_i denotes the node valence, i.e., the number of link-based springs, and $S_i < |N|$ the number of virtual springs affecting the node. The layout obtained by using these similarity forces is depicted in Fig. 2b. Note that while these forces aim at enhancing the control over force-directed layouts, our proposed TAMs concept is compatible with other layout algorithms as well (see Section 5).

3.2. Height Field Computation

Based on the embedded attributed graph G, we now define an ambient attribute field $F : \mathbf{x} \mapsto \mathbb{R}$ interpolating the attributes a_i at node positions $\mathbf{x}_i$. In our context, F is regarded to as height field, defining a parametric surface that is visualized as contextual structure for our graph. The overall goal is to produce a height field that puts the attributed graph nodes into a comprehensible ambient context that aids comparison and exploration tasks of graph elements. In particular, the topography of the height field (context) should not distract from the graph (focus of interest). To this end, the height field should be most possibly smooth and exhibit few topological features and irregularities, while still providing an appropriate and accurate context in graph layout regions of higher complexity.

Initialization. We represent F as a discrete raster image I_F of appropriate resolution, encompassing the bounding box of the embedded graph. In the first step, I_F is initialized by rasterizing the node positions $\mathbf{x}_i$ to the image and marking them as hard constraints. For the subsequent interpolation step, these values serve as interpolation bases and thus remain invariant. The topography of the resulting height field in all the remaining pixels of I_F will usually strongly depend on the used interpolation strategy.

Link embedding. An important aspect for ensuring the visual simplicity of the height field and its support for the representation of the graph is its resulting topography close to and around the different visual graph elements. One example are the links of the graph, which visually connect nodes at different altitude levels in the attribute height field. To ensure a consistent contextual representation, we want to prevent irregular changes of the underlying height field values along these links. An example is given in Fig. 3a, where highlighted links need to overcome hills of higher altitude to connect lower-valued nodes. To visually resolve such situations, we enforce a monotonous slope on the underlying height field, by linearly interpolating the attributes of the corresponding end nodes and rasterizing them to I_F as additional hard constraints. Applying a dilation operation on the resulting constraint pixels in I_F has proven to additionally relax the surface tension around the links lines. Using this approach, links become more intuitively embedded into the ambient height field, and visually reflect the attribute variations along the graph edges in an improved manner (see Fig. 3b). Links that cross at different altitude levels represent a conflict for the rasterization of interpolation sources in I_F. In such cases, we always keep the maximum value stored at the intersection point. After interpolation, this typically leads to a landscape containing an embanked bridge over the intersection point, supporting the higher-level link while exhibiting steep hillsides on either side along the crossing lower-level link, which visually resembles tunnel entry and exit points. When drawing the graph at this location, we combine these emerging topographic features with cartographic tunnel glyphs placed under these embankments (Section 3.4).

Range embedding. TAMs natively enable *visualizing range information* defined over the attribute domain. Attribute ranges such as lifespans of persons associated with the graph nodes can be visualized as *curve paths* ascending the height field and spanning their respective range. These curve paths can be similarly integrated into the current workflow by extending the original graph G by one (linear paths) or several (curved paths) additional nodes chained together and linked to their associated graph node. This allows the force-directed layout to incorporate their courses in a holistic graph layout computation. Similar to embedding links, the locally supporting height field for these paths is created by rasterizing the paths as hard constraints to I_F. After initializing I_F with node and embedded range attributes, the remaining pixels can be interpolated to a smooth and consistent height field.

Interpolation. The interpolation of scattered 2D data is a fundamental task for the construction of digital elevation models used in GIS and Geostatistics. Common methods are radial basis functions

(RBFs) [Sch95] with singular weighting kernels, Voronoi-based methods like Natural Neighbor interpolation [Sib81, BHBU06], geostatistical interpolation via Kriging [Kri51, OW90], or even hydrologically sound elevation models, simulating realistic geological ridge and streamline features [Hut88]. As we aim for height fields that exhibit smooth contours and avoid shape irregularities, we investigated two methods that were most promising to meet these requirements: (i) *Natural Neighbor* interpolation according to Sibson [Sib81], and (ii) *Diffusion*-based interpolation inspired by Orzan et al. [OBW*08]. Both produce smooth height fields, while avoiding too much bumpiness, an undesired characteristic of singular-basis RBF methods. Section 6 gives a qualitative evaluation of both methods for TAM creation based on user feedback.

3.3. Height Field Rendering

After computing the attribute field F, it is rendered in the style of a topographic map, using contour mapping, overlays, and shading.

Contour extraction. Utilizing the intuitiveness of topographic maps, we visualize the inferred scalar field with contour maps and use suitable color scales for the assignment of attribute colors. The first step comprises an isocontour extraction from the given height field I_F using Marching Quads. This leads to a stack of attributed isocontour polygons that are colored according to their atribute value and displayed below all other subsequent visual elements.

Height indicator placement. To support contextual visualization of the attribute field, we distribute local *height indicator* labels across the map, showing the height values of local contours (see Fig. 2). These indicators are placed along integral curves of the gradient field ∇F whenever these curves intersect an isocontour. To this end, we begin with a series of pseudo-randomly distributed seed points $\mathbf{x}_0$ along the boundary and center of the domain of F, and trace the gradient in both ascending and descending directions until $\|\nabla F(\mathbf{x})\| < \varepsilon$, i.e. until a saddle or local extremum is reached. An attribute indicator label is then placed at a point $\mathbf{x}$ along the curve and aligned orthogonally to the current gradient if $|F(\mathbf{x}) \; mod \; \delta_A| < \varepsilon$, with δ_A being the contour interval.

Shading. Depending on the topography of the height field, the visual discretization introduced through contour mapping can hide subtle features that lie below the contour range. To emphasize these small-scale details and improve the depth perception of the attribute landscape, we apply Lambertian shading to the final height field. We used a light direction vector pointing at the lower right area of the landscape and found an incident light angle of 60° to achieve a good shading variability and detail enhancement.

3.4. Graph Rendering

In the last step of our rendering pipeline (recall Fig. 1), the graph is rendered on top of the stylized height field. For complex and non-planar graph layouts, an important aspect in this phase is the visual resolution of links that cross at different heights. As discussed in Section 3.2, such crossings create an ambiguity at the intersection point, where different interpolated node attributes conflict in the definition of the local height field. One way of handling such

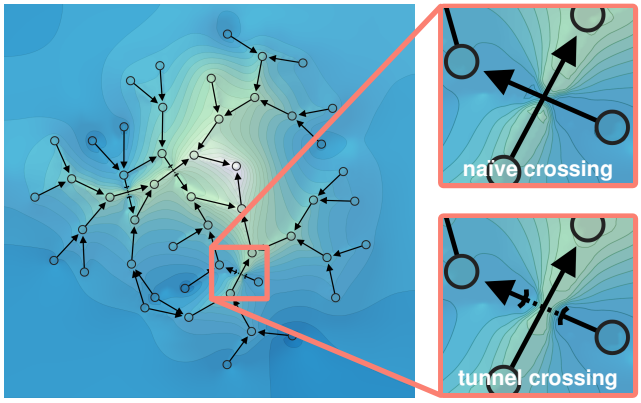


Figure 4: *Comparison between link crossings using naïve landscape averaging and visualization using a tunnel metaphor.*

conflicts is to define the altitude around a link crossing as the average of the conflicting link attributes at the intersection point, shown in the top-right image of Fig. 4. However, this can create link embeddings with non-monotonic height profiles and diminishes the visual cues about the link attributes around the crossing point. In contrast, we resort to an intuition that naturally emerges from our topographic landscape metaphor: When interpreting links as roads connecting points on a mountainous landscape, it appears natural for a link crossing another link at a higher level to be handled via the concept of a bridge. Equivalently, a link crossing *under* another link at a lower level can be intuitively thought of as a tunnel section crossing underneath the ground level of the higher road. In our implementation, we chose to consistently resort to the tunnel metaphor, which uses dashes to indicate the underground path section and thus appears visually more distinctive.

