HERMES - Particle Physics Group Project

JOOYOUNG WHANG, FADI DURAH, and CHAORAN WANG, Virginia Polytechnic Institute and State University

The HERMES data was collected from an electron-hydrogen collision experiment at DESY in Hamburg. After the statistical analysis of the data, a semantic interaction approach was proposed. Upon a successful round of attempts, more exploration of the data with semantic interaction was desired. However, there were too many options between particle, TMD factorization factor, and interest variable choices. Therefore, we come up with a Matrix View tool that allows quick and easy exploration of different options.

CCS Concepts: • **Information systems** → *Information extraction.*

Additional Key Words and Phrases: datasets, information visualization, particle physics

1 INTRODUCTION

The HERMES experiment was done in 2002-2005 at DESY in Hamburg, Germany and has been researched since. However, the work done only used static statistical plots such as scatter plots and histograms. To expand this, collaboration with Computer Science and Information Visualization involving semantic interaction was proposed. Virginia Tech and Jefferson Lab have been collaborating on this topic for about half a year, and their work has found potential in the approach. Until now, only one choice of the many interest variables in the data has been explored. In this project, we attempt to extend the prior work done with the HERMES data and semantic interaction by providing more options to explore.

2 BACKGROUND AND RELATED WORK

2.1 Background

The HERMES data is a multi-dimensional data that resulted from an energy scattering experiment done at DESY in Hamburg, Germany where highly-energized electron beams were collided with hydrogen gas. In this experiment, particles such as Pions and Kaons emerge due to the collision. Each data point contains multiple variables that have one of two kinds of information about the experiment. One type is metadata, such as charge values and particle weights that let the investigator filter different particles from one another. The other is physical data, which can be used to explain the energy interaction that occurred during the experiment. The HERMES data is rather well-understood and has been studied since 2005.

To further investigate the data, a Computer Science approach was proposed. Graphically-Linked Ensemble Explorer (GLEE) is a multi-dimensional data exploratory tool that supports semantic interaction with visualized thumbnails that represent ensembles in the data. GLEE performs Multi-dimensional Scaling (MDS) on multi-dimensional data to represent it on a 2D surface while preserving the relative distances between ensembles in regard to their attributes. It is also capable of performing Inverse Multidimensional Scaling (IMDS). The user can define a cluster from visual features that he or she notices in the thumbnails, and the system assigns a set of weights to each of the attributes, ordered by the impact an attribute had on the creation of the defined clustering. These semantic level interactions allow the users to easily explore the data domain and discover interesting features.

The creation of the visualized thumbnails was done using Paraview 5.6, a scientific data visualization and analysis tool. The tool works by processing input data along multiple filter pipelines to

Authors' address: Jooyoung Whang, joo918@vt.edu; Fadi Durah, fadid6@vt.edu; Chaoran Wang, showcy@vt.edu, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 24061.

produce a final visualization at the end of the pipeline. Paraview provides many predefined filters such as threshold, calculator, and table-to-points. The user can also use the python-programmable-filter to manually process the input at a pipeline stage.

2.2 Related Work

Munzner's textbook introduces an effective iteration cycle for creating a good visualization [1]. She writes about the importance of solidifying WHAT, HOW, and WHY for producing a visualization that can satisfy the need of a client. For our project, we took an interview with our client and identified each of WHAT, HOW, and WHY. We identified that our data type is a table with 20 columns and 2 million rows. All the data are quantitative attributes (*WHAT*). To represent our data, we filtered the raw data by several criteria such as particle types or an attribute's threshold ranges. We used a contour map of a 2D histogram to represent our data (*HOW*). Our client wanted a tool that could assist in identifying a feature hidden in the data (*WHY*).

Shneiderman discusses 3 visual information-seeking mantras, two of which are overview at first and details on demand [3]. We applied Shneiderman's advice in our design when we were considering the dashboard for our visualization. During our dashboard design step, we found that there were too many visualizations to explore all in GLEE. Therefore, we built the Matrix View to allow our client to explore an overview of the entire data first before coming back to GLEE for inspecting the details.

