Eurographics Conference on Visualization (EuroVis) (2015)
E. Bertini, J. Kennedy and E. Puppo (Editors)

Visual Techniques to Support Exploratory Analysis of Temporal Graph Data

N. Kerracher[1], J. Kennedy[1], K. Chalmers[1] and M. Graham[1]

[1]Edinburgh Napier University, United Kingdom

Abstract

Recently, much research has focused on developing techniques for the visual representation of temporal graph data. This paper takes a wider look at the visual techniques involved in exploratory analysis of such data, considering the variety of sub tasks and contextual tasks required to understand change in a graph over time, and the visual techniques which are able to support these tasks. In so doing, we highlight a number of tasks which are less well supported by existing techniques, which could prove worthwhile avenues for future research.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [INFORMATION INTERFACES AND PRESENTATION]: User Interfaces—Graphical user interfaces (GUI)

1. Introduction

Temporal graph visualisation techniques seek to support our understanding of networks that change over time. Much of the work in this area has focussed on the representation of network change, such as the development of dynamic graph layout algorithms, and assessing the performance of different visual encodings. Recent work has classified existing techniques for representing this data [BBDW14] and considered the design space of possible representations [KKC14]. In this paper we review more widely the visual techniques which can support tasks involved in exploratory analysis of temporal graph data. While the goal of analysis is often to simply understand how a graph has changed over time, many smaller sub tasks may be involved, such as comparison of the graph at different times, observing temporal trends in individual data items, or comparing evolution over different time intervals. We may also be interested in contextualising our findings, for example, by comparing the evolution in one part of a graph with that of another. Kerracher et al.'s task taxonomy [KKCss] considers the range of such possible tasks involved in exploring temporal graph data.

This work is influenced by Wehrend and Lewis [WL90], who propose a "problem-oriented approach" to tool classification: they categorise techniques according to the subproblems supported, resulting in a task-technique 'catalogue'. Their approach is intentionally application-domain independent to allow users from different domains to share methods. We consider examples of techniques from both the temporal graph literature and wider research areas for the support of the temporal graph tasks identified in [KKCss], and identify a number of gaps in task support which could benefit from further research.

2. Exploratory Tasks for Temporal Graph Data

The task categories of [KKCss] are based on the Andrienko framework [AA06], which takes a functional approach to specifying tasks. There are two components to every task: the target (unknown information) to be obtained, and the constraints (known conditions) that information needs to fulfil; a task involves finding a target given a set of constraints. Task types are distinguished according to the data items participating as targets or constraints: *lookup* (find the value of a given item or items with a given value), *comparison* (find the relation between two items), and *relation seeking* (opposite of comparison: find data items related in a given way).

[KKCss] further classify tasks in the temporal graph case according to the data items that participate, dividing them into four categories ("*quadrants*", Figure 1): Q1 individual graph elements (nodes, dyads), time points, and corresponding attribute values; Q2 individual time points, graph structures and corresponding attribute distributions; Q3 individual graph elements and corresponding attribute values, over time intervals; Q4 graph structures and corresponding

DOI: 10.2312/eurovisshort.20151133

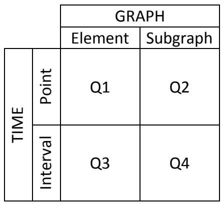

Figure 1: *Four task categories ("quadrants")*

attribute values over time intervals, and distributions of temporal trends over the graph structure.

Time	Graph Component	Attribute
Same	Same	Different
Same	Different	Same or different
Different	Same	Same or different
Different	Different	Same or different

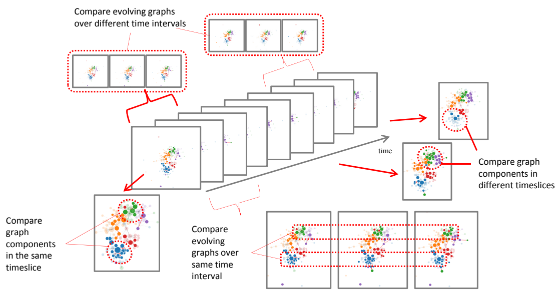

Figure 2: *Variations in comparison tasks according to the involvement of the same or different time and graph components*

Applying the task types in each quadrant produce markedly different tasks e.g. comparison in Q2 involves comparing graph structures or attribute distributions over graph structures; in Q3, comparison of temporal trends in individual attribute values, or patterns in connectivity between dyads; in Q4 comparison of evolving graph structures or distributions of temporal trends over (sub)graphs.

