Paper:

Archambault, D. & Purchase, H. (2016). On the effective visualisation of dynamic attribute cascades. *Information Visualization, 15*(1), 51-63.

http://dx.doi.org/10.1177/1473871615576758

On the Effective Visualisation of Dynamic Attribute Cascades

Daniel Archambault[1] and Helen C. Purchase[2]

[1]Department of Computer Science, Swansea University, UK*
[2]School of Computing Science, University of Glasgow, UK†

Abstract

Cascades appear in many applications, including biological graphs and social media analysis. In a cascade, a dynamic attribute propagates through a graph, following its edges. We present the results of a formal user study that tests the effectiveness of different types of cascade visualisations on node-link diagrams for the task of judging cascade spread. Overall, we found that a small multiples presentation was significantly faster than animation with no significant difference in terms of error rate. Participants generally preferred animation over small multiples and a hierarchical layout to a force-directed layout. Considering each presentation method separately, when comparing force-directed layouts to hierarchical layouts, hierarchical layouts were found to be significantly faster for both presentation methods and significantly more accurate for animation. Representing the history of the cascade had no significant effect. Thus, for our task, this experiment supports the use of a small multiples interface with hierarchically drawn graphs for the visualisation of cascades. This work is important because without these empirical results, designers of dynamic multivariate visualisations (in many applications) would base their design decisions on intuition with little empirical support as to whether these decisions enhance usability.

Keywords: Cascades, Social Networks, Animation, Small Multiples, Empirical Evaluation

*Department of Computer Science, Faraday Tower, Swansea University, Singleton Park, Swansea, SA2 8PP, United Kingdom email: d.w.archambault@swansea.ac.uk

†School of Computer Science, University of Glasgow, Room S104, 17 Lilybank Gardens, Glasgow, G12 8QQ, United Kingdom email: Helen.Purchase@glasgow.ac.uk

1 Introduction

A **multivariate network** consists of a network with attributes associated with the nodes and edges of the graph [32]. In the case of dynamic multivariate networks [3], these attributes can change their value over time. **Dynamic attributes** play an important role in several application areas. In biology, visualising expression or concentration level on interaction networks can help scientists understand the impact of experimental conditions [11]. In social media analysis, information, ideas, and influence can propagate through a social network [13, 17, 24, 47], and identifying those nodes that play an important role in rapidly increasing the rate of spread of such information thorough a social network is of particular interest [17].

In the applications mentioned above, the underlying structure of the graph usually remains static and it is the value of the attributes associated with the nodes and edges that change over time, representing the transmission of information. Nodes that take on a non-zero value of a dynamic attribute are **infected**–the term used in Yang *et al.* [47] and other works in the social media analysis literature. Long sequences of infections form **cascades** that propagate through the network along the graph edges. In this work, we look at the task of identifying nodes that amplify the spread of a cascade – nodes that infect many of their neighbours. While some empirical work has been conducted on the presentation of dynamic graphs, graphs whose structure changes over time through node and edge insertions/deletions [6, 8, 21], no empirical evaluations have been conducted on the visualisation of dynamic attributes and of the cascades that result from a sequence of node infections over time.

In many of these applications, the graph is represented visually using a node-link diagram and the progress of the infection is represented by changing node colour between timeslices [11], where each timeslice represents a snapshot of the graph at a given time. Nodes that are infected are given a saturated colour while those that are not infected are represented with a neutral colour. The resultant time series can be visualised using animation and/or small multiples presentations. In an **animation** of the data, the time series is represented as an interactive movie (or *interactive animation*). In a **small multiples** representation [42], the timeslices are depicted in a matrix of images. Small multiples has been used for visualising cascades in biological data [11], but the approach was not empirically evaluated for effectiveness.

The graph is typically depicted using an existing graph drawing algorithm. Frequently, **force-directed** graph drawing approaches are used to draw these graphs. However, **hierarchical** graph drawing techniques can also be used to support the visualisation of cascades as the cascade flows through the graph following the direction of the edges. In a hierarchical graph drawing approach, the nodes are placed into layers and the graph is drawn in such a way that directed edges, for the most part, are directed downwards. In one approach[1], two views, one force-directed and another hierarchical, are used to visualise the spread of cascades on Twitter. The presenter argues that the two views can be complementary. Barsky *et al.* [11] uses hierarchical graph drawings in his system for biological reasons. However, no experiments empirically evaluate the effectiveness of hierarchical drawings in the context of visualising cascades.

We present the results of an experiment which formally evaluated different means of visualising cascades across a static graph. Our primary research question is:

1. Which presentation method best supports the identification of those nodes which amplify an cascade: small multiples or animation?

This question is supplemented by two secondary questions:

a) Which graph layout method best supports this task: force-directed or hierarchical graph drawing methods?

b) Does encoding the history (that is, the values of the cascade in previous timeslice) help with this task?

[1] `https://twitter.com/zephoria/status/309346169732079616`

We chose these research questions and encodings for our experiment based on the design decisions of previous systems [11] in the information visualisation literature. These encodings have yet to be empirically validated. As we had no prior expectation as to their potential effectiveness, we wished to investigate whether these intuitive design decisions were appropriate.

The contribution of this paper is the presentation of the first formal experiment to test these research questions in the context of visualising cascades on node-link diagrams, providing empirical support for the design decisions taken in the development of these visualisation systems. The results of our experiment support the use of small multiples with hierarchically drawn graphs for the visualisation of cascades.

2 Related Work

We begin this section by providing an overview of previous work in dynamic data visualisation. Secondly, we provide a brief overview in terms of the simulation and analysis of cascades.

2.1 Dynamic Graphs

In this section we discuss related visualisation techniques and empirical evaluations with respect to node-link representations of dynamic graphs.

2.1.1 Dynamic Data and Networks

There are several approaches for visualising how attributes change over time on graphs. Barsky *et al.* [11] propose a small multiples representation to visualise changes in expression levels on biological networks over time. In this approach, colour is used to represent the process and only the nodes of the graph are highlighted. The results of several biological experiments can be compared side by side using the technique. A biologically inspired hierarchical graph drawing algorithm is introduced in the paper and draws the graph in such a way to orient most of the edges downwards. The layers of the drawing represent parts of the cell with the top of the diagram representing extracellular regions and the bottom of the diagram representing the interior of the nucleus. Yi *et al.* [48] describe a method to depict changing attributes over time for social networks. The approach represents the dynamic attributes using a variety of bar chart techniques embedded in the cells of an adjacency matrix. Brandes and Nick [14] present a method for representing asymmetric attributes on social networks. The technique uses glyphs in an adjacency matrix representation to indicate the evolution of attribute values at given times. Viégas *et al.* [44] present a method for visualising cascades using circle packing and containment. Cascades trees are represented using containment with the parent node of the cascade containing all of its children.

