Research Statement

Jaemin Jo

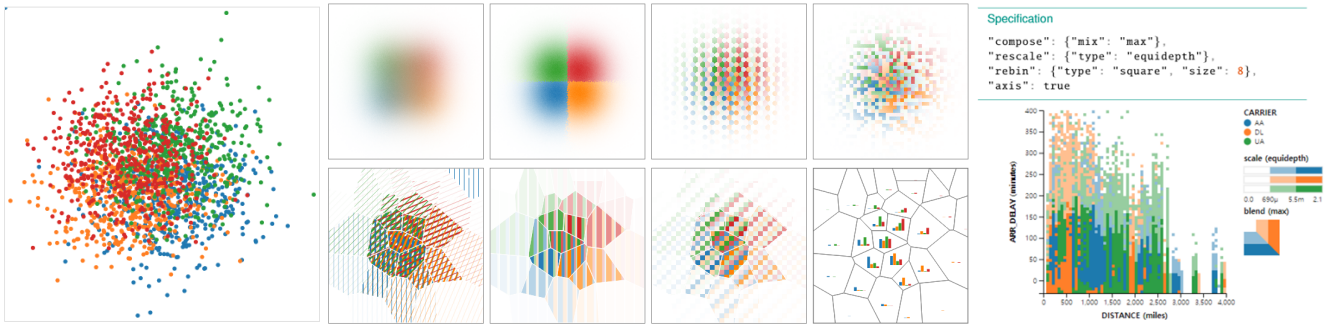
Visualization has become a necessity for data science. The ever-increasing data size and complexity that we confront in daily analytics are now challenging our perceptual and cognitive abilities, leaving the data intractable, without support. Carefully designed visualization with natural interaction idioms can amplify our abilities and promote the exploration and understanding of large-scale data. My previous research aims at designing **scalable** and **responsive** visualization systems. My ultimate research goal is empowering humans to understand large-scale data by developing scalable visual analytics systems with human-centered interaction.

Scalable and Responsive Visualization Systems

Visualization is a powerful tool for understanding, decision-making, and communication [9]. However, I have identified that previous visualization techniques and systems confront two important hurdles with the sheer number of data items: a **scalability hurdle** and a **responsiveness hurdle**.

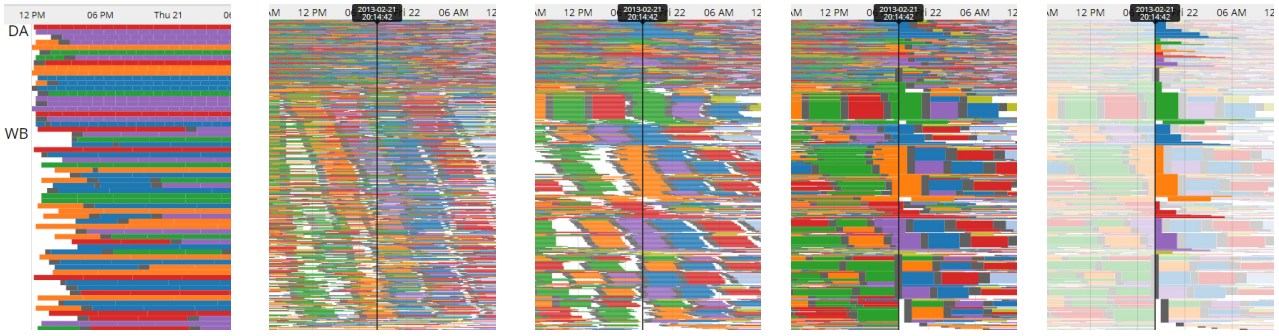
The **scalability hurdle** of a visualization technique refers to the degradation of visual quality due to overplotting or clutter, which is exacerbated when the technique draws an individual visual element for each data item. For example, a scatterplot (Figure 1a) represents a data item as one mark (i.e., circle) with its position (x, y) determined by two quantitative variables. As the number of data items increases, the scatterplot becomes too crowded, producing an uninterpretable result. Scalable visualization techniques are an essential prerequisite for large-scale visual analytics.

I am interested in designing and improving visualization techniques to overcome the scalability hurdle. In response to the above example, I generalized the design space of **multiclass density maps** that scaled up common scatterplots (Figure 1b) through binning and aggregation [7]. A large range of density map designs can be specified through a concise declarative grammar (Figure 1c) to support various tasks on scatterplots with many data items. The specification can be then interpreted by our interpreter and rendered on a web page. I am also intrigued by simplification algorithms for alleviating the scalability hurdle. For example, I developed **a reordering and aggregation algorithm for Gantt charts** [2] to visualize large-scale event sequence data.



