Exploring a Multi-focus Visual Comparison of Dynamic Graphs

Hagen Tarner *
University of Duisburg-Essen, Germany

Ivan Beschastnikh †
University of British Columbia, Canada

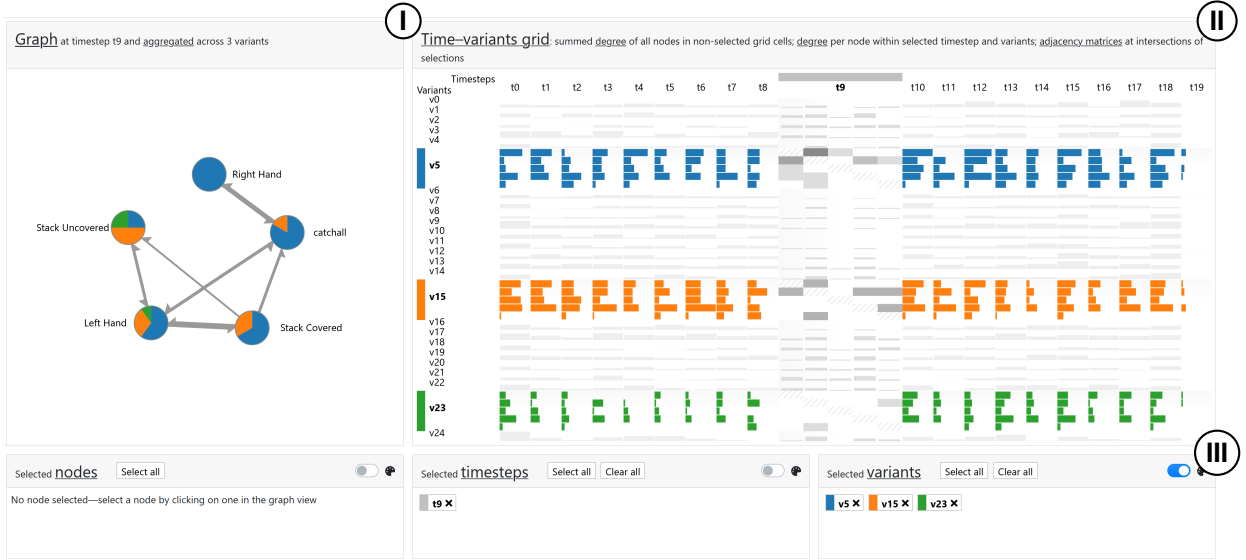Fabian Beck ‡
University of Duisburg-Essen, Germany

Figure 1: Comparing variants of dynamic graphs: (I) Aggregated Graph View shows a node-link diagram of an aggregated graph (here, across three variants in one timestep), (II) Time–Variants Grid shows all variants and timesteps at different levels of aggregation as a multi-focus table with adjacency matrices in focused cells, and (III) selection panels.

ABSTRACT

Visualizing dynamic graphs can explain how relationships between objects evolve over time. We generalize this to visually comparing variants of evolving relationships of the same set of objects. Our visualization is based on a tabular representation that shows time on one axis and different dynamic graph variants on the second axis. The option to set multiple focus points in the table supports exploration of graph details. Node-link diagrams are used for an aggregated overview while adjacency matrices show the individual graph details in selected table cells.

1 INTRODUCTION

Changing relationships and state transitions can be represented as dynamic graphs for versatile applications, such as social network analysis, software engineering, or eye tracking. In a common formal model, a dynamic graph forms a sequence $G := (G_1, G_2, \ldots, G_k)$ of static graphs $G_i := (V, E_i)$, where V is a common set of nodes (or vertices) and $E_i \subseteq V \times V$ is a (multi-)set of (directed) edges connecting pairs of nodes (note that the usage of multi-set for edges allows copies of the same edge and edges may or may not be directed). Various techniques exist for visualizing such graphs with animations or timelines [2]. We aim to study the more general scenario in which people compare different variants of dynamic graphs. This scenario is important, for instance, to evaluate the difference between similar executions of a software system or eye movements of different

*hagen.tarner@paluno.uni-due.de
†bestchai@cs.ubc.ca
‡fabian.beck@paluno.uni-due.de

participants in a user study. We define the different variants as a set of (directed) dynamic graphs $\mathscr{G} := \{G^1, G^2, \ldots, G^l\}$, each having k timesteps and being defined on the set of nodes V. Only a few approaches have so far indirectly addressed this problem, for instance, partly aggregating the temporal dimension [3], clustering similar states in a dynamic graph [1, 6], and self-comparing a dynamic graph [4]. A challenge for comparing dynamic graphs is managing the complexity of the data along nodes, edges, and time, as well as the different variants of the changing graphs.

