Visualization-aware Timeseries Min-Max Caching with Error Bound Guarantees: Supplementary Material

I. Introduction

This document presents supplementary material supporting our research paper, "Visualization-aware Time series Min-Max Caching with Error Bound Guarantees". Due to space constraints in the main paper, some details were omitted and are included here for comprehensive understanding.

Specifically, we provide detailed algorithms for the error bound calculation and query evaluation, as well as a detailed presentation of the design and results the user study we conducted.

For any queries or further information, feel free to reach out to us at the following emails:

stavmars@athenarc.gr(Stavros Maroulis); bstam@athenarc.gr (Vassilis Stamatopoulos); gpapas@athenarc.gr (George Papastefanatos); mter@athenarc.gr (Manolis Terrovitis);

II. Algorithms

A. Upper Error Bound Evaluation

By considering the definition of the error bound for the visualization and the two theorems related to inner- and inter-column errors, we can calculate the maximum error in a line chart visualization of a variable y within a time series T when it is generated using a grouping $\mathcal{G}^k(T)$. This analysis is conducted through an iterative examination of each pixel column, where we determine its potential inner-column and inter-column pixel errors. The details of this process are presented in Algorithm 1.

Initially, we iterate over the groups within $\mathcal{G}^k(T)$ and determine, for each group B_j^k, whether it overlaps with up to two pixel columns (line 4). We also ascertain whether the group is fully-contained within a single pixel column p_i or partially contained across two adjacent columns, p_i and p_{i+1}. In the case of full containment, we consider the min-max y range of the group and update the corresponding inner-column pixel range P_i with values that we can confidently determine (line 3). For partially contained groups, we adjust the pixel ranges P_r^i and P_l^{i+1} to account for the potential foreground pixels contributed by B_j^k to columns p_i and p_{i+1}, respectively (lines 9 - 10).

After establishing both the sets of correctly rendered inner-column pixels and the potential pixels that could be affected by the left and right partially overlapping groups in each pixel column, we proceed to iterate over each pixel column p_i. Within each of these columns, we determine the potential inner-column pixel errors (line 11) and inter-column pixel errors (lines 13 to 18). Specifically concerning the inter-column errors, for each pixel column and at each of its boundaries with neighboring columns, we rasterize the potentially false inter-column lines that would be rendered using $\mathcal{G}^k(T)$ across the $i-j$ intersection (line 16). Additionally, we calculate the maximum set of pixels that may be missing due to the absence of rendering the correct inter-column lines (line 17). By combining these sets of inner and inter-column errors, we can identify the maximum potential errors within the visualization, enabling us to compute the error bound ϵ (line 22).

The algorithm has a computational complexity of $O(k+w)$, since it iterates over k groups in $\mathcal{G}^k(T)$ and w pixel columns in the visualization.

Algorithm 1: Determining the Upper Error Bound in a Line Chart Visualization

Input: $\mathcal{G}^k(T)$, w, h
Output: Error bound ϵ

$E_{max} \leftarrow 0$
for $i \leftarrow 1$ **to** w **do**
  $P_i \leftarrow \emptyset$ // Initialize the foreground pixel range that can be accurately determined by the fully-contained groups in pixel column p_i
for $B_j^k \in \mathcal{G}^k(T)$ **do**
  Determine the leftmost pixel column p_i that group B_j^k spans
  // Check if B_j^k is fully contained in p_i
  if $B_j^k \subseteq B_i^w$ **then**
    $P_i \leftarrow P_i \cup [p_y(B_j^{min}), p_y(B_j^{max})]$
  else
    // For partial containment, set p_i's right partially overlapping group and the left of its immediate right neighbor
    $P_r^i \leftarrow [p_y(B_j^{min}), p_y(B_j^{max})]$
    $P_l^{i+1} \leftarrow [p_y(B_j^{min}), p_y(B_j^{max})]$
for $i \leftarrow 1$ **to** w **do**
  // Calculate inner-column errors for column p_i
  $E_{inner}^i \leftarrow (P_l^i \cup P_r^i) \setminus P_i$
  // Initialize set of potential inter-column errors
  $E_{inter}^i \leftarrow \emptyset$
  // Calculate inter-column errors for column p_i
  for *each neighbor* $p_j \in \{p_{i-1}, p_{i+1}\}$ **do**
    if p_j *exists* **then**
      Rasterize the line segment between the min or max values of $\mathcal{G}^k(T)$ before and after the intersection between the $i-j$ columns to obtain the set $F_{i,j}$
      Determine the maximum set of pixels that the actual missing correct line across $i-j$ would render to obtain the set $M_{i,j}$
      $E_{inter}^i \leftarrow E_{inter}^i \cup F_{i,j} \cup M_{i,j}$
  // Combine inner- and inter-column errors for column p_i
  $E_{inter}^i \leftarrow E_{inter}^i \setminus P_i$
  $E^i \leftarrow E_{inner}^i \cup E_{inter}^i$
  $E_{max} \leftarrow E_{max} + |E^i|$
$\epsilon \leftarrow \frac{E_{max}}{w \times h}$
return ϵ

