Evaluating Perceptually Complementary Views for Network Exploration Tasks

Chunlei Chang
Monash University
Melbourne, Australia
chunlei.chang@monash.edu

Benjamin Bach
Monash University
Melbourne, Australia
benj.bach@gmail.com

Tim Dwyer
Monash University
Melbourne, Australia
tim.dwyer@monash.edu

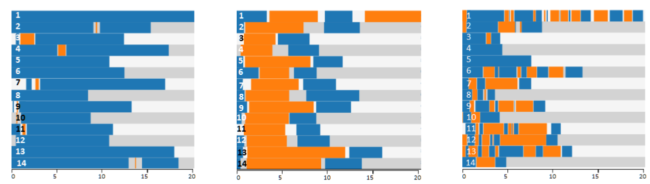
Kim Marriott
Monash University
Melbourne, Australia
kim.marriott@monash.edu

ABSTRACT
We explore the relative merits of matrix, node-link and combined side-by-side views for the visualisation of weighted networks with three controlled studies: (1) finding the most effective visual encoding for weighted edges in matrix representations; (2) comparing matrix, node-link and combined views for static weighted networks; and (3) comparing *MatrixWave*, *Sankey* and combined views of both for event-sequence data. Our studies underline that node-link and matrix views are suited to different analysis tasks. For the combined view, our studies show that there is a perceptually complementary effect in terms of improved accuracy for some tasks, but that there is a cost in terms of longer completion time than the faster of the two techniques alone. Eye-movement data shows that for many tasks participants strongly favour one of the two views, after trying both in the training phase.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous;

Author Keywords
Network visualisation, eye tracking, event sequence data, node-link diagrams, sankey diagrams, matrices

INTRODUCTION
The two most commonly used representations for network data are node-link diagrams and adjacency matrices. Studies have demonstrated distinct advantages to each, suggesting a certain complementariness for both visualisations. Similarly, for representing time-varying flows or event-sequences over networks, two complementary representations have been proposed. Sankey diagrams [19] follow the node-link metaphor, showing flows as curved lines between nodes in discrete time-steps; while MatrixWave [26] presents flows through a stacked sequence of matrices.



(a) Exclusive use (b) Sequential use (c) Parallel use

Figure 1. Examples of user behaviours while using complementary views. Horizontal bars present study trials for the same user, time running from left to right. Colours indicate the view the participant was looking at.

While node-link diagrams—including Sankey diagrams—suffer from visual clutter when displaying highly-connected (dense) networks, matrix layouts manage to avoid visual clutter by dedicating a predefined space for each link to be shown. However, matrix visualisations seem to be less intuitive and less familiar to most users and especially difficult to use for following paths. The pros and cons of each technique with respect to different tasks suggest that providing users with both views *simultaneously* might improve their performance and satisfaction during the analysis process. However, providing multiple coordinated views comes at the cost of increased visual complexity and requires users to understand when to use which visualisation. This supposed trade-off between complementary effects of simultaneous views versus their additional complexity has not previously been formally studied.

In this paper, we provide evidence from two controlled user studies, each one comparing two sets of visualisations: Node-link versus Matrix and Sankey versus MatrixWave. We consider the combined views to be *perceptually* complementary, as opposed to *informationally* complementary. Informationally complementary views show a different aspect of the data, such as different attributes, times, or levels of detail. Neither view is redundant to the analysis process and it is not possible to generate one view from the other as they contain different information. By contrast, perceptually complementary views

CHI 2017, May 06-11, 2017, Denver, CO, USA

DOI: http://dx.doi.org/10.1145/3025453.3026024

show exactly the same information and each is in some sense redundant. Perceptually complementary views may support tasks that require such a redundant but complementary visual encoding, e.g. a clear view of the network topology and a view that more efficiently encodes edge weight.

In investigating perceptually complementary views we seek answers to the following questions:

Q1: Which tasks are better supported for *i)* weighted networks (by Nodelink or Matrix) and for *ii)* event sequences over weighted networks (by MatrixWave or Sankey diagrams)?
Q2: Do side-by-side perceptually complementary views improve or decrease performance (time and error) with respect to the individual views alone?
Q3: How do users use complementary views when provided with two visualisations simultaneously (combined)?

For combined views, our studies show that there is a perceptually complementary effect in terms of improved accuracy for some tasks, but at the cost of longer completion times than the faster of the two techniques alone. Eye-movement data shows that for many tasks participants strongly favour one of the two views, after trying both in the training phase.

Different encodings for edge weight are often seen in practice and in different research on matrix representations of weighted networks but there does not seem to have been any prior formal evaluation of this topic. Since we needed this information for our studies of complementary network visualisation (Studies 2 and 3), we began with an evaluation of edge-weight encodings in matrices (Study 1).

All our datasets, study results, and visualisations of eye-tracking data are available: http://marvl.infotech.monash.edu/~dwyer/papers/chi2017/.

RELATED WORK

Our related work describes visualisations and studies on network and event-flow visualisations, as well as the use of complementary views in information visualisation.

Representations of Network Data

Ghoniem and Fekete [10] found that for networks with high link density, various tasks were more easily completed with an adjacency matrix representation, even though matrices were unordered. For sparse directed graphs Keller *et al.* [14] show that finding edges and paths for given nodes is significantly faster with node-link diagrams and they suggest that these are also more intuitive. For networks with weighted edges there appears to be little evidence of how different visualisations are complementary. Alper *et al.* [1] considered different encodings of pairs of edge weights in a single matrix to compare graphs, and found matrices greatly outperform node-link diagrams. However, the context of their study was different and their goal was not to assess complementariness.

Event Sequence Data

Event sequences describe flows of a certain strength between entities at discrete time steps. Formally, these can be considered layered networks with weighted links. Sankey diagrams [19, 24] have become a popular way to show such data, for example, they are supported by a number of visualisation tool-kits including D3 [5] and are offered by Google Analytics for analysis of web clickstream data. Sankey diagrams are easily understood, showing a very literal "pipeline" of flow between nodes at different points in time. However, like other kinds of node-link representations, Sankey diagrams can quickly become cluttered through too many links overlapping and obstructing each other and their readability has not been studied in detail. The only exception of which we are aware is a usability study of a system featuring Sankey diagrams [24] and in which a comparison to alternative visualisations was required for future work.

An alternative technique for visualising event sequence data was recently proposed by Zhao *et al.* [26]. It uses a linked sequence of adjacency matrices in a zig-zag tiling and is, hence, called *MatrixWave*. As with an earlier design for unweighted, levelled directed graphs by Bezerianos *et al.* [4], the design rationale is that adjacency matrices scale better to dense networks than node-link diagrams. MatrixWave was compared to Sankey diagrams by Zhao *et al.* for analysis of click-stream data. This study found distinct advantages of MatrixWave for tasks where participants needed to read precise values of link-weights. However, the study also suggested that Sankey diagrams remained more precise for showing path information. Qualitative feedback indicated that the Sankey diagrams were more intuitive and learnable by participants.

