Eurographics Conference on Visualization (EuroVis) 2019
M. Gleicher, H. Leitte, and I. Viola
(Guest Editors)

Volume 38 (2019), Number 3

Designing Animated Transitions to Convey Aggregate Operations

Younghoon Kim[1], Michael Correll[2] and Jeffrey Heer[1]

[1]University of Washington
[2]Tableau Research

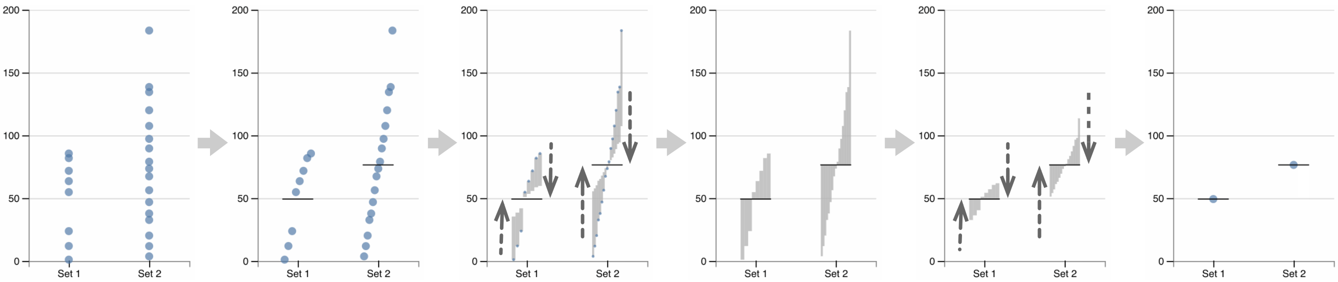

Figure 1: *Staged animated transition for conveying the arithmetic mean (*average*). Individual data points shift and transform to lines to convey residual values. The residual lines then collapse synchronously, such that the upper and lower parts cancel out to form the average.*

Abstract

Data can be aggregated in many ways before being visualized in charts, profoundly affecting what a chart conveys. Despite this importance, the type of aggregation is often communicated only via axis titles. In this paper, we investigate the use of animation to disambiguate different types of aggregation and communicate the meaning of aggregate operations. We present design rationales for animated transitions depicting aggregate operations and present the results of an experiment assessing the impact of these different transitions on identification tasks. We find that judiciously staged animated transitions can improve subjects' accuracy at identifying the aggregation performed, though sometimes with longer response times than with static transitions. Through an analysis of participants' rankings and qualitative responses, we find a consistent preference for animation over static transitions and highlight visual features subjects report relying on to make their judgments. We conclude by extending our animation designs to more complex charts of aggregated data such as box plots and bootstrapped confidence intervals.

CCS Concepts

• ***Human-centered computing*** → *Visualization design and evaluation methods; Information visualization;*

1. Introduction

Animation can help viewers track changes and stay oriented across transitions between related statistical graphics [RMC91, Gon96, BB99, HR07], with research to-date primarily focused on transitions in response to filtering, time steps, changing variables, or adjusting visual encodings [HR07, RFF*08, CDF14, DBJ*11, WASQ18]. However, visual analysis regularly involves summarizing groups of data using aggregation operations such as count, sum, and average. Sarikaya et al. [SGS18] find that aggregation is applied in 74% of the summary visualizations in their survey. Though prior work has depicted count aggregation using a metaphor of sedimentation [HVF13], we lack a more comprehensive treatment of statistical aggregates common to visualization practice.

Visualizations of aggregated data typically indicate the operations performed via axis titles (e.g., "Sum of Profit" or "Average Delay"). While helpful, viewers might overlook such titles or may be unfamiliar with the operations being performed. In this work, we seek to design and evaluate animated transitions that convey aggregation operations, with the goal of reducing ambiguity and imparting a better intuition for the semantics of each operation. Towards this aim, we consider eight common aggregation operations, including both point estimates and measures of spread: *count*, *sum*, maximum (*max*), minimum (*min*), arithmetic mean (*average*), *median*, standard deviation (*stdev*), and interquartile range (*iqr*).

We first contribute novel animation designs for aggregation operations over univariate distributions, transitioning from unaggregated to aggregated dot plots. For each operation we present our

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

design rationale, considering factors of the target concept, staging, axis scale changes, and staggering. We discuss design choices according to these factors and arrive at two staged animation designs for each operation: an *elaborate* design intended to provide a complete impression of the operation, and a simplified *basic* design with fewer animation stages.

Next, we present results from a controlled experiment evaluating these animation designs. We compare against two baseline conditions: *static* (non-animated) transitions and *interpolated* transitions that linearly interpolate between start and end states. Through an initial pilot study on Mechanical Turk, we identify which aggregation operations are most likely to be confused with each other. We then evaluate the four transition conditions with undergraduate students from a visualization course, ensuring subject familiarity with visual analysis tools such as Tableau. We ask subjects to perform binary identification tasks in which they indicate whether or not a presented transition matches a provided operation name.

Our study results indicate that staged animated transitions can improve subject's ability to correctly identify aggregation operations, though sometimes with longer response times than with static transitions. We highlight operations that benefit from *elaborate* staged transitions (such as disambiguating among *average* and *median*, or *stdev* and *iqr*) as well as those that do not (*max* and *min*). Through an analysis of participants' rankings, we find a consistent preference for staged animation designs over the baseline *static* and *interpolated* transitions. From participants' textual responses, we identify the visual features that subjects report using to make their judgments and how they vary across transition types.

Informed by our findings, we describe extensions of our design rationales to more complex charts, such as transitions to depict the construction of box plots, histograms, and means and confidence intervals calculated via bootstrapping. We conclude with a discussion of implications and areas for future work.

2. Related Work

The present work extends prior research on animated transitions in information visualization, focusing on aggregate transitions. Animation is a promising, though sometimes controversial, technique for conveying changes or processes in visualizations. Prior research has found that animation can aid the reconstruction of mental maps [BB99], decision making [Gon96], and staying oriented across transitions [RMC91]. In addition, Jessica el al. [HAS11] identify the condition under which the animation is beneficial. More recently, researchers found that animated transitions can outperform small multiple encodings in a comparison task [OJEF19].

We seek to craft animated transitions of aggregation operations that are uniquely identifiable by depicting the logic of the specific operation performed. In this regard, our design goal is similar to that of *algorithm visualization*, which conveys logical processes to aid understanding. A meta-analysis of algorithm visualization in educational contexts [HDS02] finds that they are "effective insofar as they enable students to construct their own understandings of algorithms through a process of active learning." Though we do not test educational benefits in this work, we allow subjects to play a transition as many times as they like to review what is presented.

However, there are also skeptical views of animation effectiveness, particularly for conveying processes rather than transitions. Tversky et al. [TMB02] criticize studies which show benefits for animation over static graphics, but in which the animated and static versions were not informationally equivalent. They suggest two high-level principles, *Congruence* and *Apprehension*, to promote effective animation designs.

