Executive Summary to: Gesture Kinect Data Activity Visualization

JOSEPH CALUZA, CRISTIANO FAUSTINO, & ZHEN ZHENG, University of California, San Diego

A. PROJECT INFORMATION

This section is for the purpose of describing various thoughts about the design process for the Activity Viz project.

A.1. Project Title and Members

The name of this project is called Activity Viz because of the task of visualizing kinect data. This project was completed by Joseph Caluza, Cristiano Faustino, and Zhen Zheng of Group 2 in Spring quarter of 2014 in the Cognitive Science 121 with collaboration with the Activity Visualization group from Cognitive Science 102C.

A.2. Key Problems Addressed

The gesture research group at the University of California San Diego uses the Microsoft XBOX Kinect to study body movement for American Sign Language (ASL) and also other full body gestures for non-ASL users. According to interviews conducted from the Activity Visualization group from Cognitive Science 102C, the researchers are not fully content with their current software for viewing and annotating data. Furthermore, the researchers have to write their own MATLAB scripts in order to translate Kinect comma separated value (CSV) files into readable data by their current software. Though it is a working method, it causes a hassle for the researchers and the ending result visualization is not always helpful when coding gestures. Activity Viz is an attempt to directly interpret the Kinect data and visualize the information from the CSV files in a way that is both simple and detailed. The current released version of Activity Viz is meant to be used as peripheral application to the researcher's current software in hopes that will be helpful to them until a final version is ready.

A.3. Design Ideas

When first assigned this project, the group did not know exactly what kind of data needed to be visualized or what a gesture researcher might want to see when they use the Activity Viz application. When the group finally had the opportunity to meet with the 102C Activity Visualization group, some confusion was alleviated but no one really knew what direction to take this. The first step we took was to try and understand the user base. Since the application would be used to revisit past Kinect trials, we built around what a gesture researcher might want to do. We thought of an application where a researcher could create an account. They would then see a form where they could upload three files: A CSV with video annotations corresponding with time, a CSV containing the Cartesian coordinates, and a video of the trial. The initial plan was to use Youtube API to allow the user to upload videos to Youtube to allow for later viewing. However, this idea was immediately rejected because of the researchers' need to maintain user subject confidentiality. We then thought that we would take the coordinates and times within the CSV to create an animated representation of the trial along with annotations along the way. Due to time constraints and our lack of knowledge concerning 3D animation, we also ruled this out. As a result, we decided to create one 3D visualization of the coordinates on a 3D graph which could then be filtered by body part. This would serve as our main key feature for the final submission and allows for expansion and ideas for future implementation.

A.4. Implementation

The first thing created for this project was the creation of users and setting up a working user authentication system to make sure each researcher's work remains private. This was accomplished by using Node.js to handle server side routing and for its extension libraries. Such libraries include Mongoose for creating user schemas that would save new users to the database created using MongoDB. We then used Passport.Js to handle user authentication. The next element implemented was the 3D visualization of Kinect data. X3Dom JavaScript library was used to handle 3D manipulations and the D3 JavaScript library was used to render a graph from the CSV file. An algorithm was written in JavaScript to smooth out gaps in the CSV data and store data in an array of JSONs for use by D3. Once functionality was making progress, we used the Bootstrap framework to quickly create a nice and clean user interface.

A.5. Features

This application is intended to serve as a peripheral to existing software used by gesture researchers until a more complete version is released. As a result, the main features are the ability for a researcher to visualize trial data and compare them to previous trials. The visualization is also in 3D which was previously unable to be accomplished in MATLAB and with their current software. This makes it easy for the researcher to generally see where each body part is moving.

A.6. Evaluation

As mentioned earlier, this application is more of an example that there could be a better way to visualize Kinect data. From what we have seen on the internet, this is one of the first attempts to tackle a problem involving visualizing gesture data. That being said, the way our 3D visualization is implemented right now needs a lot of rework and revisions. Some pieces of the code are hardcoded to prevent it from failing. The algorithm used to smooth data gets increasingly difficult to implement with large datasets. The current data we were given to work with has over 11,000 object entries. This causes some older browsers to lag when the visualization loads. Overall, it works for the given data but we are unsure how it will react to other datasets.

A.7. Future Directions

Gesture research seems to be growing with emergence of motion detecting devices such as the LEAP motion and the Kinect. As a result, more projects like this may emerge. For Activity Viz in particular, we ultimately hope to create a collaborative research application that will help emerging gesture research labs to have access to more trials and results (if subjects give consent to having their experiment data be distributed anonymously). This implies that we would hope to have support for more data files, video annotation, animations, and a more detailed and filterable 3D visualization. This trial would then be stored and then the researcher could compare two or more trials. The possibilities are endless and we hope that Activity Viz can help foster more ideas.