Link segmentation. When rendering the graph, the visual representation of links that are interrupted by tunnel sections have to be adjusted accordingly. To this end, we segment the given links by their intersection points with the attribute terrain, and render the resulting segments as dashed or solid lines depending on their surface or underground location. The intersection points of a link with the terrain are determined by stepping along its line $\overline{ij}$ and testing at each point $\mathbf{x}$ for a change in the sign of the difference function

$$\Delta_{ij}(t) = lerp(a_i, a_j, t) - F(lerp(\mathbf{x}_i, \mathbf{x}_j, t)), \quad t \in [0,1] \qquad (1)$$

where $lerp$ denotes linear interpolation. The bottom-right of Fig. 4 depicts the resulting visual representation of tunneled crossings.

3.5. Interaction

Our proposed graph visualization technique allows for various intuitive interaction paradigms supporting both visual analysis and exploration. Moreover, we allow users to select specific elements within the topological structure of the graph as well as within the attribute domain represented by the height field. Selecting specific nodes will highlight the node together with its immediate neighbors by increasing the thickness of their outlines. This allows users to explore the topological structure by following its connectivity. To

support the global comparability of node values in the attribute domain, users can hover over specific contour lines, which will globally highlight all contours at the given altitude on the map. Since color is already used for encoding scalar values, we use a hatched pattern to highlight the selected contour [MVB*17].

4. Implementation

We implemented our method as a web-based JavaScript application using D3 v5.9.3 to compute the force-directed graph layout and to render the height field and graph visualization into an SVG canvas. For the results presented in this paper, we employed D3 built-in collision forces to maintain a minimum distance between nodes and avoid node overlaps in dense graphs, and employed the established color scales provided by D3 as well. For an ancestral graph containing about 600 nodes, the force-directed layout algorithm takes about 20 seconds to converge to a suitable layout. This layout can also be adjusted by dragging and dropping the family nodes. On a 500×500 pixel image I_F, the height field computation requires 12 seconds using natural neighbor interpolation, and 4 seconds for hierarchical diffusion. Timings were measured on a PC with Intel Core i7-4930 CPU, using the Firefox v71.0 browser.

5. Results and Applications

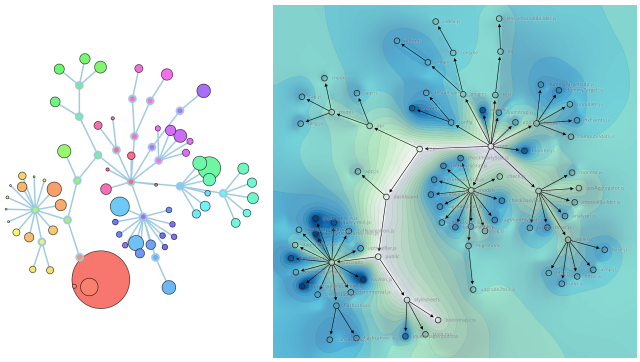
In this section, we compare the expressiveness of our method to related graph visualization techniques, and demonstrate its versatility with attributed graph data from different application domains.

5.1. Node Space Coverage

Alternatively to or in conjunction with color mapping, a popular way to illustrate attribute differences of nodes is to encode attribute values in node size. An example is given in Fig. 5, showing a dependency tree of 81 files in a software code repository. Nodes denote files, their links denote dependencies, and the node attribute represent the lines of code in each file. A visualization encoding file size in node circle radius [Zan] is shown in Fig. 5a, where the visualization space covered by large nodes quickly leads to overplotting. At the same time, comparability of absolute attribute values based on color or node size alone is not directly possible. Fig. 5b shows a TAM of the same graph that keeps nodes sizes compact, but invests the visual space to emphasize the graph structure via an underlying 3D height field, while creating a relation between distant node attributes via contour lines supporting detailed attribute comparisons.

5.2. Network Traffic Analysis

We provided our technique to researches from the field of distributed systems and computer networking. Their research often involves the analysis of traffic volumes, network utilization, and network infrastructure capacity based on traffic simulations performed on network graphs. In these graphs, nodes either represent network devices or an abstract representation of a backbone network, and their attributes denote different measures of network traffic. An interesting application of TAMs is the visualization of these traffic measurements among the graph, e.g., to provide a basis for network planning or traffic bottleneck identification. For instance, a main goal of *edge computing* [SD16] is to reduce data



(a) (b)

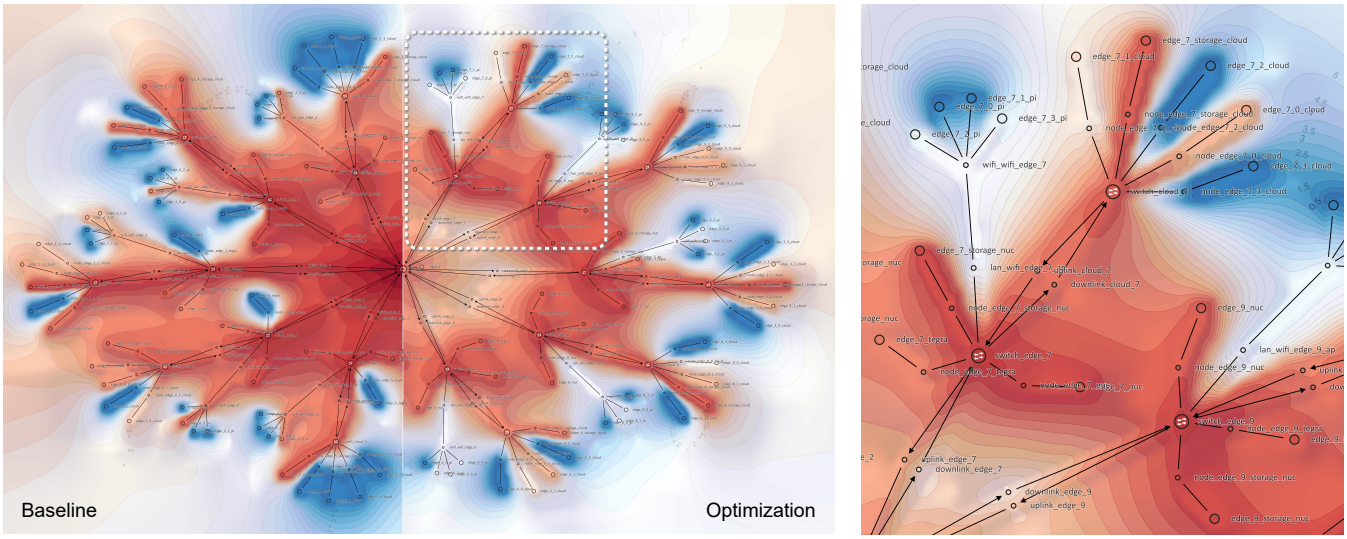
Figure 5: *(a) Visualization of a repository dependency graph, where nodes represent files, their color indicates a running node id, and node circle radii depict file sizes [Zan]. (b) The same graph with fixed-sized node radii, using a TAM to illustrate file sizes.*

