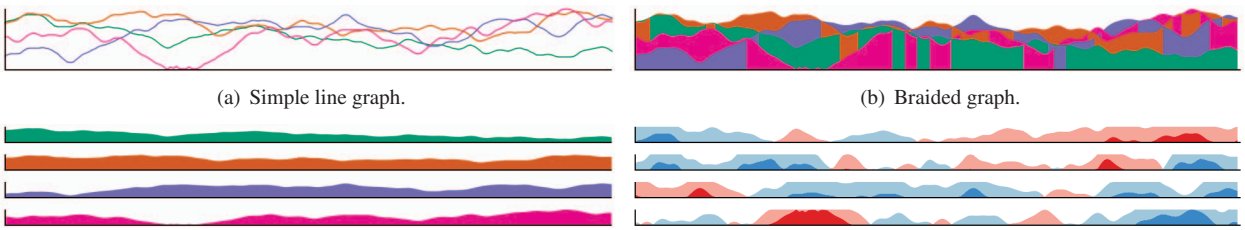
Graphical Perception of Multiple Time Series

Waqas Javed, *Student Member, IEEE*, Bryan McDonnel, *Student Member, IEEE*, and Niklas Elmqvist, *Member, IEEE*



(a) Simple line graph. (b) Braided graph.

(c) Small multiples (d) Horizon graphs.

Fig. 1. Four visualization techniques for multiple time series. This example shows the same four time series (200 data points).

Abstract—Line graphs have been the visualization of choice for temporal data ever since the days of William Playfair (1759–1823), but realistic temporal analysis tasks often include multiple simultaneous time series. In this work, we explore user performance for comparison, slope, and discrimination tasks for different line graph techniques involving multiple time series. Our results show that techniques that create separate charts for each time series—such as small multiples and horizon graphs—are generally more efficient for comparisons across time series with a large visual span. On the other hand, shared-space techniques—like standard line graphs—are typically more efficient for comparisons over smaller visual spans where the impact of overlap and clutter is reduced.

Index Terms—Line graphs, braided graphs, horizon graphs, small multiples, stacked graphs, evaluation, design guidelines.

1 Introduction

When William Playfair (1759–1823) invented the line graph in 1786 [24] to help people understand time series data, he can hardly have imagined the repercussions his work would have on posterity. Now hailed as the father of statistical graphics [11], Playfair—a Scottish engineer—used his line, bar, pie, and circle graphs to communicate political and economical data [12]. Line graphs are today one of the most common types of statistical data graphics [3], and are used to visualize temporal data in a wide array of domains such as finance, politics, science, engineering, and medicine.

However, while standard line graphs can easily deal with a few time series simultaneously, common tasks involving time series data often involve many concurrent series [17]. Consider a stock analyst surveying the history of a set of stocks in an effort to find the next investment. This comparison will have to be conducted across each of the time series representing each individual stock. While recent work [16] investigated the performance of a novel time series visualization technique—horizon graphs [26]—for different chart sizes, this study only involved **two** time series at all times. Other similar graphical perception work tend to only involve discrimination and estimation between two charts as well [27]. Lam et al. [20] studied multiple (more than two) time series, but focused on multi-resolution visualization techniques. Thus, there exists little data on graphical perception for multiple time series as a function of different line graph techniques.

In this paper, we address this lack of knowledge by rigorously evaluating graphical perception for different tasks involving multiple time series through controlled laboratory experiments. The main motivation for this work is to provide guidelines for designers who need to find a suitable method when building a temporal visualization application. Beyond studying simple line graphs [24], we also include small multiples [28] and horizon graphs [26] in our experiment. In addition, to aid perception of multiple color-coded time series, we include a novel visualization technique that we call a *braided graph* where filled areas are sorted in depth order for each position along the time axis.

Our results could influence a wide range of disciplines where temporal data are viewed and analyzed. However, there is a limit to the perceptual abilities of the human analyst, and thus there comes a point when the graphical perception task becomes impossible due to too many concurrent time series and to the correspondingly high visual clutter [8]. For these situations, we need alternative methods such as temporal queries [17], hierarchical aggregation [9], or temporal clustering [19]. While these methods are outside the scope of this paper, we are also interested in finding the point where the graphical perception of a typical user breaks down for the above techniques.

2 Related Work

Evaluation of graphical perception for statistical data graphics has a long history, originating from even before there were computers and graphics to turn charts into interactive visualizations. The pioneering work by Eells [7] set the stage for comparing different types of graphical representations. Croxton et al. compared bar charts with circle diagrams and pie charts [6], and also discussed the relative merits of bars, squares, circles and cubes to perform the comparison tasks [5]. Peterson et al. [23] measured the accuracy of reading values from eight different graphical representations of statistical data.

Early work to find the effectiveness and merits of different graph types later came under the umbrella of *graphical perception* of statistical graphics [4]. Graphical perception is defined as the ability of users to comprehend the visual encoding and thereby decode the information presented in the graph [22]. Simkin and Hastie [27] compared the accuracy of judgment while using simple bar charts, divided bar charts, and pie charts based on the comparison and estimation tasks, but they only involved two charts at a time.

- *Waqas Javed is with Purdue University in West Lafayette, IN, USA, E-mail: wjaved@purdue.edu.*
- *Bryan McDonnel is with Purdue University in West Lafayette, IN, USA, E-mail: bmcdonne@purdue.edu.*
- *Niklas Elmqvist is with Purdue University in West Lafayette, IN, USA, E-mail: elm@purdue.edu.*

Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

In the human-computer interaction field, Lohse [22] worked on developing a cognitive model to understand graphical perception. He also performed an empirical study to compare computer-simulated graphical perception based on his model with the actual performance of the user [21]. Meanwhile, Gillan [13] developed a perceptual model that explains human interaction with graphs. In later work [14], he studied this model for various representations like line graphs, scatterplots, stacked bars, and pie charts.

