Evaluating Animated Transitions between Contiguous Visualizations for Streaming Big Data

Tiago Pereira[1,*] João Moreira[1,†] Daniel Mendes[1,2,‡] Daniel Gonçalves[1,§]

[1] INESC-ID Lisboa, Instituto Superior Técnico, Universidade de Lisboa
[2] INESC-TEC, Faculdade de Engenharia da Universidade do Porto

Abstract

An approach to analyzing Streaming Big Data as it comes in while maintaining the proper context of past events is to employ contiguous visualizations with an increasingly aggressive aggregation degree. This allows for the most recent data to be displayed in detail, while older data is shown in an aggregated form according to how long ago it was received. However, the transitions applied between visualizations with different aggregations must not compromise the understandability of the data flow. Particularly, new data should be perceived considering the context established by older data, and the visualizations should not be perceived as independent or unconnected. In this paper, we present the first study on transitions between two contiguous visualizations, focusing on time series data. We developed several animated transitions between a scatter plot, where all data points are individually represented as they arrive, and other visualizations where data is displayed in an aggregated form. We then conducted a user evaluation to assess the most appealing and effective transitions that allow for the best comprehension of the displayed data for each visualization pair.

Index Terms: Human-centered computing—Visualization—Visualization Techniques; Human-centered computing—Visualization—Empirical Studies in visualization

1 Introduction

Nowadays, there are large amounts of data being constantly generated, from simple personal records to complex networks and financial transactions [13], including sensors and other electronic systems that are always producing more data at smaller intervals. As a result, exploring this vast density of information has become a complex task. Given the importance of the analysis and interpretation of Big Data, visualizations play an essential role, allowing for a better understanding and recognition of interesting patterns, behaviors, and correlations [1]. However, representing extensive amounts of data may constrain users' ability to analyze the entirety of the domain. Likewise, the limited resolution of conventional displays may be insufficient to view all this information. Some approaches tackle these challenged by applying data reduction methods and statistical measures to aggregate information [4, 16], while others provide visualizations designed to take the best of the available screen space [7, 8, 12].

When the data is continuously generated and received, tools that once processed and visualized static information now have to adapt their techniques to follow the evolution of streaming data to generate real-time visualizations. This can make the analysis of these visualizations difficult or even suffer from representations that are too dense, which can lead systems to freeze for certain periods or even crash [3]. A possible approach for visualizing streaming Big Data resorts to the concept of graceful degradation, by using several different visualizations positioned side by side, corresponding at different contiguous time spans [15]. Data can then flow through each one, and more recent data can be presented in more detail. For this, each visualization uses a different technique to aggregate and process information, which allows the most recent data to be presented with full detail, while older data is represented in aggregated form, depending on how long ago it was received. This gives the user a more detailed view of the latest data while giving an overview of the older one, saving memory but keeping its context and patterns visible.

Nonetheless, it is important to ensure that transitions between visualizations are made in such a way that the data flow is understandable, that new data can be interpreted in the context of older data, and that the visualizations are not perceived as independent or unconnected. Indeed, different animations techniques have been proposed to allow users to dynamically see information in distinct ways, the most common being zoom effects [7], clustering [2, 18], and layout algorithms [14]. Some systems apply more complex animations [11] by using target concept, stating, axis scale change and staggering [5]. Bundle movement [6] can also be applied to help guide users' sight into better tracking the important targets. However, existing approaches still struggle to handle large flows of information dynamically. This may be because the goal of existing approaches is to represent small amounts of data or because the systems are not prepared for such dimensions or to create visualizations that have large chunks of information in their composition. The Visual Segmentation Toolkit [10] contributed to create transitions between data items via streaming and well-defined aggregations, but it lacks information regarding data history. Additional research is still needed to help users analyze and explore information while it evolves.

