Compression and Shifting to Reduce Occlusion in Multiple Short Time Series

Maxime Dumas*
Croesus Finansoft

Michael J. McGuffin†
École de technologie supérieure

Patrick Chassé‡
Croesus Finansoft

Abstract

Visualization of multiple time series often suffers from overplotting, making it difficult to read the values of curves that are hidden by other curves. We present techniques for horizontally displacing the endpoints of line segments in discrete time series data that (1) enable the depiction of subsets of data and (2) reduce occlusion of endpoints. Because endpoints are not displaced vertically, their y values can still be read and compared. Unlike small multiples, our techniques do not move points with the same x or y values far from each other, making some comparison tasks easier. We present three novel techniques: compressed-superposed, compressed-juxtaposed, and shifted layers. One limitation of our techniques is that they work best when there is only a small number of x values being visualized, and additionally, one of them (compressed-superposed) modifies the slopes of curves in a way that makes the slopes incomparable. Our experimental comparison with three status quo techniques (conventional overlaid, vertically-stacked small multiples, and horizontally-stacked small multiples) shows that our proposed techniques are competitive with status quo techniques and in some cases superior.

Index Terms: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

1 Introduction

A common problem with time-series visualizations [2] is the occlusion of curves when visualizing many time-series (Figure 1A). Partitioning the data into multiple panes or windows (or "small multiples" [38]) would reduce the amount of data shown in each pane, but would also reduce the amount of space available for each time-series and prevent both axes from being shared across all data curves. This means that the user's eyes must move a longer distance to compare points with either the same x or y values.

Another way to depict distinct subsets of curves is to somehow display multiple *layers* of data in the visualization. In a 2D visualization of time-series, layers might be depicted using color coding (e.g., one subset of data curves in red, another in green), or using different curve thicknesses, or different levels of blur (one subset of curves blurred, another subset shown in sharp focus [27]), or by using movement (one subset of curves not moving, another subset vibrating [3]). Each of these approaches, however, either does not reduce occlusion of data, or does reduce the spatial resolution available for the visualization, or both. There are also interactive techniques available for temporarily reducing occlusion and/or making layers more apparent, however these require a time investment on the part of the user to move a cursor or otherwise provide input.

Jitter, i.e., the repositioning of points or endpoints of line segment by small (sometimes random) displacements, is a technique

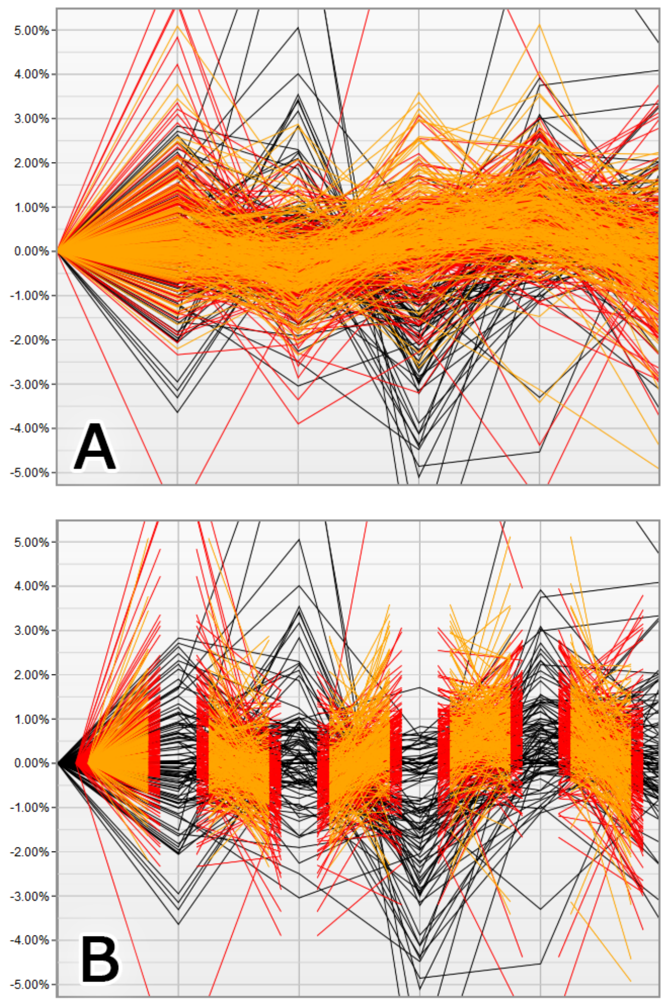

Figure 1: *A:* in this conventional overlaid view, time series curves are colored by group: black, red, or orange. *B:* with these compressed-superposed layers, the red and orange curves have been compressed horizontally to reduce occlusion of endpoints of line segments.

sometimes used to reduce occlusion in scatterplots (Figure 1 of [5]) and parallel coordinates (Figure 4 of [13]). Jitter could plausibly be used in time series visualizations, to avoid overplotting where multiple time series achieve the same value. However, displacements in the vertical direction could make it impossible for the user to accurately read the values of time series data. Small *horizontal* displacements, however, would avoid such a problem.

We observe that in discrete time series, if there is only a small number of x values to display (less than 20), the space between consecutive time steps may be underutilized. We investigate the use of horizontal offsets of different types to make subsets (layers)

*e-mail: maxime.dumas.1@ens.etsmtl.ca
†e-mail: michael.mcguffin@etsmtl.ca
‡e-mail: patrick.chasse@croesus.com

of curves more apparent. Figure 1B shows one of three approaches we have developed: To make layers of curves appear on top of a lower layer, each interval is compressed horizontally, resulting in a compressed-superposed layer. This compression changes the apparent slopes of the curve segments, but does not change the vertical positions of endpoints (meaning that values can still be compared within and between groups), and reduces occlusion with no loss in the spatial resolution in y. It also allows the same x- and y-axes to be shared by all groups of curves, meaning that comparing points with similar x or y values will still involve small eye movements. Layer compression can also be used as a focus+context technique [7], where the context is given by a background layer, and the focus is shown by a foreground (compressed) layer, as shown on Figure 2. This technique enables the display of additional contextual information without creating occlusion of the original values.

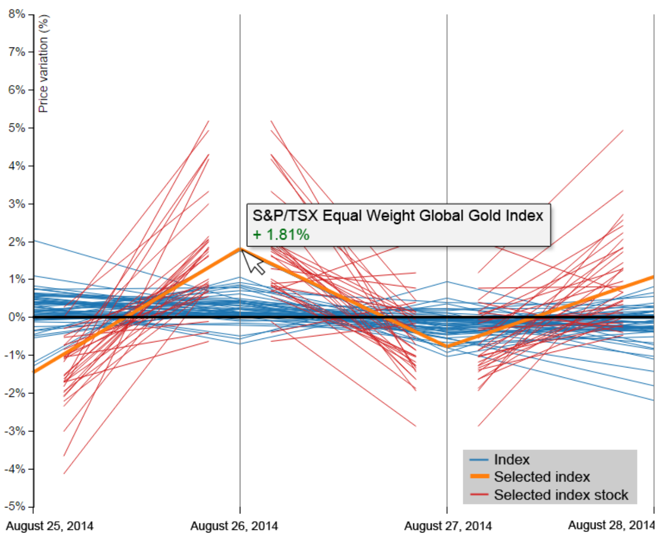

Figure 2: A compressed-superposed layer provides details of financial data. Blue curves show performance of major Canadian S&P/TSX indexes over four days. The user has selected a single index (highlighted in orange), for which the constituent companies are shown in a compressed layer (in red).

