VisMillion: A novel interactive visualization technique for real-time big data

Gonçalo Pires, Daniel Mendes, Daniel Gonçalves
INESC-ID Lisboa, Instituto Superior Técnico, Universidade de Lisboa
{goncalo.f.pires, danielmendes, daniel.goncalves}@tecnico.ulisboa.pt

***Abstract*—The rapid increase of connected devices causes more and more data to be generated and, in some cases, this data needs to be analyzed as it is received. As such, the challenge of presenting streaming data in such way that changes in the regular flow can be detected needs to be tackled, so that timely and informed decisions can be made. This requires users to be able to perceive the information being received in the moment in detail, while maintaining the context. In this paper, we propose VisMillion, a visualization technique for large amounts of streaming data, following the concept of graceful degradation. It is comprised of several different modules positioned side by side, corresponding to different contiguous time spans, from the last few seconds to a historical view of all data received in the stream so far. Data flows through each one from right to left and, the more recent the data, the more detailed it is presented. To this end, each module uses a different technique to aggregate and process information, with special care to ensure visual continuity between modules to facilitate the analysis. VisMillion was validated through a usability evaluation with 21 participants, as well as performance tests. Results show that it fulfills its objective, successfully aiding users to detect changes, patterns and anomalies in the information being received.**

***Index Terms*—Streaming Big Data, Real-Time, Visualization Technique, Aggregation**

I. Introduction

With the exponential increase of devices capable of generating ever more information, from simple sensors to complex networks and systems, it is necessary to find ways to make the visualization of this Big Data [18] simple and effective, allowing users to do an appropriate interpretation of the data. Otherwise, it may require unnecessary additional effort, users can lose context while detailing on specific periods or events, and they could be misled in their analysis.

When the data that needs to be analyzed is being generated in real time and received in streaming (Streaming Big Data [9]), the associated challenges are even greater, as the data-set cannot be processed *a priori* to create the most adequate visualization. Instead, the visualization needs to be flexible, dynamic and efficiently to deal with the data as it comes in. It should know how to adapt to information that is not available yet, and be able to process it in the limited time available in order to keep the visualization interactive.

Furthermore, existing tools for information visualization resort to idioms that either are able to display all the received information at once, using some aggregation strategy, or show the details of a specific and contained period of time, but not both, due to limitations of resolution and/or available space. However, when monitoring systems or infrastructures for instance, it is relevant to access the most recent data in detail, while having an overview of the past information so that context can be maintained.

To tackle these challenges, we propose VisMillion, a new information visualization technique targeted at time series, that is capable of representing large amounts of data in real-time while allowing users to perceive the global context of the information as new packages arrive. With it, users can get a global overview of the information and detect interesting patterns in recently received data, using multiple visualization modules positioned side-by-side. While flowing between these modules, information will be aggregated over time in such manner that recent data is always represented in full detail. Following a graceful degradation metaphor, older data is gradually stored with less detail to save memory for the system to be scalable. This means that some information will be lost in the process, ensuring that only the essential is stored in order for the user to keep context of what happened in the past. As such, the contributions of this paper are:

1) A novel information visualization technique to deal with Streaming Big Data;
2) A prototype to demonstrate the technique;
3) A user evaluation to validate the effectiveness of VisMillion;
4) A performance evaluation to assess the efficiency of the approach.

The remainder of the paper is organized as follows. Section II summarizes and discusses the state-of-the-art related to real-time visualization of Streaming Big Data. In Section III we detail our novel visualization technique. Section IV describes our implementation of the technique and the resulting prototype. In Section V we present the evaluation of the proposed technique. Finally, we present the conclusions of this work and point out directions for future work in Section VI .

II. Related Work

The challenge of visually exploring large data sets has been a relevant target of research. According to Keim [8], techniques developed to tackle this challenge can be classified according to the data to be visualized, the visualization technique, and the interaction and distortion technique used.

Regarding the data to be visualized, we are particularly interested in one-dimensional data, namely time series data. Time oriented data allows us to take lessons from the past,

and helps analyzing present events and even predict the future. As such, several visualization and interaction techniques have been proposed [1].

The visualization of large amounts of information in the limited space and resolution of a computer display requires strategies to agglomerate data, condensing several data points into a small area or even single pixel, and thus reducing the information needed to be displayed. Of course, these agglomeration strategies only generate overviews that, however good the technique may be, may leave out several relevant details. To further explore the data set, the "Visual Information Seeking Mantra" can be followed: "Overview first, zoom and filter, then details-on-demand" [16].

The recently emerged necessity of dealing with Big Data that is being produced in real time gave rise to additional challenges [5]. In this context, analysts need to be able to detect changes in the data, while maintaining an up-to-date overview, which require increased cognitive effort and is prone to misinterpretation. Naturally, applying sampling and filtering techniques to incomplete data can lead to unwanted results. In addition, to create interactive visualizations of Streaming Big Data, information needs to processed and rendered in very limited time frames.

