SCHOOL OF INFORMATION TECHNOLOGIES

ON THE FAITHFULNESS OF GRAPH VISUALISATIONS

TECHNICAL REPORT 690

QUAN NGUYEN, PETER EADES, SEOK-HEE HONG

AUGUST, 2012

On the Faithfulness of Graph Visualizations

Quan Nguyen, Peter Eades, Seok-Hee Hong

School of Information Technologies, University of Sydney, Australia
{qnguyen,peter,shhong}@it.usyd.edu.au

Abstract. Readability criteria have been commonly used to measure the quality of graph visualizations. In this paper we argue that readability criteria, while necessary, are not sufficient. We propose a new kind of criterion, generically termed *faithfulness*, for evaluating graph layout methods. We propose a general model for quantifying faithfulness, and contrast it with the well established readability criteria. We use examples of multidimensional scaling and edge bundling to illustrate faithfulness.

1 Introduction

Graphs become larger and more complex. Graph visualization techniques, to cope with this, are getting more sophisticated and involving complex parameter settings to turn graphs into drawings. Thus, the challenge is to justify how reliable visualizations methods and models are.

Graph drawing algorithms developed over the past 30 years aim to produce "readable" pictures of graphs. Here "readability" is measured by *aesthetic criteria*, such as:

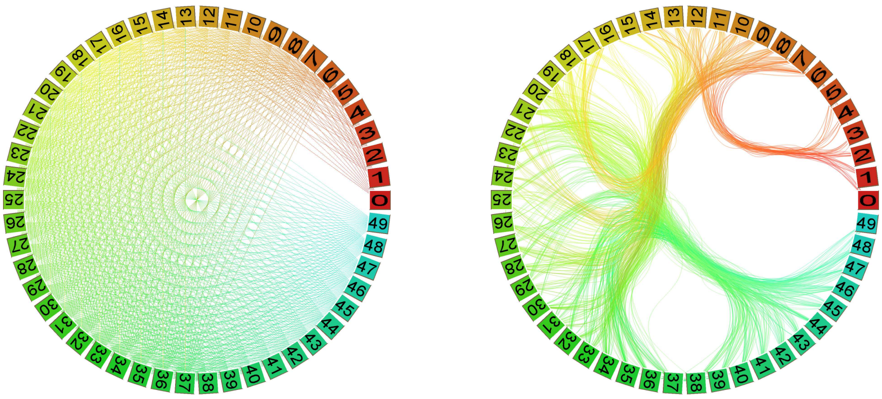
- *Crossings*: the picture should have few edge crossings,
- *Bends*: the picture should have few edge bends,
- *Area*: the area of a grid drawing should be small.

Algorithms that attempt to optimise aesthetic criteria have been successfully embedded in systems for analysis in a wide variety of domains, from the finance industry to biotechnology.

In this paper, we argue that readability criteria for visualizing graphs, though necessary, are not sufficient for effective graph visualization. We introduce another kind of criterion, generically called "faithfulness", that we believe is necessary in addition to readability. Intuitively, a graph drawing algorithm is "faithful" if it maps different graphs to distinct drawings[1]. In other words, a faithful graph drawing algorithm never maps distinct graphs to the same drawing.

Faithfulness criteria are especially relevant for modern methods that handle very large and complex graphs. Information overload from very large data sets means that the user can get lost in irrelevant detail, and methods have been developed to increase readability by decreasing detail in the picture.

[1] In Mathematics, a *faithful representation* of a group on a vector space is a linear representation in which different elements in the group are represented by distinct linear mappings.



(a) Unbundled (b) Bundled

Fig. 1: An example graph using force-directed edge bundling

As an example, *edge bundling* and its variants [15, 24, 30, 31, 40, 44, 50, 55, 63] simplify edge connections in the picture to increase readability with respect to some tasks. Figs. 1 shows two pictures of the same graph; Fig. 1a is a simple circular layout with straight-line edges and Fig. 1b has the same node positions but with "bundled" edges.

However, bundling often sacrifices faithfulness, in that two different graphs can be mapped to the same picture. For example, Fig. 2a shows a graph that differs from the graph Fig. 1a by almost 10 percent of the total number of edges. This graph is bundled in Fig. 2b. The bundled representations of the two different graphs (Figs. 1b and 2b) are identical.

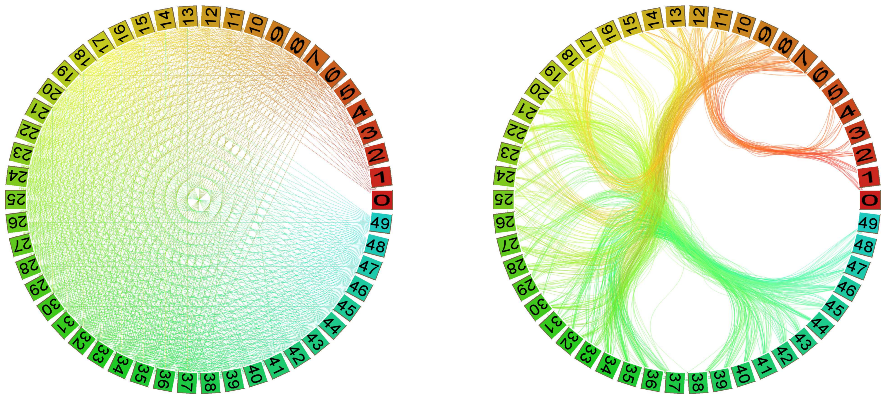
There are several notions along the lines of faithfulness in scientific visualization; these include *fidelity* of the picture [46] and visual *reconstructability* for flow visualization [37]. However, there is no model of faithfulness for graph visualization.

In order to describe the faithfulness concept, we present a general model of graph visualization process in Section 3. A general model the faithfulness of a graph visualization method is described in Section 4. Here we divide the general concept into three kinds of faithfulness: information faithfulness, task faithfulness, and change faithfulness. A model for quantifying faithfulness is in Section 6. We illustrate faithfulness with two examples in Sections 7 and 8: multidimensional scaling and edge bundling. Section 11 concludes the paper with a remark about the failure of 3D graph drawing to make industrial impact.

2 Related work

2.1 Evaluation of Visualization

Recent years have witnessed an increase of interests in evaluative research methodologies and empirical work [13, 14, 42, 53, 59].



(a) Unbundled (b) Bundled

Fig. 2: A 10 percent modification of the example graph in Figure 1(a) and the result using force-directed edge bundling

In particular, several quality metrics have been recently proposed for evaluating high-dimensional data visualization [4]. A survey of quality concerns for parallel coordinates is given in [16]. There is also proposal for judging visualizations regarding the presence of difficulties and distracting elements (or so-called "chartjunk") [35]. Other research focuses on narrative visualization and the effects on interpretations with respect to the intended story [36].

The previous research on visualization evaluation has focused much on information visualization in a broad sense, while our research has mainly targeted to the quality of graph visualizations.

2.2 Readability in Graph Visualization

Graph drawing algorithms in the past 30 years typically take into account one or more *aesthetic criteria* to aim to increase the *readability* of the drawing and to achieve "nice" drawings.

There are a wide range of aesthetic criteria proposed for graph visualizations include, for example, (a) minimizing the number of edge crossings, (b) minimizing the number of bends, (c) increasing orthogonality, (d) increasing node distribution, (e) minimizing the total area used, (f) maximizing the symmetries in the underlying network structure.