Huang et al. [18] conducted three user studies to determine the usefulness of a cognitive approach for measuring graphical perception. They argued that the cognitive behavior of the user can be useful beyond merely measuring time and error. Most recently, Heer et al. [16] performed two controlled experiments to measure the effect of chart size and layering on user performance while performing discrimination and estimation tasks on data.

Finally, Lam et al. [20] did investigate graphical perception of multiple line series, but their study focuses more on differences between low-resolution and high-resolution visual representations than on comparing the performance of line graph techniques. While their motivation is different, parts of their study are very relevant to ours.

3 Visualization of Multiple Time Series

It is not clear that the existing results on graphical perception of two data series, discussed above, generalize to the case where we have more than two time series. In this section, we will derive suitable evaluation criteria and then discuss different line graph visualization techniques individually. In the following sections, we will test them empirically using controlled experiments.

In the below discussion, let N annotate the number of time series to visualize simultaneously and let S be the total vertical space (in pixels) available for visualizing the data.

3.1 Evaluation Criteria

Graphical perception of multiple time series depend on a large number of factors. Below we list the most important of these factors and discuss how they can be used to classify actual visualization techniques:

- **Space management:** This factor describes whether space is "shared" or "split" between time series. Shared space is typically more amenable to comparison between series (because they are overlaid in the same space), while data in split space may be easier to perceive (less clutter).
- **Space per series:** The amount of vertical display space allocated to each individual time series. Some techniques allocate space proportional to the time series value.
- **Identity:** Distinguishing between time series can be difficult for shared space visualizations, where we often have to use graphical attributes such as color, fill pattern, or line style to convey identity. This is often more difficult for "line" techniques than for "area" techniques, where more of the color or style can be shown than for a thin line.
- **Baseline:** Comparison between time series is made easier with a "common" baseline than for an "individual" baseline, or one based on the "previous" time series displayed.
- **Visual clutter:** The clutter [8] associated with the visualization technique, especially for large values of N.

Table 1 summarizes our classification for the five line graph techniques surveyed in this paper. We discuss these techniques below.

3.2 Simple Line Graphs

The simple line graph technique we study in this paper is basically the original graph invented by William Playfair in 1786 [24]. Time is mapped to the horizontal (X) axis, and the value is mapped to the vertical (Y) axis. Displaying a time series on the graph simply consists of placing the points using the time and value mappings and connecting the points with lines. Adding multiple time series is easy—just assign each series a unique graphical property, such as a color or a line style, and then add them to the shared space (normalizing axes as needed).

Figure 2 shows an example with four time series, each assigned a unique color for identity. As seen in Table 1, simple line graphs use a common baseline in shared space, making comparisons across series simple. However, because each series is represented by a line, distinguishing identity without labels or interactive drill-down is challenging. Also, the graph may become cluttered at high values of N.

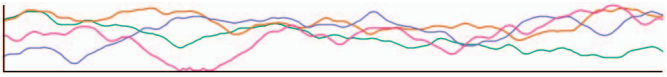


Fig. 2. Simple line graph visualization for 4 time series.

3.3 Small Multiples

Small multiples applied to line graph visualization means that instead of adding all time series to the same graph space, we split the space into individual graphs, one for each time series [28]. Because we no longer have to support several time series on the same graph space, we can turn the line into a filled area to ease identification. It is clearly important that all charts use the same axis scaling to allow for comparison across the charts.

Figure 3 shows an example scenario. Again, we assign unique colors for each of the time series, but distinguishing identity is now trivial because the graphs are separate. This also results in less visual clutter. On the other hand, this allocates less vertical resolution to each individual time series, and also makes comparison across series difficult, especially if the respective graphs are spaced far apart.

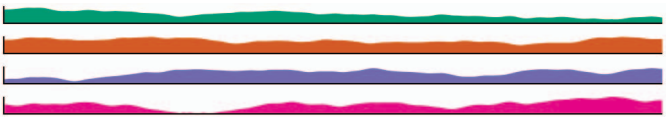


Fig. 3. Small multiples visualization for 4 time series.

3.4 Stacked Graphs

Stacked graphs have been known and used for some time, but were recently discussed and evaluated for time-series visualization by Byron and Wattenberg [2]. A stacked graph is a shared space technique where each time series in the graph is drawn sequentially, and one time series uses the value of the previous series as a baseline (the first series will use the origin of the graph as a baseline). In other words, the time series are stacked on top of each other, one at a time.

Figure 4 shows an example stacked graph for four time series. Because of this curious use of variable baselines, stacked graphs can use filled areas instead of lines to ease identification, but this comes at the cost of more complex comparison across time series [2]. It also means that the space allocation for each graph is proportional to the sum of values of all time series, so individual time series cannot use the whole vertical space despite using shared space. However, by separating the filled areas, the visual clutter can be kept reasonably low.

Fig. 4. Stacked graph visualization for 4 time series.

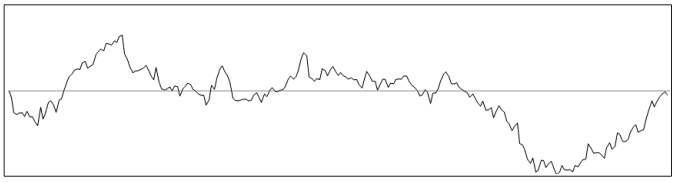
3.5 Horizon Graphs

The horizon graph time series visualization technique was originally presented by Saito et al. [26] under the name "two-tone pseudo coloring". The construction of a horizon graph is summarized in Figure 5.

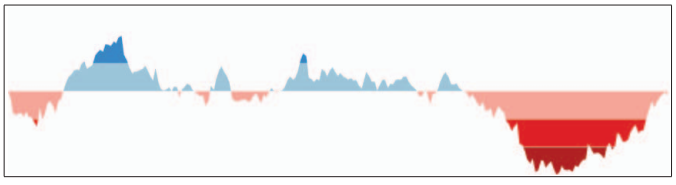
Table 1. Classification of visualization techniques for multiple time series. S is the total vertical space available for each chart, N is the number of time series to visualize, and B is the number of bands used in the horizon graph [26].