Researchers have investigated animated transitions for statistical data graphics (*e.g.,* bar charts, line charts, and scatter plots). Heer & Robertson [HR07] contribute design guidelines for animated transitions that provide strategies for achieving Tversky et al.'s principles [TMB02]. They evaluate animated transitions using both object tracking and value estimation tasks, finding significant benefits for animated transitions. Robertson et al. [RFF*08] found that, though preferred by participants, animation was less effective than static small multiples for analyzing time-series data. Participants in their experiment preferred animations and in a presentation condition completed tasks faster when using animation. However, they also exhibited more errors when using animation, both in presentation and analysis scenarios. Here we extend prior work by considering the design of transitions for aggregation operations, similarly comparing with both static and simple interpolation conditions.

Other aspects of animated transition design include trajectory paths, staging, and timing. For example, the multi-stage animations evaluated by Heer & Robinson [HR07] were typically preferred by participants but did not significantly outperform single-stage animations. Chevalier et al. [CDF14] studied staggering strategies which use different delays per visual element to reduce occlusion, but found no effect on an object tracking task. Dragicevic et al. [DBJ*11] compared time distortion methods, such as slow-in / slow-out and fast-in / fast-out, and found that slow-in / slow-out enabled users to better track visual objects. Other studies concern improving data point trajectory paths using bundling [DCZL15] or vector field [WASQ18] techniques. Taking this prior work into account, here we develop both elaborate multi-stage transitions and simpler variants for comparison.

In addition to effectiveness questions, transitions induced by aggregation operations may require more complicated designs. For unit visualizations, animated transitions conserve a one-to-one mapping between data and visual elements [PDFE17]. In contrast, aggregation transitions require visual conventions to convey many-to-one mappings of data points to an aggregate value.

Some prior projects have used animation to depict numerical aggregation. Gapminder Trendalyzer [Gap] uses bubble charts to representing the aggregated values (averages and sums) for geographical regions such as continents and countries. When drilling down to a more fine-grained level of detail (*e.g.,* from continent to country), each bubble is subdivided into smaller units. In a related vein, visual sedimentation [HVF13] depicts incoming data streams and their accumulated values by employing the metaphor of a physical sedimentation process. Over time, marks representing individual data points become part of aggregate "strata" of accumulated data. Seeing Theory [DJTD] presents interactive visualizations for basic probability and statistics concepts, including animated calculations of the mean and variance of sampled numbers. In this work, we

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

design and evaluate animated transitions for eight common aggregation functions to transition a 1D dot plot to an aggregate value.

3. Animated Transition Design

We consider eight aggregation operations common to visual data analysis: *count*, *sum*, maximum (*max*), minimum (*min*), arithmetic mean (*average*), *median*, sample standard deviation (*stdev*), and interquartile range (*iqr*). In the case of *stdev*, we focus on showing the range of [-1, +1] standard deviation from the average, rather than a point estimate. For *iqr*, we calculate the quartiles using the R7 method (as used by R and Excel) [R7]. Our primary goal is to design transitions for which viewers can accurately identify which aggregation operation is being performed.

We relied on Heer & Robertson [HR07]'s guidelines for adhering to Tversky et al.'s [TMB02] Congruence and Apprehension principles when designing our animations. We also considered four high-level design factors along which our designs might vary: *target concept*, *staging*, *axis scale*, and *staggering*. We explain our design choices for each of these factors below. Our design process began by prototyping a number of candidate designs and gathering informal feedback from members of our laboratory to inform iterative design. The candidate designs with brief comments are available in our supplement material. With the exception of *max* and *min* — which lend themselves to simpler designs — we arrived at two staged animations per operation: a *staged elaborate* version, corresponding to our full original design, and a *staged basic* version, which collapses some of the stages of the elaborate version.

3.1. Guiding Design Factors

Target Concept. The animations should concretely represent the concept we wish to convey. While one could imagine animations with arbitrary signifiers that would accomplish the task of disambiguation (for instance, points could turn green only when averaging, or blue only when summing), such designs would fail to convey the semantic content of the operation being undertaken. Instead, we wanted our transitions to convey at least some of the mathematical character of the operation being conveyed.

Staging. Rather than a single, complex transition, animations can be divided into a sequence of simpler sub-transitions. Both the choice of sub-transition keyframes and timing — including pauses between stages — are important considerations. Given a short duration, an excessive number of stages may result in too many changes in rapid succession for viewers to reliably follow [HR07]. On the other hand, if there are insufficient stages and one simply interpolates from start to end, the target concept may not be adequately conveyed. Appropriate pauses can add emphasis and prompt consideration of keyframes, while helping avoid an overwhelming experience. Following Heer & Robertson [HR07], we separate major visual changes (*e.g.,* axis transitions or the rearrangement of the data points) into stages with pauses. We combine minor adjustments to color and position into single stages to reduce the total number of stages and so the complexity of the final animation.

Axis Scales. Animations should minimize disruptive changes to the axis scale. Previous research has found that changes of scale can complicate perception of animated transitions, particularly when both axis scales and data points are changing simultaneously [HR07]. Accordingly, in our designs we seek to minimize or, if possible, avoid changes of axis scale and always use separate stages for changes to axis scale and changes to the visualized data. To ensure fair comparisons in our evaluation, we also apply this consideration to our baseline condition that otherwise performs simple linear interpolation.

Staggering. An animation should limit the number of objects that are moving simultaneously. With the exception of *min* and *max*, aggregate operations involve the simultaneous combination of multiple data points. Animations for a single output aggregate value can thus involve the movement of many input points. A natural question, then, is how to stagger or stage such movements, while still ensuring sufficient similarity of movement to give rise to perceptual grouping (the Gestalt principle of "common fate" [Pal99]). We employ different strategies for different aggregation operations, as described below. In addition, aggregates may be calculated separately over multiple groups. With group-by aggregation, we might animate all the groups at once or animate a single group first to ensure a single focus of attention, then synchronously animate the other groups to complete the transition.

3.2. Designs

We now describe our rationales and animation designs for each aggregation operation. Our descriptions focus first on the staged *elaborate* transition, then discuss how we simplify it to produce a staged *basic* version. All transition designs are intended to be comfortably presented over a duration of 2 seconds. Illustrations and specific details on keyframes and relative timing are depicted in Figure 2.

Count. The *count* operation tallies the number of data records in a group. To convey this, we consecutively stack the data points, such that the final height of the stack reflects the result. The points are spaced uniformly to emphasize that each point makes an equal contribution. We separate changes to the axis and to the points by stages: the axis fades out first, the data points are accumulated without the axis, then the new axis fade in. The goal here is to avoid misinterpretation of the point positions. We subdivide the incremental stacking into three sub-stages, such that groups with more points take longer to stack, reinforcing the larger count. For the *basic* version, we instead stack all the points in one stage.