A further task distinction relates to the different targets of analysis. *Direct* tasks concern finding or comparing structural patterns or attribute components (values or patterns) associated with known graph components at known times (e.g. finding the pattern of connectivity between a particular set of nodes at time 1). Conversely, *inverse* tasks compare the times and/or graph components associated with a particular occurrence of a structural pattern or attribute value or pattern (e.g. finding the set of nodes belonging to a particular cluster, or the time period over which a cluster grows). Task *search space* depends on which components are specified; four variations are possible: no search (time and graph components are specified i.e. direct tasks); graph search (time is specified, graph component is not—requires searching the entire graph); temporal search (graph component is specified, time is not—requires searching the entire time period); graph and temporal search (neither component is specified—requires searching the whole graph at all time points).

Further sub-variations of the comparison and relation seeking tasks depend on whether the task involves the same or two different time points/intervals, the same or two different graph objects, and the same or two different attributes; the seven possible combinations are given in the Table in Figure 2 and can be applied in each of the four quadrants, resulting in 28 task variations.

3. Mapping Visual Techniques to Tasks

3.1. Lookup

Direct and inverse lookup tasks require different techniques for their support, as they take a different starting point for the analysis; the distinction reflects the bottom up ("*search, show context, expand on demand*" [vHP09]) and top-down ("*overview first, zoom and filter, then details on demand*" [Shn96]) information seeking approaches discussed more widely in the literature. For direct lookup we must first locate the time and graph object of interest, in order to find the corresponding values and patterns. This requires navigation in both time and in the graph. Many systems employing sequential (animated) views offer temporal navigation via interactive controls such as timesliders, or thumbnails [BPF14b], often in conjunction with some statistical summary information e.g. [ATMS*11]. [CGS*09] allow users to select a time period of interest. Static graph systems, such as TaxVis [GK08] often offer a separate search box and list to filter and then highlight nodes of interest within the graph. [CGS*09] offer this functionality and also highlight in their timeline view the time points at which the selected nodes appear. Interaction techniques such as pan and zoom are also of use when locating graph elements in large graphs.

Inverse lookup tasks involve observing patterns and attribute values and identifying the corresponding graph objects and times of occurrence. Visually representing the four categories of data items involve very different techniques and research areas (Figure 3): Q1 (data elements and their attributes) is governed by general visualisation principles; Q2 is dealt with by static graph visualisation; Q3 is the domain of temporal visualisation; while Q4 is the only quadrant requiring the representation of both time and graph structure, and therefore temporal graph visualisation techniques. However, any of these data items and associated techniques may feature in the exploration of temporal graph data. Within each category, decisions as to the appropriateness of a visual representation will depend on characteristics of the specific dataset; for example, when selecting a technique to encode graph structure (Q2), the size and density of the graph must be taken into consideration; when showing structural change over time (Q4), the rate of change and length of timeseries may influence our choice of representation.

