Layer-Centered Approach for Multigraphs Visualization

Denis Redondo, Arnaud Sallaberry, Dino Ienco, Faraz Zaidi, Pascal Poncelet

Layer-Centered Approach for Multigraphs Visualization

Denis Redondo*, Arnaud Sallaberry[†], Dino Ienco[‡], Faraz Zaidi[§] and Pascal Poncelet*
* *LIRMM - Université de Montpellier, France*
Email: firstname.lastname@lirmm.fr
[†] *LIRMM - Université Paul Valéry, Montpellier, France*
Email: arnaud.sallaberry@lirmm.fr
[‡] *IRSTEA - UMR TETIS, Montpellier, France*
Email: dino.ienco@irstea.fr
[§] *City University of New York, USA - KIET, Karachi, Pakistan*
Email: faraz@pafkiet.edu.pk

***Abstract*—Recent advances in network science allows the modeling and analysis of complex inter-related entities. These entities often interact with each other in a number of different ways. Simple graphs fail to capture these multiple types of relationships requiring more sophisticated mathematical structures. One such structure is multigraph, where entities (or nodes) can be linked to each other through multiple edges.**

In this paper we describe a new method to manage multiple types of relationships existing in multigraphs. Our approach is based on the concept of pair of nodes (edges) and, in particular, we study how nodes on different layers interact which each other considering the edges they share. We propose a two level strategy that summarizes global/local multigraph features. The global view helps us to gain knowledge related to the characteristics of layers and how they interact while the local view provides an analysis of individual layers highlighting edge properties such as cluster structure. Our proposal is complementary to standard node-link diagram and it can be coupled with such techniques in order to intelligently explore multigraphs. The proposed visualization is tested on a real world case study and the outcomes point out the ability of our proposal to discover patterns present in the data.

***Keywords*-Graph Visualization; Multi-graph; Graph with multiple edge sets**

I. Introduction

Nowadays relational information is ubiquitous and complex structures are required to manage and mine this data. For instance, if we consider social interactions among two people that communicate with each other, they can connect through a variety of different media such as emails, telephones, online messengers. Similarly a user can have accounts on different online social networks. Each online network can have a specific semantic, for example (*LinkedIn* is related to work, *Facebook* to leisure, *Last.FM* to music). If these semantically different social networks are aggregated, ideally we would want to preserve these distinct links in order to better understand connectivity patterns and extract knowledge. Unfortunately, simple graphs fail to represent these semantically different relations and they are not well suited to model complex relationships.

These scenarios require a relational structure to support the representation of multiple relations among entities. One such structure is the multigraph, which is defined as a graph, with the additional feature that more than one edge can exist among the same pair of nodes. Semantically, multiple edges can be interpreted as social interactions among individuals through different channels, or layers. For example, an interaction layer can represent connectivity through *Facebook*, another layer can represent the connectivity through *LinkedIn* and so on. Furthermore, weights can be assigned to these edges and a weight of zero can be interpreted as the absence of such a connection.

Currently, standard node-link diagrams are not well suited for a number of different analysis tasks for multigraphs. For instance, it is hard to visually describe what is the weight distribution of a particular edge (or a group of edges) over the different layers. Additionally, other practical scenarios involving multigraphs can hardly be solved by classical visualization methods and thus motivate us to develop a new visualization technique for weighted multigraphs. The proposed visualization provides two types of data views: (i) a global view provides information related to layer characteristics and how different layers interact (ii) a local view to analyse each layer in terms of its edges. These two views are complementary to standard graph visualizations and they can be coupled with node-link diagrams in order to improve multigraph exploration.

The remainder of this paper is organized as follows. Section II lists the different tasks that can be achieved by the proposed visualization. In Section III we formally define a multigraph. The new visualization technique is introduced in Section IV while qualitative analyses on a real world case study are discussed in Section V. Section VI highlights possible future research directions and improvements. We briefly discuss related literature in Section VII. Finally, Section VIII concludes our contribution.

