An Interface Design for Future Cloud-based Visualization Services

Yuzuru Tanahashi       Cheng-Kai Chen       Stéphane Marchesin       Kwan-Liu Ma
Visualization and Interface Design Innovation (VIDI) Group
University of California, Davis

Abstract

The pervasive concept of cloud computing suggests that visualization, which is both data and computing intensive, is a perfect cloud computing application. This paper presents a sketch of an interface design for an online visualization service. To make such a service attractive to a wider audience, its user interface must be simple and easy to use for both casual and expert users. We envision an interface that supports visualization processes mainly directed by browsing and assessing existing visualizations in terms of images and videos will be very appealing to, in particular, casual users. That is, the aim is to maximize the utilization of the rich visualization data on the web. Without losing generality, we consider volume data visualization applications for our interface design. We also discuss issues in organizing online visualization data, and constructing and managing a rendering cloud.

1 Introduction

Cloud computing is deemed to empower the next generation web applications in several aspects. We consider its high computational power can be utilized not only in retrieving information efficiently but also in yielding new information. As the quantity of data on the web explodes, visualization, which transforms numbers and text into vivid pictures, will be a vital tool in most web applications. We also anticipate seeing a variety of online visualization services. We believe, as visualization resources become richer on the web, one’s visualization needs may be satisfied in two ways. First, the user can simply use existing visualization in terms of images or videos found on the web. Second, he can request a new visualization to be created by a visualization service. Both ways require a user interface through which the user specifies the content, look, and quality of the visualization. In this paper, we sketch the design of a user interface for future online visualization services.

A search engine can easily find existing visualization resources including images, videos, and raw datasets. However, to offer an online visualization service, such resources must be carefully organized and indexed to facilitate interactive visual exploration and selection by the users without requiring them to linearly scan through thousands to millions of found items. We advocate an interface enabling the user to obtain the exact visualization of interest or a collection of visualization contents that are similar to the desired kind through a visual-based searching method. In the latter case, the service utilizes the computation power of the cloud and computes the possible desired visualizations based on the metadata stored with the collection of similar visualizations to provide the user with rich examples. Although the concept of visualization by example is not new, the realization of this concept may be done in different ways. In the rest of this paper, we introduce our interface design while taking into account the overall architecture of a prospective online visualization service. We use volume data visualization to derive our design, but the derived design principles will be applicable to other types of visualization tasks. Internet-based computing is becoming so pervasive and powerful that it will free us from investing on and maintaining expensive technology infrastructure for demanding applications like advanced visualization. However, the usability of Internet-based applications is largely determined by properly addressing the corresponding HCI and interaction design issues. Therefore, our study also focuses on such matters.

2 Related Works

Considerations for user interface and interaction for visualization have received great interest in the visualization research community. Many [3, 6, 7, 8] have realized the concept of visualization by examples. He et al. [3] show how volume visualization can be accomplished as an iterative process of viewing and selecting from visualizations generated by computing stochastically in the transfer function space. Through this process it is possible to derive an ideal transfer function. Ma [8] presents the Image Graph design which enables sharing of visualizations and the processes to create them, and allows construction of new vi-
ualization from existing ones. Jankun-Kelly et al. [4, 5] introduce a spreadsheet-like interface for volume visualization; through this interface, the user can search for ideal visualizations by exploring in the visualization parameter space enumerating different transfer functions, views, etc. Lu and Ebert [7] present a method to generate volume visualizations by emulating example illustrations. Liu et al. [6] called their work “Visualization by Examples” but what they did is essentially to use a spreadsheet interface for composing complex visualizations from simple ones generated automatically. In addition, online visualization sites such as Many Eyes [11] have broadened the scope of visualization by enabling multiple casual users collaborate on setting up and discussing information visualization. Grochow et al. [2] studied how to best utilize client’s local processing power and cloud resources to support demanding visual data analytics applications.

3 Framework

Figure 1 shows an overview of our system. Our system is entirely web-based; that way the end user does not need to download and install a visualization tool on his computer. In particular, this enables accessing the application from a wide range of terminals, including mobile devices such as smartphones. Also, by removing the need for installing visualization tools and learning how to create effective visualizations, we consider that our system will enable casual users such as high-school teachers to compose effective and useful visualizations.

