

Understanding Pen and Touch Interaction for Data Exploration on Interactive Whiteboards

Jagoda Walny, Bongshin Lee, Paul Johns, Nathalie Henry Riche, and Sheelagh Carpendale

Abstract—Current interfaces for common information visualizations such as bar graphs, line graphs, and scatterplots usually make use of the WIMP (Windows, Icons, Menus and a Pointer) interface paradigm with its frequently discussed problems of multiple levels of indirection via cascading menus, dialog boxes, and control panels. Recent advances in interface capabilities such as the availability of pen and touch interaction challenge us to re-think this and investigate more direct access to both the visualizations and the data they portray. We conducted a Wizard of Oz study to explore applying pen and touch interaction to the creation of information visualization interfaces on interactive whiteboards without implementing a plethora of recognizers. Our wizard acted as a robust and flexible pen and touch recognizer, giving participants maximum freedom in how they interacted with the system. Based on our qualitative analysis of the interactions our participants used, we discuss our insights about pen and touch interactions in the context of learnability and the interplay between pen and touch gestures. We conclude with suggestions for designing pen and touch enabled interactive visualization interfaces.

Index Terms—Pen and touch, interaction, Wizard of Oz, whiteboard, data exploration.

1 INTRODUCTION

Pen and touch interactions have great potential to lead to more natural interactions for information visualization (InfoVis). As demonstrated by Hinckley et al., combining pen and touch can lead to powerful new tools and offer interactions that may feel more natural [16]. However, such technology remains underexplored by the InfoVis community. Relatively few InfoVis projects have investigated sketch-based interaction for data exploration without relying on menus and buttons [5][6], and some others have focused on multi-touch tables [28]. Thus, due to the lack of research on applying pen and touch interaction to InfoVis, little is known about how people would explore data using pen and touch.

Inspired by the research that has investigated the use of whiteboards as a thinking medium [4][29][44] and SketchVis [5], which uses pen-based interaction for the exploration of InfoVis charts, we explore a novel approach to InfoVis interaction combining pen and touch for data analysis on interactive whiteboards. Our goal is to design an interface for data exploration that offers interactions that minimize interface interference for the analyst and his/her reasoning. To that end, we augment whiteboard capabilities (adding computational power, access to data, and touch interactions) while preserving the basic sketch-based interactions of non-digital whiteboards (e.g., free-form pen sketching). The term natural interactions, as used in the literature [46], now encompasses all interactions that use different types of new technology to enable people to incorporate everyday physicality into their interaction. Being part of this broad picture, we focus on exploring directly manipulating visual elements, leveraging skills learned in the physical environment, and strengthening a person's focus on the task at hand while minimizing the cost of operating the interface.

When designing an interface utilizing free-form interactions such as pen and touch, there exist a plethora of possible interactions worth investigating. To explore the many possibilities in a realistic context

without implementing a wide range of costly recognizers, we opted for a Wizard of Oz study. We report our observations on how people respond to a robust and flexible pen and touch whiteboard for creating and manipulating three types of charts—bar graphs, line graphs, and scatterplots. We conducted a qualitative analysis of the interactions used by 11 participants, and we discuss our insights in the context of learnability of interactions and the interplay between pen and touch gestures. We conclude with suggestions for the design of pen and touch based interactions for InfoVis systems.

2 RELATED WORK

2.1 Pen and/or Touch Interaction

2.1.1 Sketch-based Interaction

Sketch-based interfaces are not a recent concept. One of the first pieces of research taking advantage of sketch dates back to the 1960s, with Sutherland's Sketchpad concept [39]. However, sketch-based interfaces have only recently begun to gain more attention, particularly in the context of 3D graphics modelling (see [31] for a survey) and in math and physics educational tools such as VectorPad [2], MathPad² [23], and the NiCE Formula Editor [24].

There have been only a handful of sketch-based interactions relevant to InfoVis. Graph Sketcher [38] is designed for drawing quantitative concept diagrams (charts that convey some quantitative idea without a specific underlying dataset). QuerySketch [45] and QueryLines [32] allow people to query time-series data with a sketch of the desired graph. Holz and Feiner's relaxed selection techniques let people implicitly define a level of similarity that can vary across search patterns to create a query with a single-gesture interaction [18]. NapkinVis [6], a sketch-based interface for the Protovis toolkit [1], allows people to use symbolic gestures to specify visualizations. SketchVis [5], closest to our approach, investigated hand-drawn input for exploring data through simple charts. We expand upon this direction by investigating the use of pen and touch to mitigate the limitations imposed by sketch-only interactions.

Major issues around sketch-based interfaces involve building robust recognizers (the difficulty increases as the allowable freedom in sketch-based commands increases [33]) and mode switching. Sketch-based interfaces are not inherently modeless and oftentimes require switching between, for instance, drawing and editing modes. Work by Saund and Lank [35] suggests that mode minimization is possible within a sketch-based interface, however our system uses a combination of pen and touch interaction, which can also help minimize mode-switching.

• Jagoda Walny is with the University of Calgary, e-mail: jkwalny@ucalgary.ca.

• Bongshin Lee, Paul Johns, and Nathalie Henry Riche are with Microsoft Research, e-mail: {bongshin, pauljoh, nath}@microsoft.com.

• Sheelagh Carpendale is with the University of Calgary and performed this work as a consultant for Microsoft Research, e-mail: sheelagh@ucalgary.ca.

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

2.1.2 Multi-touch Interaction

Over the last decade multi-touch interaction, especially on tabletops and surfaces, has gained significant attention. The research community has created a dedicated conference, ACM Interactive Tabletops and Surfaces (ITS), focusing on the design and use of new and emerging tabletop and interactive surface technologies. As compared to sketch-based interaction, multi-touch interaction has been applied to more InfoVis projects. Isenberg and Carpendale's interactive digital table allowed a group of people to collaboratively perform tree comparison [20]. Schmidt et al. designed a set of multi-touch link interactions for node-link graph visualizations—plucking, pinning, strumming, and bundling of edges [36]. More generally, Isenberg et al. discussed some ways that multi-touch interaction could be applied to InfoVis [21]. Leveraging multi-touch interaction especially when combined with pen interaction offers considerable potential for benefiting the InfoVis community.

2.1.3 Pen and Touch Interaction

Hinckley et al. argue that pen and touch interaction offers the opportunity to craft new interaction experiences [16]. They advocate an approach where pens are used for writing or otherwise marking the interface, touch is used for manipulating elements, and that combining pen and touch provides additional tools. Brandl et al. found the combination of pen and touch input to be not only faster and more accurate than touch-only or pen-only bimanual input, but also was more preferred [3]. Frisch et al. investigated a rich set of study-elicited pen and touch gestures for editing (rather than exploring) node-link diagrams [13]. Encouraged by all of these projects, we aim to design a pen and touch enabled interface that offers new and more direct interactions for data exploration.

2.2 Observational Studies to Inform Interaction Design

2.2.1 Wizard of Oz Studies

A Wizard of Oz study is one in which a person (the “wizard”) exerts some degree of control over how the studied system responds to a participant's actions. For systems that require an interaction recognizer, the system must be fully implemented except for the recognition layer, because the wizard can act as the recognizer. To the study participant, the prototype appears fully functioning, though slow to respond.

Nielsen, while advocating next-generation interfaces, predicted that Wizard of Oz studies may be required to design them [30]. This kind of study setup allows researchers to see how a participant might react to a sophisticated interface before this interface is actually created. It also allows a certain degree of flexibility in how the researchers choose to respond to participants' actions, something that is very difficult to achieve with rigidly implemented interfaces.

Wizard of Oz studies have been used to study interaction techniques that could potentially accept a great variety of human-generated input, or where creating the interaction recognition component is an expensive investment. This includes a wide range of domains from natural language programming [8], pen-based interfaces [9], augmented reality [43], human-robot interaction [34], to multimodal interfaces [40]. For any kind of interaction, the benefit of a Wizard of Oz study is that it enables researchers to observe whether interaction paradigms will be successful without having to implement an expensive catch-all recognizer. An added benefit is that, with enough participants, training data for the eventual recognizer can be collected.

