

Data Analysis on Interactive Whiteboards through Sketch-based Interaction

Jeffrey Browne^{1,2}, Bongshin Lee¹, Sheelagh Carpendale^{1,3}, Nathalie Riche¹, Timothy Sherwood^{1,2}

¹Microsoft Research, ²University of California, Santa Barbara, ³University of Calgary

*jbrowne@cs.ucsb.edu, bongshin@microsoft.com, sheelagh@ucalgary.ca,
nathalie.henry@microsoft.com, sherwood@cs.ucsb.edu*

ABSTRACT

When faced with the task of understanding complex data, it is common for people to work on whiteboards, where they can collaborate with others, brainstorm lists of important questions, and sketch simple visualizations. However, these sketched visualizations seldom contain real data. We address this gap by extending these sketched whiteboard visualizations with the actual data to be analyzed. Guided by an iterative design process, we developed a better understanding of the challenges involved in bringing sketch-based interaction to data analysis. In this work we contribute insights into the design challenges of sketch-based charting, and we present SketchVis, a system that leverages hand-drawn input for exploring data through simple charts.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces.

General terms: Design, Human Factors.

Keywords: Sketch-based Interaction, Visualization.

INTRODUCTION

When first brainstorming about the aspects of the data to be explored, or when multiple people are discussing a data set (e.g., meetings to discuss quarterly performance or profits), people often initially work on whiteboards. There, interaction occurs simply through a pen and eraser, allowing people to sketch visualization ideas naturally and collaboratively. However, since adding data to these sketches would require tedious repetition, the underlying data usually remain buried in a computer. Alternatively, choosing to work out visualization ideas on a computer offers data access and computational power via tools, but interaction is constrained to classic mouse and keyboard.

The freeform nature of sketch interaction lends itself to fast, naturalistic interaction without the use of widgets or menus. There is also a growing understanding that the act of sketching is a fundamental tool for promoting thinking, insight, and inspiration [2]. We explore the possibility of

combining naturalistic sketch-based interaction and powerful desktop data analysis tools by bringing data to sketched whiteboard visualizations. This enables information exploration and examination in more fluid, natural ways, extending the traditional advantages of the whiteboard.

This paper presents SketchVis, a proof-of-concept system that leverages hand-drawn input to support exploring data through simple charts. The core idea is to augment the initial structure of a bar chart or scatter plot drawn on a whiteboard with simple visualizations of actual underlying data. After sketching a chart frame, people can interact with the data through simple actions, which support filtering classes of data, applying mathematical functions, scaling axis, and selecting different aspects of the data (Figure 1). This paper also presents our insights, gathered from an iterative design process, into the design challenges present in sketch-based charting tools. Our iterative design included ten sessions in which participants explored their own data sets. This helped us refine our interface and demonstrate the potential for supporting interactive data exploration through a simple, easy-to-learn sketch-based interface.

RELATED WORK

Most information workers rely on simple, well known data graphics such as scatter plots, bar charts, line graphs, and pie charts. Our goal is to make it possible to directly sketch and to manipulate simple data graphics through interactions that can be readily used during ideation processes.

Related work includes the support of data queries by making a line-graph sketch [9]. Holz and Feiner [4] support both spatially and temporally relaxed sketch queries to indicate the level of similarity that can vary across a search pattern. Graph Sketcher [8] employs drawing interactions to ease the creation of Quantitative Concept Diagrams (QCDs), graphs that show conceptual trends (e.g., line A is growing faster than line B). Sketched objects can be directly manipulated to generate the final output. However, the resulting graphs are not generated from a data set, but rather are “made-up” to explain a concept. In this respect, Graph Sketcher is a hybrid between traditional drawing and graphing tools, but does not allow people to explore existing multidimensional data sets.

In NapkinVis [3], symbolic gestures instantiate chosen visualizations. In contrast, SketchVis directly recognizes each chart’s structure and axis labels from persistent

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ITS 2011, November 13-16, Kobe, Japan.

