TouchViz: A Case Study Comparing Two Interfaces for Data Analytics on Tablets

Steven M. Drucker[1], Danyel Fisher[1], Ramik Sadana[1,2], Jessica Herron[1], m.c. schraefel[1,3]

[1]Microsoft Research
1 Microsoft Way
[sdrucker,danyelf]@microsoft.com

[2]Georgia Institute of Technology
ramik@gatech.edu

[3]University of Southampton
mc@ecs.soton.ac.uk

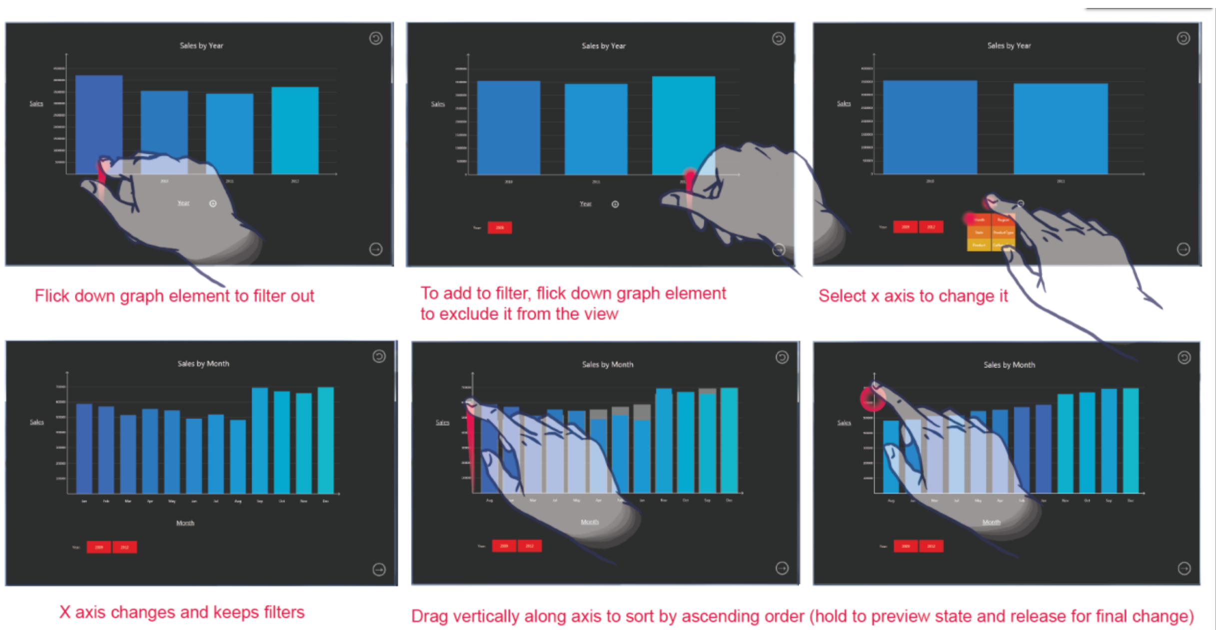

Figure 1: A touch and gesture oriented interface for visual data analytics (FLUID interface).

ABSTRACT

As more applications move from the desktop to touch devices like tablets, designers must wrestle with the costs of porting a design with as little revision of the UI as possible from one device to the other, or of optimizing the interaction per device. We consider the tradeoffs between two versions of a UI for working with data on a touch tablet. One interface is based on using the conventional desktop metaphor (WIMP) with a control panel, push buttons, and checkboxes – where the mouse click is effectively replaced by a finger tap. The other interface (which we call FLUID) eliminates the control panel and focuses touch actions on the data visualization itself. We describe our design process and evaluation of each interface. We discuss the significantly better task performance and preference for the FLUID interface, in particular how touch design may challenge certain assumptions about the performance benefits of WIMP interfaces that do not hold on touch devices, such as the superiority of gestural vs. control panel based interaction.

CHI'13, April 27 – May 2, 2013, Paris, France.

Author Keywords

Gesture interfaces; touch displays; user studies; data visualization.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms

Experimentation, Human Factors

INTRODUCTION

With the wide range of available tablet devices, an increasing number of applications are being translated from mouse-based desktop interfaces to touch-based interaction styles. Application authors are faced with a dilemma in implementing interactions: do we simply port the WIMP (Windows, Icons, Menus and Pointer)-style interfaces that are so familiar on the desktop, or should we change the interaction style to a more fluid, 'Touch-centric' interface that affords more direct manipulation of that application? This tradeoff has implications in terms of developer cost for implementation, ability to transfer interaction knowledge from one interface device to another, and in user preference and performance. We explore this question in the context of the particular domain of visual data analytics. Visual data analytics is a good model for many information worker tasks; it does not require data entry, but it has several clear task taxonomies of prototypical behaviors. In addition, there has been a historic link between direct manipulation interfaces and visual analytics [3, 16].

We wish to explore whether or not direct manipulation of data objects in the case of data visualization benefits from more direct manipulation of the data on touch surfaces, and if so, in which contexts, in comparison with WIMP-translated-to-touch interfaces. We have taken a dual approach: we first carried out a longitudinal exploration of a data manipulation design space to develop a candidate set of touch gestures. We then developed test applications to implement these gestures, and to contrast them with a control-panel-based approach. We refer to these approaches as FLUID (a gesture-centric touch based interface) and WIMP (point and click menus, icons, and other widgets that control the data). We then developed a series of prototypical questions in a study for users to address.