Gesture Kinect Data Activity Visualization

JOSEPH CALUZA, CRISTIANO FAUSTINO, & ZHEN ZHENG, University of California, San Diego

1. INTRODUCTION

The Kinect device has made its way into not only the gaming world but into the field of many researches. For example, many Gesture Researchers are taking the advantage of the Kinect's abilities - record video, audio, and the depth perception of objects - to conduct researches such as decoding hand signs or analyzing patients' activity in a hospital environment. Therefore, we are dedicated to improve the usability of Kinect data in gesture researches. The project is mainly involved a set of raw Kinect data that is in a format of CSV, and we will be turning that raw data into a visualization that will be user-friendly and beneficial for our target users – Gesture Researchers. In this paper, we will explore and touch every stage of progress and development of our project.

In section 2 we will discuss the motivation behind our project and some background information about Gesture Researcher with Kinect in a research environment. It includes the specific function of the Kinect, the existing technology that has been used by Gesture Researchers, and the evaluation of these products and Gesture Researchers' experience that will ultimately provide us with the motivation to improve the usability and user experience of Kinect data.

In section 3 we will go cover our progression of design ideas from our initial prototype to our final product. It includes our inspiration of choices of all design features in our prototypes, the pros and cons of our prototypes, how each prototype evolved over time, and the final design of this project.

In section 4 we will go into technical details of our system/application development. It explains from the general infrastructure of how our application works to the specific technical detail of what technique and data sources have been used for each of the features on our application, both front-end and back-end.

In section 5 we will explain the Human Computer Interaction principle of design on each element of our application. It will explain the reason of why every little thing on the application, such as an element's size, location, and function, is not there by accident, it is by design.

In section 6 we will cover evaluation and testing of our application. It includes what testing method we have used and the how each result impact our design.

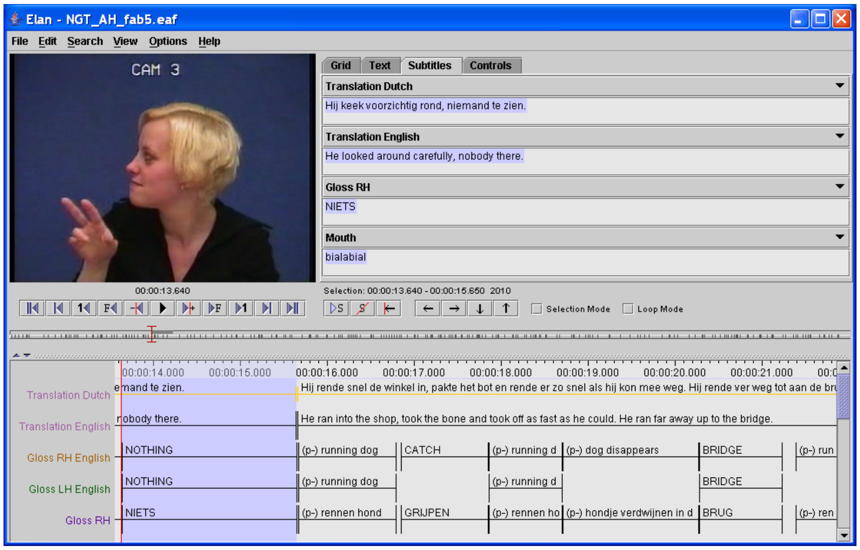
In section 7 we will discuss about the collaboration within our own group COGS121 and the collaboration between the two teams - COGS121 and COGS102C. We will go over each team member's contribution and what kind of collaboration tools we have used to keep each other updated. It also includes some of the issues we may have raised during the collaboration.

Lastly, in section 8 we will conclude our project report with a brief summary and discuss the possible future of this project is heading toward.

2. MOTIVATION AND BACKGROUND

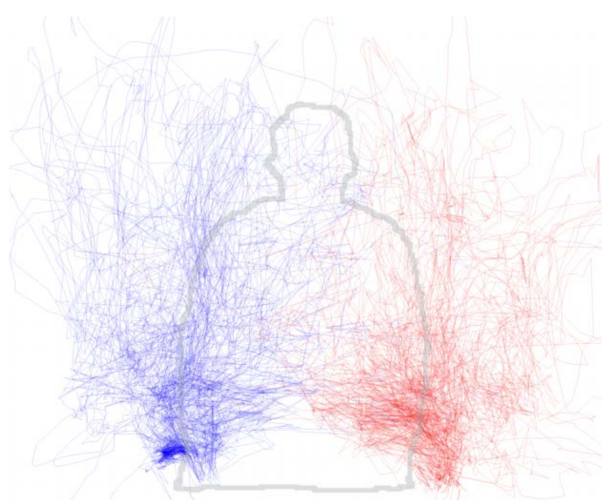
The Kinect device has the capability to capture video and audio, and it also generates body position data of a person over time (including X-Y-Z positions points). It generates raw position data in a CSV format file. Therefore many Gesture researchers are using Kinect to analyze and study hand gestures. With the collaboration with COGS102C group who gather user information from real Gesture Researchers by conducting contextual interviews, we are able to find out that Gesture Researchers are already using various existing desktop programs to analyze hand gestures. For example, the interviewed Gesture Researchers are using desktop programs such as ELAN and ChronoViz to decode hand sign gestures, transcribe audios, and make annotations on video and audio. However, the collected data and the Gesture Researchers' experiences suggest that the existing programs that are being used are not very user friendly and they are not customizable to meet the researchers' needs -

they claimed that they often have to work around the features of the programs or the programs do not have or support the features that they need for the experiment (see Figure 1).



