

Exploring Spatio-Temporal Data Modeled as Dynamic Weighted Relations

Michael Burch, Michael Raschke, Daniel Weiskopf

VISUS, University of Stuttgart
Allmandring 19, 70569 Stuttgart, Germany

Abstract. Eye tracking studies lead to spatio-temporal data in the form of gaze trajectories that show the behavior of gaze positions over time. Such data can be modeled as a dynamic graph that expresses the transitions of gaze positions between Areas of Interest (AOIs) by time-varying weighted relations. Moreover, a hierarchical organization of the AOIs may be of interest, resulting in a dynamic compound AOI digraph. Traditionally, this kind of time-based relational data is represented by animated node-link diagrams that are laid out with respect to a list of aesthetic graph drawing criteria. In our work, we propose a visual metaphor for displaying relational data that uses space-filling circle sectors to encode dynamic relations between hierarchy elements. The idea benefits from the fact that dynamic compound digraphs can be visualized with reduced visual clutter compared to node-link diagrams for dense graphs. Finally, we illustrate how interaction methods can be used to explore a dataset for trends, countertrends, and/or anomalies.

1 Introduction

Information hierarchies are present in many application domains such as in the hierarchical organization of software and river systems. Also, Areas of Interest (AOIs) are useful when exploring spatio-temporal eye tracking gaze trajectories, which can be hierarchically organized. Moreover, the eye movement behavior of study participants has a spatio-temporal nature. By using the AOI information and the gaze trajectories this data can be modeled as a dynamic weighted compound AOI digraph. Consequently, this kind of relational data can be visually explored with the general concept proposed in this paper.

The efficient representation of hierarchical data has been in focus of information visualization research ever since. Hierarchies are, for example, displayed by traditional node-link diagrams [13], Treemaps [8], indented plots [5], or layered icicles [9].

General graphs—if they do not belong to the class of planar graphs—suffer from visual clutter [14] when depicted graphically by a naive layout technique. Consequently, over the years many sophisticated graph drawing algorithms were developed to make graphs aesthetically pleasing [10–12, 16]. Apart from reducing edge crossings—which is the major criterion in graph drawing—reducing edge lengths, maximizing orthogonality and symmetries, or minimizing display space

are other aesthetic criteria among a larger set. Typically, graph data does not stay static but changes over time, leading to much research in this domain [2, 3, 6, 7].

In this paper, we propose an interactive visualization tool for representing and manipulating this kind of dynamic graph data. Our approach uses a radial visual metaphor and is based on the visualization principles proposed by Burch and Diehl in their TimeRadarTrees technique [3]. In particular, eye tracking data is of spatio-temporal nature and is visualized by heatmaps or gaze plots as traditional concepts. Heatmaps are time-aggregated representations only showing the hot spots, whereas gaze plots suffer from visual clutter. Also, AOI rivers [4] produce many crossings in the display but better show the temporal evolution of eye gaze frequencies between AOIs in displayed stimuli.

The work of Blaschek and Ertl [1] describes an approach that is useful for supporting researchers working in the field of information visualization when deriving insights from eye tracking experiments. The proposed framework uses visual analysis methods to evaluate eye tracking data. Our proposed method transforming eye movement data to dynamic graphs might be used as one candidate of such an analysis method among others.

We illustrate the visualization tool in a stepwise manner and follow the visual information-seeking mantra [15] that summarizes many visual design guidelines and provides an excellent framework for designing information visualization applications. The visual information-seeking mantra divides the visual exploration process into the following three stages: Overview first, zoom and filter, then details-on-demand.

2 Dynamic Graph Visualization

The main focus of our visualization tool is the visualization of dynamic relational data in information hierarchies as static pictures that can be manipulated and analyzed interactively. This representation stands in contrast to animation-based techniques for displaying dynamic data. The other visualization views presented here support users when they want to obtain an overview of the dataset or zoom in and apply filtering functions to the dataset.