Some approaches use dimensional reduction to deal with large data-sets, thus reducing the amount of data to render. These processes are commonly performed server-side to also lower the data needed to be transferred. For instance, instead of processing all the values that would fit on a single pixel column, the I^2 [17] environment uses the M4 aggregation technique [7] to generate loss-free plots only resorting to four values per pixel column.

Other existing solutions prefetch data to be presented in order to improve response time. For instance, ForeCache [2] provides detail on demand when users browse a large data set using a lightweight client. Using a tile storage scheme, the server prefetches the most likely tiles users are going to request, considering both users' recent movements as well as data characteristics.

Alternatively, the Event Visualizer [6] processes, analyzes and presents dynamic events in real-time, identifying the more relevant to be presented. After being processed, events are shown using relaxed timelines, allowing events to be visualized in their relative temporal order, granting that all can be equally perceived. When there are too many events to render in an interval, the visualization selects the more interesting events and removes the least interesting ones. Users can then perform pan and zoom actions to interact with the visualization and analyze the different timelines.

To navigate in the large amount of data being presented, some systems, such as LiveRAC [12] and TrendDisplay [4], resort to Semantic Zoom [15]. It starts by providing an overview of the data using a visualization idiom. Then, the user can select a period to analyze in greater detail and, after performing a zoom action, the system will switch the visualization to a different idiom to better present the data corresponding to the chosen interval. This allows big data sets to always be adequately presented within the limited space available, even when changing the focus or the level of detail.

Another possible approach is to use multiple views of the data, as also in LiveRAC [12] and imMens [10]. This requires that, when users interact one view, it communicates with the others so that they can adapt themselves, preserving the context and/or adjust to the available space. For this, techniques such as Stretch and Squish Navigation [12] and Brushing & Linking [10] can be employed.

StreamExplorer [19], using a framework to detect relevant events and cluster data efficiently, eases the visual analysis of the data stream of a social network at three levels. At a macroscopic level, it uses a glyph-based timeline visualization to present a quick overview of the data flow. Then it employs a map visualization to summarize the stream either focusing on topics or geographic information. Lastly, at a microscopic level, it resorts to interactive lenses to allow data exploration according to different perspectives.

To reduce latency when users are interacting with multiple linked visualizations, the recently proposed Falcon [13] utilizes prefetching and indexing strategies. It reindexes data when changes occur in the active view in order to support low-latency brushing actions. Also, to have faster view switching times, it starts by loading smaller resolutions and improves them progressively.

In short, several approaches have been proposed to process and visualize Streaming Big Data, tackling the challenges it poses. Some use aggregation techniques or select the most interesting information to reduce the data being transmitted and rendered. Others start by presenting only an overview of the information, allowing users to zoom in specific intervals, or show simultaneous views of the data, covering different periods or using different aggregation levels. All, however, are always restricted to limited time frames.

In this work, we propose a novel technique that uses several simultaneous visualizations with different idioms to depict all the information received in the stream so far. The visualizations cover contiguous time spans, offering more detail to the more recent data and aggregating older one. The oldest information is shown as an historical view, acting as an overview. This way, context can be maintained, while the data being received at the moment can be thoroughly analyzed.

III. VisMillion

In order to visualize streaming time series data in real-time, we propose VisMillion. This visualization technique presents simultaneously all the data received, instead of being restricted to a specific time window. To make this possible, it follows the concept of a graceful degradation, showing recent data in a more detailed fashion than older data. As the information flows through the interface, from right to left, it is aggregated following different strategies according to how long ago it was received. As such, users can focus on the latest information, presented in detail, without losing awareness of what has already been received.