Graph drawing algorithms can be classified, for example, in one the following criteria. First, one popular technique is force-directed layout, which uses physical analogies to achieve an aesthetically pleasing drawing [3, 5, 39]. Several works extended force-directed algorithms for drawing large graphs in [27, 60]. Second, multidimensional scaling is another popular method for graph visualization and visual mining [10, 12, 38, 49]. Third, many other graph drawing algorithms try to take advantage of any knowledge on topology (such as planarity or SPQR decomposition) to optimize the drawing in terms of readability. Fourth, other approaches offer representations composed of visual abstractions of clusters to improve readability.

Much research has focused on the computational efficiency while achieving the drawing readability, to the extent to which the resulting drawings conform to the aimed aesthetic criteria. In general, the problem of simultaneously optimizing two or more criteria, in many cases, is NP-hard.

In this paper, we propose another criterion, namely faithfulness, to the quality of graph visualizations. The faithfulness is different from readability criteria in the literature.

2.3 Mental map preservation in Graph Visualization

An important criterion for dynamic graph drawings is *mental map preservation* [19] or *stability* [52]. This is one of the most important concepts for visualizing dynamic graphs.

The mental map concept refers to the presentation of a person's mind while exploring visual information. The better the mental map is preserved, the easier the structure change of a graph is understood. The user's familiarity with the old drawing will transfer to the new drawing with minimal effort.

There are a huge amount of work in mental map preservation.

Some *layout adjustment* algorithms use a notion of proximity to preserve the mental map and rearrange a drawing in order to improve some aesthetic criteria. These algorithms include, for example, incremental drawing of directed acyclic graphs [51] or computing the layout of a sequence of graphs offline [17] or using different adjustment strategies in order to compute the new layout, such as the DA-TU system [33] allows navigating and interactively clustering huge graphs.

Other works have studied the mental map preservation while the graph is being *updated* [22, 45]. This can be achieved by using force directed layout techniques [22], or by using simulated annealing [45].

There are some trade-offs between the readability and the stability of (offline) dynamic graph drawing methods. The most common approach uses the preceding layout as the initial layout for each graph [34, 48]; yet layout readability may degrade over the sequence of graphs. Other approaches address stability by either placing vertices of fixed vertex locations from the layout of an aggregate of all graphs [48]; or anchoring vertices to reference positions [9]; or linking vertices to instances of themselves that are close in the sequence [20]. For details, see a recent comprehensive study of the readability and the stability in these methods [7].

In contrast to the previous work, this paper defines a new alternative for the mental map preservation and the stability criteria. Our new criterion, namely change faithfulness, helps understand and evaluate dynamic graph visualizations.

2.4 Dynamic Graph Visualization

When analyzing dynamic graphs, it is critical to be able to see the statistical trends and changes over time, while preserving user's mental map [47].

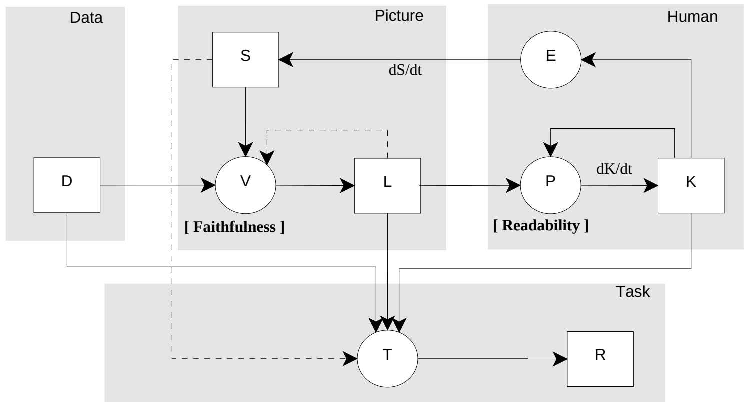


Fig. 3: Graph visualization model

The most common techniques for representing temporal data are via *animation* and *the "small multiples" display*. The animation approach shows visualizations of the sequence of graphs displayed in consecutive frames. The small multiples display uses multiple charts laid side-by-side and corresponding to consecutive time periods or moments in time [2].

Generally speaking, this previous work has been focused on the space dimension; for example, consider readability and stability or mental map preservation of 2D or 3D drawings of graphs.

In this paper, we are concerned about the faithfulness in both the *space* and the *time* dimensions.

3 Graph visualization model

In this section we describe a semi-formal model for the graph visualization. The section first gives basic annotations of graphs and then describe our graph visualization model.

A *graph* $G = (N, E)$ consists a set of nodes N and a set of edges E. In practice, the nodes and edges may have multiple attributes, such as textual labels. These attributes can be important for visualization. Further, nodes and edges may be timestamped, and the visualization varies over time.

Fig. 3 shows our visualization model. It is an extension of the van Wijk model [59]. Our model encapsulates the whole knowledge discovery process, from data to visualization to human; unlike the van Wijk model, our model includes *tasks*.

The main processes of the model are "visualization" V, "perception" P, and "task" T, and described as follows.

3.1 Visualization

The *visualization process* maps a *data* item $d \in D$ (an attributed graph) to a *layout* item (sometimes called a *"picture"*) $\ell = V(d) \in L$ according to a *specification* $s \in S$. We write this as follows[2]:

$$V : D \times S \to L \tag{1}$$

The type of data D to be visualized can vary from a simple list of nodes and edges to a time-varying graph with complex attributes on nodes and edges. The specification S includes, for example, a specification of the hardware used such as the size and the resolution of the screen.

The layout space L may consist of graph drawings in the usual sense, but more generally consists of structured objects in a multidimensional geometric space. Sometimes it is convenient to regard the layout space as the screen; in this case, using the language of Computer Graphics, it is an *image space*.

For incremental algorithms, the visualization process may use the previous layout when computing the current layout. The capability of modelling incremental algorithms as well as dynamic graphs makes our visualization model more general than the van Wijk model [59]. In this case, the general form of visualization process becomes:

$$V : D \times S \times L \to L,$$

where the previous layout is the input for computing the current layout. This is necessary, for example, when a layout algorithm attempts to preserve the user's mental map. However, unless otherwise stated we take the simpler model in equation (1).

3.2 Perception

The *perception process* maps a picture from the layout space L to the *knowledge* space K. We write this as follows:

$$P : L \to K \tag{2}$$

In this model we use the term *knowledge* – sometimes called *insight* or *mental picture* – to denote the effect on the human of his/her observation of the picture. Again, in time-varying situation, the human's perception can depend on the previous picture, and perhaps it is better to write:

$$P : L \times K \to K$$

However we use the simpler model (2) unless otherwise stated.

[2] For this semi-formal model, we use mathematical notation more as a concise shorthand rather than a precise description. For example, we describe processes as functions, but we should warn the reader that the domains and ranges of these are sometimes ill-defined.

Of course, it is difficult to formally model human knowledge or insight, and it is difficult to assess its value. In particular, in some situations such as *exploratory* visualization, the information that is contained in the data is not known a priori, and we make pictures to get serendipitous insight.

3.3 Task

Visualizations are a useful means for exploration and examination of data using visual representations or pictures. In many practical cases, visualizations are developed to serve for domain-specific *tasks*. To understand faithfulness, it is important to model these tasks.

Like visualization, the task process is executed by users or data analysers. The task process, however, is not necessarily performed by the same persons who create the pictures from the data. Furthermore, the task process can be performed directly by extracting the answers from the data with or without using a picture of that data.

Examples of common tasks include, for example, identifying important actors and communities in a social network, or exploring possible pathways in a biological network. Domain case studies (see, for example, [6, 61]) can be used to identify such tasks. Further, there are *low-level* tasks that are relevant across a wide variety of domains; such tasks have been identified and classified by psychologists (see, for example, [62]).

This paper models all tasks as processes that map the data space D, the layout space L, and the knowledge space K to a *result space* R. The result space may vary from a simple boolean space $\{\mathsf{true}, \mathsf{false}\}$ to a complex matrix space.

