FACULTY OF ENGINEERING & IT

The Design of Touchable Interactions for Data Visualization

Thesis of Master by Research Degree

Shizhe He

Supervisor: A/Prof. Mao Lin Huang

Year of submission: 2016

Acknowledgement

On the completion of my thesis, I would like to express my deepest gratitude to all the people whose kindness and advice have made this work possible.

First and foremost, I would like to thank my supervisor A/Prof. Mao Lin Huang, for his expertise and patience. His effective advice and shrewd comments kept my research and the thesis in the right direction.

Thanks to my research fellows Wenbo Wang, Jinson Zhang, Jie Hua, and Phi Giang Pham for their invaluable advice and encouragement. And I would like to extend my thanks to Bowen Xu and Lin Zhu for their help and support during the implementation and tests.

Thanks to my parents for their unconditional support and trust.

CERTIFICATE OF ORIGINAL AUTHORSHIP

This thesis is the result of a research candidature conducted at University of Technology as part of Masters by research degree. I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as part of the Masters by research degree and/or fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student:

Date: April 22, 2016

Table of Contents

List of Illustrations and Tables

Abstract

Natural User Interfaces (NUIs) are gradually becoming universal in modern user interface design. Researchers are attempting to apply novel Computer-Human into the design of interactive data visualization. Touchable interaction, an affordable and relatively mature member of NUI, has drawn much attention in today's design of personal computers. And it has been proved to be effective and efficient in accomplishing some certain tasks, for example, pinch and stretch to zoom or scale. However, the values of Interactive Tabletops and Surfaces (ITSs) that could offer to the field of data visualization still remain unclear. The derived questions become how to best leverage touchable interaction into data exploration, navigation and manipulation.

We reviewed over fifty papers and proposed a classification of touch interaction styles, based on how fingers interact with data. And then we organized the literature review according to the proposed category. Following this, we listed where touch interaction outperforms desktop and mouse, along with where the limitations and tradeoffs are. Any combination of the merits or limits could become a research question, but we are only focusing on the ones applicable with tablet-sized touch screen.

As a pioneer research, we designed a set of gestural commands to replace keyboard-mouse interactions in DA-TU. Clustered graph visualization, such as DA-TU, has been successfully applied in the field of large scale relational data visualization for data analytics. But it was very inefficient in interactions, in which a control panel was required to swap among eight different modes of operation. The proposed design was implemented on iPad, and we conducted user tests to demonstrate its higher efficiency and better user experience.

Mouse hover is a common and effective solution in data visualization applications to deal with the "representation problem". We proposed an interaction technique named Radial Cursor to address this problem on touch screens. It stimulates 'mouse hover' interaction with radial hierarchical data, coupled with a level switch, Radial Cursor provide intuitive and effective controls of 'hover to see more detail' or 'hover to highlight' over every data node. We also applied real dataset to show that Radial Cursor helps user to quickly get a general sense of an unfamiliar dataset, or review an already-known dataset.

We integrated the methodology from visualization project design study, industry of Human-Computer-Interaction, and data analysis process. Hopefully our work can contribute to clarifying why we need touch displays to support data visualization, and how to best leverage touch interaction in visual data analytics tasks.

Keywords: interactive data visualization; multi-touch; natural user interface (NUI); playful data analysis; design research

Chapter 1 Introduction

1.1 Description of Keywords in Title

Information Visualization is the use of computer supported, visual representations of data to amplify cognition [1]. It is an aid for discovery, decision making and explanation of information. There are three main components in one visualization, data, data presentation and interactions. In our research, we mainly focus on abstract data visualization, such as air quality records. Data presentation is how to 'concretely' present the data with graphs and plots. Scientists have put a lot of efforts in this area, and they have delivered plenty of solutions and are still working optimizing them. Interactions are our main interest, which we are going to discuss in the following description.

Touchable Interaction refers to a specific type of human-computer interaction (HCI), which constitutes the rising technique, NUI. In terms of the visualization, interactions such as navigation, change of encoding assist human with information perception, knowledge retaining and decision making. And we aim to figure out how to make touch interaction best assist this process. Contrast to the traditional desktop-mouse interaction, with the elements of Windows, Icons, Menus, Pointer (WIMP), we are also interested in where touch outperforms WIMP, and also how to complement touch and WIMP together.

Design Research refers to embedding design techniques, context and insights into the process of research, including work concerned with the context of designing and research-based design practice. In the context of Information Visualization, design studies are a particular form of problem-driven research, where the goal is to work with real users to solve their real-world problems [2]. In the industry of HCI, it refers to the interactive cycle of plan, observe, design, prototype and test. Different than data science and other traditional scientific research, solutions in design research are proposed based on embodied, observable phenomena related to the problem space.

1.2 Background

1.2.1 Development of HCI and Touch Interaction

The concept of HCI was first raised in 1940s, when Vannevar Bush wrote of how future technologies could augment human intellect. And that was even before the personal computer came out. Then Vannevar invented memex in 1945, which might be the first prototype of modern smartphone with camera functionality. The design sense of providing effective interface to computers began with Grace Hopper's first compiler in the early 1950's. And then the born of graphical user interface (GUI) counts as a milestone, which makes input and output a lot more intuitive.

In the process of manipulate and interact with data, user's mental model of understanding data keeps evolving and improving. Specifically, interactions could benefit data visualization in the following aspects:

1) Mitigate the conflict between limited display space and exploding data volume. Ben Shneiderman [3] proposed a basic guideline of exploring large dataset, or Visual Information Seeking Mantra as himself refers it: overview first, zoom and filter, then details-on-demand.
2) Interaction can indeed facilitate the process of understanding and analysis of data. Interactions are an especially essential part of data analysis systems, which are built to provide tools and platforms for the users to explore the data and then gain insights, instead of directly consuming insights.

The capacitive touch system was used as early as 1967 in an air traffic control system [4]. And the touch technique keeps developing in the next several decades. Apple certainly contributes a lot by making touch-enabled smart phones and tablets universal. Based on the screen size, touch surfaces can be grouped into:

- Large: screen: tabletops, whiteboard, and wall-sized screens
- medium screen: desktop sized screens, tablets
- small screen: smart phones and other small mobile touch devices

We narrowed it down to designing interactive visualizations for medium sized screen. Because large screens involve collaborative and pixel related problems, and small screens constrain the number of fingers used. To focus on our aims, and also restricted by research phase shortened (downgrading to research masters), we use tablets as main target device.

1.2.2 Why Touch/Why not Touch?

This problem is inspired by the question asked by one of my friends. After my introduction of my topic, she asked, what kind of data is best for touch interaction. That is, why do we need touch to interact with data?

Isenberg et al. [5] proposed the research questions regarding the intersection of visualization and interactive tabletops and surfaces (ITS). But it is still unclear that how can ITS contribute to efficient, effective, and satisfactory data analysis with visualizations. After surveying existing systems and designs, we listed unique features that ITSs can offer but WIMPs do not - in other words, where ITSs outperform desktop and mouse -which include:

1) Physical metaphor
2) Increased input bandwidth
3) Additional degree of freedom to express intentions
4) Panel free
5) Effective zooming/scaling

6) Trajectory enabled
7) Kinetic feedback
8) Collaborative work
9) High resolution
10) Anywhere, anytime, anyone
11) Sensors

As Bill Buxton puts it, everything is best for something and worst for something else [6]. The challenge with new input should be finding devices that work together with the mouse, or things that are strong where the mouse is weak, thereby complimenting it. Therefore we list where desktop outperform touch screens as well.