1550-6037/15 $31.00 © 2015 IEEE
DOI 10.1109/iV.2015.20

II. Tasks

Common approaches in graph visualization are mainly based on node-link diagrams [1]. This representation highlights well the topological properties of simple graphs but it is not enough to model some aspects of multigraph such as multiple edges between nodes. In the case of multigraph, where different types of edges can represent different layers of interaction, one of the key aspect to investigate is the correlation among the different layers: *How much is the overlap between two different layers in terms of nodes? Which layers are used for interaction between two specific nodes?* Such examples of questions are relevant in the case of multigraphs. Solving them by standard node-link diagrams is tedious and interactive techniques (filters, color mappings, etc.) available on most graph visualization systems such as *Tulip* [2], [3], *Gephi* [4] or *Cytoscape* [5] must be used. In this paper, with the aim to simplify the knowledge extraction process, we propose a dedicated visualization to deal with these questions.

Saket *et al.* [6] supply a task taxonomy of group-level graph visualization that categorize the tasks in four subcategories: group only tasks, group-node tasks, group-link tasks and group-network tasks. In the case of multigraphs, specific tasks concern groups of links over the different layers (group-link tasks). Following the categorization proposed in [6], we define a set of *Global* and *Edge Layer Centered (or simply, local)* tasks for multigraph visualization:

- **Global - (Task 1)**: *1.1*) Evaluate the size of each layer in terms of number of edges *1.2*) Evaluate the number of edges specific/common for a particular layer. *1.3*) Given a layer, estimate the distribution of shared edges among the other layers.
- **Edge Layer Centered - (Task 2)**: *2.1*) Evaluate the number of times an edge appears in all layers. *2.2*) Evaluate the distribution, in term of layers, for a specific edge. *2.3*) Compare and cluster similar edges inside a layer.

In order to make the connection between the proposed tasks and the ones presented in [6], we can underline that *Task 1.1* is related to both *Count the number of links in a given group* and *Find the group with the maximum (minimum) number of links*, *Task 2.2* and *Task 2.3* deal with *List groups which contain a link with specific characteristics* while *Task 1.2* casts the task *Find the group with the smallest (largest) number of links connecting it nodes outside the group* for multigraph visualization. We refer the interested to [6] in order to access the whole set of group-level visualization tasks.

In the following sections, we describe our proposal that is specifically developed to solve the tasks related to both global and local multigraph visualization. In particular, we design a data transformation and visual mappings that effectively deal with the proposed tasks.

III. Preliminaries

In this section we introduce the definition of weighted multigraph and the data transformation we developed in order to perform our pairwise analysis .

A weighted multigraph $\mathcal{G}$ is defined as a triple $(V, \{E_i\}_{i=1}^{|C|}, \{L_i\}_{i=1}^{|C|})$ where V is the set of nodes, C are the possible edge layers, $E_i : V \times V$ is the set of edges of layer i and L_i is a function $L_i : E_i \to \mathbb{R}$ that given an edge of layer i returns its weight.

A weighted multigraph $\mathcal{G}$ is the basic input to our data transformation. Given a weighted multigraph $\mathcal{G}$ a transformed data graph $\mathcal{G}^t$ is derived. The new data structure is defined as $\mathcal{G}^t = (C^t, E^t, W^t)$ where C^t is the set of meta-nodes, E^t is the set of meta-edges and W^t returns the weight of a meta-edge.

More precisely, each meta-node of C^t is a one to one mapping to an edge layer of $\mathcal{G}$, this mapping implies that $|C^t|$ is equal to $|C|$. In the following work, both the terms edge-layer and meta-nodes, convey the same meaning. The meta-edge set E^t of the transformed graph is defined as $E^t : C \times C$ where C are the layers of the original graph $\mathcal{G}$. The weight between two meta-nodes in $\mathcal{G}^t$ is the number of edges that two layers share in the original weighted multigraph $\mathcal{G}$. More formally, given $c_j, c_l \in C^t$, $W^t((c_j, c_l)) = |\{(v_m, v_k) \in V \times V | (v_m, v_k) \in E_j, (v_m, v_k) \in E_l\}|$.

We can also define an auxiliary function PW over a node of $\mathcal{G}^t$ that returns the set of edges of the corresponding edge layer in $\mathcal{G}$. Formally, we can define $PW(c_j) = \{(v_m, v_k) \in V \times V | (v_m, v_k) \in E_j\}$.

