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 Timeseries Min-Max Caching with Error Bound Guarantees". Due to space constraints in the main paper, some detailse were omitted and are included here for comprehensive understanding.

Specifically, we provide detailed proofs for the theorems related to error bound calculations, along with the main algorithms for our method.

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. Determining Upper Bound for Pixel Errors: Theorems & Proofs

The following two theorems provide the upper bound for inner and inter-column errors:

Theorem 1 (Inner-column errors): Assume a two-color line chart of a time series variable y generated using the min and max values for y on a grouping $\mathcal{G}^k(T)$. The inner-column pixel errors E^i_{inner} in a pixel column p_i is the set difference of the pixel ranges P of the *left* and *right partially overlapping* groups l, r ($B^k_l \cap B^w_i \neq \emptyset$, $B^k_r \cap B^w_i \neq \emptyset$, and $B^k_l \not\subseteq B^w_i$, $B^k_r \not\subseteq B^w_i$) and the pixel range of the fully-contained groups in p_i, respectively. Given the following pixel ranges:

$$\begin{aligned}
P_l &= [p_y(B_l^{min}), p_y(B_l^{max})], \\
P_r &= [p_y(B_r^{min}), p_y(B_r^{max})], \\
P_i &= \bigcup_{B^k_j \in \mathcal{G}^k(T) | B^k_j \subseteq B^w_i} [p_y(B_j^{min}), p_y(B_j^{max})]
\end{aligned}$$

The inner-column pixel errors are given by

$$E^i_{inner} = (P_l \cup P_r) \setminus P_i$$

Proof: Let p_i be a pixel column in a two-color line chart of the time series variable y. Define P_i as the set of foreground pixels derived from the min and max y values of the fully-contained groups in $\mathcal{G}^k(T)$ within p_i. Let P_l and P_r represent the maximum foreground pixels that could be contributed in p_i by the left and right partially-overlapping groups l and r, respectively.

Since we know the min and max y values for l and r, we can determine these sets considering the geometric transformation function:

$$\begin{aligned}
P_l &= [p_y(B_l^{min}), p_y(B_l^{max})], \\
P_r &= [p_y(B_r^{min}), p_y(B_r^{max})].
\end{aligned}$$

Given the known min and max y values for fully-contained groups, and the min and max y values of the left and right partially-overlapping groups that intersect p_i, the foreground pixels in p_i should be at least P_i. However, data points within the min and max values of the partially-overlapping groups could potentially contribute to p_i's foreground pixels, with an upper bound of $P_l \cup P_r$ resulting in missing inner-column pixels. Also, considering that we do not have the actual timestamps for the min and max values and use timestamps near the middle of the corresponding intervals, some of these min or max values may incorrectly contribute to false foreground pixels. Again, the upper bound for such pixels is $P_l \cup P_r$.

Taking the worst-case scenario where such data points span the complete min and max y range of the partially-overlapping groups, the maximum extra potential foreground pixels is the set $(P_l \cup P_r) \setminus P_i$. This results in a discrepancy in the rendering of p_i, which we define as the inner-column pixel error E^i_{inner}. ■

Theorem 2 (Inter-column errors): For two adjacent pixel columns i and j in a two-color line chart generated using the min-max values of a variable y on a grouping $\mathcal{G}^k(T)$, the inter-column pixel errors for column i because of a missing or false inter-column line between i and j are given by:

$$E^{i,j}_{inter} = (F_{i,j} \cup M_{i,j}) \setminus P_i$$

where $F_{i,j}$ denotes the pixels in column i, resulting from rasterizing the line segment between the min or max values of $\mathcal{G}^k(T)$ before and after the intersection between the $i - j$ columns, $M_{i,j}$ the maximum set of pixels that can be rendered in the worst case by the missing correct inter-column line, and P_i the set of inner-column foreground pixels that can be accurately determined by the fully-contained groups of $\mathcal{G}^k(T)$ in column i.

Proof:

Let i and j be neighboring pixel columns in a two-color line chart visualization generated using the min-max y values on a grouping $\mathcal{G}^k(T)$. Define P_i as the set of inner-column pixels in column i that can be accurately determined by considering the min-max y values over all fully-contained groups of $\mathcal{G}^k(T)$ inside the pixel column interval.

Generating the visualization using the min and max y values means that we cannot determine the two data points immediately left and right of the intersection between the two columns and rasterize the correct inter-column line between columns i and j.

Instead, the line that connects the last one of the data points with min or max value of $\mathcal{G}^k(T)$ in column i and the first min or max value in column j represents the line segment across the $i - j$ intersection that would be rasterized instead. As a result, false foreground pixels may occur due to rendering the wrong inter-column line. Denote as $F_{i,j}$ the set of pixels in column i that this false line intersects and may as a result *false render* them as foreground pixels.

Additionally, there may be missing foreground pixels in column i because of the missing inter-column line. These pixels are the ones that would have been part of the line if it were present. Let $M_{i,j}$ represent the set of maximum pixels that could be affected by the *missing line* segment in column i.

To determine the maximum inter-column pixel errors for column i due to the missing or false inter-column line, we consider both sets $F_{i,j}$ and $M_{i,j}$. The maximum set of false pixels that may not be part of the true foreground is given by $F_{i,j} \setminus P_i$, where P_i represents the set of inner-column foreground pixels that can be accurately determined within column i. Similarly, the set of missing foreground pixels is given by $M_{i,j} \setminus P_i$, where $M_{i,j}$ represents the maximum set of pixels that could be affected by the missing line.