movement through the network by performing computational tasks that operate on data, closer to where those data are stored. Fig. 6a shows a network topology of ten interconnected edge networks, where the center node represents the Internet backbone. The shown TAM compares two different resource allocation approaches and the overall traffic volume in Megabytes they produce at each device node. The data are gathered from a compute platform and network simulation developed for edge computing research [RHM*19]. The goal was to study the impact of resource allocation approaches on the traffic that leaves the edge networks. The left half of Fig. 6a shows the data for the baseline approach, where traffic often leaves edge networks. In contrast, the right half visualizes the data measured for an improved allocation strategy. While the overall network traffic remains the same, it is now much more isolated within edge networks. Note that our method supports such a direct comparison of multiple attribute channels by decoupling the graph layout from the node attributes (disregarding similarity forces). The researchers also reported that TAMs allow them to directly see that the network traffic is balanced across individual edge networks.

Another use case is the problem of *capacity planning*, where network operators are tasked with identifying overloaded hosts or network links in the topology, and finding candidates for replication and load balancing. Fig. 6b shows a detailed view of a particular edge network in the same topology. The elevation around data *storage* nodes show that the overall traffic is concentrated around these nodes. The attribute contours provided by the TAM enables the comparison of values with other storage nodes in the same network. According to the distributed-systems researchers, an actionable insight of this visualization would be that replicating storage nodes and balancing data access load between them would reduce the risk of network congestions around individual nodes.

5.3. Ancestral Graphs

Ancestral graphs are widely used to represent personal, historical, or fictional genealogical relationships between persons. In such graphs, nodes represent persons, links represent family connections, and the attribute of interest is mostly the date of birth or



(a) (b)

Figure 6: *(a) Comparison of two different resource allocation strategies in a computer network infrastructure using TAMs. Attribute values denote total network traffic in Megabytes (log-scaled, red is high). (b) Detail inspection of a single edge network, highlighting overloaded hosts for resource capacity planning.*

the time range within which persons lived. A well known visualization for such graphs are pedigree charts [Cum08], which provide an abstract temporal order by aligning nodes by generations. These, however, are only suitable for binary trees, and are therefore restricted to illustrating graphs or subgraphs that are free of circles and multiple family connections. Fig. 2 shows the ancestry graph of Marie Antoinette, where the nodes are attributed by person's birth dates. This graph is mostly binary, but contains circles as her parents were both 2nd-degree and 3rd-degree cousins. Force-directed layouts typically provide a good view on such topological circles at the cost of a scattered distribution of nodes associated to the same birth decade (Fig. 2a). Using similarity forces proposed in Section 3.1 allows to obtain a more attribute-ordered layout, as seen in Fig. 2b, which more resembles a pedigree chart. However, for larger and more complex graphs, this increasingly obfuscates topological features again. In both cases, the use of TAMs provides additional context that aids the visual analysis of the graph data, e.g., by visually connecting persons born in the same decade.

H-Tree layouts. A specific alternative layout for ancestral tree data are H-trees [NH02]. In this layout, ancestors are placed around a central person of interest in a symmetric and space-filling way. Fig. 7 shows a TAM applied to an H-tree view of Marie Antoinette's graph, demonstrating that our method is versatile and adjusts to different layouts in a suitable way. To reduce visual complexity, siblings are now grouped in *family nodes* (larger red circles), and edges only depict links between persons and the family node of their children. Our TAM allows for relative age comparisons among members of the same ancestral generation, and reveals a mostly balanced elevation of the age landscape at both sides of Marie's family, with the oldest members belonging to her father's side. Moreover, edge crossings that result from circular connections are resolved in a clean way, preserving temporal edge information by the means of the contour map and tunnel glyphs.

Lifelines. In addition to depicting birth dates, genealogists are often interested in the life spans of persons in their graph, an example for visualizing *attribute ranges*. Fig. 8 shows a TAM of a real-world dataset provided by a genealogist, where names have been anonymized. Siblings are again grouped to red family node circles, whereas empty family circles indicate two persons being married without children. Besides the underlying contour map, the distance of a child node from its family node center indicates its year of birth. Starting from the node locations that indicate the years of birth, we use red curves ascending on the underlying height field

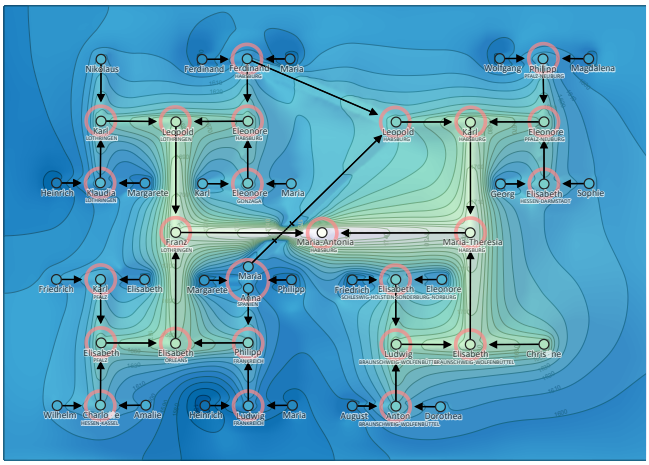


Figure 7: *Ancestral graph of Marie Antoinette visualized using a TAM computed from an H-tree layout.*