Visualization	Space management	Space per series	Identity	Baseline	Visual clutter
simple line graph [24]	shared	S	line	common	medium
small multiples [28]	split	S/N	–	common	low
stacked graph [2]	shared	proportional	area	previous	medium
horizon graph [26]	split	$S/N*2\cdot B$	–	common/individual	low
braided graph	shared	S	area	common	high

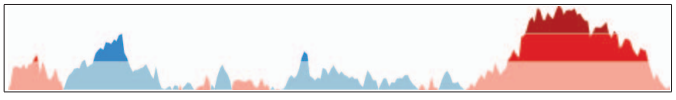
Basically, starting with a simple line graph (Figure 5(a)), we fill the area beneath the curve with a blue color for positive values, and a red color for negative (Figure 5(b)). We then split the value range into B discrete ranges, or bands, and mirror the negative values above the baseline (Figure 5(c)). In the final step, we introduce the notion of *virtual resolution* [16] by wrapping the graph space using the bands (Figure 5(d)). For multiple time series, we create one horizon graph per series; Figure 6 shows our implementation for $N = 4$ and $B = 2$.



(a) Standard line graph centered around a baseline.



(b) Color (blue is positive, red is negative) and layering.



(c) Mirroring around the baseline.

(d) Wrapping bands into a single space.

Fig. 5. Step-by-step construction of a horizon graph (adapted from [10]).

This virtual resolution and wrapping of negative values means that more space can be allocated for each individual time series despite the fact that horizon graphs use split space—instead of S/N, the space allocation for small multiples (the other split space technique), horizon graphs allocate $S/N*2\cdot B$ pixels per each graph, where B is the number of bands used (according to Heer et al. [16], $B = 2$ yields optimal performance for small vertical space allocations). Like for small multiples, the split space layout means that the visual clutter is low.

The baseline value for a horizon graph may be any value, not necessarily zero. However, one caveat with horizon graphs is that they perform best with baselines that are individual to each time series so that the ranges for the bands are utilized optimally (for example, the baseline could be the average of the vertical extents of the data, or the initial data value). On the other hand, if the baselines are not identical across all time series, it is difficult to compare them in a meaningful way. This is similar to how the value axes for other line graph techniques must be normalized for all time series being visualized.

It is also worth noting that Heer et al. [16] proposed a variant of horizon graphs where negative values were offset instead of mirrored around the baseline. This helps users perceive that negative values are, in fact, dipping and not peaking. However, Heer found no significantly better performance for these *offset graphs*, so we disregard them here.

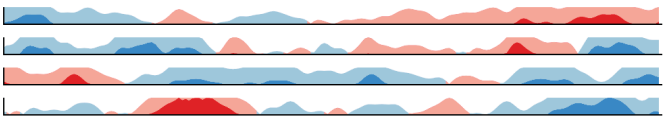


Fig. 6. Horizon graph visualization for 4 time series (2 bands).

3.6 Braided Graphs

Having come this far in our survey of static time series visualization techniques, we can note two facts: that (i) shared space layout benefits comparison across time series over split space, but that (ii) split space layout makes identification easier. The stacked graph technique tries to combine these two benefits, but the stacked baselines means that interpretation and comparison can be difficult.

The main reason why time series in simple line graphs can be difficult to identify is that the identifying graphical properties are restricted to a single (often thin) line representing the time series. This is particularly difficult if the color coding uses similar colors. If we could somehow fill the whole area beneath the line, there would be more space (i.e., more pixels) to help the viewer distinguish between different time series. However, turning the lines into filled areas means that one curve might hide the other. Even sorting the curves so that the highest-value curve is behind the lowest-value curve will not work if the two series ever change value ordering because then there exists no single depth order that avoids overlap at some point along the time axis. For example, given two series A and B, filling the area beneath the curves for these series will not work unless $A(t_i) > B(t_i)$ or $A(t_i) < B(t_i)$ for all time positions t_i in the series.

Fig. 7. Braided graph visualization for 4 time series.

Braided graphs solve the problem by identifying the *intersection points* in time where two series change value ordering, i.e., all points t_i that fulfill the condition $A(t_{i-1}) > B(t_{i-1})$ and $A(t_i) < B(t_i)$ (or vice versa). Each filled area representing a series is cut into two different segments at these intersection points, and the individual segments are then depth-sorted and drawn with the highest value segment first. This guarantees that all series segments will always be visible. Figure 8 gives a graphical view of this process.

Figure 7 shows our braided graph implementation with four concurrent time series. The technique maintains common baseline while using area curves to aid identitification. However, the resulting graph has a potentially high visual clutter for large numbers of N.

4 User Study

Our intention with this work is to study user performance for different line graph visualization techniques in the presence of multiple time series. More specifically, we are interested to see how different techniques perform under different space and cardinality constraints. In other words, is there a benefit to introducing more complex representations than simple line graphs for situations with limited vertical screen space available, or when visualizing a large number of time series?

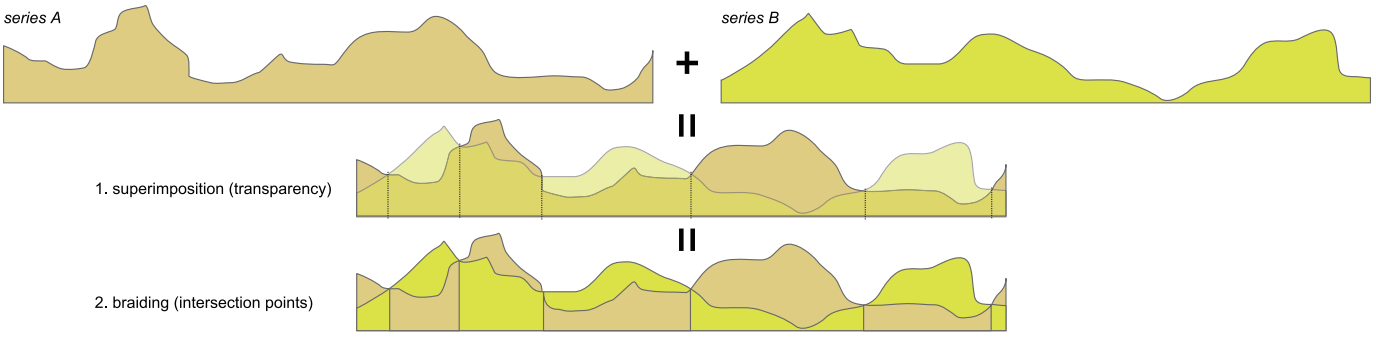


Fig. 8. Braided graphs use a common baseline but still fill the area beneath each curve. Areas are cut at points where the curves change value ordering and are depth-sorted with the highest value drawn first (in the back).