As presented later, we also propose two other techniques that horizontally displace endpoints: compressed-juxtaposed layers, and shifted layers. Table 1 compares techniques for separating subsets of curves, with our proposed techniques appearing as the three right-most columns. Our techniques are the only ones to satisfy all of the first five criteria, warranting further investigation. Note that our proposed compressed and shifted layers may not be best as a *replacement* for other techniques, but might be combined synergistically with other techniques using interaction.

In this work, we analyze design choices related to horizontal displacement of end points for the purpose of displaying multiple layers, yielding three novel techniques: two kinds of compressed layers (superposed and juxtaposed), and shifted layers. We also report an experimental comparison of our three techniques with three status quo techniques (conventional overlaid, horizontally-stacked small multiples, and vertically-stacked small multiples), over four tasks and visualizing hundreds of curves at a time. Results show that our techniques are competitive with the status quo, and in some cases superior. In two of the tested tasks (Tasks 2 and 3), users were asked to estimate y ranges. In terms of the accuracy of their estimations (denoted Δy), our three proposed techniques were never significantly worse than other techniques, and one of them (compressed-juxtaposed) was significantly better in Task 2 while another (compressed-superposed) was significantly better in Task 3. In terms of the time required to complete Tasks 2, 3, and 4, two of our techniques (compressed-juxtaposed and shifted) were never significantly worse than other techniques. Compressed-juxtaposed and compressed-superposed were also given the highest subjective ratings in 4 out of 5 criteria. As an overall summary, compressed-superposed was better for perceiving entire groups of curves (Tasks 3 and 4) whereas compressed-juxtaposed was better for perceiving individual curves (Task 2).

An interactive demonstration of all six techniques, and a video of all six, are available at http://www.michaelmcguffin.com/research/layers/

2 Related Work

Visualization of time series data [2] is a challenging topic, partly due to the long time spans involved, and due to the large number of curves that may need to be visualized, causing occlusion. Researchers have proposed collapsing curves to single rows of pixels [25] to eliminate occlusion, using interactive filtering [20] to reduce the number of curves to display, as well as other feature-rich interactive systems [24, 41, 42]. Our proposed technique is not interactive, and could be combined with ideas in these other works. Also, because we do not collapse curves into rows of pixels as in Kincaid and Lam [25], the original shape of the curves remains visible.

Recent innovations in time series visualization include horizon graphs [34, 18] and braided graphs [22]. These techniques can improve the legibility and presentation of relatively small collections of curves. For example, horizon graphs reduce the vertical space necessary to show a single curve, thereby enabling small multiples to use space more efficiently. In contrast, our work enables the visualization and comparison of hundreds of curves sharing the same axes.

Our work is based on the idea of partitioning a set of curves into subsets, and showing each subset on a different layer. Layers are a common theme in information visualization, and can be depicted in many ways. Layers can be embedded in a 3D space [8, 37] or can even be physically realized with special display equipment [1]. We seek an effective way of showing layers that will make individual subsets of curves easier to read, without requiring additional interaction, or the complexity of 3D navigation, or specialized hardware.

Macroscope [28, 29], transparent menus [15, 16], Multiblending [4], and free-space transparency [21] are all ways of using transparency to show multiple images at the same time. This general idea could also be applied to time-series visualizations, but could be confusing when curves are dense. The use of transparency can also come at the cost of some spatial resolution, or creates many possible mixtures of alpha-blended colors that are difficult to distinguish (for example, 4 different original colors can produce at least $4^2 - 1 = 15$ different mixtures, not counting different orderings of colors).

Several interactive techniques can be used to more effectively browse through time-series data or other visualizations involving thin curves. These include user-initiated deformation [40, 33], user-initiated vibration [32, 39], interactive displacement of layers with a 2.5D orthographic projection [32, 30], lens-based techniques [20, 11, 31, 9], as well as more general techniques like search and filtering. All of these can make visualizations more effective, but always at the cost of some investment in input on the part of the user. Because our proposed compressed layers have no required interactive techniques, they could be combined with many of these previous techniques to make them more powerful. The scope of our current work, however, is to study compressed layers in their static form.

The visualization of financial data has been a topic of growing interest, as shown by several recent works [23, 36, 43, 35, 10, 26], some of which have proposed techniques for aggregating curves, using small multiples, or using color coding to enable the comparison of a single time series curve with a market's performance. Our work is complementary to this previous work, as we propose a way of making multiple groups of curves visible simultaneously.

More generally, our work can be seen as another way of reducing occlusion in data visualizations [12]. In particular, our techniques might be applied to parallel coordinate plots [19], which have also seen techniques applied to reduce occlusion [11], and recently have even been integrated with time series [14]. However, applying our techniques to parallel coordinates is beyond the scope of the current work.

3 Designs to Display Layers

The partitioning of data into layers is particularly useful for tasks where the user must distinguish different types of curves. As shown in Figure 1A, the use of color is insufficient to clearly separate overlapping curves. Our techniques allow each group to be distinguished, with no occlusion of the endpoints of line segments.

3.1 Application Domain

Our research was motivated by challenges in visualizing financial data. We identified several situations where datasets containing many financial time series must be compared. For example, it might be useful to compare the historical prices of securities between multiple industries, sectors, or asset classes to visually detect trends and anomalies. Such comparisons might be used to create a more diversified portfolio with uncorrelated assets from different markets. In our visualizations, each layer might correspond to one industry or sector (Figure 3). We solicited feedback from workers in the financial industry to inform our design choices and also to later inform the choice of tasks used for experimental comparison.

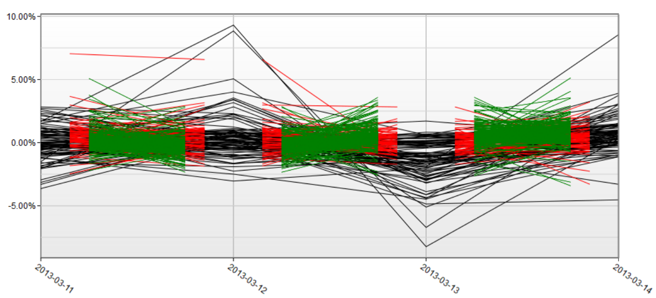

Figure 3: Daily changes in stock values in 3 sectors of the NASDAQ, with each sector on a different layer.

3.2 Compressed-Superposed vs Masked-Superposed

Consider the consecutive time steps x_i and x_{i+1} in a time series. In a visualization of the time series, there is normally no data to show between these values, and the consecutive points (x_i, y), (x_{i+1}, y) are connected by a straight line segment. This segment could be transformed, in theory, with no loss of information, as long as the user can still ascertain the original x and y values. We first consider two transformations that might allow the original values to be read, while also reducing occlusion between layers.

First, each line segment could be *compressed* (Figure 4A) horizontally. This changes the horizontal positions of the endpoints, and also changes the apparent slope. However, because it is the vertical position of each endpoint that shows its y value, these values can still be read with no occlusion from other layers.

