A Framework for Interactive Exploration of Clusters in Massive Data Using 3D Scatter Plots and WebGL

Lukas Wagner
Hasso Plattner Institute,
University of Potsdam, Germany
lukas.wagner@hpi.de

Daniel Limberger
Hasso Plattner Institute,
University of Potsdam, Germany
daniel.limberger@hpi.de

Willy Scheibel
Hasso Plattner Institute,
University of Potsdam, Germany
willy.scheibel@hpi.de

Matthias Trapp
Hasso Plattner Institute,
University of Potsdam, Germany
matthias.trapp@hpi.de

Jürgen Döllner
Hasso Plattner Institute,
University of Potsdam, Germany
juergen.doellner@hpi.de

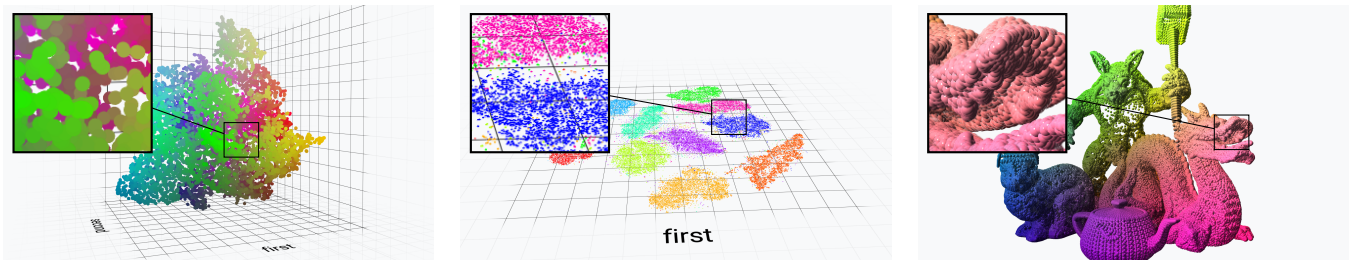

Figure 1: Various datasets visualized with our rendering framework. From left to right: 11 000 documents, arranged using dimension reduction based on topic similarities; MNIST handwritten digits dataset, arranged using t-SNE dimension reduction; Different theme (using Phong shading and larger point diameter) showing 140 000 vertices of the Khronos figures.

ABSTRACT

This paper presents a rendering framework for the visualization of massive point datasets in the web. It includes highly interactive point rendering, cluster visualization, basic interaction methods, and importance-based labeling, while being available for both mobile and desktop browsers. The rendering style is customizable, as shown in figure 1. Our evaluation indicates that the framework facilitates interactive visualization of tens of millions of raw data points even without dynamic filtering or aggregation.

KEYWORDS

WebGL, scatter plot, massive data, cluster visualization, framework

ACM Reference Format:
Lukas Wagner, Daniel Limberger, Willy Scheibel, Matthias Trapp, and Jürgen Döllner. 2020. A Framework for Interactive Exploration of Clusters in Massive Data Using 3D Scatter Plots and WebGL. In *The 25th International Conference on 3D Web Technology (Web3D '20), November 9–13, 2020, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3424616.3424730

Web3D '20, November 9–13, 2020, Virtual Event, Republic of Korea

1 INTRODUCTION

With the advance of computational capabilities, the amount of available data grows steadily, to previously unfathomable ranges [Wegman 1995]. The need for improved processing methods brought about new research topics such as machine learning and big data. While this allows utilizing the data, the human comprehension remains a challenge. It is often not feasible for a human observer to visualize and understand these huge and often high-dimensional datasets. This has prompted extensive research in Big Data visualization [Hofer et al. 2020; Keim 2001; Po et al. 2020]. End-user facing applications typically used for plotting data, such as Microsoft Excel, simply cannot cope with this amount of data. Even custom-built visualizations using plotting frameworks, e.g. the commonly used D3 library, fall short when trying to draw huge datasets. The capabilities of D3, for example, are limited by the web browsers' SVG rendering engine. In the past, it was necessary to harness the power of low-level APIs such as OpenGL to achieve the required performance. With the advance of technologies such as WebGL and the upcoming WebGPU standard, it has become feasible to create complex web-based visualizations. This allows for a operating system and (mostly) browser-agnostic implementation, resulting in very lax requirements on the end-users hardware. Following this shift, research topics, which have historically only been feasible with a low-level approach, such as point clouds, have seen successful web-based implementations. Two noticeable examples are Potree[1], a specialized point cloud viewer, and CesiumJS[2], a more generalized

[1] potree.org
[2] cesium.com/cesiumjs

geospatial toolkit [Discher et al. 2018; Schütz 2016]. Point clouds require millions of data points to be rendered. While these points do not provide any meaning individually, they can illustrate very complex scenes. This distinguishes point clouds from a regular plotting approach, where every point carries data.

In order to allow for an easier overview of the dataset, as well as to approach the occlusion problem [Elmqvist and Tsigas 2008], we analyze the point cloud using clustering algorithms such as α-shapes [Lucieer and Kraak 2004]. Reducing millions of points to significantly less clusters (typically in the magnitude of 10) gives a high-level view of the data [Linsen et al. 2008]. Additionally, the clusters can be used to reduce concealment issues. By representing multiple points with a single object of less extend, more of the dataset is visible at once, while still providing a meaningful representation. In combination with a multi-level clustering approach, the dataset can be explored both on a high level and in detail. Picking and labeling techniques can be used to quickly spot relevant cluster. After focusing on a cluster, the user can increase its level-of-detail to view subclusters and individual data points.

