Report for Design Challenge 1 - CS838-2

James Hill, Ye Liu, Shuang Huang

Description of Visualization

According to the interview and details described by the domain scientists, the major difficulty in the domain research is to determine the relationships between Epistemic Frames. Thus, in our opinion, the most important purpose of our design is to clearly show the comparison of frame matrices while providing detailed comparisons for each element in the matrices if needed. Therefore, we decided to create a hierarchical multi-interface application written in C++ including three layers of view for the frame data.

The first layer of view was an attempt to show distances and clustering of frame data (matrices) with respect to a reference (a matrix chosen by the users). By clustering the frame data, this application uses distance to compare the significance of differences between peripheral frames to the center (reference matrix). The longer the distance, the larger the difference. According to the domain scientists, the absolute value of length may not be very meaningful, but the clustering shows which groups of data frames are close, i.e. probably implying closer relationships or correlations. Our initial difference function simply sums the absolute values of the differences between each element of a given frame. Our design provides an interface for users to define their own distance functions so that the effects of different distance measurements can be explored. If the user is interested in the relationship between two elements, he is allowed to select the specific matrices and look into their differences element by element with the second and third layers of the view.

The second layer of view is an attempt to show distribution of differences in two frames of data. We decided that color saturation might be an efficient way to represent the absolute difference of each element. For each pair of frames (matrices), we create an identical matrix brick diagram and map the color saturation of each brick in the diagram to the range of difference between the corresponding elements of the target matrix element pair. This feature gives the viewer a quick visualization of the distribution of the differences between each element.

The third layer of the view was a direct comparison of each element of each matrix selected from the first layer. Similar to the second layer, a matrix brick diagram is created, but now each brick is used as an area for a bar chart generated from the value of the corresponding elements of the selected frames. This view gives the audience a detailed understanding about how and which parts of the several data frames are different. It allows multiple comparisons for more than two data frames, which is advantageous.

Technical details

The implementation of the tool was done in C++ using the open source windowing library Qt. This library provided us with the functionality to quickly build a relatively stable tool that allows the user to load and explore real data.

All three views are custom made Qt widgets. While they were a lot of work to produce, the do allow for a certain amount of polish in the final product.

Source code is available at: http://pages.cs.wisc.edu/~jshill4/projects/cs838EpVis.html and can be compiled on any platform provided the correct compiler and libraries are installed. Details for compilation are included with the source files.

Discussion Feedback

For discussion feedback, we read through the comments, summarized the questions asked and suggestions given, and then tried our best to reply according to our point of view during the implementation of the application. In the following, Question (Q) sections are the questions generated from the comment, while Answer (A) sections are our proposed reply.

Q: In the first visualization, why do you encode the frames in the second dimension and not the first as distance is a one dimensional metric?

A: 1D visualization, of course, would be more concise and might more clearly show the exact distance of two matrices. But we feel that the 2D vision is helpful in showing clustering, and is more invulnerable to the cluttering problem as it allows more area for each node (frame data) to occupy, and also properly emphasizes the reference frame. To our understanding, the purpose of the visualization wasn't to show exact distances, but to show the relative distance (correlation) of frame data.

Q: Why do all of the colors in the second view look the same?

A: We are doing this on purpose because a large difference would stand out as a more saturated red or green color. If a user is able to tell the difference between two saturation values, then he might know there is a large spike in the more saturated element.

Q: What is the scalability of the third design?

A: Scalability is the intrinsic problem of our third design. We believe it scales well for 2 ~15 frames. But it does require a large viewing area when the number of frames increases. We can try to dynamically adjust the width of the bars, however, with limited area, cluttering will eventually be a problem with a large number of frames.

Q: Your third design conceals any global information about the frame data, why is that?

A: As we explained before, our designs are not independent, but in a hierarchical system. First we would like to help the users to examine all frames at a global level, then he can compare the difference between two frames entirely if needed, then he can perform an element by element comparison to extract the inner data. That was our major goal for the design of the third visualization.

Q: The colors need to be consistent and there needs to be a key.

A:We agree and will possibly incorporate a standardized color scheme in a later version.

Q: For the first design, you should have added weight to each node (frame).

A: We considered that, but we decided the explicit display of weight by distance to the central reference node was the best channel.

Q: You should ghost the columns and rows in the second and third visualization.

A: We agree. It would be a good way to reduce visual cluttering. We will try to add this mechanism in a future version.

Q: You might want to use gradients in the color map.

