# SketchInsight: Natural Data Exploration on Interactive Whiteboards Leveraging Pen and Touch Interaction

Bongshin Lee, Greg Smith, Nathalie Henry Riche, Amy Karlson*
Microsoft Research

Sheelagh Carpendale†
University of Calgary

Figure 1. SketchInsight helps people visually explore their data, primarily by allowing them to interactively create and manipulate charts with pen and touch interaction.

**Abstract**

In this work, we advance research efforts in combining the casual sketching approach of whiteboards with the machine's computing power. We present SketchInsight, a system that applies the familiar and collaborative features of a whiteboard interface to the accurate data exploration capabilities of interactive visualizations. SketchInsight enables data analysis with more fluid interaction, allowing people to visually explore their data by drawing simple charts and directly manipulating them. In addition, we report results from a qualitative study conducted to evaluate user experience in exploring data with SketchInsight, expanding our understanding on how people use a pen- and touch-enabled digital whiteboard for data exploration. We also discuss the challenges in building a working system that supports data analytic capabilities with pen and touch interaction and freeform annotation.

**Keywords**: Pen and touch; Interaction; Visualization; Data exploration; Whiteboard.

**Index Terms**: H.5.2 [Information interfaces and presentation (e.g., HCI)]: User Interfaces.

## 1 Introduction

The use of whiteboards among information workers is ubiquitous [36]. Ideas, problem-solving, and planning are often initiated on whiteboards. However, despite its familiarity and ease of use, the traditional whiteboard is fundamentally limited by its passive nature: all content must be drawn manually. If the content of the issue to be tackled contains data, it must be tediously sketched by hand, often losing detail and accuracy in the process. Furthermore, drawn data loses flexibility: it has a fixed representation and does not afford even simple data derivations; calculations such as cardinality or arithmetic mean can be difficult to estimate with accuracy, but are vital tools in the initial steps of data analysis. In all, the limitations of human memory and dexterity make data-rich problem solving on a whiteboard extremely challenging.

The alternative to the effort of including accurate data is to make charts with the 'gist.' Here whiteboard problem-solving may be done with partial or estimated charts. A quickly sketched graph outline is easy to do but is not very accurate and could even be misleading, perhaps inappropriately influencing decisions [36].

The challenge is to have the best of both worlds. To have the ability to sketch and re-sketch data charts and graphs during a problem-solving discussion, but to be able to rely on the computer to fill in the data details, to make the data plots accurate, and to provide computed meta-data. Our goal is to close the gap between how people use whiteboards with freeform sketching while thinking, and how they manipulate their data on the computer.

The advantages of sketch-based interaction and computer-aided charting are complementary. Based on this, several research projects [6][9][37][39] have recently started to explore how to bridge the gap between whiteboard and machine, combining the casual sketching interaction of whiteboards with the availability of computational power. To further these research efforts, we designed and developed SketchInsight (Figure 1), a system that applies the familiar and collaborative features of a whiteboard interface to the data exploration capabilities of interactive visualization. SketchInsight supports visual exploration of data through interactive creation and manipulation of simple charts

*e-mail: {bongshin, gregsmi, nath}@microsoft.com, akkarlson@live.com
†e-mail: sheelagh@ucalgary.ca

IEEE Pacific Visualization Symposium 2015
14–17 April, Hangzhou, China
978-1-4673-6879-7/15/$31.00 ©2015 IEEE

with pen and touch interaction. Specifically, our main contributions are:

- Design and implementation of SketchInsight, a system that integrates pen and touch interaction with computer supported data analysis, allowing people to explore their data by creating and manipulating simple charts with pen and touch interaction.
- Elaboration on the design decisions we made to create a system with gesture recognition compared to the theoretical design goals presented in Walny et al.'s Wizard of Oz study [37].
- A qualitative user study, helping us better understand the benefits and challenges of combining pen and touch interaction with the advantages of computer-supported data analysis.

## 2 Related Work

To enable people to focus on their tasks instead of the interface, researchers have been trying to reduce the gap between people's intent and the execution of their intent. Elmqvist et al. use illustrative examples to discuss "fluid" interaction in Information Visualization (InfoVis), advocating more focus on interaction design [13]. Similarly, Lee et al. reflect on the role of more "natural" interaction for InfoVis, promoting the application of a new paradigm called post-WIMP interfaces or Natural User Interfaces (NUIs) for InfoVis [25]. In this section, we review some of the NUI interactions that could be leveraged for data exploration and visualization.

### 2.1 Sketch-Based Interaction

Sketch-based interaction goes back to the 1960s, when Sutherland first introduced the Sketchpad concept [33]. With the rapid technology advances in recent years, there has been extensive research in exploring sketch-based interaction in a wide range of applications. Examples include 3D graphics modelling [1][18], user interface prototyping [23][28], animation and texture creation [10][21][22], and educational applications [3][24].

Particularly relevant for InfoVis, sketch-based interaction was employed to support sketch-based data queries. For example, QuerySketch [38] and QueryLines [17] use drawing interaction to ease the creation of a target pattern when specifying queries for time-series data. Recently, the InfoVis community has started to employ sketch-based interaction for data exploration. For example, NapkinVis uses a set of simple pen gestures to enable fast visualization creation [9]. SketchVis leverages hand-drawn sketch input to explore data in simple charts without relying on menus or widgets [6]. SketchVis is a proof-of-concept system that uses purely stroke-based interactions and supports only two chart types, bar chart and scatter plot. Even though multiple charts can be drawn on a canvas, they are not linked and cannot be moved, resized, copied, or merged. SketchInsight takes the next step in this research direction by incorporating sketch with multi-touch, exploring the benefits and challenges of combining pen and touch interaction for data exploration.

