Using Animation to Alleviate Overdraw in Multiclass Scatterplot Matrices

Helen Chen
University of San Francisco
San Francisco, CA 94117
hhchen@usfca.edu

Sophie Engle
University of San Francisco
San Francisco, CA 94117
sjengle@usfca.edu

Alark Joshi
University of San Francisco
San Francisco, CA 94117
apjoshi@usfca.edu

Eric D Ragan
Texas A&M University
College Station, TX 77843
eragan@tamu.edu

Beste F Yuksel
University of San Francisco
San Francisco, CA 94117
byuksel@usfca.edu

Lane Harrison
Worcester Polytechnic Institute
Worcester, MA 01609
ltharrison@wpi.edu

ABSTRACT

The scatterplot matrix (SPLOM) is a commonly used technique for visualizing multiclass multivariate data. However, multiclass SPLOMs have issues with overdraw (overlapping points), and most existing techniques for alleviating overdraw focus on individual scatterplots with a single class. This paper explores whether animation using flickering points is an effective way to alleviate overdraw in these multiclass SPLOMs. In a user study with 69 participants, we found that users not only performed better at identifying dense regions using animated SPLOMs, but also found them easier to interpret and preferred them to static SPLOMs. These results open up new directions for future work on alleviating overdraw for multiclass SPLOMs, and provide insights for applying animation to alleviate overdraw in other settings.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords

SPLOM; scatterplot matrix; multi-class; multiclass; overdraw; overplotting; animation; flicker; flickering

INTRODUCTION

Scatterplot matrices (SPLOMs) are widely used to visualize multivariate data [15], and are effective tools for identifying correlations, clusters, outliers, and other features of interest [62]. Multiclass SPLOMs often use color to encode different classes in the data, enabling powerful comparisons between classes or subgroups. While color is an effective way to represent multiple classes in scatterplots [28], fundamental challenges remain when using color to represent multiclass data in SPLOMs.

Consider the widely-known diamonds dataset in ggplot2, which contains not only continuous variables, but also categorical variables that are critical for uncovering underlying patterns in the data [63]. Visualizing this dataset in a multiclass SPLOM results in significant overdraw, which occurs when points or glyphs are drawn on top of each other and occlude the underlying data. For moderate to large datasets like this one, overdraw affects the ability of viewers to accurately understand the data distribution and discern relationships between variables and subgroups in the data.

As we discuss in the next section, multiclass SPLOMs are one of the hardest and least studied settings for alleviating overdraw. In this paper, we propose and evaluate a straightforward animated technique using flickering points for alleviating overdraw in multiclass SPLOMs. Our contributions are:

1. We provide an interactive web-based open source tool demonstrating our animated approach on several canonical multiclass multivariate datasets of varying size. We encourage readers to visit vgl.cs.usfca.edu/animated-sploms/ to interact with our tool.

2. We demonstrate how to explore multiclass multivariate data with our tool using the diamonds dataset [63]. Despite needing to render over a million points across several buffers, we are able to successfully use our tool to identify regions of varying density (i.e. number of points) and diversity (i.e. number of classes) in this large complex dataset.

3. Finally, to determine the impact animation has on alleviating overdraw, we conducted a study with 69 participants who performed tasks using both animated and static SPLOMs. Participants were not only able to better identify dense regions in the animated version, but were also able to complete these tasks in times comparable to the static version. Most participants also indicated that animated SPLOMs were easier to interpret than static SPLOMs and preferred the animated approach.

CHI 2018, April 21–26, 2018, Montréal, QC, Canada.

ACM ISBN 978-1-4503-5620-6/18/04 ...$15.00.
http://doi.org/10.1145/3173574.3173991

These results show that instead of being "too distracting" for users, animation was able to alleviate overdraw in this challenging multiclass setting. We discuss other implications of these results in our future work section.

MOTIVATION AND RELATED WORK

We next provide background information on scatterplots and scatterplot matrices, and present related work to motivate our decision to explore the impact of animation on alleviating overdraw in multiclass scatterplot matrices.

Background

The **scatterplot** is a common technique introduced in the early 1800s that visualizes the relationship between two variables using horizontal and vertical position [27]. Scatterplots have been actively researched in the information visualization community. For example, Bachthaler and Weiskopf extended the power of scatterplots to visualize continuous data frequently generated in computational simulations [4]. Doherty et al. conducted a user study on estimating correlations using scatterplots and found the overall notion that individuals underestimate correlations is inaccurate [20].

The **scatterplot matrix** (SPLOM) technique visualizes multivariate data using a small multiples approach that places pairwise bivariate scatterplots of variables in a matrix [27]. SPLOMs can be effective for identifying correlations, clusters, outliers, and other features of interest in multivariate data [62]. As such, SPLOMs are a common focus in the research community and frequently used by practitioners. For example, Elmqvist et al. developed a technique that uses a SPLOM as an overview as well as a navigational interface for comparing and transitioning between individual scatterplots [23]. Lehmann et al. presented an approach to explore large SPLOMs by quickly selecting "relevant" plots [41]. SPLOMs have also been generalized to utilize other types of visualizations within the matrix beyond scatterplots [24, 37], but these generalized plot matrices (GPLOMs) are outside the scope of this paper.

There are many variants of scatterplots and SPLOMs, allowing for more variables to be visualized by modifying the shape, color, and size of the points. This is commonly done to visualize **multiclass data**, where a categorical variable is used to classify the data into multiple subgroups. Gleicher et al. performed a large scale user study that evaluated participants' ability to compare the means of multiple classes being displayed [28]. They found that using color to distinguish multiple classes is better than shape. Additionally, encoding data with redundant cues for tasks involving individual classes did not affect user performance. This work motivates our use of color to encode classes.

Why Multiclass SPLOMs?

One of the major problems with visualizing scatterplots and SPLOMs is with **overdraw** for moderate to large datasets. Overdrawing (or overplotting) refers to multiple data elements being resolved to the same pixel location in the visualization. Significant overdrawing will hinder the viewer's ability to discern meaningful information from the visualization, especially regarding the distribution and density of the data.

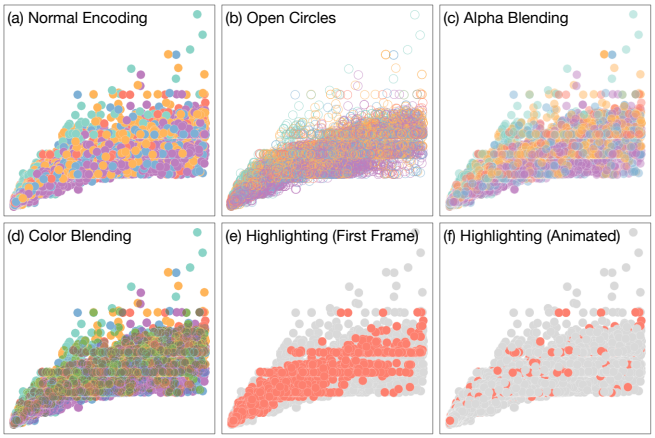

Figure 1. Examples of overdraw with common plotting techniques implemented in our tool. (a) The default point encoding with opaque circles filled by class color and a thin white stroke. (b) The open circle encoding with a thick stroke colored by class and no fill. (c) The alpha blending encoding with slightly transparent fill colors. (d) The color blending encoding using the "darken" mode where overlapping points causes colors to get darker. (e) The first animation frame from highlighting the red class using the normal encoding. When a class is first highlighted, all points belonging to that class are brought to the front and non-highlighted points are drawn in grey. (f) A snapshot of the animation after highlighting the red class. Several of the grey points have now been redrawn, covering some of the red points from before.