Although many techniques for visualising cascade data have been explored, no empirical evaluations have been performed to assess their effectiveness. Our experiment empirically evaluates node-link diagram cascade representations similar to Barsky *et al.* [11].

A number of other early efforts focused on ways ways for using the third dimension for the visualisation of dynamic processes on node-link diagrams (2.5 dimensional methods). Koike [33] present a system for visualising the execution of concurrent systems in three dimensions. The visualisation method combines processor and process execution into a singular three dimensional encoding to avoid duplication of visualisation elements. Brandes and Willhalm [15] present a landscape visualisation of citation graphs. Attributes, such as citation information are extruded into the third dimension.

Although these visualisations have been used for representing dynamic processes on node-link diagrams, they often suffer from the drawbacks of occlusion and perspective foreshortening. To our knowledge, such techniques have not been used often to visualise cascades and there is some empirical evidence [40, 41] that 2.5 dimensional methods may not be as effective as originally thought. Thus, we do not test these techniques in our experiment.

2.1.2 Experiments in Dynamic Graph Visualisation

Many formal user studies have been run to evaluate methods for visualising dynamic data. Heer *et al.* [26] evaluate a variety of animated transitions between common statistical chart such as bar graphs, pie charts, and scatterplots. The authors found that staged, animated transitions can help for tracking parts of the graph through the animation and changing values. Robertson *et al.* [36] evaluate representations of dynamically evolving scatterplots similar to those used in Gapminder. Gapminder[2] conveys dynamically evolving scatterplots of statistics about various countries to make arguments about their development through animations. Animation, small multiples, and trace line techniques were tested in the study. The study found that small multiples was the most effective method for visualising this data on the types of tasks tested. Chevalier *et al.* [18] describe an experiment to test animation as a method for highlighting changes in text. In this experiment, the authors found that animation was beneficial in helping illustrate these changes to the document when compared to no transitions where changes popped into the visualisation instantaneously. This effect is similar to a PowerPoint slide show that does not use animated transitions.

In the area of dynamic graph drawing, animated transitions have also been shown to help in a variety of tasks. Bederson *et al.* [12] found that animated transitions were useful when trying to recall parts of a tree from memory. Shanmugasundaram *et al.* [38] found that animated transitions helped determine graph connectivity when panning through a diagram that does not entirely fit inside the viewport. Archambault *et al.* [7] found that animated transitions could be helpful in a difference graph setting in determining changes in the number of edges in a dynamic graph. Zaman *et al* [49] found that, for directed graphs, tasks that involve detecting the appearance of nodes or edges could be helped through animation when differences were highlighted using colour. A

[2]www.gapminder.org

number of visualisation techniques have found that various types of animated transitions can help when compared to no transitions [10, 37].

In addition to the usefulness of animated transitions, a number of human centred experiments have also been run in the area of dynamic graph drawing to test the effectiveness of animation and small multiples. Archambault *et al.* [8] compared animation and small multiples presentations on a variety of graph readability tasks. The authors found that small multiples was significantly faster overall and for most of the questions tested. Animation, however, could be helpful in tasks involving the appearance of nodes and edges. Farrugia and Quigley [21] tested animation and small multiples on social network visualisation tasks. They found that small multiples was faster on all of the tasks tested with no significant difference in terms of error rate. The same small multiples and animation representations used in these experiments are used in this experiment as well. In the context of animation and small multiples a number of studies have also tested if the stability of the drawing, known as the mental map, affects human performance. For general graphs, surprisingly many of these studies found no positive effect [4, 8, 35] but recent work [5] has found that for both animation and small multiples, stability helps with tasks such as following long paths through the graph and revisiting specific locations.

In all of the experiments described above, the dynamic graph evolved in terms of graph structure: nodes and edges were added and removed from the graph over time. The main difference between this work and the work presented here is that we do not consider structurally evolving graphs. Instead, we consider cascades – a specific case of dynamic attributes where the attribute flows through the graph based on its structure. Even though some of these experiments consider small multiples and animation as presentation methods, none of the experiments in this section consider cascades.

2.2 Cascades and Social Media

Mathematically formalising the process of information diffusion and the properties of cascades has recently been of interest to the social media analysis community. Papers take a variety of approaches including acquiring real cascades from social media networks or simulating cascades on real data from Twitter, Facebook and blogosphere networks [17, 24, 47]. Though there has been extensive study into the area of understanding and simulating the mechanics of cascades, there have been no user studies that have tried to formally evaluate their visualisation over networks.

When depicting social media graphs, force-directed representations are commonly used [29, 34]. Force-directed algorithms are the most popular method for graph drawing and can be applied generally to any graph, regardless of structure. However, when considering the visualisation problem more closely, the force-directed approaches used in these visualisations usually do not consider edge direction. Frequently, edges in social media analysis have direction: re-posting behaviour in communities of the blogosphere and meme tracking on social media services such as Facebook and Twitter. Considering this edge direction may have an effect on visualisation effectiveness.

3 Experiment

We are interested in the interpretation task of identifying nodes that substantially increase the rate of transmission of the cascade. The experiment considers the following factors: presentation method (Animation vs Small Multiples), layout (Force-Directed vs Hierarchical), and persistence (Without History vs With History). The stimuli are based on two real-world data sets (`Facebook` and `Twitter`). In this section, we describe each of these factors and the overall experimental design.

A node is coloured differently at a particular time t, depending on if it is infected or not. A node has a saturated blue colour when it is infected and a grey colour when it is not. Nodes can take on various saturation levels of blue if history is enabled (see Section 3.3).

3.1 Presentation Methods

Animation and small multiples are tested in this experiment. Both of these methods have been tested previously in experiments involving dynamic graphs that evolve in terms of graph structure [5, 8, 21].

Animation, shown in Figure 1, is similar to a movie player. Participants could hit the play/pause button at any time or drag the slider at the bottom of the screen to advance the cascade. If the participant used the slider at the start of the task, the cascade visualisation could be interacted with immediately; otherwise, the animation started to play automatically after four seconds. To control across conditions, no other form of interaction was allowed, including zooming. The entire screen was given to the animation window. Over time, the colours of the nodes changed, depending on the state of their infection. Rather than abruptly changing node colour, the colour was gradually changed using a smooth linear interpolation.

Small Multiples, shown in Figure 2, is similar to a comic book. Each of the six timeslices was placed into its own pane and participants scanned left to right to determine the changes in the cascade. No other form of interaction was allowed, including zooming. Each timeslice took up about a sixth of the screen. The nodes in each timeslice were coloured appropriately, depending on the state of infection.

Controlling the zoom level was important and necessary for this experiment: not only did we want to control for interaction costs (as in Javed *et al.* [30]) but also for colour perception as it is highly dependent on node size. To control for zoom level, we took the graph with the largest layout across all conditions and factors and ensured it was entirely visible. Thus, for some stimuli, white space is present around the drawing (Figure 2).