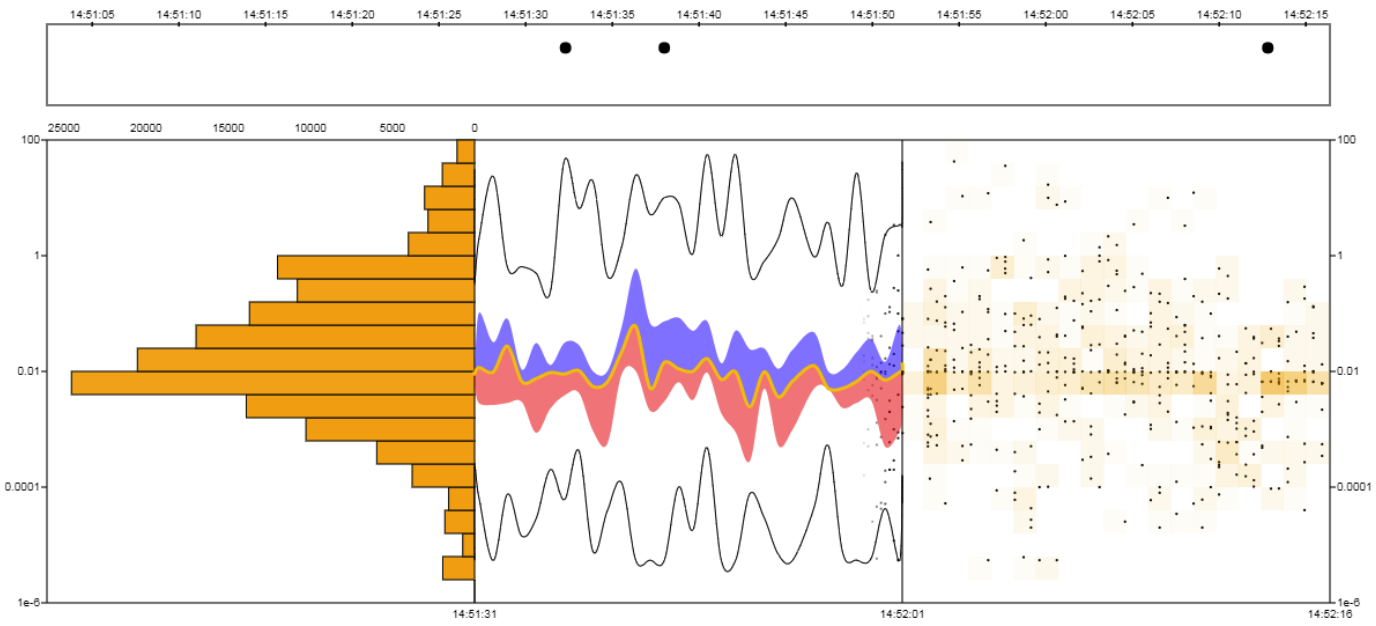

Fig. 1. VisMillion visualization: different levels of detail for the data. Data being received flows from right to left: first a Scatter Plot illustrating a short period (the most recent), then a Stream Graph representing a larger and older period, and finally a Bar Chart agglomerates all older data. On top, outliers are shown.

VisMillion is based on visualization modules, each one presenting data using a different idiom, as depicted in Figure 1. Main modules are arranged side by side, so that data can be understood to flow from one to another, creating a continuous view and connecting all the information received. The time interval covered by each module can be customized. We developed three visualizations for the modules: a Scatter Plot for the most recent data; a Stream Graph, for older data; and a Bar Chart (or histogram) for the oldest information received. While we have chosen to develop these three to illustrate the concept of graceful degradation, other visualizations that may best suit a particular scenario or domain can also be implemented.

In each module, a mouse hover action can be used to trigger a tooltip displaying information of the element below the cursor, as exemplified in Figure 2. Since the displayed data is always moving, the element below a still cursor will change over time. In order to make the information in the tooltip properly perceived, the tooltip is not updated until mouse movement is detected.

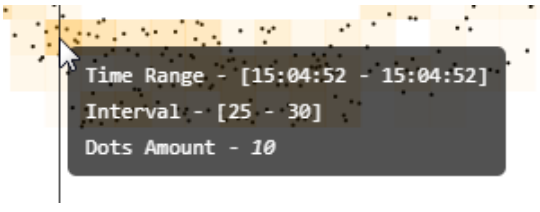

Fig. 2. Cursor hovering the Scatter Plot. A tooltip reporting the information about the hovered area of the heat map is shown.

Therefore, VisMillion was built on top of several concepts from existing literature. By using modules simultaneously showing several visualizations with different levels of detail, it is able to provide useful high information density in context [12]. Starting from left to right, the most usual reading pattern in western countries, it presents first an historic overview of the received data. Then, each module on the right zooms in a more recent period. Then details are shown on demand by activating the tooltip. This is implements the ”Visual Information Seek-ing Mantra” [16]. Lastly, modules are placed in such a way that their relation is encoded by their spatial positioning, which is recommended over other encodings [11].

Additionally, this flow of the data between different visualizations with different levels of aggregation allows to deal with large amounts of data being received in very short periods of time. By gradually aggregating more data, the memory used does not increase continuously and information to be rendered is highly controlled, which makes a system that implements the technique easily scalable. This scalability extends to the visualization itself. Even if the data stream is fairly intensive, we can still show individual datum as they arrive in real-time, because they will be displayed as such only for a limited time window. This window can be defined based on screen real-estate limitations, problem domain semantics, or visual complexity. Then, datum transition to the next visualizations, with increasing aggregation levels, as previously described. The current version of VisMillion has been tested with up to 1000 data points per second with no discerning problems.

A. Scatter Plot

The most recent data received is displayed in a Scatter Plot. All received values are represented through individual points, positioned according to their value and time (Figure 1 right). This is the most detailed visualization, as no aggregation is made. Besides showing the actual data, it also aids to understand the flow at which said data is being received, as the more points exist the greater the flow is.