2.1 Data Model

We model an information hierarchy as a cycle-free and connected graph $H = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ expresses the set of directed edges—the link information in the hierarchy. $L \subseteq V$ is the set of leaf nodes in the containment hierarchy. All other nodes $V \setminus L$ are containers that hierarchically bundle this information. The leaf nodes can be related to some extent and actually form a directed graph with edge weights $G = (L, E_G)$, where $E_G \subseteq L \times L$ and $w_G(u, v) \in \mathbb{R}_0^+$ denotes the weight function for $u, v \in L$.

A sequence of graphs can be modeled by $G_i = (L_i, E_{G_i})$ where $E_{G_i} \subseteq L_i \times L_i$. $w_{G_i}(u, v) \in \mathbb{R}_0^+$ is the weight function for $u, v \in G_i$. If the context is clear, we omit the graph G and use $w_i(u, v)$ instead of $w_{G_i}(u, v)$.

A set of gaze trajectories as generated during eye tracking experiments can be transformed into dynamic weighted directed graphs by subdividing the time into subintervals each corresponding to one graph in the sequence. The AOI information can be modeled as graph vertices and the number of eye movements between pairs of AOIs as directed weighted edges. If a hierarchical organization among the AOIs exists or one can be computed by a hierarchical clustering algorithm, then this spatio-temporal eye tracking data can be transformed into a dynamic weighted compound AOI digraph.

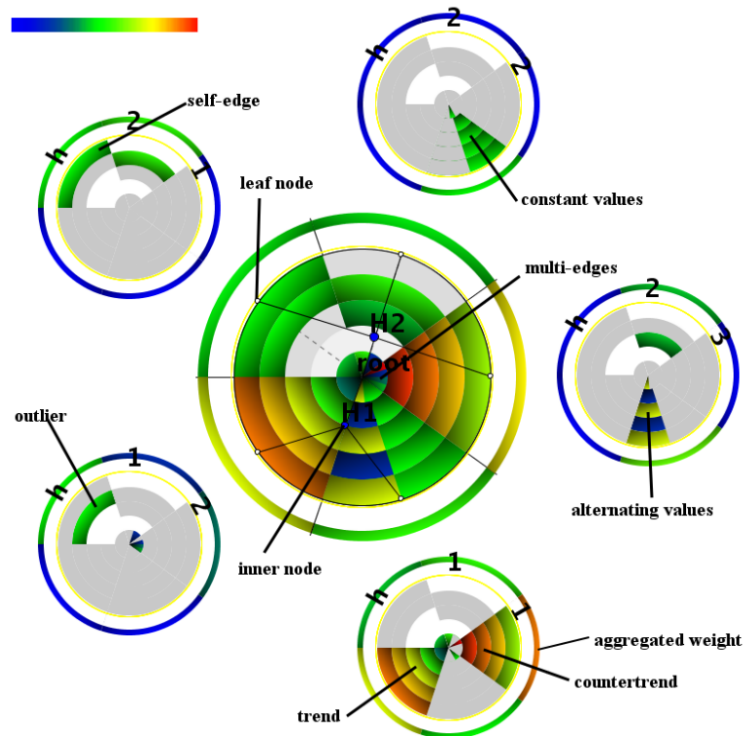


Fig. 1. A sequence of five compound directed graphs in a TimeRadarTrees representation with color-coded edge weights.

2.2 Visual Encoding

For displaying the dynamics of relations in information hierarchies, we use a TimeRadarTrees [3] visualization. Figure 1 shows a sequence of five directed and weighted graphs in this visual metaphor. The visualization is an integration of three views into one.

- **Hierarchy View:** The hierarchy of the selected elements is represented as a radial node-link diagram with the root node in the circle center and the node sizes and colors depending on the size and depth of their subtrees.
- **Time Radar View:** The large circle in the center is used to represent the sequence of graphs. Each circle slice encodes a single graph and each circle sector one specific node in this sequence. A weighted edge is represented by a color-coded circle sector. The circle is divided into as many sectors as different nodes are present in the graph sequence. The time axis starts in the circle center and heads radially to the circumference.
- **Thumbnail View:** Thumbnails—the miniature representations outside the larger circle in the middle—can be used to derive start and destination nodes of a weighted edge. By inspecting the color, the shape, the curvature, the direction, and/or orientation of a small circle sector and comparing it to the larger one in the center circle, one can derive a relation between two nodes.

