**ABSTRACT**

We present a ParaViewWeb based visual analytics application running on large high-resolution display supporting standard mouse and keyboard interaction. The application relies on SAGE2 for user interaction and multi-display visualization. We also employ a scalable middleware system called "Cloudberry" that allows users to interactively query and analyze large amounts of temporal and spatial data stored on a back end Apache AsterixDB store to enable big data analytics and interactive visualization. Our Visual Analyzing Billion Tweets application shows interactive query and visualization of result from over a billion twitter feeds streamed in real-time to the back end Apache AsterixDB. In our setup, we ran the visual analytics application on a large high-resolution display with a 24-tiled display in a 6 x 4 configuration. We also run a comparative study of the application running on a single 24 inch display and the 24-tiled display with some very interesting findings supporting the benefit of using large high-resolution display for visual analytics.

**Index Terms:** H.1.m [Information Systems]: MODELS AND PRINCIPLES—Miscellaneous

1 INTRODUCTION

The process of large data exploration, modeling, and analysis require a vastly different class of tools. Although the size of the data increased dramatically, it is still crucial to be able to quickly analyze large datasets to identify trends, anomalies, and correlations. Identification of these patterns in data improves our ability to predict future events to stay at the forefront of ongoing developments. Therefore, development of useful information visualization tools enabling interactive data exploration can result in the discovery of relationships in data for users.

The U. S. Army Test and Evaluation community tests and evaluates everything the Soldier touches. This includes network, application, vehicle, weapon, communication device, data link, etc. As a result, the community is the single largest producer of data in the Department of Defense Research, Development, Test and Evaluation community. In addition to challenges with the amount of the data collected, the many different types of data are collected, as the testing and evaluation process measures everything conceivable to assess its effectiveness, suitability, survivability, and safety. These requirements produce massive, heterogeneous, distributed data sets requiring new data analysis approaches to obtain usable information from the data. In addition, a growing requirement is time-critical analysis for the heterogeneous data collected throughout the larger Department of Defense Testing and Evaluation community [2]. We design a prototype application that visualizes a record of one billion tweets to demonstrate the viability of using the underlying technologies to enable big data visual analytics on a large high-resolution display (LHRD) system for U. S. Army Test and Evaluation heterogeneous data.

In a big data exploration, a visual analytics tool running on a LHRD system can open up the possibility for the users to visualize and interact with more of the data. The unstructured nature of the data in information visualization domain can unfold in many different ways, and visualization can be facilitated with a LHRD visualization environment. Andrews et. al. described the potential benefits of using LHRD for information visualization [1].

We developed Visual Analyzing Billion Tweets, a LHRD interactive information visualization application capable of supporting geospatial and temporal visual analytics on a large screen display. Our LHRD information visualization application takes advantage of Scalable Amplified Group Environment2 (SAGE2) [7] support for large high-resolution visualization. SAGE2 allows multiple displays to be used as a single large high-resolution multiscreen workspace resulting in a low cost LHRD system. SAGE2 users are able to access and manipulate this workspace through their own respective devices including tablets, laptops, or even smart phones. In our current development, we extended ParaViewWeb to run on SAGE2 framework to overcome the scalability limitation of our previous visualization framework [10] in the development of LHRD information visualization application targeting large data. Fig. 1 shows a picture of our Visual Analyzing Billion Tweets application running on our 24-tiled display system. A user can interact with 1,022,758,431 tweets stored on the backend AsterixDB store, and real-time interactive visualization of search results are made possible through a scalable middleware, Cloudberry [6].

There are many options in the development of an information visualization tools when one considers different combination of software platforms, APIs, and hardware. However, the most challenging design decision is the need to create an application for our specific use case while also considering the extensible and general-purpose aspect of the tool to support the next phase of the research. Specifically in our work, the goal is to support interactive visualization of big data through useful human-computer interaction on a LHRD system. Although our large screen real estate allows user to simultaneously visualize and analyze more of the data, it has also increased our data processing requirement. We are utilizing the Cloudberry middleware to progressively deliver the data to maintain a real-time usable visualization. In addition, to address our need for faster prototyping and extensibility, we have take advantage of a rich ecosystem of tools of the open source community in the development of Visual Analyzing Billion Tweets application.