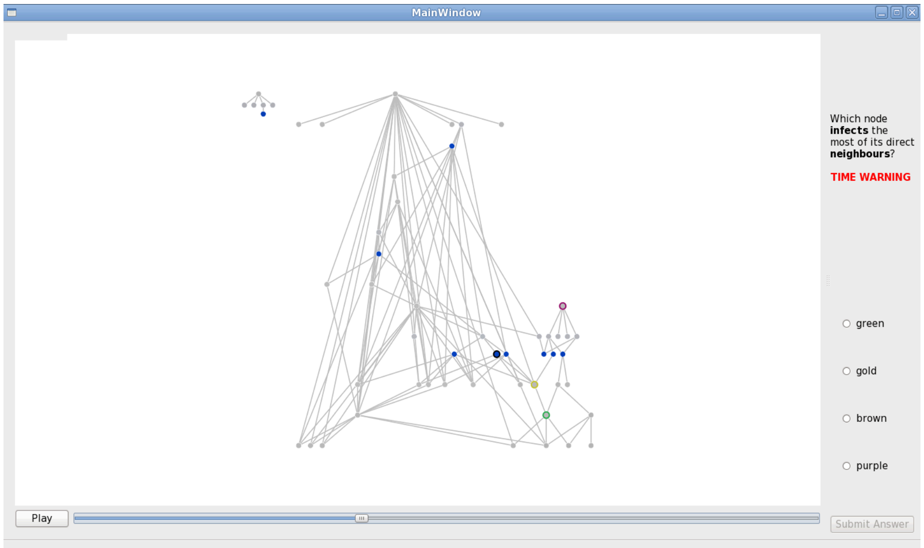


Figure 1: The animation presentation method for the experiment. Participants can pause/play the animation by clicking the button on the left. Alternatively, the participants can drag the slider at their own rate. Smooth, linear interpolation changes the colours of the nodes in the graph. A simulated cascade of `Facebook` is shown using a hierarchical layout and no cascade history. The red warning label that encourages participants to finish the task is shown.

3.2 Layouts

In this experiment, force-directed and hierarchical layouts were used to draw the cascade graphs. As graph structure did not evolve in this experiment (no nodes or edges were inserted or removed), the graphs involved in each simulated cascade were drawn once and this drawing was used for all timeslices of the graph series. The produced graphs were all directed as explained later in Section 3.4.2.

Force-directed is a typical spring layout of the graph. Figure 3(a) shows a force-directed layout for one timeslice of the data set. For this method, the direction of the graph edges are not taken into account. In this experiment, we used the GEM algorithm [22] that is implemented as part of the Tulip [9] framework. Force-directed methods that take edge direction into account [39] exist, but these approaches generally are not used in the literature. In order for our experiment to have relevance to existing

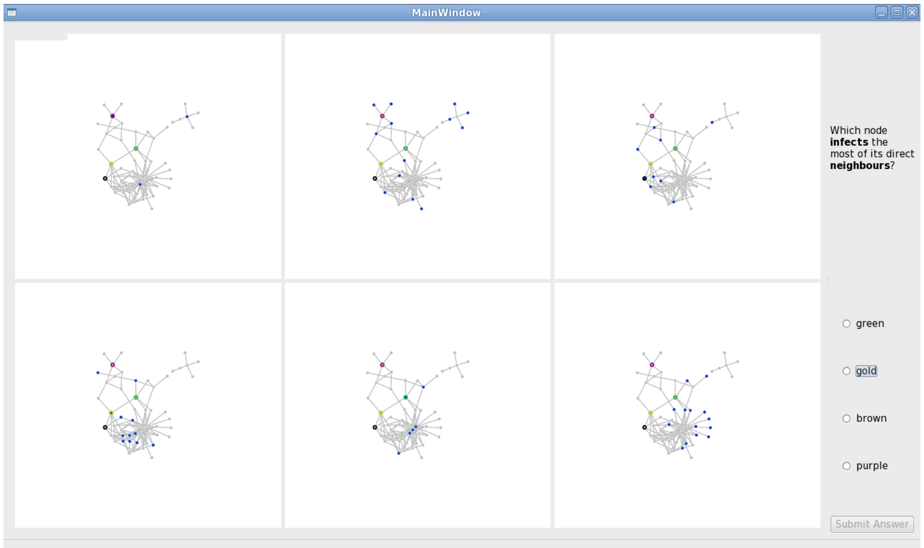


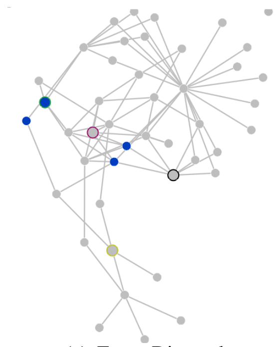
Figure 2: The small multiples presentation method for the experiment. The interface looks like a comic book where panes are read left to right and top to bottom to see how the cascade changes. The same data set in Figure 1 is shown using a force-directed layout and no cascade history.

systems that visualise cascades, we test standard force-directed approaches. One could view the force-directed condition as an *undirected* condition in the experiment.

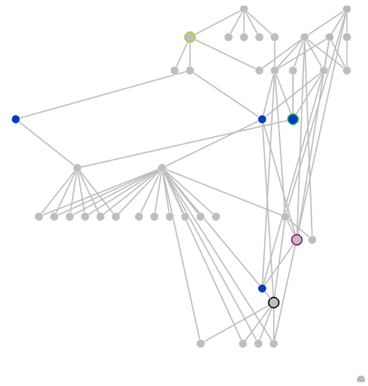
Hierarchical is a layout where most of the directed edges point downwards. Figure 3(b) shows a hierarchical layout for one timeslice of the data set. Various algorithms to draw graphs in this fashion have been used to present the evolution of dynamic graph attributes and cascades [11]. We use the OGDF [19] implementation of Gansner *et al.* [23] and chose to orient the graph top-down as this orientation has been shown to be effective [16].

3.3 Cascade History

When the cascade history condition is enabled, nodes are coloured in one of three ways (Figure 4). The saturated, blue nodes are infected in the current timeslice. The nodes which are coloured with a less saturated blue are the nodes that were infected in the



(a) Force-Directed



(b) Hierarchy

Figure 3: The force-directed and hierarchical conditions for the third timeslice of the same cascade instance of `Twitter`.

previous timeslice. Nodes that are grey are not infected in either of these timeslices. Enabling cascade history allows participants to compare multiple timeslices in a single view to alleviate slider use in animation and comparing two separate views in the small multiples condition. Encoding cascade history shares some commonalities with previous work on methods for visualising graph difference [1, 2, 7, 49]. However, this condition is inspired by Ivanov *et al.* [28] where human motion through a building was analysed. In this work, long streaks represented people moving through the hallways of the building. Our history condition implements a similar effect on node-link diagrams.

