Time Sequences

Ross Shannon

Systems Research Group

School of Computer Science &

Informatics

UCD Dublin

Ireland

ross.shannon@ucd.ie

Aaron Quigley

Systems Research Group

School of Computer Science &

Informatics

UCD Dublin

Ireland

aaron.quigley@ucd.ie

Paddy Nixon

Systems Research Group

School of Computer Science &

Informatics

UCD Dublin

Ireland

paddy.nixon@ucd.ie

Abstract

Visualisations of dynamic data change in appearance over time, reflecting changes in the underlying data, be that the development of a social network, or the addition or removal of a device node in an ad-hoc communications network. As viewers of these visualisation tools, it is up to us to accurately perceive and keep up with the constantly shifting view, mentally noting as visual elements are added, removed, changed and rearranged, sometimes at great pace. In a complex data set with a lot happening, this can be a strain on the observer's comprehension, with changes in layout and visual population disrupting their internalised "mental model" of the data, leading to errors in perception. We present Time Sequences, a novel dual visualisation technique which dilates the flow of time in the visualisation so that observers are given proportionally more time to understand changes based on the density of activity in the visualisation.

Keywords

Visualization, visual analytics, human factors, perception, dynamic data.

ACM Classification Keywords

H.5.0 Information Interfaces and Presentation: General,

H.1.2 Human information processing.

Introduction

Many data sources are *dynamic*: they change over time. Examples abound, from the evolution of a person's social network to the transitory device populations of ad-hoc networks. These type of data present the visualisation designer with options: they could take a snapshot of the data as it existed at a certain time, generate a static view from it and perform a visual analysis of its structure and layout. This lets a user analyse aspects of the data, but doesn't allow them to see the data evolve over time.

Further understanding can be facilitated by continually visualising the data at successive moments in its evolution, and transitioning smoothly between instants using animation. The view an observer sees at any particular time step is thus only a view of the system for that instant; the view may be modified or entirely replaced as the visualisation progresses. This kind of visualisation presents a different variety of information about the data: at what rate and regularity do changes occur, how does the network under study respond to changes in topology, and where do localised changes occur within the overall structure?

For instance, an ad-hoc network of wireless mobile devices will evolve over time as new devices come into range and join the network. Connections between the devices will also be transient. Thus, a visualisation of this data set will show nodes entering and leaving, edges being instantiated and deleted, and so on, all occurring at irregular intervals on different timescales. This research was motivated by difficulties users had understanding the visualisations in our previous work using traditional techniques in this area [9]. Similarly a visualisation of a person's social network will grow in complexity as time goes on, when further nodes join the network and rich interconnectivity emerges.

In this article, we are concerned with these volumes of data that are changing over time. Information visualisation is designed to exploit the bandwidth of the human visual system and avoid excess cognitive processing. However, in highly-dynamic data where changes to the display are frequent and unpredictable, the observer is tasked with maintaining their understanding of the view that they're watching no matter the complexity of the data.

Here, we present *Time Sequences*, an approach to harnessing the user's ability to understand change when presented at an accommodating pace. We enable this by presenting the flow of time within the visualisation in a non-linear way. This technique gives the human observer more time to absorb changes into their understanding of the model, while also giving the visualisation algorithm more time to manage and represent these changes on-screen.

Dynamic Information Visualisation

Sources of dynamic data exist in many fields, and depending on the speed with which the data in question changes and is recorded, have a range of tempos and rhythms. In economics, the noise-laden fluctuations of the stock market generate huge amounts of data every day and are frequently analysed in real-time to support critical quick decisions. In contrast, the rapidly-changing metabolic networks that are analysed in bio-informatics are more suitable for deeper analysis after the fact once enough data has been recorded. On the far end of this scale, the evolution of a social network happens at a comparatively slow speed, but can be
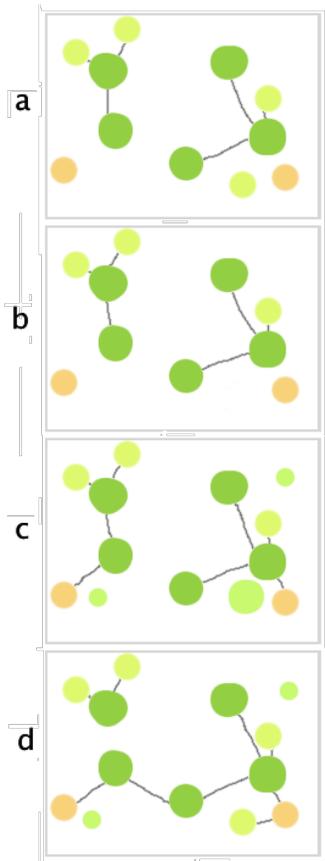
Figure 1. A network undergoing a sequence of changes to its initial node population and topology, (a). First, a node is removed (b); then, two new edges are added, along with a number of other nodes (c). Finally, one subgraph becomes disconnected from the rest (d).