In this paper we propose a set of transitions between visualization techniques targeted at time series to support the analysis of large amounts of information in real-time. In particular, we focus on transitions from a scatter plot where no aggregation exists and all individual data points are rendered as they arrive and other visualizations that use some aggregation strategy, namely heat maps, line charts, stream graphs and bar charts. We then conducted a user evaluation comparing the proposed transitions for each visualization pair. As such, our contributions are: (1) a set of animated techniques for data transitions between contiguous visualizations with different aggregation strategies; (2) a user evaluation to identify the most effective techniques.

2 Transitions between Contiguous Visualizations

To address the challenge of providing a continuous flow of data between contiguous visualizations, we propose a set of transitions between pairs of visualizations. This set is composed of multiple alternatives to create a single and uninterrupted view that joins multiple visualizations organized side by side. Those visualization

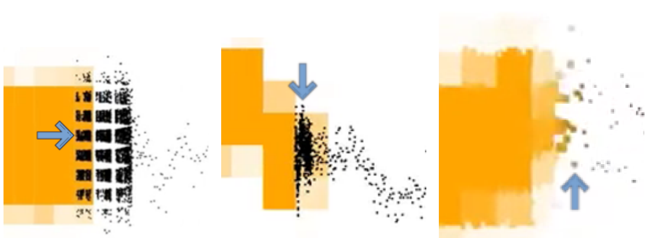
*e-mail: tiago.b.pereira@tecnico.ulisboa.pt
†e-mail: joao.pedro.lavadinho.moreira@tecnico.ulisboa.pt
‡e-mail: danielmendes@ieee.org
§e-mail: daniel.j.goncalves@tecnico.ulisboa.pt

techniques represent data that has been aggregated with some kind of statistical measure. During the development of the proposed transitions, we took into account both congruence and apprehension aspects from Heer and Robertson's design considerations [9]. We always maintain valid visual idioms in regards to congruence. The other considerations are only valid for several transitions between different visualizations, which do not apply to our study as we present alternatives for a single transition between two visualizations, simultaneously. In regards to apprehension, we aimed at: minimizing occlusion (by making each encoding always visible), maximizing predictability (by making sure users can track each data point during each transition), using simple transitions, and making sure transitions took as little time as possible. Grouping similar objects and using staging techniques are not suitable choices because we did not use complex transitions, nor have we more than one transition at a time.

We address data represented with different techniques: scatter plot, line chart, heat maps, stream graph, and bar chart. We used the scatter plot as a technique that presents the raw data, i.e. without aggregation, facilitating the analysis of the most recent data visualized in maximum detail before it gets transferred to the next modules. The line chart was used to represent the average value of data per time interval. The stream graph was used to represent multiple box plots [17] over time, joining the values from each box plot and creating colored areas representing the maximum and minimum values per time interval, and its median and interquartile range. Finally, the heat map and the bar chart were used to group multiple values to measure their density. The first has two options: a normal heat map to represent a matrix relating two variables (time and value intervals) with different tones of colors to show the frequency, and an accumulator heat map that only has one variable, similar to the bar chart, relating value intervals with its frequency.

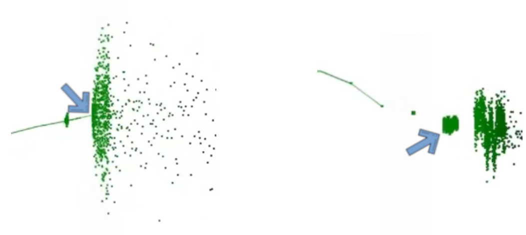
To make transitions more appealing and easier to understand as time evolves, we applied multiple techniques between the scatter plot and the other visualization techniques that can create a single data representation. Note that these are targeted only at data sets with one temporal variable and one quantitative variable.

