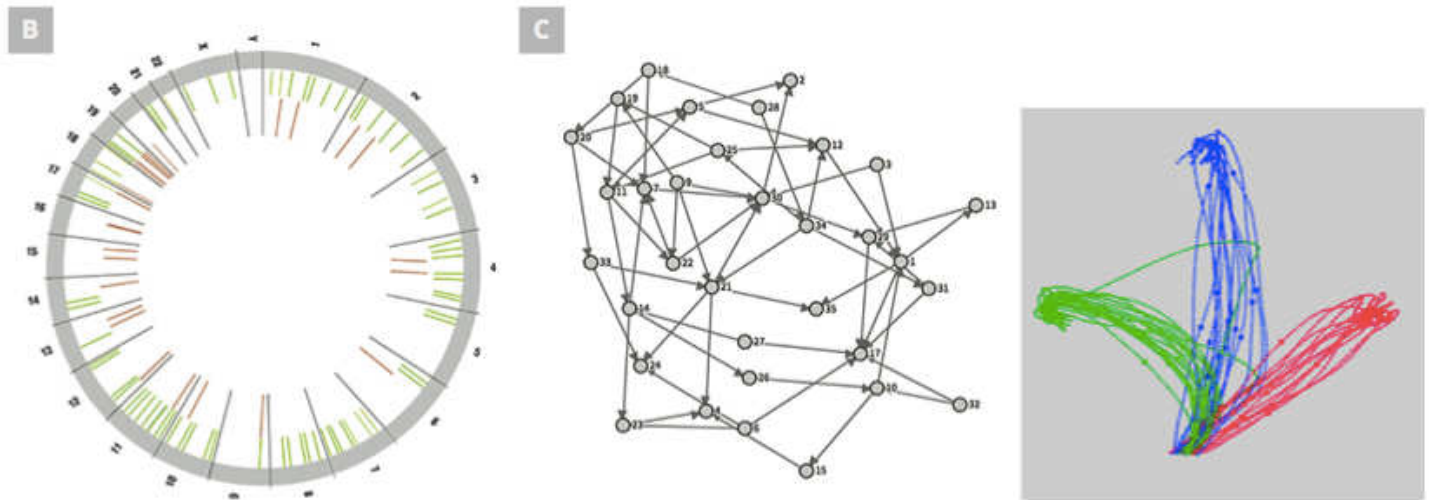


CS 237

Interdisciplinary Scientific Visualization



Final Projects 2012

Abstracts

1. Visualizing Reach and Eye Trajectories

Alicia Boucher and Hobart Reynolds

2. Measuring the Effect of Domain Knowledge on the Learnability of Visualizations

Alexandra Papoutsaki, Connor Gramazio, Hua Guo, and David Laidlaw

3. Multiple Coordinated Views for Visualizing White Matter

Ben Fichter, John Meehan, and Michail Michailidis

4. Measuring the Effect of Domain Knowledge on Visualization Learnability

Connor Gramazio, Alexandra Papoutsaki, Hua Guo, and David Laidlaw

5. Visualizing Reaching Trajectories in 3D

Hobart Reynolds and Alicia Boucher

6. Multiple Coordinated Views for Visualizing White Matter

John Meehan, Michail Michailidis, and Benjamin Fichter

7. BraCoViA: Multiple Coordinated Views for Visualizing White Matter

Michail Michailidis, John Meehan, and Benjamin Fichter

Visualizing Reach and Eye Trajectories

Alicia Boucher

Computer Science, Brown University
Alicia_boucher@brown.edu

Hobart Reynolds

Computer Science, Brown University
hobart_reynolds@brown.edu

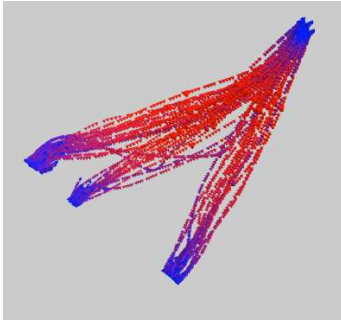


Figure 1: A group of trajectories.

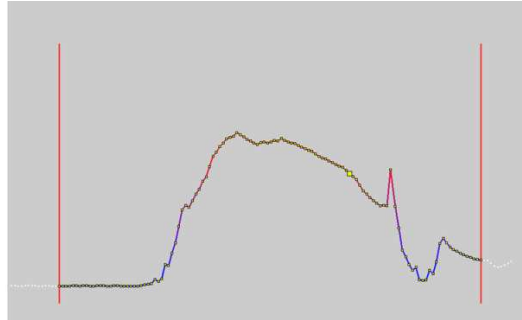


Figure 2: A velocity profile of one trajectory.

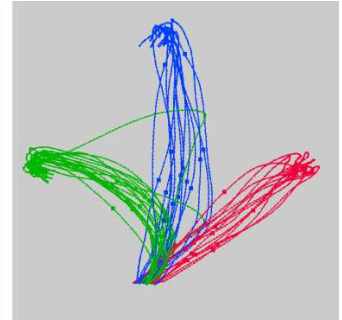


Figure 3: Subsets of trajectories.

ABSTRACT

We created a tool to aid cognitive science researchers in analyzing trajectory data from reach and visual search tasks. Studies in cognitive science suggest that analyzing relationships between eye trajectories and manual trajectories can reveal the cognitive events involved in shifting focus. One of our aims was to provide a feature that could allow analysts to observe and compare eye motion and hand motion collected from the same task. Another one of our aims was to improve upon the visual features provided in a preexisting tool for analysis. By developing this tool we wanted to aid the lab in expanding upon the known model of cognition.

Keywords

Trajectory, cognitive science, velocity profile

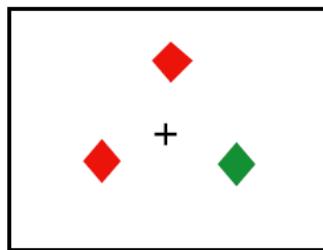


Figure 4: Color oddity task.

1. Background

Reach trajectories reveal the continuous cognitive events involved in decision making. A commonly used task for gathering trajectories is the Color Oddity Task [1-4]. In this task, several objects are projected (Figure 4) and the subject must point to the odd colored object. The trajectory data from the task is a collection of 3d points along which the subject's hand traveled.

Many trials are run for a single subject and the researchers analyze the velocities and curvature of the trajectories. The analysts also pay attention to latent time, the amount of time after which a subject responded by moving his or her hand, and movement time, the time it took from beginning the movement to reach the target.

These metrics reveal how cognitive events occur over the time of trajectory [1,3,4]. An increased curvature indicates that the subject has changed his or her decision about the location of the target [3]. The usefulness of this metric is that it provides evidence that cognitive events can happen in parallel, rather than separately and serially. For instance, decision making about the location of the target can occur while a manual movement to the target has already begun.

Previous studies have analyzed the relationship between eye motions and hand trajectories. Usually eye motions precede and guide hand motions [2,5]. The relationship between eye movement trajectories and associated hand trajectories could reveal how visual search algorithms are executed and how attention works at the level of cognition. Our tool provides a feature so analysts can observe these two kinds of trajectories together.

Implementation

Our program has two modes: Group mode in which multiple trajectories are plotted in the viewport, and Trajectory mode in which a single trajectory is plotted in the viewport. To make metrics immediately available to the user, we included a panel to display global metrics for a group of trajectories and local metrics for a single trajectory. When in Group mode, the global metrics include: average maximum velocity, average maximum curvature, average total time (or movement time), and average initial latency. When in Trajectory mode, the global metrics include the maximum metrics that occur on the trajectory, and the local metrics appear for a selected point on

the trajectory. The user can move the selected point along the trajectory.

In addition to the group averages displayed in Group mode, the program also displays the average metrics for a selected sub-group of metrics. The user can create arbitrary subgroups to analyze and render separately (Figure 3).

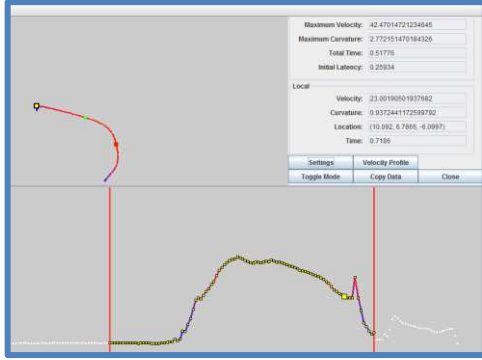


Figure 5: Adjusting the latent period and movement period of a trajectory.

When in Trajectory mode, the program can also display an interactive velocity profile of the chosen trajectory (Figure 2). The user can click on the velocity profile to choose where latent period ends and where the movement ends (Figure 5). Editing the end points of the latent and movement periods is part of the analysts' preprocessing step for calculating metrics over group data.