B. Query Processing

Next, we provide a detailed query processing algorithm for MinMaxCache. Algorithm 2 is designed to evaluate visualization queries while adhering to error bounds. It takes a query Q and a maximum acceptable error bound ϵ as input and produces query results R for visualization.

In the initialization phase, the algorithm sets up a data structure to store information for each of the w pixel columns and for each variable y in Y_Q (line 1). Specifically, it maintains the fully contained min-max range for each variable y and keeps track of left and right partially overlapped groups for each pixel column and variable.

For each variable y in Y_Q, the algorithm identifies or initializes its corresponding interval tree IT_y (lines 3-5)

Algorithm 2: Query Evaluation in MinMaxCache

Input: $Q = (I_Q, Y_Q, w, h)$: visualization query; ϵ: acceptable error bound
Output: R: query results

Initialize w pixel columns, each maintaining left and right overlapped groups and fully contained min-max range for every $y \in Y_Q$
foreach $y \in Y_Q$ **do**
    Identify the interval tree IT_y based on $\mathcal{T}_{id}$ and y
    if IT_y *does not exist* **then**
        Initialize a new interval tree IT_y
    Find groupings $G^y_{overlapping}$ in IT_y that overlap with I_Q
if $\forall y \in Y_Q, G^y_{overlapping} = \emptyset$ **then**
    `// Complete cache miss`
    Fetch data from the database for the entire interval I_Q using the initial default value for AF
else
    foreach $y \in Y_Q$ **do**
        foreach $G \in G^y_{overlapping}$ **do**
            Update pixel columns based on G
        Calculate partial error bound ϵ'_y based on updated pixel columns for y
        Determine the aggregation factor AF_y, of the overlapping grouping with the largest coverage of I_Q
        $I^y_{missing} \leftarrow$ intervals in I_Q not covered by $G^y_{overlapping}$
        if $\epsilon'_y \leq \epsilon$ **then**
            if $I^y_{missing} \neq \emptyset$ **then**
                `// Partial cache hit for variable` y
                Fetch $I^y_{missing}$ from the database with AF_y
        else
            `// Cache miss for variable` y `because error exceeds bounds`
            Fetch data from the database for the entire interval I_Q for y with $AF'_y = 2 \times AF_y$ or use raw data if $\frac{\tau_{agg}}{\tau_s} < 4$
if *data was fetched from the database* **then**
    foreach $y \in Y_Q$ **do**
        Update pixel columns and IT_y with new groupings from fetched data for y
        Reevaluate ϵ'_y
if $\exists y \in Y_Q, \epsilon'_y > \epsilon$ **then**
    Issue an M4 query for I_Q for all $y \in Y_Q$ with $\epsilon'_y > \epsilon$
Prepare R as a set of points containing the min-max values and the ones with the first and last timestamps from each pixel column for each variable $y \in Y_Q$
return R

and searches within each of them for groupings $G^{y}_{overlapping}$ that overlap the query interval I_Q (line 6).

If no overlapping groupings are found for any variable (line 7), indicating a complete cache miss, the algorithm fetches data from the database for the entire query interval I_Q using the initial default value for the aggregation factor AF (line 8).

If there are overlapping groupings for at least one variable, the algorithm proceeds with partial cache hits or cache misses for each variable individually. For each variable y, it processes its overlapping groupings and updates pixel columns, which includes updating the fully contained min-max y range and tracking left and partially overlapped groups (line 12). It calculates a partial error bound based on the overlapped groups (line 13) and determines an aggregation factor AF_y for the variable by considering the aggregation factor of the overlapping grouping with the largest coverage over the query interval I_Q (line 14).