2.2.2 Observing Individuals to Design Interactions

When designing interfaces with the goal of more natural interaction in mind, it is useful to observe how people naturally interact with systems. While most of these types of studies are qualitative in nature, some are conducted in lab settings where attention is paid to allowing as much freedom as possible to the participants. Fewer studies have been conducted ‘in the wild’ in unconstrained real



Fig. 1. A pen-and-touch whiteboard: a back-projected vertical surface that used an infrared Laser Light Plane (LLP) [41] to recognize touch, and an eBeam [25] device to provide pen functionality.

world situations. The following are exploratory lab-based studies: Grammel et al. observed how novices would construct information visualizations when unhampered by the normal constraints of an interface [15]; Dwyer et al. examined how people manipulate the node-link diagrams given different interactive paradigms (i.e., Surface and Mouse) and what kinds of layouts they produce [10]; Frisch et al. conducted a study to see what kinds of pen and touch gestures people use for editing node-link diagrams [13]. To design an interface for pen and touch, Hinckley et al. observed people's manual behaviors with physical paper and notebooks [16].

Several exploratory lab-based studies [27][47] have asked people to perform touch gestures for specific commands. In these studies, the commands are presented in an isolated way; however the work of Hinrichs and Carpendale observing gestures used in the wild suggests that temporal interaction context and social context play a role in determining what gestures are most naturally used [17]. While our study was conducted in a constrained environment, it expands upon both of these directions, allowing us to observe what gestures people invent for themselves and how these gestures fit into a sequence of interactions in more complex tasks.

3 SKETCHINSIGHT DESIGN

Inspired by the ubiquity of whiteboards in knowledge workers' workspaces, we conducted a Wizard of Oz study to develop a more in-depth understanding of how best to use pen and touch interaction for data exploration on interactive whiteboards. In particular, our study extends the ideas in the pen-based SketchVis system [5] by introducing the possibility of touch interactions and by studying how people spontaneously use and combine these two input modalities in their data explorations.

Several studies [4][29] have found evidence that freeform writing surfaces aid in thinking tasks and a study [44] has reported that charts are commonly sketched on knowledge workers' whiteboards. Blending the freeform whiteboard environment that encourages thinking with the ability to explore actual data in chart form could be a powerful combination. Therefore, the goal of our study is to mimic the natural way one might draw charts on a whiteboard and to augment the whiteboard capabilities by adding computational power, providing access to chart creation and manipulation through pen and touch interaction. To run the Wizard of Oz study, we focused our design on a flexible base system, called SketchInsight (Figure 1), for chart creation and manipulation on a pen and touch interactive whiteboard. We leave for further research opportunities such as studying these types of interactions for such features as preliminary data manipulation, inclusion of more than one data source, and drag-and-drop data adjustments.

3.1 Design Goals

SketchInsight was designed to help us investigate as many potential ways of interaction as possible on a pen-and-touch hardware setup. Because our recognizer was a person, sketched indicators or interaction gestures could be effectively performed by either pen or touch or a combination of the two. To promote the possibility of more natural interactions, we followed four design goals. The second and third align with the design guidelines for fluid interactions introduced in [11] (DG3—Minimize indirection in the interface—and DG8—Minimize explicit mode changes, respectively).

3.1.1 What You Draw is What You Get

Currently available software interfaces for creating and manipulating charts, such as Microsoft Excel [26] and SPSS [19] require people to specify charts using menus and buttons. While Tableau [12] allows people to use drag and drop interaction instead of choosing data fields from menus, people still have to drag these data fields to a dedicated area separate from the chart. In both cases, people have to first think through how to specify necessary information to see the chart they want. SketchInsight instead allows participants to draw the charts they want to see. For example, after drawing both axes, they can draw a bar shape and then write a data field name in place to map the specified data field to the bar.

3.1.2 Make Manipulation as Direct as Possible

Most interactive systems developed so far usually use indirect manipulation of the objects of interest, through direct manipulation [37] of interface elements such as sliders and menus. To leverage the benefit of pen and touch interaction (which provides an opportunity to support manipulation directly on the charts) we did not simply replace a mouse with a pen and touch. With SketchInsight we worked towards creating charts and chart elements that can be interacted with directly instead of using external controls, buttons, or commands. We minimized our use of WIMP interface controls, enabling as much functionality as possible via pen and touch. Some use of context menus was included when handwriting was recognized to be data column, data point, or function names. That is, if a participant started writing a data function name, SketchInsight could offer them a list of suggested function names before they finished writing, in the spirit of auto-completion. The context menus we used looked more like a list than standard menu controls.

3.1.3 Minimize Explicit Mode-switching

Mode-switching usually involves an additional level of indirection in an interface. For instance, in a moded interface a person would select a sketching mode first to be able to sketch an axis, or a manipulation mode to use a similar gesture to move the chart to another area of the screen. This mode switch could get in the way of an analyst’s reasoning process. To enable the analyst to fully focus on his/her tasks, it is critical not to impose the additional cognitive demands required for explicitly handling modes. SketchInsight does not include any explicit mode switching, which is made possible by the use of both pen and touch interaction. For example, using the pen for drawing charts and touch for manipulating them avoids having explicit chart creation and management modes. Studying this via a Wizard of Oz study allowed us to avoid assigning specific modes to pen or touch up front, so that we could watch how participants handled such interactions in a modeless situation.

3.1.4 Support Flexibility with Good Default Behaviors

To provide the flexibility necessary for a Wizard of Oz study and for natural interaction, SketchInsight supports multiple paths to getting a result, where possible. For instance, people can draw the axes for a chart before labelling them to specify data fields for each axis, or they can write the labels and then draw the axes, or they can draw and specify one axis at a time. Those are just a few possibilities – these actions could be performed with one hand or two hands, or in whatever way felt natural to the participant. The system responds

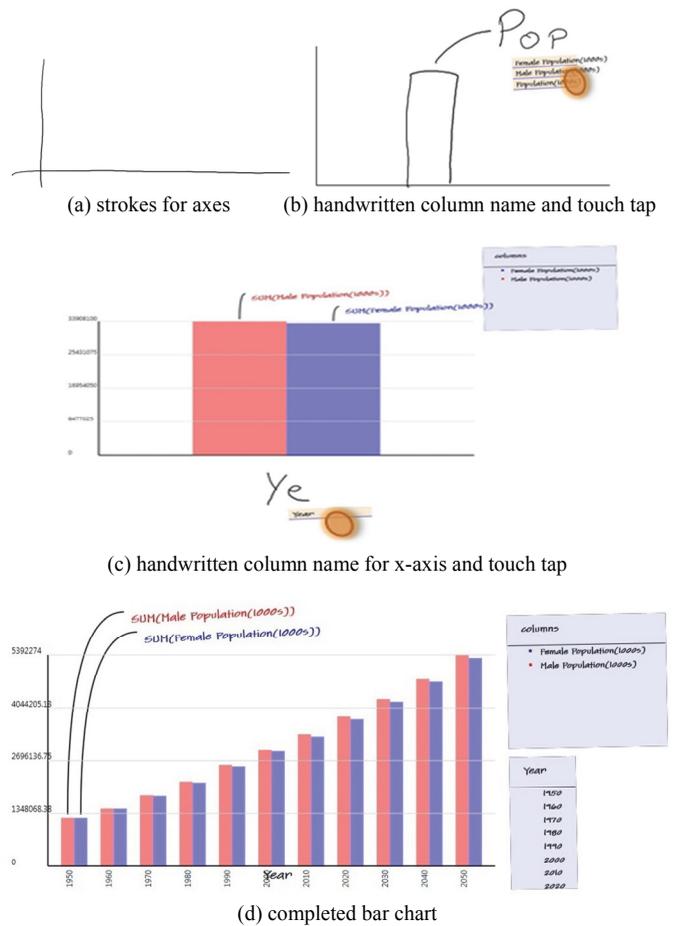


Fig. 2. Creation of a two-column bar chart to compare male and female populations by year. Orange ovals in (b) and (c) are added to show touch action.

where possible with defaults that make sense, even when participants have not provided the complete information that is necessary. For example, SketchInsight has a bar chart as a default chart type and displays data even when an x-axis is not yet specified. Also, as Tableau does, SketchInsight initially applies a default aggregation function (SUM) to avoid showing hundreds or thousands of bars.