(Figure 1)

Consequently for this reason, we believe that it is necessary to improve the existing program by designing a more flexible program for the Gesture Researchers. Initially, the obvious solution was to redesign a program that will allow Gesture Researchers to fully utilize the potential of generated Kinect data and to provide customizability for the Gesture Researchers. However, due to our technological skill limitation, resources, and time constraint, we were not able to produce such high quality functional product within just a few short weeks. Therefore, we've personally met with one of the interviewed Gesture Researchers, Dan Lenzen, to collect more information regarding the tools that he is using for his experiment. We found that there is a specific need in the visualization area of the program. During Dan's experiment, he has to filter out the Z-coordinates just to keep the X- and Y-coordinates, so that he can generate a 2-Dimensional visualization for himself to analyze the experiment (see Figure 2). Similarly, other Gesture Researchers encounter other issues with the visualization as well - such as they are not able to directly manipulate the visualization.



(Figure 2)

Finally with all of the newly gathered information, we believe that Gesture Researchers are not fully accessing the full potential of data that the Kinect generates. Therefore, we have decided and devoted to create an add-on application that is dedicated to Kinect data visualization - Visualization that will fully process X-Y-Z-coordinates and will support direct manipulation by Gesture Researchers.

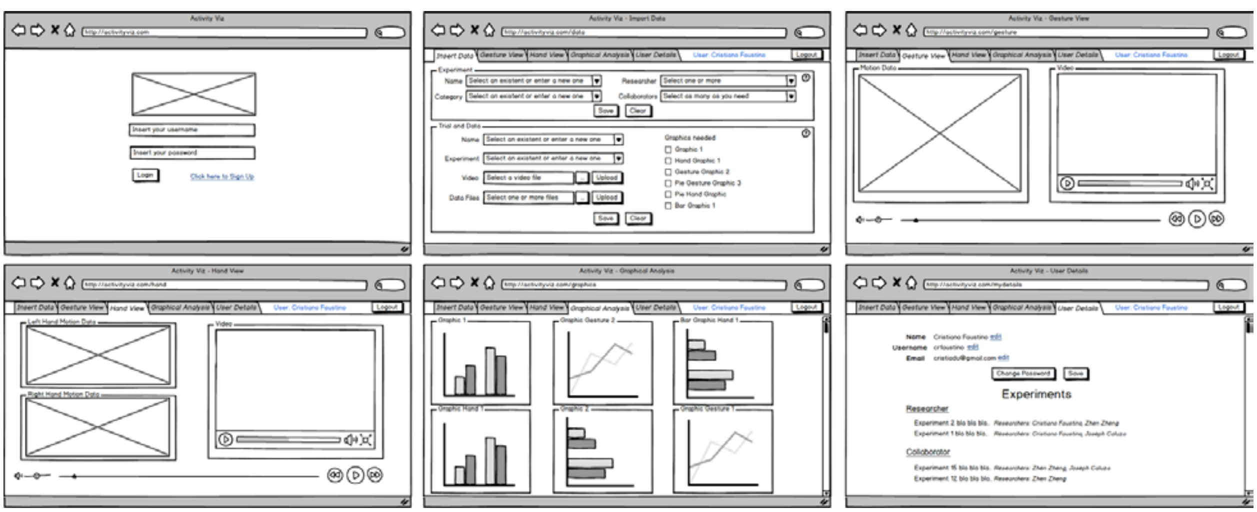
3. DESIGN

Our design ideas mainly came from the needs of our target users, in which case our target users are Gesture Researchers. We have collected valuable information regarding their work-flows and models through both in person interview and from our COGS102C team members who had conducted a few rounds of contextual interviews with the target users. As mentioned earlier, our initial idea was to redesign a program that will provide more accommodations to the Gesture Researchers because of the following reasons:

i. Gesture Researchers have to work around the features of the program.
ii. Programs do not provide that features that Gesture Researchers need.
iii. Features of the program are not customizable (i.e. video screen size is not adjustable).

Therefore we developed a low fidelity prototype based on these guidelines (See Figure 3). The earlier prototype has more functionality than our final product because our original idea was to develop an improved desktop program for our target users. Thus the initial prototype consists of more functionality, it includes:

i. Uploading video and CSV files for one trial of the experiment.
ii. Annotating video automatically by using the timestamps in the uploaded CSV file.
iii. Plotting all of the data from CSV file to produce a general overview of the whole data set.
iv. Calculating velocity to determine extreme changes in speed and direction of movement.
v. Illustrating different colors based on the calculated velocity for plotted points.



(Figure 3)

Some of these design ideas or functionalities are based on the existing programs – functions that our target users consider essential for performing their tasks such as uploading the necessary files to the program – so it is critical for us to include them in our design. The automatic annotation design idea is to minimize users' work load, performance time, and human error. Plotting all of the data and displaying velocity in different color would not only improve the users' performance as well, it also provides users a completely new angle to visualize the data itself and its relationship with one other, from general to specific.