IV. Visual Mapping

Our visual mapping is based on two complementary views. The first one is based on a global visualization that solves Tasks 1. The second one is an edge layer centered view that solves Tasks 2.

A. Global View

Figure 1 shows an example of this first view. Edge layers $c_i \in C^t$ are represented as arcs forming a ring. Their length along the ring is proportional to the number of edges they share with any other layers, which is equal to $|\{e \in E_i | \exists j \neq i, e \in E_j\}|$ for the layer c_i. In the example, the red layer shares twice as many edges as the blue one. The width of the arc associated to layer c_i represents $|c_i|$, i.e. the number of edges it contains (**Task 1.1**). White and colored areas of the arcs are respectively proportional to the number of shared and non-shared edges of the considered layer (**Task 1.2**). In the example, the red layer shares half of its edges while the blue one shares all its edges. Meta-edges from E^t are represented by curved ribbons between arcs. Their sizes and their colors are proportional to the relative number of shared edges between the two meta-nodes they link, i.e. $W^t(c_i, c_j)$ for two meta-nodes c_i and c_j (**Task 1.3**). In the example, the red layer shares 50% of its edges with the

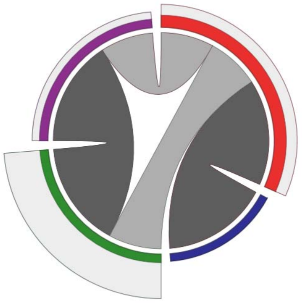


Figure 1. Global View. Arcs are layers, size of ribbons represents the number of edges shared between layers.

blue layer while the rest of the edges are shared between both purple and green layer with the same percentage (25% each). Continuing with the same example, we can see that the blue layer shares all of its edges with the red layer. For each layer, meta-edges are ordered in the opposite direction as the target meta-nodes appear along the ring. This is done to reduce edge crossing.

B. Edge Layer Centered View

This view is centered on an edge layer $c_k \in C^t$ (called focal layer hereafter). c_k is represented by a large circle. Figure 2(a) shows an example centered on the red layer. Other layers are represented by arcs forming a ring around the circle. In the example, we observe three other layers: purple, blue and green. Their length along this ring is proportional to the number of edges they contain, i.e $|c_i|$ (**Task 1.1**). The green layer in Figure 2(a) contains more edges than the purple and blue layers. A ribbon links the focal layer with the other ones. Its width depends on the number of shared edges between the external layer and the focal one, that is $W^t(c_k, c_i)$ with c_k the focal layer and c_i the external one. This mapping helps to perform what can be considered as a subtask of **Task 1.2**. In the example, we see that the blue layer shares all its edges with the red layer, the width of the ribbon is the same as the length of the blue arc. We also see that green and purple layers share a similar amount of edges with the red layer. Instead, considering a relative proportion, we observe that the green layer only shares a small part of its edges (the green ribbon is very small w.r.t. the corresponding arc) while this ratio is bigger for the purple one.

The circle of the focal layer c_k (the red layer in the example) contains pie charts representing the elements of $PW(c_k)$, that is the set of edges of the edge layer c_k. The size of a pie chart depends on the sum of the weights of this edge in the different layers: let e be an element of $PW(c_k)$, then $size(e) = \sum_{i=1}^{|C|} L_i(e)$ (**Task 2.1**). The pie chart of an edge represents the distribution of its weights among all the layers (**Task 2.2**). In the example, we see that the pie chart highlighted in Figure 2(b) represents an edge with a high weight in the blue layer, a low weight in the red one and it does not appear in the purple and green layers.