To investigate these issues, we designed a quantitative user study to measure time and correctness performance for different combinations of visualization technique, screen space, and number of time series.

4.1 Hypotheses

Our intuition is that split-space and shared-space line graph techniques have different strengths and weaknesses for tasks with different *visual span* [20]: *local*, when targets span a limited horizontal display width, and *dispersed*, when targets span the entire display width.

H1 *Shared-space techniques will perform better for tasks with local visual span.* The strength of shared space techniques is that they permit easier direct comparison across series than split-space techniques for a small visual span. Therefore, we predict that shared-space techniques (braided and simple graphs) will have better completion time for this kind of tasks.

H2 *Split-scape techniques will perform better for tasks with dispersed visual span.* Our pilot study indicated that for larger visual spans, overlap and visual clutter will become a major factor for shared-space techniques. Split-space techniques (horizon graphs and small multiples), on the other hand, avoid occlusion, and we thus predict that they will have better time performance.

H3 *Many concurrent time series will cause decreased performance.* This is the basic premise of our research: that the number of visible time series has a strong impact on the user performance of tasks that involve all series.

H4 *Small display space will cause decreased performance.* We also predict that the amount of vertical display allocated to each visualization will have a direct effect on user performance.

4.2 Participants

We recruited 16 participants (11 male, 5 female) from the student pool at our university (average age 23, median age 23). Participants were all volunteers, were paid $10 upon completing a full experimental session, had normal or corrected-to-normal vision, and not color blind (self-reporting). We also screened participants to have reasonable computer experience (which we define as using a computer more than 20 hours per week). To ensure graph reading experience, all participants were second-year or higher engineering students (average graph reading skill was self-rated as 4.5 out of 5).

4.3 Apparatus

Both experiments were conducted on a standard Dell desktop computer equipped with a mouse, a keyboard, and a 19" monitor set to 1280×1024 resolution. The experimental application was maximized on the screen. Participants only used the mouse during the actual trials.

4.4 Scenario

A single trial in our study consisted of displaying a controlled number of time series on the screen using a particular visualization type and given a particular amount of vertical screen space. Time series were labeled with letters as 'A', 'B', 'C', etc, and were deterministically assigned colors using the "Dark2" list of 8 isoluminant colors derived from [15] (so that series A was always green, B was always orange, etc). All trials used the full width of the visualization window, 900 pixels, for the time series visualization.

There was no interactive control available for any of the experimental conditions beyond inputting the answer for each trial using a dialog box showing radio buttons, one for each time series ('A', 'B', 'C', etc); the visualizations were static (i.e., no drill-down or details-on-demand). Each visualization type used a different strategy for utilizing the available vertical space as well as laying out the individual time series. These details are discussed below.

Time series data (constrained to positive values) were created for each trial using the chart generation algorithm used by Heer et al. [16]. The time series were randomized for each new trial. This was done to avoid learning effects during each experimental session, as well as to minimize the effect of any particular dataset on the final results.

4.5 Tasks

Our hypotheses are based on the premise that split-space and shared-space time series visualization techniques have different strengths and weaknesses for different tasks. Therefore, we want to include tasks that are representative for common uses of temporal visualization:

- *Maximum:* local comparison across all time series [20]);
- *Slope*: dispersed rate estimation across all time series [1]; and
- *Discrimination*: dispersed comparison of time series [27].

4.5.1 Maximum (local task)

This task required the participants to find the time series with the highest value at a specific point in time. This is a slight change from Lam et al. [20], where users were given a limited time period and not a specific point. Our pilot testing showed that this change gave the most differentiation between visual representations.

4.5.2 Slope (global task)

Assessing the global slope required users to find the time series with the highest increase during the whole displayed time period. Adapted from Beattie and Jones [1], this task can often be solved by estimating the slope of the lines in a simple line graph, but we were interested in knowing how this strategy would translate to other graph techniques.

4.5.3 Discrimination (global task)

The discrimination task, adapted from Simkin and Hastie [27], consisted of having the user determine which time series had the highest value at a point specific to each series. For example, for a trial with two time series, the user would be asked to determine whether the value of

series A at point p_A was greater than series B at point p_B. In other words, the primary task here was to find the individual values of each time series and then figure out which one was the largest one.

Discrimination points were created by evenly splitting the time dimension, adding one point per time series. Each point was indicated with a tick mark on the horizontal axis as well as a label showing the series name. The label was drawn using the corresponding color of each series. The order of discrimination points along the horizontal axis was randomly generated (but the actual positions were fixed).

4.6 Procedure

Participants were asked to fill out a demographic questionnaire prior to starting a session. They were then placed in front of the study computer and given an introduction to the goals and purpose of the experiment by the test administrator.

Trials were blocked by task type, then line graph visualization type: trials would all be of the same task, and a participant would finish all trials for a particular visualization type before moving on to another. This was an intentional design to enable participants to only deal with a single task and visualization style at a time. In addition, participants would undergo a training phase for each visualization type prior to undertaking the trials for that block. The training would consist of the administrator describing the technique (using a script) and then showing how to solve the different task for a single training trial. Participants were then asked to repeatedly solve trials on their own until they were able to correctly solve three consecutive training trials.