Given the functionality of our initial prototype, we needed to gather some samples of Kinect data to work with so that we can start implementing and testing these functions. However, we encountered a huge limitation for our design after collecting a few CSV files from a Gesture Researchers, Dan Lenzen. Both Dan and COGS102C group members informed us that we will not be receiving additional sample data because of confidentiality agreement between the Gesture Researchers and the participants. With only two short weeks left, we met with Dan and COGS102C group to discuss the possible direction and future of this project.

After meeting with Dan Lenzen, he expressed that there is need for better data visualization when analyzing the Kinect data. Therefore, we have decided to focus only on the data visualization itself and the application will be considered an add-on to the existing programs. Our final design will include the following functions:

i. Creating user login-in account.
ii. Uploading CSV files only.
iii. 3-Dimensional data visualization with direct manipulation of view angles and zoom level.

We felt that it is necessary for users to create an account before using our application because of the confidentiality issues that Gesture Researchers have to deal with. The user login feature will ensure users that all uploaded information is private. Since we only have a few CSV files to work with and our visualization is only designed for CSV data, we have put a constraint on the uploading feature that only CSV file format will be accepted. This way we can ensure that users will not make a mistake by uploading a wrong type of file and cause the application to fail. The 3-Dimensional data visualization provides users the ability of direct manipulation. Users will be able to view the visualization in any different angle and will be able to zoom in and out of the visualization as needed.

4. SYSTEM DEVELOPMENT

4.1 Architecture

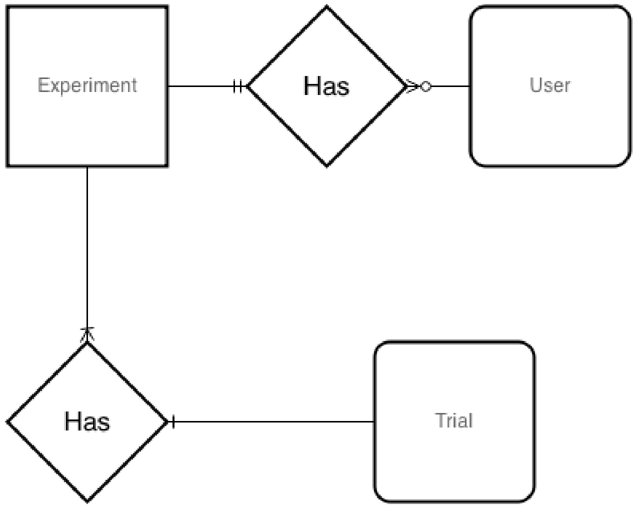
The system uses Node.js as the main back-end Framework on its architecture. It provides a better confidentiality that the code will not be shown up to the user, the server files will not be easy to discover, and the code will be organized between its different parts (routes, libraries,views, public files and models). The framework also provides the easy-to-use include of other libraries within itself; we just had to change two parts: one for including the library on the modules we are using, and the other to permit the specific route to use that library (the require). The libraries used on the project are: mongoose, mongodb, dotenv, and express.

We also used Node.js because most of the members of the group knew more about the framework than the other tools available;we all took the COGS 120 (Introduction to Human-Computer Interaction Design) class on the Winter quarter: that class show to us how Node.js could be a better tool to apply the HCI principles, and make an organized architecture of the system with visual separation between the back-end and the front-end.

The Kinect data is provided as a single CSV file (a file with fields separated by commas) for every person that is subject of the experiment, the architecture of the software define as the normal functioning of the software the importation of the file using the graphical interface page provided for it. So we don’t interact with kinect itself, but with the data kinect provide for us on saved files.

The database was handled using the mongodb as the DMS (Database Management System) , and mongoose as the tool to interact easily with that DMS, without having to create all the configuration of the database system. The database provided for the software was the following one:

(see next page)

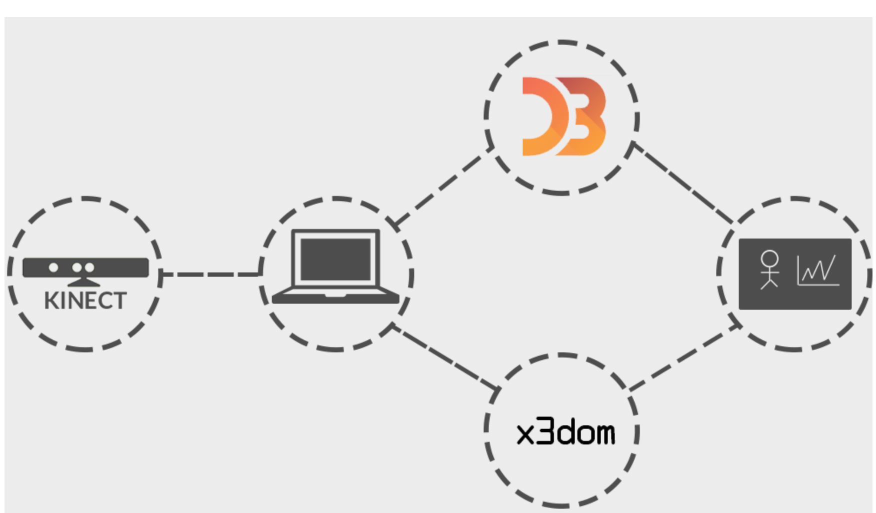


Where the Experiment is defined by a separation of what the gesture researchers are focusing on a specific research, and the trial is every single subject setted to that experiment, the trial has the saved CSV file from kinect as one of its parameters. The user control is provided to define which user created a trial or an experiment, and also keep a log of the system, preventing an unauthorized user to change the data from some trial or experiment (they can change, but the log show which user changed it).