Theresa-Marie provides many good advice for maximizing the clarity and readability of a visualization using color theory [2]. We applied Theresa-Marie's insights when we designed our thumbnails for the data. Our data points in the visualizations were shown using a white-to-red color scheme. To increase the readability, we made the background and axes labels colored in blue, which is a complementary color to red. To let our client easily read the axes labels, we made the axes labels colored in a higher value than the background. This combination of color contrast increased the effectiveness of our thumbnails. We also made sure that all of our visualizations were color-blind friendly, a point that was also made by Theresa-Marie.

3 DESIGN

3.1 Interview and Requirements

3.1.1 *Interview.* A short interview was conducted prior to the design phase with our client at Jefferson Lab. During the interview, we were able to pinpoint our client's interests and come up with the project's requirements.

Basic statistical analysis was already done by our client with the HERMES data for each of the 5 particle types that were created during the experiment. The following is a list of particles that emerged from the experiment.

- Positively Charged Pion
- Positively Charged Kaon
- Negatively Charged Pion
- Negatively Charged Kaon
- Uncharged Pion

Of these, the first 4 particles were of interest to our clients. For each of the particles, our clients were interested in looking at 6 physical data variables listed below.

- x
- y
- z
- $P_{h\perp}$

- q^2
- ϵ

In addition, a special property called *QZP*, a quantity defined with the following formula

$$QZP = \frac{q^2}{\frac{P_{h\perp}^2}{z^2}} = \frac{q^2 z^2}{P_{h\perp}^2}$$

was of special interest. *QZP* is the TMD factorization threshold for the data. Our client was interested in observing the data for the following *QZP* states.

- $QZP >> 1$
- $QZP << 1$

In our client's prior work, 2 thumbnails from each of the following *QZP* ranges were sampled (6 thumbnails in total) and put in GLEE.

- $QZP \geq 1.5$
- $1.5 > QZP \geq 0.5$
- $0.5 > QZP$

Each of the thumbnails was a 2D histogram contour plot between z and $P_{h\perp}$. In their work using GLEE, they found that sampling from an ordered range of *QZP* values resulted in a single clustering. The contributing attributes to this clustering were x, q^2, and $P_{h\perp}$, which are the three variables that are used to calculate *QZP*. No interesting results were found, but it proved that semantic interaction could be a promising approach to inspecting the data.

Therefore, for our project, the clients wanted thumbnail sets using different variable combinations that could create a more interesting clustering while sampling from only one of the *QZP* ranges at a time, instead of going across all the ranges.

3.1.2 Requirements. From the conducted interview and multiple design iterations, we were able to come up with the following design requirements:

- Create sets of visualized thumbnails of the HERMES data that can be used in GLEE.
- Each of the thumbnails should effectively show a distribution of data points between two variables as axes.
- The data should be filtered so that a set of thumbnails always show a choice of one particle type and one *QZP* range.
- The variables should be chosen from: x, y, z, $P_{h\perp}$, q^2, ϵ
- The visualization should not imply false features in the data.
- For each thumbnail in the same set, properties such as scale and contour thresholds should be consistent.
- The scale of a variable should be consistent across different sets of thumbnails.

The approach we took to satisfy our requirements is described in the below section.

3.2 Approach

Our initial approach to the problem was to generate all possible combinations of each of the particles, *QZP* ranges, and choice of two variables for the 2D histogram contour plot. However, the number of possible configurations add up to

$$4 * 2 * \binom{6}{2} = 120$$

which is too many to investigate individually in GLEE. There was also the possibility that many of them may not contain any interesting features. Therefore, we decided to devise a thumbnail explorer that would allow our client to quickly and easily traverse through the various configurations of particle portrait thumbnails before investigating further in GLEE. For this purpose, we chose to create an HTML visualization tool called the Matrix View that allows the client to quickly cycle through the different variable, particle, and QZP combinations. This tool will be described more thoroughly in 4.2.