Chen listed scalability as one of the top 10 unsolved information visualization problems [4] and our Visual Analyzing Billion Tweets application demonstrates scalability with the ability to process, dis-
We also used Cloudberry to manage the data flow between the data work. Tweets application. We then elaborate on our Visual Analyzing visualization to those LHRD software framework can be challenging a web server with the specified configuration of the client machines framework is launched on the head node of our cluster, by launching users to connect via a modern browser and network access. The multiple displays to act as one large workspace, allowing multiple SAGE2 is a software framework that uses web-browser technologies as one large desktop workspace environment. For our visualization to enable data-intensive collaboration across multiple displays acting libraries to support the development of our Visual Analyzing Billion Tweets application. We then elaborate on our Visual Analyzing Billion Tweets application, follow by our usability study to compare running our application on a single 24 inch display and on a LHRD environment.

2 Related Work

The past few years saw an explosion of web based information visualization tools. Bostock publication of D3 library [3] in 2001 has enabled countless innovative interactive information visualization projects. Although web based visualization is emerging as industry standard, using them in the development of visualization application for a LHRD environment can be challenging. Chung in his survey paper of software frameworks for cluster-based large high-resolution displays (LHRD) mentioned a number of software capable of supporting LHRD [5]. However, adding many existing information visualization to those LHRD software framework can be challenging as they are not compatible with a web based visualization framework.

The following sections describe the technologies we used in the development of our Visual Analyzing Billion Tweets application. We utilized the ParaViewWeb to run on top of SAGE2 environment to enable ParaViewWeb to be displayed on SAGE2 LHRD environment. We also used Cloudberry to manage the data flow between the data manager and visualization in our application, enabling big data visualization on big display.

2.1 SAGE2 Software Framework

SAGE2 is a software framework that uses web-browser technologies to enable data-intensive collaboration across multiple displays acting as one large desktop workspace environment. For our visualization cluster, we have three client machines driving a 24-tiled display wall, with 8 displays per machine. The SAGE2 framework combines the multiple displays to act as one large workspace, allowing multiple users to connect via a modern browser and network access. The framework is launched on the head node of our cluster, by launching a web server with the specified configuration of the client machines to host the display clients. Once the server is started, there are two types of clients that can be connected. These clients are the UI clients and the display clients. The UI clients are how the users connect to the environment, by typing the URL of the web server into a modern browser. The display clients are how the client machines of the cluster actually display what is to be shown on the wall. Multiple users can connect to a UI client on their own laptop or machine, with the ability to interact with what is displayed on the wall, as well as drag/drop files, or open SAGE2 applications on the wall. The server running on the head node is only aware of what applications and users are connected to the current SAGE2 session, but has no information about what is actually being displayed on the display clients.

The SAGE2 framework includes many applications that can be launched through a UI client and displayed across the wall. One such application is a WebView application. This application simulates a browser running within the multi-display workspace by embedding a web page via its URL into the application. The application may be resized, allowing viewing and interacting with web pages in a large display environment.

2.2 Cloudberry

Cloudberry can be described as a middleware between a data management system and the client side of the application. The middle-ware allows for building fast, real-time analytics tools that can work with on large datasets. It does so by sending back requested data to the client application in a progressive manner. Further, it completes frequently performed query requests much faster by caching results of previous series of queries and saving aggregate results. The caching improves the performance by providing potential data intermediates that can be reused to speed up future related queries. From front-end application point of view, interfacing with Cloudberry is relatively simple. It is done by passing a query to the Cloudberry layer in a JSON format as a message through a WebSocket connection. The application listen for messages that are sent back, and deal with the resulting query results. At this point, Cloudberry middleware will perform query optimization before sending the query directly to the backend database.

2.3 Cloudberry on SAGE2

After we incorporated Cloudberry middleware into the ParaViewWeb framework, using the SAGE2 framework, it is a straightforward process to run ParaViewWeb on a LHRD system. As mentioned above, the SAGE2 framework provides a WebView application that allows a web page to be embedded into a simulated web browser window. A single web server can be used to host the ParaViewWeb based application. Similar to any web application, the WebView application in SAGE2 can run this hosted application simply by specifying its IP address. This WebView simulated browser can then be resized, portraying our application across the entire display wall.