### 2.2 Multi-touch Interaction

With the prevalence of multi-touch devices, researchers have been actively exploring multi-touch interactions for InfoVis. Examples include the investigation of touch interaction techniques for InfoVis on tabletops [32][34][35], more fluid interactions for node-link graph visualizations [12][31], and expanding and improving the interaction with visualization with tablets. For example, TouchWave extends stacked graphs by alleviating their main issues through multi-touch interactions [2]. Kinetica employs physics-based affordances with multi-touch interaction to augment multivariate data visualization [29]. Ramik and Stasko explore touch interactions for interactive scatterplot visualization [30]. Drucker et al. propose a set of simple gestural interfaces for selecting, filtering, and sorting data, and compare it against a more WIMP-based one [11]. Transmogrifier leverages multi-touch interaction techniques to support efficient specification of shapes, enabling data-centric flexible analysis [5].

More generally, Isenberg et al. discuss how multi-touch interactions could be applied to visualizations [19]. Hinrichs and Carpendale observe touch and gesture interaction in the wild and suggest multi-faceted interactions [16]. In addition, Isenberg et al. discuss the challenges and research opportunities in supporting visualization on touch-enabled devices from technical, design, and social aspects [20].

### 2.3 Pen and Touch Interaction

Researchers have been exploring potential of combining pen and touch interaction to build more effective interfaces and to create new interaction experience. Brandl et al. show that combining pen and touch input is faster and more accurate than touch-only or pen-only input [4]. Frisch et al. investigate a set of pen and touch gestures for editing (rather than exploring) node-link diagrams [14]. Hinckley et al. advocate an approach where pens are used for writing and marking, touch is used for manipulating elements, and that combining pen and touch provides additional tools [15]. By recognizing handwritten formulas, the NiCE Formula Editor provides in situ computation, expanding the notion of mathematical sketching by supporting both pen and touch interactions on an interactive whiteboard [27].

Several recent projects have demonstrated the power of pen and touch interaction for InfoVis. PanoramicData lets people rapidly search structured datasets using visual pen and touch queries [39]. SketchStory leverages the expressive, freeform nature of sketch for data presentation through the dynamic creation of organic data charts [26]. Walny et al. investigate the role of pen and touch interaction for data exploration on interactive whiteboards through a Wizard of Oz (WOz) study, showing that people can transfer knowledge from previous interactions [37]. They also present design implications to inform the design of pen- and touch-enabled data exploration systems. Inspired by these projects, we designed and developed SketchInsight, a system that offers more fluid interactions for data exploration by leveraging pen and touch interaction. It also introduces multi-chart manipulation through two-handed multi-touch interaction. We describe the decisions made to create a working system compared to the theoretical guidelines from the WOz study.

## 3 SketchInsight

### 3.1 Design Rationale

SketchInsight is designed to help people visually explore data, primarily by allowing them to interactively create and manipulate charts with pen and touch interaction. Its design is mainly inspired by Walny et al.'s WOz study [37]. For example, SketchInsight fully realizes two of their four design goals; DG2 (Design Goal 2)—make manipulation as direct as possible, and DG3—minimize explicit mode-switching. We discuss why and how we adjusted some of their initial goals, and describe additional design rationale in supporting more natural pen and touch interactions.

#### 3.1.1 Support Fluid Chart Creation and Manipulation

Walny et al.'s WOz study [37] advocates an inspiring design goal called What You Draw Is What You Get (WYDIWYG—DG1). Since they assumed the "perfect" recognizer (i.e., the wizard) in their study, they could infer what participants intended when marks were drawn on the whiteboard. While this concept provides


Figure 2. SketchInsight supports eight simple pen gestures. (a)-(b) scribble and ‘X’ for erasure, (c)-(e) ‘L’, ‘O’, and inverted L for X/Y chart, pie chart, and map creation, and (f)-(h) bar shape, jagged line, and four dots for changing a chart type to a bar chart, line chart, and scatterplot.

a compelling chart creation experience, we reasoned that it may not hold up well in practice, since a real-world recognizer will have a non-zero error rate, potentially leading to misrecognitions and frustration. Thus, we opted to transform WYDIWYG into a more fluid interaction goal, in which people could create and manipulate charts with high probability of recognition. When high recognition accuracy is difficult to ensure, such as with chart type changes; SketchInsight provides alternative interaction options such as a radial popup menu.

#### 3.1.2 Support Good Default Behaviors with Controlled Flexibility

Creating a chart requires specification of several settings including a chart type, corresponding data columns, and optional grouping and aggregation functions. To respond with sensible defaults where possible, SketchInsight uses a set of simple heuristics. For example, when the x-axis is nominal and the y-axis is numeric SketchInsight initially selects a bar chart and applies the *average* default aggregation function to avoid showing hundreds of data points.

Walny et al.’s WOz study also advocates supporting multiple interaction paths when possible. For instance, their design allowed people to write the axis labels and then draw the axes, or draw and specify one axis at a time. Again, considering the accuracy of existing recognizers (especially when allowing for freeform annotation), we opted for designated areas to specify data columns, which are available only after the chart is instantiated. SketchInsight still supports the X/Y chart creation with one or two strokes and people have some flexibility to specify chart settings.

#### 3.1.3 Keep the Set of Gestures Small, Simple, and Logical

Pen and touch gestures enable people to trigger specific system responses without using menus or buttons. Well-designed gestures can be natural, fluid, and even fun. It is tempting to design and support many gestures to cover many features. However, these gestures can quickly become complex, difficult to learn and remember, and difficult for the system to correctly recognize. Therefore, we decided to keep the gesture set simple and small, and we designed the gesture set to be a logical one, mimicking the graphical primitives of the chart elements to be produced.

### 3.2 Interacting with SketchInsight

We first explain the basic mechanics of SketchInsight interaction. We then present two example scenarios using SketchInsight: Zoey explores a global energy consumption dataset, and Charley analyzes a survey dataset regarding gender difference in software feature usage.