The first idea about how the architecture of the system would be included the Youtube API as one of the tools, but the confidentiality problems with the data provided left us without any option to include the video on our system, since we couldn't just put on our server because it would be a huge flow of data on a server that is not allowed to have too much data, or even too much traffic of data (Heroku on the free version doesn't allow the users to make applications that need a huge flow of data). Therefore, the video data were not used on our system, leaving the visualization of the data our main topic for the system.

For the visualization we used D3.js library integrated with X3dom. X3dom allows us to implement easy 3D manipulations while D3.js allows us to create graphs and visualize the actual data. After the user uploads the data, it would be saved to MongoDB. The user can then go to the visualization page and view their uploaded files. When they click on the trial they want, an AJAX request is sent to the server and gets that file from the database. D3 receives the CSV files and parses it for the data. This data is then filtered and smoothed using our smoothing algorithm. Once the data is done being processed, D3 creates a graph using scalable vector graphics (SVG) within an X3Dom canvas. The graph is rendered and the user sees the visualization. They can then rotate it and view the data to their liking.

The implemented architecture of the system is the one below:



The information flow of the system is defined using a straightforward way to do it:

i. The user creates an account for himself on the system, putting the name, username, and password.
ii. He then log in on the system, providing the username and password he registered.
iii. He can see two options:
    a) Create an experiment or trial of an experiment already created, and import the CSV data.
    b) View an experiment or trial of an experiment already with data imported, using the visualization the D3.js provides for him.
iv. The user logout of the system.

The flow was implemented that way to facilitate the understanding of the system itself, and also to not provide any distraction from the actual goal the user might have. That way the system would be easy to use, and would provide the fast way to achieve user's goals and expectations. The user-experience was considered on the flow of the system, because the better our system achieve what users want it do to, the better is the user-experience related to the system, and more happy the users are with its functions.

4.2 Technology Used

For the API's and Technologies we used, there is not too much to say, as said above, we were intending to use the Youtube API, but that could not been done. We used the kinect data, but imported from CSV files, without direct interaction between the device and our system.

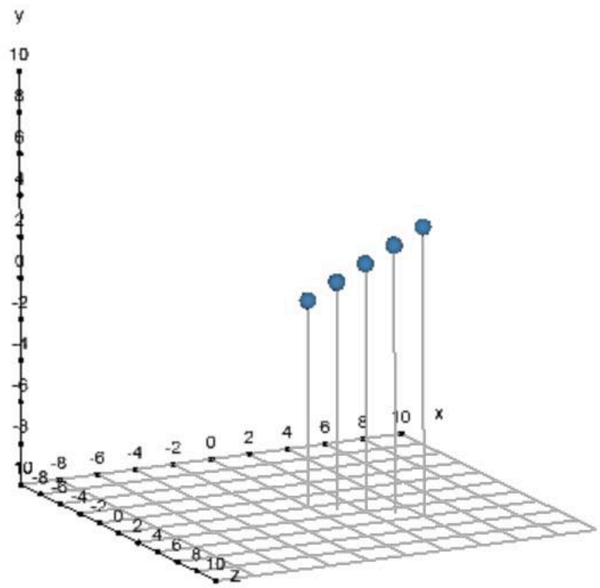
For the authentication, the Passport library were used with mongoose to provide a good authentication method for the system. The two tools integrated provided the right registration, login, and logout without too much of a problem. The Passport-local were used as the method for authentication because we did not need any authentication within any API (Facebook, Twitter, etc), but an local authentication on our system, providing security (the encryption methods were applied to the passwords), and velocity through the authentication process.

The uploaded CSVs and visualization were handled with D3. After receiving the data from an AJAX request, the data was parsed using d3.csv to extract the formatted data and we then manipulated this data to be graphed using polylines and x3dom.

4.3 Features

Due to the nature of our project and what little data we could legally be given to work with, we have few features implemented. Our main features were visualizing data from a Kinect and a way to deal with gaps in large data sets. According to the 102C group, gaps occur when the sensor loses track of the subject and researchers are currently using MATLAB to smooth out these gaps. We attempted to do the same thing but with JavaScript.

The first thing we needed to do was make the data usable by D3. The raw CSV files stores everything-- numbers, dates, and times-- as strings. When the data is received through D3, re-cast them as numbers and returns them as an array of objects for use by the callback function. The callback function counts the number of entries in the response and gets 1% of it. It will then loop through 1% of the entries and average all the values in those objects and save them in a new object which is then stored in an array of averages. This occurs until all the data has been read and averaged. This ensures that there will be no blank data points and successfully smooths gaps. The function then goes through the array and averages and tells d3 what body part is being plotted and where to plot it. The result is a 3d visualization like the one below which is populated with dummy data.