Our goal is to understand whether, and how, the fluid, touch-based gesture interaction offers subjective or performance advantages over the current WIMP approach to data exploration on touch surfaces. In this paper, after a brief review of related work, we present a review of our exploration of the design space for a fluid, touch-based data interaction gestures, and the resulting UI. We then present a study in which we compare this FLUID approach against an implementation of the WIMP approach. We show the results of the comparison and offer an analysis and observations of this comparison.

RELATED WORK

We broadly set our work in the context of the areas of direct manipulation interfaces, data visualization, and gestures and touch interfaces.

Trade-Offs of Direct Manipulation

One of the design approaches we explore in this work entails directly interacting with data. "Direct Manipulation," a term coined by Shneiderman [16], refers to the idea of being able to interact with data items on a screen. In the original 1983 work, this referred to keyboard or mouse commands; touch displays seem a logical extension. In their cognitive account of direct manipulation, Hutchins, Hollan, and Norman [10] use an example of an information visualization interface. They argue that "directness" should be seen as an orienting notion for design: a designer can build with the goal of creating a more-direct interface, one which tries to match the physical requirements of the system, the user's mental model, and the feedback users see. They also identify a number of potential trade-offs to direct manipulation interfaces: there can be difficulties with precision; with separating actions on *an* item from *a class* of items—and when a user is unfamiliar with their goals, direct manipulation may not feel intuitive.

Nielsen [14] has a similar concern with contemporary interface design. He complains that UI "chrome"—his term for buttons, menus, and status bars—can take up the majority of a user's computer screens. He suggests that replacing some chrome with gestures might allow users more screen space to do their work—but, in turn, will require users to remember these gestures. This is a concept that we explore in great detail in the designs and study presented later.

Manipulating Data Visualization

The area of data visualization has been a particularly rich one for interaction techniques. The field of visualization has spent a great deal of effort considering how users can easily create and interact with standard visualizations of their data. Polaris [17], a system for visualizing data cubes, provides an interface for creating visualization based on 'shelves'; a user drags fields to shelves to specify the data mapping in a visualization. This contrasts with techniques like Microsoft's Excel, in which a user uses a classic noun-verb paradigm: select a column from the spreadsheet, and then click "create chart".

With advent of tablet, surface, and whiteboard systems, there are new opportunities for manipulation of visualizations. SketchVis [5] is one new interface that allows users to draw data on a whiteboard. The user draws the coordinate system, writes labels on axes, and draws some initial points; the system responds by filling in the chart correspondingly. The user can filter objects from the chart by crossing them out. While there are some similarities between this work and ours, we focus less on textual or abstract gestures and more on gestural interaction with the data representation itself.

Gestures and Touch

While familiar mouse interfaces have well-understood rules, taxonomies for gestures are still being developed. Designers have learned that gestures have different affordances. Wigdor & Wixon [18] provide guidelines for both gestures, and for creating gesture recognition systems, and outline a number of issues that are raised in gesture systems, from the 'fat-finger' problem (fingers are much larger than mice) to issues around teaching users to recognize new gestures.

Yee [21] suggests guidelines for finding appropriate gestures: they should use appropriate metaphors, and should be easy for the system to recognize. Beyond that, Yee describes gestures as *abstract* versus *direct*. A direct gesture has a correlation with motion, and relates to a single object on screen; an abstract gesture is an invocation that causes an action elsewhere (such as the "three finger swipe" on trackpads to scroll a document).

Wobbrock *et al* develop a gesture vocabulary elicited from users by having them carry out mock gestures on a surface computing device [20]; they find that a second pool of users prefers the user-created set to an expert-created gesture set [12]. North *et al* similarly elicit gestures from users on a tabletop device, and find users exploring a broad selection of gestures for moving items around [15].

Guimbretiere et al [8] and others have explored the relative benefits of merging manipulation with command selection (as opposed to tool palette selection.) While we purposely chose not to use marking menus in the current prototype, the integration of selection with action is a key benefit to the touch based interface we propose.

DESIGNING GESTURES FOR DATA INTERACTION

We set out to develop what might be experienced as an "intuitive" set of gestures for touch-based manipulation of data visualizations. Our design process had three stages: we started with a **task analysis**, proceeded through multiple **brainstorming sessions**, and then created an **animated storyboard.**

Stage 1: Task analysis and a semantic action taxonomy

We first discuss the actions that we wish to accomplish separate from any particular instantiation in the interface. We call these actions 'semantic' actions; the taxonomy is intended to express the core operations that are used in visually exploring a dataset. Our taxonomy loosely follows the "Data & View" stage of visualization described in Heer & Shneiderman [11], who provide a guide to interaction with visualization. We describe a visual exploration as requiring:

- **Choose** which categories of the data to represent in the visualization
- **Select** a subset of the data,
- **Filte**r irrelevant data, focus on relevant items.
- **Order** the data in order to expose patterns,
- **Navigate** (selective zoom) in order to examine high level patterns and low level detail.

Stage 2: Gesture Brainstorming and validation

None of these operations correspond directly to well-known tablet gestures. To develop a set of candidate gestures, we ran a structured brainstorming session with 6 users and generated over 5 to 10 gestures for each semantic action in the taxonomy. Participants were knowledgeable HCI practitioners, 4 of whom were also extensively experienced with data visualization.