1) Worse precision - 'fat finger problem'
2) Education
3) Occlusions
4) No self-explaining
5) Low typing speed and high typing errors
6) Hard to implement light weighted highlight

The problem is to embrace the challenges of touch's limitation and make biggest benefits from its strength. We will discuss in detail about the above listings in the Chapter 4.

1.2.3 Background of DA-TU: Clustered Graph Visualization

Relational datasets (graphs) raised in Information Visualization applications are often with large scale: thousands, or perhaps millions of nodes. However, the classical graph drawing methods could successfully deal with only a few hundred nodes due to their computational complexity. Many alternative solutions proposed attempting to address this problem. Most of them are adding two mechanisms: 1) Advanced Navigation, 2) View Abstraction on the top of the visualization to handle the large scale data visualization.

The above strategy has been instantiated in a prototype system called DA-TU (meaning "big map" in Mandarin). It can be accessed through http://www-staff.it.uts.edu.au/~maolin/jgaa_demo/ /jgaa_demo.html. DA-TU attempts to overcome the problem of visualizing large graphs that had proceeded in two main directions:

- Clustering*:* Groups of related nodes are 'clustered' into super-nodes. The user sees a 'summary' of the graph: the super-nodes and super-edges between the super-nodes. Some clusters may be shown in more detail than others. It is the use of clustering structure for achieving 'visual abstraction', rather than 'overview diagrams' that is used by some web navigation facilities [13]

- Navigation: The user sees only a small subset of the nodes and edges at any one time, and facilities are provided to navigate through the graph.

Four layers of interaction are described as below:

1. In *graph layer*, user's operations are optional. That is, in most of cases, datasets (nodes and edges) are automatically read from the database through system inputs. However, it would be good to provides users with some optional operations allowing users to manually add (or delete) nodes and edges to (or from) the visualization. But they are out of our concerns in this case.
2. In *clustering layer*, there are two elementary operations:
    a. C_create_cluster(S: set of nodes): creates a new cluster of the set S of nodes, and returns an identifier for the new cluster.
    b. C_destroy_cluster(u: node): The children of u in the cluster tree become children of their grandparent, and u is deleted. This operation may not be applied to the root, and applying the operation of a leaf u merely deletes u.
3. In *abridgement layer*, there are three elementary operations:
    a. open_cluster(u): This is defined whenever u is in the basis U of the current abridgment. It replaces u by the children of U.
    b. close_cluster(u): All children of u in the basis of the current abridgment are deleted (from the basis), and u is added to the basis U.
    c. hide(u): If u is in the basis, then it is deleted from the basis U.
4. In *picture layer*, there are four elementary operations:
    a. Move(u: node): this is the usual operation of manually moving a node in the picture.
    b. Gathering: This operation moves sibling nodes closer together. In practice, this tends to make the cluster boundaries disjoint.
    c. Scaling: This operation can be used to increase the size of the picture to allow more detail to be seen, or reduce the size of the picture, to fit the screen.
    d. Layout operations: These are classical layout functions, which produce a picture P' of a clustered graph C' from a picture P of a clustered graph C. The layout operation is used to produce a sequence of dynamic pictures.

Despite DA-TU has been proved to be effective in dealing with large scale data visualization, its navigational operations across different layers are very inefficient. Because the ordinary WIMP interface provides users with very limited HCI vocabularies, only including mouse-click, mouse-double-click, mouse-press, etc. which are far not enough to be used for identifying the types of navigational operation in DA-TU. There are eight types of operation: create, destroy, open, close, hide, scale, gather and dismiss.

We have to design a Control Panel, setting up eight corresponding operational modes to identify these different types of operation. Before performing a particular operation, we need to swap to an appropriate operation mode by ticking the corresponding check-box on Control Panel, see Figure 1-1. So we designed a new set of touch gestures to address this problem. Chapter 5 will discuss this topic in detail.

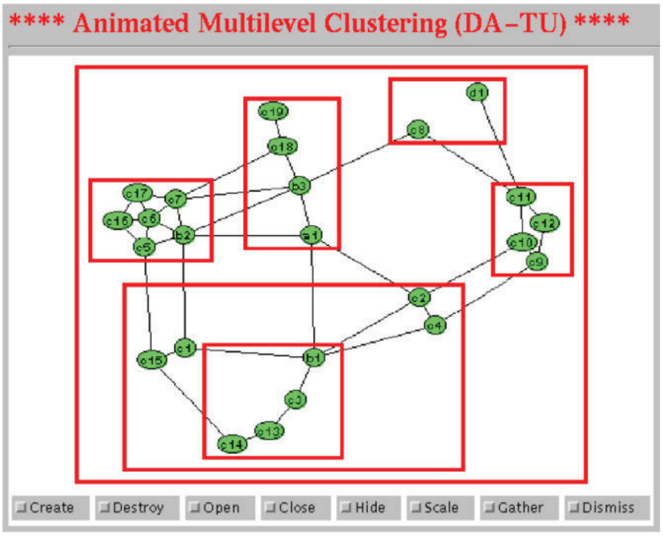


Figure 1-1 Navigating a clustered graph *C* in DA-TU.

By adjusting the inter-cluster Spring Forces in '*gather*' mode, the overlaps among clusters are eliminated.

1.3 Aims, Objectives and Significance

Stakeholders

- Visualization users: general public and domain expertise

  The users of visualization could be varied according to the application context. We simply divide them into two groups based on their background knowledge. General public might access the visualization through posters, blogs, and online communities. The data is whatever they are interested in, such as census data, survey data, or even personal data like monthly electricity usage. People quickly plays with the data and get a sense of what it is about. On the other hand, visualizations for domain experts require professional knowledge ahead of time. Besides, the data are often more complex and with larger amount. And experts would spend days, weeks or even years exploring it to get profound conclusions, driven by the questions like which medicine will works better to cure a certain disease.

- Visualization providers: researchers and industry workers

  Researchers and industry workers are actually related as interactive visualization. In fact, many research works themselves are published as applications. And the two groups both need user participation to evaluate their work.

Aims

i. Help users and designers understand the benefits and limitation of touch interactions in data visualization.
ii. Provide effective and efficient control over the visualized data.
iii. Playful and enjoyable user experience of interacting with data.

Objectives

i. To explore the benefits and limitations of combining touch interactions into the data visualization, in contrast of the WIMP interfaces. And to investigate the unique values touch interaction could provide to insights getting from visualizations.
ii. To improve the existing data visualization applications/designs with touch interactions. And there should be an executable prototype for usability testing and validation of touch's benefits.
iii. New data visualization applications/designs exclusive to touch interactions. The design needs to address touch interaction's limitation, and utilize its benefits. Reasonable user validation should be provided.

Significance

i. Determine where touch interaction outperforms WIMP interface, and also contribute to how to complement them together;
ii. Implications and reflections for the research and design of visualization with touch interaction.
iii. Bigger chance for the user to find the answers through more effective and enjoyable interactions with data.
iv. Bring innovative interaction technique to the community. Because it is versatile and cheap to implement, it can be applied to both research and business scenarios.

1.4 Outline of this Thesis

In Chapter 2 we reviewed the styles and designs of use of touch interaction in data visualization. We collected and investigated relevant papers from CHI, TVCG, AVI, ITS, InfoVis, and INTERACT. Different from the research overview conducted by Isenberg and Isenberg [7] , we mainly concentrate on touch interactions themselves and their correlation between data and contextual environment. And we developed a category of 'touch-data' styles during the reviewing process.