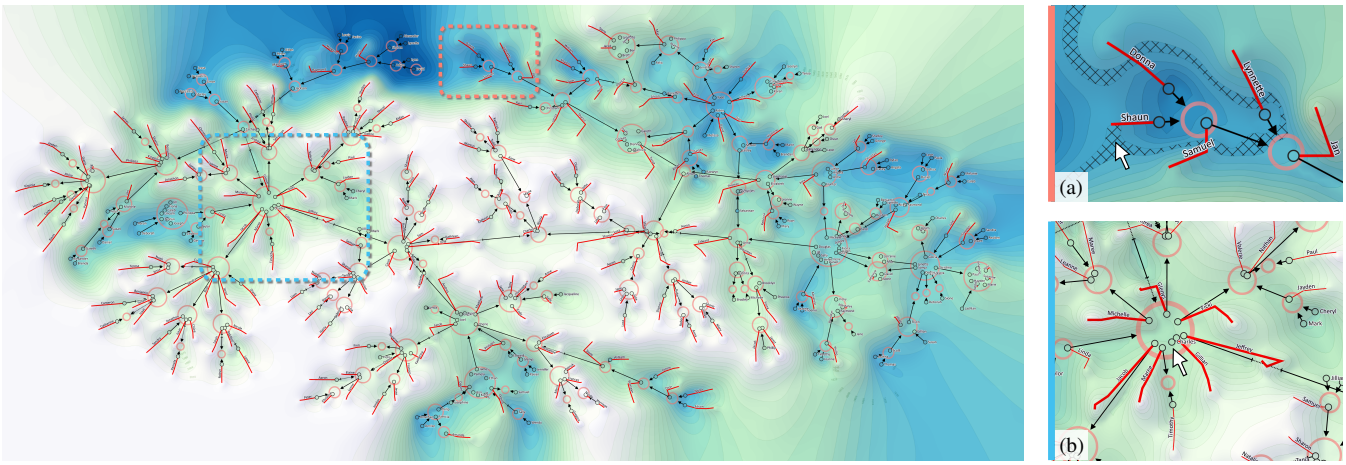


Figure 8: *TAM of an ancestry graph comprising 426 person nodes and 161 family nodes, visualized with lifelines. Siblings are grouped in red family circles, while red curves indicate a person's lifespan. The underlying contour map represents a temporal height field, immediately showing ancestral roots (topside, dark blue) as well as persons who are still alive today, indicated by life lines reaching to white hill tops. The graph layout visually preserves its major ancestral structure, but also reveals topological features like circular connections. The insets illustrate the interactive selection of (a) individual contours, highlighting intervals of equal attributes, and (b) individual family circles. (Data courtesy of Christian Ofenböck, Graz)*

to illustrate the lifespan of persons for which the date of death is known. Note that occasional bends are due to the placement of their polyline nodes using the force layout, as described in Section 3.2.

Complementing the structural assessment of the graph, these lifelines provide additional means to examine its temporal information and further support range-specific analysis tasks. The TAM shows that the majority of the contained relationship data is organized around a central family node containing the siblings *Charles* and *Nicola*. The oldest ancestral roots can be immediately spotted in the north of the landscape, and can be directly related to the father Kyle's side of their family. On the other hand, ancestors of their mother Sharon's side do not reach as far back in time, but contain more members living in the mid 19th century (blue regions). Another interesting topological aspect is that Kyle's part of the family tree does not contain a single cycle, while in Sharon's family we can count at least 5 of those. Moreover, Sharon's family exhibits a pair of two brothers, who married a pair of sisters. An overview of the entire TAM reveals that families with the most children appeared in the late 19th century, while the family sizes appeared to shrink on higher altitudes, with 1 to 3 children on average in the present. The set of persons that are still alive today are spotted via their lifelines, reaching towards the white top plateau of the landscape that represents the present. On the upper boundary of this map, the lifelines of *Max* and his children show that he lost both his sons Jeremy and Jared, while his youngest daughter Sione witnesses his death around her forties. Also on the upper center, the lifeline of the root ancestor Shaun is depicted. Following the contour of his death decade, we see that he was not fortunate to witness the birth of his grand daughter Jan.

For attribute comparisons across complex contour maps, we can additionally employ interactive selection and highlighting. In Fig. 8a we selected the decade of Shaun's death, which globally highlights all contours associated with this time interval. Moreover, selecting a family node allows to highlight all lifelines of its associated children, which is particularly helpful in densely populated map regions (Fig. 8b).

Application in genealogical research. To assess the usability of our method for genealogical research, we have provided another experienced genealogist with a large-scale TAM visualization of his largest ancestral dataset, comprising over five thousand members and multiple regions of dense cyclic connections. Fig. 9 shows this graph with surnames hidden and first names randomly anonymized. In his research, the genealogist has already worked with about seven of the most widely used software tools for family tree visualization on the market. To him, their common limitation is the limited extent of data that can be shown at once in a useful way by the means of typical time-oriented layouts like pedigree charts, fan layouts, or sandbox layouts. As these layouts only allow to show sub-trees of general family graphs, he originally had no clear structural overview of his data in terms of a mental map.

The structural overview provided by the force-based layout of his entire data allowed him to quickly identify isolated family parts and directing his focus of research to finding the missing links with the main graph. At the same time, he reported that to him, the topographic visualization of the temporal landscape was useful for (i) understanding temporal developments of family branches, and (ii) identifying focus regions of interest within the graph to attach the next genealogical research steps to. As a consequence, he reported a significant boost in his research efficiency: While using traditional software, his records show an average dataset growth of about 460 persons a year. After studying the TAM visualization and adopting his research efforts, he reported 370 new registered family members within a time span of 32 days, which he attributes to the new insights gained from our visualization.