For the example in Figure 1, one can detect that the hierarchy consists of five leaf nodes, namely h_{11} , h_{12} , h_{21} , h_{22} , and h_{23} . These are organized into two subhierarchies, H_1 and H_2 , which are again direct children of the root node. Five graphs are represented, which can be inferred from the fact that each circle is divided into five color-coded slices. The color coding in use is a vegetation color scale that maps lower values to blue and higher ones to red. Values in between are colored from green to yellow. The outer smaller slice encodes the aggregation of edge weights over the whole graph sequence and can be explored for each node pair separately.

The upper left green-colored sector in the thumbnail of node h_{21} represents a self-edge—an edge that starts and ends at the same node. The permanently green-colored sectors in the thumbnail of node h_{22} show that this relation to node h_{11} always exists in the whole sequence and moreover, has a constant weight. The thumbnail of node h_{23} on the right hand side has an alternating relation to node h_{11} . This phenomenon can be examined by the alternating color coding between blue and yellow. A trend and a countertrend are visible in the thumbnail of node h_{11} , where the strength of the relation to node h_{23} decreases at a constant level. The same is true for the relation to node h_{12} , but here we can detect an increase of the weight. An outlier may be the green sector in the thumbnail of node h_{12} that points to node h_{21} and hence into a different hierarchy level. It goes without saying that this kind of graph representation also allows multi-edges—two or more edges between the same two nodes in a graph. There are actually three edges in the first graph starting at node h_{12} and targeting to node h_{23} .

3 Working with the Tool

Figure 2 shows the graphical user interface of the visualization tool, which is divided into several components. The upper left frame shows the overview by a scrollable pixel map. Below it, the selected part of the pixel map is shown

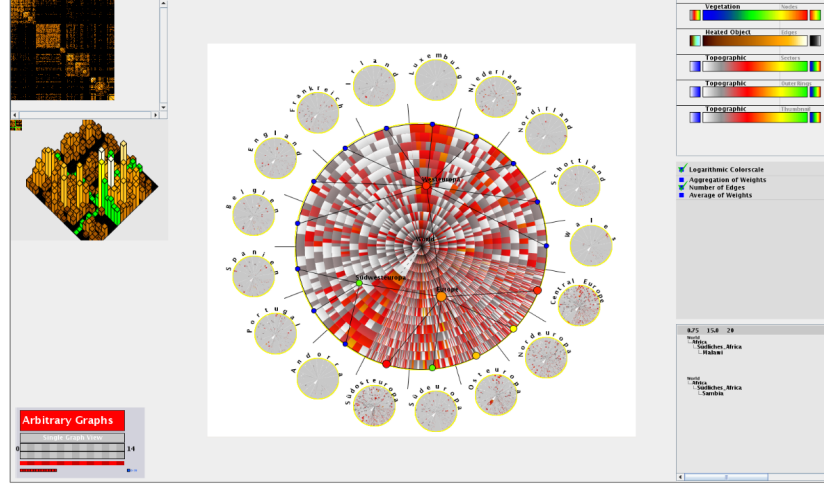


Fig. 2. The graphical user interface of the visualization tool is divided into several views. The TimeRadarTrees view in the center shows the evolution of relations in information hierarchies of the selected hierarchies and time period.

as a three-dimensional bar chart as some kind of zooming function. The filter functions are located in the bottom view on the left hand side. The three views at the right hand side are used for applying a special color coding, selecting an aggregation type or logarithmic scale, and a details-on-demand view. The main view in the center represents the TimeRadarTrees visualization for the selected nodes and time periods. In the following, we explain the single visualization components and interaction techniques to manipulate the data.