#### 3.2.1 Basic Mechanics of Interaction

SketchInsight avoids having two explicit modes for sketching and manipulation by using the pen for drawing charts or annotations, and touch for manipulating them. The pen is used to make digital ink strokes on the canvas. If any drawn stroke is recognized as one of the command gestures (Figure 2), the ink is either replaced by the corresponding system-drawn element (e.g., chart creation) or removed from the canvas after invoking the command (e.g., chart type change). Otherwise, the ink is left as-is on the canvas as annotation, and can be erased by a scribble gesture (Figure 2a). Chart objects can be manipulated (e.g., move, resize, copy) with touch interaction.

SketchInsight currently recognizes three types of chart gesture: X/Y charts use an ‘L’ shape (Figure 2c), pie charts use a circle (Figure 2d), and maps use an inverted L (Figure 2e). Scribbling (Figure 2a) on the chart type name in the upper left corner will delete the chart. Since this target region is not always easily accessible SketchInsight supports additional gesture for chart erasure, drawing an ‘X’ (Figure 2b) across the chart. SketchInsight uses handwriting text recognition to suggest column names for display.

#### 3.2.2 Scenario 1: Global Energy Consumption

Zoey first wants to see how much energy we have consumed over the years. She *draws* an L shape to create a chart with x and y axes; the size and location of the L shape determines the physical dimensions of the resulting chart (Figure 3a). As Zoey *writes* a “y” along the X axis (one of the designated areas), SketchInsight attempts to recognize the text and find matching column names in the dataset. When SketchInsight displays a popup menu with the names of the matching columns, Zoey *taps* the “Year” column from the popup to assign the column to the chart (Figure 3b). Once Zoey does the same for the Y axis to assign “Consumption,” SketchInsight populates the chart according to its heuristics, with average consumption over the years (Figure 3c). Zoey can see that average energy consumption has more than doubled in 25 years.

Now she decides to compare energy consumption trends by region. To do this, Zoey switches to a line chart by *drawing* a jagged line (Figure 2g and Figure 3d). She writes “region” in the “aggregate by” box and selects the average option. (SketchInsight supports average, sum, min, and max.) SketchInsight automatically colors the lines to distinguish them by assigning the “Region” column to the “color by” box. In addition, SketchInsight automatically provides a visual legend with map regions colored as their associated data values (Figure 3e).

Noticing that Asia-Pacific and North America are the top two consumers in recent years, Zoey focuses on these regions. She filters out other regions by *tapping* the map, and SketchInsight updates both the line chart and the map (Figure 3f). Zoey clearly sees that Asia-Pacific began to consume more energy than North America in 2005.

#### 3.2.3 Scenario 2: Survey for Gender Differences in Software Feature Usage

Charley first wants to see the distribution of genders of the survey respondents. He *draws* a circle to create a pie chart. He specifies the “Count” function in the upper title area and the “Gender” column in the center area of the chart to group the data items by gender (Figure 4a-left). Once SketchInsight completes the chart automatically, Charley notices that two respondents declined to specify their genders. He *draws* an inverted L to invoke a gender map (Figure 4a-right). Charley then filters out the two respondents by *tapping* on the corresponding icon on the gender map (or the corresponding wedge of the pie chart).

Charley wants to see how gender may have impacted responses to two different questions: how much respondents prefer to use a wizard and if they enjoy piloting. He first creates a bar chart showing the average Likert score for the wizard question by gender (Figure 4b-left). Then he creates another chart by copying the first one. When he *pins* the chart with one finger, a copy icon appears in the upper right corner of the chart. He *drags* the icon toward the right to "tear away" a copy with another finger. On the new copy, he changes the Y axis to be the "Piloting" column by *writing* over the existing column name (Figure 4b-right).

Charley notices that the ranges for the y-axes are different between the two charts, making it difficult to compare them. To combine the data into one chart, he *drags* the two charts toward each other. Combining charts is only possible if both charts have