As part of the integration of cloudberry middleware and our cluster/display wall setup, the Cloudberry middleware makes query requests to the cloudberry server from the client application. Our SAGE2 is setup with three SAGE2 display clients to manage the display across 24 screens. When an application runs on our SAGE2 framework, there will be three copies of client application physically running on the three display client machines. Therefore, when a query request is made on the display wall client, all three client machines will make independent requests back to the Cloudberry server since each client machine hosts its own instance of the application and display the results. However, due to Cloudberry’s caching capability, multiple requests can be handled with little additional data processing in the middleware layer, with potentially no additional work on the back end data servers. In this sense, Cloudberry enables our SAGE2 based applications to be scalable.
While the web component is the module responsible for rendering, within that component is performed, or frameworks used to create web based view framework, i.e. the way that the HTML rendering

The providers are to be designed in a modular way. That is, each variable, along with a subscription method that publishes events to a very large display. Cloudberry is the tool we use as the middleware between backend data management systems and front end ParaViewWeb applications. The ParaViewWeb framework is the library and ecosystem that allows us to build the front end of the data visualization solution.

In ParaViewWeb, a web component is a module that represents a section or functional unit of the UI. Some examples of web components are a set of controls designed for a certain type of interactivity, or an entire diagram, such as a collection of histograms. They form the building blocks of the UI and are solely concerned with the rendering aspects of that particular component. In order to be a ParaViewWeb component, it must implement a small number of functions that provide the interface that ParaViewWeb expects, such as binding/unbinding of the component to a DOM element, performing the actual rendering, dealing with resizing, etc. As long as these functions are implemented, the web component is agnostic to any web based view framework, i.e. the way that the HTML rendering within that component is performed, or frameworks used to create a particular data visualization. In other words, by utilizing this interface, we don’t have to worry about how the rest of ParaViewWeb operates and focus just on the UI / rendering steps.

While the web component is the module responsible for rendering, a ParaViewWeb provider is the module responsible for managing a particular piece of data, and/or a part of the application state. The providers are to be designed in a modular way. That is, each provider should only concern with providing a particular piece of the applications data and state, such as a provider that manages information about active/inactive data variables being visualized, or another provider that manages 1D histogram data. In a provider module, one must typically define methods for setting/updating various state variables, along with a subscription method that publishes events based on changes in the data or state, so that subscribed modules can receive the data and/or respond to them. This is the crucial piece in the implementation of multiple coordinated views feature of the application.

3 LARGE SCALE DATA VISUALIZATION USING PARAVIEWWEB

In this section, we briefly review the system components of our interactive information visualization application. Fig. 2 illustrates the different components of our Visual Analyzing Billion Tweets application. The main components are the ParaViewWeb framework for creating the client facing application, the Cloudberry server-side middleware, and the SAGE2 environment to allow the client app to operate on a very large scale display.

ParaViewWeb Provider

In this section, we discuss the key steps necessary for incorporating the Cloudberry middleware, as well as the UI components of TwitterMap, i.e. the showcase application for Cloudberry, into the ParaViewWeb API. The integration mainly requires work on the front end libraries. That is, our work focuses on expanding the capabilities of ParaViewWeb to be able to visualize geospatial data, and allow it to interface to the Cloudberry layer on the back end. This integration mainly involves four new API pieces that need to be implemented. Fig. 4 shows the added modules grouped by their spheres of concern.

The first part of the implementation is the development of IO interface between ParaViewWeb and Cloudberry. This part of the API should be general purpose enough to handle any relatively common query that can be accepted by Cloudberry. The second and third parts involves the TwitterMap UI portion of Cloudberry. TwitterMap is not directly a part of Cloudberry middleware itself, rather it is a showcase application that demonstrates the power of the Cloudberry middleware. It is a map visualization that contextualizes the number of twitter instances of a given word or phrase across different states.
we need the with any metadata necessary for maintaining the state of the map, counties in the US. As we are interested in geospatial visualization capability, we added the capability into ParaViewWeb. To this end, the map web component will be responsible for the rendering of this map and UI. Finally, the GeospatialDataProvider and the Map-StateProvider modules will manage and store location data along with any metadata necessary for maintaining the state of the map, respectfully. This data is to be used by the map component, but again, we strive for generality, and the provider should be usable by other components that need to work with geospatial data.