If a variable's partial error bound is lower than the acceptable error limit and there are missing intervals, this situation constitutes a partial cache hit. In such cases, the algorithm fetches only the missing data for this variable using the aggregation factor AF_y (line 18). However, if a variable's partial error bound exceeds the acceptable limit, indicating a cache miss for this variable, the algorithm retrieves data for the complete interval from the database. This retrieval is done with a doubled aggregation factor or, in cases where the aggregation interval τ_{agg} is very close to the sampling interval τ_s of the time series, raw data is retrieved (line 20). For data fetching, although not explicitly shown in the algorithm for simplicity, a single query to the database is issued to appropriately fetch the necessary data for each variable at the specific granularity needed for each of them.

When data needs to be fetched from the database, the algorithm proceeds to process the retrieved data for each variable. It performs updates on the corresponding pixel columns and reevaluates the error bounds (line 23). Subsequently, the algorithm conducts a verification step to ensure that error bounds are met for all variables. If any variable's error bound falls outside the predefined constraints, the algorithm initiates an M4 query for all variables that do not adhere to these constraints (line 26). In the final step, the algorithm compiles the query results, which include min-max values and timestamps for each variable (line 27), and proceeds to return the result set R (line 28).

III. User Study

To evaluate the efficacy of our caching approach and user reception of error-bounded visualizations, we have conducted a user study focusing on the following objectives: 1) Assessing users' evaluation of our approach's interactive performance versus the error-free approach; 2) Understanding users' perception of differences between error-bounded and error-free visualizations; 3) Determining users' confidence to use approximate, error-bounded visualizations for improving interactivity.

For this purpose, we have developed a prototype visualization tool [1]. that allows users to select a multi-variate time series dataset, choose variables for visualization, and interact with the data through a line chart, as depicted in Figure 1. The chart supports panning and zooming to navigate the results. Additionally, to enable users to assess the performance and accuracy of the results, the tool includes a secondary chart. This chart provides an error-free visualization of the same time range displayed in the first chart, allowing for a direct side-by-side comparison. Users interact with the first chart, and each interaction triggers two requests: one to our system, MinMaxCache, which updates the first chart, and a second that issues an M4 query to the same underlying database for the second chart. The M4 query follows the first, with loading indicators showing the start and end times of the query to fairly assess the comparative performance between the two systems. Furthermore, the execution time of each query is displayed to the user, along with the accuracy guarantees for the error-bounded visualization. These accuracy guarantees are always more than or equal to the accuracy constraint set by the user via a slider. It is noteworthy that, although our presentation for our method considers error bounds, the interface displays accuracy values, which tend to be more intuitive for users.

[1]An online demo is available at https://viz.pulsar.imsi.athenarc.gr/user-study/postgres/visualize/