Simple approaches for alleviating overdraw in scatterplots include using different types of glyphs (shapes [40], sunflowers [16]), varying the sizes of glyphs [42], applying alpha blending or opacity [14], and applying a small amount of noise to the dataset (e.g., jittering [12]). These simple approaches are less effective for significant overdraw, which is often the case for SPLOMs due to the small size of each individual scatterplot in the matrix (see Figure 1 for examples).

There are more sophisticated techniques for dealing with overdraw in single class settings. For example, Carr et al. [11] introduced the use of drawing contours around dense regions, hexagonal binning, and animation as alternative techniques to deal with overdraw in dense scatterplots. Keim et al. presented the "generalized scatterplots" technique which uses pixel-based techniques to address overdraw [39]. Hao et al. extend the use of aggregation to provide users with a binned representation of the data [31]. However, it is unclear how to apply approaches like hexbinning to multiclass SPLOMs. A small multiples approach (with one view per class) could be used for an individual scatterplot, but SPLOMs already use small multiples and have very little screen space to spare.

Considering other approaches to addressing the general challenge of overdraw, Dang et al. demonstrated a stacking method using three-dimensional visual space [18]. However, applying this to dense datasets in multiclass scenarios would not allow for the distinction of classes, and small multiples could extend the overplotting problem to 3D space. Sedlmair et al. [57] examined the ability of various dimensionality reduction techniques [60] (such as PCA [38] and MDS [10]) to separate classes when visualized using 2D scatterplots, SPLOMs, and 3D scatterplots. They found that users performed sufficiently

well for class separability tasks when using 2D scatterplots, and using SPLOMs were valuable at times. However, they found that 3D scatterplots hurt a user's ability to discriminate between classes due to visual clutter.

Mayorga and Gleicher introduced a contour-based technique called splatterplots, which uses color blending, and applied their approach to multiclass SPLOMs [44]. However, multiclass SPLOMs often have overlapping classes and both alpha and color blending approaches struggle in this scenario. For example, representing four classes requires 15 colors, and five classes requires 31 colors [44]. These color combinations quickly become impossible to map back to individual classes.

Sampling is another common strategy for reducing clutter and alleviating overdraw [22]. For example, Bertini and Santucci developed a framework to measure the "degradation" in quality when visualizing large datasets using a scatterplot. They developed an automatic sampling strategy that addresses the degradation in quality that a viewer experiences [8]. In subsequent work, Bertini and Santucci further fine-tuned their sampling strategy to include best uniform sampling and non-uniform sampling to reduce the observed degradation in quality [9]. However, how degradation is affected by their sampling strategy in a multiclass setting is unknown. Chen et al. instead used a multiclass blue noise sampling scheme to alleviate overdraw in multiclass settings [13]. Their sampling scheme eliminates occlusion and produces consistent scatterplots regardless of the drawing order for an individual multiclass scatterplot. These sampling strategies are related to our animated approach, which is similar to making visible different samples of data over time. However, it is unclear how to adapt these sampling strategies from individual scatterplots to SPLOMs. Should the sampling schemes be adapted to consider how the entire SPLOM would be affected (and thus each scatterplot in the SPLOM shows the same sample of data), or individually to each scatterplot such that each scatterplot in the SPLOM shows different samples?

There is also work on optimizing various parameters to alleviate overdraw in scatterplots. For example, the work by Matejka et al. uses a crowd-sourced opacity optimization technique [43]. Micallef et al. takes this a step further by optimizing other parameters in addition to opacity based on visual quality metrics [47]. Both of these are not tailored for multiclass SPLOMs, and like the sampling approaches, it is unclear how this will affect the optimization process.

Finally, Eisemann et al. presented an interactive technique using an overlapping hierarchy of scatterplots as an alternative for multiclass SPLOMs [21], but the results still suffer from occlusion as multiple scatterplots are overlaid. We primarily focus on alleviating overdraw in multiclass SPLOMs rather than explore general alternatives to the SPLOM technique.

Overall, few research efforts have targeted or applied their techniques to multiclass SPLOMs. As we discussed in this section, this is an important distinction since multiclass SPLOMs have additional constraints to consider versus single class scatterplots. This is not to say existing approaches cannot be tailored for multiclass SPLOMs, but it is clear alleviating overdraw in the multiclass SPLOM setting needs more exploration. We therefore focus on the challenging and understudied problem of overdraw in multiclass SPLOMs.

Why Animation?

Our research explores the use of animation to address the problem of overdraw in multiclass SPLOMs. Motion is easily perceived in the peripheral vision [49] and is highly effective at drawing attention [53], so it is no surprise that animation is an active research topic in data visualization. For example, Ware and Bobrow used oscillatory motion to highlight subsets of a node-link diagram [62]. Feng et al. used animation to communicate uncertainty in parallel coordinates and scatterplots [26]. Animation is often used when visualizing dynamic graphs, trees, and diagrams (e.g., [1–3, 30, 54, 55, 64]), and to visualize changes over time in other techniques (e.g., [17, 29, 59]). Research has found animation to be helpful in some situations (e.g., [7, 19, 35, 51]), though it can be detrimental in other cases (e.g., [6, 52]). For example, Heer and Robertson found that animated transitions between different representations improved graphical perception [35], but Robertson et al. later found that animation of trend traces was less effective compared to static alternatives [52].

More importantly, animation has been used to address overdraw for other visualization techniques (e.g. [22, 25, 58]). Even better, animation can be applied to many existing overdraw techniques to adapt them to multiclass settings. However, animation has not yet been studied as an approach to handling overdraw problems in scatterplots or multiclass SPLOMs. Also, there is not yet a user study confirming this approach is effective, allowing critics to dismiss animation as being "too distracting" without further consideration. Given the mixed success and problems with applying animation to different visualization scenarios [6, 35, 52], there is a need for empirical studies to assess the effects of animation on data interpretation tasks in multiclass scatterplots and SPLOMs.

When considering how best to use animation in this setting, we refer to three categories of motion for visualization: flicker, direction, and velocity of motion [34]. Our approach uses **flicker**, where points disappear and reappear over time at non-regular intervals, but do not move in position. Related work by Bartram et al. found that anchored motions like flicker were less distracting than other types of motion [5], alleviating initial concerns that animating the entire SPLOM at once would be too distracting for users. Also relevant to our design, Huber and Healy demonstrated that flicker must have a cycle length of 120 milliseconds or greater for accurate detection [36]. For our implementation and study, we animate at a rate of 30 frames per second, but the exact rate that points flicker depends on the density in that region. By design, it is possible that points flicker faster than viewers are able to identify individual points. However, we use flicker to draw attention to dense regions rather than to identify individual points. We also chose to animate the entire dataset instead of a subsample, allowing us to fulfill Munzner's requirement to provide points with "guaranteed visibility over time" [48]. Consequently, the dataset size affects the time required to animate the entire dataset, as discussed in later sections.

APPROACH

Our basic approach continuously redraws points by looping over the dataset, allowing for overdrawn points to eventually reappear. We wanted an open source interactive web-based tool to demonstrate our approach, and as such investigated various JavaScript libraries. We decided to use the HTML5 canvas to avoid the performance issues related to SVGs, and the P5.js JavaScript library [45] since our animation paradigm fit naturally with the P5.js continuous draw loop.