Various graphical features highlight features in the metrics. The trajectories can be colored by velocity (Figure 1), curvature, or subset (Figure 3). The point of maximum velocity is enlarged on each trajectory, and sliders allow the user to iterate over points in the trajectory or velocity profile. The program allows users to link the sliders on trajectories in different viewports. Our motivation in creating this feature was for analyzing eye and manual trajectory data at once.

2. Evaluation

We sought quantitative feedback about the usefulness of our tool in processing trajectory data. The existing analysis tool is a MATLAB program written and shared by the researchers. We timed one of the researchers in sorting trajectories into groups. We told the researcher to find the average maximum curvature for a specific group of trajectories. The groups were defined as trajectories from trials with: the same target color, the same target location, or the same distractor color. In these tasks, we instructed the researcher not to alter the latent times of the trajectories; we wanted to isolate the time used for grouping data. We found that the researcher performed these tasks faster in the existing MATLAB program (Table 1). Although the task took longer, our tool gave immediate visual feedback about the creation of the group.

Table 1: Finding metrics for specified groups of trajectories. Times are Minutes:Seconds.

	Target color	Target Location	Distractor Color
Existing Program	1:22.03	0:21.94	0:33.79
Our program	2:09.99	1:05.61	1:16.55

We wanted to see how long the researcher took to find the average metrics for a group when he was forced to manually select the latent period and movement period ending times for many trials. We found that the existing tool was also faster for this task (Table 2). However, our tool was superior in allowing the researcher to finely tune his selection of the points with immediate graphical feedback about how the trajectory was affected. We noticed a tendency to perfect the selection points when the researcher used our tool.

Table 2: Finding metrics for specified groups of trajectories when editing thresholds. Times are Minutes:Seconds.

	Existing Program	Our Program
Time	1:46.30	2:59.86

The researchers felt that the rendering methods provided in our tool made it ideal for analyzing two trajectories at once and for viewing data from a single subject. They thought that the program's functions to save images made it useful for generated figures and images for reports and publications.

References

- [1] Song J. -H., Takahashi N., & McPeck R. M. (2008). Target selection for visually-guided reaching in macaque. *Journal of Neurophysiology*, 99, 14-24.
- [2] Song J. -H., & McPeck R. M. (2009). Eye-hand coordination during target selection in a pop-out visual search. *Journal of Neurophysiology*, 102, 2681-2692.
- [3] Song J. -H., & Nakayama K. (2009). Hidden cognitive states revealed in choice reaching tasks. *Trends in Cognitive Sciences*, 13, 360-366.
- [4] Song J. -H., & Nakayama K. (2008). Target selection in visual search as revealed by movement trajectories. *Vision Research*, 48, 853-861..
- [5] Khan, A. Z., Song, J.-H., & McPeck R. M. (2011). The eye dominates in guiding attention during simultaneous eye and hand movements. *Journal of Vision*, 11(1), 1-14.

Measuring the Effect of Domain Knowledge on the Learnability of Visualizations

Alexandra Papoutsaki, Connor Gramazio, Hua Guo, David Laidlaw

{alexpap, connor, huag, dhl} @cs.brown.edu

Department of Computer Science, Brown University

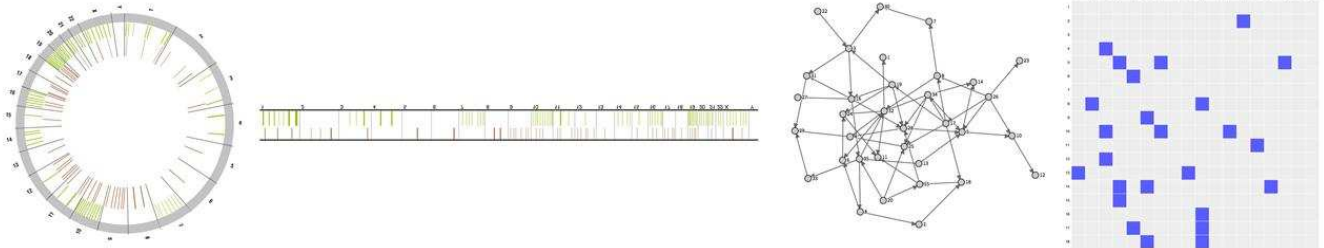


Figure 1: From left to right: Circos, Gremlin for the CompBio,

node link diagram and matrix representation for the Graph Theory

ABSTRACT

We performed a user study to measure the effect of domain knowledge on the learnability of users from two different scientific domains. Learnability is acknowledged in the field of Visualization as an important factor on the usability of visual representations. Even though great effort is being put in the design of robust and useful visualization tools, little research has been conducted in identifying the correlation between the prior knowledge for a domain specific visualization and user's performance. We provide the results that we gathered from our user study, hoping that they will trigger further research on uncovering the mechanisms that underlie learnability.

Keywords: Learnability, visual representations, domain knowledge, user study.

1 INTRODUCTION

The need of further exploring the usability and prior knowledge of the visualization tools has been repeatedly identified in the literature[2]. Even though much work is done in the development of visualization tools, scientists have not explored and discovered yet the association between the prior knowledge on a specific domain representation and the learnability, meaning the ability to learn fast and well a visual representation. Scientists have speculated in the past that users prefer a representation closer to their traditional way of viewing visualizations, often contrary to what Visualization researchers might believe [3],[4]. Motivated by these and other examples we decided to form and explore the following hypothesis: prior knowledge can be a significant factor in the improvement of the learning curve. To investigate this hypothesis we designed a user study for novices and experts in two scientific domains.

2 DESIGN OF EXPERIMENT

We designed two separate user studies for novices and experts from the fields of Computational Biology and Graph Theory. Both of the studies were conducted on an interactive web server that we created. We define an *expert* as a user with domain knowledge and prior exposure and usage of the visual representation. A

novice is defined as a user that might have seen in the past a visualization, but has not used it and does not have any domain knowledge. For both experiments we presented to the user, regardless of his level of expertise and domain knowledge, tasks from taxonomies found in [1] and [7]. In the beginning of both experiments we provided a short description of the experiment and the type of tasks. After making sure that a user was familiar with a visual representation could proceed with the actual tasks.

2.1 Computational Biology Study

We compared two visualization tools that are used by Computational Biologists in visualization and exploration of structural variations (e.g. insertions, deletions, etc.) that occur in the DNA and are usually connected to abnormalities in the phenotype of individuals. The first tool is Gremlin [6] which represents the chromosomes in a linear order, something that most Biologists are familiar with. The second tool is Circos [5] which arranges the chromosomes in a circular layout; Circos has received much attention by the Visualization community over the past years. We asked each user to perform three (3) tasks, and for each visual representation five (5) instances of the above tasks, resulting in 30 questions per user. The tasks they performed were the following:

- **Find** chromosomes with greater than a set number of inversions
- **Find** the chromosome with maximum inversions
- **Sort** chromosomes by the number of deletions

Every time that a user was starting a user study the tasks were presented in a random order. For each task all five instances were then asked, to preserve independent observations.

2.2 Graph Theory Study

Similarly with the Computational Biology study, we chose two different visual representations of the mathematical object of a graph. The first one is the classic node link diagram, while the second one is the more recent matrix representation, which represents an adjacency between two nodes as a filled entry in the matrix. The tasks that we asked our users to perform were the following:

- **Find** a specific path connectivity
- **Find** the node with maximum in-degree
- **Sort** nodes by out degree

At the end of both user studies we provided a short questionnaire asking for qualitative feedback on the preferences and the hurdles our users faced when performed the study.

3 RESULTS

For each user we kept track of the mouse movement, number of trials, actual answer, response time, and questions the participant asks the proctor. After we conducted our user study with 12 people (3 people for each combination) we continued with analyzing our results.

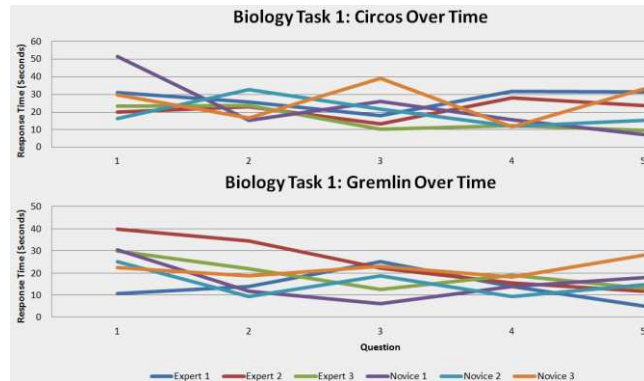


Figure 2: Progression in user performance in a single task

We started by examining the progression of the performance of users as they repeat certain tasks. Figure 2 shows the results for the Computational Biology study; similar results were gathered from the Graph Theory study. Unfortunately we were not able to detect a great improvement in the task completion time over time. We suspect that this could be a design flaw in our experiment, since the order of appearance for the tasks was not the same for each user. We continued by investigating the differences in absolute response times between novices and experts. Figure 3 shows that once more we could not find any significant difference between the two groups. There were surprising cases where expert users were taking a longer time to respond compared to novices. A possible explanation could be that experts prefer to be more thorough when examining a visual representation, in order to make sure that their answer is almost certainly correct.