We are not aware of any work that compares the effect of *two representations combined*, i.e. side-by-side. In this sense, our work is the first to investigate such a setup.

Coordinated and Complementary Views

Displaying multiple views on the same data set is common practice in information visualisation [20]. Multiple views have long been used to explore complex data sets [23] and are now standard in visualisation software such as Tableau. Differentiating between *informationally* and *perceptually* complementary views allows us to reason about if and how two views are benefiting the analysis process. Another way of understanding this difference is that informationally complementary views result from "forks" [22] earlier in the visualisation pipeline [6, pg. 17] such that the filters restricting which data is shown in each view differ. Perceptually complementary views result from forks later in the pipeline, after data values have been determined but before the representation has been chosen.

Perceptually complementary views may be beneficial for a different set of tasks than informationally complementary views. This notion, that representations that show the same information in different ways (i.e. that are *isomorphic*) may lead to different understanding, is supported by results in cognitive psychology where it is known as the *representational effect* [25]. Research in information visualisation has shown that, within a single visualisation, redundant encoding of a single data attribute with different visual channels (e.g. colour as well as size of marks) may improve readability [17]. However, the idea that there may be a similar effect across multiple views (e.g. that isomorphic views may be perceptually complementary) has not been formally tested for data visualisations prior to the studies presented in this paper.

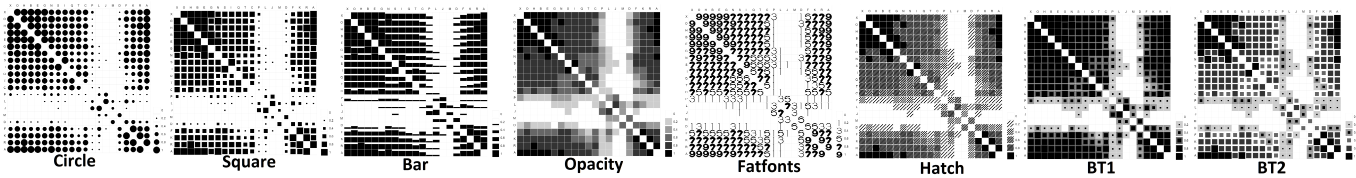


Figure 2. Samples from our encodings used in Study1. (BT = Background Transparency)

It is our view that specification and testing of perceptually complementary visualisation designs is necessary because it is something that has been proposed and tried informally by visualisation designers. For networks, Henry and Fekete proposed MatrixExplorer [11] to explore the potential of combining (coordinated) matrix and node-link visualisation side-by-side in social networks. Potentially, the dual view allows an expert analyst to use the most appropriate representation for the subtask at hand and to move smoothly between the views without loosing their mental model. They might also help novice analysts more quickly learn unfamiliar representations such as the adjacency matrix from familiar representations like a node-link diagram. Of course, this comes at the cost of more screen space, higher coding effort, and the need for the user to aggregate information across both views.

In order to facilitate their parallel usage and aggregate and transfer information between two complementary views, these views are usually *coordinated* or *linked* [16]. Common techniques for linking are brushing-and-linking or visual lines relating data elements (e.g., [7]). In our case, we opted for brushing-and-linking as it is the more flexible and widely used technique to work with coordinated views.

STUDY 1: EDGE-WEIGHTED ENCODING IN MATRICES

Our first study searches for an effective and efficient way to encode edge weights in adjacency matrices. We wanted to find an encoding that would support tasks on two levels: *overview tasks* that include spotting dense regions and finding anomalies, and *detailed tasks* that involve decoding and comparing weights for individual edges. For our study, we selected visual encodings that try to support tasks on both levels.

Encoding Techniques

Links in adjacency matrices are represented by marks in the matrix cells. For overview tasks, to detect patterns quickly within the matrix, an encoding should be pre-attentive. For detailed tasks the encoding should allow the reader to make precise assertions about the encoded value. We assembled eight techniques for our study including both encodings commonly used (e.g., [3, 8, 1, 18]) and a couple that we have not seen used in this context before (*Ff*, *BT2*, see below). The selection of these eight encodings followed discussions and pilot tests and is shown detailed in Figure 2:

- **Circle (Ci):** Edge weight is mapped linearly to the radius of a circle sitting inside matrix cells. In both this encoding and the next, a length-channel encoding is preferred over area following information visualisation standard practice informed by Stevens [21].
- **Square (Sq):** Edge weight is mapped to the size (side length) of a square inside each matrix cell, increasing linearly with edge weight.
- **Bar (Ba):** Similar to *Square*, data values are mapped to the size of the bar. Bars are aligned to the bottom line of the cell and bar height encodes edge weight [3, 18]. Cells with an edge weight of 0 are empty, cells of weight 1 are complete.
- **Opacity (Op)**: Edge weight is mapped to cell opacity between values 0 and 1.
- **Fatfonts (Ff):** Fatfonts encode data redundantly; as numerical value, and by the boldness of the font face that is used to render the numerical value [15]. Larger data values appear as thicker numbers.
- **Hatch (Ha):** Hatching fills the matrix cell with a diagonal line pattern [3, 18]. Similar to fatfonts, hatching can be use for pre-attentive encoding (cell-darkness) and to encode precise values as the number of diagonal lines.
- **Background Transparency 1 (BT1):** This technique is a combination of *Square* and *Opacity* encoding; in addition to square size, edge weight is encoded redundantly in the darkness of the cell-background. It has been used by Alper et al. [1] in a slightly different context. Lower edge weights are represented by a small square in the cell center and a light cell background. Larger values are represented by large black squares and dark cell backgrounds. Edge values of 1 result in entirely dark cells.
- **Background Transparency 2 (BT2):** This encoding is a variation of *Background Transparency 1*. Edge weights larger than 0.5 show a white background. The rational for this encoding is that it may be hard to perceive the size of black squares on a dark background.

Tasks

Detailed task: *edge-weight comparison*. *Given four highlighted cells (two with a blue frame and two with a red frame), which pair has the larger sum of cell weights?* Participants completed this task in four steps: (1) they needed to visually identify all highlighted blue and red cells; (2), they needed to assess the weight of each cell in each colour; and (3) they needed to estimate the aggregated value.

Overview task: *cluster-weight comparison*. *Given two highlighted areas (one marked as blue and one marked as red), which area has the larger sum of cell weights?* Participants completed this task in one step: they needed to estimate which coloured region contained higher aggregated edge weights (in most of our encodings this translated into assessing the overall darkness of a region.)

Both tasks offered the same answer possibilities. Once a participant felt comfortable with the answer, he/she pressed the space-bar; the matrix would vanish, the timing stopped, and participants were presented with a list of possible answers: *blue, red, too difficult, I don't know*. Participants were instructed to vote "too difficult" if they felt their confidence was too low because they could not make proper comparisons. If participants were able to read the encoding but could not make an informed decision, they were told to vote "I don't know", in order to avoid guessing.