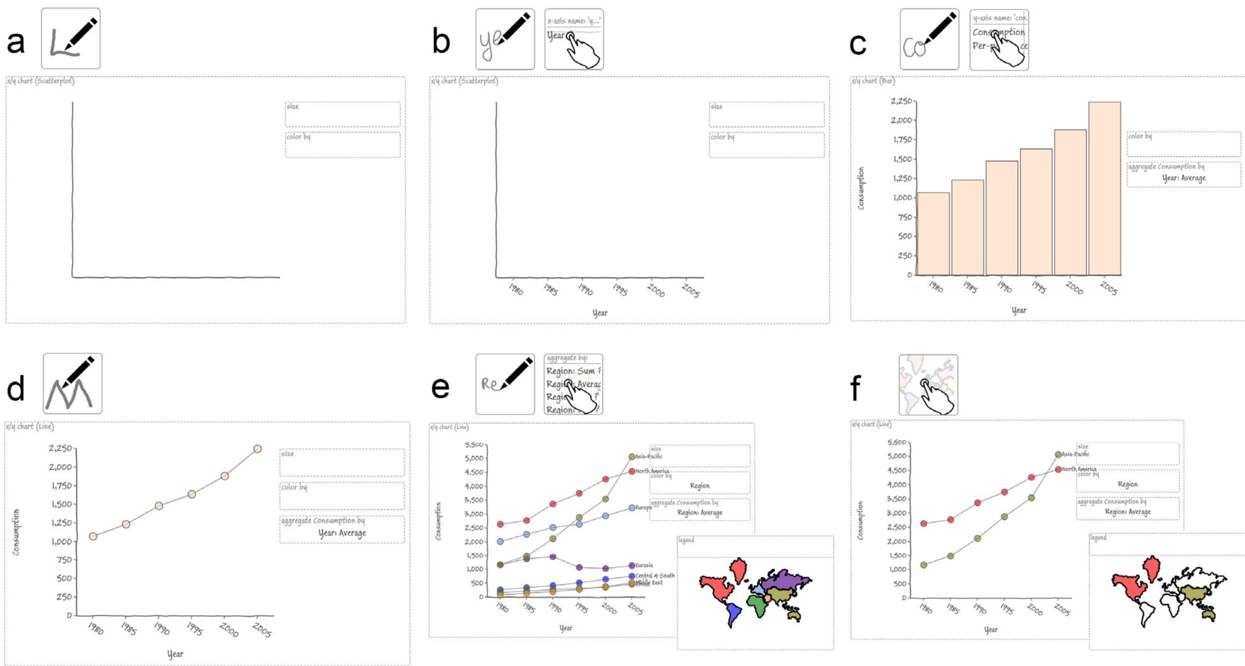

Figure 3. Exploring a global energy consumption data with SketchInsight. The icons shown in the upper-left boxes of each image indicate the type of interaction, but not the location of where it is performed.

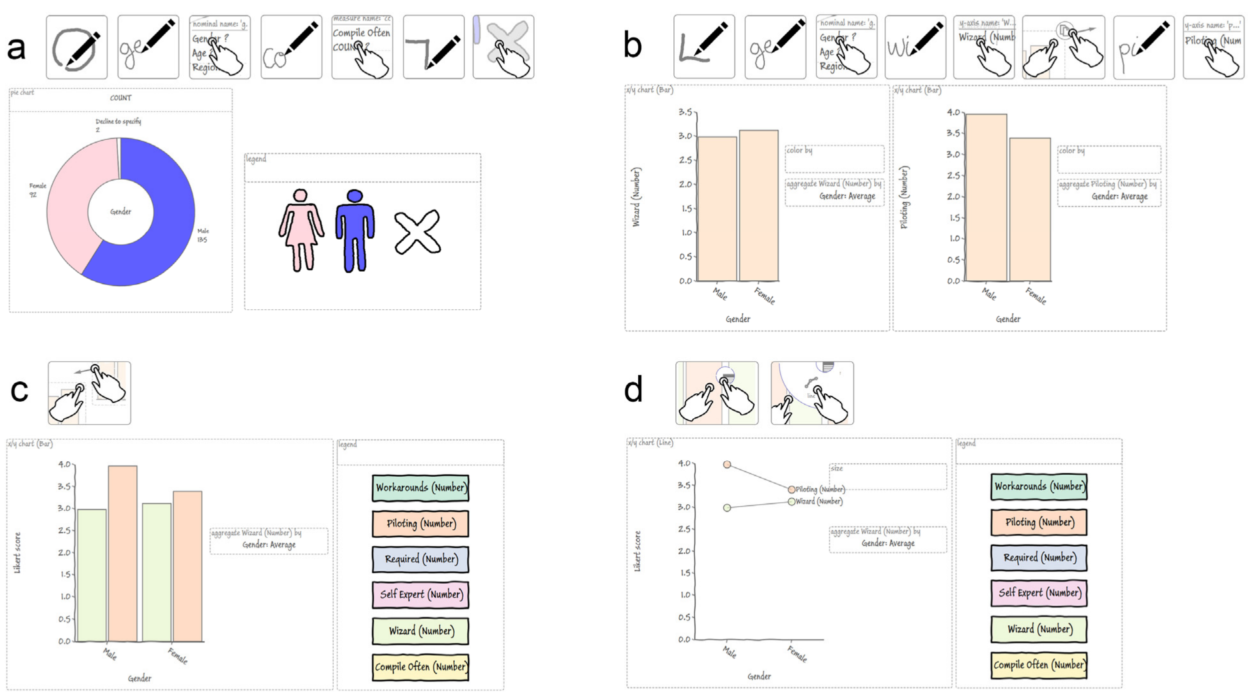

Figure 4. Exploring a survey data with SketchInsight. The icons shown in the upper-left boxes of each image indicate the type of interaction, but not the location of where it is performed.

the same X axis and Y axis of the same type; once the two charts significantly overlap, SketchInsight provides feedback indicating that the two charts can indeed be combined. Once Charley *lifts* both fingers, SketchInsight merges the two charts into one. SketchInsight color-codes the columns corresponding to each question and provides a legend to show which color is associated with which column (Figure 4c). Charley can also undo the combine operation following the same steps he used to copy them. As he *pins* the merged chart with one finger, a split icon appears (replacing the copy icon present in unmerged charts). Charley can *drag* the icon to the right with another finger to split the charts.

To switch to a line chart, Charley *pins* the chart with a finger and then *taps* the chart type icon that appeared in the center of the chart. When a radial menu appears, he selects the line chart item (Figure 4d). Now Charley can see that patterns differ between genders for these two questions.

### 3.3 Implementation Details

SketchInsight was implemented in C# using WPF as a standalone application. An input layer was written to translate basic Windows input events (Mouse, Touch, or Stylus) into unified Pointer events to mask differences in system input capabilities and to allow emulation of input modes via software configuration (e.g., to allow mouse input for inking). Inked object recognition was implemented as a hierarchical layout of recognizer modules, where each module was asynchronously invoked for any ink strokes falling within that module's region of interest: The Tablet PC Inkassembly (IAWinFX) was used to recognize handwritten text in text-input regions such as axis label locations; an IStraw implementation [8] was used to partition individual strokes into line segments in shape and gesture drawing regions; and various hand-crafted recognizers such as circle or three-sided-box were built on top of the basic segmentation primitives.

SketchInsight datasets are simple tables, where each row is a data item, and the column values represent the attributes of the data items. SketchInsight supports two basic types: numeric and nominal. Numeric columns contain numbers on a continuous scale that can potentially be aggregated with mathematical operators (e.g., sum). Nominal columns contain discrete string values, such as names or enumerations that can serve as categories for grouping data items with like values. Each SketchInsight dataset is stored as a set of two files: a tab-delimited data file and an XML-based data description file with column specifications.

