Developing Interaction Techniques for Small Multiples within a Multi-Touch Interface

Amy Jang  
Stanford University  
P.O. Box 17376, Stanford, CA 94309, USA  
isunji@stanford.edu  
+1 650 804 4763

Sakshi Agarwal  
Stanford University  
P.O. Box 13080, Stanford, CA 94309, USA  
sakshia@stanford.edu  
+1 650 450 9751

ABSTRACT
In this project we design new interaction techniques for small multiples in the setting of a multi-touch interface. We aim to introduce new interactions in these areas related to information visualization, and allow for new forms of analysis. We implement these designs, suggest other design possibilities, and conduct informal user studies.

Author Keywords  
Guides, instructions, author’s kit, conference publications.

ACM Classification Keywords  
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Interaction can make visualizations more powerful - allowing the user to more easily perform analysis tasks as well as opening up new forms of analysis. Our project looks at developing interaction techniques for small multiples within a multitouch interface. It is motivated primarily by two components.

Firstly, there is little work done in the space of developing interaction techniques for information visualization tasks for multitouch interfaces. Multitouch interfaces are becoming increasingly ubiquitous, and hence form an important space to explore for interaction work in infovis. Multitouch offers both new opportunities and challenges, and it is valuable to think about interaction working in a paradigm which is different from the current status quo of mouse clicks. To this end, we have aimed for our work to suggest possibilities for this space and nuances which might be important when making design decisions.

Secondly, small multiples visualizations are useful, but have scarcely been presented with interaction. In their static form, a specific arrangement of small multiples can convey trends about the data which might not otherwise be clear in a single merged chart or animated form. However, a static small multiple visualization is bound by a number of factors, including arrangement, size of the multiples, and details displayed. We have prototyped and implemented several interaction techniques that can be used for this visualization, and see that we have tapped a new realm that is under explored.

RELATED WORK
The simplest, motivating example of the use of interaction with ‘small multiples’ can be found in classic information visualization literature, Jacques Bertin’s “Graphics and Graphic Information Processing” [2]. In this book, Bertin describes a hotel manager who takes draws each graph of various hotel data which he has on a separate card. By doing so he keeps an “internal mobility of the image” and the “drawing is physically reclassable.” This means he is able to rearrange things and draw out trends with his perception, which he mightn’t have been able to see if all the graphs were arrange in a specific order along a single sheet of paper.

We looked at work which at the intersection of multi-touch interaction and information visualization, and found this space is still very open. Kristensson et. al [6] look at multito touch visualizations for a tagged photo interface and have interesting findings. They found that multitouch was used in about 10% of their interactions and was for complementary convenience more than necessity. Furthermore, multitouch was primarily used to organize a graph and separate many nodes at once. We take into account as we design our interactions in the small multiple space. We also learn from Chen, Yang et. al [4] that multi-touch users experience slow typing speeds and high error rates, and that a typing-free annotation approach would better help the visualization exploration process. Nacenta et. al develop interesting ideas related to the separability of multitouch gestures - which is a valuable direction to explore and keep in mind while designing we learnt, as we were implementing our interactions. More work in this area could help or at least inform the design related to issues which came up for our work in terms of gesture confounding [9].

We felt that work by Kosara provided a truly interesting and unique multi-touch data visualization, in the space of brushing in parallel coordinates [5]. Although this was done for indirect multi-touch manipulation, it was novel work and served as inspiration. Tominski et. al provide a good overview of where future work can serve to expand on the possibilities posed by new interactive and display technologies in the field of data visualization [13]. Their work on tangible views helped us think in the direction of
small multiples as we started to consider being able to act both within a space of views v.s. on a single view. Lepinski et. al focus on multitouch marking menus and present interactions which are interesting and well leverage these capabilities [8].

Research more broadly in the area of interaction and gestures was also relevant to our project. Wobbrock, Morris, and Wilson look at user-defined gestures and provide interesting insight into mental models related to gesture, and present a set of gestures which helped inform our work by providing gesture ideas and helping us take into consideration how users might relate gestures to tasks [14]. In more classic interaction work, Leganchuk, Zhai, and Buxton discuss the benefits of two-handed techniques [7]. This work serves as very interesting reference for future work, especially when thinking about which tasks can be performed in parallel or directly sequentially so as to minimize transactional costs from switching hands or tasks. North et. al claim that two-handed multi-touch surface computing provides the opportunity for gestures which are more analogous to physical interaction than the classic windowed interface [10]. They perform a study and find after users have been provided with a physical model, they perform tasks better in the multi-touch setting than the mouse setting. This helps strengthen the our choice of interface for implementation.