3.2 Pen and Touch Recognition

SketchInsight “recognizes” (through the wizard) both pen and touch gestures to support a set of functions for data exploration for three chart types (bar graphs, line graphs, and scatterplots). It makes a distinction between pen and touch actions when appropriate. For example, people can draw only with a pen but select an item from a pop-up menu with either a pen or touch. Figure 2 shows how one could create a two-column bar chart to compare male and female populations from a dataset about world population indicators (Task 1 in our study). Note that Figure 2 illustrates only one example of the many different interaction paths that can create this chart. In this section, we describe the set of baseline interactions supported by SketchInsight for each of pen and touch interaction.

3.2.1 Pen Interaction

The pen makes ink strokes on the screen. Recognized ink strokes are replaced by system-drawn elements, and any remaining ink strokes can be erased (all at once). We made this choice to provide feedback about which strokes were recognized and to simplify the study execution so that the wizard did not have to manage ink strokes. The following ink strokes were “recognized” by default:

- Handwritten data column names (e.g., year), data values (e.g., Africa), or function names (e.g., MAX) for axis labels or data aggregation functions (Figures 2b and 2c),
- Horizontal or vertical lines to draw an axis or ‘L’ shape to draw both axes together (Figures 2a and 6),
- A bar, line, or set of points to change the chart type,
- A strikethrough, check mark, or circle for filtering (Figure 3),
- Scribbles, strikethroughs, or ‘X’ marks for erasure,
- An arrow to specify a move command combined with a circle for specifying the target object to move,
- An arrow to specify a duplication command, combined with a touch and hold gesture to indicate the source object to copy, and
- A pen tap for selecting a menu item.

Other handwriting is treated like any ink stroke—it is left on the screen until the next action is recognized or until it is explicitly erased. One exception is the use of the pen for selection. When participants tapped on a menu item, we removed the ink stroke and treated it as a selection of the item.

3.2.2 Touch Interaction

When participants touch the screen, SketchInsight displays orange “blobs” that indicate where touches have been recognized. We used this to convey touch information to the wizard, but it was also beneficial to participants as feedback when they touched the screen. Also, we did not distinguish between numbers of touches, so any gestures could be performed with one finger or all five. We “recognized” the following touch gestures by default:

- Tapping for selection (Figures 2b and 2c),
- Dragging to move,
- Swiping (e.g., strikethrough with a finger or hand) for erasure or opening the dataset tray,
- One- or two-handed pinch gesture to resize, and
- Hold (with one hand) and drag (with the other hand) to copy.

3.3 SketchInsight Baseline Features and Interactions

Through pen and touch interactions, SketchInsight supports the creation and manipulation of three basic chart types: bar charts, line charts, and scatterplots. Although we designed the study to be as flexible as possible in terms of potential interactions, we did have a set of baseline interactions in mind when we designed SketchInsight. In this section, we describe these baseline interactions for supported features, along with the system’s default behaviors. Note that, when a participant employed a new way of interaction we did not anticipate, we adjusted our expectations to match his/her interactions and strove to avoid ambiguity. For example, if a participant “assigned” a specific interaction technique to an action, we responded as consistently as possible throughout the session (as it is unreasonable to expect a full-fledged system would do otherwise).

We also made several assumptions about how the system would be used. First, there would be one person interacting at a time and one active dataset at a time. Second, the person using the system would be familiar with the structure (column names) of the dataset. (We provided this information to the participants.) Third, the dataset would consist of a single table. Also, as described under our design goals, we tried to avoid making a menu-driven interface as much as possible. However, we did display context menus that showed filtered names of data columns and names of available functions, when appropriate. For example, when a data column name or data value was being handwritten, the wizard displayed a context menu showing all matching column names (Figure 2b).

3.3.1 Dataset Initialization

Loading a dataset is a prerequisite to creating a chart. A tray in the lower left corner of the screen houses a list of datasets. Loading a dataset requires opening the list of datasets by tapping or swiping the visible tab on the bottom left corner of the screen, then choosing an item from the list of datasets. SketchInsight was designed for the choices to happen through a tap (either by a pen or by touch) or by drawing a circle around the dataset name with the pen.

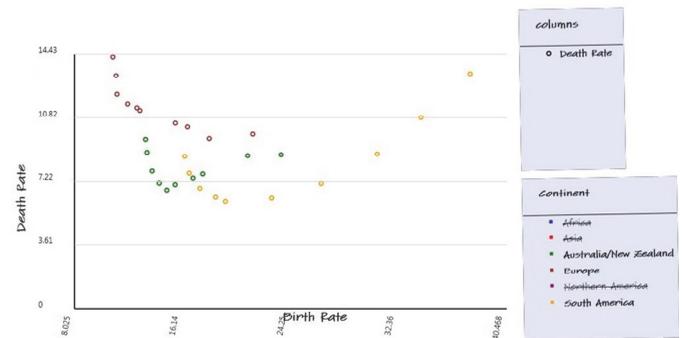


Fig. 3. A strikethrough on a legend filters out the data.

3.3.2 Chart Initialization and Removal

The minimum conditions for a chart to exist are for it to have a y-axis. A full chart has a data column associated with the x-axis and at least one data column associated with the y-axis. It is possible to associate more than one data column with the y-axis, for example, to display a multi-column bar chart. By default, as Tableau [12] and Microsoft Excel [26] Pivot do, SketchInsight applies an aggregation function (SUM) over the data—this is to avoid the possible clutter that can be caused by showing hundreds or thousands of bars. SketchInsight also applies a default chart type (bar chart) so that it can display something as soon as it has enough information. SketchInsight displays a legend for the y-axis data columns and their associated visual variable (color) as shown in Figure 2c.

To provide flexibility in creating a chart, SketchInsight was designed for chart creation to happen in two ways; 1) by first drawing both axes, then labelling each axis in turn or 2) by drawing and labelling one axis at a time. The baseline interaction for assigning the x-axis was to write a label below the axis; for assigning the y-axis, it was to write a label to the left of the axis or inside of the plotting area (particularly for multiple-column plots) as shown in Figure 2b. Participants could remove any charts that were no longer needed. For removing charts, SketchInsight was designed to support an erase gesture over one or a set of charts.

3.3.3 Data Exploration

Changing chart type – The chart type can be changed between bar graph, line graph, or scatterplot to support a variety of tasks. To change a chart's type SketchInsight was designed to support participants drawing a representation of the chart type inside the plotting area (i.e., a bar, line, or set of points). For a scatterplot, the desired shape of the points can be indicated in the drawing (e.g., circles, squares, or ‘x’s).

Data transforming functions – SketchInsight supports applying a set of aggregation functions including sum, average, max, and min. To change or remove the aggregation function, the baseline interaction was erasing and labelling over the only visible indication of the aggregation function's existence, which surrounds the label for the y-axis data (e.g., SUM (Male Population(1000s)) in Figure 2d).

Data filtering and grouping – Data can be filtered by values in a data column and grouped on the x-axis by values in a column. For filtering, SketchInsight was designed to support first a grouping action, then erasing elements inside the accompanying legend (Figure 3). To add elements back in it expected circling or checking off crossed-out elements in the legend. For grouping by another column, it expected to see either an additional label on the x-axis, or the creation of another legend by writing the column name somewhere on or outside of the plotting area (but not on the y-axis).

Managing data columns – In addition, data columns can be added or removed as desired. For adding data columns to a chart, SketchInsight was designed for people to either add another label to the y-axis or to draw and label a bar/line/point (depending on the chart type); it would also accept adding a column label to the chart's

legend. For removing data columns, SketchInsight expected an erase gesture over a column name (either an axis-label or a legend title), or over a bar/line/point or set of points in the plotting area.