## 4 Observational Study

### 4.1 Participants and Study Setup

We conducted a qualitative study to evaluate the experience of exploring data through the new ways of interaction. Our goal was to understand how people interact with SketchInsight, to learn more about its strengths and weaknesses, and to identify any major usability issues.

### 4.2 Participants and Study Setup

We recruited twelve professionals (5 female, average age = 38.1) who reported analyzing survey data at least several times a year. All participants were either current or former employees of a large software company, and were provided with a $20 cafeteria coupon as compensation.

Study participants were scheduled in pairs so that they could perform tasks as a team. By providing participants with a partner, we hoped to foster more natural interactions and discussions, and to promote a discovery mindset.

The study was conducted in a small conference room using two side-by-side 55 inch PPI displays running the SketchInsight software at 1920x1080 resolution (Figure 1). During the training phase, both participants practiced on their own using separate displays and pens. During the task phase, only one PPI (with one pen) was used for the task and the other PPI was used to show the dataset. We logged interaction such as chart creation, chart move, filter, etc. with a time stamp. We also captured screencasts of SketchInsight, and simultaneously captured video and audio of the participants interacting with SketchInsight.

### 4.3 Datasets and Tasks

To increase task validity, we borrowed two datasets from previous research: the world population dataset used in Walny et al.'s study [37], containing 9 columns and 66 rows; and the survey results conducted about software engineers, investigating gender difference [7]. Due to missing data, we discarded 13 rows, resulting in 229 rows. We excluded some questions (e.g., open-ended text) whose answers cannot be handled by SketchInsight, resulting in 31 columns. In addition, we prepared two additional datasets for training. We borrowed the energy consumption dataset from previous research [26] because its structure is similar to the population dataset. We simulated a dataset to mimic the structure of the survey dataset.

We drew six tasks from the two datasets; three tasks for each dataset (Table 1). They varied in difficulty so that we might observe differences in how well the system supported simple chart-creation tasks (e.g., T1, T4) versus those that are more exploratory in nature (e.g., T3, T6). Furthermore, the tasks were designed to encourage participants to use various system features, including chart merging (e.g., T2, T5), displaying data as a series (e.g., T6), and filtering (e.g., T3, T5). Note that for tasks T4~T6, participants were also provided with the original survey questions so that they could better understand the intention of the questions.

| Task ID | Task Text | Difficulty Level | Dataset |
|---|---|---|---|
| T1 | Show the **Birth Rate** trend by **Year** | 1 | World Population |
| T2 | Compare population changes across the years by gender (**Male Population** vs. **Female Population**). | 2 | |
| T3 | Explore correlation between **Birth Rate** and **Death Rate** for "Australia/New Zealand." | 3 | |
| T4 | Create charts to convey the demographics of survey participants:<br>a) The participant distribution by **Gender**<br>b) The participant distribution by **Age** | 1 | Survey |
| T5 | a) Create a chart to compare how the use of 3 different software engineering techniques (**Wizard**, **Piloting**, and **Workarounds**) varied by **Gender**.<br>b) Display only Male and Female (not Decline to specify). | 2 | |
| T6 | a) Explore the relationship between visual studio experience (**IDE Experience**) and whether they consider themselves a visual studio expert (**Self Expert**).<br>b) Are there any gender differences among the responses? | 3 | |

Table 1. Study tasks and their associated difficulty levels (3 is hardest).

### 4.4 Procedure

A facilitator introduced the structure of the study and then led the pair through a detailed tutorial on the features of SketchInsight. Topics covered included general ones such as the roles of pen and touch, and system-specific ones such as chart creation, copying charts, merging charts, data filtering, etc. In total, participants were introduced to 43 distinct concepts that we deemed fundamental to allowing participants to perform tasks optimally. The SketchInsight concepts were taught in six sets, after each of these the participants were given the opportunity to practice at separate PPI displays until they felt comfortable with the concepts learned. The tutorial lasted about an hour.

Following a short break, the study was begun. Tasks were presented in the same order across all sessions, T1 through T6, on a sheet of paper. Before tasks T1 and T4, the administrator introduced the dataset that would be used for the following three tasks. In addition to a verbal description of the dataset, the participants were provided with a printout of the data table. Participants were given no time constrains in completing tasks, since we were interested in natural use of the system. However, we tracked total task time from the moment the task was given to the time that the administrator deemed the task complete, which included chart creation, verbal interpretation of the chart, and requests by the administrator to explore additional chart representations. For each task we also noted the time that the “answer” chart was first created by the team; for questions with two parts, we considered the “answer” chart to be the answer for the second part. The study concluded with a paper-based questionnaire about the participants’ experience with the system, and if time allowed, a verbal request for general comments. The experiment took about two hours.

| Task | Time to “Answer” Chart | Total Time | No. of Hints |
|---|---|---|---|
| T1 | 28 | 124 | 0 |
| T2 | 114 | 248 | 0 |
| T3 | 124 | 248 | 0.17 |
| T4 | 105 | 157 | 0 |
| T5 | 203 | 352 | 0 |
| T6[1] | 582 | 756 | 2.8 |

Table 2. Average task time (seconds) and number of hints. [1]**One of the teams ran out of time and did not perform T6.**

| Interaction | Total Count | Success Rate |
|---|---|---|
| Pen | 749 (56.7%) | 0.83 |
| Touch | 572 (43.3%) | 0.95 |
| All | 1321 | 0.88 |

Table 3. Pen vs. Touch Interaction.

| Pen Gesture | Total Count | Success Rate |
|---|---|---|
| Scribble | 233 | 0.75 |
| CrossOut (X) | 18 | 0.56 |
| XYChart | 63 | 0.95 |
| Pie | 4 | 1.00 |
| Map | 34 | 0.68 |
| Bar | 17 | 0.29 |
| Line | 12 | 0.58 |
| Scatterplot | 7 | 1.00 |
| All | 388 | 0.75 |

| Touch Gesture | Total Count | Success Rate |
|---|---|---|
| Radial Menu | 43 | 0.93 |
| Chart Copy | 36 | 0.81 |
| Chart Combine | 23 | 1.00 |
| Chart Split | 1 | 1.00 |
| Resize & Move | 113 | 1.00 |
| Move | 141 | 1.00 |
| All | 357 | 0.97 |

Table 4. Pen (left) and Touch (right) Gesture Recognition.