In this model, all the details (aka. questions) of a task are considered less important, similarly to the details (aka. algorithms/mechanisms) of a visualization method; thus, the specific questions of tasks are not taken into account in our model.

The central model for the task process is $T = (T_D, T_L, T_K)$, where T_D, T_L and T_K are three functions:

$$\begin{aligned} T_D &: D \to R \\ T_L &: L \to R \\ T_K &: K \to R. \end{aligned}$$

This task model is perhaps over-simplistic; for example, it does not model the *a priori* knowledge of the human. In addition, one may make the task process more general by taking some combination of data, layout and knowledge and then mapping to the result space. However, the simple model is sufficient to demonstrate the concept of faithfulness - task faithfulness. Consequently, this simple model of task is used throughout the rest of the paper.

4 Faithfulness Model

Informally, a graph visualization is *faithful* if the underlying network data and the visual representation are logically consistent. In this section, we develop this intuition into a semi-formal model. In fact, we distinguish three kinds of faithfulness: *information* faithfulness, *task* faithfulness, and *change* faithfulness. Then we discuss the difference between faithfulness and correctness, and then between faithfulness and readability.

4.1 Information faithfulness

The simplest form of faithfulness is *information faithfulness*. This is based on the idea that the visual representation of a data set should contain all the information of the data set, irrespective of tasks. In terms of the notation developed above, a visualization V is *information faithful* if the function V is one-one, that is, V has an inverse.

As an example, consider the classical barycenter visualization function that takes as input a planar graph $G = (N, E)$, places nodes from a specified face on the vertices of a convex polygon, and places every other node at the barycenter of its neighbors (see [56]). This function is information faithful on *internally triconnected* (see [56]) planar graphs. However, if the input graph is not internally triconnected, then same picture can result from several input graphs (each internal triconnected components is collapsed onto a line), and the method is not information faithful.

4.2 Task faithfulness

The intuition behind *task faithfulness* is that the visualization should be accurate enough to correctly perform tasks. In terms of the functions V and T defined above, a visualization V is *task faithful* with respect to specification $s \in S$ if

$$T_L(V(d, s)) = T_D(d) \tag{3}$$

for every data item $d \in D$.

If a visualization is information faithful, then clearly it is task faithful; however, the converse may not hold.

Consider, for example, a visualization function V_{cir} that draws all nodes a graph G on the circle. Clearly, V_{cir} is information faithful; Fig. 1a is an example, in which we can find all nodes and edges in the drawing. Further, V_{cir} is task-faithful: all the data is represented in the drawing, and so all tasks can be performed correctly using the drawing.

On the other hand, Fig. 1b shows a graph drawing using edge bundling. Consider the task to estimate the number of edges between two contiguous groups of nodes on the boundary of the circle. The edge bundled drawing is certainly task-faithful for this task; however, it is not information faithful, as the original graph is no longer reconstructable from the bundled layout.

4.3 Change faithfulness

The intuition behind *change faithfulness* is that a change in the visual representation should be consistent with the change in the original data. Note that this is a different concept to the mental map [19] or stability [52]; while these concepts are concerned with the user's interpretation of change, the concept of change faithfulness is concerned with the geometry of change.

Change faithfulness is important in dynamic settings, such as interactive or streamed graph drawing. However, it is also valid in static settings, because the difference between two pictures should be consistent with the difference between the two data items that they represent.

Consider, for example, a function V_{groups} that visualizes the interaction networks d that occur between European Science in Society researchers in Health[3].

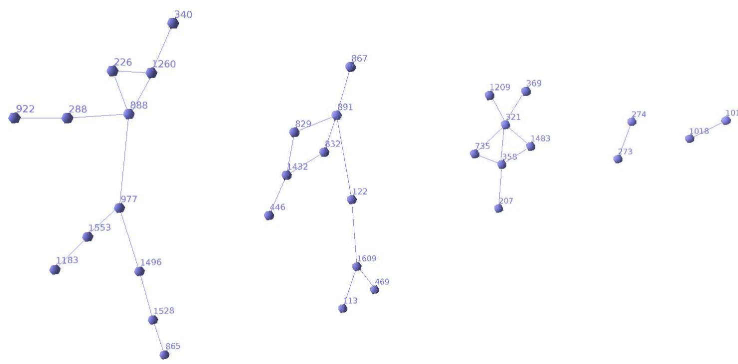


Fig. 4: Interaction groups between Health researchers in the EuroSiS dataset

Suppose that V_{groups} uses a force directed algorithm to draw the connected components of a graph $d \in D$ separately, and arranges these components horizontally across the screen, as in Fig. 4. Note that V_{groups} is information faithful. However, V_{groups} is not change faithful, because a small change in the graph d (such as adding an edge) can result in a large change in the picture.

4.4 Remarks

Remarks - faithfulness and correctness: We should stress that faithfulness is a different concept to the classical idea of *correctness* of an algorithm. An algorithm is *correct* if it does what it is required to do; correctness is an essential property of every algorithm. However, a visualization method may do exactly what it is required to do without achieving faithfulness.

Remarks - faithfulness and readability: More importantly, faithfulness is a different concept to readability. Readability is well studied in the Graph Drawing literature. It refers to the perceptual and cognitive interpretation of the picture

[3] Available at http://wiki.gephi.org/index.php/Datasets. Data use in this example are filtered for Health researchers only.

by the viewer. Readability depends on how the graphical elements are organised and positioned. It does not depend on whether the picture is a faithful representation of the data. We can divide the readability concept into three subconcepts in the same way as we divided faithfulness:

1. *Information readability*: A drawing is *information-readable* if the perception function P is one-one; that is, if two pictures appear the same to the user, then they are pictures of the same graph. Effectively, this is saying that *all* the information from the picture can be perceived by the human.
2. *Task readability*: A visualization is *task-readable* for a task $T = (T_D, T_L, T_K)$ if $T_K(P(\ell)) = T_L(\ell)$ for every layout $\ell \in L$.
3. *Change readability* is the classical *mental map*.

A good graph visualization method should achieve both faithfulness and readability; in practice, however, there may be a trade-off between the two ideals. This is especially true with large graphs, when the data size is too large for the visualisation to be information-faithful; indeed, the number of pixels may be smaller than the graph size. In such cases, faithfulness is sometimes be sacrificed for readability. In specific domains, there are important tasks for which the visualization can be both readable and task-faithful.

5 Extension models

We discuss a few extensions to our graph visualization model and faithfulness model.

5.1 Enhanced graph visualization model

Fig. 5 shows our enhanced model of the graph visualization model depicted in Fig. 3.

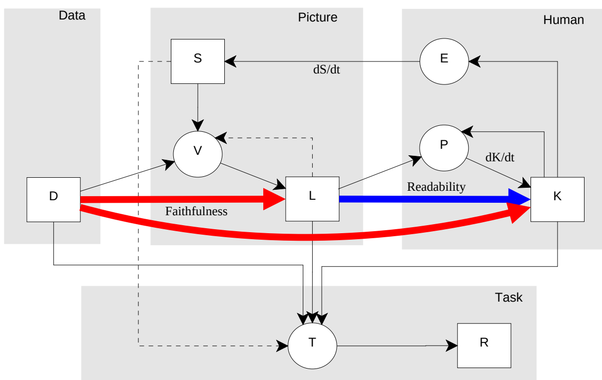


Fig. 5: Enhanced graph visualization model

In this new model, the faithfulness concept has been extended. Not only the faithfulness is defined from the visualization (from data to pictures), but also from the combination of visualization and perception (from data to pictures to knowledge). Intuitively, the new model of faithfulness is also concerned with the consistency between what the users are perceived and what is given in the data.