Finally, work more directly related to small multiples includes that by Archambault et. al who find that users perform faster on a set of 5 graphical comprehension tasks they posed using small multiples over animation [1]. Chen, Forsberg et. al pose work which is closest to what we have looked to develop in this space, and they do so in the more domain of scientific data which is heterogeneous, multidimensional, multivariate and time dependent, and also in a 3D environment [3]. They don’t use a multi-touch interface, however, which could be an interesting direction for this work. We keep our work in a 2D environment, and focus on general interaction techniques in simpler data domains.

**METHODS**

Our process can be outlined briefly as follows: Researching different task taxonomies, observing interaction techniques, finding opportunities for new design, reviewing existing literature, tightening the design space, prototyping the design, iteration, informal user testing and feedback.

The original, broader direction we started out with was to develop new interaction techniques for data visualizations in the multi-touch space. Although we narrowed our scope, the groundwork we did helped motivate our design. We started out by researching on different task taxonomies for information visualization. We came across many different breakdowns, whereby two stood out to us:

Taking these into consideration, we then brainstormed on what interaction techniques are used in various types of visualizations, including graph, map, network, and scatterplot. We compared what interactions are currently be available for these in a mouse-based environment versus a multi-touch one. This exercise helped us start to find opportunities for new design, as we thought about what tasks were easy or difficult to do in a multi-touch setting. A few of the ideas we originally had included: magnified targeting to help with finger occlusion problems, a three-finger zoom & select mechanism, and two-finger filtering across a graph.

After probing the existing literature further and evaluating opportunities, drawing in from the related work mentioned above, we found that tasks which lend themselves well to multi-touch visualizations had certain characteristics. These characteristics included tasks which have an element of ‘parallelism’ to them, that is, they can either be performed in parallel or involved manipulating multiple dimensions simultaneously. Also, multitouch lends itself well to performing tasks sequentially without necessarily having to create modes or use tools, because the gesture set is larger than mouse-click, and could also be useful for tasks which have gestures that would be natural to use with more than one contact point. We also noticed that small multiple visualizations had been treated primarily statically, and that this form seemed to have possibly valuable interactions in a multi-touch setting given the power of direct manipulation and the fact that we could think of each multiple as an object. These aspects pushed us towards designing interaction for the small multiples space, which fit an interesting problem with a lot of potential in this setting.

We detailed out various design possibilities and then chose a subset to prototype, these details are in the results section below. For our implementation, we chose to build a native application for the iPad using Objective-C to create our visualizations, because of the various touch gesture recognitions it provides. The application was built such that it will takes any well-formatted json file with sets of data which share the same x and y dimensions, so that they can be outputted into small multiples. The data will be parsed and outputted to bar charts in small multiple format. The application also used open source code for manipulation of grid view[11]. The biggest technical achievement was distinguishing different gestures and able to provide right interaction for the gesture. Although iOS seems to provide

<table>
<thead>
<tr>
<th>Authors</th>
<th>Taxonomy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shneiderman</td>
<td>Overview, zoom, filter, details-on-demand, relate, history, and extract</td>
</tr>
<tr>
<td>Yi, Kang, Stasko, and Jacko</td>
<td>Select, explore, reconfigure, encode, abstract/elaborate, filter, connect Below</td>
</tr>
</tbody>
</table>
different gesture recognizers, it posed some challenges:
touches only respond to the most frontal view, gestures get
confounded when used within the same view, and the
gesture recognizing methods will get called after every
movement of touch. To resolve this issue, we used a single
view to recognize all the gestures and an algorithm to
distinguish between the gestures: we recognized a gesture
with the number of touches, the length of a touch, the
location of touches, and the sequence of touches and
classified it with the combination of the information above.

As we were implementing our design, some iteration
naturally took place as we were testing things out for
ourselves. Finally, once we had a working prototype, we
conducted some informal user studies and gained valuable
insight into how the application could be improved.

RESULTS
Final Implementation
We came up with a number of design ideas and chose to
implement those which were compatible each other, as well
as fit within the scope of this project. We detail the design
space we explored below this section. The final interactions
which we implemented were: Rearranging multiples,
merging multiples, brushing and linking for selection, and
zooming in the multiples space.

We provide brief descriptions and rationales for our design
decisions, and for each category further justifications can be
found within the design space section below.

Rearranging multiples: we implemented this as a touch and
drag action, with a circular indicator to which the user’s
finger is matched. This made sense from the perspective of
making sure the user wasn’t occluding the graph as they
were moving it around, and also helped differentiate the
selection gesture we chose. Being able to change the order
of multiples allows the user to pick out different trends by
placing graphs next to each other and also make
comparisons between graphs for which they’d have to
original scan back and forth between different ends of the
visualization. Also, from our informal user testing, we saw
that users preferred the comfort of grabbing each multiple
by directly clicking on it than the advantage of seeing the
graph by grabbing the corner, which we had originally
implemented. We merged these benefits by allowing users
to grab a multiple by long clicking on any area of the
multiple, and having application slightly move the graph
such that the corner lines up with the touch point so that
user’s fingers will not occlude the graph while dragging.