Sum. A *sum* aggregate adds up the values of the data points. To convey an accumulation, our animation design stacks up the values. To emphasize the specific values and differentiate from *count*, we first horizontally offset the points and augment them with line segments whose length encodes the data value. We then stack the lines and grow the axis scale incrementally. We do not fade out the axis, as the final axis represents the same semantic unit, just on a different scale. We also display a supplemental tick that appears at the bottom point and moves to the top as a guide. The *basic* version reduces the visual changes by skipping the horizontal offset.

Max & Min. Maximum and minimum operations have arguably the simplest target concept: they filter the data to only the highest or lowest value. Accordingly, it is sufficient for the animation to convey which extremal point is being selected and fade out the others.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

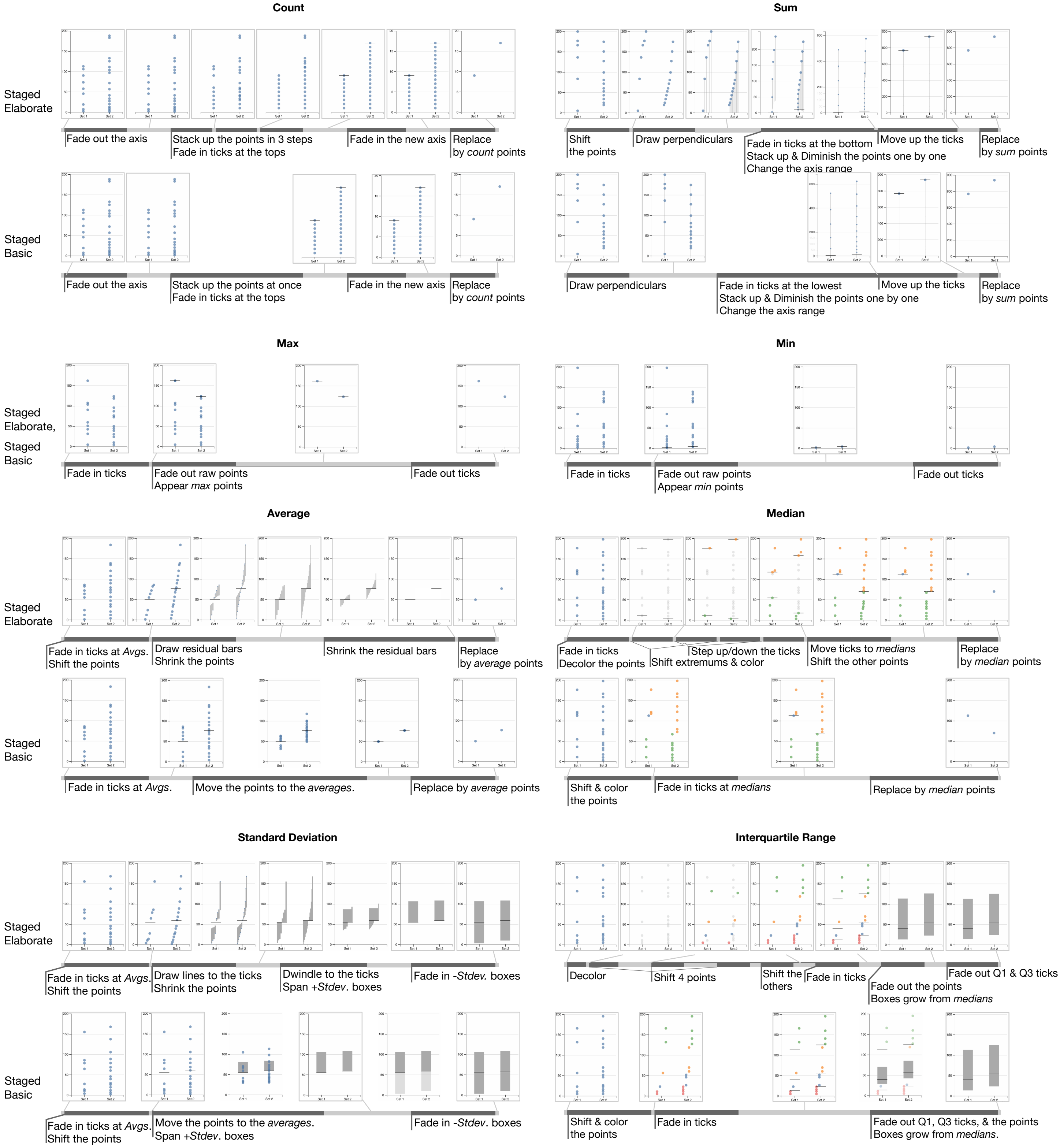

Figure 2: *Designs of animated transitions for 8 aggregate operations with captured frames. Frames with border lines are key frames so that changes happen between them. Each grey line under each series of the frames is a timeline of the animation. The dark grey spans indicate the moves or changes of the graphic objects, and the light grey spans indicate pauses.*

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

Doing so does not require axis scale changes or staggering. As a result of this simplicity, there is no difference between the *basic* and *elaborate* versions for these operations. We perform only minimal staging: we first introduce a tick element to annotate the extremal value, fade out the non-extremal points, then remove the tick.

Average. The arithmetic mean or *average* is a common measure of central tendency for a distribution. One option for depicting an average could be to show how it is calculated, for example, by first animating the sum of all values followed by division by the number of records. However, this is not the only means of calculating averages (*c.f.,* online methods) and would induce potentially jarring changes of the axis scale. Instead, we convey the property that an average is the value at which the residuals (deviations from the mean) sum to zero. Movements are staggered to arrive at the average from above and below simultaneously to signify this symmetry. The *basic* design, in contrast, does not illustrate the residuals.

Median. The *median* is a value that bisects a distribution, such that half the values lie below the median and half lie above, providing a more robust measure of central tendency. Our animation depicts this process by visually segmenting a distribution into lower and upper halves, and then highlighting the median value between the two. The difference between *elaborate* and *basic* designs is the staggering when bisecting. The *elaborate* transition counts each point from the top and bottom synchronously with incremental speed to find the median; the *basic* transition divides them at once.

Stdev. The *standard deviation* is a measure showing the degree of the dispersion. Its mathematical definition involves the square root of the sum of squared residual values. This calculation can be depicted geometrically using square shapes to convey *variance*, as done in Seeing Theory [DJTD]. Nevertheless, we abandoned this method as the two consecutive conversions (points to residual lines to variance components) was too involved and distracting. Instead, we abstract these details by starting with residual lines (initially the same as for the *average* transition) and then collapse them to form the upper range $[\mu, \mu+\sigma]$. We then fade in the lower range ($[\mu-\sigma, \mu]$) to signify that these two spans are symmetric. The *basic* version skips illustrating the residuals.

IQR. The *interquartile range* is the interval from the 25th percentile to the 75th percentile (i.e., spanning the two inner quartiles), indicating where the central half of the data values are distributed. To emphasize the quartile boundaries, we use a similar animation as for *median*, but dividing into four groups instead of the two. After the separation into quartiles, the points fade out and a range grows from the median to inner quartile boundaries. The *basic* transition collapses stages to visualize the quartile separation all at once.