In addition, a heat map is rendered highlighting areas that have higher concentrations of points, to ease the analysis of patterns and correlations between data. The size of the heat map's squares can be manipulated by the users, in order to aggregate points in an area according to their preference. Also, the opacity of the squares defines their color, and it is set between 0 and 1, corresponding to an interval set by the user. Thus, a more saturated color means that such square contains more data points than another with a less saturated color. If a very high flow of data is received, the information can appear highly cluttered, thus not understandable, and may cause performance issues. In this case, the points representation can be deactivated, leaving only the heat map. This can be triggered by the user, or automatically when a predefined threshold is exceeded.

Hovering the cursor over a squares shows the time and values intervals associated with that area, as well as the number of data points contained in it.

B. Stream Graph

The Stream Graph module performs a first aggregation of the data, allowing it to show a larger period. Although not providing the amount of received packets, this type of visualization can help finding trends in the data flow, through ascents and descents of the average values.

To illustrate the data in this module, we represent the distribution of the data in terms of their concentration, symmetry and the existence of unusual values through the median, minimum, maximum, and first and third quartiles. After splitting the time domain of the chart in several intervals with a predefined duration, the aforementioned values are calculated for each interval. Then, each of these values is connected to the corresponding value of the adjacent intervals, resulting in several lines that originate the final aspect of the module (Figure 1 middle). The areas between the first and third quartiles and the median are colored to make the visualization of this module more appealing and easy to understand. The median line is thicker than the others to indicate the overall median trend. When the data moves from the Scatter Plot to the Stream Graph, a gradual and smooth fading of the points was created, so that the points do not disappear from one moment to another, illustrating the connection between the modules and the graceful degradation of the data. The goal is to help users track where the datum move to, so that context is more easily maintained and analysis facilitated.

In this module, the mouse hover action is active within the colored areas. Here, the tooltip presents information about the minimum, maximum, median, and first and third quartiles of the corresponding interval.

C. Bar Chart

The Bar Chart module accumulates all data older than that shown on the previous two modules. The horizontal bars represent the total amount of data packages received for a given range of values, as shown in Figure 1 left, and its number can be customized by the user. This aggregation of the information can help users understand which data ranges have the most packets received, and therefore which values are most common.

After new data arrives to this module, it needs to be added to a bar. For this, a smooth animation on the bar will be performed, which consists of a blink and a gradual increase of its size. Unlike other modules, the horizontal axis does not represent time but the number of data packages received. To prevent user confusion, values of this axis are presented on the top instead of the bottom.

When the cursor hovers a bar, the data range of that bar and the number of packets received within that range are shown.

D. Outliers

In several scenarios, outliers are atypical values that may require some kind of analysis or intervention. Due to their relevance, we have a module solely dedicated to visualization outliers (as opposed to data outliers) [14]. We consider outliers all values that are above the vertical domain defined by the user.

Since the presence of data in the outliers module should not be as common as in the other modules, we opted to use a simple scatter plot, (Figure 1 top), also flowing right to left. This module is placed at the top of the others in order to maintain context, since the values represented in it are always larger than the domain covered by the visualizations of the other modules. Each outlier is represented by a dot, and has an animation in which its radius is increased and decreased so that it can draw users' attention, allowing them to take due action as soon as possible.

As in the other modules, the mouse hover action displays a tooltip. In this case, when the cursor is over an outlier, the tooltip contains the outlier's timestamp and value.

IV. Prototype

To demonstrate and evaluate VisMillion, we developed a prototype where we implemented it. The system architecture is based on a client-server approach. The client presents the visualization interface to the user, and the server (streamer) sends the data to the interface. Since our main objective is the data representation, we followed a fat-client architecture, where the client is responsible for the data processing.

The client is implemented as a web application. This allows it to run on every computer that has browser support, avoiding OS dependencies. We resorted to the D3 framework [3] to support the creation of the dynamic visualizations. Due to performance limitations of rendering SVG elements, we used

D3 combined with the HTML5 Canvas API. The data received is stored locally and filtered for each component.

The server is implemented as python script. After retrieving the data set from a CSV file, sends each row as a separate data package to the interface via a WebSocket, with the corresponding delay between packages. The use of HTML5 WebSockets make it possible to establish a bidirectional connection between the interface and the server, which can be used in future to define specific requests to the server.

V. Evaluation

To assess whether VisMillion fulfils its objective, we carried out two evaluations. One is focused on usability, while the other targets system performance.

A. Usability

We conducted a user evaluation to test if participants could successfully perceive trends, flows and patterns occurring in the information stream.

1) Method: Each evaluation session was composed of four stages. We started by introducing participants to the experiment and asked them to fill in a profiling questionnaire. Then, we explained the visualization interface, including the charts presented, as well as domain of the data utilized, through a live demonstration. In this stage, participants were encouraged to clarify any doubts they might have. After, participants were asked to analyze data streams in real-time on four different tasks. Lastly, participants fulfilled a final satisfaction questionnaire regarding their preferences about the visualization.