Scatter plot to heat map Heat map squares are formed after the transition of the points corresponding to the most recent data and according to their density by time interval. Squares get toned depending on how many points there are in that range. The Agglomeration in Squares (**Squares**) technique forms squares of multiple aggregated points to present the points that belong to each square, in addition to **NA** and **Fade** transitions. This technique clusters the squares in small groups, which themselves form a square (shown by the blue arrow, Fig. 1(a)), then moves the points with the correct acceleration and direction, so that each group will fit into the heat map squares. The Data Columns transition (**Columns**) allows users to observe the position of the points relative to the vertical axis and thus a better relation to their values (shown by the blue arrow, Fig. 1(b)). The dots corresponding to the last column of the heat map are decelerated until all of them are between the beginning and the end of that column. The Grainy Transition (**Grainy**) intends to make the dots bigger as time evolves (shown by the blue arrow, Fig. 1(c)) until they have the same size as a heat map square, inducing the creation of these squares gradually.



(a) Agglomeration in **Squares** (b) Data **Columns** (c) **Grainy** Transition

Figure 1: Transitions from scatter plot to heat map (short names in bold). Data is flowing from right to left.

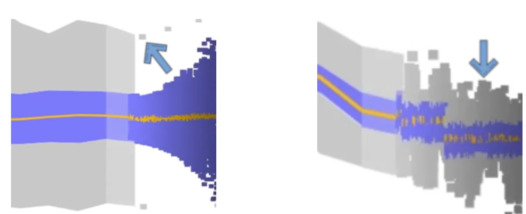


(a) **Bottleneck** (b) Points **Contraction**

Figure 2: Transitions from scatter plot to line chart (short names in bold). Data is flowing from right to left.

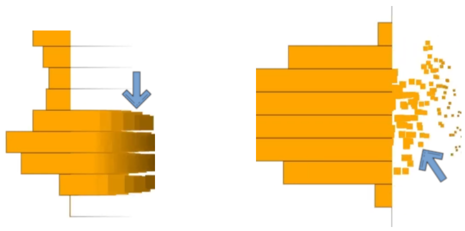
Scatter plot to line chart The line chart is formed after the transition of the points corresponding to the most recent data and according to their average by time interval. The higher the average in a time interval, the higher (relative to the vertical axis) will be the connection point of the line in that interval. Besides the **NA** and **Fade**, we conceived two more transitions. To allow the convergence of the points in multiple intervals and generate a new dot that will later be joined with the line, we created the **Bottleneck** transition. This intends to first accumulate the points that belong to a time interval, to later start converging them to the point that represents the average value of that interval (shown by the blue arrow, Fig. 2(a)). The Points Contraction (**Contraction**) transition contracts each group of points from the previous intervals, concentrating them and forming a shrinking set of points that will end up on a single one (shown by the blue arrow, Fig. 2(b)) representing the average of values on that set and then will merge it with the existing line.

Scatter plot to stream graph The stream graph is formed after the transition of points and illustrates the maximum and minimum values, median, first and third quartiles per time interval. As an alternative to the **NA** and **Fade** transitions, we created the **Narrowing** transition to delimit which points will be relevant for the stream graph areas, converging them into the areas representing the first and third quartile of each interval. The dots are initially slightly increased in size and change color to match the color of the area they represent. The points representing the maximum and minimum values of each interval are represented in the visualization without being converged into the previous areas (shown by the blue arrow, Fig. 3(a)). The **Stamping** transition, contrary to the last technique,



(a) **Narrowing** (b) **Stamping**

Figure 3: Transitions from scatter plot to stream graph (short names in bold). Data is flowing from right to left.



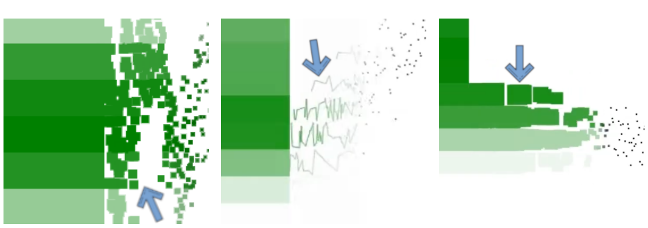
(a) Pilot **Lines** (b) Points **Consumption**

Figure 4: Transitions from scatter plot to bar chart (short names in bold). Data is flowing from right to left.

keeps all points received in the transition and tries to adapt them to the stream graph areas, i.e. changing the colors to correspond with the ones from the stream graph (shown by the blue arrow, Fig. 3(b)). Also, to reduce the visual leap it was decided to gradually increase the area of each point providing slight modifications in their position so that at the time of the transition into the stream graph, the points were aligned with the areas.