During the training session, the participants were given feedback as to whether their answer was correct and could choose to repeat a training trial (with new data), regardless of being correct or not. No such feedback or choice was given during the actual trials.

Participants were instructed to perform each trial as quickly as possible (mirroring a realistic overview task). They were allowed (and encouraged) to ask questions about trials and techniques during the training phase, but not during the actual trial phase. An average experimental session lasted up to two hours. To enable time for rest and general questions, each individual trial was interleaved with an intermission screen. While on this screen, there was no timer running.

4.7 Experimental Conditions

4.7.1 Visualization Type

This factor controlled the line graph visualization type used to display the time series data on the screen:

- **Simple graph (SG):** A basic line graph visualization where the whole vertical space was used for a single graph containing all time series, each drawn with colored lines.
- **Small multiples (SM):** A set of simple line graphs, one per time series, where each graph was given an equal amount of vertical screen space. Lines were drawn using their corresponding colors. Value (Y) axes used the same scale across all charts.
- **Horizon graph (HG):** A set of 2-band[1] horizon graphs, one per time series, where each graph was given an equal amount of vertical screen space. Graphs were drawn using the standard red/blue horizon color scheme. Value (Y) axes used the same scale across all charts, and the baseline reference for the graph was set to the average of the extents to fully utilize the graph's virtual resolution (equal ranges on each side of the baseline).
- **Braided graph (BG):** A single braided line graph using the whole vertical space where each time series was drawn as a filled line graph using its corresponding color.

[1]Two bands is the optimal number for small graph sizes [16].

4.7.2 Number of Time Series

Previous studies have mostly restricted themselves to only comparing two data series [16, 27], but we are interested in how this number would impact performance. Therefore, we included the number of time series to concurrently display as a factor. Pilot testing yielded three different levels: 2, 4, and 8 simultaneous time series.

4.7.3 Total Chart Size

In this work, we are mainly targeting the comparison and comprehension of multiple time series, such as studying trends over time for multiple stocks, disease outbreaks in different parts of a state, or visitors to various museums in a city. In these situations, the amount of screen space that the visualization consumes is an important measure, so we include this as a factor. Furthermore, because of the potential for summarization, aggregation, or windowing on the horizontal (time) axis, it is primarily vertical screen space that is important. Our pilot studies indicated three suitable levels for the chart size: 48, 96, and 192 pixels.

Note in the above description that different visualizations use different space management schemes; simple and braided graphs use a single graph area, whereas small multiples and horizon graphs split the available space into equal-sized subgraphs, one per each time series. In other words, given 8 time series to visualize, and 48 pixels to do it in, a simple line graph could use the full 48 pixels for all series (but force users to cope with occlusion for overlapping lines). A horizon graph, on the other hand, would only allocate $48/6 = 6$ pixels to each individual graph, but instead support a *virtual resolution* [16] (which for a 2-band horizon graph consequently is twice the space allocation per graph). These are all intrinsic properties of each visualization type, and we therefore incorporate this into the experimental design.

4.8 Study Design

We included the following factors in our study (all within-subjects):

- **Visualization type (V):** *Simple (SG)*, *SmallMultiples (SM)*, *Horizon (HG)*, *Braided (BG)*
- **Number of time series (N):** 2, 4, 8
- **Total chart size (S):** 48 px (small), 96 px (medium), 192 px (large) (spacing *not* included for split space graphs)
- **Task (T):** *Maximum*, *Slope*, *Discrimination*

We designed the study as a within-subjects factorial analysis on the above factors, yielding a $V \times N \times S \times T$ design with $4 \times 3 \times 3 \times 3 = 108$ different conditions. In addition, each condition was repeated 2 times to increase robustness, yielding a total of 216 trials per participant.

The order of tasks was not counterbalanced, but rather given in the order of simple to complex to better prepare participants for the more difficult task (Discrimination) at the end. The order of visualization types was counterbalanced between subjects using a Latin square to avoid systematic effects of practice; the order of size and number of time series was randomized within each visualization block.

With 16 participants and 216 trials per participant, the study system collected time and correctness measurements for a total of 3,456 individual trials for the whole experiment.

4.9 Study Design Choices

Prior to conducting the actual experiment, we performed an extensive pilot study involving 10 students from our university and including several candidate tasks and visualizations. We used the results from this pilot to inform the final design decisions for the study. In the spirit of Lam et al. [20], we here discuss these design decisions:

- *Synthetic data.* We follow the conventions of Heer et al. [16] and Lam et al. [20] in using synthetic data to allow control over the characteristics of each time series.
- *Tasks.* We limit our study to only three tasks—maximum, slope, and discrimination—to keep the experiment manageable in time and effort for the participants. Our choice of tasks was informed by the pilot study to be representative of general tasks for time series data, and also by similar studies.

- *Static representations.* Our evaluation only involves static visual representations and avoids animation. We base this on findings that suggest that animation gives significantly lower accuracy for trend visualization compared to static charts [25].
- *No interaction.* In the spirit of classic graphic perception experiments, we evaluated the different techniques based on their visual representation alone, disabling brushing, drill-down, zooming, and other interactive operations.
- *No stacked graphs.* We opted not to include stacked graphs after our pilot study showed—both from the performance results and from comments given by participants—that stacked graphs were mostly unsuitable for the tasks studied in this experiment. This decision is supported by the legibility issues of stacked graphs discussed by Byron and Wattenberg [2], and previous results on graphical perception by Cleveland and McGill [4].
- *Color choice.* We used the standard "Dark2" isoluminant color scale for categorical data proposed by Brewer [15] to ensure consistent graphical perception of each time series. However, horizon graphs by design use a specific color scheme as an intrinsic part of their mapping [10]: blue for positive values (i.e., above the baseline), and red for negative values (below the baseline). We do not see this as a confounding factor; each series is individually labeled, and a color legend is integrated with the display (below all graphs) to ease color identification.
- *Number of series.* We limited our study to include only up to eight concurrent time series to make trials tractable for shared-space line graph techniques. Realistic tasks often involve many more than eight time series, but this typically requires creating small multiples for the series, and so we regard this as being outside the scope of this experiment. For example, Lam et al. [20] study low data resolution techniques for these situations.