Data

We generated data in order to control data characteristics such as edge weights and their distribution. Edge weight varied between 0 and 1, in increments of .2, resulting in 6 levels. We generated networks with a fixed size of 30 nodes and two clusters, as indicated in Figure 2. One cluster was large and homogeneous, the smaller cluster was less homogeneous. For each network, edge weight was altered by a random function.

Participants and Setting

We recruited 19 participants (7 female, 12 male), all with normal or corrected-to-normal vision and without any colour vision impairment. Participants included university students and research staff. 11 participants were familiar with information visualisation, 8 participants where familiar with node-link diagrams. No participant was familiar with matrix visualisations. The study was run on an Intel Core i7 Surface Pro 3 and a 27 inch flat screen (resolution: 3840 $\times$ 2160). The visualisation area was centred in a full size window.

Design

We used a within-subject, full-factorial design: 8 techniques $\times$ 2 tasks $\times$ 18 trials. 18 trials included 3 training trials in the beginning of each condition (technique $\times$ task). The remaining 15 trials contained 3 levels of difficulty (5 $\times$ easy, 5 $\times$ Medium, 5 $\times$ difficult). Difficulty was introduced by edge density, the size of the selection area, and the value difference of selected areas. Each of the 18 trials showed one of the 18 networks, in fixed order and with increasing difficulty. Time per trial was limited to 20 seconds: thus, participants had insufficient time to calculate the sum of edge weights precisely by looking at the legend provided. Rather, they had to compare based on visual impression. For each trial, we measured error and task completion time. After the experiment, participants were asked to rank encodings according to their preference.

Order of techniques was randomised. Before each condition (technique $\times$ task), we instructed participants on the respective task and technique. Then, participants concluded the 3 training trials under supervision and were invited to ask questions. While training was not timed, to limit study duration and participant fatigue, each trial was limited to 20 seconds. After 20 seconds, the matrix was hidden and the participant had to enter his/her response. Excluding training, we resulted in a total of 240 trials per participant.

We had two null-hypotheses for this study:

- H_0: There is no difference in mean error between techniques.
- H_1: There is no difference in mean task-completion time between techniques.

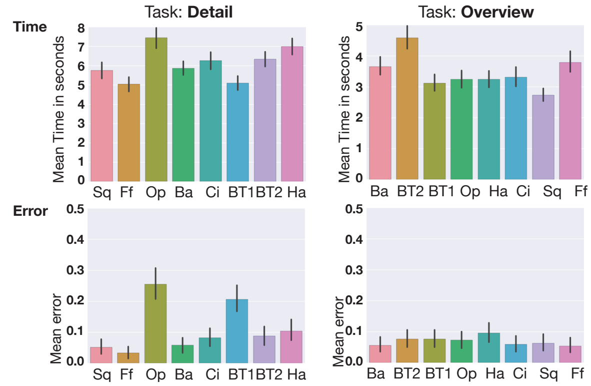


Figure 3. Results for Study 1 (Matrix encoding): task-completion *time* (top) and *error* (bottom). Error bars indicate 95% CI.

Results and Discussion

For the analysis, we removed all samples where participants took less then 1 second, as we considered these samples invalid. This left us with 8908 valid samples (out of 9120 recorded samples, training excluded). Removing samples lead to unequal sample sizes; also, we found that time and error data did not follow a normal distribution and could not be corrected. Therefore, we used the non-parametric *Whitney-Mann-U test* for unequal sample sizes.

Error: Results for error are shown in Figure 3-top. We found significant difference ($p<0.05$) between almost all pairs of techniques, leading us to reject H_0. For the *detailed-task* we found *fatfonts* and *square* to be most accurate (M=0.04 and M=0.06), and *opacity* and *BT1* to be least accurate (M=0.31 and M=0.25). For the *overview-task* we found significant difference only for *hatching*, being the least accurate technique (M=0.11). The most accurate technique on average for this task was *fatfonts* (M=0.06).

Completion Time: Results for completion time are plotted in Figure 3-bottom. We found significances ($p<0.05$) between almost all techniques, rejecting H_1. For the *detailed-task* we found *FatFonts* and *BT1* to be fastest (M=5.0s, SD=3.3 and M=5.0s, SD = 3.5) and *opacity* to be slowest (M=7.7s, SD=4.3). For the *overview-task*, we found *square* and *circle* to be fastest (2.6sec, SD=1.7 and 3.1sec, SD=2.5 no significant difference) and *BT1* and *fatfonts* to be slowest (4.4sec, SD=3.4 and 3.7sec, SD=2.9). The fastest technique (*square*) was 1.57 times faster than the slowest technique (*BT1*).

User preference: Users rated their preference on an 8-point scale ranging from 0 (strongly dislike) to 7 (strongly like). We analysed these ratings using a z-test. We analysed these ratings using a pairwise one-tailed Wilcoxon signed-rank test (with a Bonferroni correction for multiple comparisons) to determine which methods are significantly preferred over the others in the detailed and the overview tasks. We found that in the detailed task, *fatfonts* and *square* are the two most preferred methods. The preference rank changed in the overview task with *square* remaining the top.

Our goal with this study was to find an encoding that performs well across our two tasks (*detailed* and *overview*) with respect to time and error. Results suggest that there is no one encoding scoring best across all tasks but that *square* performed well in all independent measures: *time*, *error*, and *user preference*. Therefore, we decided to use *square* encoding for our two following studies.

STUDY 2: MATRIX AND NODE-LINK DIAGRAM

In a first study on perceptually complementary views, we were interested in the effectiveness of Node-link, Matrix, and a combined side-by-side representation of both Node-link and Matrix, for weighted network exploration.

Techniques

- ***Node-link* (NL):** Nodes are encoded as gray circles and edges as straight lines, laid out using WebCoLa [9]. Edge weight is encoded by line-thickness (Fig. 4 (left)).
- ***Matrix* (M):** According to the results from experiment 1, we use the square encoding to encode edge weight (Fig. 4 (right)). We used optimal leaf ordering to automatically order rows and columns in the matrices.
- ***Combined* (C):** The *Node-link* and *Matrix* views were presented side-by-side, coordinated through brushing-and-linking: i.e. hovering on a circle/row/column/cell/node/link highlights its counterpart element in the complementary view. (Fig. 4)

Tasks

We generated four representative tasks, each aiming to test for different aspects of network data and complementary views.