3.4.4 MapStateProvider

The MapStateProvider is designed to work in conjunction with the GeospatialDataProvider. While the latter is responsible for storing the map data, the MapStateProvider maintains the state of the current map settings. For example, it keeps track of the zoom level, map extent, latest map results streamed back from Cloudberry. Together they can be thought of as maintaining the model concerns of the GeospatialMap. This design helps to separate the concerns of the data portion of the model and other metadata that describes the current state of a map visualization. As a consequence, the MapStateProvider and the GeospatialDataProvider do not need to know about each other, and the GeospatialMap view module will simply utilize both of these providers. This module defines various setters to update different parts of the map state. These setters will almost entirely be called by the GeospatialMap module.

4 Visual Analyzing Billion Tweets

In order to demonstrate the capabilities of our integration of ParaViewWeb and Cloudberry, we utilize the new modules described above to create an interactive and responsive client prototype application. Visual Analyzing Billion Tweets application visualizes
The main purpose of this comparative study is to determine the benefit of running Visual Analyzing Billion Tweets application on our large display wall (24-tiled display system) compared to running the application on a normal desktop display (24 inch LCD monitor). The data being visualized was counts of Twitter tweets that contain queried information about tweet counts, by geographical region of interest (ROI) and by time, for any given keyword or combination of keywords. However, in addition to TwitterMap, we are able to stream data not only into a map visualization, but also into any other existing ParaViewWeb InfoViz component. In this demo, we visualize the geospatial mapping, a series of histograms or time bar plots, and a dynamic variable selector. The geospatial mappings provide a spatial visualization of counts by ROI, the histograms provide a temporal visualization of tweet counts over time per ROI, and the variable selector can further be used to narrow the other visualizations to ones relevant to a selected ROI. Fig. 5 shows a snapshot of the Visual Analyzing Billion Tweets application.

Because the map components interface is designed to be compatible with ParaViewWeb InfoViz components, this makes streaming data into all components fairly straightforward and consistent. That is, all visualizations are updated in a similar reactive manner once the streamed data is received from Cloudberry and the respective data models that the visualization components need are updated. From the user end, it looks as if the visualizations are updated in real time to reflect the additional ingestion of data from the server.

Finally, the existing ParaViewWeb libraries make interactive visual synchronization more easily available. For example, in our demo, we can select a particular ROI for further analysis through multiple means, such as by the name of the region (using the variable selector), or by geography (using the map). Selecting an ROI in one type of visualization component highlights the corresponding ROI representation in the other visualization components. Another use of this synchronization is the dynamic updating of ROIs based on scale. Here, this refers to different scales in geographical boundaries (state vs county boundaries). The map allows you to zoom in a region, and the app will reactively render finer grain geographical regions as ROIs. These variables are not only represented spatially on the map, but in all other visualizations. Both of these examples of synchronization makes exploratory data analysis more effective, as patterns and associations can both be hypothesized and investigated from several representations and scales of the data.

5 Comparative Study of Large Screen Display and Regular Screen Display

The main purpose of this comparative study is to determine the benefit of running Visual Analyzing Billion Tweets application on our large display wall (24-tiled display system) compare to running the application on a normal desktop display (24 inch LCD monitor). The components shown are a) FieldSelector - Dynamically select a region of interest to explore, b) CloudberryQueryInputWidget - Enter a search term to analyze, c) HistogramSelector - View tweets counts over time for each ROI, and deselect them to remove them from view, d) GeospatialMap - View aggregate of tweet counts over each ROI, and select an ROI to explore the details related to it.