### 4.5 Study Results

#### 4.5.1 Task Time and Errors

Overall, excepting T6, participating teams were able to successfully complete the tasks with few hints from the administrator (Table 2). We considered a task “correct” if one or more charts could be interpreted meaningfully.

Across all tasks, participants spent about half of each task session creating the “answer” chart(s) and the other half exploring, interpreting and discussing the results. In general, the average total task time across tasks matched our expected relative difficulties of the tasks. Teams had the most difficulty with T6, which took the longest to complete and required the most hints from the administrator. Furthermore, none of the teams created the chart that we had envisioned as the “answer” chart. However, four of the five teams that completed the task did succeed in creating a bar chart with gender as a series.

#### 4.5.2 Gesture Analysis

Since the system logs only the interactions it can recognize, we manually logged two types of failed interaction attempts by analyzing the captured screencasts and videos: incorrectly recognized gestures (e.g., a chart L shape with insufficiently sharp corners); and unsupported or invalid interactions (e.g., touching the “group by” box to invoke a context menu or trying to combine two charts that cannot be merged). The former can be considered a system error (i.e., misrecognition) and the latter a human error (i.e., misunderstanding). Among the total of 1321 interactions, 62 (4.7%) were human errors.

749 (56.7%) were pen interactions and 572 (43.3%) were touch (Table 3). Pen interactions had a lower recognition success (83%) than touch (95%). Note that we did not log the interaction for writing column names because the unit and success of action was ambiguous; sometimes people continued writing after erasing a part of the written word and repeated this multiple times before they select an item.

Among the 749 pen interactions, 388 (52%) were gestures to create and manipulate charts (Table 4-left). Among the 572 touch interactions, 357 (62%) were gestures to manipulate charts (Table 4-right). Note that we did not count a tap to select a menu item or filter an item as a gesture because it does not involve any gesture recognition. 254 (99.6%) out of 255 menu selection was done with pen interaction, whereas 49 (23.8%) out of 206 filter selection was done with pen.

#### 4.5.3 Usability Issues and Subjective Responses

Major usability issues were around poor pen gesture recognition and lack of feedback for failed attempts. Only three gestures (XY chart, Pie chart, and Scatterplot) were recognized with high accuracy (>= 95%) and others were low; two of them were attempted less than 10 times. Scribble gesture is the biggest issue in that it was attempted 233 times yet recognition rate was only about 75%. For the chart type change gestures, people could resort to the multi-touch radial menu.

SketchInsight was designed to prevent invalid chart settings and to respond with defaults that make sense, where possible. For example, when both axes are numerical data, SketchInsight did not allow people to change the chart type to be a line chart. Even though SketchInsight showed an error message on invalid requests, it was not salient enough and did not stay long enough.

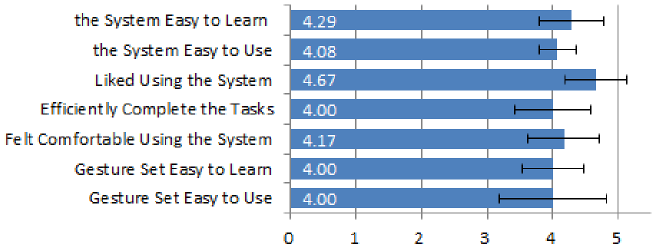


Figure 5. Average responses to subjective satisfaction questions. Error bars represent standard deviation.

Responses to the post-study questionnaire revealed that participants overwhelmingly appreciated the SketchInsight system. The average response to all subjective satisfaction questions was 4 or higher on a 5-point scale (1=Strongly Disagree, 5=Strongly Agree, Figure 5).

Participants' freeform responses highlight that they were most impressed with how quickly (10 participants) and easily (9) they were able to create and manipulate charts; seven participants specifically referred to the exploratory nature of the system. Many participants enjoyed the work style afforded by the large touch screen, particularly its support of touch interaction (8) which many associated directly with enabling faster insights, but they also valued the large screen for supporting collaboration (3), and found the form factor and interaction style a welcome alternative to traditional analysis using a desktop with keyboard and mouse (3). Consistent with these responses, every participant reported that they would be interested in using the system in their current work (M=4.50, SD=0.50 on a 5-point scale, 1=Not at All Interested and 5=Very Interested).

There was less agreement across participants regarding what they did not like about SketchInsight. Six participants were distracted by gesture recognition errors, specifically the scribble (4 participants) and line chart (4) gestures. Four participants would have liked to have a second pen to support more parallel collaboration. Three participants found it challenging to remember the range of functionality of the system and would have liked a way to look up what features were supported.

## 5 Discussion and Future Work

### 5.1 "Fluid" Exploration with Pen & Touch Interaction

When overhead is high in changing the charts, people tend to plan the charts before actually creating them. With SketchInsight, participants seem to think through interactions; overall, they interacted with data very actively, without many long pauses. It appeared that SketchInsight did lower the burden in creating and manipulating charts through fluid pen and touch interaction, allowing people to iteratively explore their data as they plot a series of charts. On the other hand, we also note that fluid pen and touch interaction is a double–edged sword. We observed that people sometimes mindlessly fill in the blanks or repeat the same set of interactions without reflecting on the results of their interactions. For example, participants repeatedly toggled a filter on and off even though the interaction would not give the final chart they needed to create.