To represent the thumbnails to use in our Matrix View, we decided to keep the 2D histogram contour plot representation that was used in the previous study. This representation was not only already familiar with the client, but it was also a good tool for inspecting the population of data points between a choice of two variables as axes for the plot. The design was optimized from previous work to help our client understand the data. Instead of improving the visuals, most of the consideration was put into automation of the export sequence for the thumbnails. For the thumbnails to be used in GLEE, there must be a set of thumbnails that each come from the same data but have different visual features that would lead into an interesting clustering. We decided to export 10 thumbnails per configuration; that is, 1200 thumbnails in total. A more detailed description will be presented in 4.1.

4 IMPLEMENTATION

4.1 2D Histogram Contour Plot

4.1.1 Overview. Figure 1 is a sample of a thumbnail that was used in the previous study by the client. First, a 2D histogram is created using a total of 10000 bins across the 2D space. Then, a contour plot is created with the histogram, using 5 contour levels and a white-to-red color scheme. To provide clear readability, a complementary color is used for the axis grid lines and the background box. The color of the grid lines has a higher value than that of the background box to further improve readability.

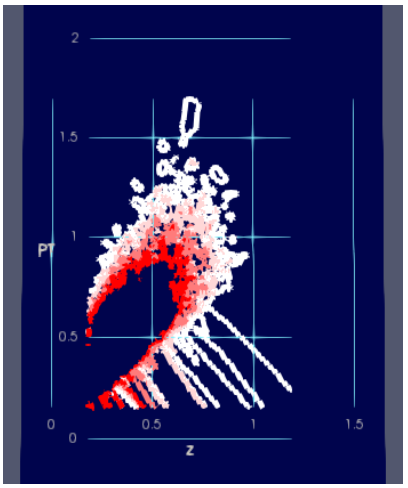

Fig. 1. thumbnail from prior work

Using the same Python script from earlier work, we produced a thumbnail with a different choice of two variables, as can be seen in figure 2.

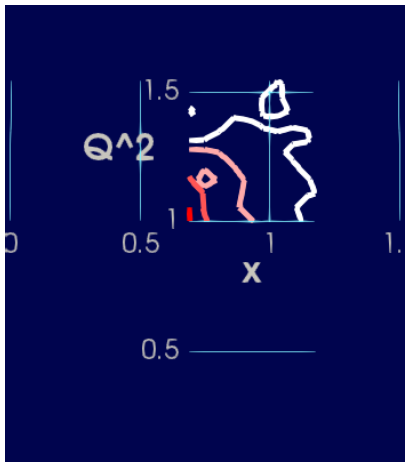

Fig. 2. Exported thumbnail with a different choice of variables

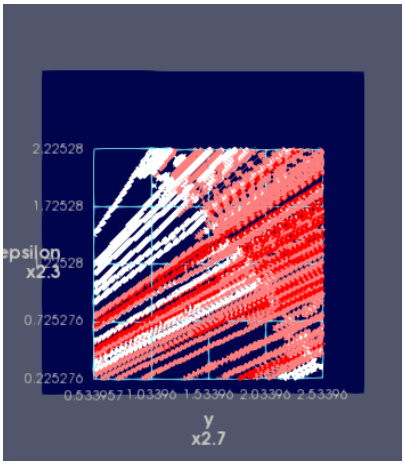

Fig. 3. End result thumbnail for Pi+, QZP ≥ 1.5, y vs epsilon

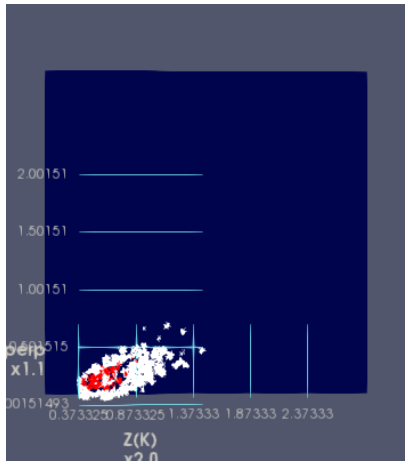

Fig. 4. End result thumbnail for K+, QZP < 0.5, z vs $P_{h\perp}$

We identified two major flaws. First, the script uses a custom range for the axes that fit the original choice of variables. Therefore, the image was not captured with the correct scale in our exported thumbnail.