Figure 5: A snapshot of our Visual Analyzing Billion Tweets application. The components shown are a) FieldSelector - Dynamically select a region of interest to explore, b) CloudberryQueryInputWidget - Enter a search term to analyze, c) HistogramSelector - View tweets counts over time for each ROI, and deselect them to remove them from view, d) GeospatialMap - View aggregate of tweet counts over each ROI, and select an ROI to explore the details related to it.

5.1 Comparative Study Objectives and Protocol

An informal usability study was performed to evaluate the scalability of the interactive information visualizations in our SAGE2, ParaViewWeb, and Cloudberry based application. Specifically, the study focused on testing the effect of having more information displayed at one time (information scaling), as well as having a given amount of information displayed at a larger scale (visual scaling), on the accuracy and speed of pattern finding tasks. To this end, a series of identification and counting tasks were designed to be run on both a normal desktop display screen and the large display wall. The data being visualized was counts of Twitter tweets that contain a given search term.

In terms of the system, Visual Analyzing Billion Tweets application was modified to allow the experiment protocol to be followed more tightly, and to remove UI functionality that may become confounding factors in the result. We ran the application in the SAGE2 environment for both display sizes to maintain consistency in UI look and feel and functionality across both.

We used two visualizations in the study. The first visualization was a choropleth diagram, which visualizes tweet counts over a map, colored by each region, and whose color scale was proportional to the number of tweets. The other visualization was a collection of histograms, which shows the number of tweets over time, over each county. Three experimental environments were designed with these visualizations to probe various questions: (1) a single choropleth map visualization with associated tasks to test the usability of visual scaling, (2) a single visualization of a collection of histograms with associated tasks to test the usability of information scaling, and (3) a combination of the map and histograms to test the usability of both visualized in tandem. The experimental environments are shown in Fig. 6, Fig. 7, and Fig. 8.
The specific tasks that were run are described in the Table 1 below. One of the tasks data (Out of the 5 selected counties on the map, select the corresponding histogram with the most area) was removed from the analysis for reasons described in the next section.

<table>
<thead>
<tr>
<th>ID</th>
<th>Environment</th>
<th>Task Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Map</td>
<td>Select the 5 counties with highest tweet counts</td>
</tr>
<tr>
<td>2</td>
<td>Map</td>
<td>Select the 3 counties with lowest tweet counts greater than 0</td>
</tr>
<tr>
<td>3</td>
<td>Histogram</td>
<td>Select the 3 histograms with the most area</td>
</tr>
<tr>
<td>4</td>
<td>Combined</td>
<td>Select the 5 counties with the highest tweet counts</td>
</tr>
<tr>
<td>5</td>
<td>Combined</td>
<td>Count the number of map regions that were colored by any color</td>
</tr>
</tbody>
</table>

The particular search terms from which the data was generated were chosen such that each resulting visualization contained a distinctive pattern for which the participants could be evaluated. Each corresponding task used the same keyword across both display types, but a different keyword was used for each experimental environment. The visualizations were only shown across the counties of California to limit the study range, and for consistency. The participants were given a short practice session on the normal display to familiarize them with the SAGE2 environment and mouse input system. In the practice session, the participants were asked to perform the same type of tasks as in the actual study. This served to remove any effects that having to learn the task and/or acclimating to the SAGE2 environment would have on the collected data throughout the experiment. Each subject was given the same tasks to perform on both the normal and large display wall. This allowed a paired analysis of task performance on both display types to be performed across each participant. The participants started the tasks on either the normal or large display, in an evenly distributed fashion. That is, the same number of participants started the tasks on the normal display as those who started the tasks on the large display.

5.1.1 Data Collection and Preprocessing

A total of 15 participants from our research laboratory were used in this study. No participants had prior experience working with large displays. Participants rated themselves on how familiar they were with the counties of California used in the study, on a scale of 1-5, 5 being most familiar. 4 participants rated themselves 3, while the remaining 11 rated themselves 1 or 2; so nobody had extensive prior knowledge of the regions used in the study.