Then we discussed research methodology in Chapter 3. We are applying a combined method in this thesis. In terms of design, we introduced design research frameworks in academic field and design process of industry. And then in the following section we investigated methodologies of scientific data analysis, which we referred to during the data acquiring and testing tasks design. Finally we presented the methods for evaluation design and measurement definition.

Then the following three chapters are the results during my candidature period.

We presented the merits and limitations of touch interaction in Chapter 4. The findings are also based on the investigation of relevant papers. This research is first initiated to address one of the research questions Isenberg et al.[5] proposed at ITS: how can ITS contribute to efficient, effective, and satisfactory data analysis with visualization? And we also considered the tradeoffs and limitations followed by values.
In Chapter 5, we described the design, implementation, and user evaluation of DA-TU on ipad. DA-TU is an interactive graph visualization prototype on desktop as 1.2.3 introduced. We designed a set of multi-touch gesture vocabulary to replace the old WIMP interface. According to the testing result, adopting such set of rich HCI vocabulary achieves better efficiency as well as better user experience.

Chapter 6 introduced a new efficient tool on touch screens, named Radial Cursor. Radial Cursor stimulates 'mouse hover' interaction with radial hierarchical data. Design and implementation process, and use cases are provided.

Then at last Chapter 7 is about conclusions and future works.

Chapter 2 Literature Review

The reviewed works are designs and applications collected from different communities, including computer human interaction (e.g., CHI), applications on touch devices (e.g., ITS) and information visualization (e.g., InfoVis). Due to the fact that visualization on touch screens is still a relatively new topic, the literatures cannot be easily classified. Inspired by the question 'what kind of data is suitable for gestures', we developed a category of 'touch-data' styles during the reviewing process, see table 2-1. Different than existing classifications, we considered both data's innate features and the action's functioning context, and mainly focused on the connection between them. That is, the classification describes 'how fingers interact with the data of visualization applications'.

Table 2-1 Categories of touch action styles

Relationship with data	Descriptive category names		Typical example
Data dependent (dd)	Direct-on-data manipulation	Reality based	drag to move
		Beyond reality	stretch to enlarge the item
	Virtual instrumental interaction		Use lens to magnify
	Unique techniques (data specific)		pinch to zoom in and stretch to zoom out
Data independent (di)	Touch panel		Widgets for changing the data encoding
Mixed	Multi-modal interaction		dd: use stylus to link the nodes
			di: put token on to show the control panel
	Gestural trigger		dd: long press to hide an item
			di: swipe to navigate between views
	Drawing gestures		dd: circle groups of items to select them
			di: draw a question mark for help

In the following sections, we first discussed about a very limited amount of existing taxonomies of touch styles used by visualization applications. And then we categorized the related works into our categories and introduced them respectively. These works also play a role of example and evidence of our proposed categories.

2.1 Existing Taxonomies of Touch Styles

In the previous works, gestures are often classified based on their physical feature (e.g., number of fingers being used)[8], [9], or physical environment (e.g., surface type[10]) of visualizations. A limited number of works were about classification of touch action styles combined with visualization. We are trying to discuss deeper intersection between visualization and touch, with the goal of best leverage touch interaction for insights.

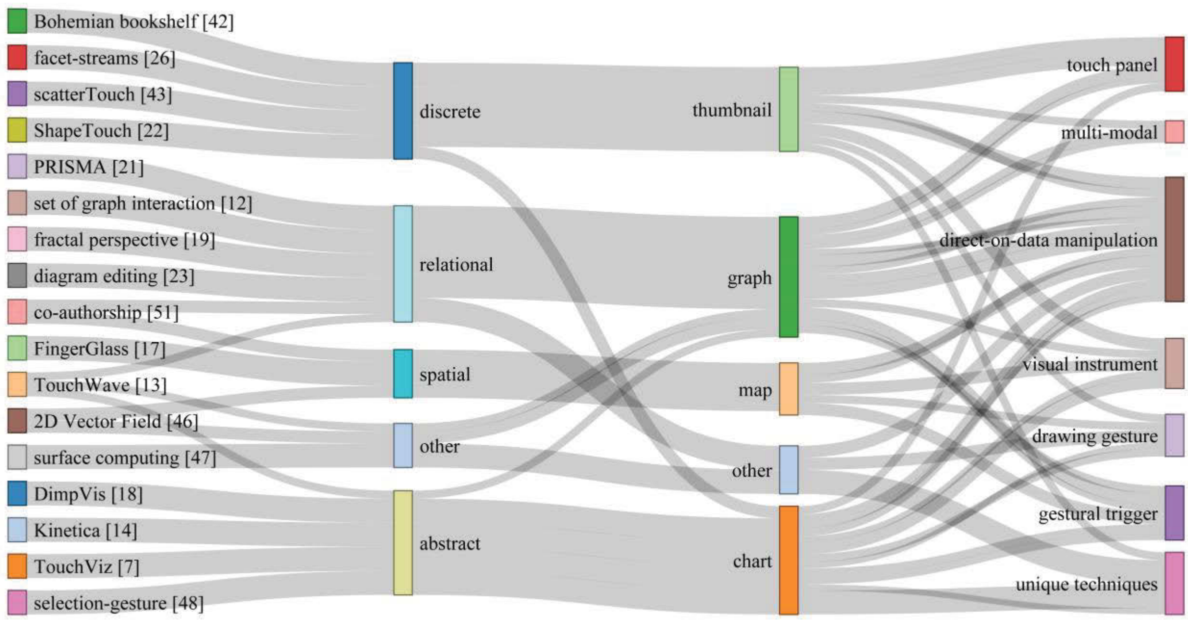


Figure 2-1 Visualization of correlations between investigated works (1st axis), data type (2nd axis), data representation (3rd axis) and our proposed classification styles (4th axis)

Yee [11] differentiate the gestures according to their directness, a 'direct gesture' manipulate on-screen objects directly, and an 'abstract gesture' do things 'indirectly', such as navigate through an information hierarchy. But the definition of 'indirect' is vague. And it is obvious that indirect gestures would significantly outnumber the direct ones, if we just categorize the gestures not performed directly on objects as abstract gesture.

Drucker [12] organized the sets of actions into the following categories at the first stage of designing TouchViz: 1) actions that mimicked the way a mouse is currently used (e.g., primarily single clicks on objects); 2) representative gestures that involve drawings either shapes or text on the screen; 3) actions that would be more convenient with touch (e.g., flicks, swipes); 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). The second category could be mapping to *drawing gesture* in our classification, while the third one is similar to gestural trigger.

Sadana and Stasko [13] roughly categorized the interaction tasks into data-driven and view-driven, and then listed the options to operate scatterplot data, which are organized based on seven categories of interaction techniques generalized by Yi et al. [14]. And Walker [15] listed InfoVis applications on interactive surfaces according to Yi's task-based categories as well. These classifications are task-specific or data-specific, while we are trying to do it in from another aspect. The seven general categories of high-level interaction proposed by Yi are: 1) Select, 2) Explore, 3) Reconfigure, 4) Encode, 5) Abstract/Elaborate, 6) Filter, and 7) Connect [14].

2.2 Data dependent touch action styles

2.2.1 direct-on-data Manipulation

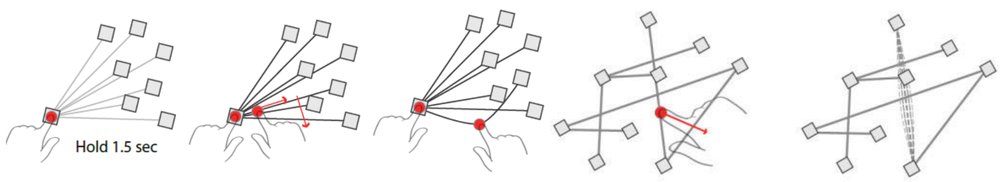
Direct-on-data manipulation is fingers directly on data objects, rather than virtual tools on objects (virtual instrumental interaction), or stylus on objects (multi-modal interaction). This might sound similar to Shneiderman's direct manipulation [16]. But Shneiderman's 'direct' is proposed comparing to the previous command-line interface, while our 'direct' is literally means fingers directly on the on-screen data objects. That is why we put 'direct-on-data' before 'manipulation' for differentiation.