Our tool supports several URL query string parameters to control how points are rendered, including animation speed, enabling pre-rendering or sampling, modifying the point encoding, and controlling the scale amount. See Figure 2 for an illustration of our approach and tool, or visit vgl.cs.usfca.edu/animated-sploms/ for a description of each parameter and a live demo.

Encoding

We start with a traditional SPLOM. Since the scatterplots along the diagonal and lower triangle of the matrix are redundant, we only draw scatterplots in the upper triangle and rotate the triangle so that it lies in the upper-left quadrant. Each individual scatterplot is fixed in size, so the overall size of the SPLOM depends on the number of variables being visualized. Users can control the size using the `scale` parameter. For each row and column in the matrix, we draw the variable name and range (minimum, midpoint, and maximum values). We place a legend and other controls in unoccupied regions in the matrix, as depicted in Figure 2c–2e.

We use color to encode the class of each point based on the work by Gleicher et al. [28], and used a qualitative color scheme adapted from ColorBrewer [33]. By default, the points have a fully opaque (not transparent) fill based on their class with a thin white stroke. See Figure 1a for an example. The white stroke ensures that points of the same color are still distinguishable from each other. We also implemented other point encodings that are commonly used to alleviate overdraw, as depicted in Figure 1b–1e. Users may change the point encoding using the `encoding` parameter. By default we draw circle glyphs for each point, but users may change this to squares using the `shape` parameter.

Using these encodings without animation can give mixed results for diverse regions with significant overdraw. For example, consider the static snapshots in Figure 1. It is difficult to determine the most dense and diverse regions (it all appears dense and diverse), as well as determine any relationships between the classes. We later demonstrate how animation allows answering such questions with a usage scenario.

Animation

We continuously loop through the dataset to animate the SPLOM, redrawing one or more rows of points from the dataset at a time. For example, drawing one dataset row with five variables (not including class) results in drawing ten points per animation frame: one for each scatterplot in the matrix. The number of dataset rows animated at once can be controlled by the `rows` parameter or the spinner widget depicted in Figure 2c. To keep the entire dataset visible at all times, we do not wipe the canvas clean between animation frames. A progress bar, located above the legend in Figure 2d, shows the percentage of dataset rows drawn, and then the percentage of rows that have been re-animated for every subsequent loop through the data.

Rows in the dataset are drawn in sequential order unless the `shuffle` parameter is enabled, which randomly shuffles the order dataset rows are animated. The order dataset rows are drawn can impact the perception of the data. For example, if the dataset is sorted by class then there will be less flickering between colors in dense regions when animating in sequential order. The time it takes to fully render the dataset can also cause issues if outliers appear towards the end of large datasets. To ensure all outliers are immediately visible, we recommend users pre-order datasets to place outliers first or use the `prerender` parameter to render the entire dataset before the animation begins. Most small to moderate-sized datasets have this parameter enabled by default in our tool.

Using our approach, the amount of flickering indicates the amount of overdraw in a region. Points that are overdrawn will reappear during later animation frames. Outliers are perceived to be static, even when redrawn. Similarly, regions with low density (i.e., few points) and low diversity (i.e., few classes) will have little to no movement and appear static. Regions with high density but low diversity will have movement due to the white outline of each point, but will not change in color (since each class is represented by a color). Regions with both high density and high diversity will change color frequently. This is where animation successfully draws attention through flickering, whereas diverse and dense regions could potentially be misinterpreted due to overdraw in static SPLOMs.

Interactivity

When users hover over a scatterplot in the matrix, the associated row and column labels on the left and top of the matrix are highlighted. We also allow users to highlight points by class using several off-screen buffers[1]. Specifically, for every animation frame, we rendered each point being animated to three buffers: a buffer with all points colored, a buffer with all points in light grey, and a class-specific buffer.

Consider the example illustrated in Figure 2. If we were rendering a point belonging to the *A* class, we would draw it in the "All," "Grey," and "Class A" buffers. Since there are four classes in this example, we need six buffers total. Users may select one or more classes by clicking on the corresponding color square(s) in the legend, as depicted in Figure 2e. When users select a class, the appropriate buffers are displayed. The example shows classes *B* and *C* highlighted, which requires the "Grey," "Class B," and "Class C" buffers to be shown. Any combination of classes may be selected. When all classes are selected, only the "All" buffer is shown. When users first highlight a class, points from non-highlighted classes are shown in the background in grey and points from all highlighted classes are shown in the foreground in their appropriate class color. For example, Figure 1e shows the first frame after highlighting

[1]These off-screen buffers are separate from any front or back buffers used for double-buffering by the P5.js draw loop internally.