Copyright 2011 ACM 978-1-4503-0871-7/11/11...\$10.00.

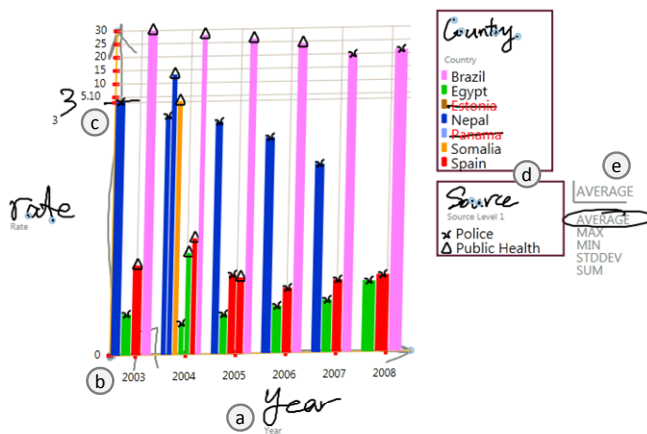


Figure 1: SketchVis integrates real data with hand written selections (a: axis labels) and sketch-based controls (b: axis arrows, bar stroke; c: tic marks; d: legend area; e: transformation menu).

sketched strokes. Since modifications to a chart in NapkinVis are made through transient gestures, people may have a difficult time remembering all relevant gestures, or even recalling the meaning of particular charts. Charts in SketchVis, on the other hand, always reflect the data as specified directly on axis labels, and the “state” of the visualization is clear as it is derived from persistent strokes.

NiCE [7] is a sketch-based tool designed to recognize and visualize mathematical formulae. Line-graphs can be generated from math functions. In contrast, SketchVis augments sketched charts by visualizing selected existing data, populating the visualization from an underlying data file.

ITERATIVE DESIGN

During the development of SketchVis, we conducted a series of iterative design sessions where we observed people analyze their own data sets using the SketchVis prototype. Our main goal was to progress to a more usable design by iteratively investigating usability and interaction methods. Between each session, we adjusted SketchVis to address issues and to introduce suggested features.

Participants and Procedure

We recruited 10 participants (7 females) internally from a large software company; seven computer science researchers/interns, two designers, and one administrator. Two sessions had multiple people (one group of two and another of three). Participants were asked to send their data sets prior to the session. After receiving a short demonstration of the system, participants began their analysis while thinking aloud. They were provided with a “cheat sheet” containing a screenshot of the results from an example analysis. Each session lasted approximately one hour. Through the ten sessions, observations and discussions, we learned several key challenges and trade-offs involved in blending sketch and data visualization.

Design Challenges

Precision vs. Freeform Input. When augmenting drawn strokes with data plots, there is a tradeoff between leaving drawn input untouched and presenting perfectly accurate renderings (e.g., axes, labels). Previous research [6,10] shows that unfinished, “messy” strokes can serve productivity since they inspire designers to think deeper and try more ideas. This may indicate that maintaining an unfinished look can inspire people to more frequently revise their charts, rapidly providing them with visualizations of their data from many different perspectives. However, freeform pen strokes come with an unavoidable level of imprecision. This conflicts with the need to accurately represent the underlying data, and could potentially undermine the process of gaining confidence in a new visualization tool. Interaction with SketchVis occurs only through hand-drawn sketches, but the data is plotted along straight axis lines.

Pure Sketch vs. Alternative Interactions. A second tension exists between creating a system that makes strict use of a stroke-based interaction model, where every interaction is expressed through drawing or erasing some strokes, versus using command widgets or a menu/button paradigm. On one hand, interaction through drawing offers a natural interaction method for many of the visual tasks in charting. Most people are familiar with the basic drawing and erasing interactions of a pen, and so they bring a great deal of prior experience and expectation when presented with a whiteboard-like interface. On the other hand, a system of purely stroke-based interactions has limitations. Some tasks may suffer in terms of speed by requiring an explicit stroke, for example long sequences of text entry or the issuing of non-persistent commands. As an exploration of sketch-based charting, SketchVis uses solely draw/erase interactions.