2) Data set and Settings: For this evaluation, we created a data set comprised of sales from a supermarket chain. Each packet sent by the server represents a single sale, containing its time and value. The value's domain ranges from 0 to 100, and some outliers exist above these values.

Regarding the interface, the Scatter Plot and the Stream Graph were set to cover intervals of 15 and 30 seconds, respectively. Although these are somewhat small intervals - in a real life scenario those might have to be larger - we decided to define them so to test the different levels of information representation without making sessions take too long.

3) Tasks: Participants were asked to identify events in a data stream on four tasks. For each event reported, we registered its type and a timestamp. The four tasks were:

1) **Finding Trends** - This task consisted in analysing the variation of the sales average value. Participants should identify patterns and changes in the stream, reporting when they found new periods of increase, decrease or stability. The stream for this task had a duration of about 63 seconds, containing seven periods with distinct trends and, thus, six variations of the trend.
2) **Identifying Outliers** - The second task focused on the analysis of outliers and atypical aggregations, simulating bizarre events in environment being monitored. Participants were asked to identify the values for the outliers shown in the respective module, as well as atypical aggregates that do not fit the main pattern of the received values. For this, four events were generated in a stream of around 62 seconds.
3) **Flow Changes** - In this task, participants needed to recognize changes in the flow of information being received. They were instructed to indicate the moments when an increase or a decrease of the amount of data received per unit of time occurred. The stream had six flow variation events and a duration of 59 seconds.
4) **Open Task** - Differently from the previous tasks, here participants observed a stream and were asked questions while it was being received. During the task, they were asked to identify the sales with the minimum and maximum values received so far At the end, they should be aware of changes in the flow and the average sales value, and able to state the range of values in which there were more sales. Participants were also encouraged to report their grasp of the visualization during the task, using a Think Aloud approach. The data set used had a duration of approximatly 55 seconds.

4) Apparatus and Participants: The evaluation sessions were carried out on a laptop PC with a 15.6" display at a 1366x768 resolution, running Windows 10. The web browser used was Google Chrome.

We counted with the participation of 21 people (17 male, 4 female), whose ages ranged from 22 to 29 years old. 86% reported to use a computer every day. On average, each participant were already acquainted with seven visualization idioms, including bar charts, line charts, scatter plots, pie charts and heat maps, among others.

5) Task Performance: The success rate achieved in identifying the events in the first three tasks is reported in Table I. In the first task, all participants successfully identified three of the six changes in the trend of the data flow (events 1, 3 and 6), while the remaining were detected by at least 86%. On average, participants took 13.15 seconds to report the change since the new period started. The variation in these delays (Figure 3) are related to the fact that participants resorted to different visualization modules for their analysis.

The second task attained a success rate of at least 95% for all events. Since only one participant failed to recognize the outliers and the atypical aggregations, this may have been due to a misunderstanding of the task. The mean delay of detecting one of these events was 8.6 seconds. The constant deviation among events (Figure 4) signifies that the analysis of both outliers and aggregations is similar.

TABLE I
Success rate for each event identification in the first three tasks of the usability evaluation.

	Event					
Task	1	2	3	4	5	6
1	100%	86%	100%	86%	95%	100%
2	95%	100%	95%	95%	-	-
3	100%	100%	95%	100%	90%	95%

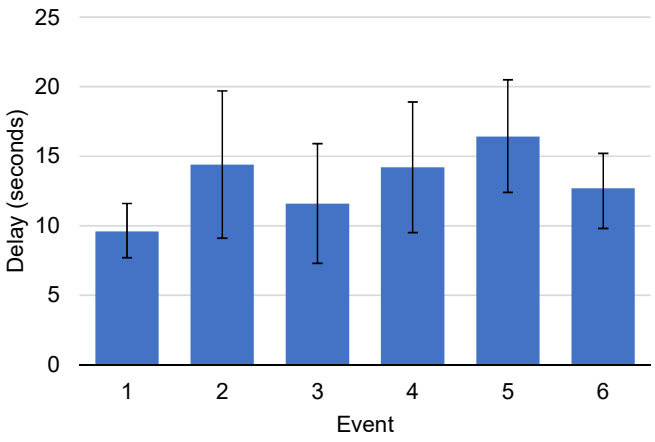

Fig. 3. Delay for participants to detect relevant events in the first task (mean and 95% confidence interval).