Generally speaking, this new concept of *perceptual faithfulness* is a compound of the visualization faithfulness and the visualization readability. This is inherently more general than our basic model:

- a faithful visualization may not be readable,
- a visualization, which is readable, may not be faithful
- a visualization is perceptually faithful if it is both faithful and readable

This notion of perceptual faithfulness is apparently useful for modern visualization techniques to handle large and complex graphs. Visualization techniques to be considered good should aim for for both faithfulness and readability.

5.2 Faithfulness in space and time

The above concepts of faithfulness concern mostly on the "*space*" dimension. That is, the visual mapping from data $d \in D$ to image $l = V(d) \in L$ does not (explicitly) consider the "*time*" factor.

We can extend the above concepts to integrate the time dimension. Let T denote the time domain and let $t \in T$ be a time point. A graph d at time t can be presented by a two-dimensional data item (d, t) of the graph d and the time t. The visualization process V transforms the two-dimensional data item into:

- a drawing $V(d, t)$ in which the drawing of $V(d)$ is placed at a location in space determined by t. For example, in small-multiple displays.
- a drawing $V(d)$ at a time frame $V(t)$. For example, in animation approaches.

In existing methods of dynamic graph visualizations, the time t is often the sequence number that represents the order of the graph d in the sequence of input graphs. Thus, the time transformation as an identity function, i.e., $V(t) = t$, is a safe assumption for these cases. The faithfulness concepts may disregard the time factors in these cases without loss of accuracy.

However, when time t is considered in a more general setting, there are several implications that need to be considered.

First, consider the faithfulness in the small-multiple approaches. The information faithfulness should consider the reversibility of the time t; for example, placing graph elements of d at a time t close together and avoid mixing elements of different times. Task faithfulness should consider the accuracy of task performance regarding the time attributes. For example, one should correctly identify if two data elements belong to the same time or not. Change faithfulness should further consider the change in time in the final visualizations. For example, two graphs d at time t and d' at time t' are placed close together if $|t - t'|$ is small; or placed far if $|t - t'|$ is large.

Second, consider the faithfulness in the animation approaches. The transformation of time t to $V(t)$ can be, for example, the identity function, a linear function, a sequence-based function, or a non-linear function. The information faithfulness should consider the inversibility of the time t; for examples, place graph elements of d at a time t in a separate frame t. Task faithfulness should consider the accuracy of task performance regarding the time attributes. For example, one should correct identify whether or not a graph element exists at time t. Change faithfulness should further consider the change of graphs together with the change in time in the animation. For example, two graphs d at time t and d' at time t' appear at frames $V(t)$ and $V(t')$ that are close/far in the animation if $|t - t'|$ is small/large.

6 Faithfulness metrics

Faithfulness is not a boolean concept; a visualization method may be a little bit faithful, but less than 100% faithful.

The classical concept of readability of a graph drawing can be evaluated using a number of metrics; these include, for example, the number of edge crossings, the number of edge bends, and the area of a grid drawing. These readability metrics are formal enough that the problem of constructing a readable graph drawing can be stated as a number of optimisation problems; thus optimisation algorithms can be used.

We aim to create a list of faithfulness metrics that play the same role. In this section we develop a framework for such metrics. We assume that the spaces D, L and R each have a *norm*, which we generically denote by $\|\cdot\|$. Roughly speaking, $\|x\|$ is the "size" of item x.

6.1 Information faithfulness metrics

The simplest way to measure the information faithfulness of a graph visualization function V is to measure its "ambiguity". For each $\ell \in L$, let $V^{-1}(\ell)$ denote the set $\{(d,s) \in D \times S : V(d,s) = \ell\}$ and let $|V^{-1}(\ell)|$ denote the number of elements in $V^{-1}(\ell)$. Then the *worst case information faithfulness* of V is a function f_{info} on the natural numbers, defined by

$$f_{info}(n) = \max_{\|\ell\| \leq n} \frac{1}{|V^{-1}(\ell)|}.$$

One can similarly define the average case information faithfulness of V.

A more subtle approach is to measure the information faithfulness of a graph visualization function V as the information loss in the channel. The loss of information during the visualization process is defined as *entropy* in information theory. The information loss is easier to measure than the total information content of a data set. There are several techniques for measuring information content and information loss (for a full discussion, see [53]).

6.2 Task faithfulness metrics

We can measure task faithfulness as the difference between the result from the visualization and the result from the data. The *worst case task faithfulness* of V is a function f_{task} on the natural numbers, defined by

$$f_{task}(n) = \max_{\|d\| \leq n} \|T_L(V(d, s)) - T_D(d)\|$$

with respect to the task $T = (T_D, T_L, T_K)$ and specification $s \in S$.

6.3 Change faithfulness metrics

Tufte [58] defines the "lie-factor" as the ratio of change in a graphical representation to the ratio of change in the data. We can express Tufte's concept in terms of our model: suppose $d, d' \in D$; denote $\|d' - d\|/\|d\|$ by $\Delta(d', d)$. Similarly, if $\ell, \ell' \in V$ then we denote $\|v' - v\|/\|v\|$ by $\Delta(v', v)$. Then the *lie factor* for $d, d' \in D$ is

$$\mathsf{lie}(d', d, s) = \frac{\Delta(V(d, s), V(d', s))}{\Delta(d', d)}.$$

Tufte's aim is to measure the quality of *static* visualizations in terms of the lie-factor, but we can apply the same principle in a dynamic setting.

Intuitively, the lie factor increases as change faithfulness decreases, and so for two data elements d' and d we can measure the worst case change faithfulness as:

$$f_{change}(n) = \max_{\|d' - d\| \leq n} \left(\frac{1}{\mathsf{lie}(d', d, s)}\right).$$

7 Example 1: Multidimensional scaling and force directed approaches

This section discusses the *multidimensional scaling (MDS)* [8] and *force directed* approaches to Graph Drawing [18, 21, 23, 25, 32, 54] in terms of faithfulness.

The MDS approach to Graph Drawing works as follows. The input is a graph $G = (N, E)$, and a $|N| \times |N|$ matrix of *dissimilarities* $\delta_{u,v}$. The goal is to map each node $u \in N$ to a point $p_u \in R^k$ such that the given dissimilarities $\delta_{u,v}$ are well-approximated by the distances $d_{u,v} = \|p_u - p_v\|$. The set of points p_u forms the layout $\ell = V(G)$ of G. In practice, k is commonly 2 or 3. In most applications, $\delta_{u,v}$ is chosen to be the graph theoretic distance between nodes u and v.

To measure the success of an MDS function, a "stress" [41] function is commonly used to compute the distortion between dissimilarities $\delta_{u,v}$ and fitted distances $d_{u,v}$. In the simplest case, the stress of a layout $\ell \in L$ is:

$$stress^{(node)}(\ell) = \sum_{u \neq v} (\delta_{u,v} - d_{u,v})^2, \tag{4}$$

MDS can be seen as an optimisation problem where the goal is to minimise this stress function. *Force directed algorithms* have a similar flavour, but view the problem as finding equilibrium in a system of forces.

7.1 Task faithfulness

The stress formula (4) can be seen in terms of our task faithfulness framework. Suppose that T is a task that depends on the graph theoretic distance between nodes; let R be the set of real-valued matrices indexed on the node set. For a graph $G = (N, E) \in D$, let $T_D(G)$ be the matrix $[\delta_{u,v}]_{u,v \in N}$. Suppose that the visualization V places node u at location p_u; let $T_L(V(G))$ be the matrix $[d_{u,v}]_{u,v \in N}$. Then define

$$f_{task}^{(node)}(n) = \max_{G \in D_n} \|T_L(V(G)) - T_D(G)\|_2, \tag{5}$$

where D_n is the set of graphs of size n and $\|\cdot\|_2$ is the Frobenius norm. Clearly, minimising task faithfulness is equivalent to minimising the stress defined by (4).