Rich Visualization vs. Visual Clutter. There is also a tradeoff between supporting as much drawing space on the board as possible and showing individually interesting, complex visualizations. The data visualization aspects of SketchVis push our design toward rich charts that portray as much information as possible. This often implies a loss of screen real estate and increase in visual complexity. In the context of sketch-based interaction, the display screen is also the input medium. Therefore, physical expansion of an overlaid chart shrinks the area available for additional strokes. SketchVis makes use of user-drawn strokes wherever possible (e.g., axis labels, tic marks) to indicate the state of the chart without cluttering the screen with large graphics.

Complex Interaction vs. Ease of Use. The more features we want to support, the more difficult it becomes for people to interact with the system. To develop this interface, everything from the initial specification of a chart needs a “vocabulary” for augmented sketch-based charting. Thus unusual data based features that do not exist with traditional whiteboards require some special gesture or an explicit

selection mechanism. If gestures are used, they need to be taught and remembered. If instead some interaction method such as a button or menu is used, people do not need to remember a gesture, but this introduces an interaction mode apart from simple stroke drawing. SketchVis strikes a balance with transformation actions occurring through persistent strokes (e.g., bar stroke, tic marks), which are implicitly reversible through erasure.

SKETCHVIS

SketchVis is a new sketch-based visualization system, where interaction is through a digital pen (stylus) either on a wall-sized screen or on a tablet PC. Through our iterative design, SketchVis grew to combine sketched frames and hand written data selections (Figure 1). We describe SketchVis and its interactions by following a simple scenario in which Alice explores her crime rates data.

After loading her data set, Alice *draws* two intersecting arrows (for X and Y axes) as she would draw a chart on a whiteboard; the size and location of these arrows determine the physical dimensions of the chart.

After reviewing the data fields provided by SketchVis as a side legend, Alice thinks of analyzing the rates for each country, so she *writes* a “c” in the X axis text entry area. Since “Country” is the first data field that starts with “c,” it is chosen and displayed. Similarly, she then *writes* an “r” in the Y axis field and “Rate” is displayed. Once the X and Y axes are mapped, SketchVis draws straight lines over the hand-drawn axes and displays the relevant data.

From the initial scatter plot, Alice sees that there are multiple data points for each country (Figure 2a), and first wants to compare the countries’ maximum crime rates. She *circles* “MAX” in the function selector menu, and SketchVis shows maximum values for each country (Figure 2b). Now Alice wishes to compare the average rate for each country,

and thus she *erases* the circle mark and selects “AVERAGE” by *writing* an “a” in the function entry box, just as she did to specify axes. Then, SketchVis updates the view to display average values for each country (Figure 2c).

To view the relative changes in crime rates between different countries more clearly, Alice *draws* a bar shape across the X axis, and the system converts the chart into a bar chart (Figure 2d). There, she sees that Brazil’s rate is, on average, more than twice that of Panama, the next highest. Now she wonders if the rates show any trends over time. So, she *erases* the “c” label and *writes* a “y” in the X axis field. Since the average function is still in effect, she can see that the average rate tended to decrease since 2003 (Figure 2e).

She now wants this information broken down by country, so she *writes* a “c” in the color legend area. Upon selection, the box expands into a legend where each value in the column is mapped to a color, and each point in the plot is colored according to its value in that column (Figure 2f). She then notices that, while Brazil’s rate tended to decrease, Panama’s rate was actually increasing.