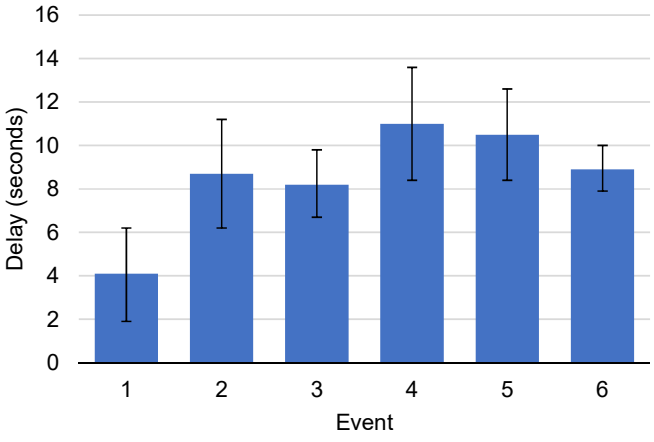

Fig. 5. Delay for participants to detect relevant events in the third task (mean and 95% confidence interval).

The success rate for the third task was always higher than 90% for every event, with three of them having a 100% rate (events 1, 2 and 4). This means that flow changes are easily perceived in the heat map of the Scatter Plot. The time needed to identify each change averaged again 8.6 seconds, and had small variations between participants (Figure 5).

When asked to report maximum and minimum values in the fourth task, 67% of participants correctly identified the minimum and 76% the maximum. The question regarding the range where there were higher sales, all participants answered correctly. However, 90% answered with a very narrow range (between 85 and 90), while two participants reported a wider range (80-100). Regarding an overall analysis of the data set, concerning the flow and mean values, all participants were able to recognize that the flow decreased over time, and 95% noted the decrease in the variation of the mean value of the data.

6) User Preferences: Through the final questionnaire, we asked participants about their experience and satisfaction with VisMillion. The questionnaire consisted of 11 closed questions to which participants had to answer using Likert scales with five values (1 - totally disagree, 5 - totally agree). Answers are reported in Table II. It also had three open questions regarding the strongest and weakest aspects of the visualization, and possible improvements or other comments.

The great majority of participants reported being very easy to notice changes in the flow of data (question 1), identify outliers (question 2) and locate larger aggregations (question 4). The zero interquartile range reveals the homogeneity of these responses. Participants also considered easy to identify variations in mean values (question 3) and ranges in the domain with more data (question 5), as well as understand the transitions between the different modules (question 7), even if with a slightly larger variation in the responses.

Concerning the verification of maximum and minimum values for a specific time within the data stream (question 6), it attained the lowest score. This means that some participants found this operation more difficult to execute in relation to the others. Additionally, and despite being overall well classified,

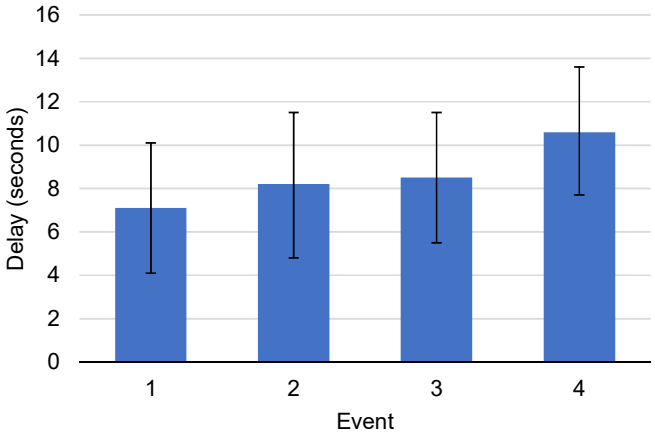

Fig. 4. Delay for participants to detect relevant events in the second task (mean and 95% confidence interval).

TABLE II
PARTICIPANTS' ANSWERS TO THE QUESTIONNAIRE. SHOWING MEDIAN AND INTERQUARTILE RANGE FOR EACH QUESTION.

Was it easy to...	Median (IQR)
1. notice changes in the flow?	5 (0)
2. identify outliers?	5 (0)
3. identify variations in the average values?	5 (1)
4. locate larger aggregations?	5 (0)
5. identify the range of values with more occurrences?	5 (1)
6. identify maximum and minimum values for a given time?	4 (1)
7. understand transitions between modules?	5 (1)
8. perceive the time interval covered by each module?	5 (2)
9. get an overall grasp the system?	5 (1)
10. interact with the system?	5 (1)
11. learn how to use the system?	5 (1)

understanding the time span covered in each module (question 8) had the highest range in participants' responses. This signifies that, against the majority, some participants found this difficult to perceive.

The last three closed questions have very similar values. These allow us to conclude that the visualization technique proposed and the implemented prototype are globally easy to understand, interact with, and learn how to use.

In the open questions, we inquired participants about the biggest difficulties felt and the strongest points of VisMillion, along with suggestions for improving our approach.

Participants mentioned that the mouse hover interaction can be difficult to perform, since the information was always moving. Others revealed some difficulty understanding the transition from the Stream Graph to the Bar Chart, because the bars appeared to grow automatically.