Scatter plot to bar chart Each bar from the bar chart is formed and incremented after the transition of points corresponding to each interval of values. Considering that the bar chart was used as an accumulator, the bigger the number of points in that range, the larger the representative bar. In addition to **NA** and **Fade**, we have two more transitions. To allow for better and more accurate tracking of the points while they move into the bars, inducing that they are being routed into the bars, we created the technique Pilot Lines (**Lines**) which firstly increases the size of the points so they end up with the same dimension as the height of the bars from the bar chart (shown by the blue arrow, Fig. 4(a)), reducing the visual leaps during the transition. This technique also uses lines that disappear gradually, providing a pilot line or a route that points the direction of the bars to the dots. With the purpose of moving the points into the bars but keeping their vertical position unchanged before the bars increase and avoiding the visual leap of the direct transition from the points to the bars, we created the Points Consumption transition (**Consuption**) that would cause a slight increase in the points, thus avoiding the formation of large groups that could overlap the others and adapting their color to the color of the bars. Finally, the points are moved into each bar, suggesting an effect of their consumption (shown by the blue arrow, Fig. 4(b)).

Scatter plot to heat map accumulator The heat map accumulator has an infinite time range which leads to one column with multiple intervals on the vertical axis, resembling horizontal bars that have the same behavior as the heat map. These bars change their colors after the transition of the points according to their concentration. We created three transitions besides **NA** and **Fade**. To increase the size of the dots and reduce the visual leap, the Point Forwarding technique (**Forwarding**) intend to route the points into each bar (shown by the blue arrow, Fig. 5(a)) while increasing its size and while they grow toward the corresponding bars, they gradually change their color to the coincident one. Then, the points are shifted into each bar causing a consumption effect on them. The Electrocardiogram transition (**ECG**) divides the data in the heat map accumulator ranges, where for each one the respective points interconnect, forming as many lines (with the color of the corresponding bar) as the number of bars in the heat map. For each interval, the user can see the variation and the peaks in the values before they coincide with the bars (shown by the blue arrow, Fig. 5(b)). Finally, the **Dilation** transition intends to increase the points until they have the same height of the bars and change their colors so they match the color of the interval to which they belong in the heat map (show by the blue arrow, Fig. 5(c)). When there are a lot of points in a row belonging to the same range, a bar forms at the transition.



(a) Point **Forwarding** (b) Electrocardiogram (**ECG**) (c) **Dilation**

Figure 5: Transitions from scatter plot to heat map accumulator (short names in bold). Data is flowing from right to left.

3 User Evaluation

We conducted a set of usability tests to compare each developed transition and understand which techniques were preferred by the participants. The tests included answering a questionnaire, in which participants were asked to answer a set of questions for each transition technique, after watching a video with it. The questions targeted trends and the evolution of data.

Method First, we introduced the questions to the participants. Then, we did a brief introduction to the used visualization techniques, the different modules, and the movement of the data flow. Participants were then encouraged to ask questions until the start of the questionnaire. The questionnaire started with a set of profiling questions and, over the subsequent sections, participants had to watch the video for each transition. Then, they had to answer a set of questions about that transition. Participants were asked to give some suggestions.

Data sets We created multiple data sets using a data generator to avoid their repetition over the various techniques. The data sets were used to create videos of the transitions with a duration of one minute, and were presented abstractly to the participants, i.e. without any semantic information. We used the same duration and number of points for all data sets, but varied trends in a controlled manner. Data could be trending upwards or downwards, oscillating, or steady around a value. Also, incoming data could be continuing the trend in the previously received data or morphing into a different one.