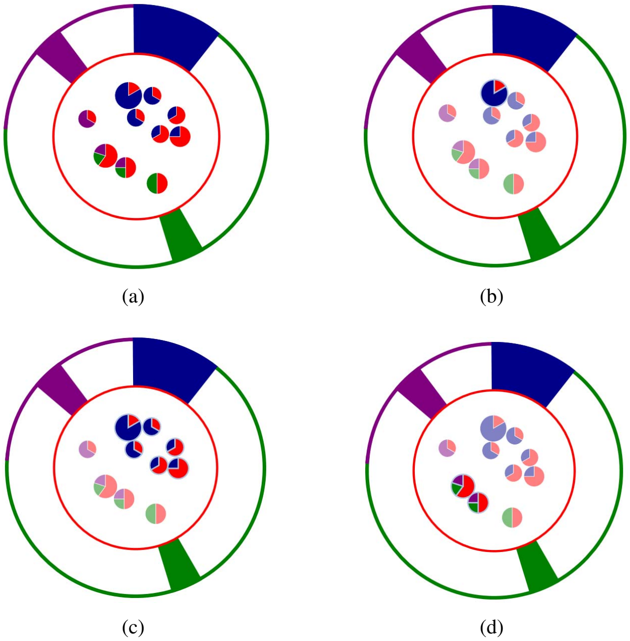


Figure 2. Edge Layer Centered View. The focal layer is represented by the red circle. Pie charts represent edges and their distribution among layers. Arcs forming a ring around the red circle correspond to the other layers. The size of ribbons linking these arcs to the focal layer is proportional to the number of edges they share.

Pie charts are organized inside the circle to locate together edges having similar distributions of their weights among the layers (**Task 2.3**). We first position them at the barycenter of the edge layers they belong to. Then, for each pair (e_i, e_j) of $PW(c_k)$, we compute a distance that corresponds to their cosine similarity:

$$dist(e_i, e_j) = \frac{v(e_i).v(e_j)}{||v(e_i)||.||v(e_j)||}$$

where $v(e_i)$ is a vector containing $L_l(e_i)$ for all the edge layers l. We then apply a force directed layout[1] on a graph where the set of nodes is $PW(c_k)$ and the edges are pairs (e_i, e_j) having the lowest distances (these distances are mapped to the lengths of the springs). Figure 2(a) shows how similar edges are grouped together producing a visual clustering. We clearly observe 5 clusters. The two clusters highlighted in Figure 2(c) are closer to each other because corresponding edges belong to the same layers.

In order to improve the visualization, before running the force directed algorithm, we also add to the dedicated graph a static node for each external layer c_i at the middle of the corresponding arc. After that, each static node is connected to all the other nodes of the dedicated graph. That's why in

[1] https://github.com/mbostock/d3/wiki/Force-Layout [Online; accessed 12-Mars-2015]

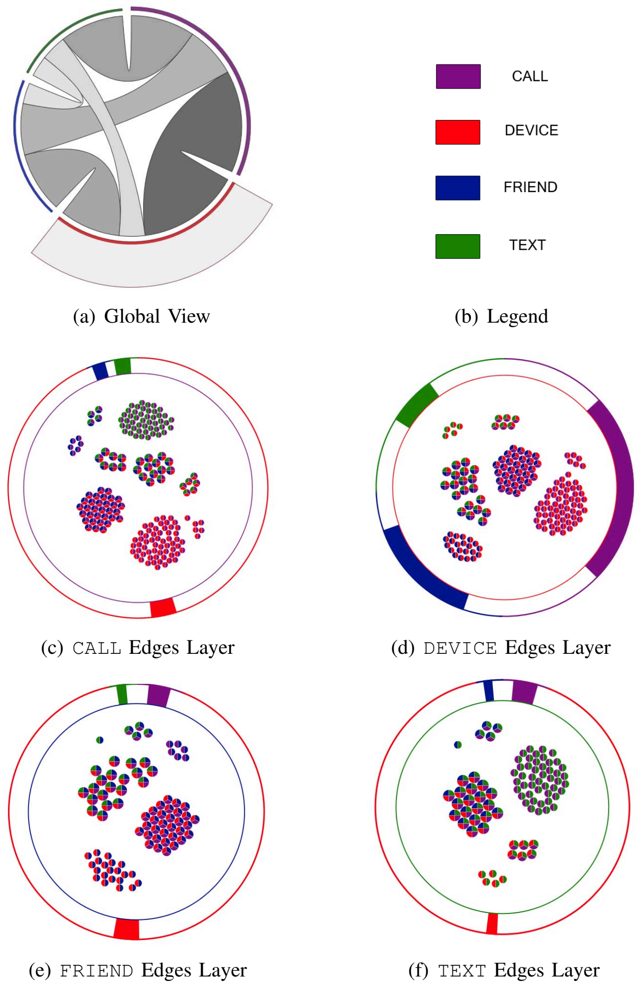
Figure 2(c), the highlighted pie charts are close to the blue ribbon while the highlighted pie charts in Figure 2(d) lie far from all the ribbons.