Contributions. We present a new web-based rendering framework for visualizing and exploring massive datasets, capable of interactively visualizing tens of millions of data points. In order to facilitate understanding, we utilize a dual rendering approach to both give a high-level overview over the dataset, as well as allow low-level interaction with single datapoints simultaneously.

2 RENDERING APPROACH

In order to render massive datasets, they first have to be loaded and possibly preprocessed by the CPU. As this task can take a while, dependent on size and format of the input file, we use a streaming approach. While the data source is being read, any chunks of datapoints that have been read already are immediately made available for rendering by handing the data over to the GPU. This allows the user to get a quick preview of the dataset without waiting for the the whole source to be loaded. Additionally, this allows us to support nonfinite data sources, such as live sensors producing an endless stream of data.

For rendering the individual data points, we utilize instanced rendering. Instead of creating an individual geometry for each point, a shared geometry is reused. The geometry is then adjusted for each data point by passing along per-point data. This technique significantly improves rendering performance and allows for custom shapes of varying complexity for multi-variate data mapping.

Since one goal of our visualization is to gain a high-level understanding of the dataset, we use clustering and dimension reduction techniques to group the data points. These subsets allow for a multi-level rendering approach, where the surface of a set is used as mesh, replacing any points contained in the set. Thus the underlying dataset could contain an arbitrary amount of points, as only the subsets currently in focus is be rendered in high detail, while everything the remaining subsets are replaced with the high-level representation. The user can then control the clusters' level of detail either indirectly through the camera position or directly by interacting with the clusters. With this multi-level approach, the user can quickly gain an overview of the overall structure of the dataset as well as retrieve selective, per-point information.

3 IMPLEMENTATION

As noted in the introduction, we chose a web-based implementation to increase portability and simplify the user experience. Thus, we were limited to WebGL 2. We chose webgl-operate[3] as rendering framework. In order to speed up loading and processing a dataset, we utilize threads to harness available CPU power. When loading the data, we use a streaming interface to support both normal files, as well as data sent by a server using a stream. The main thread receives the input in chunks. These are handed over to worker threads for parsing, which arrange the data into columns, each representing one attribute. By handling the input in chunks, any available data can be sent to the GPU immediately, enabling rendering of intermediate results. The data is managed without any optimization structure such as an octree. While this would allow for handling of even bigger datasets, the current approach already allows rendering millions of points interactively, as seen in table 1.

4 CONCLUSION

We presented a new rendering framework for visualizing and exploring massive datasets. Using a highly optimized WebGL-based implementation, we can visualize tens of millions of data points interactively. In order to facilitate understanding, we utilize a dual rendering approach to both give a high-level overview of dataset, as well as allow low-level interaction with single datapoints.

REFERENCES

Sören Discher, Rico Richter, and Jürgen Döllner. 2018. A Scalable WebGL-Based Approach for Visualizing Massive 3D Point Clouds Using Semantics-Dependent Rendering Techniques. In *Proceedings of the 23rd International ACM Conference on 3D Web Technology (Web3D '18)*. Article 19, 9 pages.

Niklas Elmqvist and Philippas Tsigas. 2008. A Taxonomy of 3D Occlusion Management for Visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 5 (2008), 1095–1109.

Peter Hofer, Lisa Perkhofer, and Albert Mayr. 2020. *Interaktive Big Data Visualisierungen – Potenzial für das Management Reporting.* Springer Fachmedien Wiesbaden, Wiesbaden, 159–187.

Daniel A. Keim. 2001. Visual Exploration of Large Data Sets. *Commun. ACM* 44, 8 (Aug. 2001), 38–44.

Lars Linsen, Tran Long, Paul Rosenthal, and Stephan Rosswog. 2008. Surface Extraction from Multi-field Particle Volume Data Using Multi-dimensional Cluster Visualization. *IEEE transactions on visualization and computer graphics* 14 (11 2008), 1483–90.

Arko Lucieer and Menno-Jan Kraak. 2004. Alpha-shapes for visualizing irregular-shaped class clusters in 3D feature space for classification of remotely sensed imagery. In *Visualization and Data Analysis 2004*, Vol. 5295. International Society for Optics and Photonics, SPIE, 201 – 211.

Laura Po, Nikos Bikakis, Federico Desimoni, and George Papastefanatos. 2020. Linked Data Visualization: Techniques, Tools, and Big Data. *Synthesis Lectures on Semantic Web: Theory and Technology* 10, 1 (2020), 1–157.

Markus Schütz. 2016. *Potree: Rendering Large Point Clouds in Web Browsers.* Ph.D. Dissertation.

Edward Wegman. 1995. Huge Data Sets and the Frontiers of Computational Feasibility. *Journal of Computational and Graphical Statistics* 4 (07 1995).

[3] webgl-operate.org

Table 1: Frames per second for different amounts of data measured on a low, mid, and high end GPU in Chrome 84.

Test System	Number of Points				
in 1920×1080px	1×10^6	5×10^6	1×10^7	5×10^7	1×10^8
Intel UHD 630	26.8	18.6	10.8	2.6	1.4
Nvidia GTX 1650	196.6	65.2	38.0	8.9	4.6
Nvidia GTX 1080	474.6	154.3	81.2	17.6	9.0