To focus on the difference between Brazil and Panama more clearly, Alice crosses out (draws a line through) all other countries from the color legend. SketchVis hides points with values matching the crossed out value and automatically adjusts the scale of the X and Y axes to fit the remaining data (Figure 2g). Continuing her analysis, Alice *erases* the filter lines and sees that the small differences in crime between Spain and Egypt are lost in the scale of the other countries’ rates. To view Spain and Egypt more clearly, she *draws* a tic as a small line high on the Y-axis and *labels* a new value “3” to force the axis to rescale. From this rescaling, she sees that Spain has a higher rate than Egypt for every year (Figure 2h).

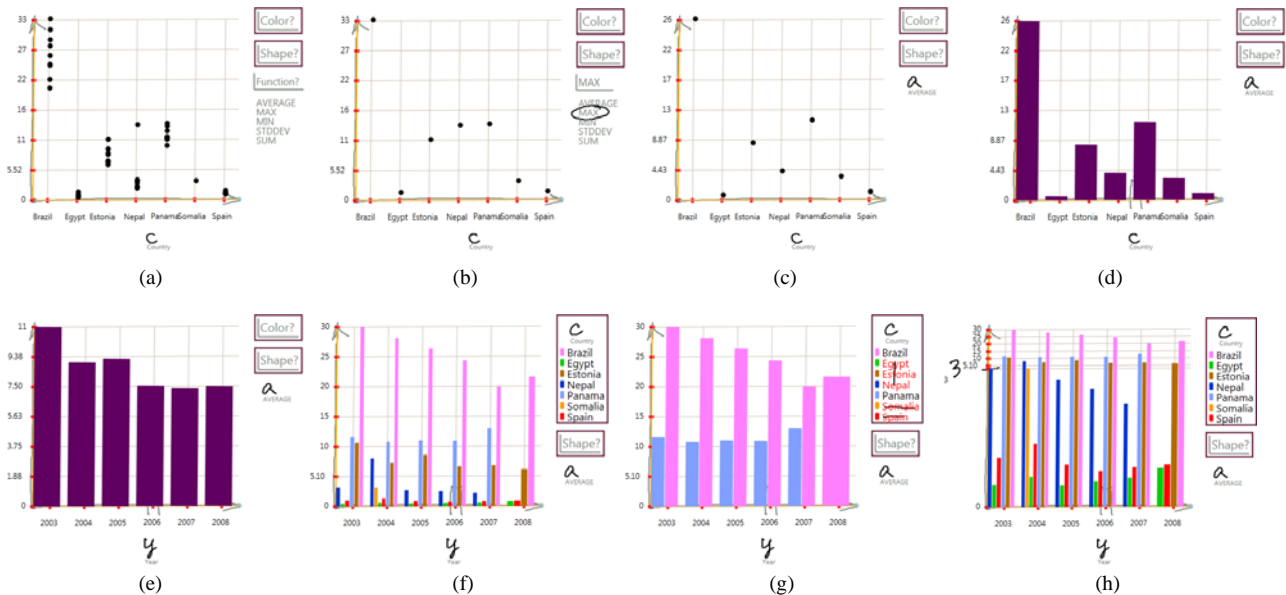


Figure 2: SketchVis helps people explore a crime rates data set through simple, interactive sketches.

DISCUSSION

Data exploration through sketch-based charting is a promising new information visualization paradigm. Many of the metaphors from charting on traditional whiteboards transfer readily to computationally supported whiteboards. However, many issues remain to be explored.

Sketch as an Exclusive Interaction Method. This choice could limit interactions with charts to creation or erasure of strokes, stopping modification (e.g., stretch, translate) of strokes after they are drawn. This results in requiring erasure and redrawing to change a chart's scale. Potentially, this could be addressed by combining the pen with touch input capabilities, where the mode could be determined through the tool being used; a stylus would imply inking mode, while touch could imply modification mode [5].