Second, each line segment could be *masked* (Figure 4B). This has the advantage of preserving the original slope, but erases the original endpoints, requiring the user to imagine where a vertical

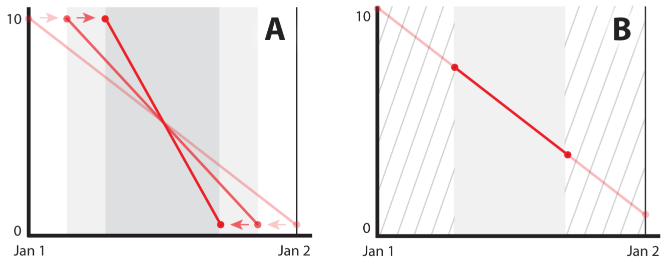

Figure 4: Approaches for separating the line segments of time series curves into layers: (*A*) compression and (*B*) masking.

line would intersect the segment if it were extended. Such masking is comparable to the effect of window blinds, picket fences, or bezels in tiled displays, and for certain tasks has been shown to not be detrimental [6].

We implemented both the compressed and masked approaches for showing layers. In both cases, layers were superposed. Our informal testing found that masked-superposed time series were much more difficult to interpret, since it was typically unclear where the imaginary intersection points were, especially with line segments having a large slope, and especially with many layers present (causing incrementally more masking of the top-most layer).

We showed both approaches to 5 financial experts. With the compressed-superposed approach, all of them found the distortion of slopes to be initially somewhat confusing, but after an explanation were comfortable with it. In contrast, all 5 of them found the masked-superposed approach to be much too difficult to interpret.

We decided that it was more important to faithfully show the original y values of the time series than to preserve slopes. This is partly justified by the fact that, to estimate the rate of change of a curve over a range of x values, a user can imagine a secant line (i.e., a line of average slope) passing through the first and last points in the series, and need not pay attention to the slopes of line segments between consecutive x values. We therefore dropped the masked approach and retained the compressed-superposed approach (shown in Figures 1B, 2, 3, 6B) for further evaluation.

We also considered whether compression could be scaled up to large numbers of x values. Figure 5 sketches out a possible way to do this. Such a view might make sense for visualizing daily stock prices grouped by financial sector: each compressed interval might be one day of activity, thus reducing occlusion of the prices at the start and end of each day, and intermediate points could show "intraday" prices (these intraday values, however, would still suffer from just as much occlusion as without compression). It is unclear how widely applicable this approach is; it seems that compression is still best used when there are only a small number of x values under consideration.

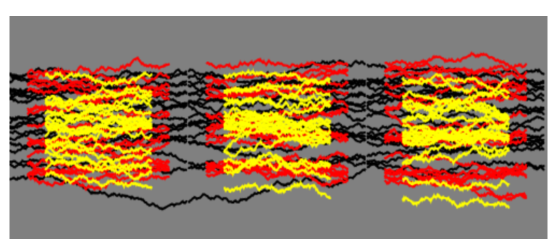

Figure 5: Compressed-superposed layers where there are three intervals along x that are compressed, but many intermediate x values within each interval.

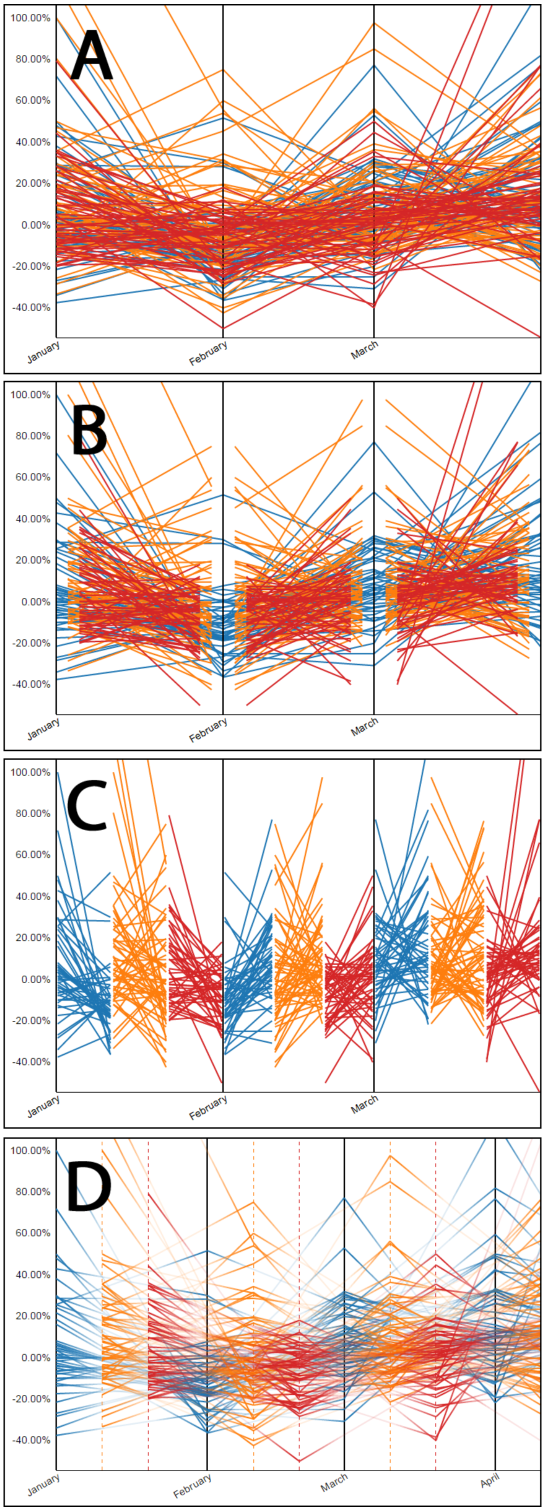

Figure 6: Layout styles for 3 layers (red, orange and blue). *A:* conventional overlaid with only color coding. *B:* compressed-superposed. *C:* compressed-juxtaposed. *D:* shifted layers.

3.3 Compressed-Juxtaposed and Shifted

We subsequently developed two more designs involving horizontal displacement of endpoints. After compressing layers, rather than superposing them, the layers could be juxtaposed (Figure 6C). This has the disadvantage of requiring each compressed interval to be more narrow than with superposition, but the benefit that now slopes *can* be compared since the segments of all layers have undergone the same (non-uniform) scaling transformation. Note that this also makes the slopes appear more extreme, however small multiples (Figure 7) also make slopes appear either more extreme or closer to zero.

Another possibility is to shift entire layers of curves to the right without introducing any gaps nor changes to slopes. In Figure 6D, the orange layer has been shifted to the right by one third of a month, and the red layer has been shifted two thirds. Shifted endpoints are always associated with the x value of the previous black vertical line, marking the month. In this shifted layer view, we smoothly vary the opacity of each curve, so that the curve is fully opaque at its segment endpoints, but transparent in between these to allow the endpoints of other layers to show through. In the experiment we report later on, the opacity of shifted curves varied smoothly between 100% at endpoints to 10% midway between them.

3.4 Comparison and Limitations

Table 1 gives a theoretical comparison of competing techniques with our proposed designs. As seen in the first five rows of the table, our techniques reduce occlusion without sacrificing spatial resolution in y, and (unlike small multiples) keep the x and y values of all groups close together, which could theoretically ease certain kinds of comparisons. For example, with horizontally-stacked small multiples, a given x value in different groups will correspond to different horizontal positions, requiring larger eye movements than with our techniques, which may make certain comparisons more difficult.