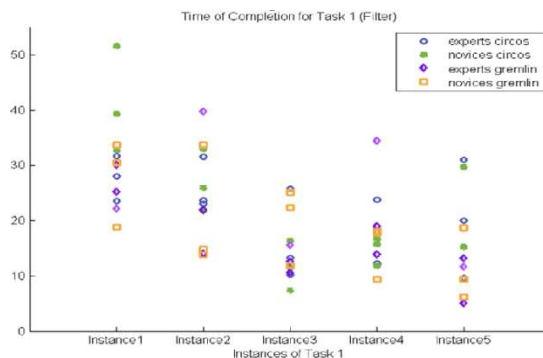


Figure 3: Differences in the time responses of all the groups

In Table 1 we report the results that we gathered from the questionnaire our users answered as the last part of the experiment. It is obvious that there is a strong preference towards the matrix representation, and many reported issues about the complexity and the difficulty of reading the node link diagram.

No definite answer can be given about the Computational Biology study, since the users were torn between Circos and Gremlin.

Preference				Reported Issues	
	Node-link	Matrix	Equal	Node-link	Matrix
experts	0	3	0	experts	3
novices	0	1	2	novices	2
	Gremlin	Circos	Equal	Gremlin	Circos
experts	1	0	2	experts	1
novices	1	1	1	novices	2

Table 1: Analysis of the Questionnaire for both studies

4 CONCLUSION

We performed and report the results of a user study to assess the significance of the effect of domain knowledge on the learnability of visual representations. Even though our results were not strong indicators in favor or against the correlation between domain knowledge and learnability we believe that they can still be used as triggers for further research. A related open question that we find worthy of exploring is quantifying the difference on the learnability and accuracy, e.g. between meticulous and fast and less careful users.

REFERENCES

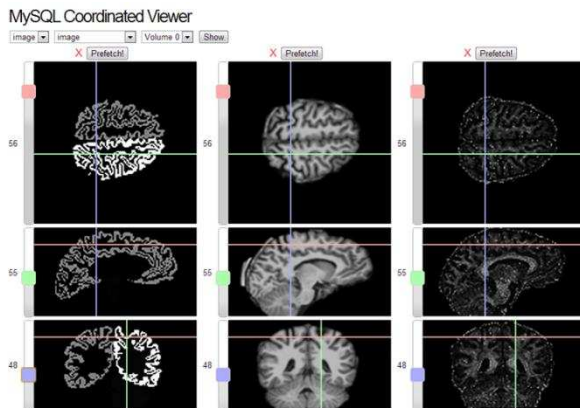
- [1] Robert A. Amar, James Eagan, and John T. Stasko. Low level components of analytic activity in information visualization. *INFOVIS*, page 15, 2005.
- [2] Chaomei Chen. Top 10 unsolved information visualization problems. *IEEE Computer Graphics and Applications*, 25(4):12-16, 2005.
- [3] Radu Jianu, Cagatay Demiralp, and David H. Laidlaw. Exploring brain connectivity with two-dimensional neural maps. *IEEE Trans. Vis. Comput. Graph.*, 18(6):978-987, 2012.
- [4] Radu Jianu, Keping Yu, Lulu Cao, Vinh Nguyen, Arthur R. Salomon, and David H. Laidlaw. Visual Integration of quantitative proteomic data, pathways, and protein interactions. *IEEE Transactions on Visualization and Computer Graphics*, 16:609-620, 2010.
- [5] Martin I Krzywinski, Jacqueline E Schein, Inanc Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, and Marco A Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 2009.
- [6] Trevor M. O'Brien, Anna M. Ritz, Benjamin J. Raphael, and David H. Laidlaw. Gremlin: An interactive visualization model for analyzing genomic rearrangements. *IEEE Trans. Vis. Comput. Graph.*, 16(6):918-926, 2010.
- [7] Ji Soo Yi, Youn ah Kang, John Stasko, and Julie Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224-1231, November 2007.

Multiple Coordinated Views for Visualizing White Matter

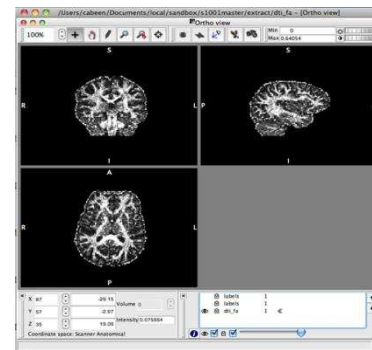
Ben Fichter

Co-Authors: John Meehan and Michail Michailidis

Computer Science Department at Brown University



A screenshot of our visualization



A screenshot of FSLView, an older application

ABSTRACT

We introduce a web-based visualization for displaying multiple MRI volumes. Prior to our tool, MRI viewers such as FSLView have been embedded applications restricted to a single brain per window. To facilitate interbrain comparisons, a helpful technique for isolating the effects of certain diseases on the brain, we have implemented a technique we call ‘multiple coordinated views.’ This describes the ability to simultaneously display three slices, top view, side view, and front view, per brain for an arbitrary number of volumes. When the user navigates through the brain for a given view, the same views of other brains automatically mimic this navigation so all slices displayed are at a consistent slice depth. Feedback suggests our tool will improve productivity of the creation of new white matter visualizations as well as being useful for collaboration between neuroscientists.

1 INTRODUCTION

White matter is the component of the brain responsible for connectivity; it is the ‘wires’ of the brain. Comparing the differences in white matter structure between healthy and unhealthy brains is a useful technique for understanding neurological diseases such as schizophrenia and Alzheimer disease [1].

White matter visualizations are fueled by MRI scans. These scans produce three dimensional datasets which typically reside in NIfTI files. These files can be converted to CSV files which can then be interpreted directly or ported into a database. The resolution of these MRI scans is usually

around 200 values per each dimension, making the CSV files around 8 million rows long.

A shortcoming of modern white matter visualizations is the inability to examine more than one brain in one window. Multiple windows become confusing, and slice depth consistency between windows must be manually maintained. Also, most MRI viewers are not web-based, which mandates software and data to be shipped to all the scientists who need it. We strive to solve these problems with our tool.

First, we had to decide how the front end would access the data. Initial testing of direct CSV reading produced slice fetching and creation times on the order of multiple seconds, indicating that this technique was not viable for an interactive web visualization. We realized using a database would be necessary for performance reasons. A matrix-based database SciDB has a function subarray() for easily fetching pieces of multiple dimensional data. We hypothesized that SciDB could provide the slice-fetching performance required for a smooth, interactive web visualization.

2 IMPLEMENTATION

2.1 Backend

To test our hypothesis, we loaded MRI data into SciDB and queried it to drive our visualization. The brain data is stored in SciDB as a three dimensional array with a single intensity value per voxel. Contrary to our hypothesis, the SciDB calls were not blazingly fast but rather on the order of a couple hundred milliseconds. Another performance problem stemmed from the output

from a SciDB query being limited to text or CSV, necessitating the creation of a PNG from python. Adding on network costs, the response time of the tool was less than desired.

In the interest of making the most usable tool possible, we adopted a new technique; loading pre-rendered PNGs into MySQL, a more mature database. This greatly increased performance, almost entirely eliminating database and PNG conversion costs.

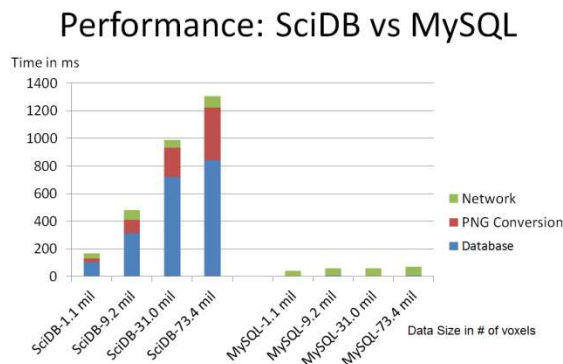
2.2 Frontend

Our tool has three drop-down menus for selecting the desired brain. The first menu displays all the possible studies that brains can be selected from. The second menu is for selecting a given patient, and the third selects the particular scan to be viewed. When the user clicks the 'Show' button, the brain described by the menus will be added to the display below.