The resultant sets of actions were organized into four primary categories: (1) actions that mimicked the way a mouse is currently used on a desktop (e.g. primarily single clicks on objects); (2) representative gestures that involve drawing either shapes or text on the screen; (3) actions that would be more convenient with touch or multi-touch (e.g. flicks, swipes, multi-touch, and smooth motion), and that mapped onto objects within the visualization; (4) and actions that involved physical movement or sensing on the device (e.g. tilting the tablet to sort the data, or blowing on the microphone to filter out unwanted data).

We chose to focus on the third category—gestures that involved manipulation mapped directly onto objects on screen—since these were overwhelmingly the favorites of the brainstorm participants.

Stage 3: Animated Storyboard for Participant Validation

We wished to refine the gestures from the brainstorm into a single vocabulary. To articulate the gesture set, we created an animated storyboard; the animation could incorporate transitions between states (which has been shown to often clarify actions to users [11]) and allowed us to see how the gestures would work together.

In the process of creating the animation, we focus on one particular visualization type. Bar-charts are familiar to many observers; in addition, most of the operations we were interested in could be carried out on a bar-chart. The rigor of working through all of the gestures into a consistent storyboard caused us to refine our gesture set to work better as a cohesive, unambiguous whole. For example, at one point we had three different ways to filter the data; one of these ways conflicted with the gestures for sorting; another conflicted with the mechanism for selecting elements. These contradictions, as well as some inconsistencies in our vocabulary, were highly visible in the video prototype.

COMPARING WIMP TO FLUID GESTURES

We wished to determine whether these touch gestures would be easier for users in carrying out tasks. Our evaluation was based on a comparative, within-subjects user study. Therefore, we implemented the gesture set as a touch-based interface and implemented an equivalent WIMP-like touch-based interface to represent the familiar buttons condition. We then selected a series of questions that would represent standard tasks.

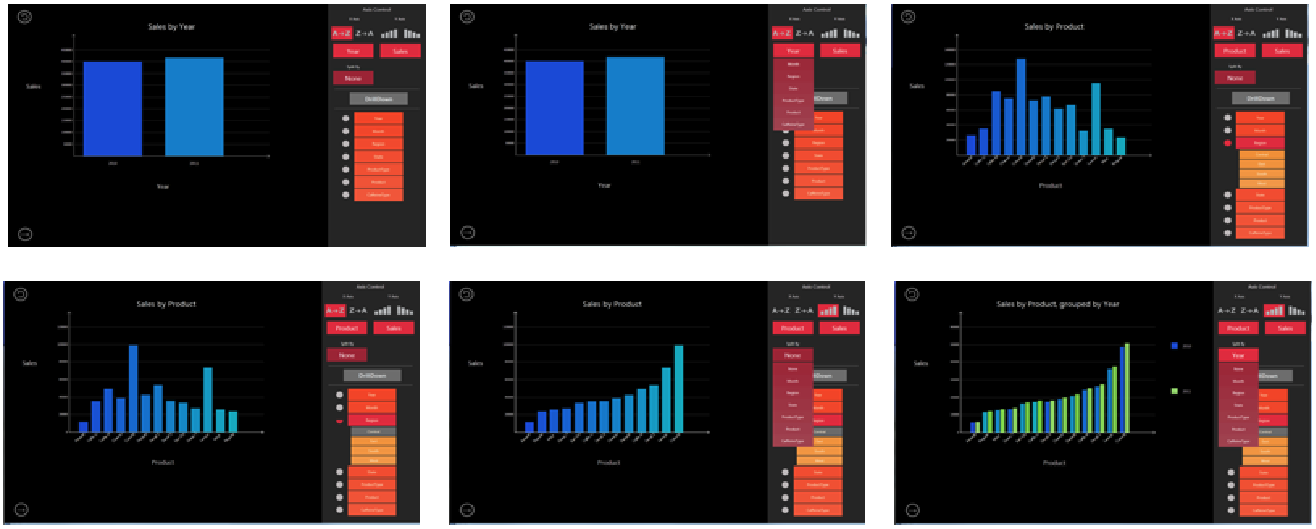

Figure 2: A walkthrough of how actions are achieved in the final WIMP interface prototype. All actions occur at the control panel. In this series of operations, the user starts with showing total sales per year, chooses to switch to showing sales of products, filters out products from the central region, sorts the products, and splits out by year.

Designing Two Interfaces

Our goal in designing these two interfaces was to ensure that they both could be easily operated with touch operations. We endeavored to make sure that both UI's were as usable as possible for their genre: no UI was a crippled or less featured version of another.

The systems were designed for an ASUS Eee Slate running the Microsoft Windows 7 operating system. The display of the device was 12.1" with a resolution of 1280× 800 pixels and a refresh rate of 64Hz. The device had multi-touch input with maximum of two touch points. The touch event information was accessed through the Windows 7 Touch and the Microsoft Surface APIs. The applications were set to full-screen mode with no OS chrome visible.

WIMP interface

The WIMP interface—that is, the one based on a traditional mouse-like interface—was designed around a control panel which was situated on the side of the screen. This is similar to the approach taken by Tableau Mobile, an iPad-specialized version of the Tableau visualization software. The mobile edition is based on the interface for their Web product; it is 'touch enabled' by growing menus and checkboxes to be large enough to be clickable by a finger and scrollable via dragging. In the WIMP interface, the only thing that the user can do on the visualization itself is selecting a bar by clicking on it. Once clicked, additional information encoded by the bar is revealed (e.g. total sales for that category).