4. Evaluation: Identifying Operations from Transitions

To evaluate our animation designs, we conduct two controlled experiments. The first, a pilot study on Mechanical Turk, was designed to assess the parameters of our identification task, and determine which aggregate measures were difficult to disambiguate. The second, a study with undergraduate computer science students in a visualization tasks, assesses both the quantitative performance and subjective preference of students for using different transition designs to identify aggregate operations. Materials, including stimuli and data tables, are available in the supplemental material and online at https://github.com/uwdata/aggregate-animation-data.

4.1. Stimuli

We compare our staged animation designs with two baseline conditions: *static* and *interpolated* transitions. Static transitions are composed of two static graphics without any animation. Interpolated transitions linearly interpolate marks in the unaggregated charts to their final values. The data points move to the positions of their aggregated values in *count*, *sum*, and *average* operations, and simply fade out in the other operations. Axes fade out and in for *count*, while axis scales change incrementally for *sum*.

All transitions are implemented using D3.js [BOH11] and initiated by clicking a "play" button. For static transitions, the unaggregated chart simply replaces the aggregated chart. Otherwise, the transition plays for 2 seconds. Upon completion, the play button changes to a "reset" button that, if clicked, swaps back to the unaggregated chart and allows subjects to replay the transition.

Each stimulus includes two groups of one dimensional quantitative values, randomly sampled from two different log normal distributions with distinct *average* and *median* values:

$$X_1 \sim \exp(\mathcal{N}(\mu = 5, \sigma = 3))$$
$$X_2 \sim \exp(\mathcal{N}(\mu = 5, \sigma = 4))$$

We sample differing numbers of data points ($N_1 \in \{6,7,8,9\}$, $N_2 \in \{12,13,...,17\}$) to ensure distinct *count* values. We also constrain the sampled distributions such that $\mu_{sample} > \sigma_{sample}$, ensuring the *stdev* range does not cross zero.

4.2. Pilot Study

To inform subsequent evaluation, we first conducted a pilot study using Amazon Mechanical Turk. While Mechanical Turk provides a convenient platform, we were concerned that this subject population was unlikely to match the expertise of people who regularly conduct analyses or consume analysis results. Nevertheless, we wanted to get an initial sense of subjects' familiarity with and confusion among aggregation operations. To do so, we employed both *comprehension* and *identification* tasks.

In a comprehension task, we first show an example transition of an aggregate operation and then show an unaggregated chart alongside four aggregated charts. The subject is asked to select the aggregated chart that correctly depicts the result of applying the aggregation operation to the unaggregated data. The three incorrect choices show aggregate values that differ by 10%, 20%, or 30% from the correct value. The primary purpose of this task is to introduce the aggregation operations and spur critical engagement from participants. After a block of comprehension tasks, participants completed a block of identification tasks, in which they are shown an aggregate transition and asked to identify which operation was performed. For both task types, participants are required to play a transition at least once before responding, but are then free to replay it as much as they like.

After completing an initial survey, participants completed 32

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

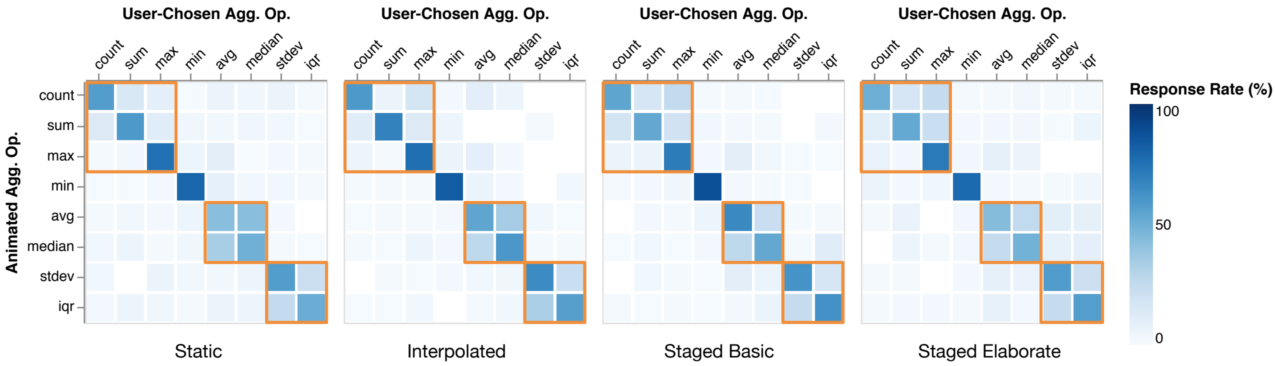

Figure 3: *Confusion matrices for pilot study judgments of transition / operation correspondences. Values along the diagonal indicate correct responses, all others are incorrect. For each transition style, we consistently find three groups of aggregate operations that are confused with each other as shown in orange squares;* $\{count, sum, max\}$, $\{average, median\}$, *and* $\{stdev, iqr\}$.

comprehension tasks followed by 32 identification tasks. In each case, participants saw four replications for each of eight aggregation operations, presented in randomized order. The transition style was assigned as a between-subjects variable, so a single participant saw a consistent style throughout their session.

We recruited roughly 30 participants per transition style (31 for *elaborate*) on Amazon Mechanical Turk. Based on participant self-reports, we filtered out subjects who reported color vision deficiencies or unfamiliarity with the concept of an *average*. Participants were paid \$1.45 US dollars. This amount was selected using internal pre-tests indicating a 12 minute session. However, the actual average completion time was 22 minutes (for \$3.95 per hour).

Figure 3 shows the collected responses for the eight-way identification task. From the responses we see a consistent pattern of confusion across the transition styles, with three recurring groups prone to misidentification: $\{count, sum, max\}$, $\{average, median\}$, and $\{stdev, iqr\}$. These groups correspond to commonalities among the operations: the *count*, *sum*, and *max* operations all visually select a “high” value (though in some cases only after initial transitions); the *average* and *median* both involve central tendencies; and *stdev* and *iqr* both depict ranges rather than point estimates.

4.3. Main Experiment

Based on our pilot study, we developed a modified identification task, shown in Figure 4. To simplify the task and streamline the analysis, we adopt a two-alternative forced choice variant: given a transition and the name of an aggregation operation, the subject must indicate whether or not the presented transition matches the purported aggregate operation. Following the confusion matrix in Figure 3, we choose our alternatives (actual vs. purported operation) from among four clusters of “similar” operations: {*count*, *sum*, *max*}, {*min*}, {*average*, *median*}, and {*stdev*, *iqr*}, resulting in $(3\times3)+(1\times1)+(2\times2)+(2\times2) = 18$ questions. The aggregate operation is a within-subjects factor, while the transition type is between-subjects — each participant saw a consistent transition style. Subjects completed four replications for each transition / operation pair, presented in randomized order, for a total of 72 stimuli.