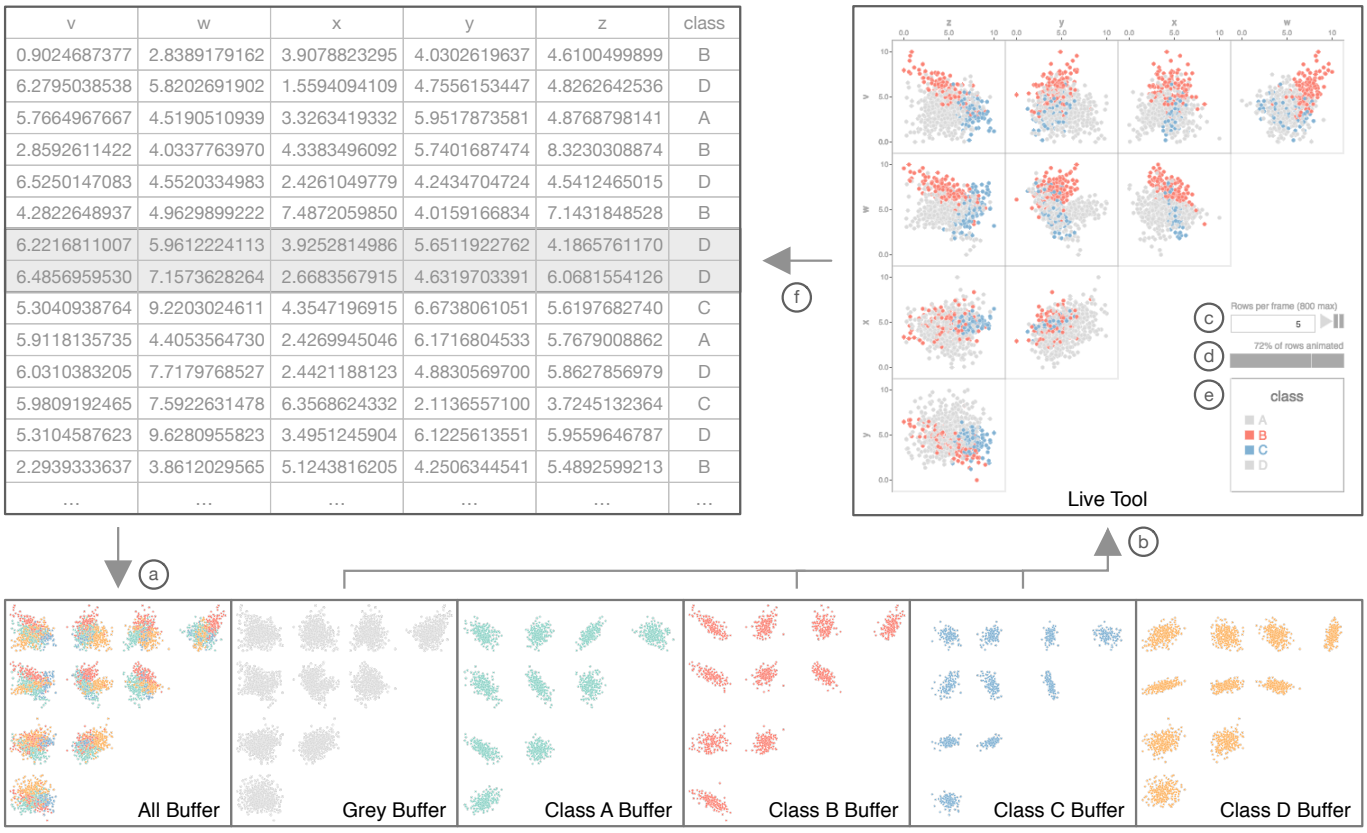

Figure 2. Illustration of our tool and approach. (a) We first draw the points for the current dataset row(s) in the all, grey, and class buffers. (b) The appropriate buffers are shown in the live tool. The buffers for highlighting the red *B* and blue *C* classes are shown here. (c) Users may use the spinner widget to change the number of rows animated per frame, and the play/pause buttons to stop and restart animation. (d) The progress bar shows the percent of rows that have been animated or re-animated since the last loop through the dataset. (e) The legend shows the color assigned to each class. Users may click on the colored boxes to highlight different classes. (f) Every frame, we advance to the next set of dataset rows and then start at (a) again.

Figure 3. Snapshot of our tool animating the diamonds dataset [63]. Only the first two rows of the SPLOM are shown, with the regions corresponding to our usage scenario highlighted. Some of these observations are only possible by interacting with our animated tool. (a) Example of an outlier in the *z* column of the SPLOM. (b) Example of a low density and low diversity region in the *depth* row of the SPLOM. (c) Example of a low density and high diversity region in the *carat* vs *table* scatterplot. (d) Example of a high density and high diversity region in the *depth* versus *table* scatterplot. (e) Highlighted region in the *price* versus *carat* scatterplot.

the red class. If animation is enabled, then points belonging to non-highlighted classes may start to be redrawn on top of the other points, as depicted in Figure 1f. Users can highlight points from different classes regardless of whether or not the animation is paused.

USAGE SCENARIO

We demonstrate our approach by visualizing the diamonds dataset included in the ggplot2 R package [63] containing the prices and other attributes of round-cut diamonds. This dataset includes 53,940 rows, 3 categorical columns, and 7 continuous columns. We visualize the *x*, *y*, *z*, *price*, *table*, *depth*, and *carat* columns in the SPLOM and use the *cut* column with 5 categories to color each point. Visit vgl.cs.usfca.edu/animated-sploms/ to view an animated live demo and Figure 3 for a static illustration.

Observations

First, we verified the detection of outliers and areas of varying density (number of points) and diversity (number of classes) with the animation technique. Consider Figure 3a, which points to values near 35 in the *z* column. The lack of flickering indicates the blue *Very Good* circle is an outlier in a low density region. We confirmed there is only one row with a *z* value of 31.8 in the dataset.

Figure 3b points to another region in the *depth* row with values greater than 65. This region has little movement and only one prominent color, indicating it has low density and low diversity. Only 2% of rows have *depth* values in this range, and 93% of those rows belong to the teal *Fair* class.

We were also able to identify regions with high diversity. For example, Figure 3c points to a region with *carat* values above 2.6 in the *carat* versus *table* plot. This region has several colors but little movement, indicating it has high diversity but low density. Indeed, this region contains all classes but less than 0.2% of all the rows in the raw data.

Our approach generates the most movement for regions with both high density and diversity. Consider the *depth* and *table* plot in Figure 3d. It has a large amount of flickering colors for values between 55 and 65, indicating it has high density and high diversity. We confirmed this region contains 80% of the points and all five classes in the raw data.

We made other general observations regarding the different classes using our tool. For example, consider again the dense and diverse region of points in the *depth* versus *table* scatterplot with values between 55 and 65 in Figure 3d. In the default view, it is difficult to determine if there are points of the teal *Fair* class in this region. This class appears infrequently in this region if we highlight only this class. We confirmed that only 1% of the points in this region belong to the *Fair* class.

Overall, the points for the teal *Fair* class are noticeable since they tend to occupy regions with low diversity. However, this class represents only 3% of the dataset. Consider the region in Figure 3e in the *carat* versus *price* plot. If we highlight only the teal *Fair* class in our tool, it is more apparent that these points make up a small percentage of the dataset.

Rendering Performance

We tested our tool on an iPad Air 2 tablet and 2013 iMac Desktop, both with 16 GB of memory. At 30 frames per second, it takes 3 minutes to loop through the entire diamonds dataset at 10 rows per frame. Both our tablet and desktop system were able to handle 10 rows per frame.

At 100 rows per frame, it takes only 18 seconds to animate the entire dataset. However, this requires animating 2,100 points per frame (100 points for each of the 21 cells in the SPLOM). This caused a sharp drop in framerate, suggesting we may need a renderer with better performance for larger datasets. As the increased rate of flickering may be overwhelming to some users, we let users control this rate. We did not observe these issues with datasets with under 10,000 rows.

We also observed performance issues prerendering a dataset of this size. To prerender all 53,940 rows in the 21 plots in our SPLOM, our tool must draw 1,132,740 points per buffer used for brushing. The browser crashed prior to rendering the first frame on both systems. Since the first frame is always the same, we may be able to eliminate that render time by storing the first frame for each buffer as an image instead. This issue can also be alleviated by ordering the dataset to place outliers first, reducing the need to prerender.

Overall, this scenario demonstrates that our interactive web-based proof-of-concept tool is capable of handling large datasets, but with some performance tradeoffs that we can address in our next iteration. We suspect much of our performance issues have more to do with our in-browser implementation of interactivity using multiple buffers than the animated approach itself.

Next, we evaluate not only whether our animated approach *can* alleviate overdraw, but also whether it is *better* than non-animated settings.

USER STUDY

We conducted a user study[2] to evaluate the effect of animation on alleviating overdraw in multiclass SPLOMs. Specifically, we focused on understanding how animation impacts a viewer's ability to find regions with high density and/or diversity, the amount of time spent on finding these regions, and whether animation is distracting or difficult to interpret.

Experimental Design

The study was run as an in-lab study administered via the Experimentr [32] framework. The experiment followed a within-subjects design with all users completing both conditions (static vs animated) for both tasks (targeted vs diverse density). We chose the default point encoding with opaque colored circles to isolate the impact animation made on user performance and preference. For this study, we removed most of the interactivity from the tool, including highlighting classes and the ability to change parameters like the animation rate and point encoding. Visit github.com/usfvgl/splom-studies for the entire Experimentr study setup.

[2]The study IRB ID 796 was reviewed by the Institutional Review Board for the Protection of Human Subjects at the University of San Francisco and approved as Exempt under 45 CFR 46.101(b).

Analysis Tasks and Data Sets

Since we focus on addressing overdraw problems, we chose tasks related to density (number of points) and diversity (number of classes) based on work on scatterplot tasks by Sarikaya and Gleicher [56]. Specifically, the two tasks were:

- *Targeted Density*: Identify the region in a specific cell with the most circles.
- *Diverse Density*: Identify the region in any cell with the most circles and at least four different classes (colors).