5. SYSTEM DEVELOPMENT

The design of our application is created from a list of human computer interaction principle requirements. To be specific, we took advantage of the Heuristic Evaluation process to implement our design.

5.1 Visibility of System Status

The system should always inform the users what they are doing, it should provide feedback to the users so they know what stage of task they currently performing. Our application provides constant feedback to the use after every interaction. During user's initial interaction with the application, they are notified to login before accessing our application, otherwise if they do not have an account, they will have the opportunity to sign up for one. After signing in, there is only task to do, upload a CSV file. The upload feature will then provide an appropriate feedback to user for the following circumstances:

- If file has not been uploaded, it will say "No file chosen"
- If file has successfully uploaded, "No file chosen" will be replace with the uploaded file name. (FILE.csv)
- If file is in a different format rather than CSV, it will notify the user to upload a CSV file via an alert box.

Finally, the users will always know if they can continue to the visualization.

5.2 Match between System and the Real World

The system should use terminology or language that the users are familiar with, and information should be displayed naturally. Even though our target is Gesture Researchers, we are trying to only use terminology that are familiar to the general public, so beginner Gesture Researchers or even a new Gesture Researcher assistance who had absolutely no knowledge about gesture research can easily get a grasp of how to use our application. The only term that they should know is CSV (Comma-Separated-Values).

5.3 User Control and Freedom

The system should support quick undo or emergency exit if a user shall make a mistake by accident. As mention earlier, if a user made a mistake by choosing the wrong format of file, a alert box will pop up instead of trying to load a new page and display the error message. By having a pop up, the background task page still remain (upload page) and user is able to quickly acknowledge the error message and return to the upload page and try again. If user uploaded a wrong CSV file and went straight to visualization page, the fixed header bar provides 3 simple shortcuts to different stages of task (Login-Upload-Visualization), so users will not have to restart the whole task.

5.4 Consistency and Standards

The system should provide consistent layout, actions, and definition. Our application has the same fixed header throughout the different stages of task and it has the same solid background. Also, by using bootstrap we have a consistent and responsive layout on tablet and mobile.

5.5 Error Prevention

The system should be designed to prevent human error rather than to correct a mistake with a nice shortcut or a good error message. Before moving on to the visualization stage, we have put a constraint on the file type that a user can upload. This constraint is to prevent user moving on to visualization with the wrong type of file and cause the application to fail. The application will only take CSV format file and if other type of file is uploaded it will notify the user immediately, so it prevents the user from making errors and our application will not fail to execute.

5.6 Recognition rather than Recall

The system should minimize user's memory work-load by making the elements and actions more visible. The contents of the application are very easy to see and the fixed header with shortcuts makes the actions of the task visible and easy to use.

5.7 Flexibility and Efficiency of Use

The system should provide accelerators that help users speed up their interaction with the system for both inexperienced and experienced users. The fixed header with shortcut that is on every page speeds up the user's performance. In addition, the ability to directly manipulate the 3-Dimensional graph provides the flexibility of data manipulation, so users may efficiently analyze their data.

5.8 Aesthetic and Minimalist Design

The system should not contain unnecessary and irrelevant content and information that will compromised the visibility of the application. Everything on each page of our application is very simple and easy to see; it has a solid background with high contrast contents. It has a very simple layout comparing to the complicated software user interface on the existing programs. Beside these visibility designs, the visualization stage contains many collapsible panels that will keep the layout content to the minimum. Since there are three stages of within the task, we have separated each stage into three different pages to show one feature at a time.

5.9 Help Users Recognize, Diagnose, and Recover from Errors

The system should provide users with simple error messages when making mistakes and the error message should provide direction to recover from mistakes. When user make a mistake from logging in, it will notify them to login again; when user make a mistake from uploading a file, it will notify them to upload a correct format type of file.