In addition, subjects performed a *ranking* task (Figure 5) that presents all transition styles for each aggregate operation and asks subjects to rank them in order of subjective preference. We then ask subjects to provide the rationale for their rank judgments using a free-form textbox and require they write more than ten words. For each of the eight ranking questions, participants could play the transitions as many times as they liked, but were required to play each transition at least once before assigning ranks.

To ensure sufficient familiarity with aggregate operations and visual analysis, we recruited computer science undergraduate students in a visualization course. All students were familiar with common visual encodings and aggregate statistics, and had experience using visualization tools including Tableau and D3. Participants were compensated with extra credit in their course.

We recruited a total of 84 subjects, and they participated through a website remotely. We filtered out four who reported color vision deficiencies and another three whose identification accuracy was below 50% (worse than chance). We also excluded two responses with completion times of more than an hour. In total, we analyzed data from 77 participants (33 female, 42 male, 2 other) distributed nearly uniformly over transition styles (20 static, 19 interpolated, 18 staged basic, and 20 elaborate).

4.4. Experiment Results

We analyze accuracy, completion time, and transition play count as performance measures for our identification task. We then examine participants’ transition rankings and text rationales to assess user preference and visual strategies. We first investigate overall differences across transition styles and aggregate operations, then examine results for each operation in detail.

We use hierarchical Bayesian models to analyze the three performance metrics. We use weakly-regularizing priors (default choices by the `brms` library in R) and use Bernoulli, shifted log, and Poisson functions as response distributions for accuracy, completion time, and play count, respectively. Each model takes the form:

$$response \sim transition * operation + mismatch \\ + order + (1|subject)$$

That is, we include the *transition* type, the named aggregation

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

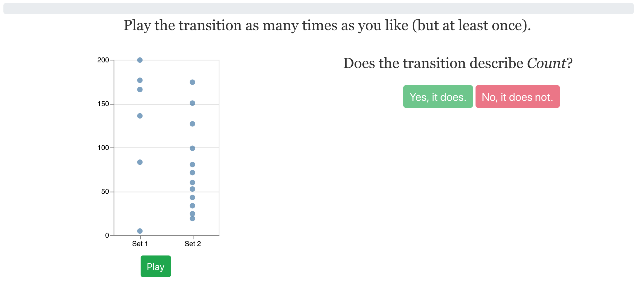

Figure 4: *Identification task interface. Subjects view a chart transition and then indicate whether or not it matches the named aggregation operation.*

Figure 5: *Rank task interface. Subjects view each transition type for a given aggregation operation, rank order them by preference, and provide a textual rationale for their choices.*

operation, and their interaction as fixed effects. In addition, the term *mismatch* is a binary indicator that is true when the transition and purported operation do not match, which we empirically observe leads to higher error — that is, accuracy was higher when the transition did in fact signify the aggregate operation being asked about. The *order* term is the index of the stimulus presentation order and accounts for a mild learning effect. Finally, the $(1|subject)$ term indicates a hierarchical (or *random effects*) term to capture varying performance intercepts per subject.

Qualitative responses from the ranking task were independently coded by two authors. The authors confirmed each others' codes and resolved conflicts via discussion.

4.4.1. Overall Performance

Across transition types, *static* transitions unsurprisingly exhibit higher play count rates across all aggregation operations (see Figure 6), though the strength of this effect diminishes for the simpler *max* and *min* operations. On average, subjects viewed animated transitions only once, but tended to replay *static* transitions one or more times. In terms of rankings (Figure 7), participants consistently prefer the staged (*basic* and *elaborate*) transitions, followed by *interpolated*, and finally *static* transitions.

Across aggregate operations, *max* and *min* exhibit the highest identification accuracy, followed by the two accumulations (*count*, *sum*), then the two central tendencies (*average*, *median*), and finally the two measures of variability (*standard deviation*, *interquartile range*) (Figure 6).

Comparisons of accuracy and completion times for transition styles pooled across aggregation operations should be cautious because there is the strong effect of aggregation operation type, and transition designs are very different across the operations. Instead, we compare these performance measures for transition types separately for each aggregation operation below. Figures 6 and 7 present performance and ranking results by aggregation operation.

4.4.2. Count Performance

For *count* aggregation, all transition styles perform similarly in terms of accuracy. In terms of completion time, the *static* and *elaborate* transitions are faster ($\Delta_\mu \sim 1$), but with 95% credible intervals that overlap those of the *interpolated* and *basic* transitions. As shown in Figure 7, subjects prefer the *basic* transition, followed by the *elaborate* transition. Participants who selected the staged *basic* transition specifically praise the visual metaphor of stacked points: *"we can see the number of dots lined up nicely making it obvious there was a count"*, *"really conveys the idea of counting obects[sic] by placing points equidistant from each other signifying that we don't care about their actual value."* The *elaborate* transition is the next most preferred, though its complexity is mentioned as detracting from its effect: *"similar to the [basic], but has some useless and meaningless animation"*; *"Staged animation is confusing when all the dots look the same and move along the same line."*

4.4.3. Sum Performance

For the *sum* aggregation, all transitions perform similarly in terms of accuracy and completion time. The staged *elaborate* transition is the most preferred, followed by the *interpolated* transition. Participants note that the horizontal offset followed by the accumulation of lines in the staged *elaborate* transition clearly conveyed the notion of sum: *"The lines in the transition I ranked first were important in my mind because it showed that it wasn't just a count, it was summing the individual distances from 0."*; *"The way each data point had a line made it easy to see they all had contribution to the ending point. This made it clear that it was a sum."*

However, those who prefer the *interpolated* transition thought that the collapse of the points to the aggregate value was a good visual metaphor for summation: *"the collapsing transition... made it obvious to me that it was summing"*; *"it seems to be merging the values which is how I usually visualize the summation."*

4.4.4. Max and Min Performance

For the *max* and *min* operations, participants in the *elaborate* and *basic* condition saw the same staged animation. Since these two populations saw different animations for all other operations, the comparisons were slightly different, explaining the differing posterior distributions of identification task performance in Figure 6. Subjects were only asked to rank three animations for these operations, hence the three options in Figure 7. Completion time was