The system architecture is as follows: in a preprocessing step the data is indexed by the data server and stored in a database. During this step, as much information as possible about the data is collected and derived. We discuss the issue of metadata information in more detail in Section 4, including ways to reconstruct missing information.

To use the system, the user connects to the server through a web browser and types his request like in any other web search engine. A series of matching datasets, images and videos are returned to him. He can then modify the visualization parameters through an intuitive interface, and at every step a new rendering request is transmitted to the rendering servers in the cloud and a new image or a new video is generated. The result is then sent back to the user who can make further refinement on the parameters.

4 Indexing and Metadata

One of the most critical problems in constructing an online visualization system is the difficulty to deal with various data formats and rendering algorithms. This problem not only gives the domain experts a hard time but also is a big reason keeping casual users from participating and relishing visualizations.

Therefore, we need a way to organize and retrieve data in order to efficiently render visualizations and respond to user queries. One way to achieve this is the use of metadata. In the context of world wide web, metadata is crucial for searching and semantic processing. Similarly, we organize and use metadata for more effective online visualization.

We thus augment all the images, datasets and videos with metadata. For example, the metadata of a volume rendered image would include the original volume data, camera position, transfer function used to generate the picture, and the creator. and the metadata of a volume dataset would include the information about its format, dimensions, etc. As such, we augment existing visualization resources in the cloud.

Because this information is not always readily available, we expect visualizations retrieved for indexing will have some metadata information missing. However, some of these unavailable visualization parameters are inferable from existing images or videos. Magnor et al. [9] use graphics acceleration to reconstruct the optical properties of planetary nebulae using genetic algorithms. We intend to apply similar algorithms to retrieve the original parameters of a visualization. Good approximations of visualization parameters such as transfer functions and camera positions are most likely recoverable in this fashion. Due to the fact
that the underlying genetic algorithms required for this task are compute-intensive and trivially parallel, it is a good candidate for cloud computing.

Once we have the metadata, we need to organize it for efficient access. This processing task is also best done in cloud due to its potentially large scale. In our design, volume data, images and videos are stored in an SQL database with their associated metadata.

5 Interaction and Interface

Our system enables the user to effectively search through a vast repository of visualizations and retrieve or create their own visualizations. The following subsections describe the user interaction and interface considerations for supporting these tasks.

5.1 Searching

Searching for visualizations is achieved in a conventional manner, such as entering a request into a search box or selecting items through indexed categories. However, the way properties are selected by the casual users differs from that of the experts. For example, while expert users typically search for a specific property, which has analytical significance, casual users would mainly make their choice based on the aesthetics of the final visualization.

Therefore, in order to facilitate an effective exploration, our system utilizes the metadata information of the found visualizations to cluster them in groups. The user can then intuitively explore various visualizations which are clustered according to the current attributes, such as color, rotation, and dataset.

5.2 Suggestive Visualization

While the initial searching provides the user with existing visualizations, as the user selects a visualization from the clustered results, our system automatically computes and generates a new set of visualizations in the cloud that the user may prefer. Such visualizations may include changes to the dataset, the transfer function, the viewpoint, or a combination of those. This situation is depicted in Figure 2. In this figure, images with a red frame are newly generated suggestions, while other images are pre-existing visualizations. The initial set of choices is depicted as clusters on top. After the user selects a cluster by clicking the image highlighted in blue, a new clustering is computed which is shown in the bottom.

We use the term "Suggestive Visualization" to differentiate it from visualization recommendation, which implies the response being a set of visualizations that already exist.

Figure 2. Clustering for suggestive visualization. Initially the user is presented with a number of choices (top). After the user selected a desired visualization, highlighted with a blue frame, a refined set of choices is generated (bottom) allowing the user to refine his selection.

5.3 Iterative Interaction

There are two primary interactions in our system, and the final visualization is achieved by a sequence of these manipulations. The first kind is changing the visualization parameters directly. This is a common way of interacting with visualization applications. The second kind is selecting visualizations from the provided collection of visualizations.