sped up to be visualised and analysed at a pace more befitting visual inspection [3].

In this article, we focus on graph visualisations, which can range from the sociograms of social science to the visualisation of telecommunications networks large and small. Graph drawings can be judged over a number of aesthetic criteria—such as the number of visible edge crossings, or the aspect ratio of the diagram—which have an effect on a user's understanding of the structure and features of the graph [8].

Dynamic graph drawing is a separate branch of research from work on the traditional static plots, and introduces techniques to redraw the graph after a change has occurred [1]. In these types of dynamic networks, new information is added, while old information is refined or replaced. When these changes occur, it is important to minimise disruption as much as possible [4], to give the observer the best opportunity to understand the changes to the graph. The changes that can occur in a graph include:

- nodes being added or removed
- edges being added or removed
- changes in the weight of either nodes or edges
- changes in the explicit clustering of nodes

Some of these change events can be seen in the succession of panels in Figure 1. From these images, you may be able to see that it can be difficult to identify where exactly the graph has changed and in how many places, as well as knowing the full extent of the change. Bear in mind that you have the benefit of being able to compare the views of the graph side-by-

side to help identify differences (what Tufte calls a *small multiples* comparison [11]), as well as an explanatory caption to confirm your observations. In an animation of a dynamic graph with nodes appearing and disappearing all the time without warning, these kind of comparisons are not possible, and the observer is forced to compare the views in time, rather than in space; leaving them to rely on their memory and observational skills to discern the differences from frame to frame rather than purely leveraging their visual perception.

The user's mental model
A visualisation in flux creates a host of problems for the viewer. While analysing any graph visualisation, dynamic or not, a user builds up what's called a *mental model* [2]. Visible changes to the graph are incorporated by the user into their own mental model as they are perceived, and many techniques have been established for minimising the visual disruption caused by a change to the graph.

When the mental model *is* disrupted, the user may struggle to reconcile a change with their understanding of the structure of the data. This will cause them to lose their built-up understanding of the data, in the worst case forcing them to have to begin again with learning the structure of the network, all while new modifications continue to be made.

Multiple fleeting changes can effectively co-occur in the visualisation, in different parts of the screen, testing the user's perceptions and running the risk of them missing vital changes. The irregularity of changes also creates difficulty for the user in anticipating *when* events they need to be aware of are likely to occur.

Change blindness

Even when the user is paying attention to the view, they may not be aware of everything that is happening, due to a perceptual weakness known as change blindness [7]. When faced with a constant flow of information, a user may be too focused on one area of the view, not recognise what they are seeing or miss visual changes entirely. Studies in cognitive science have uncovered minor or even major changes going unnoticed by observers [10].

The writer Alan Moore has argued that one of the benefits that both novels and comic books have over films is they can be taken at their own pace rather than the pace mandated by the flow of the film [6]. In these media, the reader can slow down to fully take in a complex part of the book, and speed up during passages of little consequence.

In a visualisation observer's case, this "pace'' is the rate that the data was recorded at. However, since we engage with dynamic visualisations in a different way to the passivity of watching a film, we have the opportunity to make understanding the passage of a visualisation easier, by enacting some control over the speed that it is presented at.