Disadvantages with our techniques are seen in the last four rows. Compressed-superposed is the only technique that applies a different scale factor to each layer, making slopes incomparable. Compressed-juxtaposed (along with small multiples) modifies the angles of curves, but this is done by applying the *same* scale factor to all layers, meaning that slopes can still be compared. Both of our compression techniques introduce discontinuous gaps in the x direction, which violates the gestalt principle of continuity, making it difficult to perceive each curve as a single object. It is also doubtful that our techniques can scale to large numbers of x values, though Figure 5 shows that it might be possible in limited cases involving compression.

Although not mentioned in the table, *none* of the techniques (previous or new) scale up to a large number of layers. Our proposed techniques work best with 2 to 4 layers.

4 Experimental Study

Conventional layers suffer from overplotting, sometimes making it impossible to perceive the locations of endpoints or even hiding entire curves. Small multiples, on the other hand, have the disadvantage of not sharing either x- or y-axes, which might increase the amount of eye motions when comparing curves of different groups. Our proposed techniques based on compression and shifting are not familiar to users, and may incur costs beyond the theoretical ones identified in Table 1. An experiment is thus required.

Six techniques were compared (Figures 6 and 7): three status quo techniques (conventional overlaid, horizontally-stacked small multiples (HSM), and vertically-stacked small multiples (VSM)) and our three proposed techniques (compressed-superposed, compressed-juxtaposed, and shifted).

4.1 Tasks

From a theoretical perspective, tasks with time series data could involve any of the following aspects: comparisons of x values or

	Conventional overlaid (color only)	Transparency	Blur	Movement	Random Jitter	Small Multiples		Compressed -Superposed	Compressed -Juxtaposed	Shifted Layers
						Horizontally Stacked	Vertically Stacked			
Preserves spatial resolution in y	Yes	Maybe	No	No	No	Yes	No	Yes	Yes	Yes
Reduces occlusion of endpoints of segments	No	Maybe	No	Maybe	Yes	Yes	Yes	Yes	Yes	Yes
Enables inter-group comparisons of y-values (except where occluded)	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
The same x values of different groups remain close (i.e., a single shared x-axis)	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
The same y values of different groups remain close (i.e., a single shared y-axis)	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
All groups share the same x/y scale for easier comparison of slopes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Preserves angles (i.e., no non-uniform scaling)	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes
No discontinuous gaps along x	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes
Scales to large number of x values	Yes	Yes	Yes	Yes	No	Yes	Yes	Maybe	Maybe	No

Table 1: Each row is phrased as a positive quality, so that satisfying more criteria (shown in green) is desirable. The three right-most columns show that our proposed techniques are the only ones satisfying all of the first five rows. This comes at the cost, however, of disadvantages in the lower rows.

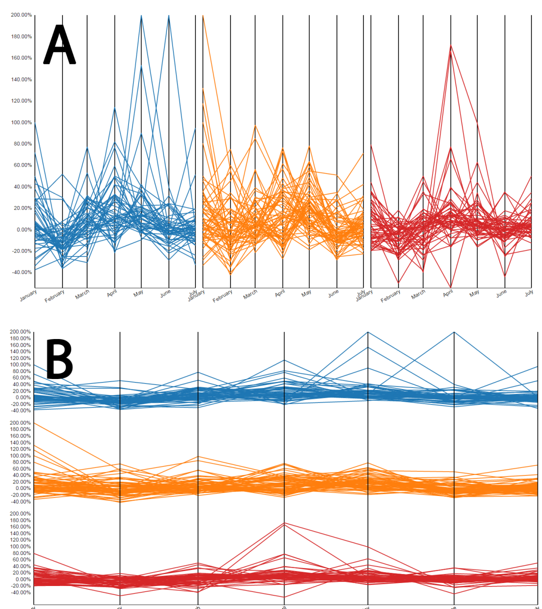

Figure 7: *A:* Horizontally-stacked Small Multiples (HSM). *B:* Vertically-stacked Small Multiples (VSM).

y values, single values or ranges of values (in both x and y), rates of change (i.e., slopes), and either individual curves or groups of curves. We chose four tasks that (1) touch on each of the theoretical aspects just mentioned, and that also (2) correspond to real-world tasks in our target application domain of financial data analysis.

Task 1 ("compare highlighted curves") asked the user to compare y values at a given x value, with no significant occlusion. One curve in each group was highlighted (by drawing it thicker and with a white outline surrounding it, and drawing it on top of the other curves). The three highlighted curves were not occluded, but there were hundreds of other curves displayed behind them. The user was asked two questions: *During the month of [month], which of the three highlighted curves performed the best, and which one performed the worst?* The user answered the first question by selecting one of 3 radio buttons, and answered the second by selecting one among another set of 3 radio buttons.

Task 2 ("compare maximal curves") asked the user to compare y values at a given x value, and to estimate the magnitude of a range in y. No curves were highlighted, and the curves identified in the questions were partially occluded. However, the questions in this task concerned curves of maximal y value at a given x value, i.e., curves that are locally outliers and thus suffer from less occlusion. The user was asked: *In which group do we find the curve that performed the best during the month of [month]? Next, estimate the vertical range covered by the 3 top performing curves (one from each group) during that month.* For example, if one group's top curve had a y value of 100%, another group's top curve had a value of 90%, and another was 105%, the range covered would be $105\%-90\%=15\%$. The user answered the first question by selecting one of 3 radio buttons, and answered the second by entering the numeric range in a text field. The difference between the y range entered and the true range is called Δy in our results, providing a measure of accuracy.

Task 3 ("compare groups") asked the user to compare ranges of y values at a given x value, and to estimate the magnitude of a range in y. In this task, no curves were highlighted, and the user was asked: *Which group exhibits the lowest volatility (smallest vertical range) during the month of [month]? Estimate the vertical range covered by that group in that month.* Notice that the group in these questions has the *smallest* vertical range, meaning that it tends to suffer more from occlusion than the other groups at the given month. The user answered the first question by selecting one of 3 radio buttons, and answered the second by entering the numeric range in a text field. Again, our results report the Δy measure of accuracy of the range entered by the user.

Task 4 ("compare slopes") required the user to compare slopes over a range of x values. In this task, no curves were highlighted, and the user was asked: *Which group increased the most over the time range from [month1] to [month2]?* This required the user to look at each group as an aggregate of curves, and estimate the average slope of the group over the given time span to select the one with the greatest slope. The user answered by selecting one of 3 radio buttons. In 50% of the trials for this task, the range in x was only one month long (e.g., "from March to April"). In 25% of the trials, the range was three months long, and in 25% of the trials, the range was five months long.

In all of these tasks, the "groups" can be thought of as representing financial sectors, such as health care, technology, and energy; and the curves can be thought of as equities such as stocks. Tasks 1, 2 and 3 displayed curves showing the percentage of change in value from one month to the next for each curve. These percentages could be positive or negative from one month to the next, and tended to cluster around $y = 0$. Task 4, however, showed cumulative curves, which could represent the current value of various equities from month to month, meaning that the slope of these curves was meaningful, and also allowing the curves to stray more and more from the $y = 0$ baseline as we move right along the x axis.