5 Results

Our experiment used two repetitions for each experimental condition. We used the average of the two repetitions for the following analysis. In the following treatment, we first discuss correctness and completion time for all tasks. We then present results for each task individually.

Correctness Table 2 summarizes the main effects on correctness for all tasks, analyzed using logistic regression. Figure 9(a) shows the correctness as a function of the visualization type and the task. Each individual task yielded the same significant effects, so we choose not to present these results in full detail.

From the table we can see that visualization type is **not** significant for correctness. This is as we expected—there should be no accuracy difference for the overview tasks included here—and it indicates that the participants were equally careful, regardless of line graph type.

Table 2. Effects of factors on correctness (logistic regression).

Task	Factors	df, den	F	p
All	Visualization type (V)	3, 45	1.40	
	Number of time series (N)	2, 30	32.95	**
	Total chart size (S)	2, 30	5.05	*
	Task (T)	2, 30	6.81	*

* = $p \leq 0.05$, ** = $p \leq 0.001$.

Completion Time The time to complete a trial was measured from when the charts were first displayed to when the user clicked the Okay button on the answer dialog. We found that the time samples violated the normality assumptions of the analysis of variance, so we analyzed the logarithm of the times (other assumptions were met).

Table 3 summarizes the main effects on completion time using a repeated-measures analysis of variance (RM-ANOVA). We analyze both all tasks combined, as well as each task individually. Figure 9(b) shows completion time as a function of both N and T, and Figure 9(c) shows time as a function of S and T. As can be seen from the table, the number of time series N has a significant main effect on completion time, but total chart size S has no significant effect.

Table 3. Effects of factors on time (ANOVA).

Task	Factors	df, den	F	p
All	Visualization type (V)	3, 45	20.49	**
	Number of time series (N)	2, 30	858.92	**
	Total chart size (S)	2, 30	1.37	
	Task (T)	2, 30	24.42	**
	V * N	6, 90	5.40	**
	V * T	6, 90	96.70	**
	N * T	4, 60	25.47	**
	V * N * T	12, 180	5.63	**
Max	Visualization type (V)	3, 45	72.95	**
	Number of time series (N)	2, 30	152.69	**
	Total chart size (S)	2, 30	0.78	
	V * N	6, 90	5.40	**
Disc	Visualization type (V)	4, 56	167.46	**
	Number of time series (N)	2, 30	611.23	**
	Total chart size (S)	2, 30	4.55	*
	V * N	6, 90	17.29	**
Slope	Visualization type (V)	4, 56	12.77	**
	Number of time series (N)	2, 30	227.65	**
	Total chart size (S)	2, 30	0.16	
	V * N	6, 90	0.63	

* = $p \leq 0.05$, ** = $p \leq 0.001$.

Completion times for each task as a function of the visualization type are shown in Figure 10. We analyzed this effect using a Tukey HSD test; Figure 11 shows pairwise relations for all tasks ($p < .05$).

6 Informal Follow-up User Study

Our results are limited to up to eight concurrent time series, but many realistic tasks involve substantially higher numbers of series. To study this effect closer, we designed an informal follow-up experiment where we investigated performance for higher numbers of time series.

We wanted experienced visualization users because of the high numbers of time series we were investigating. For this reason, we recruited four unpaid participants (all male) from another visualization laboratory at our university. As opposed to attempting to get statistically significant results with such a small participant pool, we set out only to capture completion time data for comparison in graphical form. We chose the discrimination task because it was the most taxing task (by virtue of having the highest average completion time).

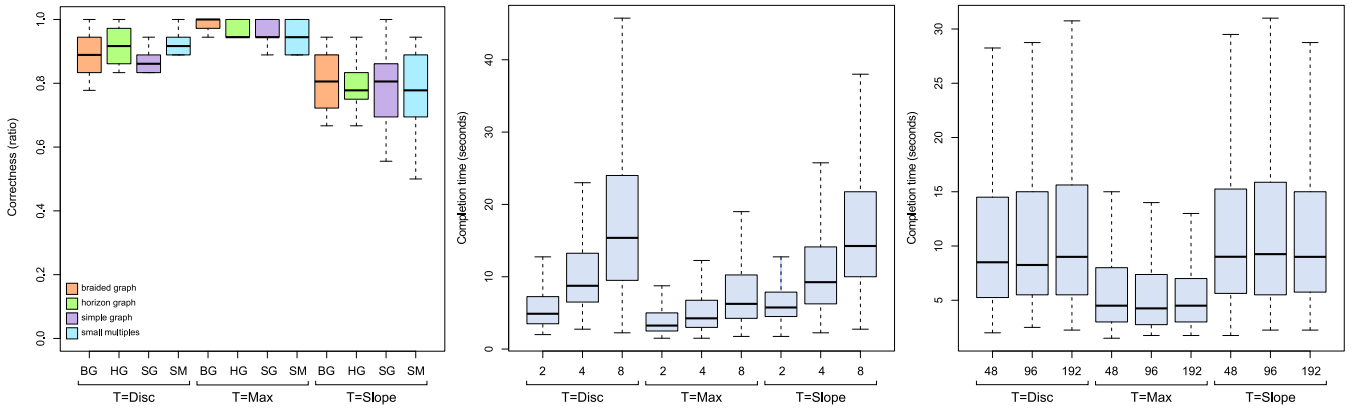
Because our main study did not show a significant effect for total chart size S given our parameters (down to 6 pixels per chart for split space graphs), we opted to fix the chart size to 6 pixels per time series, and instead allocated total chart size to be proportional to the number of time series (i.e., $S = N \times 6$ pixels). However, as for the number of time series N, we included 2, 4, 8, 10, 12, 14, and 16 concurrent series. Color choice is a problem for this many series—we based our colors on the 12-color qualitative color scheme proposed by Brewer [15].