Each brain gets its own column in the display. A column consists of a top view panel, a side view panel, and a front view panel as well as a color-coded slider for each view, a 'Prefetch' button, and a red X. Each panel has multi-colored crosshairs which tell the user where the other two slices are taken from. Pressing the 'Prefetch' button will grab all the slices for that brain from the database and cache them in the browser after some loading time in order to allow for optimal performance. Clicking the red X will close out that brain's column. Moving the color-coded slider for a given view will do a few things. Firstly, it will navigate either further or closer within the corresponding view, meaning the appropriate slice will be fetched and displayed. Secondly, the equivalent slider and view of all other displayed brains will react similarly to ensure all of these views display a consistent slice depth. The lines in the other views which match the slider color will move to indicate the new slice depth relative to the other dimensions.

3 DISCUSSION

3.1 Performance



As shown in the analysis graph, performance with SciDB is considerably worse than MySQL over all

measured resolutions. Performance for SciDB is on the order of hundreds of milliseconds with hefty database and PNG conversion costs as well as network costs. MySQL performance is almost exclusively dictated by network costs. This leads to a much smoother, more usable tool with MySQL. However, the comparison isn't completely fair as SciDB is required to fetch a range of data values which are later converted whereas MySQL only has to grab a single PNG which can be immediately displayed.

Although these differences make our current tool much more usable with MySQL, there are a few aspects of SciDB which are better. In SciDB there is no redundant storage because each voxel value is only stored once, whereas MySQL effectively stores everything three times because it needs a new set of PNGs for each possible view. Also, given the fine granularity in the output from SciDB, it is possible to do some real time data manipulations like adding 'diagonal' views or highlighting certain parts of the brain across all volumes. These manipulations would require data to be reloaded in MySQL, increasing storage redundancy and prohibiting real time interaction.

If SciDB added the ability to export PNGs directly from the database, performance would be significantly boosted and SciDB would become a much better option. With the implementation of smart prefetching, caching the next few slices in the browser while the user sits dormant, SciDB could also narrow the performance gap. Also, because network costs provide an initial hindrance to performance, SciDB may fair better in an embedded program.

3.2 Anecdotal Feedback

We had neuroscientist Daniel Dickstein and visualization specialist Ryan Cabeen use our tool and give feedback. Both collaborators were satisfied with the performance of the tool using MySQL. Ryan thought the tool would improve the productivity of the creation of new white matter visualizations. He also believed the web-based nature of the tool would facilitate remote collaboration. Some of the additional functionality the collaborators wanted to see was the ability to overlay tractography data onto the MRI scans and the option to view the position of the slices in 3D.

4 CONCLUSION

Our visualization aids in brain comparison by displaying an arbitrary number of MRI scans in coordination to be analyzed in one window. We also make MRI visualizations more accessible to more scientists by putting the tool and data on the web. Our tool is ready to be used and built upon to continue to improve white matter visualization.

REFERENCES

- [1] [Xekardaki, 2011] "White Matter Changes in Bipolar Disorder, Alzheimer Disease, and Mild Cognitive Impairment: New Insights from DTI," *Journal of Aging Research*, Volume 2011, Article ID 286564, 2011.

Measuring the Effect of Domain Knowledge on Visualization Learnability

Connor Gramazio*
Brown University

Alexandra Papoutsaki†
Brown University

Hua Guo‡
Brown University

David Laidlaw§
Brown University

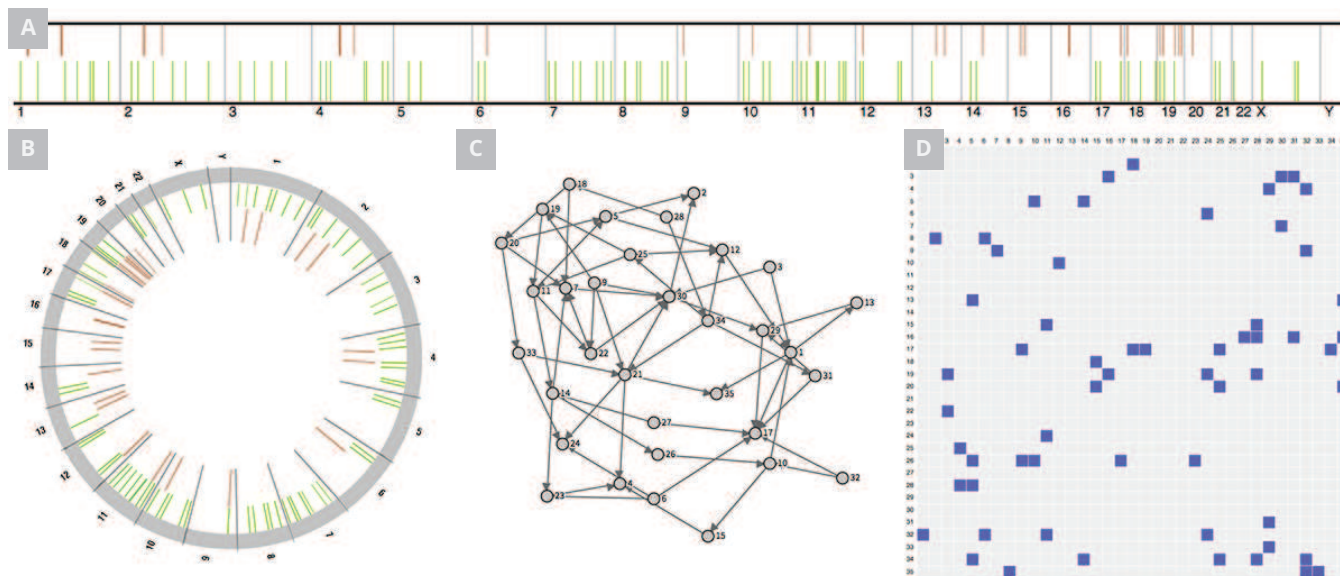


Figure 1: We discuss results of a study examining the learning of visualizations. The study compared biology experts and novices as they used Gremlin[3] (A) and Circos[2] (B), and graph theory experts and novices as they used node-link diagrams (C) and adjacency matrices (D).

ABSTRACT

Many visualization tools require users to learn how to perform new types of tasks or understand new visualizations, which can impact usability while users are still new to a tool. We describe findings from a study whose purpose was to discover ways domain knowledge can offset this learning cost, leading to what we call the learnability of a visualization. The study measured domain-area experts and novices as they completed a set of tasks on either biology or graph visualization images. Preliminary results show little difference in the learnability of visualizations and tasks between domain experts and novices.

1 INTRODUCTION

When evaluating overall utility of a visualization tool it is often useful to study how quickly users can gain proficiency using it, where proficiency is composed of accuracy and speed. We describe a study focusing on how this phenomena, which we call learnability, relates to visualization. The study examines experts and novices in biology and graph theory to determine how learnability can be impacted by familiarity with the data that is visualized. It is commonly accepted that systems for experts can sometimes be engineered differently than for novices, or vice versa, to maximize utility. Re-

search in the learnability of visualizations can help solidify that distinction when designing and implementing tools and techniques targeted at either novices or experts. We also believe this work contributes to better understanding how usability and domain knowledge impact overall utility of visualizations – problems recognized as important areas of research for the future of visualization[1].

2 EXPERIMENT

To test how learnability differs between novices and experts of a domain, we conducted a user study on twelve graduate students. The study asked participants to perform tasks on variations of visualization images shown in Figure 1. Each participant was asked thirty questions (fifteen for each visualization), where every five questions asked participants to complete a different type of task. Each task used the same set of visualizations. Task order and question order were randomized between subjects. An example of one task included in the study is, “Which node in the graph has the highest in-degree?” All tasks had participants sort or locate data shown in a visualization.

Participants were split into four groups: biology novices, biology experts, graph theory novices, and graph theory experts. Biology area participants were only given the biology version of our study, and graph theory area participants were only given the graph theory version of our study. Each version of the study took on the order of thirty minutes to complete. During the experiment each participant’s mouse interaction was tracked, task completion time logged, and all selected answers were recorded. The study was given in a web-based environment, though participants were in the same room as a proctor. Before beginning the study all participants were given instructions on how to read the visualizations included in their version of the study and how to input their answers. The instructions

*connor@cs.brown.edu

†alexandra_papoutsaki@brown.edu

‡huag@cs.brown.edu

§dhl@cs.brown.edu

were designed to minimize any impact on learning specific tasks asked in the study. The post-study questionnaire polled study participants on task and visualization difficulty.

2.1 Data and Visualization Generation

Five variations of each visualization were created using five datasets, leading to the same data being visualized in two ways.

2.1.1 Biology Variant

Biology visualizations were generated from sampled data used in O'Brien, et al.'s work[3] comparing Gremlin and Circos. We chose to sample the data to reduce the visual complexity of the diagrams to make them appropriate for the desired time allotment for each question.