We made sure that the WIMP interface could be easily operated using a tablet – menu targets and option boxes were enlarged appropriately and scrolling through lists (when necessary) was achieved by dragging the list upwards or downwards.

The control panel is shown in Figure 2. In our system, users can choose the mapping of each axis by clicking near the top of the interface (directly below the sort buttons for the

	WIMP	FLUID
Choose categories	Menus in control panel	X & Y Axis labels of visualization double as menus
Selection	Tap on a single bar	Tap on a single bar
Filtering & Focusing	Checkbox list in control panel. Toggle for selecting all or none for filtering and focusing.	Flicking down (to filter out) or up (to focus) from visualization on single element or all selected elements.
Order	4 buttons in control panel (ascending, descending along each axis)	Swipe along desired axis upwards or downwards
Zoom	Drill down button in control panel	Tap and hold on item.

Table 1: Comparison between how actions are achieved in WIMP and FLUID interfaces.

Question	Actions (FLUID)	Actions (WIMP)
What region has the largest sales?	CA	CA
What region had the biggest sales of coffee?	CA,F,CA	CA,F
In year with biggest sales, what was the product that sold the least?	F, CA,S	F, CA,S
In the eastern region, what product had the smallest sales?	CA,F,CA,S	F, CA,S
Which region had least difference in sales between decaf and regular?	CA, ADD	CA,ADD
What 4 products had the most profitability?	CA, CAY, S	CA,CAY,S
What 4 states have bought the most decaf espresso?	CA,F,CA,S	CA,F,S
In the region with the most sales, what's the worst selling state?	CA,S,F,CA,S	CA,S,F,CA,S
In product with the most profits, what state had the best sales?	CAY,CA,S,F,CA,CAY,S	CA,S,F,CA,CAY,S

Table 2: Questions used to refine both interfaces, 30 questions similar to these (15 for each condition) were used in the comparison study. (CA – Change X Axis, F – Filter, S – Sort, CAY – Change Y Axis, Add – Group By).

axes). A button below that allows users to create a clustered bar chart along a second dimension (see Figure 2). A **drill down** button allows users to select an item and filter out all other items while viewing a subcategory (for instance, you can drill down on the year 2010 to reveal the months by selecting the 2010 bar and clicking drill down). Below the drill down button is the **filter bank**. Users can open up any category and choose to turn on or off any item within that category at any time. For instance Figure 2 the user has opened up the 'Product' Category and filtered out 'Ameretto' from the view. The user can either filter out everything but a single item, or filter out only a single item.

Last, a "back" button in the top-left allows a user to undo their most recent gesture.

The accompanying Video Figure demonstrates the WIMP interface.

FLUID interface

The FLUID interface is based on the design work reported in the previous section, and strives to exclude as many buttons and controls as possible. All gestures occur on the visualization itself. This allows the visualization to use more screen real estate, but in some cases limits what actions can be performed at any time. For example, a category can only be filtered if it is on screen; there is no way to filter out a category that is not on screen. Figure 1 shows the FLUID interface.

Rather than a "back" button, the FLUID version supports a preview mode. As a user carries out a gesture, the system shows a visual confirmation. The user can finish the gesture to confirm, or 'back off' the gesture to keep the visualization state unchanged.

In this interface, the user can change an axis by touching the axis label to reveal a submenu. Sorting involves swiping on the chosen axis in the selected direction. A user can filter by flicking the screen. When they flick an item downwards, it is filtered out of the visualization; a place holder for that element is now shown at the bottom of the visualization.

If the user selects an item and flicks upwards, then the visualization filters out all items other than the current one. Multiple items can be focused on or filtered out by selecting each of them first and then flicking the group upwards or downwards. Filtered-out items are stored in a bottom storage area; the user can flick them back into the visualization either individually or all together.

Figure 1 illustrates how a user first flicks out the year 2009 and then the year 2012 to just focus on the years 2010 and 2011. The user then taps on the label for the X axis to change the X category of what is being displayed in the visualization. The user then selects the month and the visualization is updated to show a breakdown of sales by month. To sort the data, the user taps on the Y axis and drags upwards (to sort in ascending order) or (downwards to sort by descending order). Clicking on the X axis and moving to the left or right sorts the categories in chronological or alphabetical order based on what category is currently being shown.

Choosing a Task List

We developed a set of typical business intelligence style questions that could be answered with the prototype (see Table 2). These questions were adapted from the analytical task taxonomy of Amar, Eagan, & Stasko [10], which focuses on creating a common task substrate for comparing analytical systems. The common actions they find contain both low-level, fundamental tasks, including *retrieve value*, *filter*, *compute derived value*, *find extrema*, and *sort;* as well as higher-level tasks, such as determining range, clustering and correlation. All of these tasks can be done with the low-level 'semantic operations' discussed above; we confirmed that users could address these tasks with our interface. These questions also became the basis for the comparative evaluation of both prototypes.

Table 2 shows each question along with the operations required by each interface to answer the question. Note that different questions required different number of operations depending on the interface being used, e.g. the WIMP interface allows for general filtering in views not currently shown whereas the FLUID interface does not. Also note that some of the later, more complicated, questions required seeing the outcome of intermediate steps before progressing to the later steps. We made sure that all questions could be answered with both interfaces.

