Polyspector™: An Interactive Visualization Platform Optimized for Visual Analysis of Big Data

Xinxiao Li **Akira Kuroda** **Hidenori Matsuzaki**

Toshiba Corporate Research & Development Center
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki-shi, Japan, 212-8582
{xinxiao.li, akira3.kuroda, hidenori.matsuzaki}@toshiba.co.jp
Tel: +81-44-549-2232

ABSTRACT
With the advent of the "big data" era, there are unprecedented opportunities and challenges to explore complex and large datasets. In the paper, we introduce Polyspector™, a web-based interactive visualization platform optimized for interactive visual analysis with two distinguishing features. Firstly, a visualization-specific database engine based on pixel-aware aggregation is implemented to generate views of hundreds of millions of data items within a second even with an off-the-shelf PC. Secondly, a novel deep-linking mechanism, combined with the pixel-aware aggregation, is exploited to realize interactive visual analysis interfaces such as zooming, overview + detail, context + focus etc.

Author Keywords
Visualization; Interactive Visual Analysis

ACM Classification Keywords
H.5.2 Graphical user interfaces (GUI); I.3.6 Interaction techniques;

INTRODUCTION
Interactive visual analysis system used to suffer from ineffective information presentations and poor exploratory capabilities.

With the increasing volume of datasets, it is time-consuming to transfer and render the whole dataset. Furthermore, even when the visualization result is finally represented on the screen after a long wait, it is unintelligible due to overlapping and redundant visual primitives. A visualization result that is presented on screen is inherently restricted to (width × height) pixels – the area of its view regardless of the size of the original dataset. This implies that a large dataset containing far more data items than the number of pixels to be viewed should be transformed to a more effective information presentation by using a data reduction strategy.

UIST'16 Adjunct, October 16-19, 2016, Tokyo, Japan
ACM 978-1-4503-4531-6/16/10.
http://dx.doi.org/10.1145/2984751.2985720

We have investigated a pixel-aware aggregation algorithm [1, 2] to map data from a large dataset of several hundred thousand data items to a small dataset while maintaining perfect operability of visualization. The previous algorithm heavily depended on the aggregation functionality of existing databases. In the Polyspector™ visualization platform, we have improved and generalized the pixel-aware aggregation approach and developed it into a visualization-specific database engine to aggregate hundreds of millions of data items within one second even with a laptop computer locally.

Besides the response time to generate views, seeing related data in multiple views can greatly help to make sense of the data and explore data efficiently through interactive visual analysis. With pixel-aware aggregation, sub-datasets corresponding to different views appear to be independent of each other. This is especially problematic with a client-server architecture where aggregation computation is on the server side and rendering is on the client side.

Our web-based interactive visualization platform exploits a deep-linking mechanism. Multiple views on the client side are linked by sending brushing conditions to a pixel-aware aggregation module on the server side, and propagating the updating sub-datasets back to the relevant views. With the deep-linking mechanism, the common visual analysis interfaces such as zooming, overview + detail, brushing and linking can be realized as is done with small datasets.

PIXEL-AWARE AGGREGATION
Data items in a large dataset to be visualized are divided into two categories corresponding to the current view width and/or view height. The first category consists of the data items that will be mapped to geometric primitives thinner than or equal to one pixel on the current screen. The second category includes the data items mapped to geometric primitives larger than one pixel. The data in the first category are often overlapped on the screen and so they are suitable for aggregating according to their visualization coordinates on the current screen. The data in the second category are not overlapped on each other on the screen and so they are transferred and rendered as is. In a large dataset, since the number of data items in the second category is limited by the screen size, it is generally much less than the number of data items in the first category. For the same

reason, the number of aggregated data is much less than the number of data items in the first category.

The pixel-aware aggregation for the second category is implemented in three steps: (1) mapping data items to their screen coordinates; (2) clustering data items according to their screen coordinates; (3) aggregating each cluster to triple (minimum, maximum, average) values. The performance of pixel-aware aggregation is further improved by indexing and parallelizing in multicore environments in our visualization-specific database engine.

DEEP-LINKING

With the pixel-aware aggregation in a client-server architecture, sub-datasets corresponding to different views may appear to be unrelated, but the internal connection can be found by retrieving the original large dataset.

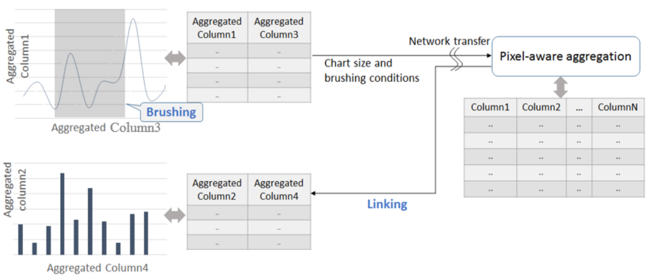

Figure 1. Illustration of brushing and deep-linking. A brushing action in a line chart (top) will update bar chart (bottom) by locating and aggregating relevant sub-datasets with pixel-aware aggregation.