Questions Each set of questions involved four multiple-choice questions on observed trends and two questions that were answered using a Likert scale. The question sets were repeated over the various existing transitions for each pair of visualization techniques. The questions were as follows. **Q1**: How do you compare the latest data with the previously received? (Continued with the same trend / The trend changed); **Q2**: What was happening to the incoming data? (Constant / Oscillating / Increasing / Decreasing); **Q3**: What happened to the previously received data? (Constant / Oscillating / Increasing / Decreasing); **Q4**: What is shown on the left side of the visualization? (Median / Sum / Count / Interquartile Range / Mean / Maximum and minimum). **Q5**: Do you agree that this transition helped to understand the evolution of the data flow? (5-point Likert Scale: 1 - Strongly Disagree, 5 - Strongly Agree). **Q6**: Overall, how do you rank this transition? (5-point Likert Scale: 1 - Awful, 5 - Excellent). **Q7** (at the end of each group of transitions): Please rank each transition worst to best (1 is best).

Participants We had 28 participants (11 female and 17 male), whose ages ranged from 17 to 56 years old (82% between 17-26 and 18% between 51-56). 89% already knew the term Big Data. 10 participants took the test in person, and 18 took it remotely.

4 RESULTS AND DISCUSSION

The initial four questions were counted as right or wrong, and the remaining three, where users provided ratings for each technique, were analyzed according to the associated numerical value. The statistical analysis of the results was performed using Cochran's test with McNemar's post hoc test for the first questions (one categorical dependent variable with two mutually exclusive categories), and Friedman's test with Wilcoxon's post hoc test for the others (one dependent variable measured at the ordinal level and one independent variable consisting of two categorical related groups). In both cases, post hoc tests were applied using the Bonferroni correction. Table 1 summarizes the statistical analysis.

Scatter plot to heat map **Squares** was better than **NA** to compare trends (**Q1**). **Columns** was worse than **Squares** and **Grainy** to identify trends in older data (**Q3**). **Squares** was worse than **Fade** and **Columns** in the rating given by participants for understanding of the data flow (**Q5**), and worse than **Fade** and **Grainy** in the transition classification (**Q6**). **Columns** was better than **NA** in the overall rating (**Q7**). **Summary**: Although there was not a single technique that stood out, the **Fade** and **Grainy** transitions were not significantly worse in any aspect than any other technique.

Scatter plot to line chart Statistical tests revealed that no significant differences existed. **Summary**: All techniques performed similarly.

Scatter plot to stream graph **NA** was worse than **Narrowing** and **Stamping** to identify trends in older data (**Q3**). **Stamping** was the one with the best overall classification, beating all the other transitions (**Q7**). **Summary**: The **Stamping** transition achieved the best results.

Table 1: Summary of the statistical analysis. Showing: χ^2 and p values for Cochran (Q1-Q4) and Friedman's (Q5-Q7) tests; p values for McNemar's post hoc test (Q1-Q4); Z and p values for Wilcoxon's post hoc test (Q5-Q7); success rate (Q1-Q4), median and interquartile range (Q5-Q7) for the techniques in each pair. Only statistically significant results are reported.