3.3.4 Management of Charts

More than one chart can be displayed on the whiteboard at a time. Charts can be moved, resized, duplicated, or deleted, but they cannot be rotated, and chart elements cannot be manipulated individually. Also, to optimize the wizard's response time, we chose not to support rotation, panning, and zooming of the canvas.

SketchInsight was designed to support moving of elements by dragging or, alternatively, drawing arrows to the desired location of the move. The baseline interaction for resizing was a one- or two-handed pinch gesture; for duplication it was a hold-and-drag gesture.

4 WIZARD OF OZ STUDY

The goal of our study was to find out what kinds of interactions people gravitated towards naturally when using SketchInsight, with its modeless, buttonless, pen and touch interaction, for data exploration on interactive whiteboards.

4.1 Study Setup

In our Wizard of Oz study setup (Figures 1, 4 and 5), the wizard acted as a moderator. The system drew ink strokes on the whiteboard and responded directly to some interactions, such as selecting a context menu item. The wizard controlled the system's other responses. Our hardware included the following three components:

- A pen-and-touch whiteboard (Figure 1) through which participants interacted with the system. It was a back-projected vertical surface that used an infrared Laser Light Plane (LLP) [41] to recognize touch, and an eBeam [25] device to provide pen functionality.
- A 21" Wacom Cintiq pen-enabled display [7] (Figure 5, right monitor) for the wizard, which mirrored the whiteboard display.
- Two network-connected desktop computers (one for the participant and the other for the wizard) to run our software.

The software consisted of two programs: the wizard's control panel on the wizard machine and the charting program on the participant machine, both connected to a shared database on the wizard machine for communication (Figure 4). The wizard's control panel (Figure 5, left monitor) stood in for a recognizer and sent commands (e.g., "Draw x-axis" or "Add column to chart") to the shared database, which was polled every 500 milliseconds by the charting program. We used two separate machines to avoid the problem of the wizard application stealing focus from the participant.

The touches detected by the LLP setup were shown to the participant as orange "blobs" on the screen, but in reality did not have any effect beyond making the wizard and the participant aware of which parts of the screen were being touched. The eBeam device captured ink strokes that were interpreted by the charting program.

We logged all ink strokes and commands polled from the database, captured screencasts of the charting software, and captured video and audio of the participants interacting with the whiteboard.

4.2 Participants and Procedure

4.2.1 Participants

We recruited 12 participants (5 males and 7 females) through mailing lists and word of mouth. However, we did not use the data from one participant due to repeated task interruptions caused by system issues during his/her session. Our participants came from a variety of different fields: 4 computer scientists; 3 management and systems analysts; and one each from math, physics, kinesiology, and design. One of the computer scientists had experience with sketch-based modelling in 3D graphics. All participants stated that they were familiar with Microsoft Excel. All but one had experience with touch-based computational devices such as smartphones and tablets.

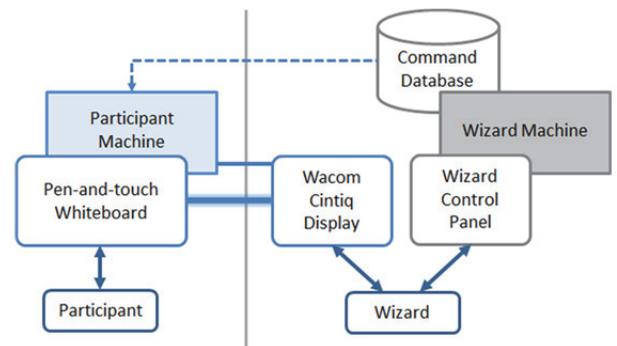


Fig. 4. Our Wizard of Oz study involved two network-connected desktop computers.

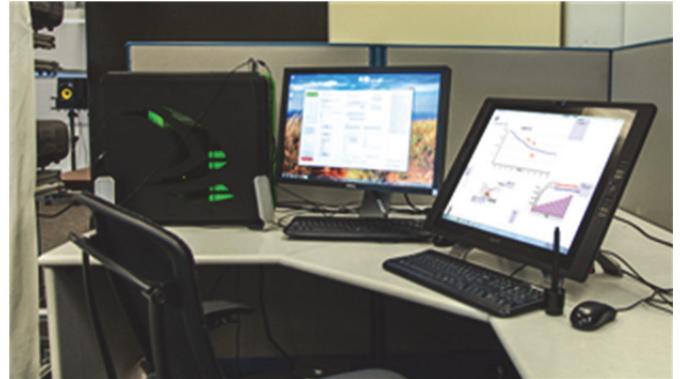


Fig. 5. The physical setup for the wizard.

Three had previous experience with touch-based and pen-based whiteboards. Participants received CAD\$20 for their participation.

4.2.2 Procedure

At the start of each session, we briefed participants by telling them they would be working with a pen and touch prototype system to perform a series of tasks for data exploration. We encouraged them to interact with the system however they wished. We asked participants to think aloud about their intentions and reactions while they were performing the tasks.

A sample of the dataset was available to participants throughout the study, since SketchInsight was designed for people working with familiar datasets. Participants could refer to the dataset or ask questions about it to the experimenter throughout the study. Then we briefly showed participants that it was possible either to draw on the whiteboard with the pen or to touch with any number of fingers; beyond this we did not provide any formal training on the system.

After this introduction, we asked participants to complete a series of eight tasks in whatever way they wished. Most of these tasks contained multiple steps. After completion of each step of a task, the participant was shown a screenshot of the expected result; if the results did not match, they were asked to repeat the step.

In a debriefing, following the completion of all of the tasks, we informed participants that this was, in fact, a Wizard of Oz study (i.e., the wizard recognized their pen and touch input). They were then asked to fill out a short questionnaire about their previous experience with pen- and touch-based technologies. We also asked a few open-ended follow-up questions about the experience with SketchInsight. Each session lasted up to 1.5 hours but was typically 45 minutes.

4.3 Dataset and Tasks

We used a dataset about world population demographics containing 9 columns (Continent, Year, Population, Male Population, Female Population, Fertility Rate, Birth Rate, Death Rate, and Net Migration Rate) and 66 rows (i.e., items). We prepared the dataset by extracting demographic data from 1950 to 2010 and projections from 2020 to

2050 from the UN World Population Prospects [42] and aggregating it by continent to simplify the data.

There were eight tasks (six of which consisted of multiple steps), presented in the same sequence to each participant. These tasks were designed to provide complete coverage of the major features of SketchInsight. While tasks were designed to be specific, their wording did not suggest *how* they were to be performed. For example, “Make a bar chart with *Year* on the x-axis, bars for *Male Population*, and bars for *Female Population*.”; “Turn this into a scatterplot.”; or “Make sure there is no aggregation function set over *Death Rate*.” The eight high level tasks were:

- T0: Loading the dataset
- T1: Creating a bar chart with multiple bars
- T2: Creating a scatterplot
- T3: Creating a line chart with multiple lines
- T4: Filtering data from a scatterplot
- T5: Changing aggregation functions
- T6: Filtering data from a bar chart
- T7: Removing all charts from the screen

Each task ended with the instruction to “keep this chart for later” so that participants would freely arrange the charts on the screen.

4.4 Wizard Response

Each time the participant performed an action, such as a pen stroke or a touch on the screen, the wizard interpreted the action and sent the appropriate command to the charting program. When necessary, the wizard temporarily blocked the participant’s interaction to allow time to send the appropriate command; a large hourglass indicated that interactions were currently blocked. The wizard followed several guidelines to determine how to interpret actions:

- *Respond in a consistent manner for the duration of a session and throughout the study as much as possible.* For example, if a participant “assigned” a meaning to a specific action, we did not change this assignment throughout that session.
- *Allow participants to perform the same action in multiple ways.* For example, moving a chart could be performed by dragging the chart using any hand position or by circling the chart with the pen and drawing an arrow to the destination.
- *Respond only to actions that could reasonably be interpreted as having a particular meaning.* For example, some participants would double tap on top of a chart but their intent was ambiguous to the wizard, so no response was given.
- *Do not respond to written commands or single taps (except for menu item selections).* This guideline was added after the first two sessions to encourage participants to explore the freedom of expression a pen offers compared to a mouse.