3.4 Data Sets

Two data sets were used in this experiment `Facebook` and `Twitter`. In both cases, the original graphs are too large for effective visualisation and real cascade data is not available. In this section, we describe how these graphs were filtered to reduce their size and the simulations that were applied to create our experimental stimuli. Generally, social media graphs have high connectivity which causes high edge occlusion when visualised.

For our experiment, we started with two real world networks that were adapted

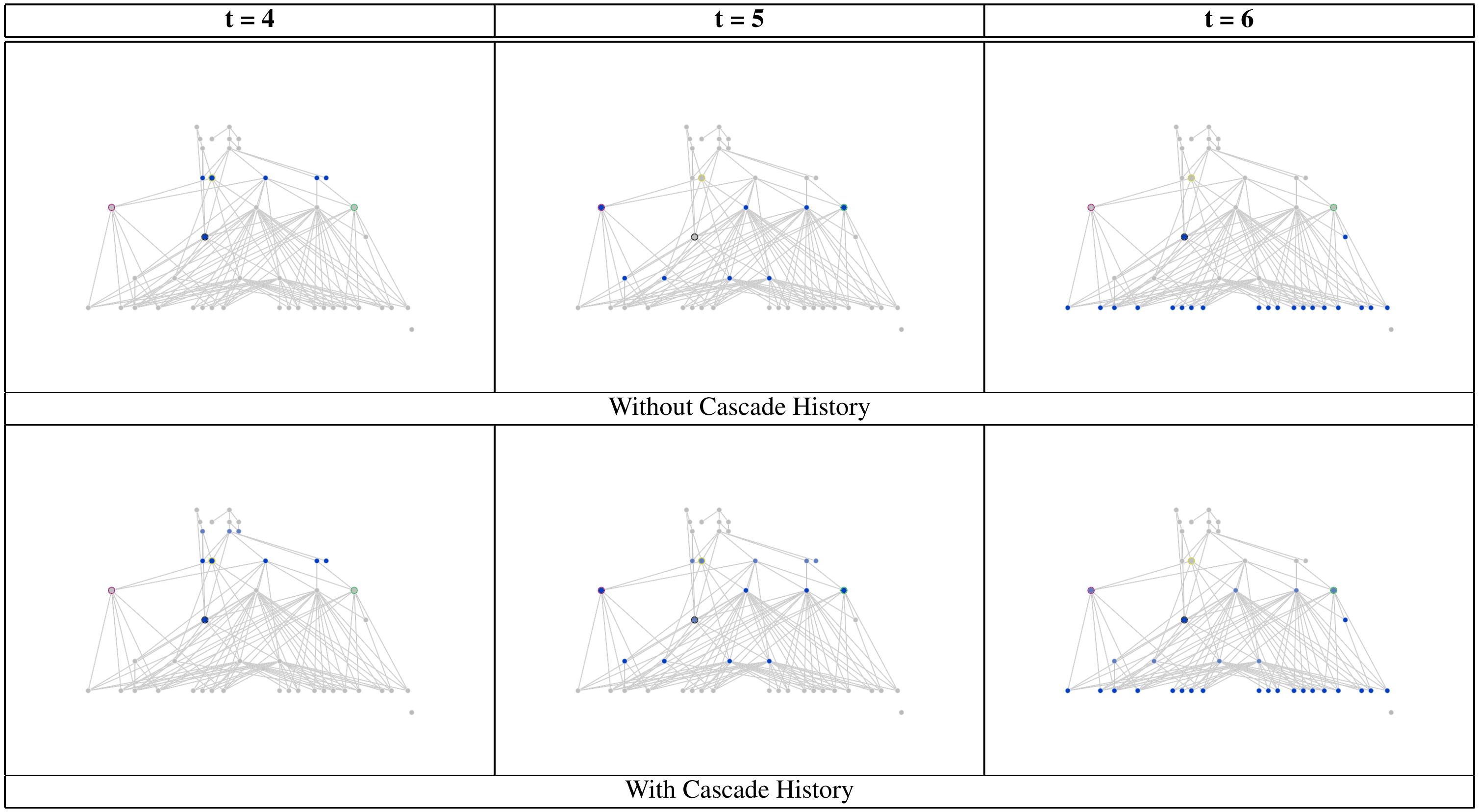


Figure 4: Cascade history factor shown with hierarchical layout. If cascade history is enabled, nodes infected in the previous timeslice appear in a light blue along with the infected nodes in the current timeslice in saturated blue. If cascade history is not enabled, only the nodes infected in the current timeslice are presented in a saturated blue. The same last three timeslices of a cascade instance of `Twitter` is drawn hierarchically with the two history conditions.

and filtered to ensure a feasible and appropriately controlled experiment. While these graphs are subsets of the data, it is preferred to base our experimental graphs on real world networks to ensure some external relevance. The alternative would be to create artificial networks that would have less external relevance.

3.4.1 Social Media Graphs

`Facebook` is an anonymised Facebook graph consisting of the public profiles of members of the New Orleans network [45]. Nodes are members and edges connect two members if they share a friendship. The full data set is an undirected graph of 63,731 nodes and 1,545,686 edges. We first reduced the size of the network by performing a breadth-first search from a central node in the data set (the node with identifier 1), of distance two, and took the resulting induced subgraph. This procedure reduced the network to 1,721 nodes and 37,722 edges for manageable cascade simulation. As Facebook graphs are undirected, to prepare this data set for cascade simulation, we inserted two directed edges (one in each direction) for every undirected edge.

`Twitter` is a Twitter follower network of British parliamentary MPs [25]. It consists of 418 nodes and 27,340 edges. The graph is directed, as it is a follower/followee network. In this network, a directed edge (A, B) indicates A follows the posts of B. As this means that B can influence A, we inverted the direction of all edges in this graph before cascade simulation.

3.4.2 Cascade Preparation

In order to create the cascades used in this experiment, we applied the commonly used Independent Cascade Model (ICM)[3] [20]. The model propagates cascades along directed edges according to probabilities associated with each of the individual edges of the graph. When a node is infected, a random number is selected and the edge prob-

[3] `http://php.scripts.psu.edu/hxc249/code_segments/independent_cascade.py`

abilities are used to determine which direct neighbours become infected. Seed nodes, where the infection begins in the first timeslice, need to be selected in this model.

For `Twitter`, we chose the leaders of the main political parties in the United Kingdom (David Cameron, Ed Miliband, and Nick Clegg). For `Facebook`, as node identities are anonymous, we took all nodes with relatively large degree (greater than 100) and selected three at random. These seeds were selected so that cascades of sufficiently large magnitude would spread through the graph. Choosing seed nodes with low degrees would cause the cascade to die out too quickly.