	Cochran / Friedman		McNemar / Wilcoxon		
	χ^2(2)	p	Pair	Z	p
Scatter plot to heat map					
Q1	19.088	.001	Squares (93%) - NA (43%)	-	.01
Q3	27.840	<.0005	Columns (32%) - Squares (82%)	-	.02
			Columns (32%) - Grainy (82%)	-	.01
Q5	18.598	.001	Squares (3.5, 3) - Fade (4, 1)	-3.079	.02
			Squares (3.5, 3) - Columns (4, 1)	-3.325	.01
Q6	11.278	.024	Squares (4, 3) - Fade (4, 1)	-2.863	.04
			Squares (4, 3) - Grainy (4, 1)	-2.917	.04
Q7	12.853	.012	Columns (2, 2) - NA (4, 2)	-2.883	.04
Scatter plot to stream graph					
Q3	18.439	<.0005	NA (32%) - Narrowing (75%)	-	.048
			NA (32%) - Stamping (79%)	-	.012
Q7	26.527	<.0005	Stamping (1, 0) - NA (3, 1)	-3.281	.006
			Stamping (1, 0) - Fade (3, 1.75)	-4.167	<.0005
			Stamping (1, 0) - Narrowing (3, 2)	-3.676	.0014
Scatter plot to bar chart					
Q2	14.417	.006	Consuption (82%) - Lines (54%)	-	.048
Q7	18.927	<.0005	Lines (1, 1) - NA (3, 1.75)	-3.460	.006
			Lines (1, 1) - Fade (3, 2)	-3.385	.006
Scatter plot to heat map accumulator					
Q1	16.393	.003	Forwarding (82%) - Dilation (39%)	-	.02
Q7	36.652	<.0005	Forwarding (1, 1) - NA (4, 3)	-4.107	<.0005
			Forwarding (1, 1) - Fade (3, 1)	-4.364	<.0005
			Forwarding (1, 1) - ECG (3, 2)	-4.378	<.0005
			Forwarding (1, 1) - Dilation (3, 2.75)	-4.365	<.0005

Scatter plot to bar chart **Consumption** was better than **Lines** when identifying trends in the most recent data (**Q2**). **Lines** was better than **NA** and **Fade** in the overall classification according to the participant's order of preference (**Q7**). **Summary**: Participants preferred the **Lines** transition, but the **Consumption** transition provided better identification of trends.

Scatter plot to heat map accumulator **Forwarding** was better than **Dilation** to compare trends along the data stream (**Q1**). **Forwarding** was better than the other transitions in terms of the overall ratings given (**Q7**). **Summary**: The **Forwarding** transition was the overall best performer.

Discussion After performing statistical analysis of the answers given by participants to the questions regarding all the transition techniques developed, there were no statistically significant differences for the transitions to line chart. The transitions to heat map were the ones with the largest number of significant differences, but it was not possible to identify a single participants' favorite transition. For the others, the best transitions as preferred by the participants are as follows: **Fade** and **Grainy** for scatter plot to heat map; **Stamping** for scatter plot to stream graph; **Pilot Lines** for scatter plot to bar chart; **Points Forwarding** for scatter plot to heat map accumulator.

Overall, results suggest that the lack of transition or a simple fade are not the best approaches, and that approaches designed to illustrate the transformation of data are more appealing and can help understand data aggregations that are being performed. Indeed, participants agreed that the use of animated transitions allows for a better understanding of the evolution of the data flow. Besides, most animated transition techniques attained better results in trend comparisons than techniques that have no animation, except for the transitions to the line chart.

Some participants reported difficulties in interpreting the statistical measures represented in some visualization techniques used. There were also cases where the visualization technique was not previously known, namely the stream graph, where there was the greatest difficulty in interpreting the statistical measures represented by it. Furthermore, some participants found it more difficult to follow the data from modules where there is a relationship with time to others with visualization where there is no temporal context, such as the bar chart and the heat map accumulator.

5 CONCLUSIONS AND FUTURE WORK

A possible approach to tackle the challenges of visualizing streaming data is to use contiguous visualizations with different aggregation strategies, following a graceful degradation metaphor, which can offer a detailed view of the most recent data while keeping the context of all the received information so far. In this paper, we proposed a set of animated transition techniques that allow users to follow the movement and transformation of the data along with visualizations with different aggregation levels, velocities and time intervals. To test the conceptualized techniques, we conducted a user evaluation with 28 participants. Results showed that thoughtful transitions that illustrate data transformation are more compelling and can help users understand the changes that are happening to data in real time.

As future work, it is relevant to study transitions between visualizations that, while having some aggregation strategy, do so on different levels. Also, it would be interesting to automatically detect changes and/or bizarre events in the data flow, highlighting them to further ease the analysis of the information being received.