Regarding the positive aspects, it was consensual that the analysis of flow variation and mean values, as well as outlier identification, were easy to perform. Participants reported that the use of the different modules enabled uncomplicated comparisons of the information in different time intervals. Still on the modules, it was stated that they facilitated straightforward analysis of large amounts of data. It was also appreciated that the use of different idioms in the visualizations allowed for different statistical measures, such as averages, quartiles, maxima and minima to be drawn from the same chunks of data.

As for improvements, participants suggested that it would be interesting to add the possibility of creating bookmarks for data points or intervals. This would allow for later analysis of the saved information. Furthermore, it was all participants' opinion that our visualization technique would fit properly into a large-scale event monitoring system.

B. System Performance

Aside from the user evaluation, we also conducted a set of tests to assess system performance. It was our objective to stress the system in order to understand the point where the visualization cannot be further done in real-time, either due to memory consumption or drops in frames per second (FPS).

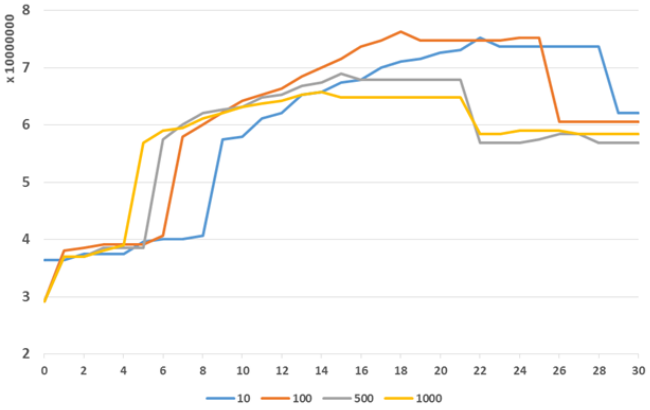

Fig. 7. Memory consumed by VisMillion for the four different flows, with the Scatter Plot rendering all data points. Showing memory (bytes) * time (seconds).

1) Method: With all four modules enabled, and each set to cover a period of 10 seconds, we tested using automatically generated streams with different flows: 10, 100, 500 and 1000 data points per second. Each stream lasted for 30 seconds. We did two trials: in the first we disabled the high flow mode of the Scatter Plot, forcing all data points contained in it to be always rendered; and in the second we enabled it, making it to be rendered only with the heat map when it contained more than 3000 points. The measures observed were FPS and memory used over time. We considered that FPS values below 30 compromise the fluidity of the visualization.

2) Apparatus: The performance evaluation was carried out on the same laptop PC running Windows 10 at a 1366x768 resolution, with an Intel Core i7-3537U CPU and 4 GB of RAM. Performance data collection was facilitated by Google Chrome DevTools.

3) Results: The charts in Figures 6 and 7 present these measures over time for the first trial. The FPS followed a

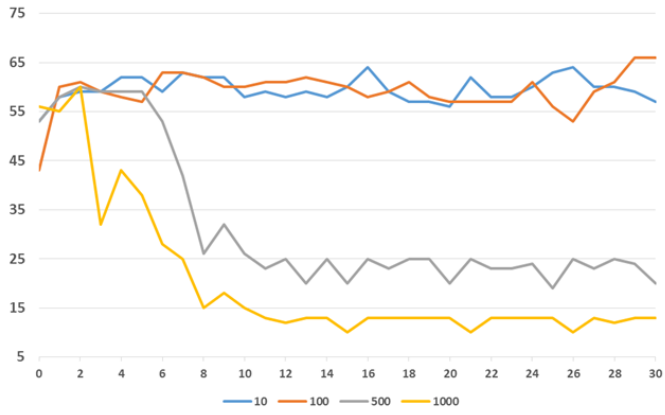

Fig. 6. Frames per second achieved by VisMillion for the four different flows, with the Scatter Plot rendering all data points individually. Showing FPS * time (seconds).

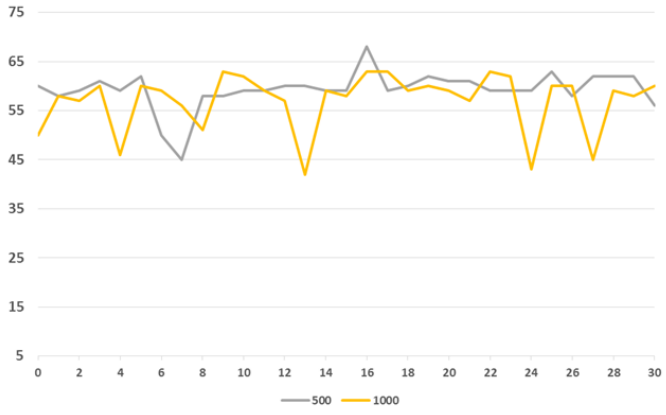

Fig. 8. Frames per second achieved by VisMillion for flows with 500 and 1000 data points per second, with the Scatter Plot rendering only the heat map when over 3000 points. Showing FPS * time (seconds).

stable evolution for flows of 10 and 100 but, for the other flows, the frame rate went below the acceptable threshold after less 10 seconds. As we confirm further down, this was due to the extremely high amount of data points within the Scatter Plot. Regarding memory consumption, we verified that the higher the flow, the faster it increases, as it would be expected. However, after some time, the memory used stabilizes.