Design Pilot

Both designs were iterated repeatedly against our test set with both expert and novice participants to make sure that the designs and layouts made sense to the user. We refined the visual presentation of the control panel several times in order to group the actions in a way that was clear to end-users. We felt confident stopping when pilot subjects were able to successfully complete tasks after minimal instruction, in either interface.

User Study

For both the WIMP and FLUID versions, all question explorations could be carried out with the bar chart visualization. For our live test data source, we used the Coffee Chain database and the Superstore Sales database available publicly through the Tableau repository as our two datasets. Table 1 shows a comparison of how actions are achieved in each interface.

The user study was designed to determine if users had either subjective or performance differences between the two interfaces, and to determine if there were particular operations that were preferable for one condition.

We implemented the study as a within-subjects design; each subject got a chance to use each interfaces on a different dataset. We counterbalanced order and dataset assignment with a latin-square design: half the subjects experienced a Coffee Products dataset with the WIMP interface and half experienced with the FLUID interface; half of the subjects used the FLUID interface first, while half the subjects used the WIMP interface first.

We created 15 questions of the same general form as those in Table 2; we then modified the fifteen to be appropriate for each dataset. Thus, questions for each dataset were controlled to be exactly the same type and requiring the same number of operations, modified appropriately for the given dataset. For instance, in the Coffee dataset we asked 'Which 4 states have bought the most decaf espresso?' while in the Office dataset we asked, 'Which 4 states have bought the most furniture?'

Protocol

Each subject was first shown a brief demonstration on how to use the interface, showing how to select, filter, sort, and change axes for the data. They were then given 3 minutes to freely interact with the interface. In the next stage, each user was asked 10 progressively more difficult questions designed as a tutorial to help subjects use the interfaces to answer questions. In this stage, if the user had questions or problems, they were encouraged to ask the experimenter for help on using the interface. After the tutorial, each subject was then given a packet of 15 questions that they were to answer on their own as quickly and accurately as possible. 2 questions were identical to tutorial questions, while 13 others were similar but required slightly different filters or axis choices. We encountered a bug in one implementation for solving one question, so that was eliminated from the packet for both interfaces. After a brief break, the process was repeated with the second interface and the other dataset. Again, the user was shown a brief demonstration, was given 3 minutes to freely interact with the interface, answered 10 tutorial questions with assistance from the experimenter, and then was given a packet of 15 questions to answer on their own. After using both interfaces, subjects filled out a number of subjective questions for each interface, including ease of use, ease of learning, perceived speed and efficiency of the interface and overall preference.

We recruited participants with at least some experience using Microsoft Excel and performing charting operations. We carried out the study with 17 participants, 11 men and 6 women, ages ranging from 32 to 60 (mean=47.1, sd=7.5). The entire study took just 35 minutes for the fastest user, and 85 minutes for the slowest. Accuracy was recorded manually for each question; the system logged and time stamped all interactions in log files for later analysis. (The log for user P2 could not be analyzed due to a technical error.)

Results

In broad summary, we found that 13 of the 17 users preferred the FLUID Interface (F) to the WIMP Interface (W). Users made significantly fewer errors on the questions when using the FLUID interface. Users were also significantly faster using the FLUID Interface (F) on answering questions.

Quantitative Results:

ACCURACY: Users were more accurate with the FLUID Interface (F) in comparison to the WIMP Interface (W). From all users, there were 13 errors made with F and 32 errors with W. For each user, the mean number of correct responses was 93.6% (F) and 84.3% (W). There was a significant effect for interface $t(355) = 355.185$, $p < 0.01$. There was no significant effect for either interface presentation order $t(405.8) = 3.121$, $p=0.8748$ (means 88.7% and 89.2% for first and second respectively), or dataset $t(404.6) = 0.4731$, $p=0.6364$ (means of 88.2% and 89.7% for Coffee and Office respectively).

SPEED: Users were faster overall with the FLUID Interface (F) in comparison to the WIMP Interface (W) with mean answer time for each question at 32.05 sec and 44.32 sec respectively. We ran a one-way repeated measures ANOVA and found a significant effect of interface on speed of completion, $F(1,14) = 9.4002$, p <

0.01, and a significant effect of stage (stage 1 for first set of questions and stage 2 for second set of questions) on performance time, $F(1,14) = 5.3298$, $P < 0.05$, but no effect of dataset on performance time and no significant interaction between interface and order.

Figure 3 shows a breakdown by interface on a per user basis. Only three users had a higher mean question time on the FLUID than the WIMP interface and they all encountered the FLUID interface first in the study.

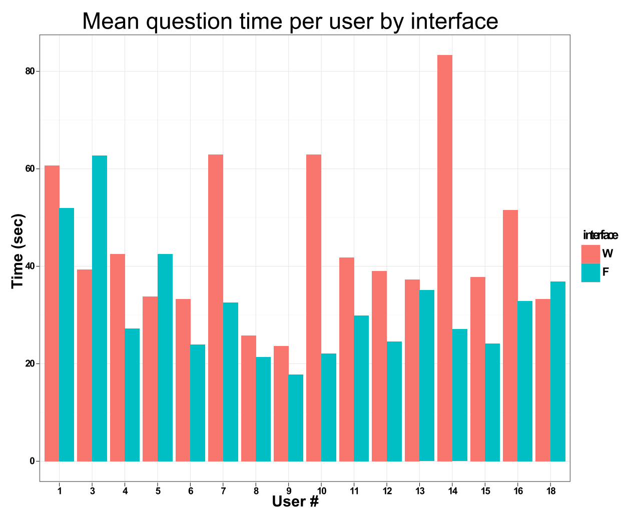

Figure 3: Mean question answering time by user by interface.