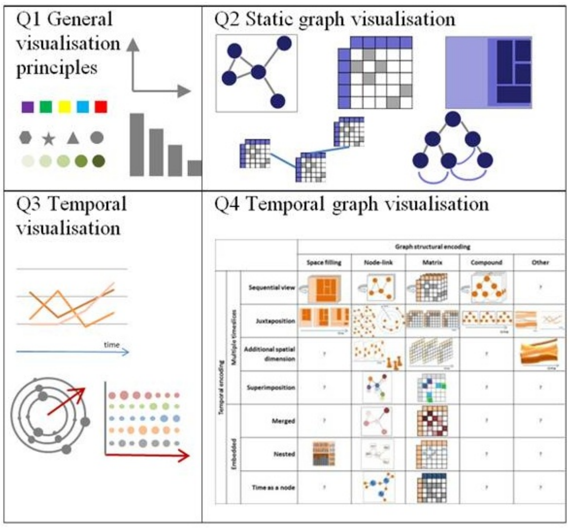

Figure 3: *Research areas and techniques associated with data items by quadrant*

When representing Q4, all of the techniques in Kerracher et al.'s design space [KKC14] show graph evolution over time, with the exception of nested views, which show the distribution of temporal trends over the graph. The timeslice views (sequential, juxtaposed, additional spatial dimension) also show a snapshot of the graph at an individual point in time (i.e. a Q2 representation). In Q2, finding structural patterns such as clusters is supported by the layout algorithm used. In dynamic graph drawing there is a trade-off between local (at each time point) and global (over all time points) layout optimisation; [BPF14b] offer layout stability controls to allow users to optimise the layout, while [PRB08] offer a choice of layout algorithm. Graph filtering techniques can help find graph objects associated with particular attribute values, while interaction techniques including filtering, clustering, grouping, simplification [SD13] and network motif glyphs [DS13] all may help find structural patterns.

For Q3, nested views e.g. [SLN05] [YEL10] show temporal trends of nodes or edges embedded within the graph structure. Temporal trends can also be combined with other representations: TimeFluxes [ITK10] connect the same node in two different timeslices of a 2.5D representation, and display timelines of attribute values for user selected nodes; vertex small multiples [BPF14a] can be selected from a matrix cube view to show connectivity patterns between individual nodes over time. Some systems focus specifically on individual temporal trends: LinkWave [RCM*14] visualises temporal trends in connectivity for all pairs of nodes in a graph, while NetVisia [GGK*11] displays temporal node statistics in a heatmap. Techniques to filter, or reduce timeseries data to reveal temporal patterns of interest e.g. [ARH12] are of potential use here.

Depending on the task search space, only a sub section of time or graph may need to be displayed. Where the search space is time, highlighting or filtering of the graph or set of timeseries can be used to show only the graph object of interest. If the search space is graph, only the time period of interest need be selected and shown e.g. [CGS*09]. Where the search space is both time and graph, showing the whole graph over all timepoints is necessary, and may require interaction techniques to allow users to navigate the whole dataset while searching for patterns of interest.

Once a pattern or value of interest is observed, the corresponding time steps and graph objects must be identifiable. Nodes are often identified using details-on-demand strategies, such as showing labels on mouse-over. Similar labelling strategies can be employed where a timeline is present, or individual timeslices can be time stamped in sequential views e.g. [LS08]. As tasks may be chained, some way of marking found graph items and/or time points for use in subsequent tasks can be supported, for example, nodes of interest are often highlighted via selection mechanisms for tracking over time e.g. [ITK10] [FAM*12] [BPF14b].

3.2. Comparison

Visual techniques appropriate to support comparison tasks depend largely on what we are comparing—graph objects, times, attribute values, structural patterns—as distinguished in the quadrants. Gleicher et al.'s [GAW*11] three basic possibilities for visual comparison—juxtaposition (placing representations side by side), superposition (overlaying representations in the same display space) and explicit encoding (where the relationship between the two items is calculated and explicitly represented)—are applicable to all quadrants. Temporal graph visualisation is heavily related to graph comparison, and Q2 can draw on a large body of literature in this area. Layout, transitioning, differencing, and matching techniques can all be used to support graph comparison. Most temporal graph systems focus on comparison between adjacent timeslices; a few systems support comparison of non-adjacent timeslices through use of transitioning techniques [BPF14b], filtering of a small multiple display to allow juxtaposed comparisons [FHQ11], or selection of timeslices for use in comparison views [ITK10] [ZKS11]. [ZKS11] offers juxtaposed and superimposed views, and 'relative re-layout' of graphs to facilitate comparison. [ITK10] offer juxtaposed, superimposed, and animated views, and consider methods for computing layouts in such cases. Positions of nodes in the timeslices are synchronised, and co-ordinate panning and zooming in the graph, and highlighting of nodes is employed. DGDtool [PRB08] allows users to select and compare multiple timeslices and apply different layout algorithms.