For our simulations, we set the probability of infection for all edges to a uniform value (2%) by following the results of Kempe *et al.* [31] which experimented with ICM probabilities between 1% and 10%. By visual inspection, we observed that the cascade activity at 1% was heavily dependant on low degree multipliers. A value of 2% produced visual stimuli with a balance between not enough and too much cascade activity. For values of 5% and above, the network was flooded. We produced sixteen different cascades for each data set, selecting ones that were not too easy or too hard for our tasks. This selection was primarily done by visual inspection using the animation presentation method. Stimuli were reconfirmed with small multiples.

Subsequently, each of the sixteen graphs produced by the cascades were further filtered. All nodes that are never infected in the six timeslices were removed from the graph. All edges that could not possibly be involved in an infection were also removed. This further filtering was necessary so as to make the graphs of a reasonable size for effective experimentation as suggested by pilot studies. As we are interested in the visualisation of cascades and the filtering process fully takes into account the propagation of the cascade on the graph, relevant graph characteristics for each cascade are preserved by the filtering process.

As there could be multiple sources of infection for each node, it is not the case that the number of neighbours a node infects corresponds its degree. High degree nodes could result from infections by multiple sources. A node u that is infected via multiple sources does not have these edges filtered out by this procedure. Thus, u can be of high degree and infect zero of its neighbours.

3.5 Task

In social networks, nodes that greatly increase the magnitude of a cascade are key [17]. These nodes often have high betweenness or degree centrality. Even though they have the potential to increase the spread of infection, for a particular cascade they may not. In this experiment, we are interested in those that actually increase the spread of infection.

- Which node **infects** the most of its direct **neighbours**?

The answer is one of four colours (green, gold, black, purple), as seen in Figures 1 and 2. These colours correspond to the outlines of four nodes in the cascade. The correct answer is the coloured node infected in a timeslice t that infects the greatest number of direct neighbours in timeslice $t+1$. In order to answer the question, participants first need to find the timeslice when one of the coloured nodes is infected. In animation, locating this timeslice is accomplished by using the slider. In small multiples, the participant searches for this timeslice. When the timeslice is located, the participant looks at the next timeslice to see how many direct neighbours are infected. The process needs to be repeated for all four coloured nodes to determine the colour that infects the most of its direct neighbours.

3.6 Experiment Design

We performed a within-participant experiment. Our primary research question focuses on whether there is any difference in performance in tasks involved in determining which nodes increase the rate of infection using animation or small multiples. Each of the secondary questions addresses an additional factor: force-directed vs hierarchical or no cascade history vs cascade history. To ensure generalisability of the results, all factors were applied to two data sets, giving us a total of 2 presentation methods (SM vs Anim) $\times$ 2 layouts (force-directed vs hierarchical) $\times$ 2 history (no cascade history vs cascade history) $\times$ 2 data sets (`Facebook` vs `Twitter`) = 16 unique stimuli. Each participant performed the task for each set of conditions and factors twice, giving a total of 32 stimuli recorded in the experiment. No time limit was enforced per task or for the experiment overall. However, a warning label appeared on the screen after thirty seconds, and participants were encouraged to finish the task quickly after that point (Figure 1).

Each stimulus was created using the procedure described in Section 3.4.2. As the cascade simulation depends on a random process, each stimulus is a different subgraph of the data sets. This variety in graph structures ensures greater generalisation of our results, as they do not simply apply to only one graph structure.

The experiment was divided into two, counterbalanced blocks, requiring participants to answer all questions under one presentation method first and all questions under the other one second. This counterbalancing ensured that any cognitive shift required for moving between two presentation methods, each requiring different interactions, occurred only once. Presentation method order was counterbalanced between participants with animation first to even-numbered participants and small multiples first to odd-numbered participants.

Within each presentation method block, a total of sixteen trials were recorded as data. The sixteen trials were randomised for each block individually and prefixed with four practice trials. The results of the practice trials were discarded and participants were not made aware that these trials did not form part of the experiment. These four trials presented each layout, cascade history, and data set of the experiment exactly twice. The inclusion of these practice trials, and the randomisation of the order of the trials assisted in countering any possible learning effect. Thus, for each presentation method, a total of 20 questions were asked (40 for the experiment including practice trials). These questions were presented in two blocks of ten. Between the two blocks of ten questions, participants could take a short, self-timed break.

At the beginning of each presentation method block, participants had a demonstration session. During this session, participants learned how to find the correct answer to the experimental task. They were encouraged to ask questions about the experimental task and about how to use the interfaces. Participants were made aware of two strategies for solving the task. The first involved looking at connections and the flow of the cascade. If the participant found the connections hard to read due to edge occlusion, they were advised to use cause and effect to see which nodes nearby lit up. Participants were also notified that if potentially two nodes could have infected a common neighbour, they were to assume that both nodes caused the infection.

Several experimental design decisions were made as a result of three pilot participants prior to the experiment. The experience of piloting revealed that four seconds was a suitable time to lapse before the animation starts (so as to give the participants time to visually explore the first timeslice and read the question). Also, about thirty seconds seemed to be an appropriate amount of time to complete each question. This soft time limit was applied to ensure that the experiment did not take too long and in an attempt to make the task cognitively challenging.

Both interfaces were rendered in real time using the Tulip framework [9]. Overall, twenty-one participants (16 male, 5 female) took part in the experiment. All were computer scientists with experience reading node-link diagrams. During the demonstration phase, all participants were tested to determine if they could see and distinguish the colours used in the experiment (all twenty-one passed). The total time for each experimental session was approximately one hour. Participants were drawn from students in computer science at Swansea University.

4 Results

We present the results of our experiment to test our primary and two secondary research questions. To answer our primary research question, we compared Animation (Anim.) to Small Multiples (SM) directly. Subsequently, as these two presentation methods are very different, we divided our data by presentation method. For each of these two presentation methods, we compared Force-Directed (FD) to Hierarchical (Hier.) layouts and No History (NH) to History (Hist.).

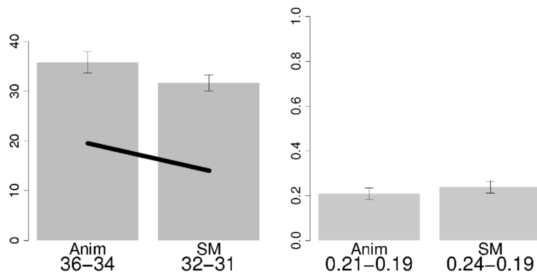
For each set of data analysed, a Shapiro-Wilk test, with a significance level of $\alpha = 0.05$, was used to determine whether or not the data was normally distributed. We found that at least one distribution in each test was not normally distributed. As a consequence, we used an exact Wilcoxon signed rank test when comparing two data sets at a significance level of $\alpha = 0.05$. When analysing the data separately according to presentation method, we applied a Bonferroni correction, thus reducing the significance level to $\alpha = 0.025$. In Figures 5 and 6, black lines connect pairs of bars with significant differences. Mean and median values, separated with a hyphen, are indicated below each bar. The standard error is indicated on each bar.