3.1 Overview: Pixel Map

As we follow the information seeking-mantra, we first give an overview of the whole relational dataset. To this end, we aggregate the graph sequence to one aggregated graph that we represent as a pixel map—a color-coded adjacency matrix. The color coding depends on the aggregation type that can be:

- **Weighted sum:** All weights in the graph sequence are summed up:

$$w_{agg}(u, v) := \sum_{1 \leq i \leq n} w_i(u, v) \quad \forall u, v \in E_i$$

- **Average weight:** The weights are summed up and divided by the number of occurrences of this edge in the sequence:

$$w_{agg}(u, v) := \frac{\sum_{1 \leq i \leq n} w_i(u, v)}{|\{(u, v) \mid (u, v) \in E_i\}|} \quad \forall u, v \in E_i$$

If the number of edges is zero, the weight is defined by

$$w_{agg}(u, v) := 0.$$

- **Number of edges:** The number of edges depends on the weight filter that is again defined by the minimum weight v_{min} and the maximum weight v_{max} . Hence, we define in this case

$$w_{agg}(u, v) := |\{(u, v) \mid v_{min} \leq w_i(u, v) \leq v_{max}\}|$$

The users can interactively change the type of aggregation until they obtain interesting insights in the dataset. At this early stage, they can detect patterns and anomalies by exploring the hierarchically ordered pixel map. If they find interesting behaviors in the dataset, they may wish to filter the data for smaller subsets and obtain a three-dimensional bar chart of the brushed elements. Further examination of the selected part leads to more specific insights in the dataset that was not possible in the overview visualization.

3.2 Zoom: 3D Bar Charts

Figure 3 shows three-dimensional bar charts for the selected area in the pixel map. One big advantage of this representation is that we can encode two different metrics at the same time—one in the height and one in the color coding. This could help find out which relations occur very often and also have a very high aggregated weight. We cannot obtain this insight in the overview-based pixel map view apart from the fact that we may switch the aggregation type.

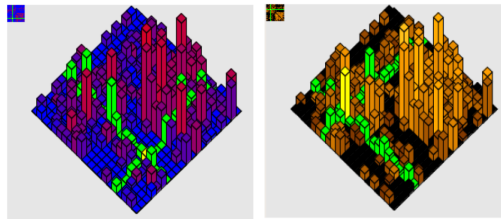


Fig. 3. Three-dimensional views of a selected subset can encode two metrics at the same time—one in the height of the bars and one in the color coding.

A crosshair function supports selecting a bar in the three-dimensional view. Selecting this bar can be difficult due to occlusion problems. The selected aggregated relation is given as a details-on-demand information with its metric values nearby.

3.3 Data Filtering

The tool is able to handle three different types of filtering functions that can be applied separately.

- **Weight filter:** First of all, one can apply a weight filter on the single values of each relation. Only those values are counted as relations that lie between the given thresholds. The aggregated values can be filtered for minimum and maximum sum, minimum and maximum average value, and minimum and maximum edge number.
- **Hierarchy filter:** The weight filter can be applied to the relations, another type of filter can be used to select a group of nodes from the hierarchy. Only those nodes are represented that belong to the selected group. This filtering technique is very important for the TimeRadarTrees visualization because it has a low scalability in this dimension.
- **Time filter:** Time is the third dimension where we can apply a filtering function. The thresholds can be given as an interval where all graphs are displayed that lie within this interval or single graphs can be selected and deselected again.

3.4 TimeRadarTrees

The main part of the visualization tool is the TimeRadarTrees representation in the center. It is used to display the dynamic graph data in information hierarchies in a static picture. Figure 1 illustrates the important visual signatures that are visible in the TimeRadarTrees representation for the example of a small dynamic graph. User interaction can be applied to modify the TimeRadarTrees view and to gain insights from the data on a dynamic level.

Interaction methods supported by the tool include filtering, adapting the color coding, aggregation of relations, collapsing and expanding subhierarchies, time warp, sector highlighting, textual search, changing of the hierarchy visual metaphor, cushion effect on the circle sectors, stacking of radial bar charts, etc.