V. Case Study

In this section we illustrate one case study that shows the practical benefits of our method to visualize multigraph data. The case study investigates social interaction among people who communicate through different media. Figure 3 shows the *Reality Mining* dataset[2] This multigraph contains data collected by the MIT Media Lab, including subjects (undergraduate and graduate CS and business students) whose interactions were monitored by a pre-installed piece of software on their mobile devices. The experiment was carried out on a total of 94 people and this also represents the number of nodes in the corresponding multigraph. The different layers offered by the dataset pertain to the means of interaction between a pair of subjects. Namely, `CALL` layer refers to subjects calling each other, `DEVICE` layer contains Bluetooth device scans, `FRIEND` layer contains friendship claims and `TEXT` layer is constructed based on text message exchanges (SMS). Each layer has, respectively, 177, 3003, 82 and 96 edges.

The *Global View* supplied by our approach is sketched in Figure 3(a). We can note that our visualization easily underlines differences between layers. For instance we can see that `DEVICE` is the most popular way to communicate while `CALL` is the second one. According to this global view, we can rank the layers by considering their arc size. This information can be exploited to accomplish **Task 1.1**. By investigating this view we can highlight other useful information. For instance, we can say that only a small percentage of people using bluetooth (`DEVICE`) uses SMS (`TEXT`) too. Conversely, people making `CALL` are also usually send SMS to each other. This kind of analysis fulfills **Task 1.3**. As additional feature, we can also evaluate how much information is shared between layers (**Task 1.2**). For example, we can observe that most of the people using bluetooth (`DEVICE` layer) mainly interact with this media while, only a small portion of them employs other kind of communication.

The *Edge Layer Centered Views* are shown in Figures 3(c), 3(d), 3(e) and 3(f) (respectively `CALL`, `DEVICE`, `FRIEND` and `TEXT`). Each *Edge Layer Centered View* allows to locally describe how the data are distributed. Considering all the different layers, we can see that our approach automatically regroups interactions according to the layers in which they appear. This organization provides a general picture about the internal distribution of a layer. This feature can be used to accomplish **Task 2.3**. Inspecting the local view for the `FRIEND` layer, we can visually depict 6 clusters with different behavior. Most of the piecharts (representing the interactions) have at least three colors. In the top left part our system is able to isolate an interaction that clearly represents an anomalous profile and the two persons involved claim that they are friends but they only communicate with SMS. Another interesting group is constituted by red/blue interactions in the bottom part of the `FRIEND` *View*. These ones highlight people that claim to be friend and really call each other. All these points help to realize **Task 2.2**.



(a) Global View

(b) Legend

(c) `CALL` Edges Layer

(d) `DEVICE` Edges Layer

(e) `FRIEND` Edges Layer

(f) `TEXT` Edges Layer

Figure 3. Reality Mining dataset

VI. Discussions

In this section we discuss some possible improvements, directions and scalability aspects of the proposed visualization.

Figure 4 shows one common drawback of visualization techniques exploiting color as visual variable. The number of employed colors is directly proportional to the number of layers. When the number of layers increases the number of used colors increases. Previous studies [7] agree on the

[2] http://realitycommons.media.mit.edu/realitymining.html [Online; accessed 12-March-2015]

difficulty to use more than 8 easily distinguishable colors. This fact negatively impacts on the visual result decreasing the usefulness of such exploration tools. In our case, this phenomenon can influence both global (the arcs associated to the layers) and local (the displayed pie chart distribution) visualizations. We are aware of this issue, but, on the other hand, recent works [8] on real multigraph data analysis show that such kind of networks often contain less than 8 layers. This number of layers can be easily managed by our approach.