2.1.2 Graph Theory Variant

Graph visualizations were created in D3 using randomly generated datasets. To equalize visual complexity between biology and graph theory variants, when generating graph data we first tested five variations of node and edge amounts ranging from eighteen nodes and twenty-five edges to fifty nodes and two-hundred edges. The dataset we used, which we believe best matched the complexity of the biology data, contained thirty-five nodes and sixty edges. Edge crossings for the node-link diagram were minimized during visualization generation.

3 RESULTS

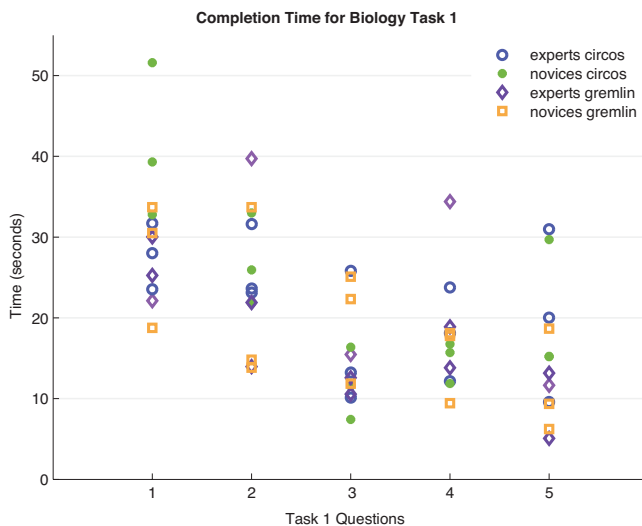


Figure 2: Completion times for Task 1 in the biology trial, where the question number corresponds to a specific question, rather than the order questions were presented in a given trial.

Our study adhered to a between-subjects design to examine how learnability differed over a population given the same visualizations. Results for all tasks show that there is not a clear separation of completion time between novice and expert groups in both biology and graph theory variants of the study, nor is there a discernable pattern between groups in task speed over time. Preliminary examination of captured data shows similar results in task accuracy. Subsections of our findings can be found in Figures 2 and 3.

The graph questionnaire showed that four of six participants found the matrix visualization easier to use for tasks and two participants found each visualization equally as difficult, with no issues reported while using the matrix and five of six participants reporting issues with the node-link visualization. The biology questionnaire showed a more equal spread with two participants preferring

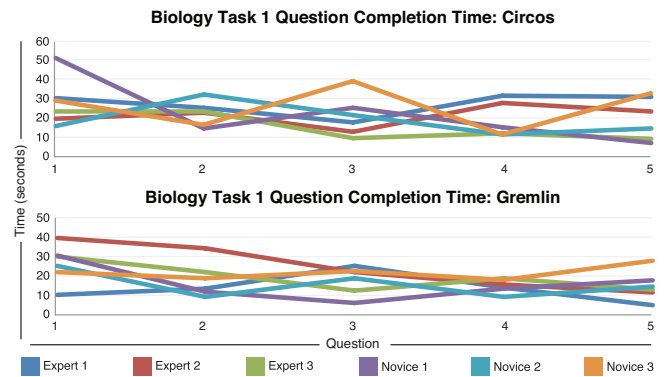


Figure 3: Completion times for Task 1 in the biology trial, where the question number corresponds to the order questions were presented (i.e., first question, second question, etc.).

Gremlin, one participant preferring Circos, and three participants preferring both equally. Three participants reported issues when using Gremlin and three reported issues when using Circos.

4 DISCUSSION

We believe that some of the noise in the results may have stemmed from how tasks were randomized in the study. Because each participant was asked questions within a given task in different orders, it is challenging to isolate learnability over time. Our analysis considered all responses regardless of accuracy, though different results might be seen should results first be filtered for correctness. Also, because experts sometimes tend to perform tasks more deliberately to make sure they “get it right,” times between experts and novices may not be directly comparable.

One possible solution to these problems is to normalize the data for each study participant and then create difficulty values for questions and question order posed to each participant based on the normalized results. Differences in approaches to complete a given question may also be discovered by looking at the captured mouse interaction data. This may indicate different initial task strategies across populations and how strategies evolved – or didn’t – over time. Learnability might also be uncovered by looking at average question completion time in each user across various time scales (e.g., the whole study or a particular task). Variation in average time and standard deviation might distill information unobservable in current analysis.

ACKNOWLEDGEMENTS

The authors wish to thank Brown University’s CSCI-0237 students for their invaluable feedback through the design of this work. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 2012148682. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- [1] C. Chen. Top 10 unsolved information visualization problems. *Computer Graphics and Applications, IEEE*, 25(4):12 – 16, july-aug. 2005.
- [2] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. Jones, and M. Marra. Circos: an information aesthetic for comparative genomics. *Genome research*, 19(9):1639–1645, 2009.
- [3] T. M. O’Brien, A. M. Ritz, B. J. Raphael, and D. H. Laidlaw. Gremlin: An interactive visualization model for analyzing genomic rearrangements. *IEEE Trans. on Visualization and Computer Graphics (Proc. Information Visualization ’10)*, 2010.

Visualizing Reaching Trajectories in 3D

Hobart Reynolds*
Brown University

Alicia Boucher†
Brown University

ABSTRACT

This paper presents a new visualization tool that was developed to display movement trajectories in three dimensions and permit greater interactivity with the trajectory data than existing applications allowed. A higher degree of interactivity grants cognitive scientists finer control over their data, while the enhanced perspective may allow researchers to gain new insights into human cognition. Though this tool will not replace existing applications, it gives researchers capabilities they did not have previously.

1 INTRODUCTION

Brown University's Perception, Action, and Cognition (PAC) Lab conducts experiments which track hand motions and eye movements. Researchers have written customized MATLAB programs to preprocess the trajectory data from the motion tracker before feeding it to statistical software for analysis. Although the trajectories contain three dimensional coordinates, the researchers are currently limited to two dimensional visualizations of the data.

This research project intended to develop a tool with a novel and interactive representation of the data. It is hoped that such a tool would not only allow for three dimensional presentation of the trajectory data, but also for the association of hand and eye trajectories so as to gain new insights into the cognitive processes behind and relationship between vision and movement.

2 BACKGROUND

2.1 Decision Making

Conventional models of cognition held that decisions were made through sequential mental processes. Under such a model, the brain would perceive stimuli, make decisions, and execute actions in order. However, a competing theory argues that the cognitive processes for planning and executing actions actually operate simultaneously in parallel.[8, 5]

Researchers at the PAC Lab have been testing this theory by analyzing the trajectories of subject's hands during reaching tasks. In one such experiment, the color oddity task, the subject is presented with three objects, one differently colored from the others. The subject is asked to reach to the odd colored target.[8, 7] While hand trajectories are straight when only one target is presented, in the color oddity task the motions swerve towards the distracting objects before moving to the target (see Figure 1). This suggests that multiple motor actions were planned and began executing, before one was ultimately chosen as the action to complete.

Other researchers have been studying further details of this model with various experiments involving hand trajectories.[8, 2] The field is also exploring the relationship between eye movements and their accompanying hand trajectories.[7, 6, 10, 9, 4]

*e-mail: hobart_reynolds@brown.edu

†e-mail: alicia_boucher@brown.edu

J.-H. Song, K. Nakayama / Vision Research 48 (2008) 853–861

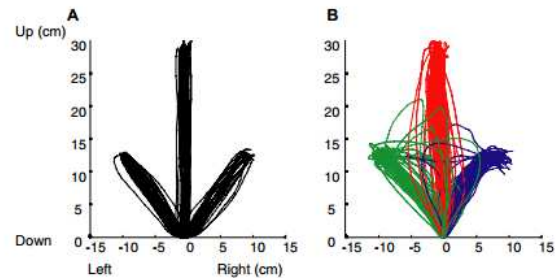


Figure 1: 2D plots of hand trajectories as projected onto a 2D screen. When one object is presented (left) the hand trajectories are mostly straight, while when multiple objects are presented (right) the hand initially curves towards the distracting targets.[7]

2.2 Related Work

Past research at the PAC lab has plotted and analyzed reaching trajectories and eye movements in 2D (see Figure 1).[8, 7, 6] Other researchers have similarly been limited to two dimensional plots. A previous effort to display trajectories in 3D was successful, but limited to analysis of a single trajectory.[1] This project intended to develop a tool that would not only allow researchers to view trajectories in 3D, but also allow them to view multiple trajectories at once and interact with their data. This interaction will be especially important when comparing hand motions and eye movements, which previous studies analyzing the two have lacked.[3]

3 METHODS

The visualization tool was implemented in Java so that the project team could take advantage of commonly used libraries. JOGL was used to create the graphical plots and Swing was used for the user interface.