Notice that Tasks 1 and 2 required comparison of individual curves, whereas Tasks 3 and 4 required comparing entire groups of curves.

All four tasks have analogs in financial data analysis: Task 1 asks the user to compare three particular equities that have been somehow selected *a priori*, and are thus highlighted visually. Task 2 asks the user to identify the equities which performed the best (and are therefore not yet selected nor highlighted) and also estimate how much better the best performed compared to the worst of the three. Task 3 asks which sector is the least volatile, which is important for reducing financial risk. Task 4 asks which sector performed the best over a given period.

Notice that none of our tasks involve intra-group comparisons, nor questions about a single group. This is because such tasks are probably best done by first filtering out or hiding the other groups, to reduce the number of curves and complexity of layout, in which case there is only one group left to display, and there are no interesting differences between the various layout techniques we consider if there is only one group to display.

4.2 Hypotheses

Task 1 asks the user to compare 3 curves that are highlighted and not occluded, but that are drawn on top of hundreds of other curves in the background. Such background noise might distract the user, however we expect the conventional overlaid technique to perform best at this task, thanks to its familiarity and other strengths and the lack of occlusion in the curves of interest. We posit hypothesis **H1:** the conventional overlaid technique will yield the best performance in Task 1.

Task 2 involves comparing y values at a given x value for curves that are only partially occluded. Because small multiples cause points of different groups with the same x and y values to appear far from each other, we expect **H2:** HSM and VSM will yield the worst performance in Task 2.

Task 3 asks the user to identify the group with the most narrow vertical range, and to estimate the magnitude of this range. Because such a group is more likely to be occluded than the other groups, we posit **H3:** the conventional overlaid technique will yield the worst performance in Task 3.

Task 4 involves comparing average slopes over a range of 1, 3, or 5 months. Because compressed-superposed is the only technique that scales each layer by a different factor, making slopes incomparable, we expect **H4:** compressed-superposed will give the worst performance in Task 4, especially over ranges of 1 month.

4.3 Data sets

The sets of curves used in the experiment were obtained by randomly choosing stocks from actual historical financial data. For Tasks 1, 2, and 3, the curves showed monthly fluctuations in percentage, displayed on a y-axis scaled to show all the y values (typically from -80% to +80%), with most y values clustered close to 0%. For Task 4, the curves showed cumulative changes in value (computed by multiplying the changes in percentage together), again with the y axis scaled (typically showing from -100% to +200%), and in this case the curves could stray increasingly away from y = 0% over time. In all trials, there were 3 groups (or 3 layers) of curves, always color-coded with the same 3 colors. The x-axis always had 12 x values: 12 tick marks labelled with the months January to December. (As explained earlier, our techniques do not scale to large numbers of x values, but appear reasonably legible with 12 x values when displayed full-screen.)

Each of the 3 groups of curves corresponded to an actual financial sector that had been randomly sampled for stocks, thus curves within the same group tended to be correlated over time. The total number of curves over the 3 groups was 300 in half of the trials, and 1000 in the other half of trials. This total number was not evenly split between the 3 groups: because of the way the historical data was randomly sampled, sectors with more stocks yielded groups with more curves. This increases the realism of the dataset.

Supplementary materials accompanying this paper (http://www.michaelmcguffin.com/research/layers/) show typical screenshots for each task and each technique.

4.4 Participants

Twelve volunteers (2 women, 10 men) participated, ranging in age from 19 to 43 (average 30.5, median 32). All were right-handed and controlled the mouse with their right hand. None reported having physical handicaps, and none were color-blind. All participants were employees of Croesus Finansoft.

4.5 Apparatus

The experiment was conducted on a laptop running Microsoft Windows 7 with a 2.3 GHz Quad Core Intel i7-2820 CPU, 8GB RAM, and an nVidia Quadro 2000 GPU, connected to an external 24 inch 1920×1080 pixel LCD display. Participants used an external USB mouse and keyboard. Mouse acceleration was disabled.

4.6 Experimental Design

Each participant performed all tasks using all techniques. The order of presentation of tasks was fixed, and the order of presentation of techniques was counterbalanced with a 6×6 Latin square. There were 12 participants × 6 techniques × 4 tasks × 4 trials = 1152 trials in total. There were also two warm-up trials for each task-technique combination, for which data were not recorded. The session for each user lasted approximately one hour, and users were allowed to take breaks.

Time series curves were generated by randomly sampling portions of actual historical financial data, yielding enough data to show a different set of curves in each of the trials seen by a single user. This collection of data was partitioned into 6 smaller datasets that were manually checked to be similar to each other, and then randomly assigned to the six techniques, still with a different set of curves shown in each of the trials of a single user. Due to a bug, the same (random) technique-dataset assignment was used from one user to the next. (In future work, it would of course be better to have a different random assignment of dataset-to-technique for each new user.)

In all tasks, at the start of each trial, the user was shown the visualization of curves, immediately below which was the question, along with radio buttons and (in the cases of Tasks 2 and 3) a text field. After selecting a radio button and possibly entering a range in the text field, the user clicked on a "Next" button to complete the trial and move on to the next trial.

To increase control over the experimental conditions, there was no way to interact with the visualizations, e.g., no interactive selection or highlighting of curves, no brushing and linking, and no interactive highlighting of specific x or y values. However, in Task 1, to make it easier to communicate the task to the user, we highlighted the curves to be compared. This simulates a scenario where the curves have been selected by the user through some other means, and can therefore be shown highlighted and on top of the other data, with essentially no occlusion.

4.7 Results

The radio button selections in each question were used to compute a success rate for each task, which was analyzed using a chi-square test of independence. Times and Δy values were log-transformed and analyzed with ANOVA. When ANOVA yielded $p < 0.05$, we also performed a pairwise t-test with Bonferroni correction.

Technique	Success rate	Avg time (s)
Conventional	99%	14.2
Superposed	95%	17.0
Juxtaposed	96%	18.3
Shifted	94%	18.2
HSM	92%	21.2
VSM	70%	33.7

Table 2: Results of Task 1 ("compare highlighted curves"). Green arcs indicate times that are not significantly different; other pairs of times are significantly different ($p < 0.05$).

Technique	Success rate	Avg Δy	Avg time (s)
Conventional	100%	5.69	30.1
Superposed	98%	5.99	35.0
Juxtaposed	96%	2.13	31.4
Shifted	94%	4.26	31.3
HSM	96%	6.69	33.2
VSM	98%	4.59	36.9

Table 3: Results of Task 2 ("compare maximal curves"). Average Δy is a measure of accuracy, with smaller values being better. Red arcs indicate Δy values and times that are significantly different ($p < 0.05$); other pairs are not significantly different.

Technique	Success rate	Avg Δy	Avg time (s)
Conventional	100%	6.51	40.6
Superposed	88%	2.77	36.3
Juxtaposed	81%	4.35	38.4
Shifted	100%	4.10	35.2
HSM	100%	4.58	35.7
VSM	96%	6.57	33.5

Table 4: Results of Task 3 ("compare groups").

4.7.1 Task 1: "compare highlighted curves"

Table 2 presents the results. There was a significant relation between technique and success rate ($\chi^2(df = 5) = 64.19$, $p < 0.001$). Technique had a significant effect on time ($F_5 = 21.48$, $p < 0.001$). The green arcs in Table 2 show the results of pairwise t-tests: a green arc indicates that the pair is *not* significantly different, whereas pairs without an arc *are* significantly different. We see that conventional overlaid was significantly faster than all other techniques, and had the highest success rate, confirming hypothesis H1.