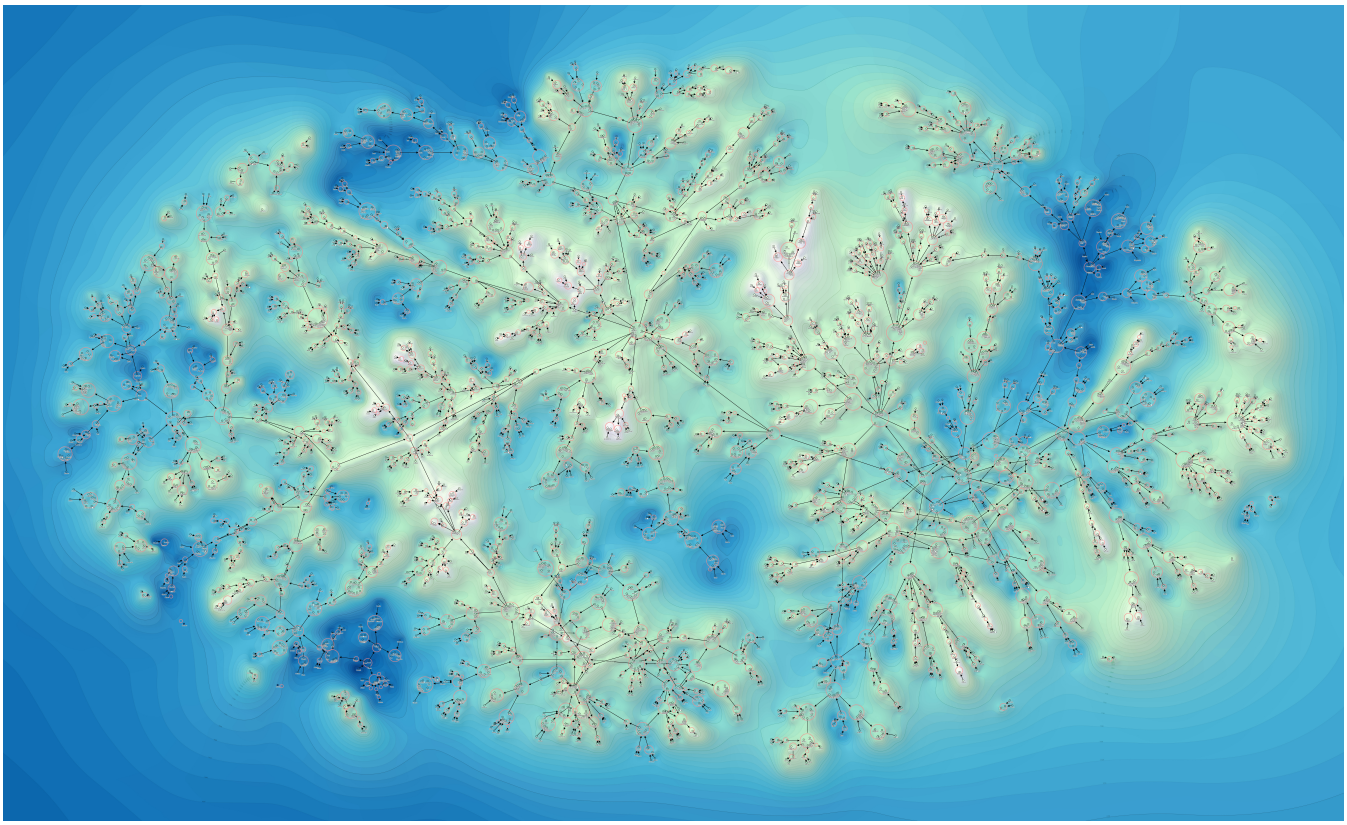


Figure 9: *Family graph of 5295 person nodes connected by 1686 family nodes, visualized as a TAM with anonymized names (zoom in to see details). The map reveals regions of ancestral roots (deep blue) as well as the temporal development to members living in the present. Regions of high topological complexity are revealed together with their temporal localization. (Data courtesy of Gerd Herud, Graz)*

5.4. Densely Connected Graphs

In this section we discuss the applicability of TAMs in graphs exhibiting a dense screen coverage of nodes and edges. To this end, we have used our method to visualize a large graph extracted from data provided by the IMDb movie database [IMD20]. The IMDb data constitute a graph of movie nodes with additional information (release year, type, rating, etc.) and person nodes (actors, producers, etc.) associated with them. From this database, we extracted movies released after 1960 that are not episodes of a series, have a runtime of at least 70 minutes, and have a valid ranking assigned. From the resulting set of records, we randomly sampled a connected subgraph of about 1634 movie nodes associated with 1673 actors. The movie nodes were attributed by the movie's rating, while the actor nodes were attributed with the averaged rankings of their associated movies. Fig. 10 shows the resulting TAM of this graph.

On the large scale, the force-based layout exhibits clusters of movies performed by groups of similar actors. To the north we observe a large cluster consisting mainly of American movies, including many well known Hollywood productions. The second largest cluster located to the south west comprises mostly Japanese movies, and is connected to the main American cluster by only a few movies like *The Small Hours*, featuring Japense actor Masaya Kato. The contour maps provided by our TAM reveal that the highest rated movie in the Japanese cluster (*Kuroku No Basket Movie 3*) is located at least one contour level below the highest rated American movies (*The Dark Knight*, *Avenger's Endgame*, etc). At closer inspection of the American cluster (see inset), we find actor Chris Hemsworth, who appeared in highly rated Marvel movies (deep red) but also in lower-rated movies like *Red Dawn* (white).

As our technique exploits unused visual space in the background of the graph, densely overplotted regions, such as the center of the main cluster, limit the usable space for the underlying TAM. Nevertheless, the embedding of links allows to identify overall structures of links and their rating level, even in dense regions. The visual accuracy can be improved by increasing the resolution of the underlying attribute scalar field, or by incorporating edge routing mechanisms. Similarly, employing node clustering techniques typically allows to adjust the topological level of detail to the current display resolution [APP10].

6. Evaluation and Discussion

We estimated the potential of TAMs for solving graph-related tasks in two user evaluations. In a first study (*S1*) we evaluated TAMs created with diffusion interpolation, while in a second study (*S2*) we evaluated Natural Neighbor interpolation (see Section 3.2).

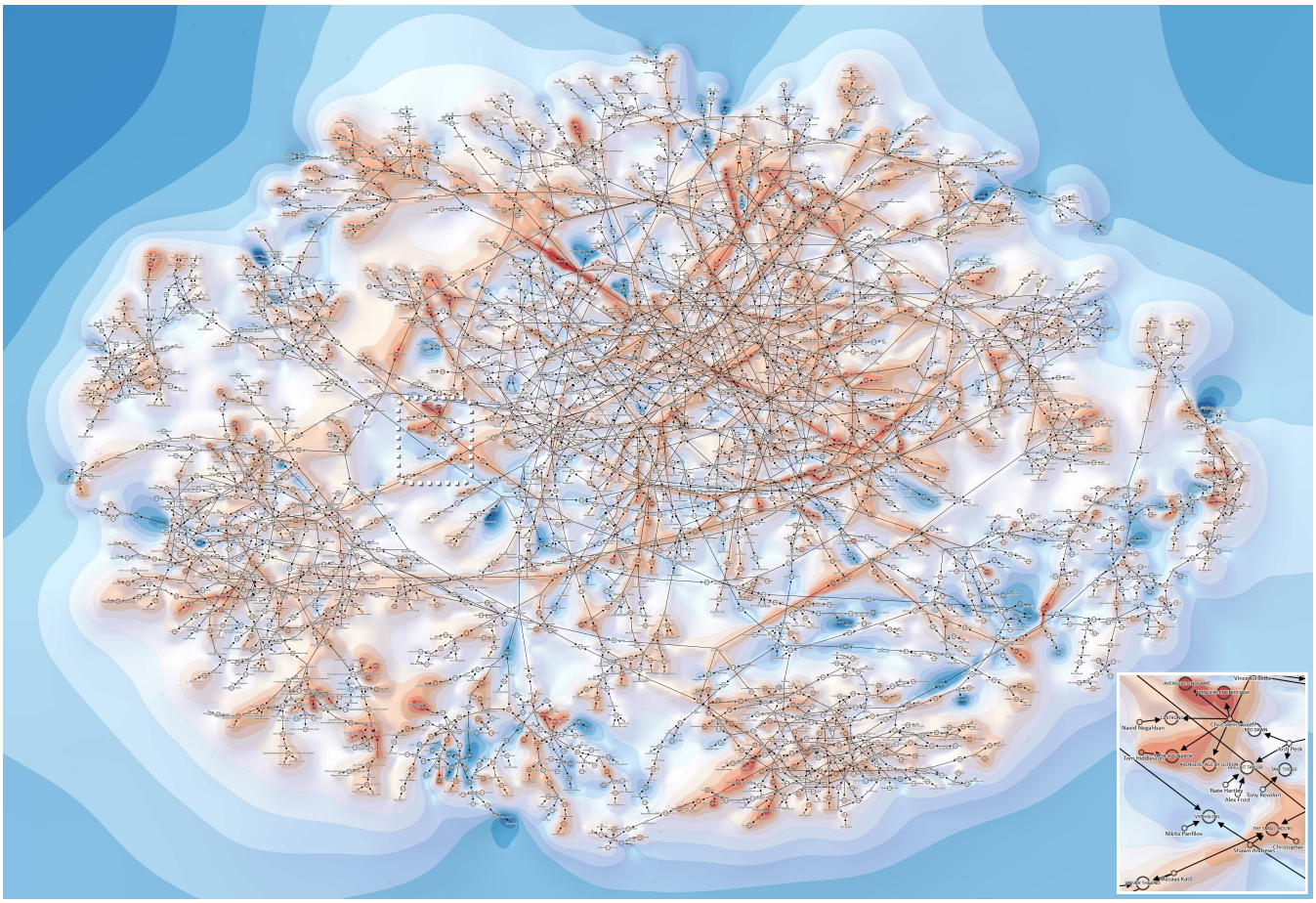


Figure 10: *TAM visualization of a graph of over 3300 nodes of movies and actors recorded in the International Movie Database (zoom in to see details). Node attributes represent ratings between 1 and 10 for movies and average movie ratings for actors (red is high). The TAM contours resolve ratings at steps of 0.5. The distinctiveness of scalar field values is naturally limited by the rasterization density of edges and nodes in the underlying scalar field. Large-scale attribute structures are still visible in this example.*