7.2 Change faithfulness

Further, we can evaluate the change faithfulness of an MDS method. In fact, MDS methods have been used extensively in dynamic settings, using stress to preserve the mental map. Suppose that at time t, we have a graph $G^{(t)}$, and the visualization function places node u at point $p_u^{(t)}$ at time t. A stress function can calculate the difference between the layout $\ell^{(t)} \in L$ at time t and the layout $\ell^{(t')} \in L$ at an earlier time t':

$$stress^{(node)}(\ell^{(t)}, \ell^{(t')}) = \sum_{u \in N} \|p_u^{(t)} - p_u^{(t')}\|^2.$$

In the so-called "anchoring" approach, t' is zero; in the "linking" approach, t' is the previous time frame before t (see [7]). These measures, however, aim for the mental map preservation - or change readability - rather than change faithfulness. For example, they aim to ensure that if the change in the graph is *small*, then the change in the layout is *small*. They do *not* ensure that if the change in the graph is *large*, then the change in the layout is *large*.

However, we can use the stress approach to define the lie factor, such as:

$$\mathsf{lie}(\ell^{(t)}, \ell^{(t')}) = (\sum_{u \neq v} \| \frac{d_{u,v}^{(t)} - d_{u,v}^{(t')}}{d_{u,v}^{(t')}} \|)/(\sum_{u \neq v} \| \frac{\delta_{u,v}^{(t)} - \delta_{u,v}^{(t')}}{\delta_{u,v}^{(t')}} \|)$$

where $d_{u,v}^{(t)} = \|p_u^{(t)} - p_u^{(t')}\|$ and $\delta_{u,v}^{(t)}$ denotes the graph theoretic distance between u and v in $G^{(t)}$.

Then the worst case change faithfulness can be measured in terms of the distortion of the data change relative to the layout change:

$$f_{change}(n) = \max_{t - t' \leq n} (\frac{1}{\mathsf{lie}(\ell^{(t)}, \ell^{(t')})})$$

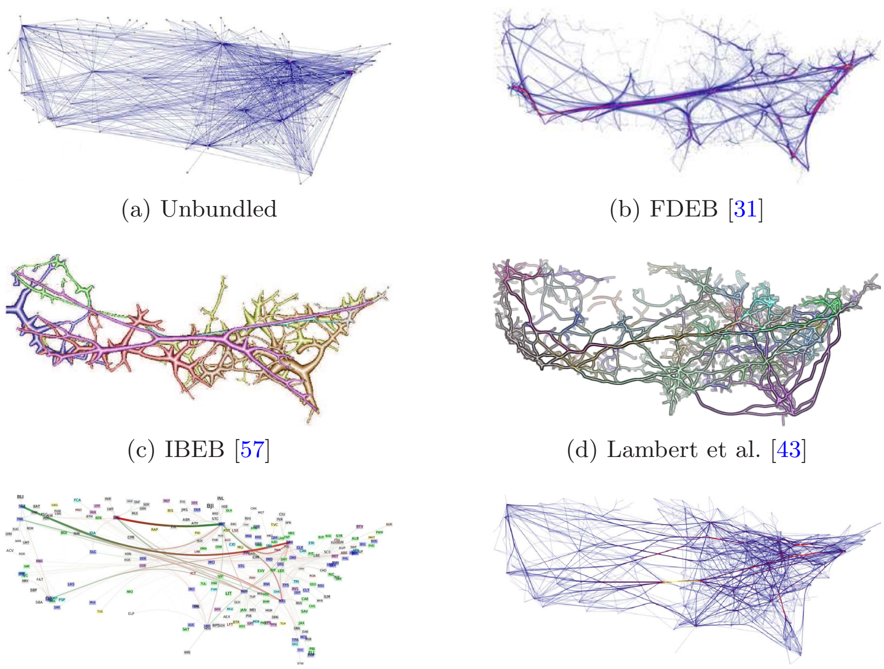
7.3 Remarks

The force directed and MDS approaches has had considerable impact on the commercial world, despite the fact that they do not have explicit or validated readability goals. We believe that the success of these approaches is due to the fact that they have explicit and validated task faithfulness goals with respect to tasks that depend on graph theoretic distances. We suggest that better MDS methods could be designed by optimising their change faithfulness using the lie factor stress above.

8 Example 2: Edge bundling

Edge bundling, as illustrated in Figs. 1b and 2b, has been extensively investigated to reduce visual clutter in graph visualizations. Many edge bundling techniques have been proposed, including *hierarchical* edge bundling [30], *geometry-based* edge clustering [15, 30, 44, 63], *force-directed* edge bundling [31, 40, 50, 55] and *multi-level agglometive* edge bundling [24].

The US airline networks are typically examples for edge bundling methods. Fig. 6 shows several edge bundling results.



(a) Unbundled

(b) FDEB [31]

(c) IBEB [57]

(d) Lambert et al. [43]

(e) TGI-EB [50]

(f) MINGLE [24]

Fig. 6: Examples of US airline networks using edge bundling

Edge bundling seems to increase *task readability* with respect to some tasks; for example, the classic bundling of air traffic routes in the USA (see [15, 24, 31, 50, 57]) seems to make it easier for a human to identify the main hubs and flight corridors. However, some readability metrics are sacrificed; for example, the number of bends is increased, making individual paths difficult to follow (the authors are not aware of any human experiments that measure readability for edge bundling). In this Section we make some remarks about the *faithfulness* of edge bundling.

8.1 Information faithfulness

As noted in Section 1, edge bundling reduces information faithfulness: as more edges are bundled together, it becomes harder to reconstruct the network data from a bundled layout. We can propose a rough-and-ready metric for this reduction based on the model presented in Section 6. Given an input graph $G = (N, E)$, an edge bundling visualization process V partitions E into *bundles* $E = B_1 \cup B_2 \cup \ldots \cup B_k$. Let G_i denote the subgraph of G with edge set B_i and node set N_i consisting of endpoints of edges in B_i. Edge bundling methods ensure that G_i is bipartite; suppose that $N_i = X_i \cup Y_i$ is the bipartition of G_i. In the bundled layout, G_i is indistinguishable from a complete bipartite graph on the parts X_i and Y_i. This representation has inherent information loss. Let $x_i = |X_i|$, and $y_i = |Y_i|$. The number of (labeled) bipartite graphs with parts X_i and Y_i is $2^{x_i y_i}$. Thus if $q = \sum_{i=1}^{k} x_i y_i$, then there are 2^q graphs that have the same layout as G. This can be used as a simple model for computing the information faithfulness of V.

8.2 Task faithfulness

Most bundling methods use a *compatibility* function; roughly speaking, a compatibility function C assigns a real number $C(e, e')$ to each pair e, e' of edges. Two edges e and e' are more likely to be bundled together if the value of $C(e, e')$ is large. A number of compatibility functions have been proposed and tested; these include *spatial* compatibility [31] from length, position, angle and visibility between edges, *semantic* compatibility [40] for bundling multi-attributed edges, *connectivity* compatibility [55], *importance* compatibility and *topology* compatibility in TGI-EB [50]. Some of these functions depend only on the input graph G, and some depend also on the layout of G.

For a number of tasks, such as identifying hubs in a network, highly compatible edges are equivalent; the correct performance of such tasks does not depend of distinguishing between them. Here we show that stress functions can be used to define metrics for computing the task faithfulness of the edge bundled layout relative to such tasks.