- ***Two-hop***: *In a train network, find the cheapest route from RED node to BLUE node*. This task translates into finding the path with the lowest overall edge weight between the two highlighted nodes. The target path contained always two hops (2 links) for this task. In *Node-link*, users had to search for narrow lines between nodes, in *Matrix* participants had to find the row with the smallest two squares (summed size) while comparing the two columns representing both highlighted nodes. In *Combined*, we hypothesised participants would search for a set of potentially low-weighted paths in the Node-link view, and then check the path weights in the matrix view. Answers involved a set of candidate paths as well as *"None of the above"* and *"Too difficult"*.
- ***Highest-degree***: *In a social network, find out which person has the most contact with other people*. This task translates into finding the most connected node (highest degree) with the strongest connections. In *Node-link*, participants had to find the node with the thickest lines attached; in *Matrix*, participants had to find the row (or column) with the largest squares (sum of size); in *Combined*, we expected participants to start with one view and check in the other.
- ***Three-hop***: *In a train network, find the cheapest route from RED node to BLUE node.* This task was a variant of *Two-hop*, except that the shortest path included 3 links. In *Matrix*, participants had to search for 2 intermediate nodes by trying different paths. We expected *Matrix* to perform badly and that in *Combined* participants would rely on node-link.

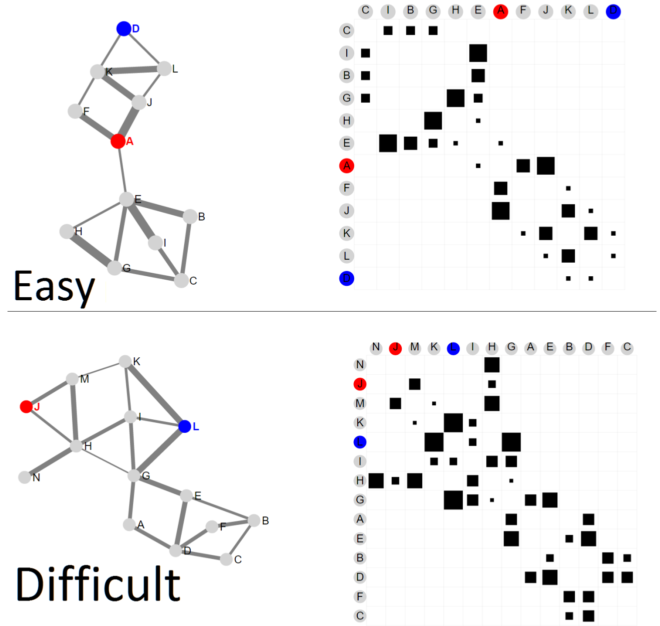


Figure 4. Samples of stimuli used in Study 2. Side-by-side node-link and matrix views with two difficulty levels. Here, participants had to find the path with the lowest cumulative edge weight between the red and blue nodes.

- ***Triangles***: *Count how many triangles are in the network.* In *Node-link*, participants counted all triples of nodes that where connected through lines; in *Combined* users had to check for triangles basically through combinations and the counting of edges. As with *Three-hop*, we expected *Matrix* to perform worse for this task and that in *Combined*, participants would rely on Node-link. Both, this task and *Three-hop* where inserted to test how do participants use complicated visualisations in a combined view setup.

Hypotheses

Based on our initial questions, stated in the introduction, we formulated three specific hypotheses.

- H_0: There is no difference in time and error across techniques, for each task (null-hypothesis).
- $H_{overall}$: *Combined* will be at least as accurate as the best of the two individual techniques *Matrix*and *Node-link*.
- $H_{combined}$: we expected *Combined* to be more accurate for *Two-hop* as participants would spot candidate paths in the node-link view, and verify path strength in the matrix view.

Data

We generated synthetic data in order to ensure the generalisability of our results. All networks had 29 nodes and a density of 5-20%. Networks were generated using Barabasi's preferential attachment method [2]. We created 3 additional datasets for training with a smaller size and density. Similar to Study 1, edge weights had 6 levels. We created 10 datasets for each trial and reused them across tasks containing 2 levels of difficulty (5 × easy, 5 × difficult). To control for difficulty level, the three conditions used the same ten networks. Labels were randomised and layout of matrix and node-link representations were rotated and mirrored differently for every trial to avoid a learning effect. Visualisation size was the same for all visualisations across all conditions. Presentation order was varied to counter-balance the learning effect. Two difficulty levels, *easy* and *difficult*, were introduced by adjusting the density of the data across the ten datasets and the differential in edge weights. Difficult datasets were two times as dense as the easier ones and featured only 10-20% difference between the correct answer the the closest alternative, while the easy datasets featured a difference of 20-60%. Examples of tasks with different difficulty levels are shown in Figure 4.

Participants and Setting

We recruited 29 participants (12 female, 17 male), all with normal or corrected-to-normal vision and without colour vision impairment. Participants included university students and researchers. The study run on an Intel Core i7 surface pro 3 and a 24-inch screen (1920 × 1080). During *Combined*, participants were equipped with a Tobii X3-120 Eye tracker, which was recalibrated at the start of each *Combined* condition. The visualisation area was centred in a full-size window and participants interacted with mouse and keyboard.

Experimental Procedure

We used a within-subject, full-factorial design: 3 techniques × 4 tasks × 13 trials. We counter-balanced the techniques, creating two groups: one group started with *Node-link* and *Matrix* (*trained*), while the other group started with *Combined* (*untrained*). We were interested in usage and performance if users are trained on both views individually, before using the combined views. There were 13 trials including 3 training trials at the start of each condition (technique × task). Each trial showed a different network, increasing in difficulty, and we measured accuracy and task completion time.

Before the controlled experiment, we instructed participants about the visualisations as well as the edge weights used in each trial, making sure none of them had any vision problems. We ask them to complete tasks as accurately and quickly as possible. Before recording the trials, three training trials were provided. Participants were guided to complete the first training task with explanation from the instructor. They answered the remaining training trials on their own unless they had any questions. Order of encoding techniques were randomised. Before each condition (technique × task), we instructed each participant on the respective task and technique. Then, participants concluded the three training trials under supervision and could ask questions. While training was not timed, we limited the time in the remaining 10 trials to 20 seconds (30 seconds for *Three-hop*) in order to avoid participants being too careful in counting and calculating. Participants were notified before the experiment and a timer progress bar was displayed on the top of the screen. To answer the trials, participants pressed the space bar to view the answer options. At that point, the timer stopped and the visualisation disappeared.

Results

For each of our 4 tasks, we obtained 10 trials × 3 techniques × 29 participants = 870 trials (3480 trials in total). We removed

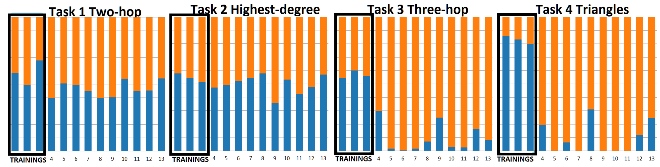


Figure 5. Relative duration (bar-height) spent looking at each view in *Combined*: orange=node-link view, blue=matrix view. Numbers indicate trials. The first 3 trials were training.

Tech.	Measure	*Two-hop*	H.-degree	*Three-hop*	*Triangles*
Time	*Node-link*	***7.8**	***12.6**	***11.4**	***6.2**
	Matrix	*9.9	13.6	*23.7	*18.7
	Combined	*11.2	13.8	*13.5	*7.6
Error	*Node-link*	0.07	*0.20	**0.13**	**0.11**
	Matrix	0.10	**0.11**	*0.40	*0.48
	Combined	0.07	**0.12**	**0.16**	**0.14**

Table 1. Mean results across techniques. Stars indicate values that are significantly different from the other two. Bold values indicate the significantly lowest (best) values.