Merging multiples: Originated from when the user was
rearranging and dragged one multiple directly on top of the
other, at which point a yellow highlight would show
indicating the graphs would be merged upon release. The
highlight helped the user to understand when a merging
could take place. The power of merging multiples meant the
user could at once pick out differences and make more
precise comparisons they couldn’t have made otherwise.
For this particular data visualization, histograms, we chose
to make the background one a line chart and the foreground
a histogram upon merging, because it was clearest for
comparison of two charts (transparent histograms on top of
each other or ones next to each other didn’t result in clear
comparison). For clearer understanding of transition and
comparison, the application could draw the line graphs such
that each segment along x axis contain vertical lines that
suggest each data points that match with foreground bars.

Brushing & linking for selection: we implemented this as a
left-to-right drag action, and also allowed the user to toggle
the left and right ends of the selection. When a selection
was made on one graph (we allowed selections within a
time series), then all the other multiples would also be
highlighted with the same selection. Being able to see this
helps focus user attention on this particular subset of the
data, serving as a ‘filter’ and more easily to pick out trends, in this case over a particular time interval. Furthermore, selection provides integer values on the graph, for further detailed information. For smoother and more powerful selection, selection could be made from any direction, from right to left, left to right, or top to bottom, where vertical direction allows selection of data in y range.

Zooming in the multiples space: The user can zoom into a set of multiples, and change from a visualization of, for example, 20 multiples and just expand 6 of them. We implemented this as a four-finger zoom - analogous to the classic zooming one uses for the iPhone with two fingers, but doubling the finger count. Our original motivation for this implementation had been we’d use the classic zoom (‘pinch’) to zoom just within a graph, and use four-finger for the multiple space zoom. Doubling the fingers was a way to differentiate between these different ‘levels’ of zooming. From informal user studies, this feature was found to be powerful but awkward in terms of interaction at times. It is crucial to distinguish zoom within a graph and multiple space, so the application could have a switch to indicate which zoom action to be recognized for pinch gesture.

Figure 2. Merging multiples

Figure 3. Brushing & linking for selection

Figure 4. Zooming in the multiples space
**Design Space: Conceptual Ideas**

We laid out the following as interesting areas for which to develop multi-touch interaction with small multiples: Reconfiguration of multiples, connecting multiples, generating views of additional interest, changing the encoding of multiples, and performing aggregation tasks on the data. We also noticed that they interactions could be initiated by various forms of gestures varying number of touch points (e.g. single, four-finger), length of touch (e.g. short tap, long click), sequence of touches (e.g. tap, double tap), and of course these things in combination motion (e.g. dragging, pinching). Here we describe some of the various design ideas we had.

To this end, we explored the following ideas and justifications:

Reconfiguration of multiples: Here we looked at reordering, whereby the user can rapidly switch between the arrangement of the small multiples. Certain layouts of graphs can convey information about trends and relationships between the data, however, in a static setting we are limited to one of these. We implemented this using a single-finger drag & circle indicator, but other possibilities could have included a different number of fingers & dragging, or changing the length of the initial touch before dragging. Another idea could be to take the idea of rearrangement further, and allow the user to create ‘groupings’ by for example annotating or using space. Also, we can foresee functionality which allows the users to save these different views, or allow the user to automatically create an arrangement based on a predefined function.

Connect: This is a powerful task to perform in a small multiple space with interaction techniques, because it allows users to really see and understand the relationship between their data. Since we have the multiple views here we looked at brushing & linking for selection. What we implemented is detailed in the previous section, but other ideas could include enabling users to change the actual ‘encoding’ of the small multiples simultaneously. We only implemented linked selection along one dimension, time, but the user could possibly select along other dimensions or have some kind of linked filtering function as well. Another possibility is being able to zoom in within a particular graph and have the other multiples follow suit - we had implemented this as the classic two-finger zoom but found that it was confused with other gestures in the interface.

Generating views of additional interest: We looked at merging multiples in our implementation. Other possibilities could have included being able to dynamically add or remove multiples. One practical application this could be useful in is a stock market application; whereby the user could easily generate different sets of stocks, and quickly do analysis on them, even be able to generate multiples which for example are a ‘moving average.’ In terms of implementation ideas, one could imagine the use of brushing & linking for selection. With the data, one could select along other dimensions or have some kind of linked filtering function as well. Another possibility is being able to zoom in within a particular graph and have the other multiples follow suit - we had implemented this as the classic two-finger zoom but found that it was confused with other gestures in the interface.