In the second trial, we enabled the Scatter Plot to hide individual data points representation when it has more than 3000 to show, rendering only the corresponding heat map. Since we only had issues with the flows of 500 and 1000 packages per second, we run the trial solely for these. The attained results (Figure 8) confirmed that plotting all individual points points significantly slow down the rendering of the module and, consequently, of the whole system as well. Indeed, comparing the new results with the previous ones show that, where less than 10 FPS were reached before, at least 40 FPS were now sustained. As for memory consumption, we observed that it was not affected, indicating that VisMillion is indeed easily scalable.

VI. Conclusions and Future Work

The challenges of intelligibly present large amounts of data are exacerbated when this data is being received in real-time. In this paper, we proposed VisMillion, a novel visualization technique for Streaming Big Data. It aims at allowing users to have satisfactory insights about recently received data and its current flow, while preserving context of past events. Resorting to several visualization modules, trough which the information flows and is aggregated over time, it lets user both focus and have an overview, without needing to change time windows or use explicit strategies.

To validate our approach, we implemented a prototype based on a web application and a custom streamer, and conducted both usability and performance evaluations. The high percentage of participants that could identify the relevant events in the data stream means that they could successfully follow the information across the different visualizations, maintaining context and perceiving newly arrived data in light of that previously received. When tested with high throughput streams of up to 1000 data packets per second, VisMillion uphold constant frame rates and a stable memory consumption, thus confirming the easy scaling of the system.

As future work, we intend to further explore transitions between modules, using smooth animations, in order to make the graceful degradation concept clearer and prevent abrupt changes or the disappearance of elements. Also, as proposed by Khan et al. [9], it would be interesting to use some kind of machine learning approach to detect relevant events or features in the information received. This could then be used to let the visualization adapt itself, highlighting specific events or even changing to an idiom that is better suited for the data in question.

Acknowledgments

This work was supported by FCT, through grants VisBig PTDC/CCI-CIF/28939/2017 and UID/CEC/50021/2019.

References

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

[2] L. Battle, R. Chang, and M. Stonebraker. Dynamic prefetching of data tiles for interactive visualization. In *Proceedings of the 2016 International Conference on Management of Data*, pages 1363–1375. ACM, 2016.

[3] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011.

[4] D. Brodbeck and L. Girardin. Interactive poster: Trend analysis in large timeseries of high-throughput screening data using a distortion-oriented lens with semantic zooming. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis03)*, pages 74–75. Citeseer, 2003.

[5] R. J. Crouser, L. Franklin, and K. Cook. Rethinking visual analytics for streaming data applications. *IEEE Internet Computing*, 21(4):72–76, 2017.

[6] 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*, pages 801–806. ACM, 2012.

[7] U. Jugel, Z. Jerzak, G. Hackenbroich, and V. Markl. M4: a visualization-oriented time series data aggregation. *Proceedings of the VLDB Endowment*, 7(10):797–808, 2014.

[8] D. A. Keim. Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[9] A. M. Khan, D. Gonçalves, D. C. Leão, A. Puig, and T. Isenberg. Towards an adaptive framework for real-time visualization of streaming big data. In *EuroVis 2017-Posters*, pages 13–15. Eurographics Association, 2017.

[10] Z. Liu, B. Jiang, and J. Heer. immens: Real-time visual querying of big data. In *Computer Graphics Forum*, volume 32, pages 421–430. Wiley Online Library, 2013.

[11] J. Mackinlay. Automating the design of graphical presentations of relational information. *Acm Transactions On Graphics (Tog)*, 5(2):110–141, 1986.

[12] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. Liverac: interactive visual exploration of system management time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1483–1492. ACM, 2008.

[13] D. Moritz, B. Howe, and J. Heer. Falcon: Balancing interactive latency and resolution sensitivity for scalable linked visualizations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 694. ACM, 2019.

[14] M. Novotny and H. Hauser. Outlier-preserving focus+ context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, 2006.

[15] K. Perlin and D. Fox. Pad: an alternative approach to the computer interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 57–64. Citeseer, 1993.

[16] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343. IEEE, 1996.

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

[18] J. S. Ward and A. Barker. Undefined by data: A survey of big data definitions. *CoRR*, abs/1309.5821, 2013.

[19] Y. Wu, Z. Chen, G. Sun, X. Xie, N. Cao, S. Liu, and W. Cui. Streamexplorer: A multi-stage system for visually exploring events in social streams. *IEEE transactions on visualization and computer graphics*, 24(10):2758–2772, 2018.