Qualitative results:

On the post-study questionnaire, users were asked to individually rate ease of use (W1, F1), ease of learning (W2, F2) and speed and efficiency (W3, F3) for each interface on a -3 to 3 Likert scale.

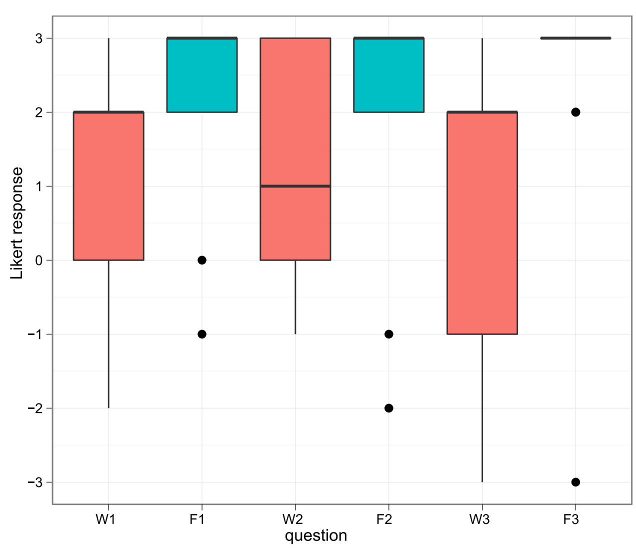

Figure 4: Likert responses for ease of use (W1,F1), ease of learning (W2,F2), and speed & efficiency (W3, F3). Higher numbers indicate increased satisfaction.

UI PREFERENCE: In interviews, five of the users volunteered that they found the FLUID interface fun to use and engaging. P12 said "the [FLUID] interface focuses more on the data and less on the interface" which he liked.

Four users preferred the WIMP interface to the FLUID one. In interviews, they reported that the WIMP interface was more familiar to the ways that they currently work; they liked to have the power of being able to perform any operation at any time as well as being able to quickly see what was currently filtered in and out of the dataset. They cited familiarity with working style, and liked the notion of having all controls available at any time. P3 said, "I'm a pilot and I like to see everything at once and know what's going on".

FUNCTIONALITY: Three users commented that they felt that the WIMP interface had more functionality or was more powerful, but that they liked the FLUID interface better since it matched the way they liked to work through the problem better. Another user felt that the WIMP interface had too many options which made him confused about how to start answering some of the questions. He said that the WIMP reminded him of a "Reverse-Polish" calculator, somewhat non-intuitive.

Regardless of interface, filtering was a difficult concept to many of the users. In the F interface, users could only filter fields that were currently visible on screen; in W, however, users could filter fields that were not currently visible which could be confusing. Even in the W condition, many users decided to change their X- axis over to a field that they were about to filter in order to observe the immediate effects of the filter.

DISCUSSION: IMPLICATIONS FOR DESIGN.

Overall, there seemed to be clear gains in translating the interface from a more conventional desktop WIMP based interface to a more gestural based direct interaction when using a touch device. This was clear from the subjective preference reports as well as speed and accuracy improvements.

In trying to unpack the attributes of the design that garnered both the performance boost and the enjoyments of participants, we noted two major aspects:

- **gestural interaction** seemed to have particular benefits for interacting with objects, compared to the more indirect interface
- A **problem solving approach** that was particularly appropriate for the gestural interface that encouraged users to deal more directly with the data

Gesture:

People claimed to find the FLUID interface felt more intuitive and more natural, at least in comparison to button presses or making selections for operations on a control panel. One user said that 'if touch is available in the device, it should be used and it's weird when it's not available'. They also felt that 'the [FLUID] interface focuses more on the data and less on the interface'.

Flick gestures allow the near simultaneous specification of the 'noun' (the object upon which interaction is performed), the 'verb' (the action such as filter or sort), and even occasionally an 'adjective' (filter in or filter out, sort ascending or descending). In contrast, the control panel interface enforced more of a separation between objects and the actions taken on them. This is similar to the findings of Gumbretiere et al [8]. However the Gumbretiere work focused on rapid interactions, often without a great deal of cognitive overhead for accomplishing their tasks. Overall interaction times for our work was much slower for both interfaces and the performance advantages might have been due to a different process entirely detailed in the following section.

Problem solving approach

Perhaps more intriguing than just the preference and performance results of the two interfaces is an exploration of the reformulation of the problem solving approach that was encouraged by the more direct manipulation of data that the touch approach afforded. People were able to perform both faster and with fewer errors in the FLUID version; based on people's responses, it seems that this result is based less on the particular interaction type–the touch gestures–and more about the type of problem model represented in the UI.

One way to describe the differences between the WIMP and FLUID is that one uses additional space to trade off with user memory. The additional space allows all options to be in view while the other forces users to remember gestures and commands. Touch devices, especially those with limited screen-size, often force us to remove menus and buttons from the screen. The cost is that the interfaces become more difficult to remember how an action is performed. Perhaps ironically, interfaces that require gestural memory may facilitate a type of interaction that has been largely deprecated on the desktop given the larger available screen-space. The heuristic from research of desktop design is that it is better to recognize than recall, for instance; hence the presence of persistent menus and controllers and performance evaluations have supported that heuristic.