Directly manipulating data item enabled by interactive surface could be one of the greatest values it offers to visual data exploration. And we further divide the direct acting on data items into two sub-categories, in terms of whether it is just a mimic of reality or beyond reality:

- Reality-based: physical simulation (e.g., drag to move)
- Beyond reality: 'magic' power (e.g., stretch to enlarge the item)

Reality metaphor enables us to manipulate data items as we treat them in physical world. We could move one or several items to any desired position, or rotate them to any angle easily.

Multi-touch graph interaction is a relatively popular area regarding data visualization in interactive surface. The set of graph interaction techniques proposed by Schmidt et al. [17] includes an action to pluck a single edge out of a congested area with two touches. It mimicked the solution we are very likely to take in real world, when faced with the problem to identify a single wire of a certain target application out of a mess of wires. Likewise, a short flick gesture on an edge would strum it as people expect it, since it is the same if we flick a string in reality.



(a) (b) (c) (d) (e)

Figure 2-2 pluck a single edge (a-c), flick an edge (d-e) proposed by Shimidt et al. [17]

Direct manipulation on surfaces could help you 'realize' the fantasy that user can duplicate an item simply by some spelling and a wand, or enlarge or shrink objects by stretching them with bare hand? You could first tap the source, and then the destination where you wish to place the copy [9], the duplication is done.

When inspecting hierarchy stacked graph, you can stretch or pinch a layer vertically to see its contained sub-layers or hide them [18]. New objects can also be created by a simple gesture, e.g., a scatterplot can be created just by placing two fingers to indicate the diagonal corners [19].

2.2.2 Virtual Instrument

The effectiveness of virtual instruments in WIMP interface such as sliders, cursors or lasso has already been discussed by Beaudouin-Lafon [20]. Touch-based interfaces enable some unique actions that can be more convenient, and easier to understand and perform than those on traditional WIMP interface.

For instance, In order to facilitate comparison of stacked graph representing numeric flow data (e.g., box office results for movies), a clever tool vertical names ruler [18] is introduced. The ruler is bound to the finger. As the finger moves around, ruler shows the size of all layers at the finger's current position. See figure 2-3 (right).

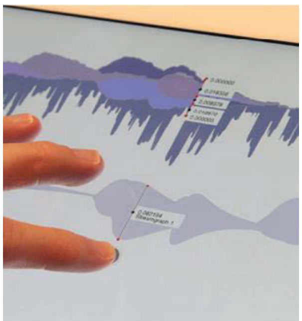


Figure 2-3 left: ruler from TouchWave [18] right: virtual lens from FingerGlass [21]

Meanwhile, virtual interactive lens have attracted much attention [22]. For example, FingerGlass (see figure 2-3 (left) [21] is designed to improve the precision during execution of tasks on multi-touch screens, using bimanual technique and the metaphor of lens. Users use one hand (coarse hand) to locate an area to focus and magnify, and at the same time another hand (dominant hand) could interact with objects easily inside the magnified view. This technique works particularly well in a city maps, the context is the whole city while the focused area in the lens shows specific roads or even bus stops.

2.2.3 Unique Techniques: Data Specific

There exist unique and novel styles of touch action that cannot be easily grouped into categories. Or we could say that each one of these can be a category itself. Most of them are self-developed and data-specific, and very likely to be extraordinarily effective to explore a certain type of data.

The most famous example should be 'pinch to zoom out and stretch to zoom in' for map visualizations. As you stretch over a city, you get to see the districts and maybe streets, and you can further zoom in to reveal buildings.

DimpVis [23] employs one of those novel techniques to support object-centric temporal navigation. Users can manipulate any data item along its hint path to navigate temporally, so as to focus and explore the local spatial changes of an individual data item (see figure 2-4). For example, to find out 'whether this bar is ever at 500' in a time-varying bar chart, a user just needs to drag the bar to that height. If there exists a moment that the bar is at the height, the visualization including unselected items would be updated to reflect the data at that time.

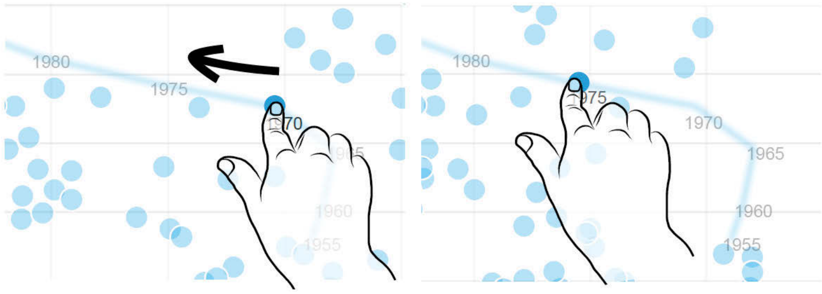


Figure 2-4 Navigating a data item temporally along its hint path [23]

There exists a innovate or even bold technique from Kelleher and Grinstein [24], named Fractal Perspective, see figure 2-5. Nodes are encoded as circles while edges are encoded as squares. And distant objects appear as smaller circles. The layout is quite similar to Circular Treemap [25]. Just zooming and panning this nested object visualization, which represents a node's viewpoint, are enough to travel through the node-link graph. Kellerher and Grinstein claim that their proposed technique fits perfectly with URL identified nodes, where incremental data requests need to be dynamically visualized.

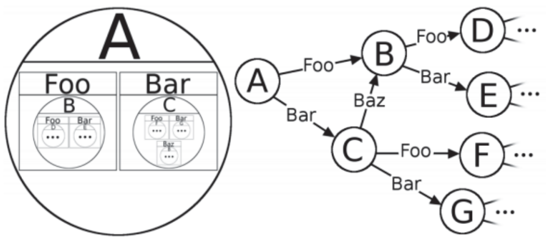


Figure 2-5 left: fractal perspective right: corresponding node-link representation

2.3 Data Independent Touch Action Styles

2.3.1 Touch Panel

WIMP based interface could be effectively ported to interactive surface, simply by replacing a mouse click with a finger tap [12], [26]. In fact, most of the hand hold devices such as smart phones and tablets still use this desktop metaphor as main interactive style, where operations are achieved through control panel, push buttons, checkboxes or dialog windows. But the exciting fact is, even with WIMP metaphor, touch enabled panel could still outperformed mouse with its increased input bandwidth and directness, such as operating multiple sliders simultaneously either through multiple fingers selection or vertical hand swipe [27], see figure 2-6.