Technique	Success rate	Avg time (s)
Conventional	58%	20.4
Superposed	67%	23.1
Juxtaposed	50%	23.3
Shifted	44%	24.5
HSM	52%	22.4
VSM	42%	19.7

Table 5: Results of Task 4 ("compare slopes")

Technique	1-month interval	3-5 month interval
Conventional	67%	50%
Superposed	71%	63%
Juxtaposed	46%	54%
Shifted	63%	25%
HSM	63%	42%
VSM	38%	46%

Table 6: Task 4 success rates broken down by different x-intervals

Technique	Success rate (Tasks 1-4)	Avg Δy (Tasks 2,3)
Conventional	91%	6.10
Superposed	88%	4.38
Juxtaposed	84%	3.24
Shifted	85%	4.18
HSM	86%	5.64
VSM	75%	5.58

Table 7: Overall statistics

Technique	Ease of use	Speed	Learnability	Precision	Satisfaction
Conventional	2.75	2.75	4.67	2.42	2.67
Superposed	3.58	3.67	3.67	3.50	3.75
Juxtaposed	3.33	3.42	3.58	3.42	3.25
Shifted	2.75	2.75	3.08	3.17	3.17
HSM	3.00	2.17	4.25	2.50	2.25
VSM	3.08	2.95	3.85	3.00	3.02

Table 8: Overall subjective ratings on a 1-5 Likert scale. In all cases, a higher score is better.

Both small multiples techniques were significantly slower than conventional overlaid ($p < 0.001$) and yielded more errors, especially VSM. All participants rated VSM as the worst technique for this task. Since VSM uses a different y axis for each layer, participants have to read y values for each layer and compare the results mentally. They also found it more difficult to read y values because of the reduced spatial resolution in y with VSM.

5 out of the 12 participants preferred conventional overlaid for this task, and 5/12 preferred compressed-superposed. The preference by many users for compressed-superposed is surprising since occlusion was not a major problem in this task. However, examination of the data revealed that, because the highlighted curves were drawn with a thick stroke and a white contour, some occlusion between the highlighted curves could appear when curves were close enough. Compressed-superposed avoids such occlusion of endpoints entirely.

4.7.2 Task 2: "compare maximal curves"

Table 3 shows the results. There was no significant relation between technique and success rate ($p > 0.1$). Technique had a significant effect on Δy ($F_5 = 15.13$, $p < 0.001$), and also on time ($F_5 = 2.56$,

$p < 0.05$). The red arcs in Table 3 show the results of pairwise t-tests: a red arc indicates that the pair *is* significantly different, whereas pairs without an arc are *not* significantly different. We see that compressed-juxtaposed yielded a significantly smaller Δy than other techniques ($p < 0.001$), and the times for conventional overlaid, compressed-juxtaposed, and shifted were significantly smaller than for the other techniques.

We suspect that compressed-juxtaposed did well because, by not overlaying groups of curves, it made it easier for users to see the highest curve in a given group without visual interference from other layers. In certain cases, participants missed the maximal curve on the inner layers of compressed-superposed, shifted and conventional overlaid because they were hidden in the background noise, resulting in large estimate errors. Layers are also presented close to each other using juxtaposed, which also helps comparing the values.

Hypothesis H2 is only partially confirmed. HSM and VSM did perform poorly, but were not the worst two techniques. Instead, HSM was worst in terms of Δy, VSM was worst in terms of time, and neither was significantly better than the other by either metric.

Most techniques yielded success rates similar to those in Task 1, except VSM. One participant gave us a hint as to the reason why: "I felt that VSM was less painful [in Task 2 than in Task 1] since I already had to measure the Y values for the second part of the question." Since participants were forced to measure the values, they did not try to guess the answer simply by looking at the curves.

5/12 participants preferred compressed-juxtaposed for Task 2, and 5/12 preferred compressed-superposed. Users preferred juxtaposed because of the complete separation of layers, and superposed because layers are positioned closer to each other. Although the results for VSM were much better than in Task 1, 10/12 participants still selected VSM as the worst technique for Task 2. The other 2 participants chose conventional overlaid and HSM, respectively, as the worst techniques for Task 2.

4.7.3 Task 3: "compare groups"

Table 4 presents the results. There was a significant relation between technique and success rate ($\chi^2(df = 5) = 27.32$, $p < 0.001$). Technique had no significant effect on time ($p > 0.05$), but did have a significant effect on Δy ($F_5 = 2.84$, $p < 0.05$).

Hypothesis H3 is partially confirmed. Conventional overlaid yielded the worst time and 2nd worst Δy, but certain other techniques were not significantly better, and some techniques yielded a worse success rate than conventional overlaid.

Compressed-superposed and compressed-juxtaposed yielded the worst success rates. Some people found compressed-juxtaposed especially difficult when the layers were not vertically aligned (i.e., similar ranges in y but with different minima and maxima in y).

Subjective opinion for this task is much more divided than for the previous tasks, with preferences split between compressed-juxtaposed (4/12 users), compressed-superposed (3/12 votes), shifted (3/12 votes) and VSM (2/12 votes). Shifted performed quite well in this task, although some participants complained that it was difficult to interpret when the number of curves was unevenly distributed between layers. Many participants mentioned that VSM was the easiest technique for finding the least volatile sector, but was harder to use when estimating the y range (this is reflected in VSM having the best time but the worst Δy). Because VSM "squashes" curves vertically, there is less spatial resolution in y with which to judge the thickness of a group of curves. Finally, compressed-juxtaposed was especially appreciated by users for the first part of the task.

7/12 participants chose conventional overlaid as the worst technique since occlusion made it difficult to answer the task questions. In several trials, participants simply guessed that one layer was the smallest since they could barely see it. VSM (3/12 users) and HSM (2/12 users) were judged as the worst techniques by the remaining users.

4.7.4 Task 4: "compare slopes"

Table 5 presents the results. There was no significant relation between technique and success rate ($p > 0.1$). Technique had no significant effect on time ($p > 0.1$).

Recall that Task 4 involved estimating average slopes, and our hypothesis H4 that compressed-superposed would yield the worst performance. Despite how compressed-superposed introduces discontinuous gaps in x and scales each layer by a different factor, this technique achieved the best success rate! We thus reject H4. 4/12 users even preferred compressed-superposed for Task 4. We suspect that, when users were estimating the average slope of groups of curves, they were able to attend to each of the colored layers in turn and see it as a whole, with compressed-superposed reducing occlusion enough to help but without distorting slopes as much as compressed-juxtaposed, HSM, or VSM.

Curiously, VSM was also preferred by 4 users, while yielding the worst success rate. Some participants commented that VSM gave them the best picture to guess the average slopes, but users ended up giving invalid answers in many cases. Because of the small space available for each layer vertically, VSM curves appear flattened with slopes that were difficult to interpret.

Half of the trials involved a single-month interval, and the other half involved multiple months. Table 6 reports success rates for the two cases. Interpreting trends over multiple months was clearly more difficult. But surprisingly, despite discontinuous gaps and distorted slopes, compressed-superposed and compressed-juxtaposed yielded the best two success rates in trials involving multiple months. By contrast, shifted layers, which were proposed to avoid gaps and changes to slopes, did poorly in trials involving multiple months.