Study setting. We recruited $40/47$ participants in S1/S2 with different backgrounds (computer science, engineering, archaeology, HR, radiology). Three participants in total $(1/2)$ reported to have a color deficiency. We asked the participants to rate their experience with using graph data visualization on a three-step scale. $12/15$ answered to have little to no experience, $25/27$ claimed to have moderate experience, and $3/5$ judged themselves as experienced users. Both studies have been realized in web-based form with static images, and used both small graph datasets of actors performing in movies (about 20 nodes) and ancestral graphs showing genealogical relations. Participants had to solve the following tasks:

- **T1** (movie graph) Find the actor that has the highest rating.
- **T2** (movie graph) Find the three actors that have the same rating.
- **T3** (movie graph) Count how many actors played in a movie, where at least 2 other actors with lower rating appeared as well.
- **T4** (ancestral graph) State if a person was still alive when another person was born.

T1 targeted finding extrema in the attribute values. T2 focused on whether participants were able to identify nodes that have similar values. T3 was chosen to test whether the participants were able to combine both the topological and the attribute domain information for task solving. T4 was used to assess the effectiveness of the life lines and has no baseline equivalent for comparison. We compared error rates and the confidence rated by the participants on a 10-point Likert scale, from 1 (highly unconfident) to 10 (highly confident), both for TAMs and a baseline graph visualization. As baseline we chose a common and intuitive visual mapping approach for attributed graphs, that colors nodes according to their attribute value [CD14]. Different values were used for the baseline and for TAMs to avoid any bias from participants becoming familiar with the data. Introductions about attributed graphs and TAMs were given to the participants before starting the tasks.

Quantitative analysis. Since the web-based framework did not provide a controlled test environment, response time was not accounted for. In contrast, the study focused on the potential of TAMs for decreasing error rates and increasing trust of users in the provided method. In both studies, TAMs led to lower error rates for tasks T1–T3 compared to the baseline (Fig. 11). Participants noted

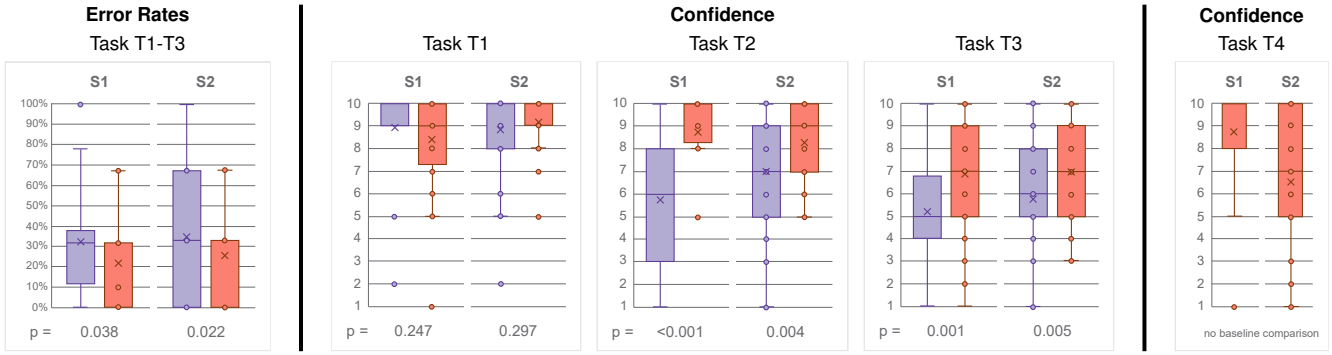


Figure 11: *Box-plots of error rates and confidence values for our tasks, together with P-values from Wilcoxon signed-rank tests. In S1 and S2 error rates (accumulated over all tasks) for TAMs (orange) were significantly lower than for the baseline (violet). For T2 and T3, the confidence values for solving the tasks with TAMs were significantly higher. Confidence values for T4 are shown without comparison due to the missing baseline equivalent. Error rates for this task were* 12.5% *and* 17.0% *for S1 and S2, respectively.*

that "it needs time to get used to see and read the encoding properly", which indicates that a longer adaption and learning period could help to further decrease the error rates. We deemed confidence an important indicator of a user's willingness to embrace and use a new technology, and therefore accessed confidence feedback for all three tasks as well. Fig. 11 shows that participants were significantly more confident when solving more complex tasks (T2, T3) with TAMs compared to the baseline visualization.

Qualitative analysis. We also collected qualitative feedback to estimate the effectiveness of the new TAM visualization concepts, comparing diffusion and Natural Neighbor interpolation, and comparing different layouts produced by applying similarity forces. Participants stated that they *"liked the landscape representation"*, that TAMs were *"very helpful as it allowed to establish equivalences and differences between the nodes more easily"*, that the layout makes *"it easier to detect clusters with the same values"*, and that for comparisons they *"just had to follow the height-lines"*. However, some also noted that the encoding *"can become complicated for larger graphs"*. Finally, participants were asked to rate how helpful they considered TAMs for the task completion on a scale from 1 (not helpful at all) to 5 (very helpful), which yielded an average rating of (2.95 ± 1.02) for S1 and (3.45 ± 0.69) for S2.

In S2, we additionally asked participants to state which interpolation algorithm and which layout algorithm they prefer, and why. Participants preferred the Natural Neighbor interpolation (87.23%) over the diffusion interpolation (12.77%), because it was *"easier to parse since the isolines are not squeezed"*, it shows *"smoother transitions between the different zones and clearer alignment"*, and it produces *"more homogeneous/symmetric looking structures"*. Also, they preferred the layout produced by applying the similarity forces (70.21%) over the layout disregarding attribute similarities (21.28%). For ancestry data, participants found that the similarity force layout made the *"timeline more obvious"* and made it *"easier to see how many generations have passed"*.

Threats to validity. In our studies, users first performed the tasks on the baseline visualization, and then on TAMs. The same tasks used different but comparable data sets for both the baseline and the TAMs. Furthermore, the majority of participants had at least some experience with graph data visualization, i.e., interpreting a graph was not a completely new concept to them. For these reasons, we did not expect a significant learning effect to occur. However, without randomizing or counterbalancing the order of the tasks and testing for a learning effect, it cannot be ruled out completely.