Given a pair of edges e and e' in an input graph G, let $C(e, e')$ denote their compatibility. We assume that this compatibility function depends only on G and not on its layout. Let $\ell \in L$ be the layout of G. For two edges e and e', let $d(e, e')$ be the distance between the curves representing e and e' in ℓ. We can choose from a number of distance functions for curves, such as the *Fréchet distance* [1] and several distance measures for point sets [11]. The stress in ℓ is then defined as:

$$stress^{(edge)}(\ell) = \sum_{e \neq e'} (C(e, e') - d(e, e'))^2. \tag{6}$$

We can then define the *worst (average) case task faithfulness* $f(n)$ of an edge bundling visualization function V as the maximum (average) of $stress^{(edge)}(\ell)$ over all input graphs of size n.

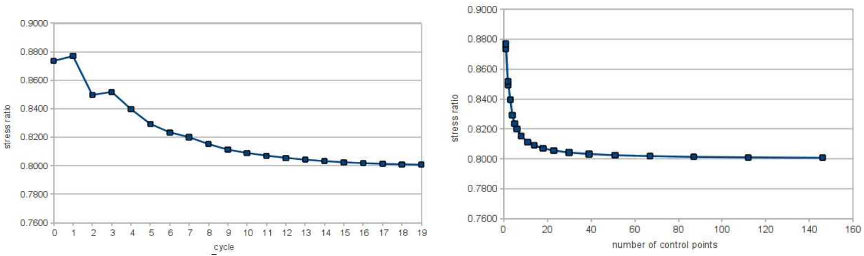
8.3 Remarks

Despite the plethora of recent papers in edge bundling, there are few evaluations of effectiveness. Using the formal models outlined above, one can begin to evaluate faithfulness and compare bundling methods. For example, one can test the following hypotheses:

H 1 *The force-directed edge bundling (FDEB) [31] methods and its force-directed variants [40, 50, 55] are task-faithful.*

H 2 *Force-directed edge bundling methods are* more *task-faithful when using* more *control points per edge.*

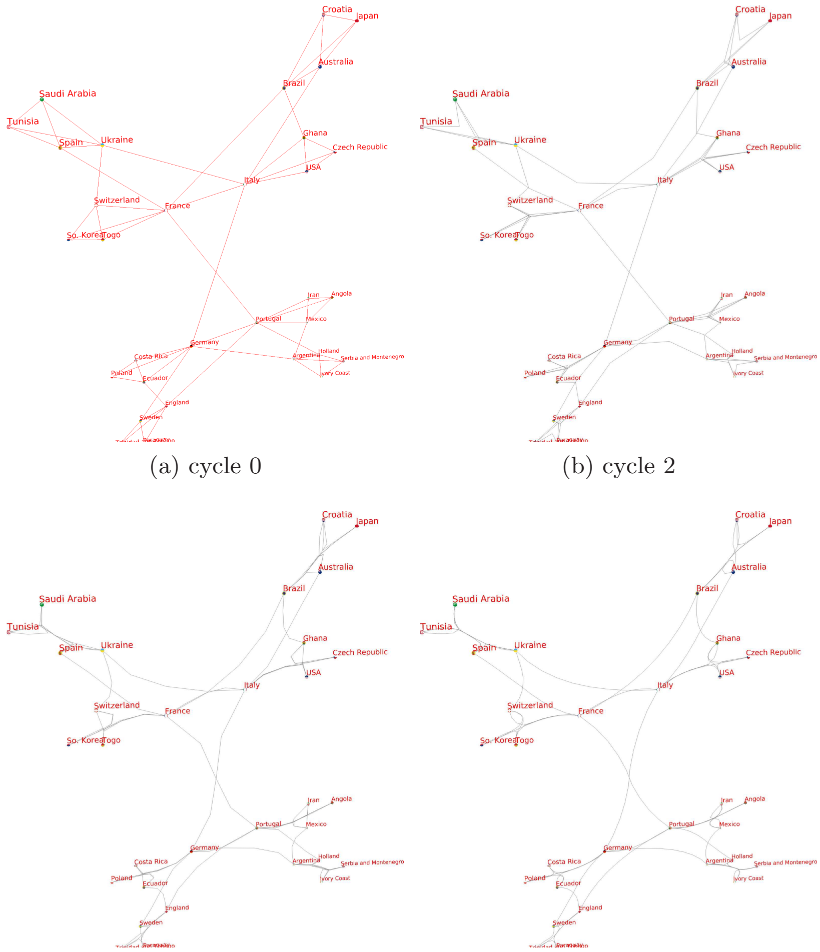
Our initial studies (see Appendix 8.3) using the metrics above have shown a confirmation of the above hypotheses.



(a) cycle vs stress ratio

(b) number of control points vs. stress ratio

Fig. 7: Comparisons of the faithfulness of the edge bundled worldcup visualizations



(a) cycle 0 (b) cycle 2

(c) cycle 5 (d) cycle 9

Fig. 8: Visualization of FIFA worldcup data year 2006

Hypotheses on the Faithfulness of Edge bundling We have conducted several experiments for the faithfulness metrics, described in Section 6. We use the worldcup data[4] as our examples.

Fig. 7 shows statistics of stress values at different iteration cycles of force-directed edge bundling algorithm (FDEB) [31]. The figure has shown that FDEB achieves more faithful results when more number of cycles are performed. Furthermore, in later cycles the algorithm uses more control points to improve its layout, the result becomes more faithful.

[4] data available at http://gd2006.org/contest/WCData/

Fig. 8 shows the visualizations of the football matches in the year 2006. The figure also shows the bundled results computed by FDEB.

9 Example 3: Visualization metaphors

This section discusses our new notions of faithfulness for several representative metaphors for graph visualization.

9.1 Matrix representation

Besides node-link diagrams, visualization of graphs as matrix form is also popular. Generally speaking matrix metaphor is very faithful, as all the nodes and edges are represented in the visualization. However, for certain tasks such as path tracing, matrix metaphor is not task-readable.

There has been a number of research aiming to improving readability of matrix metaphors; for instance, reordering columns/rows to show highly connected groups of nodes.

9.2 Cartography

There is a recent upsurge in interests of map-based visualizations. For example, the music land visualization [26] gives attractive visualizations depicted in Fig. 9.

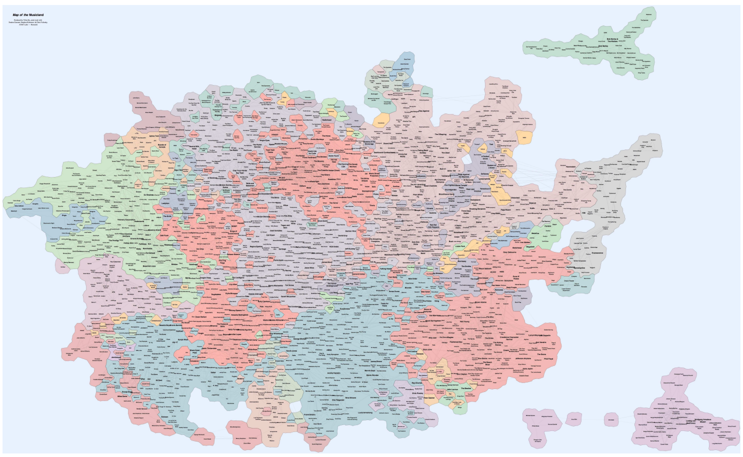


Fig. 9: Visualization of music categories

These map-based approaches do increase task faithfulness as many tasks such as identifying similar topics become straight-forward. However, they do sacrifice information faithfulness to increase readability. Typically, links with small weights can be discarded to focus on more important links and to create appealing maps with more readable boundaries.