4.5 Analysis and Results

We started our analysis process by collaboratively developing a set of themes that we saw in the data and subsequently using an open-coding approach similar to that described by Jordan and Henderson [22]. Two of the researchers each coded the same session independently, using both session video and screencast as data. The resulting codes were then discussed, refined, and verified for coding consistency until the Jaccard similarity score reached 86.5%. The two researchers subsequently coded all sessions using the mutually agreed-upon codes.

We examined our data for each participant by the intended functions (e.g., *erase*) of the participant’s actions (e.g., *swipe*) used to achieve high-level tasks. These intended functions were generally readily apparent because of the ‘think aloud’ protocol, the way that the tasks were designed, and most clearly by the participant’s apparent satisfaction upon success. For each participant we coded and counted all of the different ways that they performed each intended function using either pen or touch or pen and touch in combination. The intended functions we coded for were: initializing a chart, changing the chart type, filtering, grouping, erasing, duplicating, resizing, and moving. We also coded for sketch-based

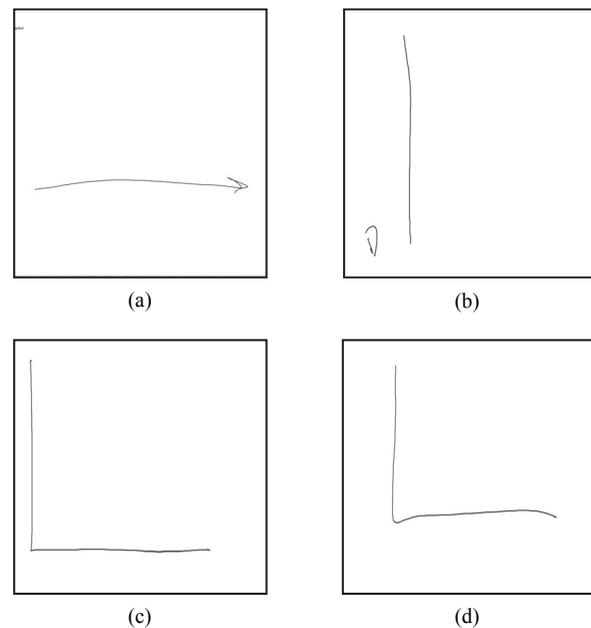


Fig. 6. Progression of ways to initialize a chart throughout P5’s session. In (a): x-axis drawn first, with an arrowhead; (b) y-axis drawn first, no arrowhead; (c) two axes drawn in two strokes; (d) two axes drawn in a single stroke.

actions such as drawing the chart axes (initializing a chart), as well as for touch gestures.

Here, we first report the range of actions participants performed for each intended function. When possible the distribution (i.e., the number of participants) is numerically appended in brackets. These guided us to draw some higher-level findings, which are discussed in the next section.

Initialize a chart: Each participant created several charts; some of them used more than one method; however, considering the variety possible when using freeform pen and touch interaction, there was consistency across participants. The first time this task was presented, 6 out of 11 participants first tapped the “dataset loaded” icon, wanting to start by looking at the data. However, since the intention was to study chart creation and manipulation, there was no available system response for this. Only two of the participants repeatedly tried invoking a menu of options before moving on to drawing axes. There were several styles of drawing axes: drawing two axes with one stroke in an L-shape (e.g., Figure 6d) was used by five participants, three of them almost exclusively throughout their session; drawing two axes with two strokes in an L-shape (e.g., Figure 6c, used by seven participants, four of them almost exclusively throughout the session); and drawing one axis at a time (used by five participants; two of them exclusively throughout the session). Some participants changed their style throughout the session. For example, two participants used arrowheads the first time they drew axes but not subsequently. Participant 5’s progression of axis drawing is shown in Figure 6.

Change chart type: This was a difficult function when it was presented the first time. Four participants tried several combinations of written commands and invoking a list of options (neither of which we supported), before a series of hints from the researcher guided them into trying a pen- or touch-based interaction for this task. The hints progressed from a vague one (e.g., “you have a pen and two hands.”) to more specific ones (e.g., “how would you draw a scatterplot?” and “try drawing a scatterplot.”). Three participants had no problems coming up with a pen- or touch-based action; two participants needed no hints but took a few tries; one participant needed one vague hint; and five participants needed the more specific hints. We allowed one participant to change the chart type using a written command, which resulted in this participant using

written commands for all subsequent similar tasks. (We no longer recognized written commands after this session). In summary:

- 8 participants used sketch-based actions throughout. Two of them did this with no hints, and one with only a vague hint,
- 2 participants used touch to mimic the chart type at first, but ended up using the pen later, and
- 1 participant used written commands throughout.

Filter: Participants often used slightly different actions to indicate that they wanted a data item included (filter-in) and that they wanted a data item excluded (filter-out). Filtering-in actions included using checkmarks, encircling, touching, scrubbing, and writing the data point name. Filtering-out actions included most of the same actions: strikethroughs, 'x' marks, touching, and scrubbing. Some people used the same action to indicate both filtering in and out. For instance, touch and scrub were used this way, as a toggle.

Group by: Participants had two chances to group the chart by a particular data column. One was with a line chart and the other a scatterplot. We saw the following range of actions:

- Draw multiple lines on chart (5: 4 with a pen, 1 with touch),
- Write data point name next to the y-axis, inside the plotting area, to the right of the plotting area, or in a legend (8), and
- Write data column name next to the y-axis, in plotting area, or to the right of the plotting area (3).

Erase: Participants erased elements a total of 139 times, using a wide variety of different actions to indicate their intent to erase. These included:

- Drawing an 'x' mark with the pen (8),
- Striking through the object to be erased with a pen (8),
- Swiping across the object with their hand (3),
- Scrubbing the object out with repeated finger movements (2),
- Scribbling over the object with the pen (1), and
- Drawing an 'x' mark with a finger (1).

Duplicate: We saw a wide variety of actions for duplicating a chart, for example anchoring and dragging, anchoring (with touch) and drawing a line away from the chart, circling and drawing an arrow, circling and writing a command, simply writing a command, and various combinations of tapping and double tapping on the chart and desired location for the copy. In general, participants tended to indicate a source chart, optionally an intended function (copy), and a destination location in various ways:

- Source: anchor with finger, draw a circle, tap, double tap, tap with two fingers,
- Copy action: written command or implied through the way the destination was specified, and
- Destination: drag gesture, arrow, or a single or double tap.

Resize: This was done with amazing consistency. All but one participant used bimanual touch, drawing their hands together or apart to indicate the change in size. This was for a total of 75 resize actions over ten participants. The only other method used (4 times by only one participant) was to circle the object using the pen and then to draw a new smaller circle to show the new size.

Move: While not quite as consistent as resize, move was indicated with a touch-drag gesture by seven participants for 78 actions. Other methods included using the pen to circle the item and an arrow to indicate the destination (3 participants). One participant simply used a line without an arrow and another participant once added a second circle in the destination next to the arrow.

Pen-in-Hand: When we coded for a touch gesture, we noted whether they did their touch action with their pen in their hand, moved the pen elsewhere, or used their opposite hand (for variations see Figure 7). Most participants used a combination of techniques, but they were also fairly consistent. As a result participants either performed:

- Over half of their touch actions with their pen in hand (6),
- Over half of their touch actions with pen moved aside (3),
- Over half of their touch actions with the opposite hand (1), or
- Roughly equal distribution among these three options (1).