4.1 Animation vs. Small Multiples

Figure 5 shows the response time and error rates when comparing animation and small multiples. We found that small multiples resulted in significantly faster task performance than animation (Anim.:35.7s, SM:31.6s, $p = 0.019$). In terms of error rate, no significant difference was found.

4.2 Force-Directed vs Hierarchical

The first row of Figure 6 compares force-directed and hierarchical layouts for animation and small multiples separately.



(a) Response Time (sec.) (b) Error Rate

Figure 5: Response time and error rate when comparing animation to small multiples overall. Black lines connect pairs of bars with significant differences.

- For, **animation**:
  - Hierarchical resulted in significantly faster performance when compared to force-directed (FD:38.7s, Hier.:32.7s, $p = 0.003$).
  - Hierarchical had significantly fewer errors than force-directed (FD:0.26, Hier.:0.15, $p = 0.020$).
- For, **small multiples**:
  - Hierarchical resulted in significantly faster performance when compared to force-directed (FD:33.7s, Hier.:29.5s, $p = 0.012$).
  - No significant difference in terms of error rate.

4.3 History vs No History

The second row of Figure 6 compares no history and history factors for animation and small multiples separately.

- For, **animation** and **small multiples**:
  - No significant difference in either response time or error rate.

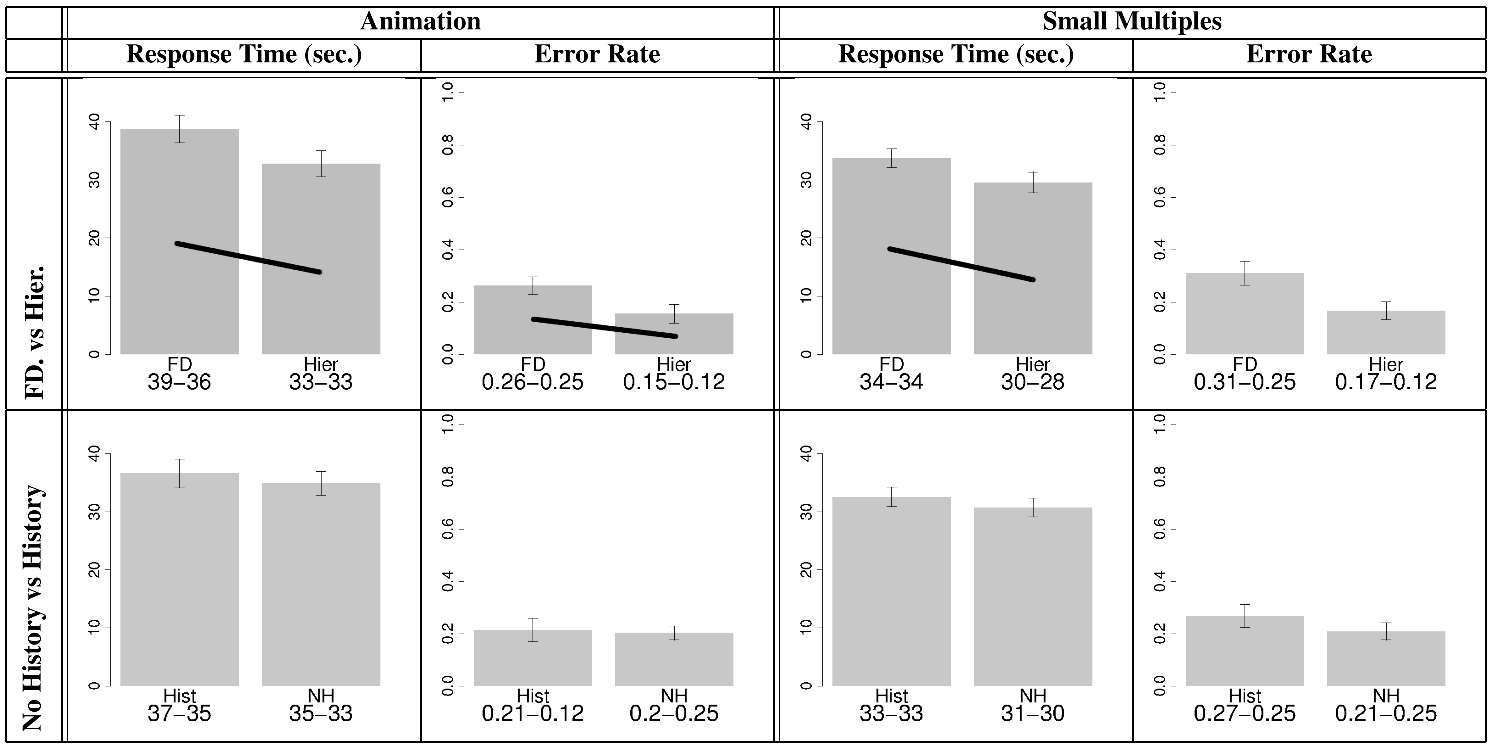


Figure 6: Response time and error rate for our secondary research questions. The first row of the table compares force-directed to hierarchical while the second row compares no history to history. In this table, **FD** is force-directed layout, **Hier** is hierarchical layout, **NH** is the no history factor, and **Hist.** is the history factor. Black lines connect pairs of bars with significant differences.

		Layout		History	
	All	**FD**	**Hier**	**NH**	**Hist.**
Animation	14	2	19	14	7
Small Multiples	7	3	18	15	6

Table 1: Preference data for the primary and secondary research questions of the experiment. In this table, **FD** is force-directed layout, **Hier** is hierarchical layout, **NH** is the no history factor, and **Hist.** is the history factor.

4.4 Survey Data

We asked participants to rate the presentation methods, layouts, and cascade history at the end of the experiment by asking them which presentation method, layout, and history methods they preferred. Table 1 records these results. Overall, animation was preferred to small multiples. According to the qualitative data, participants found interacting with the animation useful. For the animation condition, most participants manipulated the slider directly around the moment before and after answer nodes. No participants waited for the animation to start automatically and manipulated the visualisation immediately once they understood the task.

The participants preferred the hierarchical layout to the force-directed layout for both presentation methods. Finally, the participants preferred, to less of an extent, no cascade history to cascade history.

5 Discussion

For our primary research question, we found that small multiples resulted in significantly faster performance with no significant difference in terms of error rate. This result supports the argument that small multiples is more effective for the visualisation of cascades. It further confirms the design decisions of Barsky *et al.* [11]. The result is in line with Tversky *et al.* [43] who states that animations can take longer to fully understand so they have a cost. This result also agree with many studies [8, 21, 36] on various types of dynamic data.