9.3 Compound visualizations

Compound visualization techniques combine several types of visualization metaphors into the final result. Examples include, for example, MatLink [28], NodeTrix [29], which integrate matrix views with node-link diagrams. Fig. 10 shows an example of the hybrid visualizations by NodeTrix [29].

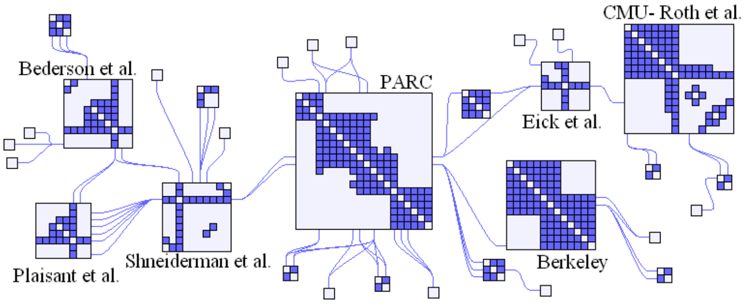


Fig. 10: A Hybrid Visualization of Social Networks using NodeTrix

In general, these techniques [28, 29] seem to increase readability for some parts of the networks that are of most interests and at the same time may sacrifice faithfulness and readability in other parts of the network that are less important. Furthermore, there can be a trade-off between global readability and local readability, and between global faithfulness and local faithfulness.

10 Discussions and Future work

This section discusses about recent advances and important factors in graph visualization.

Research in visualization in general and graph visualization in particular is influenced by two main factors: (1) display capacity (hardware), and (2) interaction. The former is related to the availability of improved display resources. The latter refers to the development of novel interaction algorithms to adapt to main display parameters such as the level of detail, data selection and aggregation, by which the data is presented.

10.1 Display device

Desktop displays include the high-resolution displays and advanced display devices, such as large-scale power walls and small portable personal assistant, graphically-enabled devices. Faithfulness metrics should be addressed in a specific formula depending on the characteristics of the available output devices.

10.2 Interaction

The analyst needs to be able to fully focus on the task irrespective of the complexity of the user interface, which could be potentially distracting. Novel interaction techniques need to support seamlessly visual communication of the user with the system. To this end, user inputs are a must to support of the user in navigating and analyzing the data, memorizing insights and making informed decisions.

Consider the case when the user's attention concerns only a certain part of the graph, for example, through data queries or by repeated direct manipulations of the same graphical elements. The layout adjustment could sacrifice the overall faithfulness to the local faithfulness (for the part of the graph in concern)

Beyond that, distortion techniques such as lens effects and occlusion reduction also provide the analyst with trade-off between faithfulness and visual clarity. Each of them typically transforms object's position to improve local readability at the cost of accuracy of global relations.

10.3 Level of details

Appropriate solutions need to intelligently combine visualizations of selected analysis details and a global overview. The relevant data patterns and relationships need to be visualized on several levels of detail, and with appropriate levels of data and visual abstraction. The overall goal is to balance between the faithfulness and the readability to maximise the user's expectations.

10.4 Affine transformation

The intuition of faithfulness does not depend on global affine transformations such as rotation, translation and scaling. However, some transformations of the local parts of the graphs may increase readability, yet may degrade faithfulness.

10.5 Metrics for compound visualizations

It would be interesting to define faithfulness metrics for compound visualizations. Intuitively, the compound methods should balance between faithfulness values of different parts of the networks, and they should also balance between faithfulness globally (overall) and locally (sub-networks).

11 Conclusion

This paper has introduced the concept of faithfulness for graph visualization. We believe that the classical readability criteria are necessary but not sufficient for quality graph drawing; faithfulness is the generic criterion that is missing. The paper describes the faithfulness concept and gives a semi-formal model.

We give a model for metrics for faithfulness. We illustrate the concept with two examples. The first of these is multidimensional scaling / force-directed methods. We believe that future directions of these methods would need to balance the aims of readable outputs versus faithful representations. The second example is edge bundling. Despite the current upsurge of interest in edge bundling, there are very few evaluations; we show that faithfulness metrics may prove the key to evaluation.

We conclude with a remark about 3D graph drawing. The occlusion problem for 3D means that, even with binocular displays, some part of the graph is hidden. This can be seen as a lack of not only readability but also information faithfulness. We believe that the lack of commercial impact of 3D graph drawing is partially due to its inherent lack of faithfulness.

References

1. H. Alt and M. Godau, *Computing the fréchet distance between two polygonal curves*, International Journal of Computational Geometry and Applications **5** (1995), no. 1, 75–91.
2. N. Andrienko and G. Andrienko, *Exploratory analysis of spatial and temporal data: a systematic approach*, Springer Verlag, 2006.
3. Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis, *Graph drawing: Algorithms for the visualization of graphs*, Prentice-Hall, 1999.
4. E. Bertini, *Quality metrics in high-dimensional data visualization: An overview and systematization*, Visualization and Computer Graphics, IEEE Transactions on **17** (2011), no. 12, 2203–2212.
5. U. Brandes, *Drawing on physical analogies*, Drawing Graphs (2001), 71–86.
6. U. Brandes and T. Erlebach, *Network analysis: methodological foundations*, Springer Verlag, 2005.
7. U. Brandes and M. Mader, *A quantitative comparison of stress-minimization approaches for offline dynamic graph drawing*, Graph Drawing, Springer, 2012, pp. 99–110.
8. U. Brandes and C. Pich, *Eigensolver methods for progressive multidimensional scaling of large data*, Graph Drawing, Springer, 2007, pp. 42–53.
9. U. Brandes and D. Wagner, *A bayesian paradigm for dynamic graph layout*, Graph Drawing, Springer, 1997, pp. 236–247.
10. Ulrik Brandes and Christian Pich, *Eigensolver methods for progressive multidimensional scaling of large data*, Graph Drawing, 2006, pp. 42–53.
11. S.S. Bridgeman and R. Tamassia, *A user study in similarity measures for graph drawing*, J. Graph Algorithms Appl. **6** (2002), no. 3, 225–254.
12. A. Buja, D.F. Swayne, M.L. Littman, N. Dean, H. Hofmann, and L. Chen, *Data visualization with multidimensional scaling*, Journal of Computational and Graphical Statistics **17** (2008), no. 2, 444–472.
13. S. Carpendale, *Evaluating information visualizations*, Information Visualization (2008), 19–45.
14. M. Chen, D. Ebert, H. Hagen, R.S. Laramee, R. Van Liere, K.L. Ma, W. Ribarsky, G. Scheuermann, and D. Silver, *Data, information, and knowledge in visualization*, Computer Graphics and Applications, IEEE **29** (2009), no. 1, 12–19.
15. W. Cui, H. Zhou, H. Qu, P.C. Wong, and X. Li, *Geometry-based edge clustering for graph visualization*, TVCG (2008), 1277–1284.