4.7.5 Overall

Table 7 summarizes over all tasks. Technique had a significant effect on Δy ($F_5 = 8.31$, $p < 0.001$). Although conventional overlaid yielded a better success rate than our proposed techniques, our techniques yielded better Δy values than the state of the art. Compressed-juxtaposed performed significantly better than all other techniques overall ($p < 0.001$). Our proposed techniques were also preferred by users overall: 6/12 for compressed-superposed, 4/12 for compressed-juxtaposed, and 2/12 for shifted. Compressed-superposed received the highest subjective scores in 4 out of 5 criteria (Table 8).

5 Discussion

Because Task 1 displays the curves of interest highlighted and in the foreground, in a real scenario this would require the user to somehow first select these curves. The experiment results, showing that conventional overlaid was best in this task, confirm that conventional overlaid works well when there is little or no occlusion, despite the background visual noise of hundreds of irrelevant curves. It also suggests that, in a real user interface, if a user selects individual curves for further examination, they should be displayed conventionally without compression.

In each of the 4 tasks, there is at least one criterion (success rate, time, or Δy) by which VSM or HSM or both are the worst techniques. For example, in Task 2, HSM is worst as judged by Δy, and VSM is worst as judged by time. This is explained by the cost of the user having to move their eyes a greater distance than in the other techniques to compare points with the same x or y values. This also suggests that our proposed techniques actually do benefit from always having a shared x and y axis for all groups.

Despite how compressed-superposed applies a different scale factor to each layer, theoretically making their slopes incomparable, this technique resulted in the best success rate in Task 4 where users had to compare slopes. This may be because users were comparing the average slopes of entire *groups* of curves, and in some cases slopes over multiple months (which require the user to imagine a secant line rather than pay attention to slopes of individual line segments). If, instead, the task had been to compare slopes of individual curves over a single time step, compressed-superposed might have fared poorly.

Shifted layers were designed to avoid the problems of discontinuous gaps in x and changes to slopes, but performed relatively poorly in our experiment. Many users found the use of opacity (alpha blending) to be visually noisy and difficult to understand.

To summarize our results in a single sentence, it seems that compressed-superposed was better for perceiving entire groups of curves (Tasks 3 and 4) whereas compressed-juxtaposed was better for perceiving individual curves (Task 2). Task 2 asked the user to compare individual curves from different groups at a given x value, and compressed-juxtaposed may have performed best here because it reduces occlusion of individual curves even more than compressed-superposed.

It is possible that other variants of our designs could do even better in certain cases. Figure 8(bottom) shows a variant of compressed-juxtaposed where slopes are not shown, thereby preventing the user from following any curve from one x value to the next, but perhaps making it easier to compare ranges in y and trends of entire groups over many x values. For tasks involving the perception of groups of curves, the variant shown might help the user by not distracting them with irrelevant details about slopes at each x value.

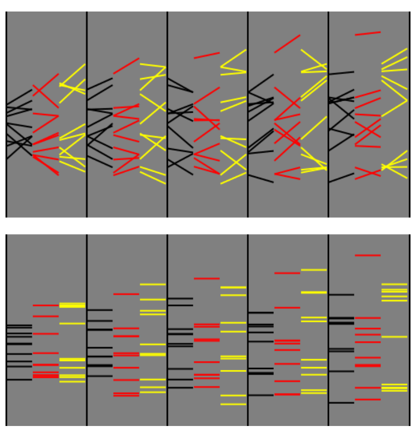

Figure 8: *Top:* Compressed-juxtaposed layers. *Bottom:* An alternative design, where short horizontal strokes show y values with no slope.

6 Conclusion

We have presented three novel techniques for reducing occlusion in time series visualizations, that can be applied when there is only a small number (≈ 12) of x values to visualize. We also presented the design rationale behind these techniques, and demonstrated experimentally that they can facilitate tasks involving hundreds of time series curves. In terms of Δy (error in estimated vertical range), our three proposed techniques were never significantly worse than other techniques, and one of them (compressed-juxtaposed) was significantly better in Task 2 while another (compressed-superposed) was significantly better in Task 3. In terms of the time required to complete Tasks 2, 3, and 4, two of our techniques (compressed-juxtaposed and shifted) were never significantly worse than other techniques. Compressed-superposed also achieved the highest success rate in Task 4. Compressed-juxtaposed and compressed-superposed were also given the highest subjective ratings in 4 out of 5 criteria.

We were also surprised that the compressed-superposed technique yielded the highest success rate in Task 4, a task requiring users to estimate slopes, despite the fact that compressed-superposed distorts slopes with a different x/y scale for each group of curves. Although there is no single "winner" across all tasks, we have established that the status quo techniques are sometimes not best, and that further investigation into these or hybrid techniques is warranted.

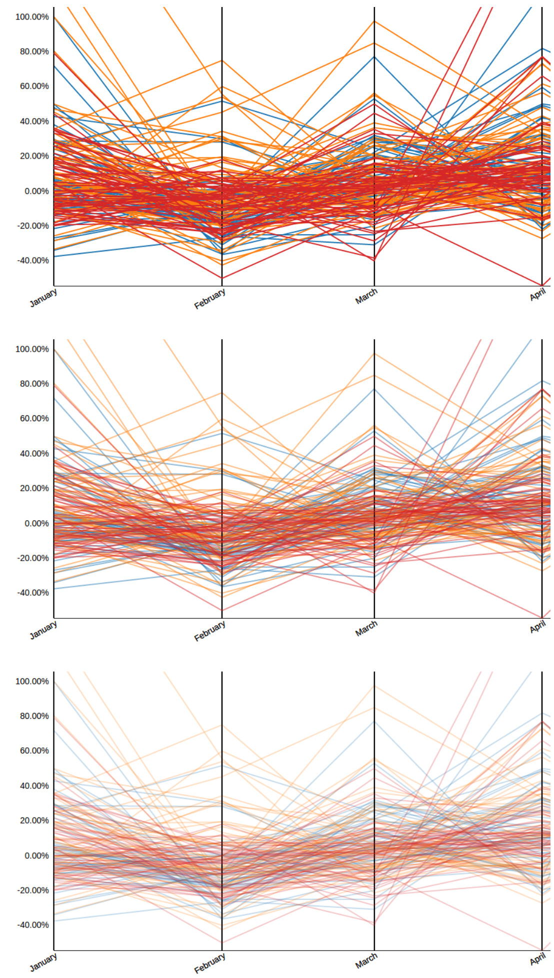

Figure 9: Conventional overlaid with opacity of 100% (*Top*), 50% (*Middle*), 25% (*Bottom*). Future work could compare these experimentally.

7 Future Work

We were surprised that shifted layers did not perform better in our experiment, given the theoretical advantages they have of preserving slopes and of introducing no gaps. Perhaps, by using different opacity levels or gradient patterns, they could be made easier to interpret. The design in Figure 8(bottom) could also be investigated, perhaps combining it with additional (semi-transparent) strokes to indicate slope at each x value.

Although this work has focused on time series, the idea of compression to reduce occlusion might also be applied to parallel coordinate plots, when the data can be partitioned into meaningful layers or subsets.