With 4 participants and 28 trials for each, we collected data for 112 individual trials. Figure 12 plots time for V and N, where we can note that split-space techniques seem to scale better for higher values of N.

7 Discussion

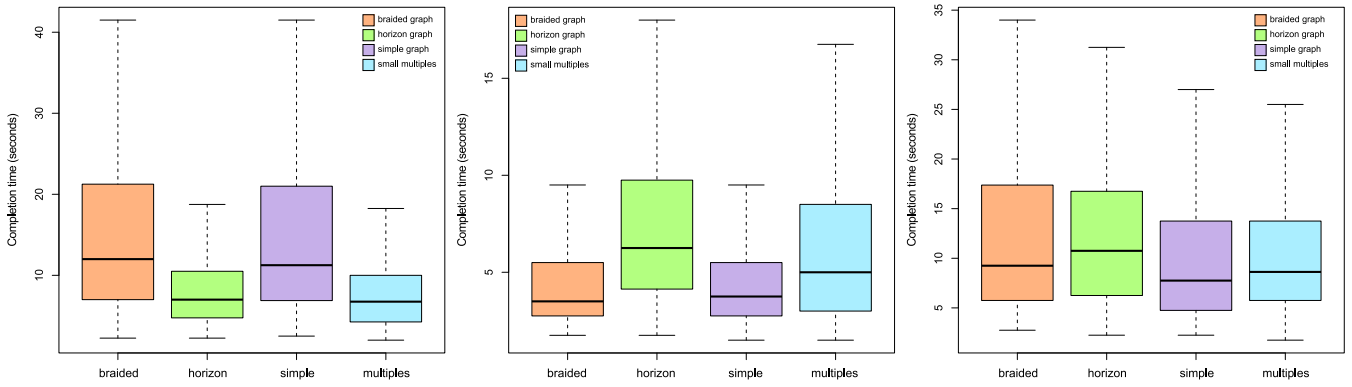
We can summarize our findings as follows:

- Shared-space techniques (SG and BG) were faster than split-space techniques for the local Maximum task (confirming H1);
- Split-space techniques (SM and HG) were faster than shared-space techniques for the dispersed Discrimination task (H2);
- The Slope task, with dispersed visual span, was special—SM and SG were fastest here;



(a) Correctness ($V \times T$). (b) Completion time ($N \times T$). (c) Completion time ($S \times T$).

Fig. 9. Correctness and completion time plots for the overall study (all tasks combined).



(a) Discrimination task. (b) Maximum task. (c) Slope task.

Fig. 10. Completion time as function of visualization type V for each task.

- Higher numbers of concurrent time series caused decreased correctness and increased completion time (confirming H3); and
- Decreased display space allocation had a negative impact on correctness, but had little effect on time (partially confirming H4).

7.1 Explaining the Results

The results from this study confirm our intuition that different visual representations have different strengths and weaknesses—a perhaps not surprising result. However, these findings can be useful as guidelines for designers looking to build visualization systems based on line graph representations. We believe that superimposed (shared-space) techniques excel at comparisons with a local visual span because these techniques have the benefit that the comparison is done in the same space. Juxtaposed (split-space) techniques require the user's gaze to travel vertically between different screen regions for comparison.

However, for dispersed visual span tasks, we introduce an additional horizontal travel distance necessary for the comparison. In a shared-space representation, clutter and overlap between time series is inevitable and increases as the number of series increases, and this makes following individual lines over a distance difficult. Split-space techniques, on the other hand, do not have this problem because they disambiguate each series using vertical distance, and managing large visual spans becomes easier. For this reason, split-space techniques are more robust against high numbers of concurrent series for dispersed tasks. This is particularly clear from the results from the follow-up study, where the overlap and clutter is significant at high values of N.

Having said that, the pairwise comparisons in Figure 11 do show that for these three tasks at least, simple graphs (SG) and small multiples (SM) end up at the top more often than horizon (HG) and braided

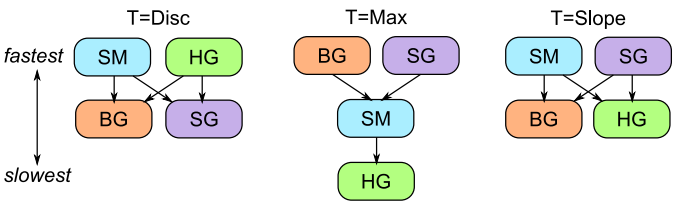


Fig. 11. Pairwise relations for completion time for all three tasks. Arrows indicate that the source is significantly faster than the destination. Techniques on the same level have no significant completion time difference.

graphs (BG). This indicates that these former two representations are more robust towards different task types than the latter two.

For shared-space techniques, the clutter problem is compounded by the limited color acuity of the human visual system—it is plain difficult to differentiate between eight (or sixteen, for the follow-up study) unique colors, particularly when each color is represented by a thin line. Our braided graph was designed to improve color perception by filling the area under each curve. However, while braided graphs were better than both small multiples and horizon graphs for the Maximum task, it was only equivalent and never better than the simple graph. Therefore, it is not clear this technique fully reached its design goals.

One surprising outcome of our study is that the total display chart size had no significant effect on completion time. In other words, it appears that participants did not become slower when the chart size decreased. We explain this by that participants were asked to solve each trial as quickly as possible and thus tended to use the same amount of

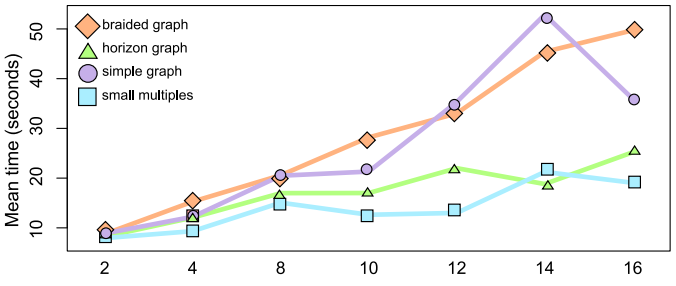


Fig. 12. Interaction of $V \times N$ for completion time for the follow-up study.

time regardless of chart size. This was manifested in decreased correctness instead, on which chart size **did** have a significant effect; in other words, a classic time/accuracy trade-off.