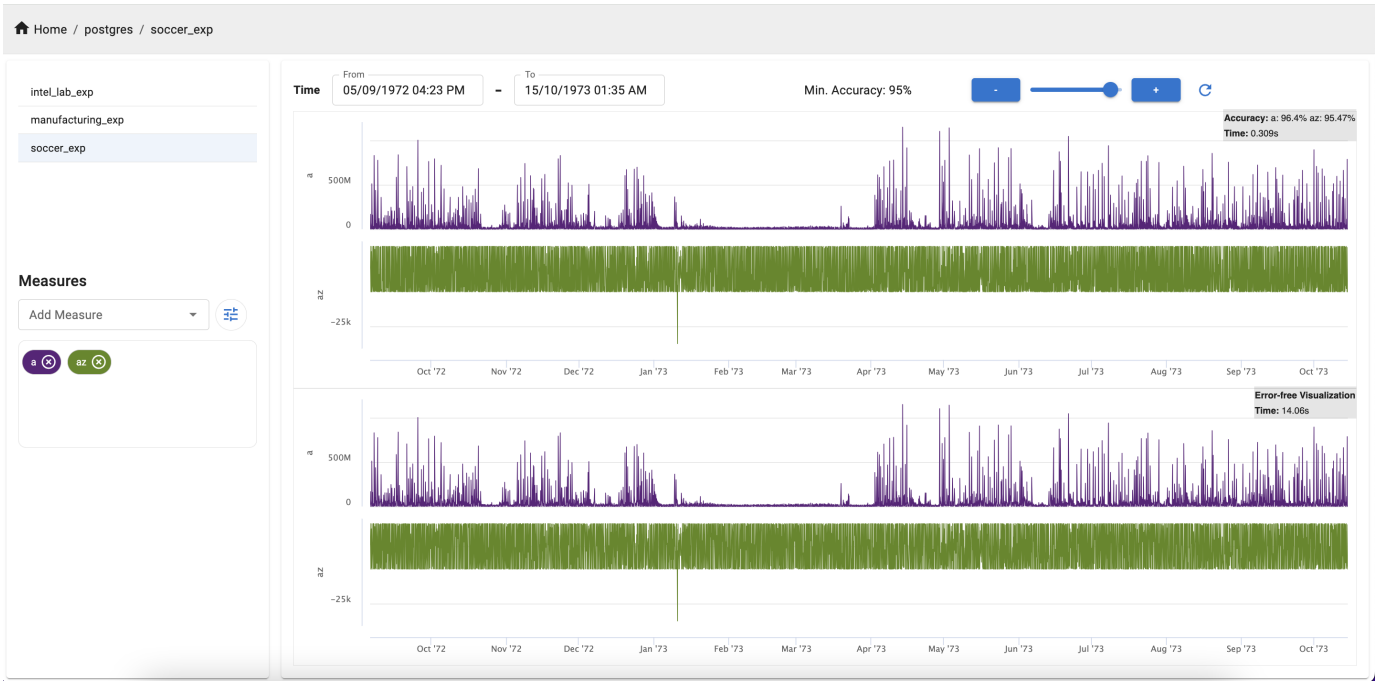

Fig. 1: User Study UI

A. *Study Design*

The user study involved 12 participants, comprising 4 researchers, 3 software engineers, and 5 students from a master's program in Data Science, all with some knowledge of data analytics. Conducted in one-on-one sessions, this setup allowed personalized interviews and a detailed evaluation of our caching mechanism from a cold start, ensuring the cache wasn't influenced by data from other users and providing an unbiased view of its performance.

For the user study, we utilized PostgreSQL as the database backend and the SOCC dataset, which consists of 350 million data points and has a size of 64GB. During the user study, each participant was initially provided with an overview of the user interface and requested to familiarize themselves with it. Subsequently, participants were asked to navigate to a specific period within the time series dataset by interacting with the chart through panning and zooming operations. Following this, they were instructed to select two additional variables for visualization and continue navigating the chart. During this stage of the user study, the accuracy slider was set to a minimum accuracy of 0.95.

Participants provided feedback through Likert scale questions focused on the responsiveness of our approach, MinMaxCache, and the M4 method, which necessitates a new database query with each interaction. Additional questions assessed the perceived accuracy relative to the error-free visualization offered by M4. To qualitatively compare error-bounded and error-free visualizations, participants also performed simple data analysis tasks, including identifying minimum and maximum variable values within the viewport, as well as detecting trends, patterns, and outliers. These tasks aimed to test the precision and reliability of the error-bounded visualizations and determine if the results varied between the two visualization types.

Subsequently, each participant was instructed to adjust the accuracy slider to 0.99 and interact with the chart to visualize data from the previous year. They then answered questions aimed at evaluating the impact of accuracy adjustments on the system's responsiveness and the perceived accuracy of visualizations. These questions sought to understand how a higher accuracy level influenced the perceived responsiveness and accuracy of the visualizations compared to the error-free benchmark. The study concluded with interviews that gathered detailed feedback on their experiences, including suggestions for system improvements and features they found either useful or challenging.

B. User Study Results

1) Quantitative Feedback: The survey results were generally favorable for our method. Participants highly rated the overall responsiveness of the caching approach compared to the error-free approach using M4, with scores indicating a strong consensus on our system's improved performance ($\mu = 4.5$). When asked to assess whether differences between error-bounded and error-free visualizations were noticeable, participants indicated that such differences were perceived as only marginally noticeable ($\mu = 2.8$). Further, upon performing simple analysis tasks such as detecting outliers, determining min/max points, and identifying trends, participants unanimously reported no discernible differences between the error-bounded and error-free visualizations.

Moreover, participants expressed a willingness to accept minor accuracy compromises in favor of greater interactivity ($\mu = 4.1$). However, there was a challenge in interpreting the accuracy guarantees on the approximate visualizations, as highlighted by participants finding this aspect difficult to comprehend ($\mu = 2.3$), an issue that was also discussed further during the interview phase.