Comparison in Q2 also relates to comparison of graph objects in the same timeslice: additional support for this may be required for large graphs where the components being compared are distantly positioned in a crowded display. DualNet [SNGS07], a static graph technique, allows selection and comparison of two different parts of the same network, in linked side-by-side views.

Comparison of graph attributes is generally not well considered in the literature. [ABR*13] consider techniques for weighted graph comparison, and some support is offered in temporal graph systems e.g. [EHKW04] [TS07]. Little attention has been given to comparison of *different* attributes e.g. comparing the distribution of attribute A with attribute B. This could be an interesting challenge, where the graphs in question occur at different timepoints or are selected from different parts of the graph.

Comparison in Q3 is little considered by the temporal graph literature. Nested views show individual temporal trends in the same display space, which allows comparison to some extent. However the conditions are not optimal due to the limited space available to show each timeseries, and their spatial positions are determined by the graph layout. Comparison is better supported where timeseries are aligned, as in [RCM*14] and [GGK*11]. With regard to comparison of different attributes, TimeMatrix [YEL10] supports comparison of the temporal behaviour of two different types of edges between the same pair of nodes, and comparison of different attributes over time for an individual node or edge. Techniques for more flexible selection of timeseries associated with different graph objects, time periods, and attributes, for use in comparison tasks could be considered when designing temporal graph systems.

Comparison of data items in Q4 (evolving graphs or temporal distributions over graph structures) is not well documented in the literature. [SLN05] and [YEL10] allow multiple attributes to be displayed in their timeseries glyphs, potentially supporting comparison of temporal distributions of different attributes over the graph. MatrixFlow [PS12] offer a juxtaposed view of the evolution of three co-occurrence matrices aligned over the same time period. [ITK10] support comparison of evolution of two different graphs: at each time point, a timeslice from each graph is combined in one of three ways (aggregate, pile, or split view, reflecting Gleicher's [GAW*11] approaches). These combined timeslices can then be visualised using the temporal layouts offered by their system (animation, juxtaposition, 2.5D, merged and superimposed views). An interesting direction for future research would be to adapt these techniques to explore the possibilities relating to comparison of different parts of the graph, different time periods, and different attributes, and also assess the effectiveness of combinations of comparison techniques and the temporal encodings (e.g. is comparing sequential (animated) views side by side an effective way to compare structural change over time?).

In inverse comparison, in order to compare the times or graph objects associated with a value or pattern of interest, these must be identifiable to the user (as discussed in the lookup tasks). Assessing the connectivity of two graph objects can be supported by highlighting the edge/path between two selected nodes. PaperLens [LCRB05] (for static graphs) allows the selection of two nodes from a drop down list, and displays the "degree of separation links" between them.

One final variation of comparison tasks is where the comparison involves a specified value or pattern (i.e. one not necessarily found in the data), e.g. a particular graph motif or pattern of graph evolution; in this case a system may need some way to visually represent this for use during analysis.

3.3. Relation Seeking

Relation seeking is the opposite of comparison, in that we want to find items—graph objects, times, attribute values, patterns—related in a given way. Many of the comparison techniques also support relation seeking. Matching techniques—which find common elements between two graph representations—can be considered relation seeking techniques; [HD12] note the use of three different approaches: visual links, colour coding, and brushing and linking. In the graph structural case, relation seeking may involve connections between graph objects e.g. 'find the nodes connected to node A'; highlighting nodes linked to a selected node is a common technique to support this task.

TimeSearcher [HS04] is a good example of a technique supporting relation seeking in Q3: specifying a slope and tolerance results in all timeseries with a similar slope being selected. An opportunity for future research could be the development of similar visual analytics tools to find structural patterns in Q2 and Q4 e.g. highlighting potentially similar clusters or motifs, or similar patterns of graph evolution.