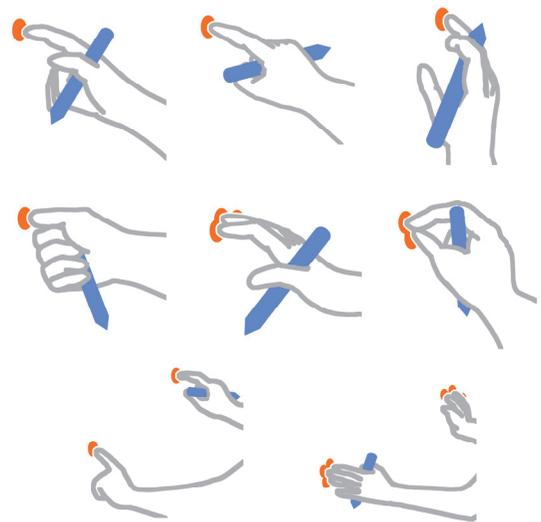


Fig 7. Various hand positions observed when performing pen and touch interactions.

We saw some other attempted actions that we did not respond to. Participants did fall back on the familiar and try to invoke menus by pressing and holding on the screen, and to replace commands by writing the command as a word. However, since we had not built SketchInsight to have complex menus, we simply told participants that these commands were not supported.

5 DISCUSSION: INSIGHTS FROM THE STUDY

Here, we discuss higher-level findings grounded in how participants performed the lower-level actions delineated in Section 4.

5.1 Learnability

We originally thought that what is natural would be easily discoverable (e.g., writing on a whiteboard with a marker or pen). However, we observed that this was not necessarily the case and that people became accustomed to different interactions if they are learnable and memorable, even if they were not simple to discover. Also, people drew from past experiences both on and off computers in choosing interactions.

We found that it is possible to design learnable interactions, and that these interactions can become natural once learned. In fact, we had expected that participants would try a wide range of actions, but instead they quickly learned the system's boundaries and stayed within them. While humans deftly adapt to new environments, this providing learnable interactions can ease this adaptation.

5.1.1 Transferred from the Physical World

Participants transferred a significant amount of knowledge from the physical world. When tasks had direct physical counterparts, these were commonly used for interaction, as in moving or stretching.

For touch-based interactions, we found evidence that physical metaphors were an important factor in the gestures people used. Gestures for moving, resizing, and duplication were overwhelmingly performed with touch. The resize gesture was most consistent among participants, almost always being a two-handed stretching gesture—10 of 11 participants resized exclusively by spreading two hands in opposite directions. Moving had some variation, but was generally performed as a drag gesture with varying numbers of fingers—9 of 11 participants generally moved objects using a drag gesture, with only three participants performing moves using the pen.

We found people often used conventions with counterparts from paper usage. For example, when moving objects with a pen, they circled the object and drew an arrow to the destination, as one would illustrate the concept of moving an object when sketching. Erasing or filtering was similarly done with sketch of strikethrough, 'x', etc.

Initializing charts by drawing axes was a common action that relied on previous chart-drawing knowledge. Although 6 participants initially wanted to see or manipulate the dataset before creating a chart, their next interaction was drawing one or two axes, as on paper or a whiteboard. Subsequently, no participants had trouble initializing charts, indicating this was an easily learnable interaction. Similarly, labelling the axes was a natural step. Once the sketched axes were recognized, no participant hesitated to label them.

Duplication was interesting as it has no strict physical counterpart, and so it required some creativity in terms of conveying the desired action to the system. It appeared that the closest mental model some participants had was a physical action: dividing something into two. P9, when trying to think of ways to duplicate a chart, commented: *"Maybe I would try to grab with both hands and divide it into two,"* then placed the sides of his/her hands on the chart and spread them apart. P10 remarked, *"I was thinking of the apple. If you give it to two children, you cut it in half. But that's not what I want to do. I want to get another apple."* P10 then attempted to copy the chart by tapping first on the chart, then on another part of the board. However, duplication gestures generally followed the "specify source, specify command, specify destination" model, suggesting a close association with move. This was one task for which some participants combined pen and touch gestures, via an "anchor and draw line" interaction.

These findings suggest that leveraging pre-existing conceptions about the world can lead to interactions that come naturally once learned. In cases where there is no direct physical counterpart, as in duplication, tasks may still follow a pre-existing metaphor.

5.1.2 Transferability within the System

More importantly, in some cases, knowledge gained through feedback while using the system was then used to infer how to perform similar actions later on. This aligns with the findings from the psychology research on analogical transfer, where people learn a principle and apply it to different cases [14].

The most prominent example of this is changing the chart type. Our baseline interaction for changing the chart type—drawing a representation of the chart type (e.g., drawing a bar, a line, or points) inside the plotting area—turned out to not be so easily discoverable, but was readily transferred. When initially asked to change the bar chart to a scatterplot, only three participants immediately tried drawing points inside the plotting area. Two people at first used touch to indicate the switch to a scatterplot (making points by tapping inside the plotting area multiple times in a random fashion), but both ended up switching to pen-based actions for later, similar actions. The idea of drawing what they wanted to see, where they wanted to see it was arrived at with varying degrees of difficulty for the first chart type switch. However, all participants readily extrapolated to analogous actions for subsequent chart changes and swiftly performed future switches. For example, none of them hesitated to use a similar strategy later, when they were asked to change the chart type to a line chart or a bar chart.

Another example is the task of grouping the chart by the values of a particular column. For this task, knowledge was transferred in one of two ways: participants either mimicked how they had previously learned to retrieve columns (through labelling), or they demonstrated how they wanted the chart to look (by drawing more lines or points).

5.1.3 Feedback Aids Learning

It is well recognized that providing the right feedback leads to a more satisfying experience, especially when no other instructions are given as with our study. SketchInsight did not provide much feedback when people's actions were not recognizable. For example, it did not highlight an active element as traditional desktop applications do. This caused some confusion and frustration. We saw a few cases in which a combination of pen and touch was used for moving in that the element to be moved was "selected" first through circling. This was especially true of the smaller elements, such as chart labels. The participants later explained that they had wanted to

ensure that the correct elements were going to be moved. Since people learn how to use the system through how the system reacts to their actions, it would be important to rectify this in a real system and perhaps to offer suggestions when an action is not recognized.

We suspect that our visual feedback inevitably had an effect on how people drew things later. For example, most people converged to drawing axes in an L-shape once they saw this in system chart axes. Thus, to leverage what people draw naturally it might be preferable to augment strokes to indicate they were recognized instead of replacing them. For example, SketchInsight could use an axis with an arrow if people draw an axis with an arrow and put the axis labels where people wrote them instead of using a fixed location.

Recognizing drawn axes is an example of effective feedback for learnability. Most of our participants were not experienced with sketch-based interfaces, so it was asking a lot to stand them in front of a blank screen and ask them to communicate with the system. It was apparent that while participants were unsure of what to expect from the system upon drawing the axes, seeing the system recognize these axes gave them an idea of how the system would work.

5.1.4 The Effect of Preconceptions about Interfaces

We also saw evidence of knowledge transferred from previous computer experience (including multi-touch capable devices). Conceptually, it was evident that our participants were accustomed to choosing from a list of possible options rather than sketching what they wanted to see, and resizing items with multi-touch, drawing their hands together or apart to indicate the change in size. However, despite the fact that we only supported a limited number of familiar interactions, 10 of 11 participants completed all of the tasks, and in follow-up interviews most were confident that they would have no trouble using the system in the future.

5.2 Pen vs. Touch Interactions

Our participants had a clear distinction in their minds regarding what types of actions should be touch- or pen-based. This was evident both in the way that they used the system and in their responses to our follow-up questions (10 of 11 participants described this distinction confidently and in a way consistent with our observed results). For example, P12 remarked, *"It's nice that I can use the pen for some stuff like writing for example, but my hand for other stuff, like moving objects."* Such a sentiment was consistent with other participants, and seemed to be encouraged by our study setup. For example, P11 said, *"Well if it's set up the way it is here, then dragging is definitely the finger because otherwise you're drawing lines, so it's nice to have that separation, to have control over whether you're drawing a line or dragging."* These results are consistent with Hinckley et al.'s [16] pen vs. touch results, despite the fact that our study context had fewer real-world counterparts (artifacts are not normally moved around on a vertical whiteboard as they are on flat surfaces like tables). In general, touch was used for arrangement actions that had a counterpart in the real world, such as moving and resizing (stretching) objects, and pressing elements as if they were buttons (though buttons were not supported in our study).