We use the term *cell* to refer to a specific scatterplot within the SPLOM. The *targeted density* task focused on density without consideration for diversity, and asked participants to focus on a specific cell. Participants could not continue to the next task until they made a selection within the correct cell in the SPLOM. The *diverse density* task focused on both high density and high diversity. Participants could select regions in any cell within the SPLOM, but had to select a region with at least four or more classes to continue. Therefore, our two hypotheses for the user study are as follows:

- *Hypothesis 1*: Participants will identify *targeted density* more accurately using an animated *vs.* static SPLOM.
- *Hypothesis 2*: Participants will identify *diverse density* more accurately using an animated *vs.* static SPLOM.

For user testing, we generated synthetic datasets tailored for each task using the pandas [46], scikit-learn [50], and numpy [61] libraries in Python. We experimented with different dataset sizes, and choose to generate datasets with 300 rows, 5 columns with values between 0 and 10, and 1 class column with 5 possible classes. We found that datasets of this size created multiple regions with overdraw, could be pre-rendered quickly in the browser, resulted in SPLOMs that fit entirely in the browser window without scrolling, and would loop more than once within a reasonable amount of time when animating 1 row per frame and 30 frames per second.

Specifically, we generated two training datasets and one larger core dataset per task. We generated 6 additional datasets (3 per technique) from each of the core datasets by shuffling the rows, randomly assigning variable labels v, w, x, y, and z to columns, randomly flipping all values in the dataset, and randomly assigning class labels A, B, C, D, and E to the class numbers. The goal was to generate datasets with comparable patterns, density, and diversity, but different enough to avoid learning effects as the study progressed.

Study Procedure

Before beginning the trials, participants completed a questionnaire about basic demographic information such as age bracket and education level. The application then provided a brief introduction to scatterplots and SPLOMs and asked participants how comfortable they were with these techniques.

Next, participants were provided instructions on how to perform each task. They selected regions by dragging a fixed-size black rectangle on the plot. After the instructions, participants practiced each task on both static and animated SPLOMs before beginning the main task trials. The practice questions provided immediate feedback on how many circles and colors were currently selected. Both the instructions and training always showed a static SPLOM first followed by an animated SPLOM. Participants then completed each task six times total (three trials for static and three trials for animated). We randomized the technique order and which datasets were used for each technique for each participant, but not the task order. For example, users always saw task 1 before task 2, but some users saw dataset 1 first using a static SPLOM and others saw dataset 1 last using an animated SPLOM.

After completing both tasks, participants were asked whether they found each technique easy to interpret, which technique they preferred, and whether they found animation distracting.

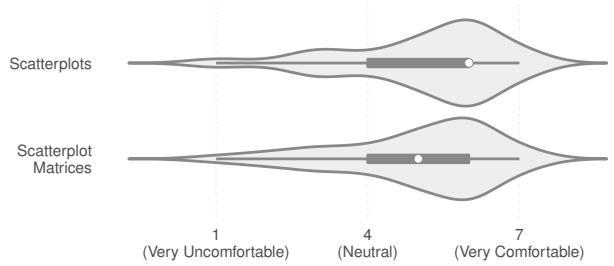

Figure 4. Participants were asked to rate their comfort level with scatterplots and SPLOMs on a scale of 1 to 7. Most participants were somewhat comfortable interpreting these plots. *Interpretation:* The violin plots show the density of ratings, the box plots show the first and third quartiles, the whiskers show 1.5 times the inter-quartile range, and the white circle shows the median. This encoding is consistent across all figures.

Participants

We recruited 69 undergraduate and graduate university students to participate in the study. To be eligible to participate in the study, participants had to confirm they were 18 or older, did not have a known form of color blindness that interfered with their ability to interpret the class colors, and did not have photosensitive epilepsy.

Approximately one third of participants identified as female and the other two thirds identified as male. Over 78% of participants were under 24 years old, and the remaining 22% of participants were 25 to 44 years old. Participants took between 6 and 26 minutes to complete the study (about 15 minutes on average).

Participants provided ratings about their comfort level with both scatterplots and scatterplot matrices. Figure 4 shows a summary of the participants' comfort levels. Most participants were moderately comfortable interpreting the plots.

Results

We analyzed the results in terms of task performance as well as for subjective preference.

Task Performance

Both the *targeted density* and *diverse density* tasks asked participants to select a region with the most circles (with the first specifying a cell, and the second specifying a number of classes). To assess task performance, we considered the average number of points in the selected regions and the amount of

time taken to complete the tasks per participant. See Figure 5 for an overview of those results.

We conducted statistical tests for the performance results for each task to test the hypotheses that the animated SPLOMs would enable better performance than static SPLOMs. For the *targeted density* task, neither task time nor the number of identified points were normally distributed. Time was skewed right, which was likely due to the online nature of the study and individual preference; most participants preferred to complete the tasks quickly, but some participants opted to spend more time answering questions. The identified points were skewed left, which was likely due to an upper limit in the number of points in any region of the plots; in other words, it was not possible to identify more than the maximum number of points. Since the measures did not meet the assumptions for parametric testing, we therefore opted for nonparametric Friedman tests for statistical comparison of the animated and static SPLOMs.

To test hypothesis 1, we examined the number of points identified, and found a significant effect with $\chi^2(1) = 10.88$ and $p < 0.001$. Participants identified regions with significantly more points when using the animated SPLOMs ($M = 60.63$, $SD = 13.24$) compared to the static SPLOMs ($M = 56.22$, $SD = 12.24$), supporting hypothesis 1. Regarding task completion time for the *targeted density* task, each trial took approximately 23 seconds to complete. No evidence of a difference was detected ($\chi^2(1) = 0.71$, $p = 0.40$) between animated and static SPLOMs.

To test hypothesis 2, we examined the number of points identified from the *diverse density* task. Assumptions of sphericity and normality were met for parametric testing, so we used a repeated measures ANOVA (analysis of variance) test. The effect was significant with $F(1,68) = 18.97$ and $p < 0.001$, showing more points were identified with animated ($M = 42.65$, $SD = 6.53$) than with static SPLOMs ($M = 39.45$, $SD = 5.98$), supporting hypothesis 2. Average task completion time for the *diverse density* task was approximately 21 seconds per trial. Completion times for this task were not normally distributed, so we again used a Friedman test. No evidence of a difference was found, with results yielding $\chi^2(1) = 2.45$ and $p = 0.12$ between animated and static SPLOMs.

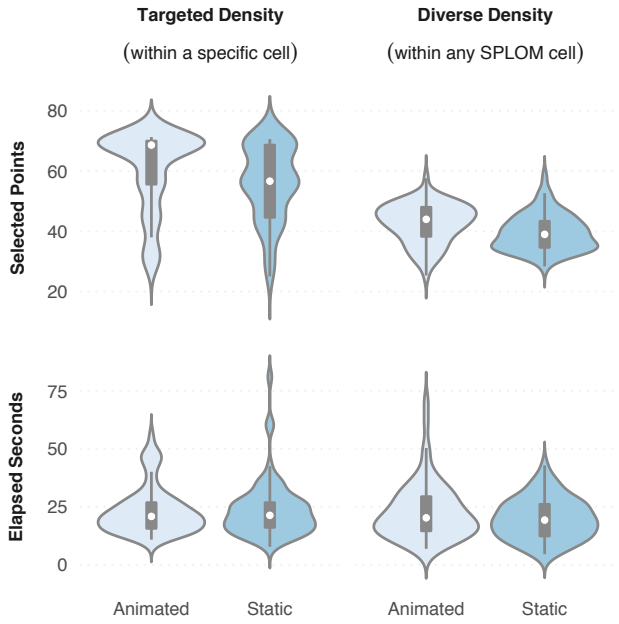

Figure 5. Performance results from our user study. See Figure 4 for how to interpret these plots. *Top*: The number of points selected for each task and technique. Participants were able to select denser regions using animated versus static SPLOMs on average. The effects were statistically significant for both tasks. *Bottom*: The elapsed seconds it took to complete the tasks per participant. No evidence of difference was detected between techniques for either task.