Further exploration into user perceptions regarding the impact of adjusting the acceptable accuracy level on visualization responsiveness and accuracy revealed that an increase in the acceptable accuracy level led to noticeable responsiveness degradation ($\mu = 3.9$). Meanwhile, participants generally did not perceive significant improvements in visualization accuracy compared to the error-free reference following such adjustments ($\mu = 2.8$). These findings suggest that increasing the visualization's accuracy level did not substantially enhance perceived accuracy, whereas the impact on responsiveness was more pronounced. This observation may encourage users to trust error-bounded visualizations more for improving interactivity.

2) Qualitative Feedback: Following the survey, participants engaged in brief interviews, complemented by open-ended questions aimed at eliciting more comprehensive feedback. These discussions were structured to understand the challenges participants faced, any comments they had, and the reasons behind their confidence levels or lack thereof in utilizing the error-bounded approach. Additionally, the interviews sought to identify difficulties in understanding the approach and to identify factors that could potentially enhance their trust in using this method for data analysis.

Regarding their confidence in utilizing the error-bounded approach for analysis, most participants expressed willingness but also expressed concerns about relying solely on approximate results for their analyses. Some noted that while the visual differences between error-bounded and error-free visualizations were almost indiscernible, they would prefer the option to verify results against error-free data for assurance. Nevertheless, participants were willing to use the error-bounded approach for rapid, interactive exploration, suggesting they would reserve requests for error-free verification for specific visualizations of interest.

During the interviews, some participants mentioned the challenge of comprehending the accuracy guarantees provided. While they acknowledged that these guarantees could help them trust the visualized output, understanding how these accuracy compromises translated into the visualizations proved difficult—a sentiment anticipated to some extent, given that the accuracy guarantees in our approach are expressed in the visualization domain, not the data domain. To make the concept of accuracy guarantees more comprehensible, a potential solution that emerged involves employing visual cues on the line chart. For instance, highlighting pixels on hover that may contain errors could effectively communicate the extent of possible accuracy compromises.

In a similar context, one participant expressed their concern about determining the minimum acceptable accuracy guarantees (or maximum error bound) in real-life scenarios. They highlighted the challenge of making informed decisions about setting these thresholds without a clear understanding of their impact on the visualization's accuracy and the resulting interactivity. A suggestion that emerged involved the system to automatically adjust the accuracy level to maintain interactive responsiveness, while allowing the user the option to refine and fetch the error-free visualization if the provided accuracy guarantees are not satisfactory to them.

Finally, one participant pointed out a discrepancy in the tooltip information when hovering over data points in the visualizations generated by the error-bounded approach versus the error-free M4 approach. Specifically, they noted differences in the timestamps displayed between the two methods. This observation provides valuable insight into the subtleties of our Min Max aggregation approach, which approximates the timestamps for the minimum and maximum values with the mid-interval timestamp of the corresponding group. In contrast, the M4 method incorporates a join operation in its query to retrieve the exact timestamps for these min-max values. As a result,

visualizations derived from M4 can display precise timestamp information, explaining the noted differences in the tooltip content.

3) Future Work: Based on insights from the user study and the limitations highlighted, several areas for future work have been identified to enhance the usability and effectiveness of our caching approach.

Firstly, to avoid misleading users of the visualization tool and to mitigate the observed discrepancy in tooltip information between the timestamps of the min and max data points, an appropriate solution would involve presenting timestamp ranges instead of exact timestamps to users. This approach acknowledges the inherent limitations of data reduction methods in general, including those employed by M4 and similar techniques, where a large number of data points may be approximated by a single data point.

Secondly, enhancing the intuitive communication of accuracy guarantees within the visualizations is a key area for improvement. Introducing visual cues in the next iteration of the user interface, such as highlighting pixels on hover that may contain errors, will provide users with a direct visual understanding of the data's accuracy. Such enhancements could ultimately enhance user confidence in the error-bounded visualizations.

Lastly, addressing the challenge of determining acceptable accuracy levels for specific tasks and domains is crucial. Complementary approaches could be helpful in this regard, where, instead of users setting accuracy constraints, they could specify an acceptable maximum latency. The system would then adjust the error guarantees to satisfy this latency constraint, while offering users accuracy guarantees tailored to their visualization needs. Machine learning (ML) methods could be particularly beneficial for this purpose, utilizing historical interaction data and specific system characteristics (e.g., type of database). This strategy aims to improve the user experience by finding a balance between responsiveness and data fidelity that meets individual user requirements and expectations.