### 5.2 Gap between the Wizard of Oz Study and Real System

Implementing SketchInsight as a real system allowed us to explore the limitations and inconsistencies inherent in the initial goal of What You Draw Is What You Get (WYDIWYG) from Walny et al.'s Wizard of Oz study [37]. The basic WYDIWYG chart creation gestures worked well, and despite issues with poor recognition of the chart-type gestures, they seemed to be easily remembered. But the desire to support freeform annotation, particularly in a collaborative or presentation format, conflicted directly with some of the WYDIWYG examples described in the paper: in particular, judging by much simpler recognition issues in other parts of the system, attempting to allow column specification by drawing arbitrary lines "connecting" chart elements with handwritten text would have had, in practice, an extremely low overall success rate. We believe that allowing column specification by writing in specific spaces is not highly restrictive yet reduces frustrating false positives and negatives.

This relates to one fundamental difficulty in attempting to allow inking for both command gestures and freeform annotation. It is extremely difficult to perfectly disambiguate between the two because there is no simple command gesture that could not, in theory, be a useful pure annotation graphic in some given user scenario. For example, even a "perfect" circle command recognizer will fail whenever a person simply intends to draw a circle as annotation. Furthermore, the SketchInsight command set was deliberately designed to mimic the simple graphical primitives of the chart elements a person wants to produce, so creating elaborate, visually unique, easily-differentiable command gestures simply to reduce the false positive recognition rate defeats the purpose. Disambiguation can always be achieved by introducing additional modes of interaction, but adding more modal operations conflicts with the SketchInsight design goal of minimizing explicit mode-switching for fluid interaction.

### 5.3 Whiteboards for Collaborative Data Exploration

Whiteboards intrinsically afford multi-user scenarios, inviting collaboration. We envision two types of collaboration. First, a presenter reports his or her analysis to the audience and interacts with the system to answer any questions. SketchInsight could support this collaborative discussion performed after initial individual exploration but before final presentation. Second, two or more people actively explore data together. The SketchInsight approach could be extended to support this collaborative data exploration. Enabling the simultaneous use of multiple pens would be a good start. We also need to be able to differentiate the individual people when they interact with the system.

### 5.4 Lack of Feedback and Study Limitations

One of the major usability issues of SketchInsight was lack of feedback for failed attempts. For invalid requests, to fix this issue we could provide a more readable, salient error message longer. For misrecognition, we could help people offer better text input to SketchInsight, which does not clearly mark the text-input regions to avoid clutter. For example, when people start writing (close to the text-input region) we could highlight the valid region (in an unobtrusive way) until they finish writing. In addition, we could show the actual stroke from the bounding box along with the recognition result as a way to show an input to the text recognizer.

In addition to introducing a radically different system with a slew of functionality, the datasets and tasks we presented did not necessarily reflect the data exploration process to which the participants are accustomed. Furthermore, the experiment setup where an administrator is watching over the shoulder may have imposed a high level of stress. We suspect this pressure could have caused some of the mindless exploration, trying to do something without being idle. A longer term deployment study where people could use the system with their own data in a normal work environment would shed more light on how people would use pen and touch interaction for data exploration.

## 6 CONCLUSION

Inspired by the the power of pen and touch interaction for InfoVis as shown in recent research and the dominant use of whiteboards as a thinking medium, we designed and developed SketchInsight to combine pen and touch for data exploration on interactive whiteboards. SketchInsight pushes the boundary of fluid data exploration, introducing new experiences. We discussed design rationales and the main challenges in developing a system that supports more fluid pen and touch interaction with freeform annotation. Through a qualitative study with a working system, we learned how people react to and interact with such a system, achieving a better understanding of the benefits and challenges of the combining pen and touch interaction with computer supported data analysis.

## REFERENCES


[1] Bae, S., Balakrishnan, R., & Singh, K. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. *Proc. UIST*, 2009, 151–160.

[2] Baur, D., Lee, B., & Carpendale, S. TouchWave: kinetic multi-touch manipulation for hierarchical stacked graphs. *Proc. ITS*, 2012, 255–264.

[3] Bott, J.N., LaViola, J., & Joseph, J. A pen-based tool for visualizing vector mathematics. *Proc. SBIM*, 2010, 103–110.

[4] Brandl, P., Forlines, C., Wigdor, D., Haller, M., & Shen, C. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. *Proc. AVI*, 2008, 154–161.

[5] Brosz, J., Nacenta, M.A., Pusch, R., Carpendale, S., & Hurter, C. Transmogrification: causal manipulation of visualizations. *Proc. UIST*, 2013, 97–106.

[6] Browne, J., Lee, B., Carpendale, S., Riche, N., & Sherwood, T. Data analysis on interactive whiteboards through sketch-based interaction. *Proc. ITS*, 2011, 154–157.

[7] Burnett, M.M., Fleming, S.D., Iqbal, S.T., Venolia, G., Rajaram, V., Farooq, U., Grigoreanu, V., & Czerwinski, M. Gender differences and programming environments: across programming populations. *Proc. ESEM*, 2010.

[8] Cheema, S., Gulwani, S., & LaViola, J.J. QuickDraw: improving drawing experience for geometric diagrams. *Proc. CHI*, 2012, 1037-1064.

[9] Chao, W.O., Munzner, T., & van de Panne, M. Napkinvis: Rapid pen-centric authoring of improvisational visualizations. *Posters InfoVis*, 2010.

[10] Davis, R.C., Colwell, B., & Landay, J.A. K-sketch: a 'kinetic' sketch pad for novice animators. *Proc. CHI*, 2008, 413–422.

[11] Drucker, S., Fisher, D., Sadana, R., & Herron, J. TouchViz: a case study comparing two interfaces for data analytics on tablets. *Proc. CHI*, 2013, 2301–2310.