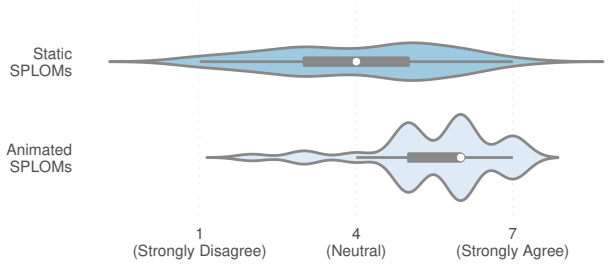

Figure 6. Users were asked to rank how strongly they agreed with the statement that each technique (static versus animated SPLOMs) was easy to interpret. The animated plots had statistically significantly higher ratings than static. See Figure 4 for how to interpret these plots.

Preferences

For subjective measures, participants found animated SPLOMs easier to interpret on average than static (see Figure 6). Due to the ordinal nature of interpretability ratings, we compared ratings for static and animated SPLOMS with a Friedman test. The animated plots had significantly higher ratings with $\chi^2(1) = 16.29$ and $p < 0.001$. These responses align with the task performance results, suggesting that participants were aware that they could more easily interpret region density with the animated SPLOMS.

Participant responses also clearly indicated a strong overall preference for the animated SPLOMS over the static variants (see Figure 7, top). Of the 69 participants, 77% preferred animation. Despite the clear preference and superior interpretability of animated SPLOMS, some participants found the animation distracting (see Figure 7, bottom). However, only 30% of participants indicated agreement (at any level above neutral) that animation was distracting during the study.

Discussion

Applying animation is a powerful and straightforward approach to handling overdraw in multiclass SPLOMs. The human visual system is well-suited to interpreting differences in motion, and leveraging motion differences can help users to identify differences in densities and class compositions in SPLOM visualizations. The results of the user study suggest the animation technique is both effective at alleviating overdraw and is easy to understand. Participants performed better using animated versus static SPLOMs, confirming that animation helps distinguish region density in SPLOMs with high

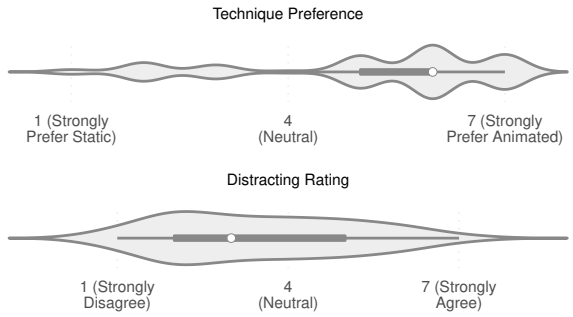

Figure 7. Users were asked to rate their preference between static and animated SPLOMs, and whether they agreed with the statement that animation is distracting. Most users preferred animated but some found it somewhat distracting. See Figure 4 for how to interpret these plots.

overdraw without the animation being too distracting. Furthermore, participants found animated SPLOMs easier to interpret and preferred them over static SPLOMs. It is a promising result that the animation approach did better in both performance and interpretability in a comparable amount of time as a static SPLOM under the same circumstances.

While distraction is a valid common concern for the use of animation in information visualization, the results of the user study indicate that distraction was not an issue for most participants. The experiment did not produce evidence that animation increased the amount of time it took participants to complete those tasks, and the majority (70%) of participants reported no problems with being distracted by the motion.

However, the benefits were not universal. Approximately 29% of participants on the *targeted density* task (selecting a dense region in a specific cell) and 28% on the *diverse density* task (selecting a dense and diverse region in any cell) performed worse on average with the animated SPLOM, and a similar number (30%) found animation at least somewhat distracting. We plan to further explore the relationship between how distracting a viewer finds animation, the rate of animation and overdraw, the amount of time spent training, and overall performance in the future. For now, these results highlight the need to allow viewers to disable or adjust the rate of animation (which our tool supports).

FUTURE WORK

While our work does not establish the best approach for alleviating overdraw in this setting, it does show that animation is a promising approach for multiclass settings without being too distracting. This opens up many other promising directions of future research.

For example, we only focused on the impact of animation for the simplest point encoding in our user study. Additional study is needed to understand the impact of animation on other encodings, especially when combined with other techniques for alleviating overdraw such as alpha blending, color blending, and sampling.

We also fixed the animation parameters in our study. Future research can explore how to optimize the rate of animation based on user perception, similar to the work already done on optimizing opacity [14]. Another direction of study is just noticeable differences (JND) between frames in an animation and the ability of users to make quantitative judgments with them. While the rows per frame can be controlled by the viewer when viewing the animation, its role in the perception of diversity in a region needs to be studied.

Our study results showed that animation did not increase the time required to complete tasks, but this could be related to the small dataset size used in our study. Another promising direction of research is to explore the relationship between the time required to complete tasks using animation with respect to dataset size. There is also the related question on how the rate of animation and the time required to see all of the data affects performance. These additional studies can help us understand the scalability of this technique, and whether there is a dataset size for which the approach is no longer beneficial.

Finally, this work was motivated by the lack of research focused on overdraw in multiclass SPLOMs. Animation could be useful for alleviating overdraw in scatterplots with a single class as well. Previous work by Shearer et al. also proposes using a similar animated approach with other visualization techniques [58], but there has been little work studying how to do this effectively. Clearly more study is needed on whether animation is effective at alleviating overdraw in other settings.

CONCLUSION

Multiclass SPLOMs are one of hardest and least studied settings for dealing with overdraw, and there are not yet studies examining how well animation alleviates overdraw. Our work is a first step to remedying that. Specifically, we provide and demonstrate an interactive web-based tool capable of rendering of over a million points. We then studied the impact of animation on the simplest of encodings with a 69-participant user study. Participants performed better with the animated SPLOMs, found them easier to interpret and *not* too distracting, and preferred animated versus static SPLOMs.

ACKNOWLEDGMENTS

This work was funded by the Faculty Development Fund from the University of San Francisco.