The project team wrote an intuitive tool that could provide researchers with three dimensional representations of trajectory data. The application allows researchers to plot trajectory data in two or three dimensions. As in previous tools, the user may group trajectories together and plot them as groups (see Figure 2). The points of a trajectory may also be colored according to metrics such as velocity or curvature, giving the user a sense of how the motion develops over time (see Figure 3). The application has multiple viewports and enables the user to view two sets of data at once. This allows trajectories from different data sets to be plotted.

Because the hand tracker recorded measurements taken before the hand moved and after it had reached the target, a crucial preprocessing step for the researchers was truncating the trajectory so that metrics would only be computed for the actual motion. The existing MATLAB tools required the researcher to approximate a velocity threshold and only retain points for which the velocity was above the threshold. The new tool gives the user more control over this process by enabling them to select the starting and ending points of the motion on the velocity profile (see Figure 4). When viewing a velocity profile, the associated point in space is highlighted in the

other viewport, so the process is more intuitive than in MATLAB.

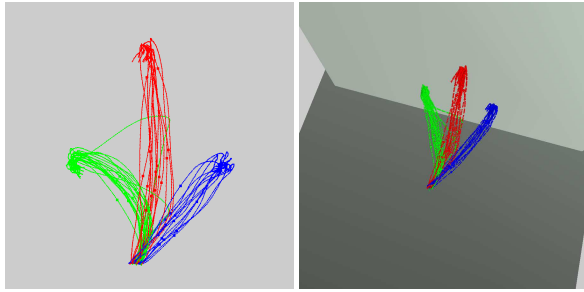


Figure 2: Trajectory data grouped according to the target. The data can be presented in 2D (left) as in Figure 1, or in 3D (right).

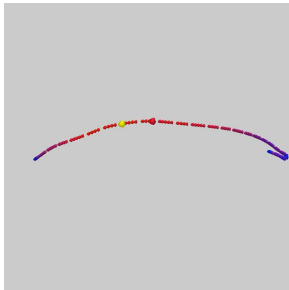


Figure 3: A single trajectory in 3D with points colored by velocity. Blue points have low velocity while red points have high velocity. The big yellow point indicates the currently selected point, while the big red point indicates the point of maximum velocity.

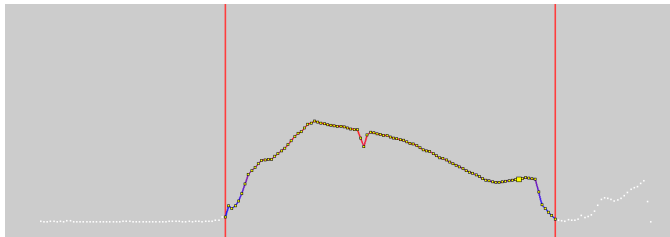


Figure 4: The velocity profile of the trajectory. The red lines indicate the user-selected starting and ending points, while the white dots indicate the points which will be ignored.

4 RESULTS

To evaluate the application, the project team asked a researcher to perform an analysis task in his MATLAB tool and in the application. The researcher first had to group the trajectories according to experimental attribute and then compute a metric. On average, the researcher needed 95% more time to complete the task in the new application than in MATLAB. This was primarily due to the fact that researchers use widely varying experimental attributes to group their data, and so it is more difficult to partition the data in the tool than it is in the customized MATLAB programs. The researcher also performed the preprocessing task of cropping the trajectories, which he needed 69% more time for in the new application than in MATLAB, once again due to the difficulties selecting trajectories.

However, the application introduced capabilities to the researchers which they had not had previously. As mentioned before,

the ability to select specific points enabled researchers to crop the trajectories more accurately. Additionally, some trajectories reflect irregular motions (such as the user slowing down in the middle of the reach). These trajectories could not be cropped with the threshold method and would previously have been rejected. The new tool allowed these trajectories to be retained and analyzed.

The researchers believe that the innovative 3D rendering of trajectories could provide new insights into the data which the original 2D plots could not. They also believe that new ability to view associated hand and eye data will greatly assist in future research.

5 CONCLUSION

It remains to be seen whether this new perspective of the trajectory data will allow researchers to gain new insights into cognition. The researchers themselves are enthusiastic about the possibilities presented by this tool. The application as it stands will not replace the current MATLAB tools entirely, as it is currently too cumbersome to be used for analysis. However, it supplements them by allowing greater interactivity with the data and may supplant them in preprocessing the trajectories.

Future research in the field will require additional visualizations to enable viewing of more varied data. The Lab hopes to view fMRI data imaged during reaching tasks so as to observe the relationship between motion and neural activity. Above all, these tools will require interaction so that the researchers can better analyze and understand their data.

ACKNOWLEDGEMENTS

The authors would like to thank the researchers at the Perception, Action, and Cognition Lab for all their help and support. In particular, we would like to thank Professor Joo-Hyun Song for her guidance of the development of the tool and Jeff Moher for his extensive assistance in its implementation and evaluation.

REFERENCES

- [1] D. R. Faria and J. Dias. 3d hand trajectory segmentation by curvatures and hand orientation for classification through a probabilistic approach. *2009 IEEE/RSJ Inter. Conf. on Intel. Robots and Systems*, pages 1284–1289, 2009.
- [2] A. P. Georgopoulos and et al. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J. Neurosci.*, 2:1527–1537, 1982.
- [3] P. J. and et al. The coordination of eye, head, and hand movements in a natural task. *Exp. Brain Res.*, 139:266–277, 2001.
- [4] R. M. McPeck. Incomplete suppression of distractor-related activity in the frontal eye field results in curved saccades. *J. Neurophysiol.*, 96:2699–2711, 2006.
- [5] D. A. Rosenbaum and et al. The problem of serial order in behavior: Lashley’s legacy. *Hum. Mov. Sci.*, 26:525–554, 2007.
- [6] J.-H. Song and R. M. McPeck. Eye-hand coordination during target selection in a pop-out visual search. *J. Neurophysiol.*, 102:2681–2692, 2009.
- [7] J.-H. Song and K. Nakayama. Target selection in visual search as revealed by movement trajectories. *Vision Res.*, 48:853–861, 2008.
- [8] J.-H. Song and K. Nakayama. Hidden cognitive states revealed in choice reaching tasks. *Trends in Cogn. Sci.*, 13:360–366, 2009.
- [9] S. van der Stigchel and et al. Eye movement trajectories and what they tell us. *Neurosci. Biobehav. Rev.*, 30:666–679, 2006.
- [10] T. Yang and M. N. Shadlen. Probabilistic reasoning by neurons. *Nature*, 447:1075–1080, 2007.

Multiple Coordinated Views for Visualizing White Matter

John Meehan*

Michail Michailidis
Brown University

Benjamin Fichter

ABSTRACT

We present BraCoViA (Brain Coordinated Visualization Application), a tool that provides simultaneous coordinated navigation of multiple brain scans over a web interface. BraCoViA gives the user the ability to quickly make comparisons between several subjects at once, allowing neuroscientists to better analyze connections between white matter patterns and brain disease. To accomplish this, we employ SciDB, a multi-dimensional array database, and evaluate its use within the context of web visualizations.

Keywords: White matter, neuroscience, coordinated views, SciDB, databases

1 INTRODUCTION

Advances in magnetic resonance imaging (MRI) have created opportunities to map the human brain never before possible. Neuroscientists are able to determine the structural integrity of cerebral white matter, the connections between various sections of the brain [1]. These pathways are crucial for understanding the differences between a healthy brain and a diseased one, and can be visualized as hundreds of curves running through a model of a brain.

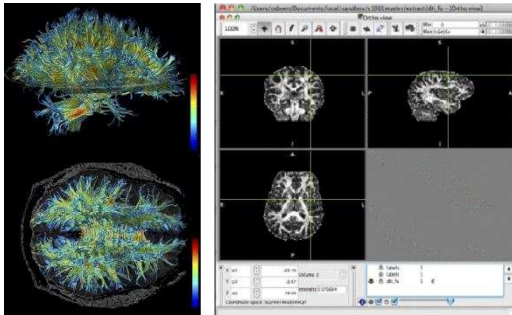


Figure 1: Examples of current white matter visualizations

While very sophisticated, current visualization tools have several shortcomings. Due to a lack of centralized controls, it is difficult to efficiently analyze multiple sets of brains at once. Such functionality would be extremely helpful to neuroscientists, as it would allow far easier comparison between healthy brains and abnormal brains. In addition, it is currently difficult for scientists to share their findings with collaborators across the country, as specific software is required and all files must be copied from network to network. We address both of these issues with our tool.