In this work, we propose a novel visualization technique for the comparison of dynamic graphs. It addresses the above challenge using a node-link representation as an overview and a table-based multi-focus approach for showing temporal and comparison aspects. A prototype of the proposed approach using *d3.js* is available online at `https://vis.hagen-tarner.de/dgc/`.

2 VISUAL COMPARISON OF DYNAMIC GRAPHS

Fig. 1 overviews our approach: the node-link diagram on the left (I) shows an aggregated representation of several graphs and adapts to the current selection of graph entities. On the right (II), the multi-focus table view—inspired by *Table Lens* [5]—shows different graph variants in rows and timesteps in columns. The user can focus on multiple rows and columns independently, as in the original *Table Lens* approach. Cells in focus show an adjacency matrix of the respective graph at the respective timestep. Unfocused or partly focused cells display aggregate information in the form of bar charts depicting the number of edges of the respective graph. Finally, the bottom panel (III) provides information on the current selection of nodes, timesteps, and variants.

The Multi-focus Selection Concept of our approach allows users to independently focus multiple entities from the three categories: *nodes* $V' \subseteq V$, *timesteps* $I' \subseteq I = \{1, 2, \ldots, k\}$, and *variants*

$J' \subseteq J = \{1, 2, \ldots, l\}$. This is done by clicking a node in the Aggregated Graph View (node selection), a column header (timestep selection), a row header (variant selection), a grid cell (simultaneous selection of timestep and variant), or some combination of these. Selected entities are listed in the respective Selection Panel (III). Entity selection also works as a filter mechanism. In its initial state the application shows different aggregations of all nodes, timesteps, and variants in the dataset. Upon selection only the edges that are connected to the selected entities are visualized. The Selection Panels are also used to control the colors, which are used to support the comparison of selected entities. The color mode can be set exclusively to one of the three categories. If color mode is set to timesteps or variants, the nodes in the Aggregated Graph View are replaced with pie charts showing the distribution of edge counts (incoming and outgoing) in the selected timesteps/variants.

The Aggregated Graph View contains a node-link diagram that shows, by default, an aggregation of all graphs in the whole dataset (i.e., the *supergraph*, a union across all edge sets in all timesteps and variants: $\cup_{i,j} E_i^j$ where $G_i^j = (V, E_i^j) \in G^j$ and $G^j \in \mathscr{G}$). A force-directed layout is initially computed for the supergraph and the layout remains unchanged during usage to not break the user's mental map. The number of aggregated edges connecting two nodes is visualized using stroke thickness. If edges are directed, we use half-drawn arrow heads to indicate the source and target of an edge. Edges will be hidden if not present in the selected timestep/variant.

The Time–Variants Grid is a highly-interactive component with a multi-focus concept inspired by *Table Lens* [5]. Each timestep in the dataset forms one column in the grid, and each variant one row. Grid cells are displayed depending on their selection state; the cell in the i-th column and j-th row shows a meaningful level of aggregation of the respective graph $G_i^j \in G^j$ where $G^j \in \mathscr{G}$ (i.e., i-th timestep, j-th variant). If neither timestep nor variant of the cell is selected, the cell is filled with a single vertical bar, depicting the number of incoming and outgoing edges, summed over all nodes. In case either timestep or variant of the cell is selected, the cell gets filled with one bar per node of the graph. The bar then encodes the same metric of incoming and outgoing edges, but it is only aggregated per node. If the timestep *and* variant of the cell are both selected, the cell is filled with an adjacency matrix encoding the number of edges connecting each combination of two nodes for this timestep and variant as a monochrome fill color.