16. A. Dasgupta and R. Kosara, *Adaptive privacy-preserving visualization using parallel coordinates*, Visualization and Computer Graphics, IEEE Transactions on **17** (2011), no. 12, 2241–2248.
17. S. Diehl and C. Görg, *Graphs, they are changing*, Graph drawing, Springer, 2002, pp. 23–31.
18. P. Eades, *A heuristic for graph drawing*, 1984, pp. 149–160.
19. P. Eades, Lai W., K. Misue, and Sugiyama K., *Preserving the mental map of a diagram*, Proceedings of COMPUGRAPHICS, 1991, pp. 34–43.
20. C. Erten, S. Kobourov, V. Le, and A. Navabi, *Simultaneous graph drawing: Layout algorithms and visualization schemes*, Graph Drawing, Springer, 2004, pp. 437–449.
21. B. Finkel and R. Tamassia, *Curvilinear graph drawing using the force-directed method*, Graph Drawing, Springer, pp. 448–453.
22. Y. Frishman and A. Tal, *Dynamic drawing of clustered graphs*, Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on, IEEE, 2004, pp. 191–198.
23. T. M. J. Fruchterman and E. M. Reingold, *Graph drawing by force-directed placement*, Softw. Pract. Exper. **21** (1991), no. 11, 1129–1164.
24. E. Gansner, Y. Hu, S. North, and C. Scheidegger, *Multilevel agglomerative edge bundling for visualizing large graphs*, IEEE PacificVis, 2011, pp. 187–194.
25. E. R. Gansner and S. C. North, *Improved force-directed layouts*, Graph Drawing, 1998, pp. 364–373.
26. E.R. Gansner, Y. Hu, and S. Kobourov, *Gmap: Visualizing graphs and clusters as maps*, PacificVis, IEEE, 2010, pp. 201–208.
27. D. Harel and Y. Koren, *A fast multi-scale method for drawing large graphs*, Graph drawing, Springer, 2001, pp. 235–287.
28. N. Henry and J.D. Fekete, *Matlink: Enhanced matrix visualization for analyzing social networks*, Human-Computer Interaction–INTERACT (2007), 288–302.
29. N. Henry, J.D. Fekete, and M.J. McGuffin, *Nodetrix: a hybrid visualization of social networks*, TVCGn **13** (2007), no. 6, 1302–1309.
30. D. Holten, *Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data*, TVCG (2006), 741–748.
31. D. Holten and J. J. van Wijk, *Force-directed edge bundling for graph visualization*, Computer Graphics Forum **28** (2009), no. 3, 983–990.
32. Y.F. Hu, *Efficient and high quality force-directed graph drawing*, The Mathematica Journal **10** (2005), 37–71.
33. M. Huang and P. Eades, *A fully animated interactive system for clustering and navigating huge graphs*, Graph Drawing, Springer, 1998, pp. 374–383.
34. M.L. Huang, P. Eades, and J. Wang, *On-line animated visualization of huge graphs using a modified spring algorithm*, Journal of Visual Language and Computing **9** (1998), no. 6, 623–645.
35. J. Hullman, E. Adar, and P. Shah, *Benefitting infovis with visual difficulties*, Visualization and Computer Graphics, IEEE Transactions on **17** (2011), no. 12, 2213–2222.
36. J. Hullman and N. Diakopoulos, *Visualization rhetoric: Framing effects in narrative visualization*, Visualization and Computer Graphics, IEEE Transactions on **17** (2011), no. 12, 2231–2240.
37. H. Jänicke, T. Weidner, D. Chung, R.S. Laramee, P. Townsend, and M. Chen, *Visual reconstructability as a quality metric for flow visualization*, Computer Graphics Forum, vol. 30, 2011, pp. 781–790.
38. N. Jaworska and A. Chupetlovska-Anastasova, *A review of multidimensional scaling (mds) and its utility in various psychological domains*, Tutorials in Quantitative Methods for Psychology **5** (2009), no. 1, 1–10.

39. T. Kamada and S. Kawai, *An algorithm for drawing general undirected graphs*, Information processing letters **31** (1989), no. 1, 7–15.
40. W. Kienreich and C. Seifert, *An application of edge bundling techniques to the visualization of media analysis results*, InfoVis, 2010, pp. 375–380.
41. J.B. Kruskal, *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis*, Psychometrika **29** (1964), no. 1, 1–27.
42. H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale, *Seven guiding scenarios for information visualization evaluation*, Technical Report, UNIVERSITY OF CALGARY Calgary (2011).
43. A. Lambert, R. Bourqui, and D. Auber, *3D Edge Bundling for Geographical Data Visualization*, InfoVis, IEEE, 2010, pp. 329–335.
44. ______, *Winding Roads: Routing edges into bundles*, Computer Graphics Forum, vol. 29, Wiley Online Library, 2010, pp. 853–862.
45. Y.Y. Lee, C.C. Lin, and H.C. Yen, *Mental map preserving graph drawing using simulated annealing*, Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation-Volume 60, Australian Computer Society, Inc., 2006, pp. 179–188.
46. T. Matsuyama, X. Wu, T. Takai, and S. Nobuhara, *Real-time 3d shape reconstruction, dynamic 3d mesh deformation, and high fidelity visualization for 3d video*, Computer Vision and Image Understanding **96** (2004), no. 3, 393–434.
47. K. Misue, P. Eades, W. Lai, and K. Sugiyama, *Layout adjustment and the mental map*, Journal of visual languages and computing **6** (1995), no. 2, 183–210.
48. J. Moody, D. McFarland, and S. Bender-deMoll, *Dynamic network visualization1*, American Journal of Sociology **110** (2005), no. 4, 1206–1241.
49. E. Namey, G. Guest, L. Thairy, and L. Johnson, *Data reduction techniques for large qualitative data sets*, Handbook for team-based qualitative research (2007), 137–162.
50. Q. Nguyen, S.H. Hong, and P. Eades, *TGI-EB: A new framework for edge bundling integrating topology, geometry and importance*, GD, Springer, 2011, pp. 123–135.
51. S. North, *Incremental layout in dynadag*, Graph Drawing, Springer, 1996, pp. 409–418.
52. F.N. Paulisch and W.F. Tichy, *Edge: An extendible graph editor*, Software: Practice and Experience **20** (1990), no. S1, S63–S88.
53. H. Purchase, N. Andrienko, T. Jankun-Kelly, and M. Ward, *Theoretical foundations of information visualization*, InfoVis (2008), 46–64.
54. A. Quigley and P. Eades, *Fade: Graph drawing, clustering, and visual abstraction*, Graph Drawing (London, UK), Springer-Verlag, 2001, pp. 197–210.
55. D. Selassie, B. Heller, and J. Heer, *Divided edge bundling for directional network data*, TVCG **17** (2011), no. 12, 2354–2363.
56. R. Tamassia, *Handbook of graph drawing and visualization*, Discrete Mathematics And Its Applications. Online draft available at http://www.cs.brown.edu/~rt/gdhandbook/.
57. A. Telea and O. Ersoy, *Image-Based Edge Bundles: Simplified Visualization of Large Graphs*, Computer Graphics Forum, vol. 29, Wiley Online Library, 2010, pp. 843–852.
58. E.R. Tufte, *The visual display of quantitative information. 2001*, Visual Explanations, 194–95.
59. J.J. Van Wijk, *The value of visualization*, VIS '05, IEEE, 2005, pp. 79–86.
60. C. Walshaw, *A multilevel algorithm for force-directed graph drawing*, Graph Drawing, Springer, 2001, pp. 31–55.
61. Stanley Wasserman and Katherine Faust, *Social network analysis: Methods and applications*, 1 ed., Cambridge University Press, November 1994.

62. Stephen Wehrend and Clayton Lewis, *A problem-oriented classification of visualization techniques*, VIS '90, IEEE, 1990, pp. 139–143.
63. H. Zhou, X. Yuan, W. Cui, H. Qu, and B. Chen, *Energy-based hierarchical edge clustering of graphs*, IEEE PacificVis, IEEE, 2008, pp. 55–61.

ISBN 978-1-74210-283-2

School of Information Technologies
Faculty of Engineering & Information Technologies
Level 2, SIT Building, J12
The University of Sydney
NSW 2006 Australia

T +61 2 9351 3423
F +61 2 9351 3838
E sit.information@sydney.edu.au
sydney.edu.au/it

ABN 15 211 513 464
CRICOS 00026A