REFERENCES

1. Daniel Archambault and Helen C. Purchase. 2016a. Can Animation Support the Visualisation of Dynamic Graphs? *Information Sciences* 330, Supplement C, SI Visual Info Communication (February 2016), 495–509. DOI: http://dx.doi.org/10.1016/j.ins.2015.04.017

2. Daniel Archambault and Helen C. Purchase. 2016b. On the Effective Visualisation of Dynamic Attribute Cascades. *Information Visualization* 15, 1 (January 2016), 51–63. DOI:http://dx.doi.org/10.1177/1473871615576758

3. Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. 2014. GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks. *IEEE Transactions on Visualization and Computer Graphics* 20, 5 (May 2014), 740–754. DOI: http://dx.doi.org/10.1109/TVCG.2013.254

4. Sven Bachthaler and Daniel Weiskopf. 2008. Continuous Scatterplots. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (November 2008), 1428–1435. DOI:http://dx.doi.org/10.1109/tvcg.2008.119
5. Lyn Bartram, Colin Ware, and Tom Calvert. 2003. Moticons: Detection, Distraction and Task. *International Journal of Human-Computer Studies* 58, 5 (May 2003), 515–545. DOI: http://dx.doi.org/10.1016/S1071-5819(03)00021-1
6. Patrick Baudisch, Desney Tan, Maxime Collomb, Dan Robbins, Ken Hinckley, Maneesh Agrawala, Shengdong Zhao, and Gonzalo Ramos. 2006. Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 169–178. DOI: http://dx.doi.org/10.1145/1166253.1166280
7. Benjamin B Bederson and Angela Boltman. 1999. Does Animation Help Users Build Mental Maps of Spatial Information?. In *Proceedings of the 1999 IEEE Symposium on Information Visualization (INFOVIS '99)*. IEEE Computer Society, Washington, DC, USA, 28–35.
8. Enrico Bertini and Giuseppe Santucci. 2004. Quality Metrics for 2D Scatterplot Graphics: Automatically Reducing Visual Clutter. In *Smart Graphics*, Andreas Butz, Antonio Krüger, and Patrick Olivier (Eds.). Lecture Notes in Computer Science, Vol. 3031. Springer Berlin Heidelberg, 77–89. DOI: http://dx.doi.org/10.1007/978-3-540-24678-7_8
9. Enrico Bertini and Giuseppe Santucci. 2006. Give Chance a Chance: Modeling Density to Enhance Scatter Plot Quality through Random Data Sampling. *Information Visualization* 5, 2 (20 June 2006), 95–110. DOI:http://dx.doi.org/10.1057/palgrave.ivs.9500122
10. Ingwer Borg and Patrick JF Groenen. 2005. *Modern Multidimensional Scaling: Theory and Applications*. Springer-Verlag New York. DOI: http://dx.doi.org/10.1007/0-387-28981-X
11. Daniel B Carr, Richard J Littlefield, WL Nicholson, and JS Littlefield. 1987. Scatterplot Matrix Techniques for Large *N*. *J. Amer. Statist. Assoc.* 82, 398 (June 1987), 424–436. DOI:http://dx.doi.org/10.2307/2289444
12. John M. Chambers, William S. Cleveland, Beat Kleiner, and Paul A. Tukey. 1983. *Graphical Methods for Data Analysis*. Chapman and Hall/Cole Publishing Company.
13. Haidong Chen, Wei Chen, Honghui Mei, Zhiqi Liu, Kun Zhou, Weifeng Chen, Wentao Gu, and Kwan-Liu Ma. 2014. Visual Abstraction and Exploration of Multi-Class Scatterplots. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (December 2014), 1683–1692. DOI:http://dx.doi.org/10.1109/tvcg.2014.2346594
14. Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. 2013. UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (December 2013), 1992–2001. DOI:http://dx.doi.org/10.1109/TVCG.2013.212
15. William C. Cleveland and Marylyn E. McGill. 1988. *Dynamic Graphics for Statistics* (1st ed.). CRC Press, Inc., Boca Raton, FL, USA.
16. William S Cleveland and Robert McGill. 1984. The Many Faces of a Scatterplot. *J. Amer. Statist. Assoc.* 79, 388 (1984), 807–822.
17. Paul Craig, Jessie Kennedy, and Andrew Cumming. 2005. Animated Interval Scatter-Plot Views for the Exploratory Analysis of Large-Scale Microarray Time-Course Data. *Information Visualization* 4, 3 (21 September 2005), 149–163. DOI: http://dx.doi.org/10.1057/palgrave.ivs.9500101
18. Tuan N. Dang, Leland Wilkinson, and Anushka Anand. 2010. Stacking Graphic Elements to Avoid Over-Plotting. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (28 November 2010), 1044–1052. DOI: http://dx.doi.org/10.1109/tvcg.2010.197
19. David DiBiase, Alan M MacEachren, John B Krygier, and Catherine Reeves. 1992. Animation and the Role of Map Design in Scientific Visualization. *Cartography and Geographic Information Systems* 19, 4 (1992), 201–214. DOI:http://dx.doi.org/10.1559/152304092783721295
20. Michael E. Doherty, Richard B. Anderson, Andrea M. Angott, and Dale S. Klopfer. 2007. The Perception of Scatterplots. *Perception & Psychophysics* 69, 7 (2007), 1261–1272. DOI:http://dx.doi.org/10.3758/bf03193961
21. Martin Eisemann, Georgia Albuquerque, and Marcus Magnor. 2014. A Nested Hierarchy of Localized Scatterplots. In *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE Computer Society, Los Alamitos, CA, 80–86. DOI: http://dx.doi.org/10.1109/sibgrapi.2014.14
22. Geoffrey Ellis and Alan Dix. 2007. A Taxonomy of Clutter Reduction for Information Visualisation. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (November 2007), 1216–1223. DOI: http://dx.doi.org/10.1109/tvcg.2007.70535
23. Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. 2008. Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (November 2008), 1539–1148. DOI: http://dx.doi.org/10.1109/tvcg.2008.153
24. John W. Emerson, Walton A. Green, Barret Schloerke, Jason Crowley, Dianne Cook, Heike Hofmann, and Hadley Wickham. 2013. The Generalized Pairs Plot. *Journal of Computational and Graphical Statistics* 22, 1 (2013), 79–91. DOI: http://dx.doi.org/10.1080/10618600.2012.694762
25. Sophie Engle, James Shearer, Michael Ogawa, Steve Haroz, and Kwan-Liu Ma. 2006. Free Your Data! Cenimation: Visualization for Constrained Displays. (2006).

26. David Feng, Lester Kwock, Yueh Lee, and Russell Taylor. 2010. Matching Visual Saliency to Confidence in Plots of Uncertain Data. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (November 2010). DOI: http://dx.doi.org/10.1109/tvcg.2010.176

27. Michael Friendly and Daniel Denis. 2005. The Early Origins and Development of the Scatterplot. *Journal of the History of the Behavioral Sciences* 41, 2 (2005), 103–130. DOI:http://dx.doi.org/10.1002/jhbs.20078

28. Michael Gleicher, Michael Correll, Christine Nothelfer, and Steven Franconeri. 2013. Perception of Average Value in Multiclass Scatterplots. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (December 2013), 2316–2325. DOI: http://dx.doi.org/10.1109/tvcg.2013.183

29. Amy L Griffin, Alan M MacEachren, Frank Hardisty, Erik Steiner, and Bonan Li. 2006. A Comparison of Animated Maps with Static Small-Multiple Maps for Visually Identifying Space-Time Clusters. *Annals of the Association of American Geographers* 96, 4 (December 2006), 740–753. DOI: http://dx.doi.org/10.1111/j.1467-8306.2006.00514.x

30. David Guilmaine, Christophe Viau, and Michael J. McGuffin. 2012. Hierarchically Animated Transitions in Visualizations of Tree Structures. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. ACM, New York, NY, USA, 514–521. DOI: http://dx.doi.org/10.1145/2254556.2254653

31. Ming C. Hao, Umeshwar Dayal, Ratnesh K. Sharma, Daniel A. Keim, and Halldór Janetzko. 2010. Variable Binned Scatter Plots. *Information Visualization* 9, 3 (21 September 2010), 194–203. DOI: http://dx.doi.org/10.1057/ivs.2010.4