7. Conclusion and Future Work

Conclusion. In this paper we presented a new approach for the visualization of attributed graphs. Our visualization concept employs the metaphor of a continuously varying landscape for the visual representation of node attributes, within which a force-directed layout of a graph is embedded. Attribute values at node locations and along edges are interpolated to form an underlying height field. This height field is visualized as topographic landscape, allowing us to adopt established techniques from topographic mapping to create a visual context of the spatial distribution of the attributes over the graph. Contour lines reestablish the visual connection between similarly attributed nodes across the graph. Other visual mappings include tunnel glyphs and range indicators to further enhance the representation. Interaction techniques enable users to highlight certain graph nodes and height field contours. Moreover, we have shown that TAMs are versatile to be used for different graph layouts. We successfully applied our technique to various use cases from different domains, such as genealogy, network simulation and large graph analysis. Finally, the usefulness of TAMs was evaluated in two user studies resulting in quantitative and qualitative results. Overall, our results show that TAMs support users in analyzing attributed graphs, and were well received by experts from different application domains.

Future work. Similar to other graph visualizations, one limitation of our technique is scalability. Different aggregation techniques for large graphs have already been proposed [vLKS*11], including edge bundling and filtering. Integrating edge bundling and parallel tracks into TAMs to solve occlusion problems will be an interesting topic in the future. We would also like to explore other application areas like cultural heritage, which might also require the integration of metadata attributes.

Acknowledgments

We thank Christian Ofenböck and Gerd Herud for providing their genealogical datasets and their valuable feedback on ancestral research applications, and Thomas Rausch from TU Wien for providing data and valuable input on use cases in network traffic anlaysis. Many thanks to all participants of our user studies. VRVis is funded by BMVIT, BMDW, Styria, SFG and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (854174), managed by FFG.

References

[AB14] ARENDT D. L., BLAHA L. M.: SVEN: Informative Visual Representation of Complex Dynamic Structure. *CoRR abs/1412.6706* (2014). 2

[APP10] ARCHAMBAULT D., PURCHASE H. C., PINAUD B.: The Readability of Path-Preserving Clusterings of Graphs. *Computer Graphics Forum 29*, 3 (2010), 1173–1182. 9

[BCB03] BÖRNER K., CHEN C., BOYACK K. W.: Visualizing knowledge domains. *Annual Review of Information Science and Technology 37*, 1 (2003), 179–255. 2

[BHBU06] BOBACH T., HERING-BERTRAM M., UMLAUF G.: Comparison of voronoi based scattered data interpolation schemes. In *Proceedings of International Conference on Visualization, Imaging and Image Processing* (Palma de Mallorca, Spain, Aug. 28–30 2006), IASTED '06, pp. 342–349. 5

[CD14] CARR H., DUKE D.: Joint Contour Nets. *IEEE Transactions on Visualization and Computer Graphics 20*, 8 (2014), 1100–1113. 10

[CLY17] CHEN S., LIN L., YUAN X.: Social Media Visual Analytics. *Computer Graphics Forum 36*, 3 (2017), 563–587. 2

[CPC09] COLLINS C., PENN G., CARPENDALE S.: Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1009–1016. 2

[Cum08] CUMMINGS M.: *Human Heredity: Principles and Issues*. Available Titles CengageNOW Series. Cengage Learning, 2008. 2, 7

[DR08] DRAPER G. M., RIESENFELD R. F.: Interactive Fan Charts: A Space-saving Technique for Genealogical Graph Exploration. In *Proceedings of the 8th Annual Workshop on Technology for Family History and Genealogical Research* (Provo, UT, USA, Mar. 13 2008), FHTW '08. 2

[FK14] FRIED D., KOBOUROV S. G.: Maps of Computer Science. In *Proceedings of the IEEE Pacific Visualization Symposium* (Yokohama, Japan, Mar. 4–7 2014), PacificVis '14, pp. 113–120. 2

[FP13] FIONDA V., PIRRO G.: Querying Graphs with Preferences. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management* (San Francisco, CA, USA, Oct. 27 – Nov. 1 2013), CIKM '13, ACM, pp. 929–938. 1, 2

[FR91] FRUCHTERMAN T. M. J., REINGOLD E. M.: Graph drawing by force-directed placement. *Software: Practice and Experience 21*, 11 (1991), 1129–1164. 1, 2, 3

[GHK10] GANSNER E. R., HU Y., KOBOUROV S. G.: GMap: Visualizing graphs and clusters as maps. In *Proceedings of IEEE Pacific Visualization Symposium* (Taipei, Taiwan, Mar. 2–5 2010), PacificVis '10, pp. 201–208. 2

[GHK14] GANSNER E. R., HU Y., KOBOUROV S. G.: *Viewing Abstract Data as Maps*. Springer New York, New York, NY, 2014, pp. 63–89. 2

[GHKV09] GANSNER E. R., HU Y., KOBOUROV S. G., VOLINSKY C.: Putting recommendations on the map: visualizing clusters and relations. In *Proceedings of the ACM Conference on Recommender Systems* (New York, NY, USA, Oct. 23–25 2009), RecSys '09, pp. 345–348. 2

[GJ12] GRONEMANN M., JÜNGER M.: Drawing clustered graphs as topographic maps. In *Proceedings of the International Symposium on Graph Drawing* (Redmond, WA, USA, Sept. 19–21 2012), GD '12, Springer Berlin Heidelberg, pp. 426–438. 2

[Hut88] HUTCHINSON M. F.: Calculation of hydrologically sound digital elevation models. In *Proceedings of the 3rd International Symposium on Spatial Data Handling* (Sydney, Australia, Aug. 17–19 1988), vol. 133 of *SDH '88*, International Geographical Union Sydney, pp. 117–133. 5

[HW10] HARVEY W., WANG Y.: Topological Landscape Ensembles for Visualization of Scalar-Valued Functions. *Computer Graphics Forum 29*, 3 (2010), 993–1002. 2

[IMD20] IMDB.COM I.: Imdb datasets. https://www.imdb.com/interfaces, 2020. [accessed 2020-02-22]. 9

[JRHT14] JIANU R., RUSU A., HU Y., TAGGART D.: How to Display Group Information on Node-Link Diagrams: An Evaluation. *IEEE Transactions on Visualization and Computer Graphics 20*, 11 (2014), 1530–1541. 2

[KCA*16] KO S., CHO I., AFZAL S., YAU C., CHAE J., MALIK A., BECK K., JANG Y., RIBARSKY W., EBERT D. S.: A Survey on Visual Analysis Approaches for Financial Data. *Computer Graphics Forum 35*, 3 (2016), 599–617. 2

[KCH10] KIM N. W., CARD S. K., HEER J.: Tracing Genealogical Data with TimeNets. In *Proceedings of the International Conference on Advanced Visual Interfaces* (Roma, Italy, May 26–28 2010), AVI '10, ACM, pp. 241–248. 2

[KO10] KRAAK M.-J., ORMELING F. J.: *Cartography: Visualization of spatial data*, 3 ed. Guilford Publications, 11 2010. 2

[Kri51] KRIGE D. G.: A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy 52*, 6 (1951), 119–139. 5

[LL18] LICHTENBERG N., LAWONN K.: Parameterization and Feature Extraction for the Visualization of Tree-like Structures. In *Proc. of Eurographics Workshop on Visual Computing for Biology and Medicine* (Granada, Spain, Sept. 20–21 2018), VBCM '18, pp. 145–155. 2