5.10 Help and Documentation

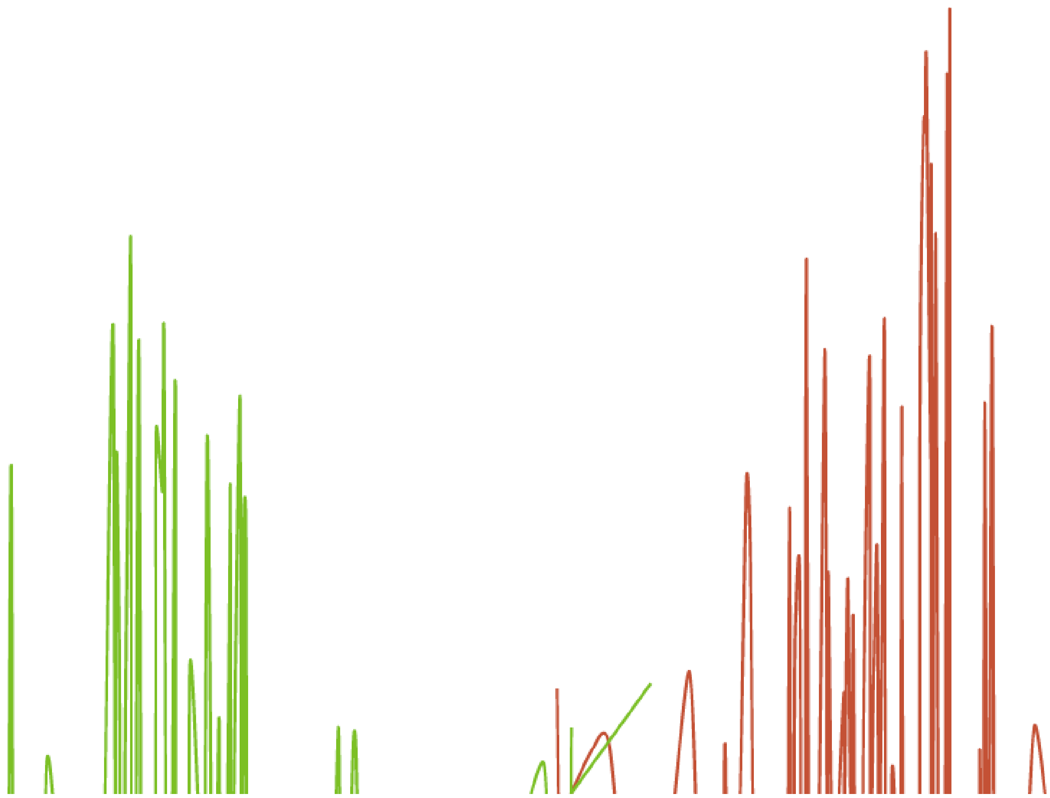
The system should provide help instruction to guide the user through the application if necessary. The login page and visualization page are very self-explanatory, but we included an instruction text for the uploading task. “Must be Kinect Data CSV”.

The reason why we use the Heuristic Evaluation as a guideline is because the listed requirements of the Heuristic Evaluation tackle all of the basic and common usability issues, it helps developers to debug their products and improve the usability of the product and the user experience. We believe that our application complies with the Heuristic Evaluation requirement and we believe the Heuristic Evaluation have improved the usability of our application significantly.

6. TESTING AND EVALUATION

Due to the lack of sample data to visualize, we were not really able to test as much as we would have liked. As a result were only able to test two different data sets: a filtered CSV and an unfiltered CSV. The filtered CSV only contained timed and wrist data and only x and y coordinates resulting in a 2D visualization. When the result was graphed, the gaps in the data became apparent as seen in the photo below. Since we were going to focus more on 3D visualizations, we decided to implement the smoothing algorithm only for the 3D visualization. When the algorithm was finished being written, it was discovered that there were a lot of bugs and that the data was not always being averaged correctly. Whether this will happen to other data or not is something we are unsure of.

(2D representation of wrist data with gaps)



In the end, the algorithm may or may not be necessary for this iteration since data is not plotted in relation to time. This may change before the final release on the day of our groups presentation.

7. COLLABORATION

7.1 Group Collaboration and Single Contribution

All of the group members had the same level of knowledge on the best field of each one; there was no member that would not help enough because of his level of expertise. Also, all the group members worked on the definition of the process of development, division of work, definition about the presentations and the next steps, throughout the 102C meetings, and with the conversion of the needings of the 102C group into feasible features of our system.

Starting with the capacities of each group member: Cristiano has more experience with the back-end development using Node.js, and a little experience with HCI and D3.js provided by COGS120 and COGS121; Joseph has experience with both Cognitive Science and Computer Science area, with the programming knowledge, but also the HCI, and a few of Gesture research knowledge; and Zhen has the experience with Cognitive Science, using D3.js, and about the main topics of Cognitive Science related with the research of the Cognition of the human brain. The division of work were thought using the actual knowledge of each member of the group, and it was divided on the following way:

- Cristiano: Took care of the back-end development related with Node.js implementation; Implemented the authentication methods; Implemented the database and the routes related with the interaction with the database; Went to meetings with Dan, and communicated the information from the meetings with him to the other elements of the group.
- Joseph: Took care of the front-end development, using Bootstrap tools for that; Took care about the visualization using D3.js with Zhen, and tried to make the 3D visualization to use different quadrants to show different graphs of the gesture data provided; Helped Cristiano with the back-end development, creating a boilerplate of the application using Node.js and Mustache; Went to the 102C meetings, and collected useful information to make better visualizations.
- Zhen: Made the 3D visualization; used his Photoshop skills to make the group presentation of each week; Used D3.js to make a prototype of the 3D graph using different quadrants for different visualizations (that part with collaboration of Joseph); Used D3.js to make visualizations to show using the provided Kinect data.

All of the elements of the group worked properly on their duties although our results did not get a huge system working on, but a prototype of the designed system. The D3.js tools using the 3D graph were too hard to make, and the results were good, but not sufficient to make an actual application out of it. Therefore, yet the effort of our group to make an actual application were excellent - and the collaboration were good between its members - , the group could not achieve the goal we tried to.