2 trials that had a task-completion time inferior than 1 second as we considered them accidental clicks on the space-bar. We counted the answers '*Too difficult*' and '*None of above*' as error. Reported times are for correct answers only. We decided to keep trials where the participant had hit the time limit for each task as participants were still able to give an answer and we were more interested in accuracy than in completion time. We rerun our analysis removing those turnout-trials but found the results the same with respect to significances and ranking of technique performance.

After cleaning our data, we found time and error measures to be *not* normally distributed and we could not correct this using any standard transformation (Box Cox transformation, log-transformation). We made sure that this was not an artifact of including trials that hit the time-limit. Time and errors were analysed individually for each task. Using the non-parametric FRIEDMANS' TEST for one-way factorial analysis between techniques per test with a significance level of $p < 0.05$. For pair-wise comparison, we used MANN WHITNEY U TEST as removing trials resulted in unequal sample sizes. We did not find any effect of task difficulty on time or error.

In the following, we report on results with a significance level of $p < 0.05$ (*), $p < 0.01$ (**), and $p \leq 0.001$ (***) , for each task individually. Numbers in brackets indicate mean values in seconds (time) and mean-errors. Results are summarised in Table 1 and Figure 6.

Eye-tracking data: For each trial in *Combined* we measured the duration spend looking at each view (Figure 5). We also observed eye-tracking data individually for each participant and identified three main strategies (Figure 1): (a) participants sticking to one view to solve the task (exclusive use), (b) participants using both but spending a long time on each view (sequential use) and (c) participants switching frequently between both views (parallel use). We refer back to these strategies, for each task individually.

***Two-hop*:** For *time*, high significance (**) was found between all techniques; *Node-link* was fastest (7.8sec, SD=3.9), *Ma-*

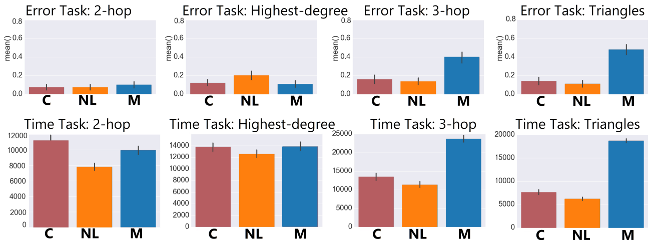


Figure 6. Results for Study 2 (Matrix–Node-link): *error* (top) and *time* (bottom) for all techniques.

trix ranking in the middle (9.9sec, SD=4.4) and *Combined* was slowest (11.2sec, SD=5.5). For *error*, FRIEDMANS' TEST did *not* found any significant difference between *Node-link* and *Combined* both having 0.07 errors on average, and *Matrix* with 0.1 error on average. These results do not support $H_{combined}$, though supporting at least $H_{overall}$. Time was surprisingly higher for *Combined* than for *Node-link*, the fastest individual view for this task. Eye-tracking data suggested that participants spend an equal amount of time looking at each view relying on both strategies: sequential and parallel (Figure 5).

***Highest-degree*:** For *time*, we found *Node-link* to be significantly faster (**), though at a very narrow margin; 12.6sec, SD=5.45 for *Node-link*, and 13.9sec, SD=5.7 for *Matrix* and 13.8sec, SD=5.4 for *Combined*. For *errors*, we found significance (**) for *Node-link* (0.2) being about twice as inaccurate as *Matrix* (0.11) and *Combined* (0.12). Again, we found support for $H_{overall}$ but contradict H_0, with the same result as *Two-hop* that *Combined* did not increase performance. However, *Combined* was not very much slower than *Matrix*. Surprisingly, participants where faster with *Node-link* but yielded more errors. This may suggest an effect of false confidence in a technique well known by participants. Eye-tracking data shows a similar pattern as for *Two-hop*; participants spend roughly the same time on looking at both views.

***Three-hop*:** For time, we found significance (**) between all techniques; *Matrix* was slowest (23.7sec, SD=6.4), *Node-link* fastest (11.4sec, SD=6.4), and *Combined* (13.5sec, SD=7.7) slightly slower than *Node-link*. With *Matrix*, participants hit the time limit in 35% of the trials (98 out of 280). We conjecture that some participants did this on purpose in order to have more time to solve the task. For *error*, we found *Matrix* (0.4) to be significantly (***) less accurate than *Node-link* (0.13) and and *Combined* (0.16). We expected *Matrix* to be slowest and less accurate due to increased cognitive effort in finding the three hops. Again, our results support $H_{overall}$ and contradict H_0. The eye-tracking data for *Combined* (Figure 5(c)) shows that during training with *Combined* participants spend about the same amount of time on both views. However, after training (trials 1-3), participants mainly decided to work with mostly the Node-link view using an exclusive strategy.

***Triangles*:** For time, we found significance (***) between all techniques: *Node-link* was fastest (6.2sec, SD=3.2), followed by *Combined* (7.6sec, SD=4.7) and *Matrix* was slowest (18.7sec, SD=2.7). Similar to *Three-hop*, we found a high rate of time-turnouts: 39% (111 out of 280) of trials showed

Technique:	*Node-link*	*Matrix*	*Combined*
Percentage using view in *Combined*	44%	5%	53%
User Preference:	34.48%	13.79%	51.72%

Table 2. View usage and user preferences in study 2.

participants hitting the time limit. For error, we found *Matrix* (0.48) significantly (***) worse than the other two techniques (*Node-link* 0.11, *Combined* 0.14). Again, these results support $H_{overall}$. During training in *Combined*, participants did spend a larger amount of time on the matrix view, however relying almost exclusively on the node-link view. One participant used the matrix view for the first few trials after training but then switched to node-link and one participant exclusively used the matrix view.

View Use: Based on the eye-tracking data, visualised as shown in Figure 5, we counted the number of participants using both views, and those exclusively using one of the two views. Results are summarised in Table 2. When we compared the actual task performance for each of these user groups, but could not find any significant difference, for no task. Yet, mean-error rate for users actively switching between both views was slightly higher. We also compared strategies of users who started with *Combined* (*trained* group), against those who started with the individual views and were already trained in both views. We found that participants starting with *Combined* had a learning curve to find their strategy. Eventually, participants from this group consistently used both views. However, participants who used *Combined* last had a clear preference and were more likely to stick to one visualisation. In these cases, node-link was generally preferred for task *Three-hop*, and *Triangles* task. Even those participants who generally preferred matrix were more likely to check their guess with node-link.

Overall User Preferences: After the study, we asked each participant which visualisation technique they did most prefer overall. *Combined* was most preferred by half of the participants (51.72%), *Matrix* was the least preferred (13.79%) and *Node-link* ranking in between (34.48%). We can see this as evidence that users felt more comfortable with both views. However, half of the participants where fine with a single view and did not seem to prefer multiple views.