[LLW16] LI J., LIU Y., WANG C.: Evaluation of Graph Layout Methods Based on Visual Perception. In *Proceedings of the 10th Indian Conference on Computer Vision, Graphics and Image Processing* (Guwahati, Assam, India, Dec. 18–22 2016), ICVGIP '16. 2

[Mac82] MACEACHREN A. M.: Map Complexity: Comparison and Measurement. *The American Cartographer 9*, 1 (1982), 31–46. 2

[MVB*17] MEUSCHKE M., VOSS S., BEUING O., PREIM B., LAWONN K.: Combined Visualization of Vessel Deformation and Hemodynamics in Cerebral Aneurysms. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 761–770. 6

[NH02] NGUYEN Q. V., HUANG M. L.: A space-optimized tree visualization. In *Proc. of the IEEE Symposium on Information Visualization* (Boston, MA, USA, Oct 28–29 2002), INFOVIS '02, pp. 85–92. 7

[NMSL19] NOBRE C., MEYER M. D., STREIT M., LEX A.: The State of the Art in Visualizing Multivariate Networks. *Computer Graphics Forum 38*, 3 (2019), 807–832. 2

[OBW*08] ORZAN A., BOUSSEAU A., WINNEMÖLLER H., BARLA P., THOLLOT J., SALESIN D.: Diffusion Curves: A Vector Representation for Smooth-Shaded Images. *Communications of the ACM 56*, 7 (2008), 101–108. 5

[OW90] OLIVER M. A., WEBSTER R.: Kriging: a method of interpolation for geographical information systems. *International Journal of Geographical Information Systems 4*, 3 (1990), 313–332. 5

[PXY*05] PHAN D., XIAO L., YEH R. B., HANRAHAN P., WINOGRAD T.: Flow Map Layout. In *Proceedings of the Symposium on Information Visualization* (Minneapolis, MN, USA, Oct. 23–25 2005), InfoVis '05, pp. 219–224. 2

[RHM*19] RAUSCH T., HUMMER W., MUTHUSAMY V., RASHED A., DUSTDAR S.: Towards a Serverless Platform for Edge AI. In *2nd*

USENIX Workshop on Hot Topics in Edge Computing (Renton, WA, USA, July 9 2019), HotEdge '19, USENIX Association. 6

[SBB13] SKUPIN A., BIBERSTINE J. R., BÖRNER K.: Visualizing the topical structure of the medical sciences: A self-organizing map approach. *PloS one 8*, 3 (2013). 2

[Sch95] SCHABACK R.: Creating surfaces from scattered data using radial basis functions. *Mathematical methods for curves and surfaces 477* (1995). 5

[SD16] SHI W., DUSTDAR S.: The promise of edge computing. *Computer 49*, 5 (2016), 78–81. 6

[SdJ05] SKUPIN A., DE JONGH C.: Visualizing the ica: A content-based approach. In *Proceedings of 22nd International Cartographic Conference* (A Coruña, Spain, July 9–16 2005), ICC '05. 2

[Sib81] SIBSON R.: A brief description of natural neighbour interpolation. *Interpreting multivariate data* (1981). 5

[Sku02] SKUPIN A.: A cartographic approach to visualizing conference abstracts. *IEEE Computer Graphics and Applications 22*, 1 (2002), 50–58. 2

[Sku04] SKUPIN A.: The world of geography: Visualizing a knowledge domain with cartographic means. *Proceedings of the National Academy of Sciences 101*, suppl 1 (2004), 5274–5278. 2

[SLK*09] STREIT M., LEX A., KALKUSCH M., ZATLOUKAL K., SCHMALSTIEG D.: Caleydo: connecting pathways and gene expression. *Bioinformatics 25*, 20 (2009), 2760–2761. 2

[SSK16] SAKET B., SCHEIDEGGER C., KOBOUROV S.: Comparing Node-Link and Node-Link-Group Visualizations From An Enjoyment Perspective. *Computer Graphics Forum 35*, 3 (2016), 41–50. 3

[SSKB14] SAKET B., SIMONETTO P., KOBOUROV S., BÖRNER K.: Node, Node-Link, and Node-Link-Group Diagrams: An Evaluation. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2231–2240. 2

[SSKB15] SAKET B., SCHEIDEGGER C., KOBOUROV S. G., BÖRNER K.: Map-based Visualizations Increase Recall Accuracy of Data. *Computer Graphics Forum 34*, 3 (2015), 441–450. 3

[TKE12] TARAWANEH R. M., KELLER P., EBERT A.: A General Introduction To Graph Visualization Techniques. In *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering - Proceedings of IRTG 1131 Workshop* (Dagstuhl, Germany, 2012), vol. 27 of *OpenAccess Series in Informatics (OASIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 151–164. 1

[TSW*07] TORY M., SPRAGUE D. W., WU F., SO W. Y., MUNZNER T.: Spatialization Design: Comparing Points and Landscapes. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1262–1269. 2

[VBW17] VEHLOW C., BECK F., WEISKOPF D.: Visualizing Group Structures in Graphs: A Survey. *Computer Graphics Forum 36*, 6 (2017), 201–225. 2

[vLdL03] VAN LIERE R., DE LEEUW W.: GraphSplatting: visualizing graphs as continuous fields. *IEEE Transactions on Visualization and Computer Graphics 9*, 2 (2003), 206–212. 2

[vLKS*11] VON LANDESBERGER T. T., KUIJPER A., SCHRECK T., KOHLHAMMER J., VAN WIJK J. J., FEKETE J.-D., FELLNER D. W.: Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum 30*, 6 (2011), 1719–1749. 2, 11

[WSA*16] WANG Y., SHEN Q., ARCHAMBAULT D., ZHOU Z., ZHU M., YANG S., QU H.: AmbiguityVis: Visualization of Ambiguity in Graph Layouts. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 359–368. 2

[WT08] WU Y., TAKATSUKA M.: Visualizing Multivariate Networks: A Hybrid Approach. In *Proceedings of the IEEE Pacific Visualization Symposium* (Kyoto, Japan, Mar. 5–7 2008), PacificVis '08. 2

[WT17] WANG C., TAO J.: Graphs in Scientific Visualization: A Survey. *Computer Graphics Forum 36*, 1 (2017), 263–287. 2

[WTP*95] WISE J. A., THOMAS J. J., PENNOCK K., LANTRIP D., POTTIER M., SCHUR A., CROW V.: Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proceedings of the IEEE Symposium On Information Visualization* (Atlanta, GA, USA, Oct. 30–31 1995), InfoVis '95, pp. 51–58. 2

[XChHT07] XU K., CUNNINGHAM A., HEE HONG S., THOMAS B. H.: GraphScape: integrated multivariate network visualization. In *Proceedings of the 6th International Asia-Pacific Symposium on Visualization* (Sydney, NSW, Australia, Feb. 5–7 2007), APVIS '07, pp. 33–40. 2

[Zan] ZANINOTTO F.: Codeflower source code visualization. https://www.redotheweb.com/CodeFlower. [accessed 2019-12-03]. 6

[ZWP17] ZHANG Y., WANG Y., PARTHASARATHY S.: Visualizing Attributed Graphs via Terrain Metaphor. In *Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada, 2017), KDD '17, ACM, pp. 1325–1334. 2