For experiment data collection, the participants UI selections and/or verbal responses to each task were recorded. This provided us a measure of how accurately each task was performed. The time to complete each task was also recorded. After all the tasks were performed on both displays, participants were given a survey asking their opinion about which display was better for different criteria, such as speed of completing the tasks, accuracy completing the tasks, and level of high picture / low level details that could be inferred. This served to gather attitudinal data about what participants say about the visual analytic experience on the display sizes.

One of the participants was removed from the analysis due to various difficulties the participant experienced with using the mouse and UI, as well as difficulties with understanding the task instructions. Further, the combined histogram task was removed from analysis because the collected data values mostly reflected error due to recording the task completion time rather than the participants actual completion time.

5.2 Analysis of Comparative Study Data

Various factors of the study data were analyzed. First, to get a general sense of how performance varied between the display types, we compared how time to complete each of the tasks varied across the normal and large displays. We then analyzed potential existence of conditioning effects to the tasks and to the display types. This was to assess if there were any changes in task performance and behavior that may have occurred as the experiment progressed, changes that may not have been accounted for in the practice session. We finally discuss other patterns found in participant performance, including the ability to find small regions, and the ability to keep count of items in the visualization. These patterns further elucidate how users are responding to the tasks on the different display sizes.

5.3 Comparing Task Time to Completion on Normal and Large Displays

To get an overall sense of how long each task occurred on the normal and large displays, we analyzed the time to completion of each given task on the normal display, and on the large display.

Fig. 9 shows a series of boxplots for the distributions of the time to complete each task on the normal and large display. Each pair showing the distributions over the same task but on the different display sizes. Generally speaking, tasks take slightly longer to complete on the large display than on the normal display. Further, the times for the histogram selection task (task 3), and the county counting task (task 5), exhibit the most variance for the time to completion. The map selection task time measurements all have smaller variance.
5.4 Conditioning To Experimental Tasks and Display Size

5.4.1 Comparing Earlier and Later Task Completion Times

We assessed the possibility of user conditioning to the tasks that may have occurred among the participants as the study session progressed. While a practice round was given to each participant to pre-condition participants to the study tasks, this assessment served as an additional check. We analyzed the difference in the time to completion for the tasks for selecting the regions on the map with highest tweet count (tasks 1 and 4), one which occurred a bit earlier in the study than the other.

Fig. 10 shows a boxplot for the difference in times for the task completed on the large display, and for the task completed on the normal display. The data indicates that there is not much time difference between completing the task earlier vs completing the task at a later time. Also, there does not appear to be a significant difference between the time differences on the normal display vs the large display. This suggests that there is no further task conditioning that occurred.

5.4.2 Effects of the choice of display the participants started on

We further studied the effect of the order of the display size in which each participant performed the given tasks on the time to completion of each task. We reiterate that participants were asked to complete the same tasks on both normal and large display sizes, but with the order of display sizes randomly selected.

Fig. 11 shows a series of plots for the time to complete each task that is similar to Fig. 9, but with participants grouped according to which display size on which they started the study. We see that, overall, there seems to be some difference in time to completion of many of the given tasks when first performed on the large display vs when first performed on the normal display. To get a clearer picture, we compute the difference between time to completion of a given task on the large display and on the normal display.

Fig. 12 shows a series of boxplots over these time to completion differences for each task. The interesting thing to observe here is that when participants do the large display tasks first (left), they end up spending a bit more time on the large display tasks compared to the normal display tasks, but when participants do the normal display tasks first, they do not spend more time on either the normal display tasks, or the large display tasks. This is true for the tasks 1-4. It’s important to note that for these tasks, it is not the case that normal display tasks take longer when participants start out with normal display tasks, which can be seen in Fig. 11. In fact, when starting with the large display, the participants take as much time or more time on the tasks than the participants who started with the normal display. This indicates that corresponding large display tasks take longer when done before normal display tasks compared with when performed after. Normal display tasks do not necessarily take longer when performed before large display tasks.