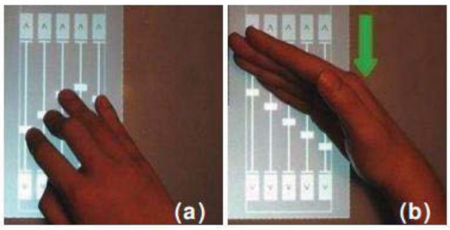


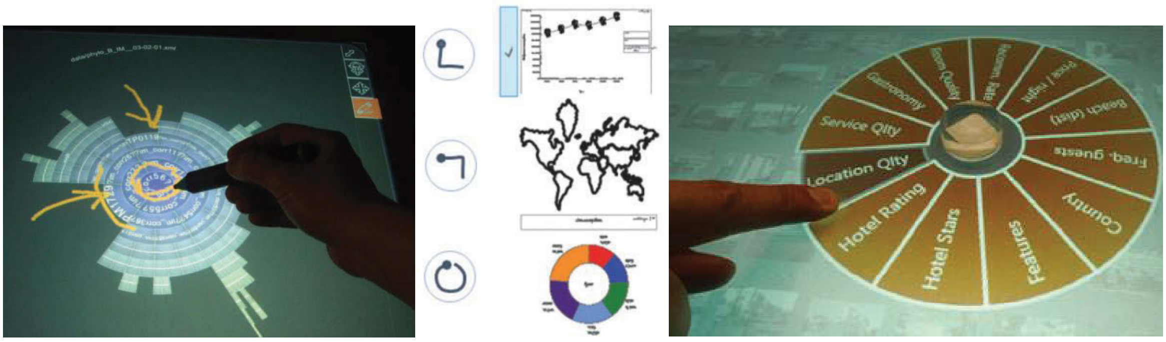
Figure 2-6 operating multiple sliders simultaneously

The control widgets are sometimes indispensible in visualization systems, particularly under the situation where exist multiple options of reconfiguring, encoding, filtering, or connecting [14]. And Wobbrock et al. [9] also suggest the needs of widgets for the operations whose gestural agreement are very low. As an example, menus are still needed when it comes to change x axis, even in FLUID version (which focuses on touch interaction with limited panels) of TouchViz [12].

2.4 Mixed Touch Action Styles

2.4.1 Multi-Modal Interaction

Hybrid interactive surface combines multi-touch and an additional device, such as pen [10], [28]–[30], and token [31]. As Hinckley [32] puts it, Sometimes a single device may not be enough for all the requirements of tasks. Furthermore, some certain devices would be more intuitive and nature to use than others. For example, people are more used to write using pens rather than fingers.



(a) (b) (c)

Figure 2-7 use pen to annotate [10] (a), create charts [29] (b) and employ a token to do query [31] (c)

Most of the applications requiring additional devices are implemented on large screens, such as tabletops, interactive whiteboard or wall-sized screens. Isenberg and Carpendale [10] introduced an interactive tree comparison system which support both finger and pen input. While finger is frequently used to drag items or change views, pen is much more convenient to make annotation. SketchStory [29] exploits a deeper degree of sketch ability for presenting and communicating. An example icon created by the presenter could be used in the chart, and the desired type of chart could be automatically completed based on the axis drawn by the user. Instead of pens, Facet-Streams [31] use a glass token to facilitate query process. The token's physical appearance already reveals its possibilities of translation and rotation. Virtual elements would be displayed around each token, forming a 'wheel' as user puts it on the screen, and then the user could do further filtering by touching different range of the wheels.

2.4.2 Gestural Trigger

Gestural trigger is a relatively complicated gesture (we certainly do not want to evoke a certain action by a single tap and block everything else, since every gesture begins as a tap) replacing the button to be the trigger of an effect. The gesture should be distinctive and unambiguous, and at the same time reliably be detected by the system and easily learned by users [33]. This becomes a gesture recognition related technical problem and a user-centric design related problem. The action performed by users should map to the effect it will cause, in other words, matches user's expectation. So training time will be minimized, and retaining phase can be longer. Mature employments of this technique lie in navigation, such as "swipe to change between views".

However, designers must be aware that not every operation performs better with gestures than with traditional control panel. Assigning inappropriate gesture to a task, even as simple as double tap or long press, would add up user's cognitive overhead. For instance, in TouchWave [18], long press will reorder the layers according to their values, and double tap would change the stacked graph's layout. Users might wonder if the result is in ascending order or descending order, or get confused about whether long press results in resort or double tap does so after a while. These consequences can no longer be discussed since no usability study has been conducted during the study. Instead, TouchViz [12] provides a more intuitive and well-matched way for sort task: drag vertically along the axis from small value to larger value would sort the data by ascending order, while the moving in the opposite direction will sort it in descending order. But please note that this action is categorized as a drawing gesture, we mention it here only as a contrast for reorder in TouchWave. Developing new gestures other than widely used ones (e.g., defined by iOS, they are tap, pan, long press, pinch, rotate and swipe [34]) might be a wise idea. Especially when we run out of options, or no custom ones could perfectly match the effect of the operation.

2.4.3 Drawing Destures

Gestures involve drawing simple shapes, paths or symbols are classified as drawing gestures. With the constraint of the difficulty for users to perform or memorize, and for system to detect, no complex moving should be employed. On the other hand, the finger would be too 'fat' to do delicate drawings. And stylus would be a more intuitive tool to draw or sketch, as we discussed before.

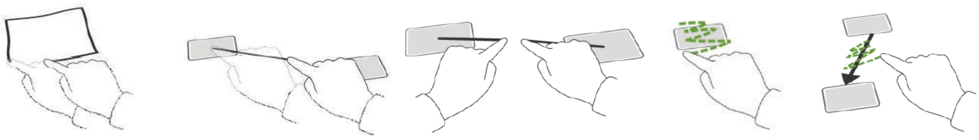


Figure 2-8 drawing gestures used to manipulate diagram

One common use of drawing gesture is to encircle an object or group of objects to select them. Also, to edit node-link graph, user could draw a square to create a node, or draw a line between two nodes to create an edge [28]. Similarly, connections between tokens in Facet-Streams [31] could be created by drawing path between them, or be deleted by crossing them out. Kinetica [19] allows user to draw a region that filters points. Once they finish drawing, the points are immediately pushed outwards.

It is easy to tell that the above examples are all data dependent. There also exist some actions performed as system commands, which are data independent. The user-defined gesture set proposed by Wobbrock et al. [9] includes drawing a check to accept, drawing a 'X' to reject and drawing '?' for help. And Magllenes et al. [35] combined such drawing command with graphical elements in their proposed system.

Chapter 3 Research Methodology

3.1 Design Research and Design Studies as Projects

Michael Sedlmair and his colleagues [2] define a design study as a project in which visualization researchers work with domain experts on a real-world problem and deliver a visualization system. Design study is different from the technique-driven research, where the goal is to develop new and better technique without necessarily establishing a strong connection to a particular user need. Design study is a particular form of problem-driven research, where the goal is to work with real users to solve their real-world problems. The definition implies that a design study paper does not require a novel algorithm or technique contribution.

While in the context of interaction design, design research is an integral part of the user-centered design process, which is an interactive cycle of plan, observe, design, prototype and test. These steps align with the core process of Sedlmairet et al. [2] 's framework. In fact, design research has been practiced since 1960s in the architectural, industrial, and academic communities. We will be discussing the first four components in the following paragraph, and then test in 3.3 Evaluation.

3.1.1 Precondition: Learn, Winnow and Cast

Before conducting the study, researcher should gain a solid knowledge of the visualization literature, including design guidelines, visual encodings, interaction techniques and evaluation methods. In the next stage, winnow, the goal is to identify the most promising collaborations. Sedlmairet et al. [2] suggest holding initial meetings and considering questions about data accessibility, engagement and task validation. And then identify collaborator roles during cast stage. There will be a front-line analyst, gatekeeper, which are crucial, and additional roles such as co-authors or fellow tool builders.

3.1.2 Core: Discover, Design, Implement and Deploy

For observation, we need a plan first. Designers in HCI community will need an interview guide before the user research. This guide should include the goals, problems, and contexts of the people who might use their design, and also how the settings and goals inspire the designers to create new applications, products, and services. And then perform the interview under certain environment, and record the process for further investigation.