Some actions blurred the line between pen and touch: particularly erasing and selecting menu items. Both of these were fairly common interactions. Erasing tended to elicit pen interactions, but we also saw a significant number of touch-based erasing interactions. 6 of 11 participants erased exclusively with the pen, using strikethroughs or 'x' marks (rarely scribbles). One participant erased exclusively with touch, and several used mixed pen and touch. Interestingly, for selecting items from the column and function context menus, the balance was shifted slightly in favour of touch. 6 participants used touch or mostly touch interactions; 4 used pen or mostly pen; and 1 used a mixture of the two. The many examples of interactions where people used touch intentionally while still holding the pen is clear evidence that people had active ideas about when they wanted to use pen or touch (see Figure 7). For 7 of 11 participants, a significant number of touch gestures occurred with the pen still in their hand.

5.3 Integrated Interactions

We use the term *integrated interactions* to refer to interactions where a person's hands, tools, actions, interactions, visual response, and feedback are in situ where the data is visualized. That is, to effect an interaction, a person's attention is not drawn away from the visual representations of data in which they are interested. Aside from some participants' initial searches for fixed interface elements akin to menus, our participants' interactions remained focused on the area of their current chart. These interactions were almost entirely in location—on the item, in the plotting area, on the data bar, and so on.

5.4 Implications for System Design

Our findings comprise a collection of insights pointing to the feasibility of pen and touch interfaces for some InfoVis applications. Although a preliminary study such as this cannot offer precise declarations for designing a pen and touch enabled chart exploration system, we offer new insights and many confirmations that interactions suggested in HCI research are applicable to InfoVis.

We have seen that pen and touch interactions have great potential to lead to more integrated interactions for InfoVis. While we are not suggesting that interface widgets be placed within representations, we are suggesting locating modeless data-direct interactions in situ.

The idea behind the design goal of 'what you draw is what you get' seemed readily transferable. While only 5 participants immediately tried sketching what they would like the system to do and the others needed varying degrees of hints to make this attempt, all 11 subsequently made use of this idea. This suggests that while this was a novel concept to several participants, they picked up the idea and extrapolated related interactions—assuming that if drawing dots in the plot area triggered a scatterplot, drawing a line would change it into a line chart. This idea is at the heart of sketch-based interfaces and seems to be worth further exploration in InfoVis.

The clear use of pen and touch in combination is another useful direction for InfoVis interfaces. InfoVis interfaces usually need interactions for both data access, often involving many small and precise details, and data navigation, which may use broader, less precise actions. Participants had clear distinctions about when to use pen or touch for such interactions, as can be seen in their consistent, smooth transitions between pen and touch interactions.

There are also many interaction ideas that were confirmed as useful in InfoVis chart manipulation. These include: consistent use of metaphors leads to more learnable and explorable interactions, good feedback gives people confidence in their interactions and helps people learn the system, and leveraging existing knowledge of the physical world to inform the design of pen and touch interactions helps to make interactions more discoverable. While these last four ideas are well-established interaction concepts, it is useful to find them effective in a pen and touch InfoVis interface.

We found little evidence to support distinguishing between multi-finger and single-finger gestures. Few participants tried using more or fewer fingers to make a gesture work. This may be partially due to the physical world metaphor described above; to drag or stretch physical objects there is rarely a distinction between using one or many fingers (except when an anchor point is needed, e.g., rotation).

5.5 Limitations

Our study focused on the creation and manipulation of InfoVis charts. It was readily apparent that there are broader aspects that are closely related that would be well worth studying. For instance, our participants actively looked for interactions enabling direct access to the original data values, however, our system did not support these.

We did not provide much feedback—for example, indicating selected items or real-time movement—partly because we wanted to minimize our influence on people's interaction. However, it was impossible to avoid influencing their learning completely, since we had to respond to people's interaction in a consistent way based on our intention to learn more about pen- and touch-based interactions. Additionally, we restricted the interactions we responded to: if we

had supported invoking menus or written commands, it is likely that our participants would have stayed with familiar WIMP-type actions and we would not have seen such diverse actions. We wanted to focus on the feasibility of a sketch-based pen and touch interface, rather than simply appropriating current WIMP-based InfoVis interfaces for a vertical pen-and-touch display, since there are already many successful WIMP-based InfoVis interfaces.

It was difficult for the wizard to be fully consistent across participants and sometimes even within a participant. In addition to the cognitive load and stress on the wizard to make quick, consistent decisions, the infrared Laser Light Plane (LLP) [41] we used to recognize touch was not 100% reliable. For example, sometimes the system did not recognize touch when it happened near corners, or the system recognized a pen or person's sleeve as part of touch.

6 CONCLUSION AND FUTURE WORK

Inspired by the dominant use of whiteboards as a thinking medium [4][29][44] and SketchVis [5], we explored a novel approach to InfoVis interaction combining pen and touch for chart creation and manipulation on interactive whiteboards. To begin to learn more about how to design an interface that does not get in the way of the analyst's reasoning, we conducted a Wizard of Oz study to observe how people would interact with a modeless, buttonless, pen and touch system. Our results show a clear interplay between pen- and touch-based interactions. In Section 5 we discussed how these results provide new insights into how to best support pen and touch for InfoVis chart interactions. Specifically, we discussed:

- *Division of labor for pen and touch:* Our participants clearly distinguished between appropriate pen and touch interactions and would perform some actions via touch even while the pen was still in their hand (Figure 7),
- *Explorations into new interaction paradigms:* While the predominant menu-based interfaces certainly influenced initial interactions, participants readily switched to new expectations set by SketchInsight, using system feedback for guidance,
- *Leveraging two types of transferable learnability:* Participants readily extrapolated new interactions by transferring knowledge about both physical world interactions and those already learned from using the system.
- *Working towards integrated interaction:* Where interactions are located within the visualization in proximity to the elements they act on, rather than using externally-located interface widgets.

In our study, we asked people to create and configure charts rather than explore data. It would be useful to investigate if pen and touch enabled interactions also enhance people's analysis capabilities. Thus, a next step is to implement the system based on the lessons learned from this study, and test it with analysts exploring data.

Our study assumed one active person at a time, however, since whiteboards are often used for collaboration, it would be interesting to see how groups of people would interact with the system. Also, we assumed that the person using the system would be familiar with the structure of the dataset. It is possible that using whiteboards to explore and understand data new to the analyst would be a different and interesting research topic.

Furthermore, especially with recent advances in interaction technologies, there is a vast space of using next-generation interaction for data exploration on whiteboards that offers the enormous opportunity to craft new interaction experiences. It would be interesting to extend SketchInsight's interaction by using additional modalities such as speech and body gesture.

ACKNOWLEDGMENTS

We would like to thank Andy Wilson for helping us set up the whiteboard at Microsoft Research and providing the touch recognition library, Piali Choudhury for writing the code to provide feedback on touch using the library, Travis Kirton for setting up touch and pen interaction at University of Calgary, and Dominikus Baur for the sketched images.