Discussion

Our study aimed to assess for which tasks the individual techniques differ in performance (Q1), what is the effect of showing both views side-by-side (*Combined*) (Q2, $H_{overall}$, $H_{combined}$), and which strategies participants employ while working with two views (Q3). For the two tasks that were comparable to Ghoniem's *et al.* [10] path-following task (two-hop, three-hop) we confirmed their findings. However, our study involves weighted edges, ordered matrices, and both visualisations simultaneous.

Node-link vs. Matrix (Q1) A task-by-task analysis revealed significant differences between techniques for individual tasks, i.e. the choice of technique influences accuracy and time. Thus we can reject H_0. We found there may be general bias towards *Node-link*, being the more familiar visualisation technique. We found each *Matrix* and *Node-link* are each better for different tasks. Yet, overall, we found more positive results and user preferences for *Node-link*. Thus, we conjecture both individual views to be complementary in their overall use.

Performance of Combined (Q2) Half of the participants (51.72%) preferred working with the *Combined* condition. The overall preference for *Combined* and the number of participants using both views, supports this complementariness. For *Combined*, we did not find a significant effect proving that side-by-side views increase error compared to the individual techniques. We can still accept $H_{overall}$ but have to reject $H_{combined}$. Our eye-tracking data suggests that participants use both views when they are provided with the opportunity and in many cases use both views, independent from whether there is one view that is clearly better (e.g. tasks *Three-hop*, *Triangles*). However, in general, if participants found one view is clearly better, they tended to mainly use this view.

User Strategies (Q3) Only half of the participants actively switched between both views, when provided with *Combined* and switching had no effect on the performance. For tasks *Three-hop* and *Triangles*, the eye-tracking data suggests that participants were correctly using the better visualisation (*Node-link* in both cases). We could not observe the same effect for *Highest-degree*, as participants spend the same amount of time on both views, though *Matrix* was significantly more accurate. Though the difference between *Node-link* and *Matrix* was of the amplitude of 2, we have several hypothesis; participants may either: a) not have noticed a difference in accuracy; b) felt more confident with the node-link visualisation; or c) used the second visualisation for cross-checking.

STUDY 3: MATRIXWAVE AND SANKEY DIAGRAM

In our third and last study, we extended our investigations to complementary views for flows between nodes in a networks over several timesteps. We compared two visualisations commonly used for this task; MatrixWave [26] and Sankey diagrams [19]. As in our Study 2, we recorded eye-tracking data for every participant in the *Combined* condition.

Techniques

- ***Sankey*** diagrams visualise the weights of flows between nodes in a network using curved lines between vertical bars (Figure 7-right), Each bar represents a different timestep.
- ***MatrixWave*** is a zig-zag tiling of matrices showing the flow of data. Each cell in the matrix represents the volume of the flow between the corresponding two nodes (Figure 7(left)). Each matrix represents transitions between two timesteps.
- ***Combined*** shows MatrixWave and Sankey side-by-side, coordinated through brushing-and-linking; hovering on a path/rectangle/row/column/cell/node highlighting its counterpart element in both complementary views (Figure 7).

Tasks

We selected four representative tasks for each level-of-detail. We used the scenario of companies and flows of employees to better explain the tasks.

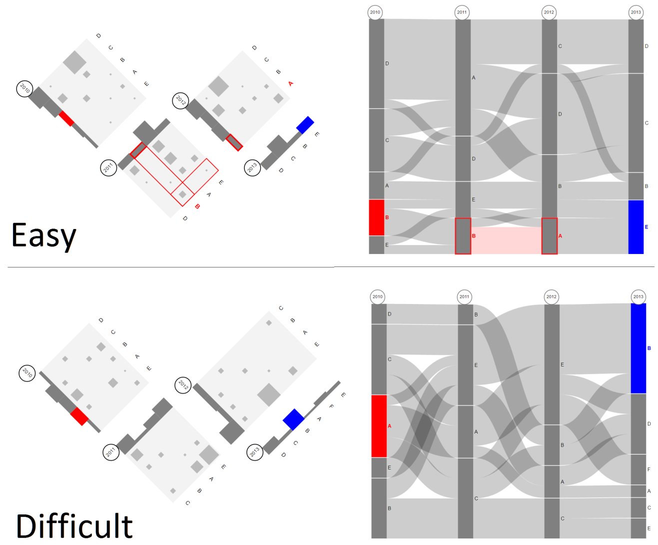


Figure 7. Examples of Easy and Difficult side-by-side view stimuli used in the *Path-following* task for Study 3. Participants need to find the most likely path from red node to blue node. The link is highlighted on both views when the mouse is hovered over any part of it.

- ***Out-link***: *For an employee working in the red organisation in 2010, which organisation are they most likely to be working for in 2013?* This task required participant to identify the largest flow over time; between the red node in the first stage (2010) and any node in the final stage (2013). Participants needed to identify and estimate the possibilities on each stage and summarise in the final stage.
- ***Largest-flow***: *Which is the largest flow in the graph?* In *Sankey*, participants had to find the link with the thickest stroke; in *MatrixWave*, participants need to identify the largest square in any matrices.
- ***Path-following***: *If someone works for the red organisation in 2010 and for the blue organisation in 2013, which organisations are they most likely to have worked at in 2011 and 2012?* This task translates into finding the strongest path between two nodes. Participants needed to locate the source and target nodes, then identify the most possible routes in the middle 2 hops (Figure 7).
- ***Return***: *Which organisation do people leave but never return to?* In this task, participants needed to quickly iterate all the nodes and verify if there is any possible path between the node in the first and last stage.

Hypotheses

Hypotheses were similar to those in Study 2, reflecting our initial questions Q1-Q3.

- H_0: There is no difference across techniques in time and error for each task.
- $H_{combined}$: *Combined* will be at least as accurate as the best of the two individual techniques (*MatrixWave*, *Sankey*).

Participants, Setting, and Data

We recruited 24 participants (9 female, 15 male). Participants were university students and staff; all of normal or corrected-to-normal vision and without any colour vision impairment. The survey equipment configuration was exactly same with

Study 2. Again, data was generated synthetically. In each dataset, there were 4 time steps, each with 4-6 nodes and 5-20 flows for each step. Flow weight was assigned randomly ranging over 6 levels. We introduced two levels of difficulty ($6 \times$ easy, $6 \times$ difficult), varying in flow density and value difference. The same twelve graphs were used across conditions with labels and presentation order randomised, and layouts mirrored or rotated in each condition to avoid a memory effect. Visualisation size was the same in all conditions.

Experimental Procedure

We used a within-subject, full-factorial design: 3 techniques $\times$ 4 tasks $\times$ 15 trials. We counter-balanced techniques in four orders to obtain our 2 groups (*trained, untrained* as in Study 2). 15 trials per condition included 3 training trials in the beginning of each condition (technique x task). Each of the 15 trials showed one of 15 networks, in a fixed order and increasing in difficulty. For each trial, we measured accuracy and task completion time.