Figure 5 shows an edge-layer visualization of a multigraph bigger than the ones employed in the case studies. We can observe that also in this example our method is able to visually depict the cluster structure. In the case where the number of edges grows too much, we could also introduce an interactive exploration system for the local view inspired by the ones proposed in [9], [10]. More in detail, firstly we can perform a hierarchical clustering on the pie charts and then allow to navigate over the different granularity levels supplied by the hierarchical organization. At each level of granularity the cluster can be replaced by a pie chart obtained as the average of the original pie charts belonging to that group. In this way our system could easily scale up to multigraphs containing thousands of edges per layer. We leave this amelioration as a possible future work direction.

Technically, our proposal has been developed using the D^3 library [11]. This tool is a data-driven Javascript library that supplies visualization components and interaction capabilities in order to develop web based prototypes.

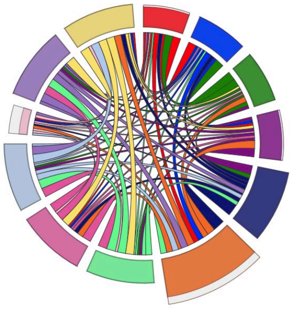


Figure 4. Common limits supplied by the use of colors as visual variable

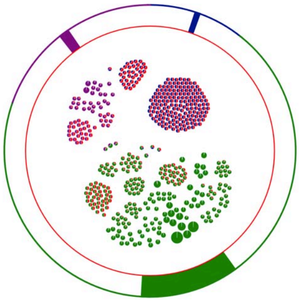


Figure 5. Visual scalability of Edge Layer View in terms of number of edges

VII. Related Work

Although graph visualization techniques have become a popular way to analyze networks, little effort has been made to develop techniques that support visualization of particular aspects of multigraphs. Furthermore, the focus of these works is not to highlight the different pair-wise interactions as compared to what we have proposed in this paper.

For example, [12] proposes a system to visualize multiple relationships among nodes by iterating between link types keeping the spatial layout of nodes stable. This type of visualization is useful to study relationships of nodes in certain layers but has limitations if relationships across different layers need to be studied at the same time. The authors also propose a matrix visualization which summarizes relationships of different link types among individuals to discover patterns. The visualization is effective for small size networks but it becomes very difficult to find patterns in large networks.

A recent work [13] discusses a number of different well-known visualization techniques and methods to help analyze complex networks. One of these methods is particularly suited for networks containing multiple edges which is parallel coordinates. Each column in the parallel coordinates view corresponds to a replica of nodes, one for each edge set to help analyze different relationship sets in a network. Different from this work, we used a ring based visualization to improve space utilization and we also supply a local view (edge layer centered) devoted to inspect each layer.

The problem of tracing edge paths and rendering them easily readable has been well addressed by [14], although in a completely different context. The authors present a new visual representation for state transition graphs. The focus is to uniquely visualize higher-order state transitions. They use smooth curved lines to make it easier to trace state transitions. Edges are encoded with animated textures to represent different types of links and the orientation of edges to enhance their representation. On similar lines, [15] explore the use of link curvature proposing four families of techniques. The authors used different visual encodings from these four families to facilitate the visual representation and interaction with multiple edges. They also suggested extensions to existing techniques to further ameliorate their representation.

Hybrid visualization techniques have also been introduced to visualize large instance sets with multiple relations. One such technique [16] combines the classical node-link diagrams and the adjacency matrix representation to visualize different parts of the data. The technique uses different glyphs to encode multiple types of links making it easier to visualize different relations.

Elzen and van Wijk recently proposed a tool for exploring multivariate graphs. It combines two views, a detail view to select areas of interest and an high-level infographic-style overview of the selected areas [17]. The originality of this approach lies to start from the detail view instead of global one as usually done following the strategy proposed by Shneiderman's mantra [18]. The resulting tool combines node-links diagrams with standard diagrams (scatter plots, histograms, treemaps and parallel-coordinate plots).