Second, the contour level thresholds were also configured for the original choice of variables which didn't align with the other choice of variables. Some of our exported thumbnails only had white contour regions that didn't show any interesting features.

Therefore, the script was modified to automatically produce and use reasonable axes ranges and contour levels for each configuration. The following figures 3 and 4 show the end results. The following section explains the detailed implementation.

4.1.2 Pipeline. In Paraview, the following pipeline was used to generate the thumbnails for a choice of particle type and a QZP range.:

(1) Configure the camera view and a background box for color contrast
(2) Get a pre-filtered data that only contains the chosen particle type.
(3) Filter the data to only contain the chosen QZP range.
(4) Choose two variables from: x, y, z, $P_{h\perp}$, q^2, ϵ
(5) Take a tenth of the data randomly that haven't been selected before.
(6) Create a 2D histogram.
(7) Create a contour plot from the 2D histogram.
(8) Scale the axes to fit the camera view and save the image.
(9) Return to step 5 until all data have been selected and processed.
(10) Return to step 4 to repeat the process until all combinations of variables are processed.

Aside from the existing filter that creates a contour plot, the other pipeline stages had to be manually programmed in Python 2.7.15. Statistical data such as the mean and standard deviation is also collected during the process to be used in GLEE if desired in the future.

The extremum of the selected variables and the calculated bin values of the histogram are all recorded in the process to generate a reasonable scale and contour thresholds. For each of the axes, the minimum and the maximum value are scaled to fit the camera view. The scaling factor n is written below the axes label in the notation: xn. The maximum and minimum bin values are linearly interpolated in 5 steps to be used as the contour thresholds.

A flaw with Paraview that we noticed while implementing scaling is that the tool does not support scaling the axes length without also changing the label value. For example, in figure 3's x-axis, the maximum range value is shown as 2.53396. However, this is not the true value. To retrieve the true value, the shown value must be divided by the scaling factor 2.7, which is 0.938503704.

4.2 Matrix View

4.2.1 Overview. As stated in 3.2, the Matrix View is an exploratory tool that our client can use to quickly and effectively compare the thumbnail distributions of the many different combinations in the data. The Matrix View is shown in figure 5.

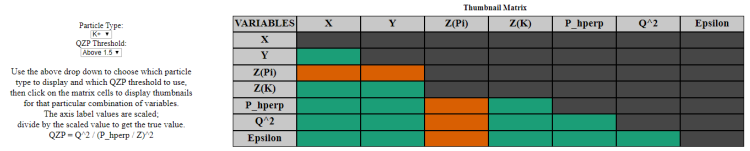

Fig. 5. Matrix View HTML visualization tool

The tool was kept relatively simple and easy to use to facilitate the thumbnail comparison process. Below is an explanation of the tool's capabilities and design choices.

4.2.2 Functionality. As seen in figure 5, the names of each of the interest variables are shown in the light gray cells in the topmost row and in leftmost column. The interest variables are then used in the corresponding row or column to plot the variable combination and display the correct thumbnails.

Inside of the matrix, the green-jade colored cells represent an available combination between two interest variables, and the orange-red colored cells represent a combination that is currently unavailable. This distinction is important since the type of particle chosen will limit which z interest variable is available for plotting. For example, if the chosen particle type is $K+$, all cells involving $z(Pi)$ will become unavailable, since we do not have any data for $z(Pi)$ that can be mapped for the particle type $K+$. Similarly, if the particle type chosen was $Pi+$ or $Pi-$, then the cells involving

$z(Pi)$ would become green (available) while $z(K)$ becomes unavailable. Note that the dark gray cells inside the matrix are "dead" because they represent a combination that either is already covered by another cell (green cell) or is unnecessary (a combination of two of the same variable such as (y, y)), therefore they are not interactable.

Aside from choosing the interest variable combination via the available matrix cells, the client also has the ability to choose the Particle Type and QZP Threshold used to filter through the data set by using the drop-down lists to the left of the Matrix. Selecting options from these drop-down menus will result in different thumbnails being displayed. The Particle Type and QZP Threshold Lists have the options $K+$, $K-$, $Pi+$, $Pi-$ and Above 1.5, Below 1.5, respectively.