(a) An overplotted scatterplot (b) Various designs of density maps (the data is the same as (a)) (c) A specification and its rendition

Figure 1: Multiclass density plots can overcome the scalability hurdle of previous scatterplots through binning and aggregation. Various map designs can be specified through a single unifying grammar with flexibility in creating a task-specific design.



(a) Small Gantt chart (b) Cluttered Gantt chart (c) After reordering (d) After aggregation (e) Simplified chart

Figure 2: (a) A Gantt chart gives an overview of tasks scheduled on multiple resources over time. (b) With a larger number of task bars, a Gantt chart suffers from clutter, becoming uninterpretable. (c) Resources that have similar task sequences are placed nearby to alleviate the clutter. (d) Adjacent similar task bars are aggregated into a single task bar, simplifying the chart further. (e) Highlighting allows people to focus on tasks that are active at a time point of interest.

On the other hand, the **responsiveness hurdle** comes from the latency of interactive systems when the systems deal with large-scale data. Reducing the latency has been a core concern in designing visualization systems, because long latency can outpace humans' ability to focus their attention, eventually degrading the quality of the analysis [11]. Nielsen [10] suggested 10 seconds as the time limit for keeping the user's attention, but only judiciously designed systems can meet such a requirement.

My research interests also lie in developing responsive visualization systems to support fluent visual analytics, even on large-scale data. With the great advances in modern computing technologies, our first approach was to **tightly couple visualization systems with a distributed computing engine**, Apache Spark [1]. Our SwiftTuna system [3] (Figure 3) took a *progressive* computation paradigm, delivering intermediate results of computation on the fly without blocking analysis until the complete result becomes available. This allows people to obtain feedback from the system within the attention-preserving latency of 10 seconds. Developing the system, we identified and addressed new design challenges that were not apparent in previous visualization systems. For example, our system manages visualizations that are updated in real time, but those visualizations are approximate (i.e., data values are estimated through statistical procedures), so it is important to enable users to assess the uncertainty of the data values as well as interpreting the data values themselves.

(a) The main interface of the SwiftTuna system (b) Parallel execution of tasks on a Spark cluster

Figure 3: (a) The SwiftTuna system allows people to explore a large-scale multidimensional dataset interactively with scalable and approximate visualizations. (b) Samples are created from the dataset and processed on a Spark cluster in parallel, giving users partial responses every few seconds.

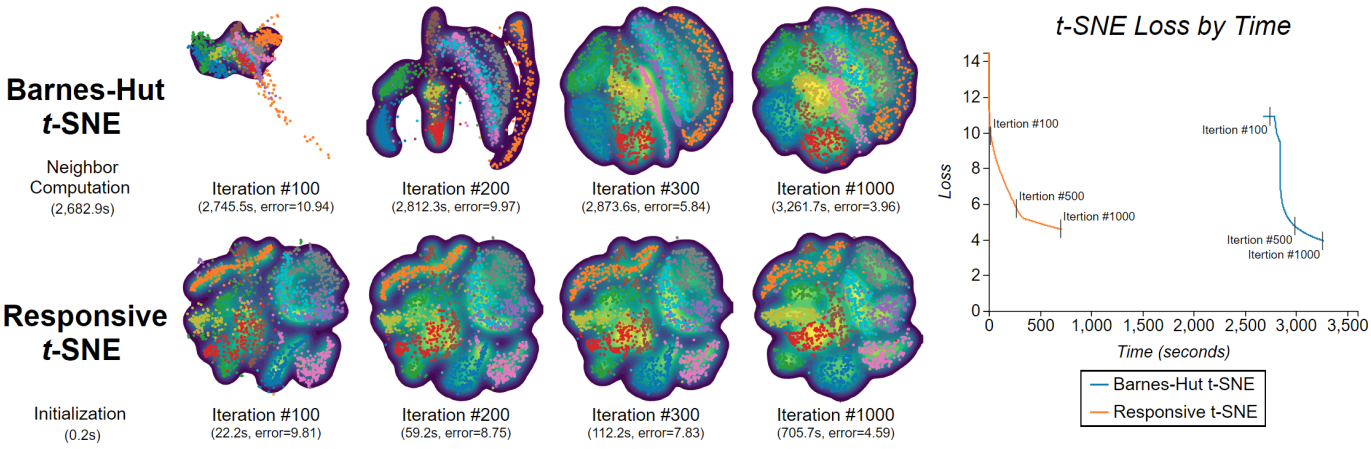


Figure 4: Despite the popularity of Barnes-Hut t-SNE, an efficient variant of the original t-SNE [8], its long initialization time resulting from neighborhood computation has restricted its use in interactive systems. Our responsive t-SNE greatly reduces the upfront overhead while giving a comparable embedding quality.