REFERENCES

- [1] M. Bostock and J. Heer. Protovis: A graphical toolkit for visualization. *IEEE TVCG (InfoVis 2009)*, vol. 15, no. 6, pp. 1121–1128, 2009.
- [2] J.N. Bott, J. LaViola, and J. Joseph. A pen-based tool for visualizing vector mathematics. *Proc. SBIM 2010*, pp. 103–110, 2010.
- [3] P. Brandl, C. Forlines, D. Wigdor, M. Haller, and C. Shen. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. *Proc. AVI 2008*, pp. 154–161, 2008.
- [4] S. Branham, G. Golovchinsky, S. Carter, and J.T. Biehl. Let's go from the whiteboard: Supporting transitions in work through whiteboard capture and reuse. *Proc. CHI 2010*, pp. 75–84, 2010.
- [5] J. Browne, B. Lee, S. Carpendale, N. Riche, and T. Sherwood. Data analysis on interactive whiteboards through sketch-based interaction. *Proc. ITS 2011*, pp. 154–157, 2011.
- [6] W.O. Chao, T. Munzner, and M. van de Panne. Poster: Rapid pen-centric authoring of improvisational visualizations with NapkinVis. *Posters Compendium InfoVis 2010*, 2010.
- [7] Cintiq 21UX Pen Display | Wacom Americas. <http://www.wacom.com/en/Products/Cintiq/Cintiq21UX.aspx> (accessed June 26, 2012).
- [8] N. Dahlbäck, A. Jönsson, and L. Ahrenberg. Wizard of Oz studies — why and how. *Knowledge-Based Systems*, vol. 6, pp. 258–266, 1993.
- [9] R.C. Davis, T.S. Saponas, M. Shilman, and J.A. Landay. SketchWizard: Wizard of Oz prototyping of pen-based user interfaces. *Proc. UIST 2007*, pp. 119–128, 2007.
- [10] T. Dwyer, B. Lee, D. Fisher, K. Inkpen, P. Isenberg, G. Robertson, and C. North. Understanding multi-touch manipulation for surface computing. *IEEE TVCG (InfoVis 2009)*, vol. 25, no. 19, pp. 961–968, 2009.
- [11] N. Elmqvist, A. Vande Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T.J. Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, vol. 10, pp. 327–340, 2011.
- [12] Fast Analytics and Rapid-fire Business Intelligence from Tableau Software | Tableau Software. <http://www.tableausoftware.com> (accessed June 26, 2012).
- [13] M. Frisch, J. Heydekorn, and R. Dachsel. Investigating multi-touch and pen gestures for diagram editing on interactive surfaces. *Proc. ITS 2009*, pp. 149–156, 2009.
- [14] M.L. Gick and K.J. Holyoak. Schema induction and analogical transfer. *Cognitive Psychology*, vol. 15, pp. 1–38, 1983.
- [15] L. Grammel, M. Tory, and M.A. Storey. How information visualization novices construct visualizations. *IEEE TVCG (InfoVis 2010)*, vol. 16, no. 6, pp. 943–952, 2010.
- [16] K. Hinckley, K. Yatani, M. Pahud, N. Coddington, J. Rodenhouse, A. Wilson, H. Benko, and B. Buxton. Pen + touch = new tools. *Proc. UIST 2010*, pp. 27–36, 2010.
- [17] U. Hinrichs and M.S.T. Carpendale. Gestures in the wild: Studying multi-touch gesture sequences on interactive tabletop exhibits. *Proc. CHI 2011*, pp. 3023–3032, 2011.
- [18] C. Holz and S. Feiner. Relaxed selection techniques for querying time-series graphs. *UIST 2009*, pp. 213–222, 2009.
- [19] IBM SPSS software for predictive analytics. <http://www-01.ibm.com/software/analytics/spss> (accessed June 26, 2012).
- [20] P. Isenberg and S. Carpendale. Interactive tree comparison for co-located collaborative information visualization. *IEEE TVCG (InfoVis 2007)*, vol. 13, no. 6, pp. 1232–1239, 2007.
- [21] P. Isenberg, U. Hinrichs, M. Hancock, and S. Carpendale. "Digital ables for Collaborative Information Exploration." *Tabletops - Horizontal Interactive Displays*, C. Müller-Tomfelde, ed., Springer-Verlag, pp. 387–405, 2010.
- [22] B. Jordan and A. Henderson. Interaction analysis: Foundations and practice. *J. Learning Sciences*, vol. 4, pp. 39–103, 1995.
- [23] J. LaViola, J. Joseph, and R.C. Zeleznik. MathPad²: A system for the creation and exploration of mathematical sketches. *ACM SIGGRAPH 2007 Courses*, 2007.
- [24] J. Leitner, C. Rendl, F. Perteneder, A. Gokcezade, T. Seifried, M. Haller, R. Zeleznik, and A. Bragdon. NiCE formula editor. *Proc. ACM SIGGRAPH 2010 Talks*, pp. 55:1–55:1, 2010.
- [25] Luidia Inc.: eBeam Classic Whiteboard: eBeam Whiteboard Turns Any Whiteboard into a Digital Copyboard. <http://www.e-beam.com/products/ebeam-classic-whiteboards.html> (accessed June 26, 2012).
- [26] Microsoft Excel - Spreadsheet - Office.com. <http://office.microsoft.com/en-us/excel> (accessed June 26, 2012).
- [27] M.R. Morris, J.O. Wobbrock, and A.D. Wilson. Understanding users' preferences for surface gestures. *Proc. GI 2010*, pp. 261–268, 2010.
- [28] C. Müller-Tomfelde. *Tabletops - Horizontal Interactive Displays*, Springer-Verlag, 2010.
- [29] E.D. Mynatt. The writing on the wall. *Proc. Interact 1999*, pp. 196–204, 1999.
- [30] J. Nielsen. Noncommand user interfaces. *CACM*, vol. 36, no. 4, pp. 83–99, 1993.
- [31] L. Olsen, F.F. Samavati, M.C. Sousa, and J.A. Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, vol. 33, pp. 85–103, 2009.
- [32] K. Ryall, N. Lesh, T. Lanning, D. Leigh, H. Miyashita, and S. Makino. QueryLines: Approximate query for visual browsing. *Ext. Abst. CHI 2005*, pp. 1765–1768, 2005.
- [33] F.F. Samavati, L. Olsen, and J.A. Jorge. *Sketch-based Interfaces and Modeling*, pp. 1–15, 2011.
- [34] P. Saulnier, E. Sharlin, and S. Greenberg. Exploring interruption in HRI using Wizard of Oz. *Proc. HRI 2010*, pp. 125–126, 2010.
- [35] E. Saund and E. Lank. "Minimizing Modes for Smart Selection in Sketching/Drawing Interfaces." *Sketch-based Interfaces and Modeling*, J. Jorge and F. Samavati, ed., Springer-Verlag, pp. 55–80, 2011.
- [36] S. Schmidt, M. Nacenta, R. Dachsel, and S. Carpendale. A set of multitouch graph interaction techniques. *Proc. ITS 2011*, pp. 113–116, 2011.
- [37] B. Shneiderman. Direct manipulation: A step beyond programming languages. *IEEE Computer*, vol. 16, no. 8, pp. 57–69, 1983.
- [38] R. Stewart and M.C. Schraefel. Graph sketcher: Extending illustration to quantitative graphs. *Proc. CHI 2009*, pp. 1113–1116, 2009.
- [39] I.E. Sutherland. Sketchpad: A man-machine graphical communication system. *Proc. AFIPS Spring Joint Comp. Conf.*, pp. 329–346, 1963.
- [40] R. Taib and N. Ruiz. Wizard of Oz for multimodal interfaces design: Deployment considerations. *Proc. HCI 2007*, pp. 232–241, 2007.
- [41] A. Teiche, A.K. Rai, C. Yanc, C. Moore, D. Solms, G. Cetin, J. Riggio, N. Ramseyer, P. Dintino, and L. Muller. Multi-touch technologies. *NUI Group*, 2009.
- [42] United Nations, Department of Economic and Social Affairs, Population Division. *World Population Prospects: The 2010 Revision*, CD-ROM Edition, 2011.
- [43] S. Voidsa, M. Podlaseck, R. Kjeldsen and C. Pinhanez. A study on the manipulation of 2D objects in a projector/camera-based augmented reality environment. *Proc. CHI 2005*, pp. 611–620, 2005.
- [44] J. Walny, S. Carpendale, N. Henry Riche, G. Venolia, and P. Fawcett. Visual thinking in action: Visualizations as used on whiteboards. *IEEE TVCG (InfoVis 2011)*, vol. 16, no. 6, pp. 2508–2517, 2011.
- [45] M. Wattenberg. Sketching a graph to query a time-series database. *Ext. Abs. CHI 2001*, pp. 381–382, 2001.
- [46] D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann, 2011.
- [47] J.O. Wobbrock, M.R. Morris, and A.D. Wilson. User-defined gestures for surface computing. *Proc. CHI 2009*, pp. 1083–1092, 2009.