Generating views of additional interest: We looked at merging multiples in our implementation. Other possibilities could have included being able to dynamically add or remove multiples. One practical application this could be useful in is a stock market application; whereby the user could easily generate different sets of stocks, and quickly do analysis on them, even be able to generate multiples which for example are a ‘moving average.’ In terms of implementation ideas, one could imagine the use of brushing & linking for selection. With the data, one could select along other dimensions or have some kind of linked filtering function as well. Another possibility is being able to zoom in within a particular graph and have the other multiples follow suit - we had implemented this as the classic two-finger zoom but found that it was confused with other gestures in the interface.

Performing aggregation tasks on the data: We can imagine that being able to perform simple calculations such as average, sum, range, growth rate, etc. would be very useful to a user performing analysis. One real world example, which we were given as feedback, would be for consultants presenting to executives - such tasks can be more cumbersome to perform using traditional tools because one has to interact directly with the data and re-generate visualizations; also there is no template for arranging these visualizations. In this case, we would provide a way to apply these operators directly to a visualization, making it quick and easy to do in an executive setting and allow users to focus on analysis rather than direct data reconfiguration. To this end, one viable model we can think of, borrowing from interaction work, is making this a tool-based system. That is, displaying tools which can be applied to graphs, even merged graphs, and can return to the user such calculations. As far as direct interaction with the graphs for aggregation tasks goes, one design idea is to use two fingers to touch points on a graph and have the slope be displayed.

**Informal User Testing**

We tested our application with four subjects who are undergraduate students, and also got feedback during a poster session. With our four subjects, we conducted an informal walk-through, which roughly followed this format:

Tester briefly introduced the visualization, data set. Then asked the subject to pick out any interesting trends they noticed off-hand. Then tester stepped through the interaction techniques available to the visualization. After this, the tester asked a series of questions which included asking the subject to “Pick out the country with the fastest rate of growth in time period X” and “Choose two countries with similar growth in time period X.” Finally, the tester surveyed the subject and asked about experience using the application, and whether the interaction changed analysis. From this informal study, we came upon several interesting results.

There were a few common threads, behaviors and breakdowns, which came up during testing. Users would immediately notice in the static visualization that mobile subscriptions were increasing across the board. To answer the questions that we posed, users were quite actively using the interaction facilities. They would have an initial hypothesis as to the answer, but the interaction techniques would allow them to make more precise claims and validate their answers. They would typically rearrange the multiples,
DISCUSSION

Our system has introduced new interaction techniques which will help push forward both the space of creating a unique connection between analysis tasks and physical gestures in touch space, and also that of developing such techniques for a small multiples interface. We suggest that multitouch platforms can be a powerful medium for data analysis and that they offer a range of manipulations which are more seamless and intuitive. Furthermore, we provide a platform in which analysis tasks are conducted directly on visualizations with instant feedback, rather than having the user go back to the data and wait on the feedback loop. We provide a new way of performing analysis by designing rearrangement and merging in the small multiples space - for example, allowing the user to take advantage of the best parts of a single, completely merged graph and different interesting layouts in a purely static multiple space. We provide a solid prototype which can be developed further into a powerful visualization tool. We also address various possibilities in the design space so as to fuel further thought in this area.

FUTURE WORK

We hope that future work address these new areas of at the intersection of interaction and information visualization in analysis and multi-touch settings. Our system in particular can be extended by further building upon the findings we had from initial user studies and during the implementation process. It could look at understanding which gestures are compatible and unlikely to be confounded, and it could also borrow more heavily from work related to multi-touch gestures and mental models. It would be also great for more exploration related to the ideas we mention in our section on the design space.

Particularly compelling possibilities from this include: building in more connected manipulations, like the within multiple zoom we originally had; being able to generate multiples dynamically from selection; performing aggregation tasks on the visualizations (sum, average, rate, etc.); introducing interactive techniques for small multiples of data with higher dimensionality; and even being able to look at multiples from different data domains at the same time, introducing spaces to drag and for collections for multiples. We foresee interesting applications being developed for more specific data domains using these general techniques which we introduce, and are excited to see where this work can go.

ACKNOWLEDGEMENTS

Thank you to Jeffrey Heer, Jason Chuang, Sanjay Kairam, Wendy Mackay, and Michel Beaudouin-Lafon for providing us with discussion and insight which greatly helped to inform our work.

REFERENCES

1. Archambault, D., Purchase, H., and Pinaud, B.
   Animation, Small Multiples, and the Effect of Mental


11. Quatemain, A. AQGridView. Available at https://github.com/AlanQuatermain/AQGridView