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

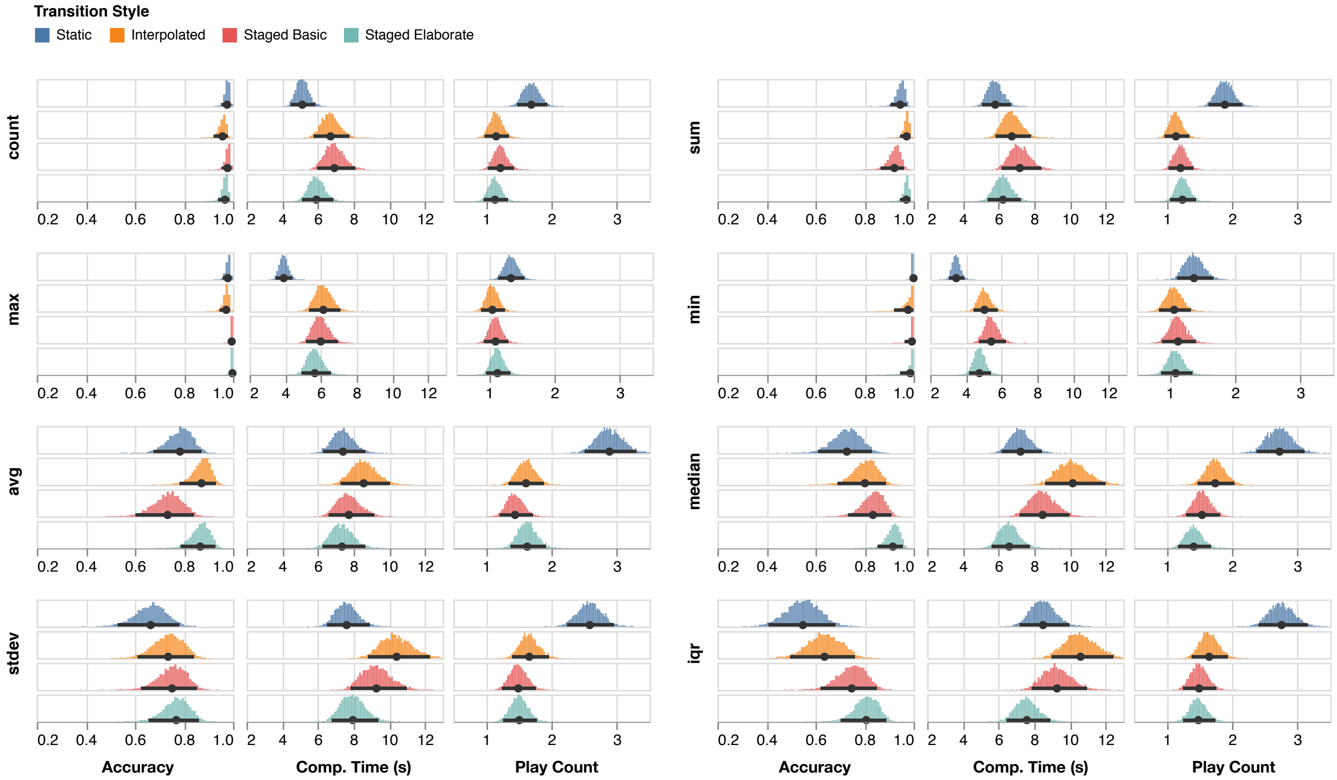

Figure 6: *Posterior probability distributions of expected accuracy, completion time, and play count of each transition style within each aggregate operation. Overlaid points indicate means, and horizontal lines indicate 95% quantile credible intervals.*

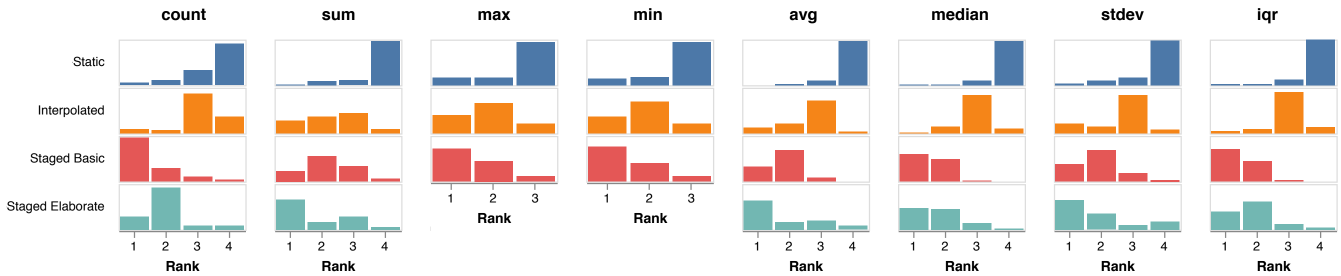

Figure 7: *Rankings of participants' preferred transition style across aggregate operation.*

faster for the *static* transitions ($\Delta_\mu > 1.3$), with 95% credible intervals that do not overlap any of the other transitions. Accuracy was high (mean of greater than 95%) across all conditions, although for *max*, the staged transition outperformed the *interpolated* transition.

While participants prefer the *staged* transition most often, many prefer the *interpolated* transition. The rationales reveal two conflicting opinions. Those who favor the *staged* transition find the tick annotations helpful: *"Showing the line as a new feature draws your eye to the maximum before getting rid of the data which makes it first place."*; *"Clear selection of top value by using a thick line while all points were visible"*. Those who favor the *interpolated* transition were skeptical of the utility of the animated elements in the *staged* transition, and feel that the aggregate operation is easy enough to identify without additional aids: *"I felt that no animation was necessary to show that the topmost element is the maximum of each set."*, *"Same rationale as the maximum, minimum is an easy concept to present and doesn't need that much animation."*

4.5. Average and Median Performance

For *average* and *median* operations, the staged *elaborate* transition meets or exceeds the others in terms of accuracy and completion time. For the *median* operation, the 95% credible interval for the accuracy of the *elaborate* transition is fully separated from that of the *static* transition ($\Delta_\mu = 0.19$). The staged *elaborate* transition is the most preferred for the *average* operation, whereas the *basic* version is most preferred for *median*. For participants who prefer the staged *elaborate* transition for the *average* specifically mention the gray residuals as matching their conception of the target concept, helping with explainability and interpretability: *"for average, it really helps to see the 'area under the curve', to emphasize that*

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

selected line really is an average, supported by equal shaded area (by lines)"; *"The explicit vertical lines show the different weightings by value, which is quite compelling."*

For the *median* operation, participants who prefer the staged *basic* transition like the use of color, but felt that the additional elements in the *elaborate* transition (such as the removal of points one by one), are distracting: *"The colors were incredible helpful. Of the two colored ones, the simpler one was easier to understand at a glance."*, *"The use of color to delineate elements on either side of the median is really helpful, but in this case the additional animations don't provide additional clarity. Perhaps here, color is *so* useful, that the additional steps seem a superfluous distraction."* That said, others did appreciate the per-element animation in the staged *elaborate* transition for the *median* operation, as it lined up with their internal conception of the target concept: *"The first place one is great here – does median the way i think about median, taking off values from each end of the sorted values."*, *"I like the counting motion in [the staged elaborate transition] which is how I usually visualize finding a median."*