To describe each of the components of the visualization we use dynamic captioning. Generated from templates, these captions respond to selection and describe the current state of the application in a meaningful way. To aid users, the captions provide a textual summary of the current selection and explain basic terminology via tooltips on underlined terms.

3 APPLICATION EXAMPLE

To demonstrate our approach we use eye-tracking data from showing a video (a static top view of two players playing a card game) to participants. This eye tracking dataset contains 25 graph variants (1 per participant), 20 timesteps (each timestep is an aggregation of 158 frames of the original video material), and five nodes (the *Areas of Interest* (AOI) that identify the main objects in the video). Edges within one timestep and variant describe the eye movements a participant made between two AOIs.

Now to compare the eye movement of different participants at one timestep, we first load the dataset into the application, and then select timestep *t9*, and participants *v5*, *v15*, and *v23*, based on the observed patterns in the Time–Variants Grid. To support the comparison of participants, we set the color mode to *variants*. The resulting application state can be seen in Fig. 1.

The Aggregated Graph View gives an overview, of where each participant looked at in this timestep. While *v5* looked at each of the five AOIs, *v15* did not look at the *Right Hand* AOI, and *v23*

concentrated on the *Stack Uncovered*, and *Left Hand* AOIs. The most frequented AOI pair is the connection between *Left Hand* and *Stack Covered*. An interpretation of this could be, that the left player in the video had to draw multiple cards from the stack.

Further analysis of the dataset reveals overall low activity for *v23*, when compared to other participants. The dataset contains multiple timesteps, in which *v23* did not look at all available AOIs, but instead focused on only a subset of AOIs. This could indicate a level of attention to detail, that other participants lack.

4 DISCUSSION AND FUTURE WORK

Our approach is designed to support the visual comparison of dynamic graphs. Since the complexity of the input data is large (i.e., nodes, edges, timesteps, and variants), we work with aggregated representations, both in the node-link diagram and the tabular representation. However, details can be accessed on demand using the versatile selection mechanism.

Whereas we support a variety of comparison modes, scalability remains limited. Our running example uses a small graph with 5 nodes, 25 variants, and 20 timesteps. Graphs with more nodes can be shown (see other datasets available in the tool), but one may need to compromise on the number of variants and timesteps. To scale our approach, we plan to explore clustering of nodes, timesteps, and graph variants—clusters can be interactively collapsed to hide unnecessary information.

We have integrated both node-link and adjacency matrix representations: the node-link diagram as an intuitive representation for the enlarged aggregated visualization, where nodes can be colored as a pie chart; the matrix in the small inter-cell representation as the more scalable approach that integrates well with the tabular layout. However, we also want to explore other combinations and layouts, for instance, how small-scale variants of node-link diagrams can be best integrated into the tabular representation.

5 ACKNOWLEDGMENTS

REFERENCES

[1] B. Bach, N. Henry-Riche, T. Dwyer, T. Madhyastha, J.-D. Fekete, and T. Grabowski. Small MultiPiles: Piling time to explore temporal patterns in dynamic networks. *Computer Graphics Forum*, 34(3):31–40, 2015. doi: 10.1111/cgf.12615

[2] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 36(1):133–159, 2017. doi: 10.1111/cgf.12791

[3] T. Blascheck, M. Schweizer, F. Beck, and T. Ertl. Visual comparison of eye movement patterns. *Computer Graphics Forum*, 36(3):87–97, 2017. doi: 10.1111/cgf.13170

[4] M. Burch. The Dynamic Call Graph Matrix. In *Proceedings of the 9th International Symposium on Visual Information Communication and Interaction*, VINCI, pp. 1–8. ACM, 2016. doi: 10.1145/2968220.2968226

[5] R. Rao and S. K. Card. The Table Lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI, pp. 318–322. ACM, 1994. doi: 10.1145/191666.191776

[6] T. von Landesberger, F. Brodkorb, P. Roskosch, N. V. Andrienko, G. L. Andrienko, and A. Kerren. MobilityGraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):11–20, 2016. doi: 10.1109/TVCG.2015.2468111