As in the SwiftTuna system, providing intermediate results through progressive algorithms is suitable for large-scale visual analytics in that users can conduct analysis without interruption. However, despite the advantage, it is not always simple or even possible to convert a previous sequential algorithm directly to a progressive one. In our article [6], we address one important problem: progressively finding the k-nearest neighbors (KNN) of a given point in a multidimensional space, i.e., the KNN problem. One most promising application is **Responsive t-Distributed Stochastic Neighbor Embedding (t-SNE)** [8]. t-SNE is a popular nonlinear dimensionality reduction algorithm, but the long initialization time due to KNN computation has limited its potential use in interactive visualization systems. Our Responsive t-SNE spreads the load of neighborhood computation to later iterations, alleviating the initial overhead coming from the blocking KNN methods (Figure 4) while giving a comparable embedding quality.

Future Research Agenda

My mission is to design scalable, responsive, and usable visualization systems that can facilitate the understanding and exploration of large-scale complex data. I plan to research the following exciting topics.

Human Factors in Progressive Visualization Systems

With the popularity of progressive visualization systems, a variety of algorithms and systems have been proposed for large-scale data analysis. However, little has been done regarding how people use and interact with those systems. Recent studies on progressive visualization systems have revealed essential concepts that were not apparent before, such as *approximation*, *uncertainty*, and *scheduling*. However, important questions still remain regarding such concepts. When do people make decisions when running uncertain visualizations? How can the system prevent people from going further with an early wrong decision? Studying those **human factors** can imply general but crucial design guidelines when designing effective progressive visualization systems.

Building a Design Space of Scalable and Uncertain Visualization

To facilitate data exploration at scale, visualization techniques (1) have to be **scalable** to the number of data items and (2) support the visualization of **uncertain** data values. Designing visualization techniques that satisfy all the two requirements is important but challenging. As part of a future research agenda, I would like to build a design space of such visualization techniques based on successful case studies from the literature. I believe the design space will not only give designers useful guidelines on creating a novel technique but also provide clues on underexplored design alternatives.

Multimodal Interaction for Large-Scale Data Exploration

Recent studies [4, 5] have shown the effectiveness of multimodal interactions, such as multitouch or pen interaction, in visualization systems. Encouraged by these results, I am interested in **designing a multimodal interface for large-scale data exploration**. For example, one can apply a filtering transform by drawing a lasso on a scatterplot with a pen. This interaction itself may be easy for people to remember and use, but the usability issues regarding the interaction, such as response latency due to the size of data, must be revisited in the context of progressive visualization systems.

References

[1] Apache Spark. https://spark.apache.org/, May 2018.

[2] Jaemin Jo, Jaeseok Huh, Jonghun Park, Bohyoung Kim, and Jinwook Seo. Livegantt: Interactively visualizing a large manufacturing schedule. *IEEE transactions on visualization and computer graphics*, 20(12):2329–2338, 2014.

[3] Jaemin Jo, Wonjae Kim, Seunghoon Yoo, Bohyoung Kim, and Jinwook Seo. Swifttuna: Responsive and incremental visual exploration of large-scale multidimensional data. In *Pacific Visualization Symposium (PacificVis), 2017 IEEE*, pages 131–140. IEEE, 2017.

[4] Jaemin Jo, Bongshin Lee, and Jinwook Seo. Wordleplus: expanding wordle's use through natural interaction and animation. *IEEE computer graphics and applications*, 35(6):20–28, 2015.

[5] Jaemin Jo, Sehi L'Yi, Bongshin Lee, and Jinwook Seo. Touchpivot: Blending wimp & post-wimp interfaces for data exploration on tablet devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2660–2671. ACM, 2017.

[6] Jaemin Jo, Jinwook Seo, and Jean-Daniel Fekete. PANENE: A Progressive Algorithm for Indexing and Querying Approximate k-Nearest Neighbors. *IEEE transactions on visualization and computer graphics, to appear*, 2018.

[7] Jaemin Jo, Frédéric Vernier, Pierre Dragicevic, and Jean-Daniel Fekete. A Declarative Rendering Model for Multiclass Density Maps. *IEEE transactions on visualization and computer graphics, to appear*, 2018.

[8] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[9] Tamara Munzner. *Visualization analysis and design*. CRC press, 2014.

[10] Jakob Nielsen. *Usability engineering*. Elsevier, 1994.

[11] Emanuel Zgraggen, Alex Galakatos, Andrew Crotty, Jean-Daniel Fekete, and Tim Kraska. How progressive visualizations affect exploratory analysis. *IEEE transactions on visualization and computer graphics*, 23(8):1977–1987, 2017.