Event Sequences

The techniques described so far to help the viewer make sense of dynamic graphs have involved visual changes to the graph and intelligent algorithms to lessen visual churn on the screen, but keep the data flowing in at the rate that it is recorded. We suggest that it would be useful to lessen the cognitive load on a user in a second way; by affording them additional time in which to make sense of the changes to the graph,

and breaking up the constant stream of modifications into discrete blocks of changes.

When developing a dynamic visualisation, the developer will typically load in a log file of changes, or mutations, which occurred to the data structure. Here we'll use a file of ordered mutations, with one per line, describing the generation of a graph. Each line takes the form of a timestamp and the corresponding mutation:

```
010704120856-0700 add node1
010704120858-0700 add node2
010704120867-0700 link node1 node2
⋮
```

We refer to each of these lines as *events* in the system, and the queue of events in the input file as an *event sequence*. The visualisation will render the data structure as it stands, and then loop to continually read in the next few events for that time step. This might be a single event, many events all happening in a short space of time, or nothing for that time step, depending on the data. The visualisation will continue until it is quit or it reaches the end of the file.

Throughout these changes, the user is at the mercy of the data: the visualisation will continue to render new events at the rate and rhythm that is defined in the input file, leaving the user to keep up with this pace or miss potentially important information.

This imposition is not ideal for a number of reasons. First of all, events can occur at a volatile rhythm, which may be rapid if events are clustered together in the log file. This means the visualisation will be updated

frequently, without regard for the difficulty the user is having at incorporating previous events into their mental model. To take an example from life, the reader may have themselves experienced a seminar speaker proceeding through material at too quick a pace, and leaving the attendees wishing they would slow down their delivery so that they could properly note down the important concepts without fear of missing something.

Conversely, there may be long passages of time when nothing of much consequence is happening, wasting the observer's time by giving them nothing to analyse.

Perceiving Time and Space

Humans have an internal sense of time passing, but it's well known to be inaccurate and susceptible to significant influence from many external factors, including visual stimulus, caffeine intake and so on [5]. Readers will identify the feeling of time passing quickly or slowly depending on their engagement with a task, whether they find it engrossing or tedious.

On the other hand, our visual perception is highly adept at accurate local comparisons (see Figure 2). When presented with two parallel lines, observers can instantly see which is the longer of the two. We seek to invert the balance here: in the application of our Time Sequences technique, we take time out of the equation by only allowing visual data changes to occur at certain regular, predictable time intervals. By marshaling events in this way, the user always knows that they have a certain constant amount of time to understand a particular scenario before the next change arrives. It also gives the visualisation algorithm more time than it might usually receive to optimally lay out the graph for legibility in between changes. This helps it avoid visual

oscillation, harsh transitions and the ensuing damage to the user's mental model.

Visualising Time

Though events now occur at a constant rate for the user, they will not have occurred regularly in the event sequence, and these relative delays are important information for the user's understanding of the structure. The user needs to be informed of how long has passed since the last event. To do this, we introduce a set of visual elements that show the speed of the passage of time between each event, so that the user gets a sense of how long has passed in "real-time" since the previous event.

First, we introduce a visual timeline which flows down the left side of the view, as seen in Figure 3. Each segment of this timeline is sized proportionally to the period of time it represents. Traditional timelines are presented horizontally, but when presented vertically like this the user can easily compare the lengths of two segments (this is another application of the small multiples technique). The first segment at the top of the screen represents the unit length of a period in the simulation (this is currently set to two seconds), so the observer has a frame of reference for comparison.

Besides encoding the relative lengths of time between events, the timeline acts as a visual history of the data stream. The user can visually inspect the timeline to see if the events are happening close together (many short lines) or far apart (long lines), and at a regular or chaotic rhythm, but they don't have to deal with any chaos or long waits themselves.

This non-linear time series technique can be applied to
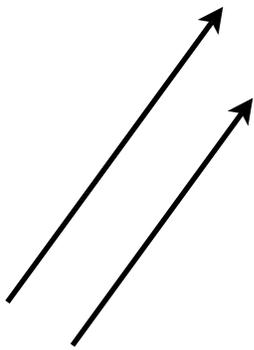
Figure 2. The human eye's ability to discern length is much better than the brain's ability to estimate time.