It is worthwhile to note that, while there may not be conditioning as suggested in Fig. 10, the order in which the tasks were performed seemed to matter, as suggested here in Fig. 12. Indeed, Fig. 11 indicates that the participants who had to perform the large display tasks first performed more slowly in both map tasks they encountered (earlier and a bit later). This may suggest that while there is no conditioning to the tasks themselves, there was conditioning to performing the task on different display sizes size (e.g. participants acclimating to the unfamiliar large display).
5.5 Ability to Identify Items on Normal and Large Displays

5.5.1 Finding and Selecting Small Map Details

While we had several indicators of task accuracy, one indicator of particular interest was whether or not participants selected San Francisco in the choropleth map for the map task of selecting the 5 regions with highest tweet count (in task 1). This is indicative because San Francisco is a small region relative to other defined regions in the study, and there could be differences in the ability to detect it on different display sizes.

The results in Fig. 13 indicate that there isn’t a difference in ability to detect San Francisco on the normal versus large display, as a roughly equal proportion detected the region on the normal and large display.

It is noteworthy that, as a general pattern, the people who detected San Francisco took about the same time or longer to complete the study tasks overall, which is observed in both the normal and large display tasks, and can be seen in Fig. 14 and Fig. 15. One explanation for this may be that participants who detected San Francisco were more performant overall, and focused on doing the task well, at the expense of completing it quickly.

5.5.2 Counting Regions on the Map

One quantifier of participant performance we computed was the difference between the participants’ estimated and actual count of colored counties in the count task (task 5). An assessment of the data shows that 3 of the participants estimated correctly on one of the devices, 1 of the participants got the count correct on both, and 6 of the participants consistently either overestimated or underestimated the counts. For these participants, we note that while consistent, it seems as if the counts on the large display was either the same, or slightly higher than the counts on the normal display. There were 4 participants who overestimated on one display size, but underestimated on the other. Here, we note that 3 out of the 4 participants overestimated on the large display. The fact that the count estimates on the large display are slightly greater may be attributable to the fact that some county map regions have non-contiguous regions as part of its makeup, making it ambiguous when counting.

Taking a look at the difference between each participants’ county count obtained on the large and normal display provides more insight. Overall, 10 of the participants had different count estimates for each of the two display sizes, despite both displays portraying the same map and colored regions. Of those 10, 7 of them counted more
regions on the large display map than they did on the normal display. In total, that equals to half the participants counting more colored regions on the display wall than the normal display. On the display wall, many finer details can be observed when looking at the map as a whole due to the resolution and real estate that the map occupies. Because of this, participants were more apt to count smaller regions that they could not necessarily see on the normal display screen map size.

5.6 Overview of User Survey Data

We overview the responses to the user survey completed by each participant at the conclusion of the experiment. This attitudinal data serves as a complement to the quantitative task data, and can help to provide a top level view of users’ receptivity to analytics tooling on a large display. It also helps us to discover important issues that were not considered in the design of the visualization application and large display set up, as well as in the usability study tasks’ design itself. In this manner, the survey data will serve as one piece for redefining successive stages in our research. Specifically in this survey, participants were mainly asked which display, if any, they felt was better in regards to several criteria. They were instructed to base their answers on their experience particularly in the experiment. The survey questions were given as:

1. In which display size did you more quickly find what you were looking for?
2. In which display size do you feel you found more of the patterns of interest and less often missed important pieces of the visualization?
3. In which mode was UI interaction between the histograms and map more intuitive?
4. In which mode was it easier to understand the big picture?
5. Which mode allowed for greater understanding of fine detail?
6. Which mode did you prefer?

Fig. 17 shows the distribution of responses to each of the survey questions. The vast majority of study participants felt that they could more quickly and accurately find what they were looking for on the large display wall, and preferred the display wall in general for being able to see overall patterns and specific details.

The survey also included a space for users to provide free responses as to why they preferred one type of display. Some of the reasons the users gave for preferring the large display include being able to see everything at once without needing to scroll, being easier to compare amongst regions, having more granular detail making it easier to discern between counties, and being easier to understand/see the data, especially with a high resolution data set. These are all points that we hypothesized are advantages to the large display. However, we note that the users survey responses do not tell the same story as the quantitative task data. For example, the time to completion of tasks in the large display seems to be greater than the time to complete on the normal displays, as was shown in Fig. 9.