More generally, one of the major usability issues was handwriting-based axis selection. Our participants became frustrated by having to rewrite misrecognized text, effectively letting the interface get in the way of quickly selecting a new axis to view. Axis selection that supports the full freedom of input afforded by freehand input could alternatively be accessed through pen-based widgets.

Eager vs. Lazy Recognition. SketchVis uses eager recognition [1]. This means that a data attribute could be recognized from a single letter if no other data attributes start with that letter. This can have advantages in quick recognition that might not require the full label to be written. However, we noticed that participants would often pause between each letter to see if the writing was recognized. If not, they would add another letter, wait, and repeat this process. After each letter, the data displayed on the chart might change, possibly distracting our participants, though sometimes illustrating alternatives. Further, eager recognition on every stroke can lead to performance degradation, which would need to be addressed.

Rapid Data Exploration through Charting. Traditionally, novices have had to put a significant amount of effort into generating a meaningful chart. This has led to common practices of planning charts and visualizations before actually creating charts. By lowering the barrier to creating charts on a whiteboard display, SketchVis allows people to approach the visualization of data sets as an exploration, developing their understanding of their data as they plot a series of charts. Some participants found the speed and direct nature of our system's sketch interaction as a great boon. For example, one participant, self-identified as a novice to charting and data exploration, compared SketchVis to chart generation tools she had used before, saying, "I don't spend a lot of time working with graphs, and [SketchVis] feels like I can get my head around it better."

Support for More Chart Types. Our participants made significant progress in analyzing their data sets with only the two chart types, scatter plot and bar chart. A much more

capable system that supports more chart types (e.g., pie charts, line charts, node-link diagrams) will enable richer analysis covering a wider range of data sets.

CONCLUSION

In this paper, we have described the design and realization of SketchVis, a sketch-based proof-of-concept application that leverages hand-drawn input for exploring data through simple charts. Scatter plots and bar charts created with SketchVis retain the traditional benefits of sketch-based creation and manipulation of whiteboards, yet these charts move beyond simple indications of sketched trends by displaying real data alongside free form strokes. Our iterative design demonstrates the potential for supporting interactive data exploration through a simple, easy-to-learn sketch-based interface. We observed people uncover insights into their data during the process of interface discovery, most of which occurred without explicit training.

REFERENCES

1. Blostein, D., Lank, E., Rose, A., Zanibbi, R. User interfaces for on-line diagram recognition. *Graphics Recognition: Algorithms and Applications. LNCS 2390*, Springer-Verlag, 2002, pp. 92–103.
2. Buxton, W. *Sketching user experiences: getting the design right and the right design*. Morgan Kaufmann, 2007.
3. Chao, W.O., Munzner, T., van de Panne, M. Poster: Rapid Pen-Centric Authoring of Improvisational Visualizations with NapkinVis. *Posters Compendium InfoVis '10*, 2010.
4. Holz, C., Feiner, S. Relaxed selection techniques for querying time-series graphs. *Proc. UIST '09*, 2009, pp. 213–222.
5. Hinkley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., Buxton, B. Pen + touch = new tools. *Proc. UIST '10*, 2010, pp. 27–36.
6. Landay, J.A., Myers, B.A. Interactive sketching for the early stages of user interface design. *Proc. CHI '95*, 1995, pp. 43–50.
7. Leitner, J., Rendl, C., Perteneder, F., Gokcezade, A., Seifried, T., Haller, M., Zeleznik, R., Bragdon, A. Nice formula editor. *Proc. SIGGRAPH '10*, 2010, pp. 55:1–55:1.
8. Stewart, R., Schiraefel, M. Graph sketcher: extending illustration to quantitative graphs. *Proc. CHI '09*, 2009, pp. 1113–1116.
9. Wattenberg, M. Sketching a graph to query a time-series database. *Ext. Abs. CHI '01*, 2001, pp. 381–382.
10. Wong, Y.Y. Rough and ready prototypes: lessons from graphic design. *Ext. Abs. CHI '92*, 1992, pp. 83–84.