4.6. Standard Deviation and Interquartile Range Performance

For *stdev* and *iqr*, the staged *elaborate* transition performs best in terms of accuracy and completion time. It has overlapping 95% credible intervals with the other conditions, with an exception for the *iqr* operation, where it exhibits higher, non-overlapping accuracy relative to the *static* transition ($\Delta_\mu = 0.26$). The staged *elaborate* transition is the most preferred for the *stdev* operation. Participants note that its depiction of the residuals of the average matches their internal target concept: *"they were more clearly calculating an average and then showing the same size range on either side of the average line."*; *"The fourth graphic's discretization makes average very clear, and the standard deviation logically follows."*

The staged *basic* transition is more often preferred for the *iqr* operation, with the staged *elaborate* as the second most preferred. People who favor the *basic* transition appreciate the use of color to identify quartiles in both staged animations as a way of clearly communicating the inter-quartile range, but find the *elaborate* transition needlessly ornate: *"The 4 colors made it really easy to understand that it was splitting it into 4 quarters"*; *"I thought the [staged elaborate transition] added too much animation without it being helpful."*; *"[the staged elaborate transition] just was constant movement and made the transition a little more confusing."*

5. Discussion

We now discuss the effectiveness of animated transitions for aggregation operations, emphasize the importance of conveying a target concept, and extend our transition designs to additional operations.

5.1. Transition Effectiveness varies by Aggregation Operation

The observed effectiveness of the tested transitions strongly depends on the particular aggregation operation. The *max* and *min* operations are accurately and quickly identified regardless of transition type: while animation is preferred, it does not appear to provide performance benefits. For *count* and *sum* operations the performance distributions for each transition type overlap, but with *elaborate* and *static* exhibiting slightly shorter response times (~ 1 sec.). Participants' text responses express appreciation for design differences that disambiguate the two: equidistant stacking for *count* and summation lines for *sum*. Meanwhile, staged *elaborate* transitions exhibit benefits for conveying central tendencies (*average*, *median*) and measures of spread (*stdev*, *iqr*). In these cases, participants value the depiction of residuals (for *average* and *stdev*) and quantile segmentation (for *median* and *iqr*) to convey the target concepts.

5.2. Static Transitions Benefit from Replay

Unlike the animated transitions, static transitions were typically played more than once. This result suggests that participants used the replay feature to aid identification, for example by further comparing start and end states. However, in some real-world scenarios (such as presentations) it may not be feasible for viewers to replay a transition. Our results in favor of animation may thus be conservative, showing static transitions in a stronger light than may actually apply. In addition, we intentionally cue participants attention up front. In more realistic solutions, viewers may not attend to features of the start and end states in sufficient detail prior to a change of view. These observations, in conjunction with the equivalent or better performance of animated transitions, strengthen the argument for the use of appropriately-designed animations.

5.3. Target Concept Congruence Drives Preferences?

Our results provide evidence that user preferences are impacted by how the concept depicted by an animation relates to participants' understanding. As seen in the results and quotes above, subject rationales explain differences among ranking choices in terms of how well the animation aligns with their mental model of the aggregation operation. For the *sum* operation, the *interpolated* transition is ranked highly by some participants, who report interpreting the (upward) coalescing of points as a summation. For the *stdev* operation, multiple participants state that they prefer the *interpolated* transition because it introduces the lower and upper ranges simultaneously, whereas the staged animations show the construction for one side first, then extend it to the other. Echoing Tversky et al. [TMB02], it appears that *congruence* between prior mental models and the animation design correlates with user preference.

5.4. Preferences and Potential Novelty Effects

For the most part, subject preferences either align with the performance results or at least do not contradict them. We observe a consistent preference for the staged animated transitions, then *interpolated* transitions, and lastly *static* transitions. However, we acknowledge that our results may also suffer from a novelty effect. For example, with prolonged use, viewers might prefer shorter or less elaborate transitions. Some rationales specifically acknowledge potential novelty issues: *"I like the last one with the fancy animation, but I do think the second from the left is enough for most cases"*; *"I am not sure what interquartile range with median is, but the 1st one I choose looks [like the] more complicated animation so I like it more."*; *"I like more [feature-]packed animations."*

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

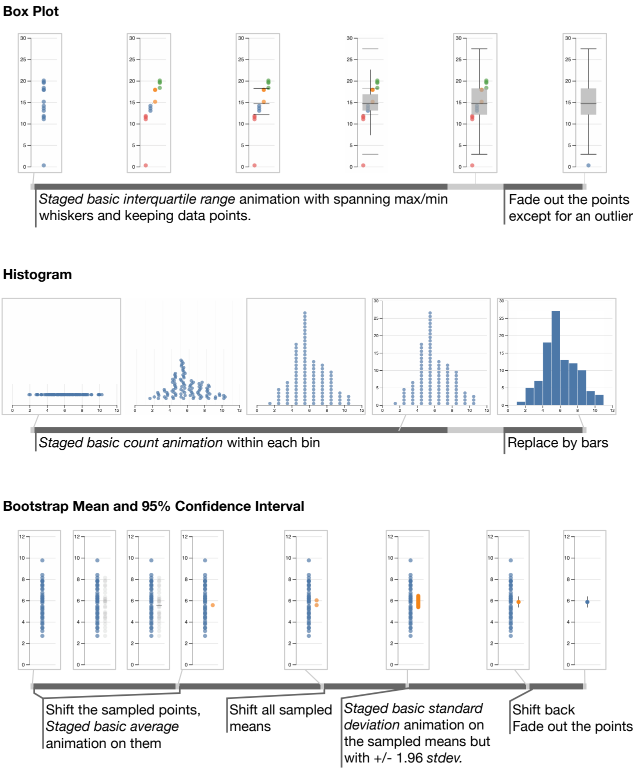

Figure 8: *Animated transition designs for more complex aggregate operations, from a univariate dot plot to a box plot, histogram, or bootstrapped mean and confidence interval.*

5.5. Reusing Animation Designs for Additional Operations

We focused on our eight aggregation operations because they are commonly used and address ambiguities that we believe are likely to arise in practice. That said, we also look upon our animations as reusable "component" designs that can be combined with other animations to convey more complex operations. As shown in the top of Figure 8, we can animate the construction of a box plot by extending our *iqr* animation. A histogram can be created by applying a *count* animation, where the histogram bins define the groups (Figure 8, middle). As a more complex example, bootstrapped means and confidence intervals can be depicted using *average* and *stdev* animations: a sampling plus *average* animation can be applied repeatedly, followed by a *stdev* animation to form a parametric confidence interval over the bootstrapped means (Figure 8, bottom).

5.6. Limitations

Of course, our work also exhibits a number of limitations. Our examples focus on relatively small datasets, involving aggregation of only a dozen or so points per class. The designs we propose may fail to be comprehensible or suffer rendering performance degradation as the number of points increases. That said, we note that this critique is also applicable to existing animation techniques in the literature. Sub-sampling may be one means to address scalability issues, but requires future study and potential refinement. We also focus only on transitions that aggregate univariate data in the form of dot plots. As such we do not explicitly address animation designs concerning other mark types, multivariate plots, or deaggregation (*e.g.,* drill-down). Nevertheless, variants of our presented techniques are applicable in other cases; for instance stacking animations for *count* or *sum* can be applied to other mark types.