4. Conclusions

We have presented our work in mapping visual techniques to the exploratory tasks for temporal graphs which they support; a summary is included in the supplementary material. We anticipate the usefulness of this mapping during the design process, particularly when selecting techniques for inclusion in systems.

The four categories of data items which may participate in tasks distinguished in [KKCss] demonstrate the need to incorporate techniques from a wider range of research areas than those specifically associated with temporal graph representation, in order to effectively support the tasks which may be of interest when exploring temporal graph data.

We identified a number of areas for further research, including techniques to support the comparison of data items in Q4, and plan to investigate this area in our future work. We also plan to make use of our mapping in the design of a new temporal graph visualisation application.

References

[AA06] Andrienko N., Andrienko G.: *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer, New York, 2006. 1

[ABR*13] Alper B., Bach B., Riche N. H., Isenberg T., Fekete J.-D.: Weighted graph comparison techniques for brain connectivity analysis. In *Proc. CHI'13* (2013), ACM, pp. 483–492. 4

[ARH12] Aigner W., Rind A., Hoffmann S.: Comparative Evaluation of an Interactive Time-Series Visualization that Combines Quantitative Data with Qualitative Abstractions. *Computer Graphics Forum 31*, 3 (2012). 3

[ATMS*11] Ahn J.-w., Taieb-Maimon M., Sopan A., Plaisant C., Shneiderman B.: Temporal visualization of social network dynamics: Prototypes for nation of neighbors. *Social computing, behavioral-cultural modeling and prediction* (2011), 309–316. 2

[BBDW14] Beck F., Burch M., Diehl S., Weiskopf D.: The state of the art in visualizing dynamic graphs. In *Proc. EuroVis'14 - State of The Art Reports.* (2014). 1

[BPF14a] Bach B., Pietriga E., Fekete J.: Visualizing Dynamic Networks with Matrix Cubes. In *CHI 2014* (2014). 3

[BPF14b] Bach B., Pietriga E., Fekete J.-D.: GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks. *IEEE TVCG 20*, 5 (Nov. 2014), 740–754. 2, 3

[CGS*09] Chang D., Ge Y., Song S., Coleman N., Christensen J., Heer J.: Visualizing the Republic of Letters, 2009. 2, 3

[DS13] Dunne C., Shneiderman B.: Motif Simplification : Improving Network Visualization Readability with Fan, Connector, and Clique Glyphs. In *Proc. CHI '13* (2013), pp. 3247–3256. 3

[EHKW04] Erten C., Harding P., Kobourov S., Wampler K.: GraphAEL : Graph Animations with Evolving Layouts. *Graph Drawing: Lecture Notes in Computer Science 2912/2004* (2004), 98–110. 4

[FAM*12] Federico P., Aigner W., Miksch S., Smuc M., Windhager F.: Vertigo zoom : combining relational and temporal perspectives on dynamic networks. In *Proc. AVI'12* (Capri Island (Naples), Italy, 2012). 3

[FHQ11] Farrugia M., Hurley N., Quigley A.: Exploring temporal ego networks using small multiples and tree-ring layouts. In *Proc. ACHI* (2011), pp. 79–88. 3

[GAW*11] Gleicher M., Albers D., Walker R., Jusufi I., Hansen C. D., Roberts J. C.: Visual comparison for information visualization. *Information Visualization 10*, 4 (Sept. 2011), 289–309. 3, 4

[GGK*11] Gove R., Gramsky N., Kirby R., Sefer E., Sopan A., Dunne C., Shneiderman B., Taieb-Maimon M.: NetVisia : Heat Map & Matrix Visualization of Dynamic Social Network Statistics & Content. In *Proc. IEEE Conference on Social Computing* (2011), IEEE Press, pp. 19–26. 3, 4

[GK08] Graham M., Kennedy J.: Multiform Views of Multiple Trees. *12th International Conference Information Visualisation* (July 2008), 252–257. 2

[HD12] Hascoët M., Dragicevic P.: Interactive graph matching and visual comparison of graphs and clustered graphs. In *Proc. AVI '12* (2012), pp. 522–529. 4