7.2 102C and 121 Collaboration

The collaboration between the 121 and 102C groups started on a very small time after the 121 group attribution, we already did the Facebook group to interact with each other on the first week designed for the project, and the 102C members were included on that group a few days later, after we get the email from the elements of the group from the Jim Hollan class.

On the second project week we started going on their meeting, after an initial talk on the Facebook group about the project definition and its actual goals - it was really hard to define their goals with the document provided from 102C about the project - , and they started to talk about the project and give us a clearance about our function on the development of the project, not just about the programming itself, but “What to code?”. After the initial meeting our groups were centered on the actual design of the Actviz system.

On that first meeting, the 102C group gave us a broad idea about what they wanted to do, but it was not feasible for our project on COGS121, so we had to start making some restrictions about what would be doable within four weeks we had left for the project. We came with a more clear idea; however, not the final one because we needed to define which visualizations we should do. Also, on that meeting, they told us that the Kinect data were extracted using another system, and it would provide us the CSV file of the data; that was a problem we were discussing because, on the beginning of the project, none of us from COGS121 knew how the data were extracted and if we needed to make a tool for that function or not.

To try to define the visualizations that we might do, and to see the patterns and organization of the Kinect data extracted, we talked with Dan Lenzen, that is the researcher of gesture cognition that the 102C group told us they were talking to. He explained how the CSV data was organized, and told that almost any visualization would be helpful to the gesture researchers. That gave our group a tough spot, because if all visualizations would be helpful, we do not have any idea of an actual visualization to make. None of us had too much knowledge about gesture research to decide what would be good to do.

However, the COGS102C group helped us a little with that, providing some of the ideas they had using the interviews they made with some gesture researchers on the academic community. We started from that, and had a good idea about what we could really implement as the visualizations from our system.

One of the problems that the COGS102C group brought to us was related with our idea about using Youtube API to upload the kinect video data, instead of storing on our server or database; They said that due to confidentiality issues that would not be possible, and the researchers would not allow it. We decided then to not use the video data on our actual system, because of lack of other options to store the video data, since our server does not has the sufficient space to store the videos.

As deadlines got closer, we had to find one key feature to implement that would be the most useful to gesture researchers. This is when we decided that a single, filterable 3D visualization would be the best option with the limited time and the learning curve required to implement them. By the time our visualization demo was complete, the 102C group had already finished some high-fidelity wireframes for us to use as reference for what gesture researchers might expect.

Both the 121 and 102C groups thought that this particular project was difficult and challenging but are proud of trying something new in such a unique field.

8. CONCLUSION AND FUTURE WORK

Overall, this project has been a rewarding experience. There are little no gesture data visualization currently in existence that are web based. Group 2 has been able to join a very unique and groundbreaking project that has a lot of potential to become a staple application for every gesture researcher. That being said, this application still has a long way to go.

With gesture driven devices such as the Kinect, LEAP motion, the Wii, and even cellular devices, more support for different types of data will be a must. Furthermore, the data we worked with was very limited so we do not know how future data will act when inputted. It also must be formatted with the correct CSV header or else it will not be read. This should be changed so that any file can be accepted and the system automatically adjusts to that file.

Although we assume that a 3D representation of the data might be helpful to researchers, we are unsure if our particular visualization will be helpful or not. When you only visualize data in 2 dimensions, you miss an entire dimension of motion that could reveal a lot about subject behavior. This is why we opted for a 3D representation. However, this a more special representation of data instead of temporal. We plot points without any relation to time. We do not know if this sacrifice in time data will affect research or not and should probably implement some sort of time annotation or some equivalent in future implementations.

There are other elements we hoped to add to Activity Viz. One problem mentioned by gesture researchers is having to use a separate program to annotate video data. We hope to implement our own video upload and annotation system without the help of Youtube to protect subject confidentiality. This will allow the researcher to upload, annotate, visualize, and compare data all within the same tool. We also thought that creating a 3D plot that animates over time may also be helpful. We also believe that 2D visualizations should not be eliminated because they can still be useful tools and can help simplify the viewer experience. For example, the user may not always want to look at a 3D scatterplot but instead how the subject is moving from left to right. This would be more easily visualized with a 2D plot.

The 2D plot could either be a standalone visualization or maybe a filtering option derived from the 3D visualization. This is one of many filtering options that we have thought of implementing. A researcher may want to filter the data by showing where velocity changes rapidly, at a certain time, a certain body part or intersections in data points. They may also want to customize how it is visualized such as plot color, line width, or being able to zoom in and out.

These are just some ideas that we thought of as we were completing Activity Viz. It is important to note that none of the members of Group 2 are not gesture researchers so what we may believe to be helpful may actually be trivial or actually not helpful at all. Further collaboration would be needed to move on with development with testing done along the way. Hopefully, this becomes a tool that researchers everywhere can use.

Due to confidentiality issues, we were very limited in the data we had to work with. As a result, we could not implement an extensive, complete application for this project. 3D visualizations are also relatively new and there is hardly any documentation on them at all. The learning curve turned out to be steeper than we thought. That on top of receiving last minute notification that we need to smooth data using our own algorithm, we were severely limited in what we were able to implement and accomplish. However, I feel that our trials and errors will be helpful for implementing a better system in the future.