This work primarily seeks to support unambiguous identification of aggregation operations, assuming basic statistical literacy on the part of the viewer. To be clear, we do not make any claims regarding the educational value of our animation designs for *learning* about aggregation operations. We leave investigation of potential pedagogical uses to future work. We expect that additional concerns, including longer playback, pauses with descriptive annotations, as well as accompanying text and graphics, might play a role. Moreover, for such cases *animation* itself may not be particularly valuable (c.f. Tversky et al. [TMB02]), but the choice of keyframes underlying our elaborate designs might prove a useful starting point for either animations or static multi-panel explanations.

6. Conclusion & Future Work

We presented animation designs for eight common aggregate operations and conducted a controlled experiment to assess their effectiveness for identifying what operation is being performed. We found that our staged animated transition designs were able to meet or exceed the accuracy of static and simpler interpolated transitions for identifying measures of central tendency (average, median) and spread (standard deviation, interquartile range). In other cases (count, sum, maximum, minimum) static and animated transitions fared similarly. Participants generally prefer our staged animation designs, but in particular prefer those transitions that are congruent with their individual mental model of the operation. Our results extend existing design treatments of animated transitions and provide new evidence of animation effectiveness and how it varies with the specific operation depicted.

Still, many challenges remain. Our proposed animation designs only concern aggregate operations for position encodings of univariate data. Future work might consider techniques for other visual variables and multi-dimensional displays. In addition, techniques and effectiveness studies for large data volumes remain an open problem. For example, our *count* and *sum* animations are not effective with many data points, leading to overplotting when stacked. Regarding evaluation, we focus on using animation to better identify which operation is being performed, and measured accuracy, completion time, and play count alongside subject-reported rankings and rationales. Eye-tracking or other physiological measures might enable a more detailed account of participants' strategies and experienced difficulties. Moreover, future design and evaluation work is needed to consider other measures, such as memorability, and other uses of animation, such as helping to teach aggregation operations to an unfamiliar audience.

7. Acknowledgements

This work was supported by a Moore Foundation Data-Driven Discovery Investigator Award.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

References

[BB99] BEDERSON B. B., BOLTMAN A.: Does animation help users build mental maps of spatial information? In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis'99)* (Oct 1999), pp. 28–35. doi:10.1109/INFVIS.1999.801854. 1, 2

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D^3 data-driven documents. *IEEE Transactions on Visualization & Computer Graphics*, 12 (2011), 2301–2309. doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.185. 5

[CDF14] CHEVALIER F., DRAGICEVIC P., FRANCONERI S.: The not-so-staggering effect of staggered animated transitions on visual tracking. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (Dec 2014), 2241–2250. doi:10.1109/TVCG.2014.2346424. 1, 2

[DBJ*11] DRAGICEVIC P., BEZERIANOS A., JAVED W., ELMQVIST N., FEKETE J.-D.: Temporal distortion for animated transitions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), CHI '11, ACM, pp. 2009–2018. URL: http://doi.acm.org/10.1145/1978942.1979233, doi:10.1145/1978942.1979233. 1, 2

[DCZL15] DU F., CAO N., ZHAO J., LIN Y.-R.: Trajectory bundling for animated transitions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2015), CHI '15, ACM, pp. 289–298. URL: http://doi.acm.org/10.1145/2702123.2702476, doi:10.1145/2702123.2702476. 2

[DJTD] DANIEL K., JINGRU G., TYLER D. D., DANIEL X.: Seeing theory. [Online; accessed 26-March-2018]. URL: https://students.brown.edu/seeing-theory. 2, 5

[Gap] Gapminder. URL: https://www.gapminder.org. 2

[Gon96] GONZALEZ C.: Does animation in user interfaces improve decision making? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1996), CHI '96, ACM, pp. 27–34. URL: http://doi.acm.org/10.1145/238386.238396, doi:10.1145/238386.238396. 1, 2

[HAS11] HULLMAN J., ADAR E., SHAH P.: Benefitting infovis with visual difficulties. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec 2011), 2213–2222. doi:10.1109/TVCG.2011.175. 2

[HDS02] HUNDHAUSEN C. D., DOUGLAS S. A., STASKO J. T.: A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages & Computing 13*, 3 (2002), 259–290. doi:https://doi.org/10.1006/jvlc.2002.0237. 2

[HR07] HEER J., ROBERTSON G.: Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov 2007), 1240–1247. doi:10.1109/TVCG.2007.70539. 1, 2, 3

[HVF13] HURON S., VUILLEMOT R., FEKETE J.: Visual sedimentation. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (Dec 2013), 2446–2455. doi:10.1109/TVCG.2013.227. 1, 2

[OJEF19] ONDOV B., JARDINE N., ELMQVIST N., FRANCONERI S.: Face to face: Evaluating visual comparison. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (Jan 2019), 861–871. doi:10.1109/TVCG.2018.2864884. 2

[Pal99] PALMER S. E.: *Vision Science: Photons to Phenomenology*. MIT press, 1999. 3

[PDFE17] PARK D., DRUCKER S. M., FERNANDEZ R., ELMQVIST N.: Atom: A grammar for unit visualizations. *IEEE Transactions on Visualization and Computer Graphics PP*, 99 (2017), 1–1. doi:10.1109/TVCG.2017.2785807. 2

[R7] Quantile. [Online; accessed 12-December-2018]. URL: https://www.itl.nist.gov/div898/software/dataplot/refman2/auxillar/quantile.htm. 3

[RFF*08] ROBERTSON G., FERNANDEZ R., FISHER D., LEE B., STASKO J.: Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (Nov 2008), 1325–1332. doi:10.1109/TVCG.2008.125. 1, 2

[RMC91] ROBERTSON G. G., MACKINLAY J. D., CARD S. K.: Cone trees: Animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1991), CHI '91, ACM, pp. 189–194. URL: http://doi.acm.org/10.1145/108844.108883, doi:10.1145/108844.108883. 1, 2

[SGS18] SARIKAYA A., GLEICHER M., SZAFIR D.: Design factors for summary visualization in visual analytics. *Computer Graphics Forum 37*, 3 (2018), 145–156. doi:https://doi.org/10.1111/cgf.13408. 1

[TMB02] TVERSKY B., MORRISON J. B., BÉTRANCOURT M.: Animation: can it facilitate? *International Journal of Human-Computer Studies 57*, 4 (2002), 247–262. doi:https://doi.org/10.1006/ijhc.2002.1017. 2, 3, 9, 10

[WASQ18] WANG Y., ARCHAMBAULT D., SCHEIDEGGER C. E., QU H.: A vector field design approach to animated transitions. *IEEE Transactions on Visualization and Computer Graphics 24*, 9 (Sept 2018), 2487–2500. doi:10.1109/TVCG.2017.2750689. 1, 2

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.