Another option is to get the question from literature reviews. The problems might show from user study, case study, or the author's explicit statement. Result of Chapter 4 originated from this. Activity analysis will be conducted during the process. We need to find out steps, artifacts, goals and pain points from it.

Then it is time to collect the observation result, and connect it to the actual design. During this transformation, prototype plays a role for need finding and feedback. Prototyping is also a strategy for efficiently dealing with things that are hard to predict. Prototype is supposed to be incomplete, easy to change and get to retire. There are three types of prototype, feel, implementation or role. That is, the prototype can either look like, or work like, or provide the similar experience as the final delivery. There can also be several prototypes, the best way to have a good idea is to have lots of ideas. And the design process might start as multiple ideas, then the flaw ones get to retired as the new ones keep coming up. Finally there might be two or three ideas left to be prototyped and tested.

Industry practitioners often use tools to do interactive prototyping, such as Dreamweaver, Axure. Different than the goal to deliver a product, researchers are exploring a potential or 'perfect' solution for a certain problem. And for the visualization field, interactions are embedded within the visualization, which existing prototyping tools cannot nicely support. Actually MS PowerPoint and Word are powerful and more frequently used by researchers. We can easily and quickly create an earlier prototype that looks like our final delivery, and use panels to mimic how it works like. And instead of user testing, researchers tend to employ peer review for prototype tests.

The implementation of software systems and tools actually tightly intersect with the design process. But under the circumstances when we do need a final delivery, here are some implementation options:

i. Use existing data visualization frameworks, such as OpenGL-based Caleydo, or Javascript-based D3.
ii. Use data analysis tools and language which support interactive visualization, such as R.
iii. Other Graphic Framework, such as Core Graphic from Cocoa (for iOS).

3.1.3 Analysis: Reflect and Write

Reflect is about improving and proposing guidelines. Based on new findings, currently available guidelines can be either confirmed by further evidence of their usefulness; refined or extended with new insights; or even rejected when they do not work when applied.

Paper writing can be started at any point, which would make it a lot easier according to my experience. Writing is the time to revisit abstractions, to identify contributions, and to gain a coherent and understandable line of argumentation.

3.2 Data Analysis Methods

3.2.1 Problem Definition

To define a research problem in data analysis, we could start with a general question, for instance, can I automatically detect emails that are SPAM that are not? Then make it concrete and specific: can I use quantitative characteristics of the emails to classify them as SPAM/HAM? Then we need to define the ideal data set. The data set may depend on the analysis type and research goal. But his approach might be not exact proper in design research, where questions might come after the solution.

There are six different types of data analysis:

- Descriptive - a whole population
- Exploratory - a random sample with many variables measured
- Inferential - the right population, randomly sampled
- Predictive - a training and test data set from the same population
- Causal - data from a randomized study
- Mechanistic - data about all components of the system

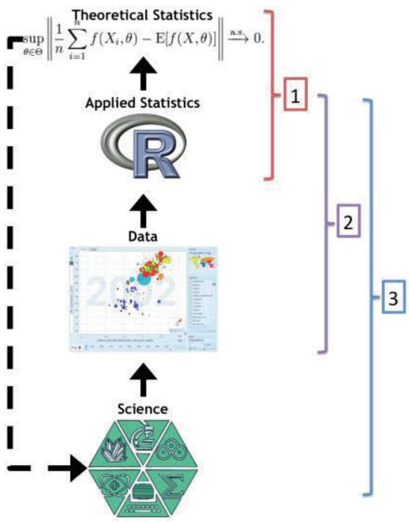


Figure 3-1 Pipeline of data analysis [36]: 1. Statistical methods development 2. Danger zone!!! 3. Proper data analysis

3.2.2 Getting and Cleaning Data

First we need to determine what data we can access. Sometimes we can find data free on the web, other times we may need to buy the data. Or we may need to generate it ourselves if the data don't exist.

The following table shows free public data sources.

Table 3-1 list of public open data sources

Data description	Links
Open government sites of United Nations	http://data.un.org/
Open government sites of U.S.	http://www.data.gov/
Open government sites of United Kingdom	http://data.gov.uk/
Open government sites of France	http://www.data.gouv.fr/
Open government sites of Ghana	http://data.gov.gh/
Open government sites of Australia	http://data.gov.au/

Open government sites of Germany	https://www.govdata.de/
Open government sites of Hong Kong	http://www.gov.hk/en/theme/psi/datasets/
Open government sites of Japan	http://www.data.go.jp/
Open government sites of others	http://www.data.gov/opendatasites/
Gapmider	http://www.gapminder.org/
Survey data of United States	http://www.asdfree.com/
Data science competition	http://www.kaggle.com/
Collections by Peter Skomoroch	https://delicious.com/pskomoroch/dataset
Collections by Jeff-Hammerbacher	http://www.quora.com/Jeff-Hammerbacher/Introduction-to-Data-Science-Data-Sets
Collections	https://github.com/caesar0301/awesome-public-datasets
Standard Large Network Data	https://snap.stanford.edu/data/
UCI Machine Learning	http://archive.ics.uci.edu/ml/
R datasets package (clean)	https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html

The data we obtained from the web is usually raw data, which means we need to clean the data.
For instance, raw data could be

- the strange binary file your measurement machine spits out;
- the unformatted excel file with 10 worksheets the company you contracted with sent you;
- the complicated JSON data you got from scraping the Twitter API;
- the hand-entered numbers you collected looking through a microscope.

Raw data often needs to be processed. Processed data is the data that is ready for analysis, such as a neat json file used in D3 Website. Processing can include merging, subsetting, transforming, etc. If it is pre-processed, we need to make sure that we understand how. Also, we need to understand the source of the data (census, sample, convenience sample, etc.) Don't forget to determine if the data are good enough - if not, quit or change data.

The four requirements of tidy data [36]

1. Each variable you measure should be in one column;
2. Each different observation of that variable should be in a different row;
3. There should be one table for each "kind" of variable;
4. If multiple tables are needed, they should include a column in the table that allows them to be linked.

According to the structure of data analysis, we need to do exploratory data analysis as a fore step of statistical prediction/modelling. Since visual analysis here actually focus on visualization, we should see exploratory data analysis as a source of inspiration, and help us getting a general sense of the data. The basic steps include:

1. Look at summaries of the data
2. Check for missing data
3. Create exploratory plots
4. Perform exploratory analyses (e.g. clustering)

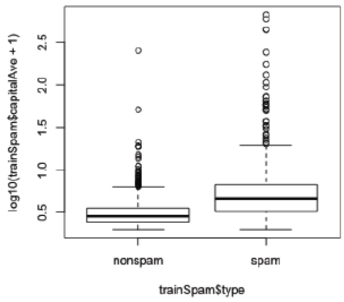


Figure 3-2 result of "plot(log10(trainSpam$capitalAve + 1) ~ trainSpam$type)" in R

3.3 Evaluation

3.3.1 Evaluation Designs

There are three types of evaluation design. In an experimental research design, one or more of the independent variables are manipulated to ascertain their impact on the dependent variables. A quasi-experimental design distinguishes itself from a real experiment by its lacking random assignments of subjects to different treatments – in other words, subjects decide on their own about their treatment, e.g., online user tests. Non-experimental designs include all other forms of quantitative research, as well as qualitative research. E.g., longitudinal research, questionnaires, case studies.