Experimental procedure was the same as in Study 2. Before each condition, participants concluded 3 training trials under supervision and could ask questions. While training was not timed, we limited the time in the remaining 12 trials to 20 seconds (30 seconds for *Path-following*). Participants were notified before the experiment that a timer progress bar was displayed on the top of the screen. To answer the trials, participants pressed the space bar.

Results

Similar to our second study, we analysed data for each task individually. We obtained 24 participants $\times$ 12 trials $\times$ 3 techniques = 864 trials per tasks. We found the time-out rate relatively high for this study, but decided to include these trials since time-out was consistent across techniques and depended on the task only. We repeated our analysis *excluding* these trials, and could not find major differences. Again, we found time and error not normally distributed and used FRIEDMANS' TEST for one-way analysis and MANN WHITNEY U TEST for pair-wise comparisons. Significance levels are indicated as $p < 0.05$(*), $p < 0.01$(**), and $p \leq 0.001$(***). We refer to the results for each task individually, summarised in Table 3 and Figure 8.

For the eye-tracking data, we found the same three strategies as in Study 2 (Figure 1). Duration spent on each view is shown in Figure 9.

Out-link: For time, we found *Sankey* significantly (**) faster (9.8s, SD=5.3s) than any of the other other two techniques (*MatrixWave*: 11.9s, SD=5.5 and *Combined*: 11.2, SD=5.9s). We did not find significant difference for error (0.09, 0.1, 0.08).

Largest-flow: For time, we did not find any significant difference between *Sankey* (7s, SD=3.8) and the other techniques (6.6s, SD=3.8 for *MatrixWave* and 6.7s, SD=4.0 for *Combined*). For error, we found a slight trend (p=0.068) for *Combined* being more accurate (0.05) than *MatrixWave* (0.08) and *Sankey* (0.08). Our results contradict H_0 with respect to time, but confirm it with respect to error. We could not support $H3$. What we did find surprising was the time participants

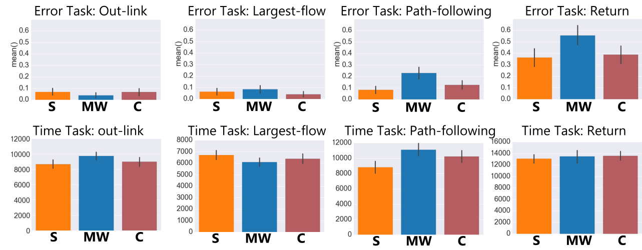


Figure 8. Results for Study 3: MatrixWave (MW), Sankey(S), and combined (D): *error* (top) and *time* (bottom) for all tasks.

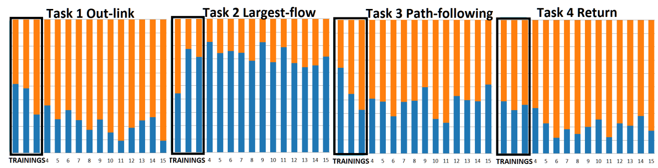


Figure 9. Overall time spent looking at each view in the *Combined* condition: orange=*Sankey*, blue=*MatrixWave*.

spend looking at each view, in *Combined*; despite *Sankey* being about as accurate as *MatrixWave*, overall participants spend much more time looking at *MatrixWave*. Eye-tracking data for *Combined* by participant showed that 4/9 exclusively used *MatrixWave* while briefly gazing over to *Sankey* (similar to Figure 1(a)). Other participants switched between views and were faster in completing the trials. We believe there is a clear advantage to switching between both views.

Path-following: For time, we found significant differences between all techniques: *Sankey* was fastest (9.2s, SD=6.8), *MatrixWave* was slowest (11.6s, SD=6.81) and *Combined* ranking in between (10.8s, SD=7.0). For error, we found *MatrixWave* significantly (***) less accurate (0.24) than *Sankey* (0.09) and *Combined* (0.12). we observed a light trend (p=0.068) between *Combined* and *Sankey*. These results support both hypothesis; H_0 as we did found *Sankey* to be more accurate than *Matrix*, and $H_{combined}$. Individual eye-tracking revealed most participants employing a sequential strategy, 2 used a parallel strategy, and 1 exclusively used *MatrixWave*. We hypothesise that participants searched for the path in MatrixWave and quickly confirmed via Sankey.

Return: For time, we found no significant difference between techniques. We conclude that the time limit was too short and too many participants ran out of time. Though the error rate was generally high for each technique (*Sankey*: 0.36, *MatrixWave*: 0.55, *Combined*: 0.39) we found a significant difference (***) for *Matrix* being most inaccurate, but found no difference between *Sankey* and *Combined*. Again, these results contradict H_0, but support $H_{combined}$. Similar to *Path-following*, the difference in error was high between *Sankey* and *MatrixWave*, but still participants spend some significant amount of time looking at *MatrixWave*. The individual data revealed that all participants used the sequential strategy.

View Use: Comparing the performance of participants in *Combined* by view use (Sankey-exclusive, MatrixWave-exclusive, and both) we did not find any significant difference but

Tech.	Measure	*Out-link*	Larg.-flow	Path-foll.	*Return*
Time	*Sankey*	***9.8s**	7.0s	***9.2s**	16.3s
	MatrixWave	11.9s	6.6s	*11.6s	16.8s
	Combined	11.2s	6.7s	*10.8s	16.5s
Error	*Sankey*	0.09	0.08	**0.09**	**0.37**
	MatrixWave	0.10	0.08	*0.24	*0.56
	Combined	0.08	0.05	0.12	0.40

Table 3. Study 3: mean results across techniques. Stars indicate values that are significant different from the two other. Bold values indicate the significantly lowest (best) values.

Technique:	*Sankey*	*MatrixWave*	*Combined*
User Preference:	54.17%	12.50%	33.33%
Percentage using view in *Combined*	47.6%	28.6%	28.8%

Table 4. View use and user preferences in study 3.

we found consistently lower error rates for those using MatrixWave. Percentages of participants using views exclusively, or relying on both views together, are given in Table 4. When comparing strategies of users starting with *Combined* (*untrained*) to those starting with the individual views (*trained*), we found very similar results to Study 2; *untrained* participants did rely on both views while *trained* participants tend to stick to one of the two views.

Discussion

Sankey vs. MatrixWave (Q1): Compared to Matrix and Node-link in Study 2, complementaryness of *Sankey* and *MatrixWave* was not as strongly evident. We can still reject H_0, and found *MatrixWave* to result in more errors than *Sankey*. In a way, this contradicts the results found in [26]. However, study set-up, tasks, and data sets varied across studies. Our data sets were smaller and their tasks were about comparisons.