Using the mouse and hovering over any available (green) cells, a yellow highlight will cover the cell to indicate the selection, and clicking on that available cell will cause the 10 corresponding thumbnails to be displayed below the matrix (according to interest variable and drop-down menu selections). Consider the figure below (Figure 6):

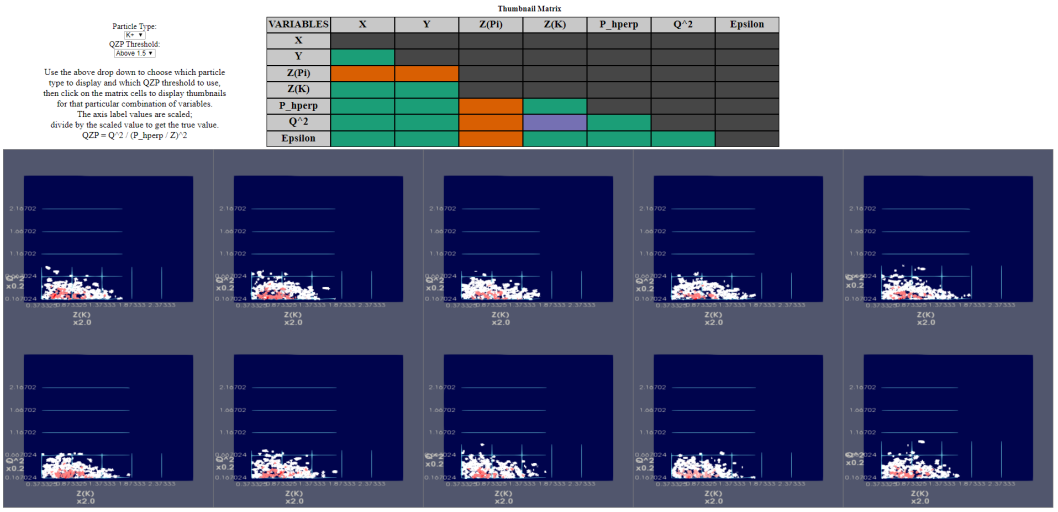

Fig. 6. Once an available cell is selected, its thumbnails are displayed

As shown, the available cell chosen is now "highlighted" (gains a purple color and becomes unclickable) and its thumbnails are shown below the matrix. In this figure, the thumbnails being shown are for the variable combination ($z(K)$, q^2) as evident by the cell's position ($z(K)$ column and q^2 row) along with the axis shown on the thumbnails. From here, the client would compare the different thumbnails and decide whether or not this particular combination (for this particle type and this QZP threshold) has any interesting features that could be significant enough to study more closely and import into more dedicated software for examination.

4.2.3 Accessibility. We have done multiple milestone evaluation at various points during development to get feedback on our design and point out things that we can do to improve our client's satisfaction. For the Matrix View itself, it's relevant to note one previous, major design that we had during our project. It is shown in the below figure (Figure 7):

During one such evaluation, our team made improvements to the above design according to our advisor's feedback. Ignore the abundance of red cells, as this was at a point where we did not have many of the thumbnail combinations ready for display. The followings are the improvements made to the old design.

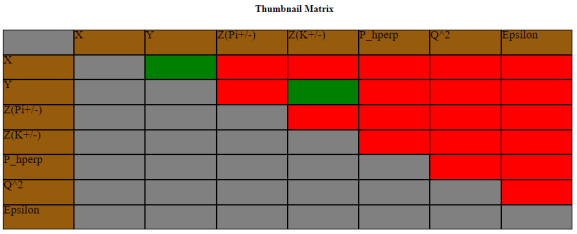

Fig. 7. Early Design for the Matrix View

- Improved Variable Name Readability
  - This included replacing the murky brown color while also increasing font size, using bold font, and centering the names.
- Restructured combination cells to be aligned towards the left of the matrix.
  - This seemed to be more visually pleasing and made more sense considering the typical structure of a matrix.
- New color scheme to support colorblind accessibility along with improving overall design aesthetic.
  - This was done while maintaining the clear red-green color coding for the availability category.
  - The highlight purple color was included to indicate when a cell was being displayed.
  - The color scheme was chosen using an online tool called the Color Brewer, which is a well known tool for choosing a good color scheme for various data types.