5.7 Discussion

Given the responses from the study participants, one would expect users to perform more quickly and accurately on the large display wall. However, our results show that the display wall tasks took longer than the normal display tasks on average, and the accuracy of the tasks completed was very hard to assess. Creating a study with more quantifiable tasks, and/or more difficult tasks could help uncover this desired result in the future. Also, more study participants will probably be needed so that out of ordinary study sessions can be thrown out. More participants will also help to get a better general consensus and to see more concise trends in the data collected. The data is not very conclusive in either direction and we can only pose our own speculative analysis here.

As has been noted, the survey responses were majority in favor of the large display wall being quicker to find what the users were looking for, despite that not necessarily being true. The positive response from users in favor of the large display wall seems promising and shows that users may opt to use the wall as a data analysis tool. The slower task completion time is a little disconcerting, but there are many factors that could contribute to this. Some of these factors include the users unfamiliarity with large displays, the amount of data being portrayed at once being much greater than the amount seen on a normal display, not knowing what or where to look at first one the wall, as well as the case where something catches their eye and they reevaluate their initial selections, leading to deselecting and reselecting regions or histograms. On the large display wall, all data can be considered at once and in comparison with each other, so although it takes longer to analyze the data, the response may be much more accurate due to the comparison of all angles. On a normal display, they may have been quicker to choose a response since there was less data shown on the screen at one time; however, the response may not have been accurate since data was missing and not considered. This leads into how do we define better and more accurate. Unfortunately, due to ambiguity in task requirements and lack of a concise answer for many of the questions posed, the ability to capture an accuracy rating evaded us. It is overly simplistic and incorrect to conclude that the more time to complete a task implies that that display size is worse. Thus, in future iterations we need to have a better way to analyze the correctness and accuracy of the users tasks to be able to compare against the time to completion.

5.8 Future Work

The user study conducted was a first iteration to collect initial feedback from users on normal display size versus large display size. After conducting the study, we have learned just as much about the study design as we have about our desired results. The main goal for future iterations will be to make the tasks less ambiguous with a means to collect and evaluate the accuracy of tasks performed. We also hope to pose survey questions that better reinforce instead of contrast with task performance. Much of the future work will also be attributed to means of collecting and portraying the data for the study. We plan to study what kind of pieces or slices of data are the best to show in an incremental streaming fashion. In addition, we
plan to compare different sized data sets to see if the large display may be preferred for large datasets whereas the normal display may be more appropriate for small, sparse datasets. This could help us answer the question, How much data is too much data? With more sophistication and planning involved, one potential way to collect more relevant data could be tracking the users eyes as they conduct the tasks on the large display wall. This could provide insight into why the large display tasks take longer to complete. A last thing we would like to test in future iterations is the full complex system. This would include more visualization types than just the choropleth diagram and histograms, as well as testing SAGE2, Cloudberry, and ParaViewWeb individually and in unison.

6 Conclusion

We presented our Visual Analyzing Billion Tweets application for performing visual analytics over large resolution and big data, for geospatial and temporal visualization of Tweeter data. Our main goal is to demonstrate the viability of using similar technologies for development of similar scalable visual analytics tool to analyze heterogeneous data from the U. S. Army Test and Evaluation data community. We also ensure continued SAGE2 compatibility by iteratively testing the application on SAGE2’s WebView. Taking advantage of scalability of SAGE2 framework, our information visualization application is capable of running on various display system configuration; single display system or multi-display system. Through a comparative study, we demonstrated the benefit of running our Visual Analyzing Billion Tweets application on our LHRD system compare to running the application on a 24 inch monitor. We also demonstrated the feasibility of quickly prototyping different type of visualization and visualization functionality using the ParaViewWeb API. Notably, our LHRD geospatial visualization has multiple coordinated views, and is tailored to work in SAGE2, together which facilitate similar user experience compare to a desktop display but with added benefits of LHRD for data exploration. The resulting seamlessness of transitioning into this new user experience should encourage user adoption of Visual Analyzing Billion Tweets application for visualization.

Acknowledgments

This work was supported in part by the DOD High Performance Computing Modernization Program at The Army Research Laboratory (ARL), Department of Defense Supercomputing Resource Center (DSRC).

References