Many of the display parameters in our techniques could be tested to find their optimal values. For example, we would like to investigate the effect of different curve thicknesses, and perhaps develop a rule of thumb or algorithm for automatically choosing curve thickness that increases visibility without aggravating occlusion. We could also investigate the effect of aspect ratio of charts on different tasks. For example, one rule of thumb [17] states that line graphs have a scale chosen so that the average slope of curves is 45 degrees. We suspect that changing the aspect ratio of charts could have an effect on tasks like Tasks 3 and 4.

Transparency is another parameter to explore experimentally. Table 1 suggests that transparency has many theoretical benefits (its column is the "most green"). Figure 9 shows examples of applying it to our data.

Finally, interactive techniques could be evaluated, like brushing and linking, animated transitions between the different techniques, and filtering.

Acknowledgements

Thanks to the participants in our study for their valuable time, to the members of the HIFIV research group at ÉTS for their feedback, and to our reviewers for their suggestions. This research was supported by an IIS grant from NSERC, by FRQNT, by Croesus Finansoft, and by a scholarship from ÉTS.

References

[1] W. Aboelsaadat and R. Balakrishnan. An empirical comparison of transparency on one and two layer displays. In *Proc. British HCI*, 2004.

[2] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visualizing time-oriented data — a systematic view. *Computers & Graphics*, 31(3):401–409, 2007.

[3] L. Bartram and C. Ware. Filtering and brushing with motion. *Information Visualization*, 1(1):66–79, 2002.

[4] P. Baudisch and C. Gutwin. Multiblending: displaying overlapping windows simultaneously without the drawbacks of alpha blending. In *Proc. CHI*, 2004.

[5] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J. Fekete. GraphDice: A system for exploring multivariate social networks. In *Proc. EuroVis*, 2010.

[6] X. Bi, S.-H. Bae, and R. Balakrishnan. Effects of interior bezels of tiled-monitor large displays on visual search, tunnel steering, and target selection. In *Proc. CHI*, 2010.

[7] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys (CSUR)*, 41(1), 2008.

[8] C. Collins and S. Carpendale. VisLink: revealing relationships amongst visualizations. In *Proc. InfoVis*, 2007.

[9] M. Dumas, M. J. McGuffin, and P. Chassé. VectorLens: Angular selection of curves within 2D dense visualizations. *IEEE TVCG*, 21(3), 2015.

[10] M. Dumas, M. J. McGuffin, and V. L. Lemieux. Financevis.net: A visual survey of financial data visualizations. In *Poster abstracts at IEEE InfoVis 2014*, 2014. http://financevis.net/.

[11] G. Ellis and A. Dix. Enabling automatic clutter reduction in parallel coordinate plots. *IEEE TVCG*, 12(5), 2006.

[12] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE TVCG*, 13(6), 2007.

[13] N. Elmqvist and P. Tsigas. A taxonomy of 3D occlusion management for visualization. *IEEE TVCG*, 14(5), 2008.

[14] H. Gruendl, P. Riehmann, Y. Pausch, and B. Froehlich. Time-series plots integrated in parallel-coordinates displays. In *Computer Graphics Forum*, volume 35, pages 321–330, 2016.

[15] B. L. Harrison, G. Kurtenbach, and K. J. Vicente. An experimental evaluation of transparent user interface tools and information content. In *Proc. UIST*, 1995.

[16] B. L. Harrison and K. J. Vicente. An experimental evaluation of transparent menu usage. In *Proc. CHI*, 1996.

[17] J. Heer and M. Agrawala. Multi-scale banking to 45 degrees. *IEEE TVCG*, 12(5), 2006.

[18] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *Proc. CHI*, 2009.

[19] J. Heinrich and D. Weiskopf. State of the art of parallel coordinates. *STAR Proceedings of Eurographics*, 2013:95–116, 2013.

[20] 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.

[21] E. W. Ishak and S. K. Feiner. Interacting with hidden content using content-aware free-space transparency. In *Proc. UIST*, 2004.

[22] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE TVCG*, 16(6), 2010.

[23] D. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in visual data analysis. In *Proc. IV*, 2006.

[24] R. Kincaid. SignalLens: Focus+context applied to electronic time series. *IEEE TVCG*, 16(6), 2010.

[25] R. Kincaid and H. Lam. Line graph explorer: scalable display of line graphs using Focus+Context. In *Proc. AVI*, 2006.

[26] S. Ko, I. Cho, S. Afzal, C. Yau, J. Chae, A. Malik, K. Beck, Y. Jang, W. Ribarsky, and D. S. Ebert. A survey on visual analysis approaches for financial data. *Computer Graphics Forum*, 35(3):599–617, 2016.

[27] R. Kosara, S. Miksch, and H. Hauser. Semantic depth of field. In *Proc. InfoVis*, 2001.

[28] H. Lieberman. Powers of ten thousand: Navigating in large information spaces. In *Proc. UIST*, 1994.

[29] H. Lieberman. A multi-scale, multi-layer, translucent virtual space. In *Proc. IV*, 1997.

[30] R. Lopez-Hernandez, D. Guilmaine, M. J. McGuffin, and L. Barford. A layer-oriented interface for visualizing time-series data from oscilloscopes. In *Proc. IEEE PacificVis*, 2010.

[31] E. Pietriga, O. Bau, and C. Appert. Representation-independent in-place magnification with sigma lenses. *IEEE TVCG*, 16(3), 2010.

[32] G. Ramos, G. Robertson, M. Czerwinski, D. Tan, P. Baudisch, K. Hinckley, and M. Agrawala. Tumble! Splat! Helping users access and manipulate occluded content in 2D drawings. In *Proc. AVI*, 2006.

[33] N. H. Riche, T. Dwyer, B. Lee, and S. Carpendale. Exploring the design space of interactive link curvature in network diagrams. In *Proc. AVI*, 2012.

[34] 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 *Proc. InfoVis*, 2005.

[35] M. Schaefer, F. Wanner, R. Kahl, L. Zhang, T. Schreck, and D. A. Keim. A novel explorative visualization tool for financial time series data analysis. In *Proc. UKVAC Workshop on Visual Analytics*, 2011.

[36] T. Schreck, T. Tekušová, J. Kohlhammer, and D. Fellner. Trajectory-based visual analysis of large financial time series data. *ACM SIGKDD Explorations Newsletter*, 9(2), 2007.

[37] M. Streit, H. Schulz, A. Lex, D. Schmalstieg, and H. Schumann. Model-driven design for the visual analysis of heterogeneous data. *IEEE TVCG*, 18(6), 2012.

[38] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 2nd edition, 2001.

[39] N. Wong and S. Carpendale. Supporting interactive graph exploration with edge plucking. In *Proc. VDA*. SPIE, 2007.

[40] N. Wong, S. Carpendale, and S. Greenberg. EdgeLens: an interactive method for managing edge congestion in graphs. In *Proc. InfoVis*, 2003.

[41] J. Zhao, F. Chevalier, and R. Balakrishnan. KronoMiner: using multi-foci navigation for the visual exploration of time-series data. In *Proc. CHI*, 2011.

[42] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory analysis of time-series with ChronoLenses. *IEEE TVCG*, 17(12), 2011.

[43] H. Ziegler, M. Jenny, T. Gruse, and D. A. Keim. Visual market sector analysis for financial time series data. In *Proc. VAST*, 2010.