In this paper, we address a different set of objectives and thus, clearly differ from the previously existing works. A good recent resource highlighting the problems and state of the art research for multivariate graphs is [19]. This resulted from a seminar where well known researchers from the graph visualization domain discussed several issues pertaining to the problem of visualizing multivariate graphs.

VIII. Conclusion

In this paper we have presented a new approach to visualize multigraph layers. Our proposal showed ability to capture both global and local behaviors of the underlined data. We demonstrated the quality of our proposal on a real world dataset highlighting how our visualization can accomplish tasks described in Section II that common graph based visualization strategies fail to solve. Our proposal is complementary to standard graph visualization techniques (such as node-link diagrams). As future work, we plan to investigate those techniques that can be coupled with our approach to develop a complete exploration system for multigraph data.

Acknowledgment

This project was funded through the French National Research Agency (ANR) project ANR 11 MONU 14 FRESQUEAU.

References

[1] R. Tamassia, *Handbook of graph drawing and visualization*. Chapman & Hall/CRC, 2007.

[2] D. Auber, "Tulip - a huge graph visualization framework," in *Graph Drawing Software*, ser. Mathematics and Visualization Series, P. Mutzel and M. Jnger, Eds. Springer Verlag, 2003.

[3] D. Auber, D. Archambault, R. Bourqui, A. Lambert, M. Mathiaut, P. Mary, M. Delest, J. Dubois, and G. M. con, "The tulip 3 framework: A scalable software library for information visualization applications based on relational data," INRIA Bordeaux Sud-Ouest, Tech. Rep. RR-7860, 2012.

[4] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," in *Proceedings of the Third International Conference on Weblogs and Social*, 2009.

[5] P. Shannon, A. Markiel, O. Ozier, N. Baliga, J. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome Research*, vol. 13, no. 11, pp. 2498–2504, 2003.

[6] B. Saket, P. Simonetto, and S. Kobourov, "Group-level graph visualization taxonomy," in *Proceedings of the Eurographics/IEEE VGTC Symposium on Visualization (EuroVis)*, 2014, pp. 85–89.

[7] C. Ware, *Information visualization : Perception for design*. Morgan Kaufmann, 3rd edition, 2013.

[8] F. Bonchi, A. Gionis, F. Gullo, and A. Ukkonen, "Distance oracles in edge-labeled graphs," in *Proceedings of the 17th International Conference on Extending Database Technology (EDBT)*, 2014, pp. 547–558.

[9] F. V. Ham and J. J. V. Wijk, "Interactive visualization of small world graphs," in *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, 2004, pp. 199–206.

[10] J. Abello, F. van Ham, and N. Krishnan, "ASK-GraphView: A large scale graph visualization system," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 669–676, 2006.

[11] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.

[12] A. Perer and B. Shneiderman, "Balancing systematic and flexible exploration of social networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 693–700, 2006.

[13] T. Crnovrsanin, C. Muelder, R. Faris, D. Felmlee, and K.-L. Ma, "Visualization techniques for categorical analysis of social networks with multiple edge sets," *Social Networks*, vol. 37, pp. 56–64, 2014.

[14] J. Blaas, C. P. Botha, E. Grundy, M. Jones, R. S. Laramee, and F. H. Post, "Smooth graphs for visual exploration of higher-order state transitions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 969–976, 2009.

[15] N. H. Riche, T. Dwyer, B. Lee, and S. Carpendale, "Exploring the design space of interactive link curvature in network diagrams," in *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI)*, 2012, pp. 506–513.

[16] B. Bach, E. Pietriga, and I. Liccardi, "Visualizing populated ontologies with ontotrix," *International Journal On Semantic Web and Information Systems (IJSWIS)*, vol. 9, no. 4, pp. 17–40, 2013.

[17] S. Elzen and J. van Wijk, "Multivariate network exploration and presentation: From detail to overview via selections and aggregations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2310–2319, 2014.

[18] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings of the 1996 IEEE Symposium on Visual Languages (VL)*. IEEE Computer Society, 1996, pp. 336–343.

[19] A. Kerren, H. Purchase, and M. O. Ward, "Information visualization-towards multivariate network visualization (dagstuhl seminar 13201)," *Dagstuhl Reports*, vol. 3, no. 5, 2013.