Therefore, the maximum inter-column pixel errors for column i due to the missing inter-column line at its boundary with column j can be expressed as:

$E^{i,j}_{inter} = |(F_{i,j} \cup M_{i,j}) \setminus P_i|$ ■

To calculate potential false pixels, $F_{i,j} \setminus P_i$, for a column i, we need to figure out which pixels could be affected when we rasterize the line between the data points determined by the Min Max aggregation across the $i - j$ intersection. However, since some of these pixels might already be accurately identified within the column, we subtract these correct pixels (P_i) from the total.

Calculating potential missing pixels, $M_{i,j} \setminus P_i$, due to the absence of a line at the border between columns i and j, is a bit more complex and depends on two situations: i) if there's a single partially-contained group overlapping the intersection between columns i and j (denoted as B_{ij}), the pixels that might be missed by our line are those within the vertical range of this group. So we define $M_{i,j}$ as the pixels between the lowest and highest points of the group, or $M_{i,j} = \{p \mid p_y(B_{ij}^{min}) \leq p \leq p_y(B_{ij}^{max})\}$.) if we have fully-contained groups on either side of the border (B_i in column i and B_j in column j), then missing line could start from any point in group B_i (or opposite from B_j) and increase towards group B_j (or B_i). In the worst case, the missing pixels are identified by the min max values of the adjacent groups, i.e., either $M_{i,j} = \{p \mid p_y(B_i^{min}) \leq p \leq p_y(B_j^{max})\}$ or $M_{i,j} = \{p \mid p_y(B_i^{max}) \leq p \leq p_y(B_j^{min})\}$.

It's important to note that we consider all possible scenarios and account for the existing correctly identified inner-column pixels. The maximum possible error, $E^{i,j}inter$, is computed as the set of potential false or missing pixels that are not part of the correctly identified pixels, i.e., $E^{i,j}inter = (F_{i,j} \cup M_{i,j}) \setminus P_i$. So, even if one scenario creates a longer line segment, it doesn't necessarily mean it will result in a larger error if many of its pixels are correctly identified and included in P_i.

III. ALGORITHMS

A. Upper Error Bound Evaluation

These steps for determining the error bound for a line chart visualization, which involve analyzing the potential inner-column and inter-column pixel errors, are outlined in Algorithm 1:

Algorithm 1: Determining Upper Error Bound

Input: $\mathcal{G}^k(T)$, w, h
Output: Error bound ϵ
$E_{max} \leftarrow 0$
for *each pixel column* c_i **do**
    Determine the set of correct inner-column pixels P_i for c_i
    Determine the set of potential missing inner-column pixel errors Q_i for c_i
    $M_i \leftarrow \emptyset$
    $F_i \leftarrow \emptyset$
    for *each neighbor* $c_j \in \{c_{i-1}, c_{i+1}\}$ **do**
        if c_j *exists* **then**
            Rasterize the line segment between the two min-max of c_i and c_j immediately across the $c_i - c_j$ intersection to obtain the set $F_{i,j}$
            Determine the maximum set of pixels that the actual missing line across $i - j$ would render to obtain the set $M_{i,j}$
            $F_i \leftarrow F_i \cup F_{i,j}$
            $M_i \leftarrow M_i \cup M_{i,j}$
    $E_{max} \leftarrow E_{max} + |(Q \cup M_i \cup F_i) \setminus P_i|$
$\epsilon \leftarrow \frac{E_{max}}{w \times h}$
return ϵ

B. Query Processing

Next, we provide in detail the query processing algorithm of MinMaxCache:

Algorithm 2: Query Processing

Input: Query Interval Q, Interval Tree IT, Width w, Height h, Sampling Interval s, Error Bound ϵ
Output: Query Results R
Initialize w pixel columns each maintaining min-max value and fully contained range
Find overlapping groupings $G_{overlapping}$ in IT that overlap with Q
for *each grouping* $s \in G_{overlapping}$ **do**
    Update corresponding pixel columns using s
Evaluate ϵ'
Identify the grouping G_{max} with maximum query interval coverage in $G_{overlapping}$
Determine the aggregation factor AF for G_{max}
Set the aggregation interval of the grouping G_{agg} to $w * AF$
if $\epsilon' \leq \epsilon$ **then**
    Identify the intervals of pixel columns with missing data or uncomputable error bound
    if $G_{agg} < 4 * s$ **then**
        Issue a raw query for these intervals to the database
    else
        Issue an aggregate query for these intervals with aggregation interval G_{agg} to the database
    Update pixel columns based on the query results
else
    Double the aggregation interval of the grouping G_{agg}
    if $G_{agg} < 4 * s$ **then**
        Issue a raw query to the database
    else
        Issue an aggregate query with aggregation interval G_{agg} to the database
    Update pixel columns and $G_{overlapping}$ based on the query results
    Reevaluate ϵ'
    if $\epsilon' > \epsilon$ **then**
        Issue an M4 query to the database
        return *M4 query results for visualization*
Update IT with new groupings
Prepare R as a set of tuples containing min-max values and first-last tuples from each pixel column
return R