ACKNOWLEDGMENTS

This work was partially supported by FCT through projects PTDC/CCI-CIF/28939/2017, UIDB/50021/2020, SFRH/BD/143496/2019 and POCI-01-0145-FEDER-030740 – PTDC/CCICOM/30740/2017. We also would like to thank the reviewers for their valuable input.

References

[1] S. M. Ali, N. Gupta, G. K. Nayak, and R. K. Lenka. Big data visualization: Tools and challenges. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 656–660. IEEE, 2016.

[2] J. Alsakran, Y. Chen, D. Luo, Y. Zhao, J. Yang, W. Dou, and S. Liu. Real-time visualization of streaming text with a force-based dynamic system. *IEEE Computer Graphics and Applications*, 32(1):34–45, 2011.

[3] L. Battle, M. Stonebraker, and R. Chang. Dynamic reduction of query result sets for interactive visualizaton. In *2013 IEEE International Conference on Big Data*, pp. 1–8. IEEE, 2013.

[4] D. M. Best, S. Bohn, D. Love, A. Wynne, and W. A. Pike. Real-time visualization of network behaviors for situational awareness. In *Proceedings of the seventh international symposium on visualization for cyber security*, pp. 79–90. ACM, 2010.

[5] F. Chevalier, P. Dragicevic, and S. Franconeri. The not-so-staggering effect of staggered animated transitions on visual tracking. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2241–2250, dec 2014. doi: 10.1109/tvcg.2014.2346424

[6] F. Du, N. Cao, J. Zhao, and Y.-R. Lin. Trajectory bundling for animated transitions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. ACM Press, 2015. doi: 10.1145/2702123.2702476

[7] F. Fischer, F. Mansmann, and D. A. Keim. Real-time visual analytics for event data streams. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pp. 801–806. ACM, 2012.

[8] M. Hao, D. A. Keim, U. Dayal, D. Oelke, and C. Tremblay. Density displays for data stream monitoring. In *Computer Graphics Forum*, vol. 27, pp. 895–902. Wiley Online Library, 2008.

[9] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, nov 2007. doi: 10.1109/tvcg.2007.70539

[10] S. Huron, R. Vuillemot, and J.-D. Fekete. Visual sedimentation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2446–2455, dec 2013. doi: 10.1109/tvcg.2013.227

[11] Y. Kim, M. Correll, and J. Heer. Designing animated transitions to convey aggregate operations. In *Computer Graphics Forum*, vol. 38, pp. 541–551. Wiley Online Library, 2019.

[12] M. Krstajić, E. Bertini, F. Mansmann, and D. A. Keim. Visual analysis of news streams with article threads. In *Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques*, pp. 39–46. ACM, 2010.

[13] M. Krstajić and D. A. Keim. Visualization of streaming data: Observing change and context in information visualization techniques. In *2013 IEEE International Conference on Big Data*, pp. 41–47. IEEE, 2013.

[14] F. Mansmann, M. Krstajic, F. Fischer, and E. Bertini. Streamsqueeze: a dynamic stream visualization for monitoring of event data. In *Visualization and Data Analysis 2012*, vol. 8294, p. 829404. International Society for Optics and Photonics, 2012.

[15] G. Pires, D. Mendes, and D. Goncalves. VisMillion: A novel interactive visualization technique for real-time big data. In *2019 International Conference on Graphics and Interaction (ICGI)*. IEEE, nov 2019. doi: 10.1109/icgi47575.2019.8955070

[16] J. Traub, N. Steenbergen, P. Grulich, T. Rabl, and V. Markl. I2: Interactive real-time visualization for streaming data. In *EDBT*, pp. 526–529, 2017.

[17] D. F. Williamson, R. A. Parker, and J. S. Kendrick. The box plot: a simple visual method to interpret data. *Annals of internal medicine*, 110(11):916–921, 1989.

[18] J. Xia, F. Wu, F. Guo, C. Xie, Z. Liu, and W. Chen. An online visualization system for streaming log data of computing clusters. *Tsinghua Science and Technology*, 18(2):196–205, 2013.