As the complexity of the data, the phenomena to be captured by the visualization, and the visualization methods to apply continues to grow, advanced knowledge about the application domain and visualization techniques are required to generate satisfactory visualization results. For casual users to make compelling visualizations, it seems only feasible through selecting from suggestive visualization based on image evaluation [3, 6]. Consequently, the second kind of interaction is anticipated to receive more attention and popularity.

Figure 3 is a sketch of our user interface. Area A depicts the current visualization and its properties. The user then chooses one cluster of suggestions from area C. Then the contents of this cluster are shown in area B, and the user selects the most suitable item. The system then discards the current search results and provides the user with a new
Figure 3. A mock-up of the user interface. Area (A) provides the user with the primal interface for operations such as undo, redo, and save. It also displays the current visualization, along with the adjustable metadata. Here, we show the transfer function and the viewpoint control. Area (C) displays the thumbnails of the clustered suggestive visualizations. The current selection is highlighted with a red outline. The selected set of visualizations is presented in Area (B). The current set of suggestive visualizations with various transfer functions applied.

set of suggestive visualizations calculated based on the selected visualization. This process is repeated until the user is satisfied with the generated visualization.

6 Rendering in the Cloud

We propose to use the computational resources of the cloud to compute the set of suggestive visualizations. After the user selects a desirable visualization, our system submits the corresponding settings of the visualization, including its indexing and metadata properties to the rendering cloud to produce a collection of suggestive visualizations. For this set of visualization parameters, the rendering cloud explores different combinations of parameters, and determines the relevant ones based on its built-in criterial and empirical rules. The settings for these suggestive visualizations are then distributed to each individual server for rendering.

In a previous project [10], we have implemented a rendering cloud for interactive and high-quality visualization of large-scale volumetric datasets. The rendering cloud utilizes MapReduce model to take full advantage of the distributed computing resources in the cloud. We believe that MapReduce is a good fit for generating suggestive visualizations since the rendering for different settings are independent and can be trivially parallelized. The two main components of the MapReduce workflow are Map and Reduce functions. In the rendering cloud, a given data is bricked into small pieces and sent to each Mapper for rendering. Once the Mappers have finished the task, all the rendering fragments are sent to the Reducer for composition.

In order to achieve high performance rendering for large-scale data in the rendering cloud, a number of challenges have to be overcome. For instance, before the cloud starts to process the suggestive visualization, the data needs to be transferred to each rendering server over the network, which could incur additional latency due to network congestion or other unexpected situations. One solution is to minimize the amount of data that must be transferred over the network. By indexing the data locations, our system could enable each rendering server to obtain the data from the closest resource in the cloud. Thus, the cost due to the data migrations over the network could be alleviated.

7 Conclusions

In this paper, we present design considerations for online visualization service based on cloud computing. Our interface design mainly targeting casual users was made to
maximize utilization of the rich visualization and growing cloud computing resources that we anticipate to have. At the same time, the layout of our system succeeds in seamlessly integrating existing and derived data (and visualizations) to support the user’s task. As we have mentioned, we expect such utilization of the cloud will become more common in the coming generation of web applications.

We have identified a number of key components which we believe are crucial to the robust realization of a usable system:

- A simple and intuitive interface
- An efficient data indexing system
- A suggestive visualization system allowing the user to refine its previous results in a relevant fashion
- An efficient implementation of visualization techniques using the MapReduce paradigm.

Because our system requires little configuration and domain-specific knowledge for use, we expect it to see wide usage from people who would not have access to visualization otherwise. Similarly, the use of visualization for large datasets has so far been mostly restricted to domain scientists with expensive visualization clusters. Since we attempt to leverage cloud computing resources, we are potentially able to handle visualization at large scale without requiring the users to invest in their own visualization clusters. As more people gain access to the use of advanced visualization techniques, we expect that one of the key values of this work will be to make visualization tools a much more accessible. By a cascading effect, this should enable the use of visualization tools in a much wider range of situations and to a much bigger audience.

References