Table 3-2 Basic characteristics of evaluation designs. Modified from [37]

Data	Historical datasets, real user studies
Subject	Online customers, students, historical user sessions, simulated users, computers
Research method	Experimental, quasi-experimental, or non-experimental
Setting	Real-world scenarios, lab

Table 3-3 evaluating methods for evaluation, partially derived from [38]

method	advantages	disadvantages	RP	G	R	C
Usability studies	Learn a lot; Avoids developer blinders	Environment / motivations not representative; lack of comparison; experiments bias	?	?	?	-
Surveys	Quick; cheap; large sample size	Reporting vs. actual behavior	+	+	-	+
Expert feedback (peer critique, heuristics)	Quick valuable insights; deep insights via dogfooding	Developer blinders; inbreeding; not generalized	?	-	-	-
Comparative experiments (A/B testing)	Identify key variables; see actual behavior; actionable	Expensive	+		+	+
Participant observation (in real use environment)	See real, sometimes longitudinal behaviors	Expensive, requires mature prototype	?	+	+	-
Case studies (applying real data; minimum user participation)	Quickly shows the applicable contribution	little data from user	+	-	+	+
Theoretical comparison	Quickly shows the contribution in theory	Not practical	-	-	-	+

*Note: 'RP' = 'Reliability/Precision (repeatability)'; 'G' = 'Generalizability (representativeness of the sample)'; 'R' = 'realism (applicable vs. environment/population)'; 'C' = 'Comparison (alternatives considered)'; '+' represents positive; '-' represents negative; '?' represents depending on condition; ' ' represents unsure

Different methods answer different questions, and are best used at different times in the design cycle.

Keller et al. [39] categorized testing tasks for visualization systems into nine category: 1) identify, 2) locate, 3) distinguish, 4) categorize, 5) cluster, 6) rank, 7) compare, 8) associate, and 9) correlate. But as the data itself and analysis techniques keep developing, tasks in visualization systems and tools will also be growing in quantity and diversity.

3.3.2 Measurements

3.3.2.1 Quantitative Measurements

Accuracy. Accuracy is an easy way to evaluate the effectiveness of a design. For example, the question could be how many clusters can user observe from the visualization, or which item is most important.

Time spent in tasks. It is also quite common to compare time spent in same tasks to demonstrate that the proposed design outperforms some former ones.

User feedback and satisfactory. After the user study, subjects could fill in a rating survey/sheets based on their experience of the RS.

3.3.2.2 Qualitative Measurements

User behaviour. We can observe user's behaviour during usability studies so that we can tell where users encounter a problem we never expected, or how easily they learn to use the system.

Case studies. Apply several real-world dataset to the proposed visualization design; trial experience and interviews with domain experts or actual users.

Open ended questions. Tasks and questions with exact in experimental studies are normally too simple to truly exam the design, or hard to apply to reality. But answering open ended question would require more background knowledge from users and more time during the studies.

Qualitative comparison with the traditional approach, such as the description of overcoming a problem as Hernando and his team did [40], or list of advantages and contributions.

Chapter 4 Merits and Limitations of Touch interaction

4.1 Unique Values Interactive Surfaces Offer

Isenberg et al. [5] proposed the research questions regarding the intersection of visualization and interactive tabletops and surfaces (ITS). We will be discussing one question here: how can ITS contribute to efficient, effective, and satisfactory data analysis with visualizations? We listed unique features that ITSs can offer but WIMPs do not - in other words, where ITSs outperform desktop and mouse -which include:

1) Physical metaphor
2) Increased input bandwidth
3) Additional degree of freedom to express intentions
4) Panel free
5) Effective zooming/scaling
6) Trajectory enabled
7) Kinetic feedback
8) Collaborative work
9) High resolution
10) Anywhere, anytime, anyone
11) Sensors

Physical metaphor

Physical metaphor make it possible to treat on-screen objects freely as they are in physical world. As demonstrated in direct-on-data manipulation section, directly manipulating the targets could be one of the biggest merits interactive surface could offer. Reality-based interaction [41] is built on users' pre-existing knowledge of the everyday and non-digital world, to a much greater extent than before. People meditate a sense of control and the model-world system provides the desired feedback [42].

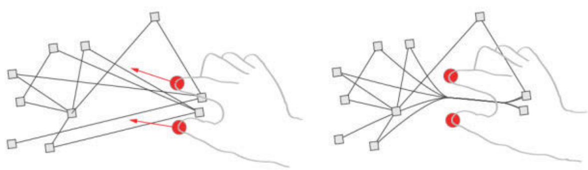


Figure 4-1 funnel gesture in graph visualization [17]

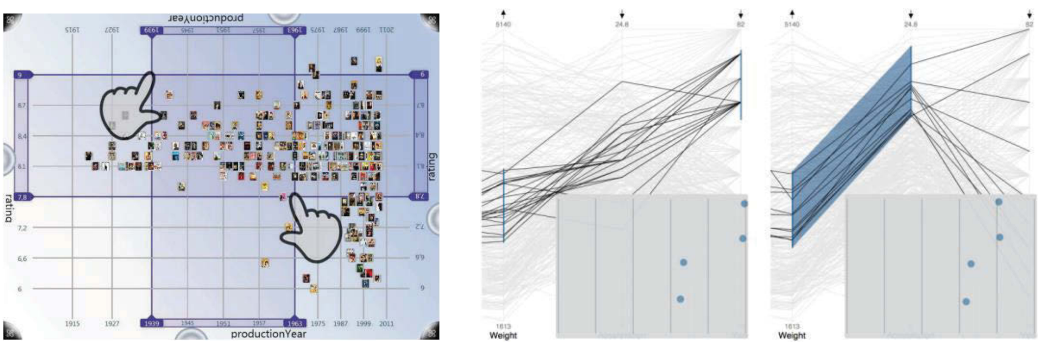
On one hand, by removing the mediator (e.g., mouse cursor) between user's hand and the object, extra mental effort is reduced. Less usage of cognitive resources enables concentration on tasks rather than the interface. And the time spent on visually track an on-screen cursor which is away from the hand [43], or constantly clicking for choices are saved. Previous studies have shown that direct-touch outperforms mouse for both single and multi-target selections [43], [44]. On the other hand, treating objects as they are in reality could benefit the data exploring tasks. When a manager of a restaurant tries to analyze the budgets of a year, he/she could place all the budgets per month on the table, and might move some of the sheets away or place several of them side by side for comparison, or even overlay two of them, if the paper is not completely opaque. Touch surface enables this kind of activities in the digital world in an intuitive and flexible way, virtual items could be manipulated as freely as they could be in the physical world.

This reality metaphor has been widely adopted in numerous information visualization systems [17], [19], [45]–[47]. An interesting example is focused on edge interaction. To avoid cluttering or to highlight relationships in a node-link graph, one could also bundles the edges by performing funnel gesture along with the direction of edges [17].

Increased input bandwidth

Rather than a single point controlled indirectly by mouse, interactive surfaces provide simultaneous control point of direct touches, which increases parallelism and makes users feel more 'powerful' with data.

For instance, in the machine-learning step, a user could move a data point with one hand, while another hand specifying target data points to indicate which similarity relationships he or she intends. The added bandwidth of these multi-touch interactions can more effectively and accurately define such manipulations' analytical reasoning. Another example is, by performing pinch or stretch gesture inside the scatter grid, both dimensions can be distorted simultaneously [48], see figure 4-2 (a). Although gestures are performed indirectly on a trackpad, Poster [49] leverages the multi-point feature to implement brushing non-adjacent axis and adjacent axis of a parallel coordinates. These are enabled through four touch points, two points on each coordinate respectively deciding the range of selection, see figure 4-2 (b-c).