When we compared force-directed to hierarchical layouts, we found that hierarchical layouts resulted in significantly faster performance. Hierarchical layouts also produced significantly fewer errors for animation. Therefore, this provides evidence that hierarchical layouts can improve performance. The study provides support for the design choices of Barsky *et al.* [11], independent of domain requirements. For our tasks, hierarchical layout was probably more suitable because the cascades are easier to perceive and analyse if they are shown to flow in a direction consistent with the layout. Thus, the spread of the cascade was easier to anticipate based on the structure of the graph layout.

We did not find a significant difference between the no history and history conditions in our experiment. The history condition presented the previous timeslice in a less saturated version of the infection colour when the no history condition did not represent history at all. Other colour schemes and methods for encoding cascade history should be considered and tested, particularly varying hue to represent history.

We found that animation was preferred to small multiples. The main objection to the small multiples approach was the smaller size of the graphs and many of the participants wanted zooming enabled. As only a sixth of the screen was used for each timeslice and no zooming was allowed for the interface, it is a clear disadvantage. A few participants also found that with animation actually showing the dynamic process that they felt more confident in their answers. In contrast, our quantitative data suggests that they were able to perform the tasks significantly faster with a small multiples representation with no significant difference with respect to error rate.

Participants preferred hierarchical to force-directed almost unanimously. This preference is consistent with the quantitative data recorded in the experiment. It seems that organising the layout to show consistent direction for the cascade was beneficial in determining which nodes would be infected next. Participants, in general, did not like the cascade history as implemented. Many found, especially during the animation condition, that it was hard to determine if the de-saturation of the node was a transition between timeslices or the presentation of history. It is clear that if history is to be shown, it should be done so less ambiguously.

6 Limitations, Conclusions, and Future Work

We have presented an experiment to test factors involved in visualising cascades. Our task was based on cascades in a social media context. We found that small multiples was significantly faster than animation for this task. When divided by presentation method, for animation, a hierarchical drawing of the graph was significantly faster and produced significantly fewer errors. For small multiples, a hierarchical drawing was significantly faster. We therefore conclude that effective visualisation of cascades through graphs is best supported using small multiples and a hierarchical layout.

There are several implications for our work to the problems in biology and social media as described in the introduction. Firstly, both design choices, small multiples and hierarchical layout, of Barsky *et al.* [11] were validated formally. In the original system design, small multiples was chosen based on results in related literature–works such as Tversky *et al.* [43]. The developed techniques were not empirically validated to test their effectiveness with humans. The hierarchical drawing method chosen was chosen because it was natural for biologists. As long as a large portion of the process flows downwards in the diagram, the results of this experiment further support these arguments from a perceptual sense. In terms of social media graphs, neither small multiples nor hierarchical layouts are all that prevalent. Usually, force-directed are used to draw the diagrams. Our study suggests that when a cascade flows along the direction of graph edges, a hierarchical layout of the graph can help.

Empirical research on cascade visualisation is at a very early stage, and future experiments would produce results that extend beyond the scope of this first experimental study. We have focused on node link diagrams, the presentation methods of animation and small multiples, and highlighting only the nodes of the data set using colour. Our reason for testing these particular encodings was due to the fact that they had been used in previous work [11]. Through our experimental design, we were interested in validating whether or not these design choices were effective in terms of human performance. Other visual encodings are available and could be used in the experiment for depicting cascade propagation: integrated visualisations showing the time dimension in a single view [46], possible matrix representations [14], and encoding the propagation edges [27]. In a node-link context, drawing algorithms besides force-directed and hierarchical could have been considered. Broader graph reading tasks could be investigated, in particular, those inspired by applications (e.g. social media, simulations of epidemic spread of diseases, geospatial information visualisation, and expression levels in biological networks). While we chose the commonly used ICM cascade simulation model, other simulation models could be used and follow-up studies would usefully validate the results presented here.

Scalability, in terms of the number of timeslices and nodes in the dynamic graph, is an important factor in this research area. The number of nodes, edges, and timeslices in the data sets used in this experiment are small when compared to existing data sets. However, most scalable methods for dynamic graph visualisation either use filtering or aggregation to reduce the number of time steps and/or the number of nodes and edges to a reasonable level. In these cases, the results of this experiment can be applied directly as the information visualisation technique has reduced the large data set to one of this size. It would be interesting to further investigate techniques that are able to visualise large dynamic graphs directly without the need of aggregation or filtering, but such techniques remain future work.

We did not find an effect of the history condition in our experiment. In the experiment, we used saturation level to encode this history. Different encodings similar to those used for visualising graph differencing [1, 2, 7, 49] could provide more of a benefit. By considering other encodings, besides colour, visualising history could have more of an impact on cascade visualisation tasks. Testing these other possible encodings remains future work.

This paper reports results of the first empirical study of the effective visualisation of cascade information on node-link diagrams. We also raised further questions regarding the appropriate means of representing cascade history and the applicability of these results to other real-world application domains.

Acknowledgments

We thank Donn Morrison of NUI Galway (DERI) for his advice on cascade simulation and social media analysis. We also thank Derek Greene for providing the `Twitter` data set and processing advice.

References

[1] K. Andrews, M. Wohlfahrt, and G. Wurzinger. Visual graph comparison. In *International Conf. on Information Visualisation (IV '09)*, pages 62–67, 2009.

[2] D. Archambault. Structural differences between two graphs through hierarchies. In *Proc. of Graphics Interface (GI '09)*, pages 87–94, 2009.

[3] D. Archambault, J. Abello, J. Kennedy, S. Kobourov, K.-L. Ma, S. Miksch, C. Muelder, and A. C. Telea. *Temporal Multivariate Networks*, volume 8380 of *LNCS*, chapter 8, pages 151–174. Springer, 2014.

[4] D. Archambault and H. C. Purchase. The mental map and memorability in dynamic graphs. In *Proc. of the IEEE Pacific Visualization Symposium (PacificVis '12)*, pages 89–96, 2012.

[5] D. Archambault and H. C. Purchase. Mental map preservation helps user orientation in dynamic graphs. In *Proc. of Graph Drawing (GD '12)*, pages 475–486, 2012.

[6] D. Archambault and H. C. Purchase. The "map" in the mental map: Experimental results in dynamic graph drawing. *Int. J. of Human-Computer Studies*, 71(11):1044–1055, 2013.

[7] D. Archambault, H. C. Purchase, and B. Pinaud. Difference map readability for dynamic graphs. In *Proc. of Graph Drawing (GD '10)*, volume 6502 of *LNCS*, pages 50–61, 2010.

[8] D. Archambault, H. C. Purchase, and B. Pinaud. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Trans. on Visualization and Computer Graphics*, 17(4):539–552, 2011.