A: We thought about this method, there are indeed merits to using gradients which were demonstrated by a different group, but they're equaled by the merits of not using them since we are really looking at graph data and not sampling data from a higher resolution source. There is no meaning to values "between" the node connections which would be implied by gradients.

Q: You might want to show direction on the first layer of design.

A: We technically show the magnitude of the distance, and most of the properties might depend on the actual distance function defined by the user.

Q: In the first layer of your design, why don't you use color coded nodes to represent each matrix?

A: We believe that color coding each node would clutter the image and disturb the prominence of the reference node. Instead, we added dynamic labels for each node. If the user wants to locate a node, he can mouse over it. We can also added a "search" function in the future version of design, so a user might be able to easily locate a certain node in a large amount of data.

Q: You should remove your redundant data.

A: We agree and will try to implement that in our next version.

Q: You might want to show labels for each frame in your first design layer.

A: We considered that, but there doesn't seem to be a good way to do this, since the nodes will consume too much space and the cluttering problem will get worse with increased frame data.

Discussion of Problems

As pointed out by the comments and discovered by ourselves, there are a number of possible improvements that could be made to our tool. The following are some of the more prominent:

First, we do have a lot of redundant data. In the second and third layer of our design, we are using half of our resource to displaying redundant data. Removal of the redundant data would be relatively easy, but how to efficiently utilize the resources spared by redundancy removal might be a problem for future refinement. For now, we believe the redundancy has not prevented the user from seeing important information.

Second, since we rescaled/normalized the distance for each reference node to make the best use of space in the first visualization, comparing two distances with different reference nodes is difficult, and a consistency problem arises for identical distances. So we are considering using a global scaling factor.

Third, the third view has no key for determining which frame an element belongs to. This is a requirement for successful usage of the tool and will need to be added in a later version of the software.

Fourth, scalability in the third layer of our visualization will eventually be a problem with a large number of data frames. As this problem happens in most visualizations when the number of matrices being compared increases, we can only come up with solutions such as dynamically adjusting the width of the bars, but a more effective method is needed.

Fifth, as pointed out by the audience and the domain scientists, colors should be consistent across different runs of the third layer, otherwise it will cause unnecessary confusion to the user.

What can easily be seen with this visualization

In the first layer of the visualization, we can clearly:

1. observe clustering of frames given a reference frame
2. get a general idea of the relative relationship of frames to the reference frame, i.e., which frame is most related to the reference frame? Which frames are not related?
3. get a general idea of which frames are outliers

For the second layer of visualization, we can clearly tell:

1. Between the nodes of two matrices/ frames, where the greatest differences are.
2. How the differences are distributed over the frames.

For the third layer of visualization, we are able to:

1. Compare each element of different data frames in a very precise manner;
2. Obtain a very detailed view of the differences between each data element

Usage of Design Principles

First, according to the requirement of the domain scientists, we thought a hierarchical system with several layers would be easier for the user to interact with, i.e. observing relationships at a larger scale, finding interesting areas and subjects and comparing them in detail to extract the inner data and view the source of the difference. Thus, we developed a design with a hierarchical concept, with a layer for large scale comparison between data frames, difference distribution between two frames, and a detailed view and comparison for each element of the frame data.

For each layer, we have specific concerns according to the design principles. For instance, in the first layer, since most frame data is quantitative, encoding them with position and length is a good choice (Munzner, Fundamentals of Computer Graphics, 2009 pg. 683).

In the second layer, we consider each block as a representation of the difference between it's two frame elements. While it is possible to order the blocks, we wanted the user to categorized the blocks based on the differences in element values. For that reason we considered this data categorical and used saturation and hue to encode the data.

For the third layer, we display large sets of quantitative data, thus using length would be a better choice as our area is limited. So we came back to conventional bar charts which are very easy to understand and compare, and relatively easy to be extended for multiple comparison.

Team Dynamics

We held three one hour meetings where we laid out the idea for the visualizations. We discussed the requirement of the domain scientists, and according to the interview, we thought a hierarchical system with several layers of design would be easier for the users to “use”, i.e. observing relationships in a larger scale, finding interesting areas and subjects and comparing them in detail and extract the inner date to view the source of the difference. With this principle, every individual in the group brought out a design. Most of the implementation of designs is developed by James in C++.

A lot of the design was done in the three meetings, however many emails were sent back and forth with the goal of refining our designs and determining when meetings would be held.

The most difficult part of the team dynamic was finding times that worked for all members. We where able to accomplish this and we believe our development process has been very successful.