Performance of Combined (Q2): We can confirm that *Combined* was always as good as the best of the individual techniques (accepting $H_{combined}$). However, *Combined* was never more accurate than any individual techniques. Similar to Study 2, we did not find any difference in performance for people using both views together or exclusively using one view in the the *Combined* condition.

User Strategies (Q3): Contrary to our findings from Study 2, more participants preferred one of the individual views, than the combined one. *Sankey* was preferred by more than half of the participants, while *Matrix* was ranked last. These results are in accordance with the measures we found for performance. We also think that combing *Sankey* and *MatrixWave* resulted in an increased visual complexity, which for users did not pay off. Still, a combined display attracted much greater preference than a single matrix and most participants used both views in a sequential manner (25/36 cases). Only 6/36 cases used *MatrixWave* exclusively.

SUMMARY AND CONCLUSIONS

We set out to explore the effect of perceptually complementary views for visualising weighted networks and event flows. Within the limits of our tested conditions, we can summarise our findings as follows:

Square and FatFonts are good choices for encoding edge weights in matrices. Both encodings performed well for our tasks and attracted a high user rating. The widely used opacity encodings performed badly for detailed value comparison.

Combined views are not worse than individual views. We found coordinated views as good as individual views and participants did not take longer to work with them. We see this as evidence that participants are able to chose the more performant view. However, we did not find evidence that combined views are more accurate for our tested tasks. More complex and compound tasks may yield different effects. However, studies involving more complex tasks also involve different strategies and require more experienced users. Both may increase result variability.

Trained users tend to stick to the view of their preference. Users new to combined views tend to explore both views and try to solve the task with both views. Users trained in (both) individual views tend to use individual views on a per-task basis. Our participants, though untrained in *each* of our tested visualisations, were quickly able to decide on the better view for a certain task. As we did not obtain bad results about combined views, both for time, error, and user preference, our results generally support the use of perceptually complementary views for network visualisation tasks.

FUTURE WORK

In future, we want to extend our studies towards other perceptually complementary views as well as informationally complementary visualisations, e.g., for tree and hierarchical data and dynamic networks (e.g., views showing topology combined with views showing temporal evolution for dynamic networks).

Future investigation is needed to understand how to measure the degree of perceptual complementariness of two representations. Compressing side-by-side complementary views into hybrid techniques is an interesting direction for future research. There is currently no empirical evaluation of hybrid techniques such as NodeTrix [13] and MatLink [12]. Eventually, we aim to identify tasks where perceptually complementary views actually improve task accuracy and/or completion time.

ACKNOWLEDGEMENTS

This research was supported under Australian Research Council's Discovery Projects funding scheme (project number DP140100077).

REFERENCES

1. Basak Alper, Benjamin Bach, Nathalie Henry Riche, Tobias Isenberg, and Jean-Daniel Fekete. 2013. Weighted graph comparison techniques for brain connectivity analysis. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 483–492.
2. Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.
3. Jacques Bertin. 1973. Sémiologie graphique: Les diagrammes-Les réseaux-Les cartes. (1973).
4. Anastasia Bezerianos, Pierre Dragicevic, Jean-Daniel Fekete, Juhee Bae, and Ben Watson. 2010. Geneaquilts: A system for exploring large genealogies. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1073–1081.
5. Michael Bostock. 2012. D3. js. *Data Driven Documents* (2012).
6. Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. 1999. Readings in information visualization: using vision to think. (1999).
7. Christopher Collins and Sheelagh Carpendale. 2007. VisLink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1192–1199.
8. Antoine de Falguerolles, Felix Friedrich, Günther Sawitzki, and StatLab Heidelberg. 1997. A tribute to J. Bertin's graphical data analysis. *SoftStat* 97 (1997), 11–20.
9. Tim Dwyer. 2016. Constraint-Based Layout in the Browser: WebCoLa. `http://marvl.infotech.monash.edu/webcola/`. (2016). Accessed: 2016-08-08.
10. Mohammad Ghoniem, J-D Fekete, and Philippe Castagliola. 2004. A comparison of the readability of graphs using node-link and matrix-based representations. In *Proceedings of IEEE Symposium on Information Visualization, 2004 (InfoVis)*. IEEE, 17–24.
11. Nathalie Henry and Jean-Daniel Fekete. 2006. MatrixExplorer: a dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 677–684.
12. Nathalie Henry and Jean-Daniel Fekete. 2007. Matlink: Enhanced matrix visualization for analyzing social networks. In *IFIP Conference on Human-Computer Interaction*. Springer, 288–302.
13. Nathalie Henry, Jean-Daniel Fekete, and Michael J McGuffin. 2007. NodeTrix: a hybrid visualization of social networks. *IEEE transactions on visualization and computer graphics* 13, 6 (2007), 1302–1309.
14. René Keller, Claudia M Eckert, and P John Clarkson. 2006. Matrices or node-link diagrams: which visual representation is better for visualising connectivity models? *Information Visualization* 5, 1 (2006), 62–76.
15. Miguel Nacenta, Uta Hinrichs, and Sheelagh Carpendale. 2012. FatFonts: combining the symbolic and visual aspects of numbers. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, 407–414.
16. Chris North and Ben Shneiderman. 1997. A taxonomy of multiple window coordination. (1997).
17. Lucy Nowell, Robert Schulman, and Deborah Hix. 2002. Graphical encoding for information visualization: an empirical study. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*. IEEE, 43–50.
18. Charles Perin, Pierre Dragicevic, and Jean-Daniel Fekete. 2014. Revisiting bertin matrices: New interactions for crafting tabular visualizations. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2082–2091.
19. Patrick Riehmann, Manfred Hanfler, and Bernd Froehlich. 2005. Interactive sankey diagrams. In *Proceedings of IEEE Symposium on Information Visualization, 2005. (InfoVis)*. IEEE, 233–240.
20. Jonathan C Roberts. 2007. State of the art: Coordinated & multiple views in exploratory visualization. In *Proceedings of International Conference on Coordinated and Multiple Views in Exploratory Visualization, 2007.* IEEE, 61–71.
21. Stanley S Stevens. 1971. Issues in psychophysical measurement. *Psychological review* 78, 5 (1971), 426.
22. Matthew Tobiasz, Petra Isenberg, and Sheelagh Carpendale. 2009. Lark: Coordinating co-located collaboration with information visualization. *IEEE transactions on visualization and computer graphics* 15, 6 (2009), 1065–1072.
23. Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. 2000. Guidelines for Using Multiple Views in Information Visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '00)*. ACM, New York, NY, USA, 110–119. DOI: `http://dx.doi.org/10.1145/345513.345271`
24. Krist Wongsuphasawat and David Gotz. 2012. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2659–2668.
25. Jiaje Zhang and Donald A Norman. 1994. Representations in distributed cognitive tasks. *Cognitive science* 18, 1 (1994), 87–122.
26. Jian Zhao, Zhicheng Liu, Mira Dontcheva, Aaron Hertzmann, and Alan Wilson. 2015. MatrixWave: Visual comparison of event sequence data. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. ACM, 259–268.