Figure 3. A mockup of the visualization of a dynamic graph using the event sequences model. The vertical timeline tracks the time between mutations of the data structure shown on the right (from [9]). As each change occurs, the timeline scrolls down, so that the earliest event moves downward beyond view, and a new event marker is added to the top. This is so that the marker's length can be compared to the unit length at the top (which designates a two second duration of "real time").
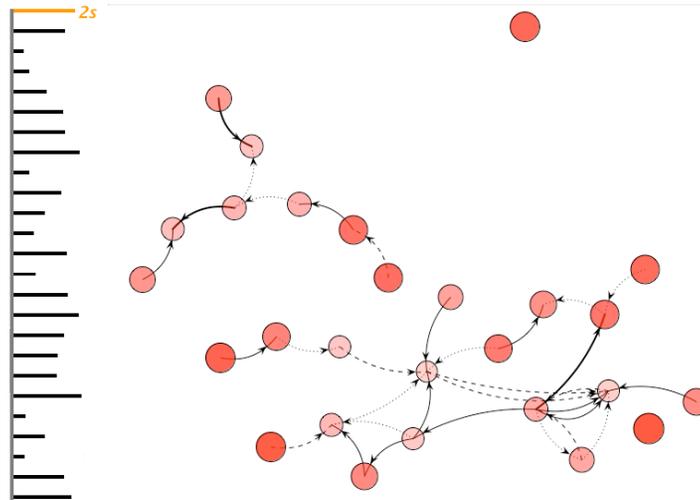
any dynamic data source, whether for offline analysis of a log file, or for analysis of data being streamed directly from a data source. The technique is applicable as long as time-critical responses are not required of the observer. Events will be buffered into a single continuous feed of changes, and will take longer in real-time to go through if there are many events bunched close together. By the same token, the viewer can watch more data then they could in real-time if the event queue is sparse. This acts in a give-and-take fashion: when events occur frequently, the visualisation expands this time and takes longer to get through these events while buffering future events. Then when nothing much is happening in the log file for a particular period, it can "catch up" to more contemporary events.

Conclusion

Historically, most graph visualisation research has been task-oriented. Here, we are exploring methods to improve the user's experience of understanding and interacting with the graph. Time Sequences helps the user by giving them a continuum of events where changes to the data structure occur deterministically on

a set rhythm, and quiet periods where no events are occurring can be quickly skipped past. A timeline tracks the relative differences in time between events, providing a visual history of the data set.

References

[1] Branke, J. Dynamic graph drawing. In *Drawing graphs: methods and models*, Springer-Verlag (2001) pp. 228–246.

[2] Eades, P., Lai, W., Misue, K., & Sugiyama, K. Preserving the mental map of a diagram. In *Proc. Compugraphics, 91*(9), (1991) pp. 24–33.

[3] Farrugia M. and Quigley A. Visual data exploration of temporal graph data, In *Proc. VDA 2009*. (2009).

[4] Friedrich, C., & Eades, P. Graph Drawing in Motion. In *Journal of Graph Algorithms and Applications, 6*(3), (2002) pp. 353–370.

[5] Harrison, C., Amento, B., Kuznetsov, S., & Bell, R. Rethinking the progress bar. In *Proc. UIST 2007*, ACM Press. (2007) pp. 115–118.

[6] Moore, A. Writing for comics. Avatar Press. (2007)

[7] Nowell, L., Hetzler, E., & Tanasse, T. Change Blindness in Information Visualization: A Case Study. In *Proc. Infovis 2001,* IEEE (2001) pp. 15–23.

[8] Purchase, H. C. (1998). Which Aesthetic Has the Greatest Effect on Human Understanding? In *Graph Drawing, 1994*, Springer (1994) pp. 248-261.

[9] Shannon, R., Williamson, G., Quigley, A., & Nixon, P. Visualising Network Communications to Evaluate a Data Dissemination Method for Ubiquitous Systems. In *Proc. UbiComp 2007 Workshop Proceedings* (2007) pp. 288–291.

[10] Simons, D. J., & Chabris, C. Gorillas in our midst. In *Perception, 28*, (1999) pp. 1059–1074.

[11] Tufte, E. R. Envisioning Information. Graphics Press. (1990) p. 67.