32. Lane Harrison. 2013. Experimentr Framework. (March 2013). https://github.com/codementum/experimentr

33. Mark A. Harrower and Cynthia A. Brewer. 2003. ColorBrewer.org: An Online Tool for Selecting Color Schemes for Maps. *The Cartographic Journal* 40, 1 (2003), 27–37. http://colorbrewer2.org/

34. Christopher Healey and James Enns. 2012. Attention and Visual Memory in Visualization and Computer Graphics. *IEEE Transactions on Visualization and Computer Graphics* 18, 7 (July 2012), 1170–1188. DOI: http://dx.doi.org/10.1109/TVCG.2011.127

35. Jeffrey Heer and George Robertson. 2007. Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (November 2007), 1240–1247. DOI: http://dx.doi.org/10.1109/TVCG.2007.70539

36. Daniel E. Huber and Christopher G. Healey. 2005. Visualizing Data with Motion. In *IEEE Visualization*. IEEE Computer Society, 527–534. DOI: http://dx.doi.org/10.1109/VISUAL.2005.1532838

37. Jean-Francois Im, Michael J. McGuffin, and Rock Leung. 2013. GPLOM: The Generalized Plot Matrix for Visualizing Multidimensional Multivariate Data. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2606–2614. DOI: http://dx.doi.org/10.1109/TVCG.2013.160

38. Ian T Jolliffe. 2002. *Principal Component Analysis*. Springer-Verlag New York. DOI: http://dx.doi.org/10.1007/b98835

39. Daniel A. Keim, Ming C. Hao, Umeshwar Dayal, Halldor Janetzko, and Peter Bak. 2010. Generalized Scatter Plots. *Information Visualization* 9, 4 (21 December 2010), 301–311. DOI:http://dx.doi.org/10.1057/ivs.2009.34

40. Martin Krzywinski and Bang Wong. 2013. Points of View: Plotting Symbols. *Nature Methods* 10, 6 (May 2013), 451–451. DOI: http://dx.doi.org/10.1038/nmeth.2490

41. Dirk J. Lehmann, Georgia Albuquerque, Martin Eisemann, Marcus Magnor, and Holger Theisel. 2012. Selecting Coherent and Relevant Plots in Large Scatterplot Matrices. *Computer Graphics Forum* 31, 6 (September 2012), 1895–1908. DOI: http://dx.doi.org/10.1111/j.1467-8659.2012.03069.x

42. Jing Li, Jean-Bernard Martens, and Jarke J. van Wijk. 2010. A Model of Symbol Size Discrimination in Scatterplots. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2553–2562. DOI: http://dx.doi.org/10.1145/1753326.1753714

43. Justin Matejka, Fraser Anderson, and George Fitzmaurice. 2015. Dynamic Opacity Optimization for Scatter Plots. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2707–2710. DOI: http://dx.doi.org/10.1145/2702123.2702585

44. Adrian Mayorga and Michael Gleicher. 2013. Splatterplots: Overcoming Overdraw in Scatter Plots. *IEEE Transactions on Visualization and Computer Graphics* 19, 9 (September 2013), 1526–1538. DOI: http://dx.doi.org/10.1109/tvcg.2013.65

45. Lauren McCarthy. 2014. P5.js JavaScript Library. (July 2014). http://p5js.org/

46. Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman (Eds.). 51–56. https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf

47. Luana Micallef, Gregorio Palmas, Antti Oulasvirta, and Tino Weinkauf. 2017. Towards Perceptual Optimization of the Visual Design of Scatterplots. *IEEE Transactions on Visualization and Computer Graphics* 23, 6 (2017), 1588–1599. DOI: http://dx.doi.org/10.1109/TVCG.2017.2674978

48. Tamara Munzner, François Guimbretière, Serdar Tasiran, Li Zhang, and Yunhong Zhou. 2003. TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. In *ACM Transactions on Graphics*, Vol. 22. ACM, New York, NY, USA, 453–462. DOI: http://dx.doi.org/10.1145/882262.882291

49. Stephen E Palmer. 1999. *Vision Science: Photons to Phenomenology*. MIT Press.

50. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

51. George Robertson, Kim Cameron, Mary Czerwinski, and Daniel Robbins. 2002. Animated Visualization of Multiple Intersecting Hierarchies. *Information Visualization* 1, 1 (March 2002), 50–65. DOI: http://dx.doi.org/10.1057/palgrave/ivs/9500002

52. George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. 2008. Effectiveness of Animation in Trend Visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (November 2008), 1325–1332. DOI: http://dx.doi.org/10.1109/TVCG.2008.125

53. Claudia Roda. 2011. *Human Attention in Digital Environments*. Cambridge University Press, New York, NY, USA.

54. Sébastien Rufiange and Michael J. McGuffin. 2013. DiffAni: Visualizing Dynamic Graphs with a Hybrid of Difference Maps and Animation. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (December 2013), 2556–2565. DOI: http://dx.doi.org/10.1109/TVCG.2013.149

55. Sébastien Rufiange and Guy Melançon. 2014. AniMatrix: A Matrix-Based Visualization of Software Evolution. In *2014 Second IEEE Working Conference on Software Visualization*. 137–146. DOI: http://dx.doi.org/10.1109/VISSOFT.2014.30

56. Alper Sarikaya and Michael Gleicher. 2018. Scatterplots: Tasks, Data, and Designs. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (January 2018), 402–412. DOI: http://dx.doi.org/10.1109/TVCG.2017.2744184

57. Michael Sedlmair, Tamara Munzner, and Melanie Tory. 2013. Empirical Guidance on Scatterplot and Dimension Reduction Technique Choices. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (December 2013), 2634–2643. DOI: http://dx.doi.org/10.1109/tvcg.2013.153

58. James Shearer, Michael Ogawa, Kwan-liu Ma, and Toby Kohlenberg. 2008. Pixelplexing: Gaining Display Resolution Through Time. In *IEEE Pacific Visualization Symposium (PacificVIS '08)*. IEEE, 159–166. DOI: http://dx.doi.org/10.1109/pacificvis.2008.4475472

59. Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. 2002. Animation: Can it Facilitate? *International Journal of Human-Computer Studies* 57, 4 (October 2002), 247–262. DOI: http://dx.doi.org/10.1006/ijhc.2002.1017

60. Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. 2009. *Dimensionality Reduction: A Comparative Review*. Technical Report TiCC-TR 2009-005. Tilburg University.

61. Stéfan van der Walt, S. Chris Colbert, and Gaël Varoquaux. 2011. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science Engineering* 13, 2 (March 2011), 22–30. DOI: http://dx.doi.org/10.1109/MCSE.2011.37

62. Colin Ware and Robert Bobrow. 2005. Supporting Visual Queries on Medium Sized Node-Link Diagrams. *Journal of Information Visualization* 4, 1 (March 2005), 49–58. DOI:http://dx.doi.org/10.1057/palgrave.ivs.9500090

63. Hadley Wickham. 2009. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. http://ggplot2.tidyverse.org/

64. Loutfouz Zaman, Ashish Kalra, and Wolfgang Stuerzlinger. 2011. The Effect of Animation, Dual View, Difference Layers, and Relative Re-layout in Hierarchical Diagram Differencing. In *Proceedings of Graphics Interface 2011 (GI '11)*. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 183–190. http://dl.acm.org/citation.cfm?id=1992917.1992947