5 EVALUATION

Our final design does a great job of showing the correlations between variables and the distributions of the data points given specific filters. The redder the clusters are, the more data points are concentrated. The Matrix View provides easy navigation across many combinations of attributes and particle types. It helped our client view the results and figure out interesting facts about the specific combinations simultaneously. The color-coding of the matrix helped remind the client which combination the view is showing and clearly show what the differences are among plots. The updated thumbnail export script also increased the export efficiency as we previously needed to run different combinations 120 times in total. The new script allowed us to run it only 8 times (for each choice of particle type and QZP range) to get all results.

However, there are still some problems with our design. The biggest problem is that the design does not show the actual quantitative value of the variable because the axes are scaled. We do not consider this a major problem since this visualization is an exploratory design that is focused on recognizing the shape and the correlation between variables instead of retrieving the actual values. Hence, we believe our work meets the client's requirements such that they can easily find the interesting features and variable correlations, which then they could investigate the interesting facts further in other tools such as GLEE.

The most important objective of our project was to satisfy our client, so we kept in touch with our client throughout the process of this project. We received lots of suggestions that helped us improve our design. We also presented our final design demonstration to our client. Our client concluded that this is a very successful visualization and it is helpful for building the information our client needs.

Our client was initially confused why we divided the data randomly into 10 pieces, as our client's advice was to sort by one variable and then cut it into 10 segments. However, we could ultimately agree on the choice because sorting the data along a particular variable will introduce a similar problem our client had in the previous experience with GLEE, where only a single cluster would occur and the contributing attribute is the sorting variable. We agreed that randomly picking the data points would provide a better chance of forming an interesting cluster.

We were also given some suggestions for future work. As we expected, our client was unhappy with the scaling problem. It was very inconvenient for our client to have to retrieve the exact value after going through the process of reading the values and then dividing them by the scaling factor.

Our client also felt that there were too many blank areas in the thumbnails. Our client suggested that it might be helpful if we further settle the scales to decrease the blank areas.

Finally, our client pointed out some minor issues, such as the matrix view not showing the correct notation of the variables (For example, $P_{h\perp}$ written as P_hperp) and the lack of definition for QZP for other physicists that encounter our tool. The property QZP was a temporary term that was defined between Jefferson lab and Virginia Tech. As this was a simple fix, we applied it in our final iteration.

Overall, our client confirmed that our design is exactly what our client was looking for.

6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

We believe our design creates a foundation for our client to start deeper analysis of the HERMES data. There is a good chance that many interesting facts may be found by exploring the data combinations using the Matrix View and later moving over to GLEE for complex observation.

6.2 Future Work

Based on the discussion in section 5, we found some potential future work work that could further improve our design.

First, different contour threshold scales such as log-scale could be applied to allow easier identification of a cluster in regard to contour color. The current design only uses a linear scale, and there is a possibility that it might hide some interesting shapes.

Second, we could fix the axes labels to show the real values instead of showing scaled values. This would greatly shorten the time for our client to retrieve the true value of variables.

Third, we could implement a zoom feature to the Matrix View thumbnails which could help solve the blank space problem while maintaining the current thumbnails.

Finally, we should represent the variables in the correct mathematical notations. This would improve the readability for other physicists.

REFERENCES

[1] Tamara Munzner. 2014. Visualization Analysis and Design. In *A.K. Peters visualization series.*

[2] Theresa-Marie Rhyne. 2012. Applying Color Theory to Digital Media and Visualization. In *ACM SIGGRAPH 2012 Courses (SIGGRAPH '12).* ACM, New York, NY, USA, Article 1, 82 pages. https://doi.org/10.1145/2343483.2343484

[3] Ben Shneiderman. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages (VL '96).* IEEE Computer Society, Washington, DC, USA, 336–. http://dl.acm.org/citation.cfm?id=832277.834354