Finally, horizon graphs were significantly slower than all other techniques for the maximum task, the quickest (and presumably easiest) task. We speculate that this is because horizon graphs are not really preattentively perceivable, but require some cognitive effort to decode.

7.2 Generalizing the Results

The design of this experiment is broad enough that our results should be generalizable to other settings and tasks. The results are also consistent with previous work on graphical perception for statistical data graphics [1, 16, 27]. In particular, according to the reasoning above, simple graphs and small multiples in general appear to be most versatile of the visual representations tested in this work.

On the other hand, our study, while fairly comprehensive, only includes three of the many potential tasks users may want to perform on a time series visualization. In particular, our chosen tasks are designed for quick overview and not detailed drill-down, whereas the virtual resolutions of horizon graphs and the shared space of superimposed techniques may lend themselves to this kind of detail task. Conducting an accuracy-based evaluation is left for future studies on this topic.

Finally, as our results show, there clearly is a limit to how many time series are practical to display simultaneously before the visual clutter becomes too high for effectively perceiving individual series. For higher values of N, we must instead turn to methods involving compact visual representations [20], temporal queries [17], or aggregation [9]. In other words, it is clear that visualizing individual temporal data series is not generalizable to any number of concurrent series.

8 Conclusion and Future Work

We have presented results from a user study on the graphical perception of multiple simultaneous time series. Our results show that superimposed line graph techniques work best for local tasks, whereas juxtaposed techniques work best for dispersed ones.

In our future endeavors, we would like to study graphical perception for massive numbers of time series. Virtually all data can be analyzed with respect to time, and it is not unusual to have datasets consisting of hundreds, if not thousands, of time series. One promising approach is to aggregate time series (e.g., [29]) and visualize the aggregates.

Acknowledgments

This research was partly funded by Google, Inc. under the project "Multi-Focus Interaction for Time-Series Visualization."

References

[1] V. Beattie and M. J. Jones. The impact of graph slope on rate of change judgments in corporate reports. *ABACUS*, 38(2):177–199, 2002.

[2] L. Byron and M. Wattenberg. Stacked graphs — geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, Nov./Dec. 2008.

[3] W. S. Cleveland. *Visualizing Data*. Hobart Press, Summit, NJ, 1994.

[4] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, Sept. 1984.

[5] F. E. Croxton and H. Stein. Graphic comparisons by bars, squares, circles, and cubes. *Journal of the American Statistical Association*, 27(177):54–60, 1932.

[6] F. E. Croxton and R. E. Stryker. Bar charts versus circle diagrams. *Journal of the American Statistical Association*, 22(160):473–482, 1927.

[7] W. C. Eells. The relative merits of circles and bars for representing component parts. *Journal of the American Statistical Association*, 21(154):119–132, 1926.

[8] G. Ellis and A. J. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, 2007.

[9] N. Elmqvist and J.-D. Fekete. Hierarchical aggregation for information visualization: Overview, techniques and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 2010. to appear.

[10] S. Few. Time on the horizon. *Visual Business Intelligence Newsletter*, June/July 2008. `http://www.perceptualedge.com/articles/visual_business_intelligence/time_on_the_horizon.pdf`.

[11] P. J. FitzPatrick. Leading British statisticians of the Nineteenth Century. *Journal of the American Statistical Association*, 55(289):38–70, Mar. 1960.

[12] M. Friendly. A brief history of data visualization. *Handbook of Computational Statistics: Data Visualization*, III, 2007.

[13] D. J. Gillan. A componential model of human interaction with graphical displays. *ACM SIGCHI Bulletin*, 25(3):64–66, 1993.

[14] D. J. Gillan and A. B. Callahan. Componential model of human interaction with graphs: VI. cognitive engineering of pie graphs. *Human Factors*, 42(4):556–591, 2000.

[15] M. A. Harrower and C. A. Brewer. ColorBrewer.org: An online tool for selecting color schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[16] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualization. In *Proceedings of the ACM CHI 2009 Conference on Human Factors in Computing Systems*, pages 1303–1312, 2009.

[17] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.

[18] W. Huang, P. Eades, and S.-H. Hong. Beyond time and error: a cognitive approach to the evaluation of graph drawings. In *Proceedings of BELIV*, pages 1–8, 2008.

[19] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[20] H. Lam, T. Munzner, and R. Kincaid. Overview use in multiple visual information resolution interfaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1278–1285, 2007.

[21] G. L. Lohse. A cognitive model for understanding graphical perception. *Human-Computer Interaction*, 8(4):353–388, 1993.

[22] J. Lohse. A cognitive model for the perception and understanding of graphs. In *Proceedings of the ACM CHI'91 Conference on Human Factors in Computing Systems*, pages 137–144, 1991.

[23] L. V. Peterson and W. Schramm. How accurately are different kinds of graphs read? *Educational Technology Research and Development*, 2(3):178–189, June 1954.

[24] W. Playfair. The commercial and political atlas: Representing, by means of stained copper-plate charts, the progress of the commerce, revenues, expenditure and debts of England during the whole of the Eighteenth Century, 1786.

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

[26] T. Saito, H. N. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda. Two-tone pseudo coloring: Compact visualization for one-dimensional data. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 173–180, 2005.

[27] D. Simkin and R. Hastie. An information-processing analysis of graph perception. *Journal of the American Statistical Association*, 82(398):454–465, June 1987.

[28] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut, 1983.

[29] J. J. van Wijk and E. R. van Selow. Cluster and calendar based visualization of time series data. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 4–9, 1999.