* john@cs.brown.edu

2 IMPLEMENTATION

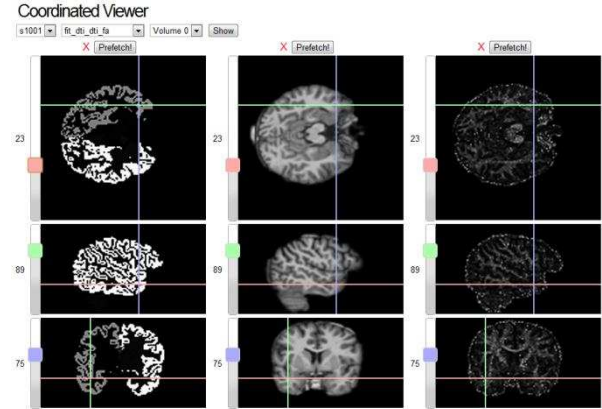


Figure 2: A screenshot of multiple MRI scans of the same brain, as viewed with BraCoViA

2.1 Front End Visualization

As shown by Figure 2, BraCoViA displays as multiple brain scans across the browser page. The windows representing the three directions of view for a single scan (top, side, and front views) are placed vertically. Each slider to the left of a window controls which “slice” of the brain is displayed for that view. In addition, moving the slider for one view also moves the corresponding sliders for the same view of other scans. This allows the user to move through all scans at once, providing for better comparisons. As a slider is moved for one view, the corresponding sliding bar (matched by color) is moved within the two other views. This bar aids the user in knowing which point he or she is currently viewing within the three-dimensional space.

For example, suppose that the user wants to view a set of three brains that have the same MRI scan type. She first selects the brains desired from the dropdown menu, ordered by subject first, followed by scan type. Once the scans are selected, she wishes to see the exact center of each brain from the “top” view. She moves the “top” slider (red) to the desired position. Simultaneously, the identical sliders for brains two and three move as well. As these three sliders are moved, the “top” views change to display the desired slice of the brain. Additionally, the red indicator bars on the “front” and “side” views of the brain also move to show the user’s position within the brain.

2.2 Database Infrastructure

MRI scans are stored in three-dimensional image files known as “nifti” files. These scans must be converted to a format that is quickly and easily readable from a web application. Initially, SciDB seemed to be a perfect database for this purpose. Because SciDB is designed for array-based storage, it

is possible to store each voxel in its own cell [2]. This allows the system to query the data in such a way that it can pull individual “slices” of data, which can then be displayed on screen (Figure 3). However, due to performance concerns (discussed in section 3.2), it became apparent that a different approach would be needed to provide the same interactivity as comparable software.

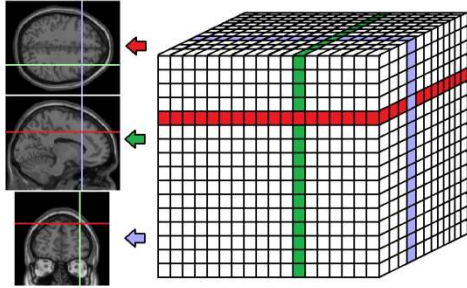


Figure 3: A visual representation of voxels stored in SciDB, and how images are generated from the data

The eventual alternative was storing each brain slice as its own image within a relational database (MySQL). This is a more traditional approach, and allows for any slice to be queried extremely quickly since the actual image is pulled directly from the database with no conversion required. The downside to this approach is that each voxel within a brain must effectively be stored as three pixels, one for each view.

3 DISCUSSION

3.1 Feedback

To evaluate the scientific contribution of BraCoViA, we presented the tool to neuroscientist Daniel Dickstein for feedback. He was impressed with the results given the scope of the project, and stated that the tool would improve comparisons between brains. He also proposed numerous additions to the project, viewing BraCoViA as more of a starting point for future visualizations rather than a completed project.

We also presented BraCoViA to fellow collaborator Ryan Cabeen, a specialist in neuroscience visualizations. Cabeen indicated that the tool would be particularly useful for drawing connections between various white matter filters, which would improve his productivity in the creation of new visualizations. He also noted that our project would be very helpful for collaborating with Dickstein and other scientists, since it would provide a way for all parties to interact with the same data remotely.

3.2 Performance

3.2.1 SciDB Challenges

There were numerous challenges that prevented the use of SciDB for our infrastructure. Most challenges were caused by an absence of features that are present in more mature databases. For instance, while SciDB allows for voxel-by-voxel storage and querying, there is no easy method to retrieve the information from the database. Currently, the best way to pull data from SciDB is to output the query results as text. However, this means that the backend of BraCoViA is required to traverse the data line by line, which is quite slow. In addition, our program then must convert the data to a PNG image file. By the time the image is ready, too much time has elapsed for the tool to be truly interactive. If SciDB were able

to directly output its data to a PNG file rather than text, we believe that the performance would be far more acceptable.

3.2.2 Comparison to MySQL

When we switched BraCoViA to use pre-generated images stored in MySQL, performance improved dramatically. On an average brain scan, a slice image was generated in less than 40 milliseconds, far faster than the nearly 200 milliseconds required by SciDB. The majority of these 40 milliseconds were due to network costs, which are unavoidable in a web application. We also compared the performance of prefetching all of the data, which involves loading every slice and caching it in the browser. Again, we found the MySQL performance to be vastly superior, loading in less than a second as opposed to SciDB’s 32 seconds.

We also compared the performance across multiple sizes of brain volumes, increasing the size from 1.1 million voxels to 73.4 million voxels. MySQL continued to scale far more impressively than SciDB, despite effectively storing the data three times (once for each dimension).

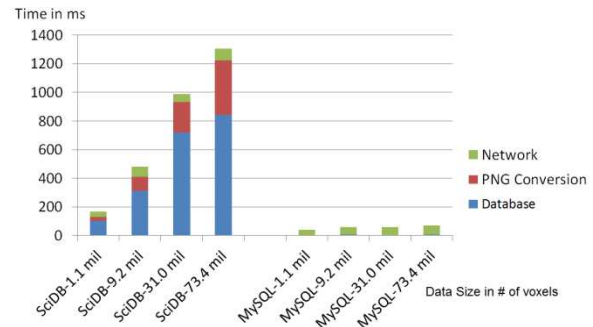


Figure 4: Performance comparison between SciDB and MySQL

While SciDB performed poorly for our purposes, we believe that it may still be useful for visualizations that involve more complicated data interaction. For instance, suppose our visualization allowed the user to highlight certain portions of the brain in one view with the same portions highlighted in corresponding views. This would require more granular access to the data, separating a brain by region rather than simply displaying it as unified images. Because SciDB stores each voxel separately, it would be possible to retrieve and change portions of the data without affecting the others.

4 CONCLUSIONS

By coordinating windows for multiple brain volumes at once within a web application, BraCoViA is able to improve the productivity of neuroscientists and allow for better collaboration across the globe. Our research shows that SciDB is currently not advisable for visualizations of this type. However, we believe that with improvements, SciDB may still have applications for data-intensive visualizations.

REFERENCES

- [1] Stephen Correia, Stephanie Y. Lee, Thom Voorn, David F. Tate, Robert H. Paul, Song Zhang, Stephen P. Salloway, Paul F. Malloy, David H. Laidlaw. Quantitative tractography metrics of white matter integrity in diffusion-tensor MRI. *NeuroImage*, Vol 42, Issue 2, pages 568-581, 15 Aug 2008.
- [2] Michael Stonebraker, Jacek Becla, David Dewitt, Kian-tat Lim, Stan Zdonik. Requirements for science data bases and SciDB. CIDR 09, 2009.

BraCoViA: Multiple Coordinated Views for Visualizing White Matter

Michail Michailidis
Brown University

John Meehan
Brown University

Benjamin Fichter
Brown University

ABSTRACT

Visualizing white matter within multiple MRI brain scans, simultaneously, can provide much insight in comparisons between healthy and unhealthy subjects. Current visualizations either have coordinated views within one brain volume, or have disconnected views from more than one brain volumes. Furthermore their implementations are standalone applications that use files as a form of storage, and can have performance issues when multiple brain coordination is needed. We hypothesize that by implementing a web application that uses an array data storage engine like SciDB, we can provide multiple brain coordination, solving the resulting performance drop due to more data fetched and accessibility issues of standalone applications.

Keywords: multiple brain coordination, white matter visualization, SciDB.

1 INTRODUCTION

Advances in magnetic resonance imaging (MRI) have created opportunities to map the human brain as never before possible. By analyzing the directions by which water diffuses throughout the brain, we are capable of determining the structure of cerebral white matter. New methods have been developed by visualization researchers to use tractography methods to better represent the white matter pathways throughout the brain (see Fig. 1). These pathways or “streamtubes” are crucial in understanding the differences between a healthy brain and a diseased one[2], and can be visualized as hundreds of curves running through a model of the brain.