[HS04] Hochheiser H., Shneiderman B.: Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization 3*, 1 (2004), 1–18. 4

[ITK10] Itoh M., Toyoda M., Kitsuregawa M.: An Interactive Visualization Framework for Time-Series of Web Graphs in a 3D Environment. *Proc. IV 2012*, vi (July 2010), 54–60. 3, 4

[KKC14] Kerracher N., Kennedy J., Chalmers K.: The Design Space of Temporal Graph Visualisation. In *Proc. EuroVis'14, Short Papers Track* (Swansea, 2014). 1, 3

[KKCss] Kerracher N., Kennedy J., Chalmers K.: A Task Taxonomy for Temporal Graph Visualisation. *IEEE TVCG* (In press). 1, 4

[LCRB05] Lee B., Czerwinski M., Robertson G., Bederson B. B.: Understanding research trends in conferences using paperLens. *Extended abstracts on Human factors in computing systems - CHI '05* (2005), 1969–1972. 4

[LS08] Leydesdorff L., Schank T.: Dynamic animations of journal maps: Indicators of structural changes and interdisciplinary developments. *Journal of the American Society for Information Science and Technology 59*, 11 (Sept. 2008), 1810–1818. 3

[PRB08] Pohl M., Reitz F., Birke P.: As Time Goes by âĂŞ Integrated Visualization and Analysis of Dynamic Networks. *Proc. AVI '08* (2008), 372–375. 3

[PS12] Perer A., Sun J.: MatrixFlow : Temporal Network Visual Analytics to Track Symptom Evolution during Disease Progression. In *AMIA Annual Symposium Proceedings.* (2012), Association A. M. I., (Ed.), p. 716. 4

[RCM*14] Riche N. H., Carpendale S., Madhyastha T., Roussel N., Grabowski T. J.: LinkWave: a visual adjacency list for dynamic weighted networks. In *Proc. IHM '14* (New York, NY, USA, 2014), ACM, pp. 113–122. 3, 4

[SD13] Shneiderman B., Dunne C.: Interactive network exploration to derive insights: filtering, clustering, grouping, and simplification. In *GRAPH DRAWING Lecture Notes in Computer Science*. Springer, 2013, pp. 2–18. 3

[Shn96] Shneiderman B.: The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. IEEE Symposium on Visual Languages* (1996), pp. 336–343. 2

[SLN05] Saraiya P., Lee P., North C.: Visualization of Graphs with Associated Timeseries Data. In *IEEE Symposium on Information Visualization* (Minneapolis, MN, USA, 2005), IEEE, pp. 225–232. 3, 4

[SNGS07] Staats B., Namata Jr G. M., Getoor L., Shneiderman B.: A Dual-View Approach to Interactive Network Visualization. In *Proc. CIKM '07* (2007), pp. 939–942. 4

[TS07] Tu Y., Shen H.-W.: Visualizing changes of hierarchical data using treemaps. *IEEE TVCG 13*, 6 (2007), 1286–93. 4

[vHP09] van Ham F., Perer A.: ÂŞ Search, Show Context, Expand on Demand: Supporting Large Graph Exploration with Degree-of-Interest. *IEEE TVCG 15*, 6 (2009), 953–960. 2

[WL90] Wehrend S., Lewis C.: A problem-oriented classification of visualization techniques. *Proc. First IEEE Conference on Visualization: Visualization '90* (1990), 139–143,. 1

[YEL10] Yi J. S., Elmqvist N., Lee S.: TimeMatrix: Analyzing Temporal Social Networks Using Interactive Matrix-Based Visualizations. *International Journal of Human-Computer Interaction 26*, 11-12 (Nov. 2010), 1031–1051. 3, 4

[ZKS11] Zaman L., Kalra A., Stuerzlinger W.: The Effect of Animation , Dual View , Difference Layers , and Relative Re-Layout in Hierarchical Diagram Differencing. In *Proceedings of Graphics Interface* (2011), pp. 183–190. 3