So why is an interface that requires users to recall the gestures performing better than one where the commands are available on the screen? Is it because of a particular combination of attributes between gesture and spatial such that the gestural benefits outweigh the additional memory cost?

We suspect that in focusing on the affordances of what could be easily manipulated directly, and by restricting generality, we ended up with an interface that both performed better and was preferred by an overwhelming majority of the users.

In the data analyst tasks performed, the problem solving approach required for using the FLUD interface scaled more clearly from answering simple questions to answering more complicated questions. For example, to answer the question, 'Find the state that sold the most coffee", users could answer the question in the WIMP approach by switching to a view of sales by state and then filter out the coffee product type. In the FLUID approach, they needed to switch to a view of products types, filter out coffee, and then switch to a view of states.

For harder questions, such as 'Find the best-selling states within the worst-selling regions of coffee', users had to work first to find the worst-selling regions of coffee, and, focusing only on that region, find the best-selling states. In the FLUID interface, this was a natural extension to the workflow for the simple problem, but in the WIMP interface, people could not know a priori which region to filter out so some users were at a loss of what to do when they switched to the state view. Thus perhaps enforcing this workflow on simpler questions made it easier to know what to do on more complicated questions. This is borne out to a certain extent by the data which shows a more significant difference in performance by interface on more complicated questions but not with simple questions.

This finding is at least partially hinted by the preferences of the top two overall performers. They both preferred the WIMP interface (but they actually performed better on the FLUID interface). Post experiment interviews with these subjects confirmed that they already had a clear model for how to solve the problems and therefore did not need the extra workflow guidance that the FLUID interface effectively enforced, and in fact, felt constrained by that.

Limitations and Future work

These results all raise a number of additional interesting questions and issues that our current work cannot explain and will be explored in future work.

Questions on learnability and discoverability

In our experiments, each section had a tutorial phase where we demonstrated how the interfaces were meant to be used. There are still many questions on how to make gestures discoverable and memorable. While most users stated that they felt the FLUID interface was easier to learn than the WIMP interface, it is not clear whether people could sit down with no assistance at all and use the FLUID interface (how to discover that sorting was done through swiping, or filtering via flicking up or down). Ways of helping people discover the affordances of the FLUID interface is an interesting area to explore. Work by Bau & Mackay [2] as well as others explore techniques for assisting in learning gestures which would be useful in exploring in a more longitudinal study.

Comparisons using mouse based interaction

The two conditions that we explored are both on a touch device, and we do not compare these interfaces with operations on a mouse based/desktop display. Other research, including Forlines et al [6], found some encouraging results for multi-touch interaction on a tabletop display, but also found benefits for single-user single input using a mouse. It would be interesting to see if the results in this study hold for mouse based interaction in addition to touch.

Questions on scalability and additional features

Adding functionality in a WIMP type of interface is fairly straightforward (add a button or a menu), though at the risk of potentially overwhelming a user with complexity. In fact, several users already commented that they felt confused by where to start with a problem in the WIMP interface. While some research already exists to help ease the complexity of additional features (adaptive UI, ribbons, etc), finding good ways to add new functionality and features to the more FLUID type direct manipulation interfaces needs to be explored more. There has been encouraging work on a variety of gestural techniques (from marking menus to bimanual interaction) which would all provide interesting extensions to the current work.

Longitudinal studies moving back and forth between desktop and touch device

Users often move between different applications and different devices with different affordances. There is clearly some benefit of having a familiar interface metaphor between each application and each device. This study showed that there is also clearly some benefit of tuning the interaction to the affordance of a particular device. Future work should examine this problem in a more holistic fashion, perhaps as a longitudinal study where users need to move back and forth between the desktop and a touch oriented device.

Hybrids

In fact, many users felt that a combination of features from both interfaces would really present the best overall experience. Several users liked the sort button in the WIMP interface, but the filtering gesture in the FLUID approach. It would be interesting to consider ways to select the best parts of each.

Presentation

Finally, nearly all participants commented that they would appreciate the FLUID system when doing a presentation since they felt that each gesture they made would be apparent for an audience that was observing them work. This is an intriguing direction.

CONCLUSION

In this series of design explorations and studies, we set out to understand when porting a data visualization application to a tablet whether there were particular performance or other benefits like user enjoyment in crafting more gesture-specific interactions, as opposed to using a conventional WIMP UI. We studied this question by first exploring a design space for data visualization gestures, next by coming up with and testing candidate gestures for this domain, and finally by implementing a prototype with these gestures to support exploring real data sets. In a head to head comparison of a WIMP translated design and our gestural prototype, testing the same typical kinds of data questions for each case, we found that the majority of participants both preferred and performed better on the fluid, gesture-based, touch UI.

We have postulated that there are two attributes informing these results: the directness of hands or fingers on the object being manipulated, and second, the use of more constrained stepping through a problem that this gestural interface seems to enforce. As noted above, this latter result is somewhat counter-intuitive, but there may be some quality in the combination of gesture and problem solving process that seems a higher payoff than the more traditional spatial-memory/recognize UI that only experts seemed to prefer, and a small set of these. Understanding this interplay is a task for future work. In the meantime, the main design implication from the current work is that for this class of common data visualization interaction, it is certainly worth the design effort to deploy an interface that moves beyond the conventional WIMP based interfaces commonly used on desktops today and include those extra control panels only as optional expert panels.