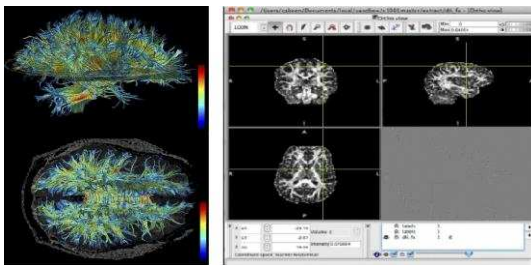


Fig.1: White matter structure of human brain (taken by MRI) Fig.2: Fslview Brain Vis. Tool

Current implementations, like Fslview (see Fig.2) provide single brain volume visualization. That makes the comparison of multiple brains difficult. Our implementation provides the ability to compare many coordinated analyses of the same brain. Also it supports multiple brain coordination visualization from different subjects. Furthermore being a web application improves accessibility. The implementation section describes the infrastructure and the frontend visualization. The discussion

section describes the challenges and the evaluation.

2 IMPLEMENTATION

Our application, named BraCoViA (Brain Coordinated Visualization Application) being online and distributed needs to be based on a database storage engine, instead of raw nifti files. That also provides the ability to change the stored volumes, universally. We built the needed infrastructure with 1) SciDB database support and 2) MySQL database support, which are discussed below. The multiple approaches will help in quantitative and qualitative evaluation of BraCoViA.

2.1 SciDB

MRI scans are essentially three views of a single brain (top, side and front), and brains are obviously three dimensional. As such, the most natural way to represent the color intensities in a three dimension space is to use an array database, like SciDB[1]. Our initial intention was to put a single brain volume, but since SciDB provides more than 3 dimensions, we were able to put multiple brain volumes in a single array/database. Some nifti files can have more than one volumes in them, so the 1-to-1 mapping from nifti files to arrays was possible. So for a brain that is 181x181x216 slices for the three views we are storing 7076376 pixel color intensities. The storage of a single brain volume looks like Figure 3. The fetching is being done by querying the slice needed from SciDB. The SciDB output is in CSV flat format that is being converted with server side code to a 2d matrix, which is then transformed in PNG with base64 encoding and returned to the frontend.

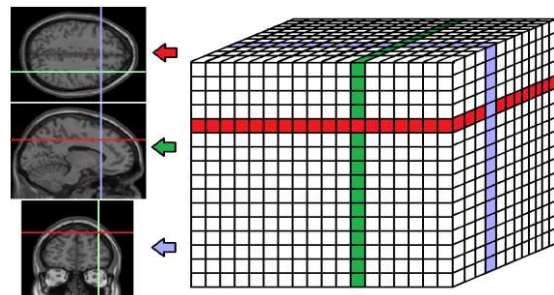


Fig.3: Single brain volume stored in SciDB and corresponding slices

2.2 MySQL

The MySQL database is loaded after SciDB load has finished. The code that converts the SciDB output to PNG is used and the PNGs for the slices are stored in the database. Each PNG is either for top, side or front view. So for a brain that is 181x181x216, 578 PNG slices are stored as text in the database.

MySQL was used to provide the fastest performance possible, and be compared with SciDB and raw file loading. MySQL is very good at caching the slices that were requested before, so it is faster than raw file loading and provides the same capabilities with the nifti files, and that's why direct file loading wasn't included in the evaluation.

2.3 Frontend

In the frontend (see Fig. 4) the user can have as many brain

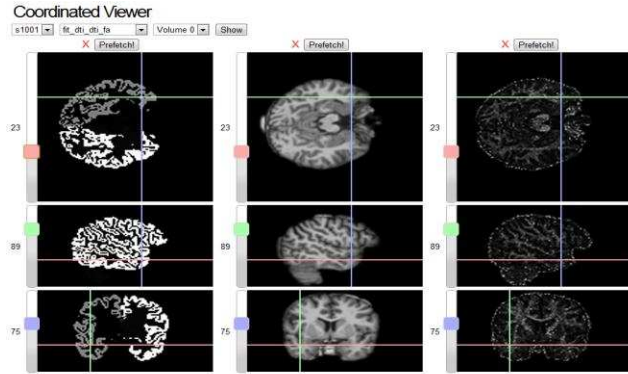


Fig.4: BraCoViA Multi-Brain Coordinated Viewer

volumes as she prefers. She can use a set of drop-down menus that are populated according to the paths of nifti files. Each brain volume has three views: top, side and front. Each view has a uniquely colored slider (three colors in total). Coordination is being done in a single view across multiple brains. For example moving the slider on the front view of a brain, moves the front sliders to all other brains, synchronizing the front view of all the loaded brain volumes. In addition moving the sliders in one view changes the corresponding colored bars in the other views, so the user knows where she is within the volume brains.

3 DISCUSSION

The implementation of BraCoViA faced many challenges. We discuss the performance and flexibility issues we had with SciDB and MySQL, the latter being used, because SciDB didn't perform as expected. The discussion covers quantitative and qualitative evaluation.

3.1 Performance

We did a performance comparison between SciDB and MySQL (see Fig. 5). We used brain slices with four different resolutions to

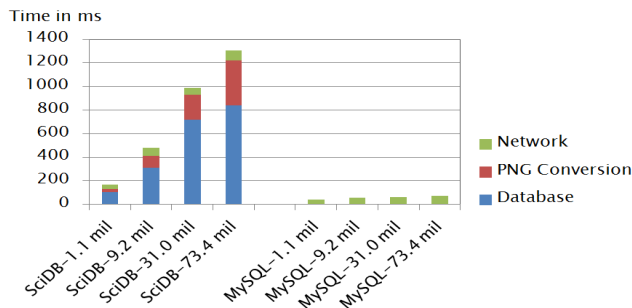


Fig.5: SciDB and MySQL performance evaluation

assess how these two implementations scale.

According to the measurements, SciDB had to spend about 20% of the time converting from stored color intensities to PNG format while MySQL had PNGs prestored. Even if SciDB had a way to output in PNG format directly, 70-75% of the time is due to the database being slow to fetch the requested slice. Small part of the performance hit is accounted to network costs. As we can see MySQL is almost instant to fetch the requested slices, which is expected. SciDB with its multidimensional array schema

disappoints on this application. The numbers on the vertical axis denote the millions of pixels each slice has.

It is quite clear, that if we want to use SciDB as a storage engine without mutating or doing any calculations on color intensities stored, we will have as a result a not so interactive application with only 1-5fps. If we want to have an interactive/high frame rate application, MySQL serves very well this opposite extreme, but sacrificing the fine grained nature of SciDB.

One should note that even though smaller chunk sizes in SciDB, changed the performance slightly, made the loading time of a single brain to take hours and the rendering problematic. With chunk sizes equal to 1 the performance improvement wasn't enough to support our hypotheses.

3.2 Feedback

Our application was firstly evaluated by Daniel Dickstein, who is a neuroscientist. Although he liked the multiple coordinated views and current application capabilities, he suggested that this could be the start of a more sophisticated web application.

Ryan Cabeen, who is a visualization specialist, found the application interesting and very promising in providing a more helpful way for him, to make new brain, or improve current visualizations, out of MRI scans.

3.3 Other challenges and future work

The inconsistency of brain positions, different color intensity ranges and different resolutions in MRI scans, are some challenges that make brain volume coordination more difficult than expected. We addressed the different color intensity ranges issue but more work is needed to automate the solution of the other too.

Future work can involve tractographies integration with MRI scans and 3d visualization of the three slices cutting each other, and providing a more intuitive mechanism for browsing through brain volumes.

4 CONCLUSION

BraCoViA certainly helps brain scientists or visualization scientists in various ways. It provides accessibility and easy distribution of the information. SciDB doesn't seem to be the best candidate for this kind of application but possibly it is not mature enough, for the time being. Even with MySQL, prefetching of the whole brains was needed to have performance(fps) comparable to fslView, something expected due to network costs in web applications. Finally, visualizations that involve any on-the-fly calculations of data bigger than RAM, would take advantage of SciDB's more fine-grained nature compared to raw files or MySQL prerendered pngs.

REFERENCES

- [1] [Stonebraker et al. 2009] Michael Stonebraker, Jacek Becla, David Dewitt, Kian-tat Lim, Stan Zdonik, "Requirements for Science Data Bases and SciDB" CIDR 09,2009.
- [2] [Xekardaki, 2011] Aikaterini Xekardaki, Panteleimon Giannakopoulos, Sven Haller, "White Matter Changes in Bipolar Disorder, Alzheimer Disease, and Mild Cognitive Impairment: New Insights from DTT" Journal of Aging Research, Volume 2011, Article ID 286564, 2011.