[9] D. Auber. Tulip : A huge graph visualization framework. In P. Mutzel and M. Jünger, editors, *Graph Drawing Software*, Mathematics and Visualization, pages 105–126. 2003.

[10] B. Bach, J.-D. Fekete, and E. Pietriga. GraphDiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):740–754, 2014.

[11] A. Barsky, T. Munzner, J. Gardy, and R. Kincaid. Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE Trans. on Visualization and Computer Graphics (Proc. InfoVis '08)*, 14(6):1253–1260, 2008.

[12] B. B. Bederson and A. Boltman. Does animation help users build mental maps of spatial information? In *Proc. of the IEEE Symposium on Information Visualization*, pages 28–35, 1999.

[13] S. P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.

[14] U. Brandes and B. Nick. Asymmetric relations in longitudinal social networks. *IEEE Trans. on Visualization and Computer Graphics (Proc. InfoVis '11)*, 17(12):2283–2290, 2011.

[15] U. Brandes and T. Willhalm. Visualization of bibliographic networks with a reshaped landscape metaphor. In *Proc. of the symposium on Data Visualisation (VISSYM '02)*, 2002.

[16] M. Burch, N. Konevtsova, J. Heinrich, M. Hoeferlin, and D. Weiskopf. Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *IEEE Trans. on Visualization and Computer Graphics (Proc. InfoVis '11)*, 17(12):2440–2448, 2011.

[17] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 199–208, 2009.

[18] F. Chevalier, P. Dragicevic, A. Bezerianos, and J.-D. Fekete. Using text animated transitions to support navigation in document histories. In *Proc. of SIGCHI conference on Human Factors in Computing Systems (CHI '10)*, pages 683–692, 2010.

[19] M. Chimani, C. Gutwenger, M. Jünger, K. Klein, P. Mutzel, and M. Schulz. The open graph drawing framework. In *Proc. of Graph Drawing (GD '07)*, 2007.

[20] E. T. D. Kempe, J. Kleinberg. Influential nodes in a diffusion model for social networks. *Automata, Languages and Programming*, 3580:1127–1138, 2005.

[21] M. Farrugia and A. Quigley. Effective temporal graph layout: A comparative study of animation versus static display methods. *J. of Information Visualization*, 10(1):47–64, 2011.

[22] A. Frick, A. Ludwig, and H. Mehldau. A fast adaptive layout algorithm for undirected graphs. In *Proc. Graph Drawing (GD'94)*, volume 894 of *LNCS*, pages 388–403, 1995.

[23] E. R. Gansner, E. Koutsofios, S. C. North, and K.-P. Vo. A technique for drawing directed graphs. *IEEE Trans. on Software Engineering*, 19(3):214–230, 1993.

[24] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *ACM Trans. Knowledge Discovery from Data*, 5(4):21:1–21:37, 2012.

[25] D. Greene and P. Cunningham. Producing a unified graph representation from multiple social network views. In *Proc. ACM Web Science 2013 (WebSci'13)*, pages 118–121, 2013.

[26] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Trans. on Visualization and Computer Graphics (Proc. InfoVis '07)*, 13(6):1240–1247, 2007.

[27] D. Holten, P. Isenberg, J. J. van Wijk, and J.-D. Fekete. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In *Proc. of the Pacific Visualization Symposium (PacificVis '11)*, pages 195–202, 2011.

[28] Y. A. Ivanov, C. R. Wren, A. Sorokin, and I. Kaur. Visualizing the history of living spaces. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1153–1160, 2007.

[29] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proc. Joint 9th WEBKDD and 1st SNA-KDD Workshop*, pages 56–65, 2007.

[30] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):927–934, Nov 2010.

[31] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

[32] A. Kerren, H. C. Purchase, and M. O. Ward, editors. *Multivariate Network Visualization*, volume 8380 of *LNCS*. Springer, 2014.

[33] H. Koike. The role of another spatial dimension in software visualization. *ACM Transactions on Information Systems*, 11(3):266–286, 1993.

[34] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proc. 19th international conference on World Wide Web*, pages 591–600, 2010.

[35] H. C. Purchase and A. Samra. Extremes are better: Investigating mental map preservation in dynamic graphs. In *Diagrams '08*, volume 5223 of *LNAI*, pages 60–73, 2008.

[36] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Trans. on Visualization and Computer Graphics (Proc. InfoVis '08)*, 14(6):1325–1332, 2008.

[37] S. Rufiange and M. J. McGuffin. DiffAni: Visualizing dynamic graphs with a hybrid of difference maps and animation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2556–2565, 2013.

[38] M. Shanmugasundaram, P. Irani, and C. Gutwin. Can smooth view transitions facilitate perceptual constancy in node-link diagrams? In *Proc. of Graphics Interface (GI '07)*, pages 71–78, 2007.

[39] K. Sugiyama and K. Misue. Graph drawing by the magnetic spring model. *Journal of Visual Languages and Computing*, 6(3):217–231, 1995.

[40] M. Tory, D. Sprague, F. Wu, W. Y. So, and T. Munzner. Spatialization design: comparing points and landscapes. *IEEE Trans. on Visualization and Computer Graphics (Proc. InfoVis '07)*, 13(6), 2007.

[41] M. Tory, C. Swindells, and R. Dreezer. Comparing dot and landscape spatializations for visual memory differences. *IEEE Trans. on Visualization and Computer Graphics (Proc. InfoVis '09)*, 16(6):1033–1040, 2009.

[42] E. Tufte. *Envisioning Information*. Graphics Press, 1990.

[43] B. Tversky, J. Morrison, and M. Betrancourt. Animation: Can it facilitate? *Int. Journal of Human-Computer Studies*, 57:247–262, 2002.

[44] F. Viégas, M. Wattenberg, J. Hebert, G. Borggaard, A. Cichowlas, J. Feinberg, J. Orwant, and C. Wren. Google+ripples: A native visualization of information flow. In *Proc. of the International Conference on World Wide Web*, WWW '13, pages 1389–1398, 2013.

[45] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in facebook. In *Proc. of ACM SIGCOMM Workshop on Social Networks (WOSN'09)*, 2009.

[46] T. von Landesberger, S. Diel, S. Bremm, and D. W. Fellner. Visual analysis of contagion in networks. *Journal of Information Visualization*, 2013.

[47] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. In *Proc. of the IEEE International Conference on Data Mining (ICDM '10)*, pages 599–608, 2010.

[48] J.-S. Yi, N. Elmqvist, and S. Lee. TimeMatrix: Visualizing temporal social networks using interactive matrix-based visualizations. *Human-Computer Interaction*, 26(11-12):1031–1051, 2010.

[49] L. Zaman, A. Kalra, and W. Stuerzlinger. The effect of animation, dual view, difference layers, and relative re-layout in hierarchical diagram differencing. In *Proc. of Graphics Interface (GI '11)*, pages 183–190, 2011.