4 Conclusion and Future Work

In this paper, we proposed a visualization tool for exploring dynamic compound digraphs in information hierarchies. Such datasets can be generated by taking spatio-temporal eye movement data and corresponding areas of interests into account. By doing this, a dynamic weighted digraph is produced that can then be visually explored by our dynamic graph visualization tool. In general, the difficulty is to first present the data in a pixel-based overview and allow the user to zoom and filter the dataset in several dimensions, i.e., weight, hierarchy, and time.

The TimeRadarTrees visualization technique is used to give insights in the dynamic data. The strength of this technique is the static representation of dynamic data that shows several dimensions in a single view—relation weights, hierarchical organization, and time periods. Interaction techniques support the user in obtaining insights in the dataset.

We plan to implement more interaction techniques such as a radial and a sector distortion. To this end, the TimeRadarTrees visualization will be mapped

to a circular shape. A sector distortion would allow unequal sized sectors, but the whole structure would still be mapped on a circle. As an enhancement, we could allow a distortion of the circular shape by dragging and dropping the circumference, thus obtaining an irregular shape that may improve the visual exploration when also applied to the thumbnail view in the same way.

The weakness of our approach—in contrast to node-link diagrams—is the lack of support for solving path-related tasks. To address this problem, we plan to implement an interaction method that highlights paths in the graph data displayed as TimeRadarTrees.

References

1. Blascheck, T., Ertl, T.: Techniques for analyzing empirical visualization experiments through visual methods. In: *Proceedings of Workshop on Visual and Spatial Cognition (2013, to appear)*
2. Burch, M., Beck, F., Diehl, S.: Timeline trees: Visualizing sequences of transactions in information hierarchies. In: *Proceedings of 9th International Working Conference on Advanced Visual Interfaces*. pp. 75–82 (2008)
3. Burch, M., Diehl, S.: TimeRadarTrees: Visualizing dynamic compound digraphs. *Computer Graphics Forum* 27(3), 823–830 (2008)
4. Burch, M., Kull, A., Weiskopf, D.: AOI rivers for visualizing dynamic eye gaze frequencies. *Computer Graphics Forum* 32(3), 281–290 (2013)
5. Burch, M., Raschke, M., Weiskopf, D.: Indented Pixel Tree Plots. In: *International Symposium on Visual Computing*. pp. 338–349 (2010)
6. Burch, M., Vehlow, C., Beck, F., Diehl, S., Weiskopf, D.: Parallel edge splatting for scalable dynamic graph visualization. *IEEE Transactions on Visualization and Computer Graphics* 17(12), 2344–2353 (2011)
7. Greilich, M., Burch, M., Diehl, S.: Visualizing the evolution of compound digraphs with TimeArcTrees. *Computer Graphics Forum* 28(3), 975–982 (2009)
8. Johnson, B., Shneiderman, B.: Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In: *Proceedings of IEEE Visualization Conference*. pp. 284–291 (1991)
9. Kruskal, J., Landwehr, J.: Icicle Plots: Better displays for hierarchical clustering. *The American Statistician* 37(2), 162–168 (1983)
10. Misue, K., Eades, P., Lai, W., Sugiyama, K.: Layout adjustment and the mental map. *Journal of Visual Languages and Computing* 6(2), 183–210 (1995)
11. Purchase, H.C.: Which aesthetic has the greatest effect on human understanding? In: *Graph Drawing*. pp. 248–261 (1997)
12. Purchase, H.C., Cohen, R.F., James, M.I.: Validating graph drawing aesthetics. In: *Graph Drawing*. pp. 435–446 (1995)
13. Reingold, E.M., Tilford, J.S.: Tidier drawings of trees. *IEEE Transactions on Software Engineering* 7(2), 223–228 (1981)
14. Rosenholtz, R., Li, Y., Nakano, L.: Measuring visual clutter. *Journal of Vision* 7(2), 1–22 (2007)
15. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: *IEEE Symposium on Visual Languages*. pp. 336–343 (1996)
16. Ware, C., Purchase, H.C., Colpoys, L., McGill, M.: Cognitive measurements of graph aesthetics. *Information Visualization* 1(2), 103–110 (2002)