[12] Dwyer, T., Lee, B., Fisher, D., Inkpen, K., Isenberg, P., Robertson, G., & North, C. Understanding multi-touch manipulation for surface computing, *IEEE TVCG (InfoVis 2009) 25*, 19 (2009), 961–968.

[13] Elmqvist, N., Moere, A.V., Jetter, H.-C., Cernea, D., Reiterer, H., & Jankun-Kelly, T.J. Fluid interaction for information visualization, *Information Visualization 10*, (2011), 327–340.

[14] Frisch, M., Heydekorn, J., & Dachselt, R. Investigating multi-touch and pen gestures for diagram editing on interactive surfaces. *Proc. ITS*, 2009, 149–156.

[15] Hinkley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., & Buxton, B. Pen + touch = new tools. *Proc. UIST*, 2010, 27–36.

[16] Hinrichs, U. & Carpendale, S. Gestures in the wild: studying multi-touch gesture sequences on interactive tabletop exhibits. *Proc. CHI*, 2011, 3023–3032.

[17] Holz, C. & Feiner, S. Relaxed selection techniques for querying time-series graphs. *Proc. UIST*, 2009, 213–222.

[18] Igarashi, T., Matsuoka, S., & Tanaka, H. Teddy: a sketching interface for 3D freeform design. *Proc. SIGGRAPH*, 1999, 409–416.

[19] Isenberg, P., Hinrichs, U., Hancock, M., & Carpendale, S. Digital tables for collaborative information exploration. *Proc. Tabletops*, 2010, 387–405.

[20] Isenberg, P., Isenberg, T., Hesselmann, T., Lee, B., von Zadow, U., & Tang, A. Data visualization on interactive surfaces: a research agenda from the DEXIS workshop. *IEEE CG&A 33*, 2 (2013), 15–24.

[21] Kazi, R.H., Chevalier, F., Grossman, T., Zhao, S., & Fitzmaurice, G.W. Draco: bringing life to illustrations with kinetic textures. *Proc. CHI*, 2014, 351–360.

[22] Kazi, R.H., Igarashi, T., Zhao, S., & Davis, R. Vignette: interactive texture design and manipulation with freeform gestures for pen-and-ink illustration. *Proc. CHI*, 2012, 1727–1736.

[23] Landay, J.A. & Myers, B.A. Interactive sketching for the early stages of user interface design. *Proc. CHI*, 1995, 43–50.

[24] LaViola, J.J. & Zeleznik, R.C. MathPad$^2$: A system for the creation and exploration of mathematical sketches. *ACM SIGGRAPH Courses*, 2007.

[25] Lee, B., Isenberg, P., Riche, N.H., & Carpendale, S. Beyond mouse and keyboard: expanding design considerations for information visualization interactions. *IEEE TVCG (Proc. InfoVis) 18*, 12 (2012), 2689–2698.

[26] Lee, B., Kazi, R.H., & Smith, G. SketchStory: telling more engaging stories with data through freeform sketching. *IEEE TVCG (Proc. InfoVis) 19*, 12 (2013), 2416–2425.

[27] Leitner, J., Rendl, C., Perteneder, F., Gokcezade, A., Seifried, T., Haller, M., Zeleznik, R., & Bragdon, A. Nice formula editor. *Proc. SIGGRAPH*, 2010, 55:1.

[28] Lin, J., Newman, M.W., Hong, J.I., & Landay, J.A. DENIM: an informal tool for early stage web site design. *Ext. Abs. CHI*, 2001, 205–206.

[29] Rzeszotarski, J.M. & Kittur, A. Kinetica: naturalistic multi-touch data visualization. *Proc. CHI*, 2014, 897–906.

[30] Sadana, R. & Stasko, J. Designing and Implementing an Interactive Scatterplot Visualization for a Tablet Computer. *Proc. AVI*, 2010, 265–272.

[31] Schmidt, S., Nacenta, M., Dachselt, R., & Carpendale, S. A set of multi-touch graph interaction techniques. *Proc. ITS*, 2010, 113–116.

[32] Spindler, M., Tominski, C., Schumann, H., & Dachselt, R. Tangible views for information visualization. *Proc. ITS*, 2010, 157–166.

[33] Sutherland, I.E. Sketchpad: A man-machine graphical communication system. *Proc. AFIPS Spring Joint Comp. Conf*, 1963.

[34] Vlaming, L., Collins, C., Hancock, M., Nacenta, M., Isenberg, T., & Carpendale, S. Integrating 2D mouse emulation with 3D manipulation for visualizations on a multi-touch table. *Proc. ITS*, 2010, 221–230.

[35] Voida, S., Tobiasz, M., Stromer, J., Isenberg, P., & Carpendale, S. Getting practical with interactive tabletop displays: designing for dense data, "fat fingers," diverse interactions, and face-to-face collaboration. *Proc. ITS*, 2009, 109–116.

[36] Walny, J., Carpendale, S., Riche, N.H., Venolia, G., & Fawcett, P. Visual thinking in action: visualizations as used on whiteboards. *IEEE TVCG (Proc. InfoVis) 17*, 12 (2011), 2508–2517.

[37] Walny, J., Lee, B., Johns, P., Riche, N.H., & Carpendale, S. Understanding pen and touch interaction for data exploration on interactive whiteboards. *IEEE TVCG (Proc. InfoVis) 18*, 12 (2012), 2779–2788.

[38] Wattenberg, M. Sketching a graph to query a time-series database. *Ext. Abs. CHI*, 2001, 381–382.

[39] Zgraggen, E., Zeleznik, R., & Drucker, S.M. PanoramicData: Data analysis through Pen & Touch, *IEEE TVCG (Proc. InfoVis) 20*, 12 (2014), 2112–2121.