As in Figure 1, the brushing conditions in a line chart are sent to the pixel-aware aggregation module on the server side to find the relevant sub-dataset of the bar chart. The sub-dataset is further aggregated and sent to client to update the bar chart.

INTERACTION OF DEMO

Our visualization platform can interactively explore large datasets containing up to hundreds of millions of data items. The following typical interaction interfaces will be shown in the demo.

Zooming

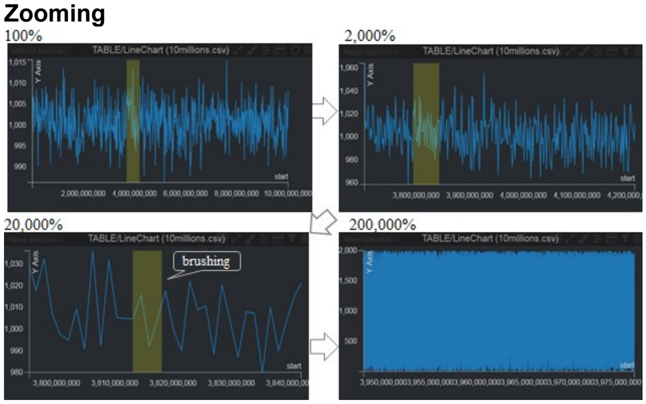

Figure 2. Zooming into a large dataset by continual brushing. Each subsequent view is automatically fitted to the current view size.

The zoomable user interface (ZUI) is used to interactively change the scale to drill down into more details of a large dataset. As shown in Figure 2, the line chart for an example dataset containing 10 million data items is zoomed in from a full range view (zoom scale 100%) to several detail views (zoom ratio of 2,000%, 20,000%, 200,000%, separately). The details in the preceding three views are smoothed out by averaging multiple neighboring values. The last view is rendered with the original data items as is without aggregation. Our current implementation has capacity to render or update views of hundreds of millions of data items within one second.

Overview + detail, brushing and linking

With the deep-linking mechanism, multiple views of one or multiple large datasets can be organized together to make sense of data. Figure 3 shows a dashboard including two scatter plots and one line chart created with our visualization platform. The top-left scatter plot is an overview and the top-right scatter plot is a detail view changed by the brushing conditions in the overview. The line chart shown at the bottom is also linked to the overview so that it can be updated with the shared brushing conditions. The overview is in highlight display mode to emphasize the brushing conditions with coloring.

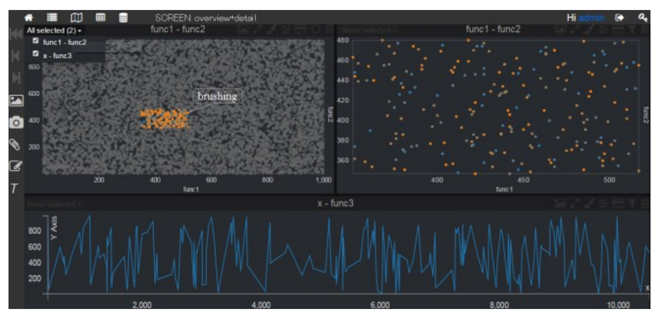

Figure 3. A dashboard showing the overview + detail, brushing and linking interfaces. The overview (top-left scatter plot) is linking to the other two detail views.

CONCLUSION

A visualization platform optimized for interactive visual analysis is proposed. In the current implementation, we have verified its capacity to render or update views of hundreds of millions of data items within one second. The deep-linking mechanism combined with pixel-aware aggregation makes it possible to explore large datasets with common visual analysis interfaces such as zooming, overview + detail, brushing and linking. It has been applied to efficiently analyze massive cases of simulations with various SSD design choices [3]. The pixel-aware aggregation engine is scalable and so its performance can be further improved to deal with various big data visual analysis tasks.

REFERENCES

1. Li, X., Kuroda, A., Matsuzaki, H., Nakajima, N., Distributed Aggregate Computation between Server and Client for Interactive Visualization. 2015 IEEE 5th Symposium on Large Data Analysis and Visualization, pp. 135-136.
2. Li, X., Kuroda, A., Matsuzaki, H., Nakajima, N., Advanced aggregate computation for large data visualization. 2015 IEEE 5th Symposium on Large Data Analysis and Visualization, pp. 137-138.
3. Hasegawa,Y., Kuroda,A., Matsuzaki,H., Asano,S., Architectural Exploration Platform for Solid State Drive Designs. BEST PRESENTATION AWARD in 52nd Design Automation Conference, Design Track, June 2015.