In sum our work has made the following contributions:

- Proposed and tested a simple fluid gestural interface for choosing data, filtering, and sorting in data interactions.
- Expanded on a set of what may be further developed as benchmark questions for comparing data visualization interfaces.
- Compared the gestural oriented interface with a WIMP oriented spatial interaction to find a strong result in favor of the gestural oriented design.
- Proposed several further directions to explore why the FLUID gestural oriented interface may be preferred to the WIMP interface.

ACKNOWLEDGMENTS:

The authors wish to thank Asta Roseway, Hrvoje Benko, Roland Fernandez, Merrie Morris, and Mary Czerwinski for their comments and suggestions.

REFERENCES

1. Amar, R., Eagan, J., and Stasko, J., Low-Level Components of Analytic Activity in Information Visualization, *in Proc. of IEEE InfoVis '05*, Minneapolis, MN, October 2005, pp. 111-117.
2. Bau, O. and Mackay,W.. 2008. OctoPocus: a dynamic guide for learning gesture-based command sets. In Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST '08). ACM, New York, NY, USA, 37-46.
3. Baudel, T. 2006. From information visualization to direct manipulation: extending a generic visualization framework for the interactive editing of large datasets. In *Proc. of UIST '06*, New York, NY, USA, 67-76.
4. Bostock, M. , Ogievetsky, V., Heer, J. , D³ Data-Driven Documents, In *IEEE Trans. Vis. and Comp. Graph (InfoVis '11)*, pp.2301-2309, Dec. 2011.
5. Browne, J., Lee, B., Carpendale, S., Riche, N, and Sherwood, T. Data analysis on interactive whiteboards through sketch-based interaction. In Proc *Interactive Tabletops and Surfaces (ITS)*, November, 2011.
6. Forlines, C., Wigdor. D., Shen. C., and Balakrishnan, R. 2007. Direct-touch vs. mouse input for tabletop displays. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07). ACM, New York, NY, USA, 647-656.
7. Grammel, L., Tory, M., Storey, M. , How Information Visualization Novices Construct Visualizations, in Proc *IEEE Trans. Vis. and Comp. Graph. (InfoVis, '10)*, vol.16, no.6, pp.943-952, Nov.-Dec. 2010.
8. Guimbretiere, F., Martin, A. and Winograd, T., Benefits of Merging Command Selection and Direct Manipulation. Transactions on Human-Computer Interaction, 12(3), pp 460-476, 2005.
9. Heer, J., Robertson, G.G., Animated Transitions in Statistical Data Graphics, in *IEEE Trans. Vis and Comp. Graph. (Infovis '07)*., vol.13, no.6, pp.1240-1247, Nov.-Dec. 2007.
10. Hutchins, E.L., Hollan, J.D., and Norman, D.A. 1985. Direct manipulation interfaces. *Hum.-Comput. Interact.* 1, 4 (December 1985), 311-338.
11. Heer, J. and Shneiderman. B. Interactive Dynamics for Visual Analysis. *Queue* 10, 2, Pages 30 (February 2012), 26 pages.
12. Kolb, J., Rudner, B. and Reichert, M, Towards Gesture-Based Process Modeling on Multi-touch Devices. Lecture Notes in *Business Information Processing*, 1, Volume 112, Advanced Information Systems Engineering Workshops, Part 3, Chapter 8, Pages 280-293. 2012.
13. Morris, M.R., Wobbrock, J.O. and Wilson, A.D. (2010). Understanding users' preferences for surface gestures. *Proceedings of Graphics Interface (GI '10).* Ottawa, Ontario (May 31-June 2, 2010). Toronto, Ontario: Canadian Info. Proc. Soc., pp. 261-268.
14. Nielsen, J. 2012. "Jakob Nielsen's Alertbox: Browser and GUI Chrome." http://www.useit.com/alertbox/ui-chrome.html
15. North, C., Dwyer, T., Lee, B., Fisher, D. , Isenberg, P., Robertson, G. and Inkpen, K., Understanding Multi-touch Manipulation for Surface Computing, in *12th IFIP TC13 Conf. in Human-Comp. Interaction (INTERACT 2009)*, Springer Verlag, August 2009.
16. Shneiderman, B.1983. Direct Manipulation: A Step Beyond Programming Languages. *IEEE Computer* 16:8 (August 1983).
17. Stolte, C., and Hanrahan, P., 2000. Polaris: A System for Query, Analysis and Visualization of Multi-Dimensional Relational Databases. In *Proc. IEEE Infovis, 00*. IEEE Computer Society, Washington, DC, USA.
18. Wigdor, D., and Wixon, D. *Brave NUI World*. 2012. Morgan-Kauffamn.
19. Wickham, H.: [http://had.co.nz/ggplot2/] webcite ggplot2: An implementation of the Grammar of Graphics. 2008. [R package version 0.8].
20. Wobbrock, J.O., Morris, M.R. and Wilson, A.D. (2009). User-defined gestures for surface computing. In *Proc. of SIGCHI '09*. Boston, Massachusetts (April 4-9, 2009). New York: ACM Press, pp. 1083-1092.
21. Yee. W. 2009. Potential Limitations of Multi-touch Gesture Vocabulary: Differentiation, Adoption, Fatigue. Human-Computer Interaction. Novel Interaction Methods and Techniques. *Lecture Notes in Computer Science*, 2009, Volume 5611/2009, 291-300, ACM Press (2001), 9-18.