(a) (b) (c)

Figure 4-2 multi-point input in scatter grid (a) and parallel coordinates (b, c)

Additional degrees of freedom to express user's intentions

Multiple types of actions can largely facilitate target-specific tasks like to delete one target item. There are traditionally two ways to achieve target-specific tasks with WIMP: 1) define operation before target (DOT), and 2) define target before operation (DTO). DOT uses the technique of 'mode'. For example, if users tend to remove an object, they must first enter 'remove mode', then select the object, and finally exit 'remove mode'. DTO utilizes the prompt panel, to remove the object, users need to perform an action (e.g., mouse right click) on the certain object to elicit a prompt panel, then select 'remove' out of a list of options from prompt window. But both are problematic somehow, for example, switching between modes can easily confuse users, and the DTO can only deal with single object at a time. Touch interaction achieves 'modeless', you can slide on the target item directly to remove it. Yet DTO is still constantly and necessarily adopted in visualization systems [12], [13], [18], [19], [50], with improved bandwidth and freedom, and minimized UI components.

The richer expressions enabled by touch screens provide additional degree of freedom for users to express their intentions, which is significantly useful for exploring visualization containing spatial information. Isenberg et al. [51] proposed a set of postures to facilitate exploring and annotating the 2D vector fields. They employed one-finger posture to do local exploration, while fist posture to do a larger scaled one. And de-amphasis and erasing could be achieved through flat hand wiping. Moreover, North et al. [52] suggest that manipulating node location with multi-touch technique could encourage people to "think with their hands".

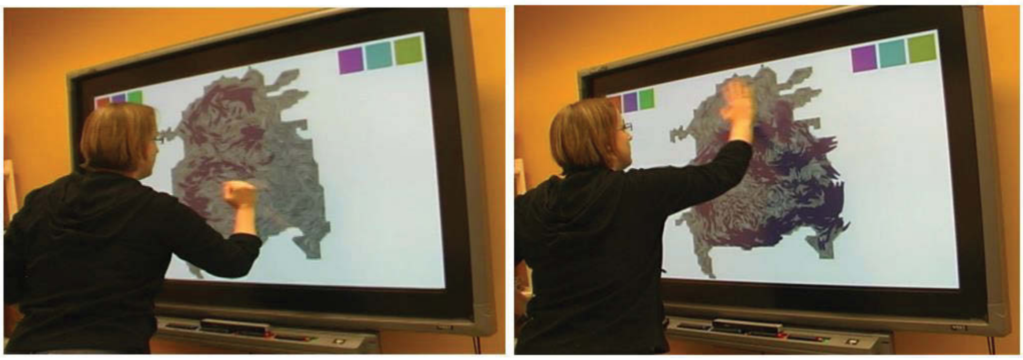


Figure 4-3 Exploring larger areas of the dataset using a fist (left) Glyph de-emphasis & erasing using the flat hand (right)

From user's perspective, people tend to feel amazed at the interaction free from mouse and keyboard with richer expression, thus they are more likely to feel engaged and willing to do data exploration and chances of unexpected findings is increased. As Elmqvist [53] says, manipulating visualization using touch screens is in many aspects similar to video games, thus lead to the transformation of time and loss of self-consciousness. On the other hand, as a powerful modality to make demonstration and presentations, interactive surfaces (e.g., whiteboards) are able to attract more attention of audiences [29], [30].

Panel free

Lee et al. [54] argue that exploring different input modalities could limit the number of menus and controls, allowing analysts to concentrate on data exploration tasks rather than operating an interface. WIMP could easily get users 'drowning in the functionality' [55] because of unlimited buttons and menus. People might get distracted by the enormous number of widgets, mouse clicking, and travelling through levels of menus. With touch enabled interface, we can say goodbye to overwhelming menus.

Another advantage of panel-free is space utilization. In TouchViz [12], space is more effectively used in FLUID (a gesture-centric touch based interface) version than WIMP version, see figure 4-4. Because we no longer need to list every option in the side menu, there is more space available for the visualization.

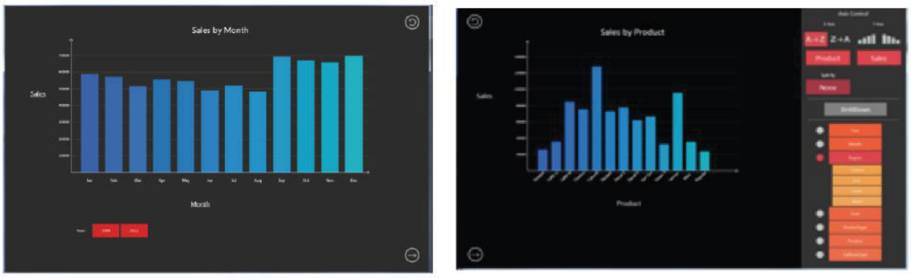


Figure 4-4 advantage of space utilization, screen shots from [12]

Effective zooming and scaling

Interactive tabletops and surfaces are particularly effective and efficient in exploring maps [50], [56]. 'Pinch to zoom out and stretch to zoom in' has almost evolved into a behavior. Apart from free control towards degree of scale and easily understandable pinch/stretch action, also there is no need to adjust the area of interest as we have to do when using buttons.

Figure 4-5 effectively interact with map visualization

Trajectory tracking

Mouse is precise and can move fast, but it is hard to make it move stably along a certain path. Everyone would be surprised to find themselves such terrible painter, when experiencing computer painting tools for the first time. And to keep pressing the button during dragging would be struggling and annoying. At this point finger interaction again shows higher performance by providing nicer and friendlier tracking experience.

With trajectory detection, the dimensions of data being explored are increased. Spatial information can be interrogated more deeply. Advanced operations such as visual query are enabled. There exist several tools that employ freeform sketch query technique to explore spatial or temporal data [53], [57]–[59].

Furthermore, interacting with data by moving finger flexibly on the screen can build a 'playful exploration' [47]. The Bohemain Bookshelf [47] creates the impression of book covers bubbling up to the surface and disappearing again, by enlarging and shrinking the thumbnails of book covers along the trajectory of touch point, which could potentially motivate serendipitous discoveries.

Figure 4-6 "playful exploration" in Bohemain Bookshelf

Kinetic feedback

Interacting with data on a surface provides somatic feedback [60], which can enhance contextual awareness during navigation, as well as help users to memorize or recall the exploring process of a certain visualization result. The reason is simple enough. Sequence of hierarchical menu operations is very hard to remember. But finger actions can be automatically retained after practice, which is much easier, somehow similar to muscle memory gained through playing piano [61].

The time line hint paths of DimpVis [23] serves as an guidance, and helps analyzing trend at the same time. As mentioned before in self-developed technique in literature review section, the hint path of one target point shows its temporal changes and the visualization will updates as finger move along it, see figure 2-3. So as users move along the path, they are feeling the changes both visually and somatically.

Collaborative work

The information space can simply be too complicated for an individual to interpret entirely [10], especially as data scale and complexity keeps exploding. On the other hand, the dataset may be susceptible to a variety of interpretations. In this case, if analysts may discuss and negotiate their interpretations of data, and work in group, the chance of serendipity will be increased.

Shared analysis of information visualization can increase information processing power, and also enable team members to share, negotiate, and discuss their interpretations on a dataset, thus contribute unique perspectives on a given problem. Isenberg and Carpendale [10] proposed a system supporting collaborative tree comparison. Visualization planes can be freely rotated, translated, and scaled. And it enables both individuals exploration and group inspection, as well as fluid switch between individual view and collaborative view.

There are a lot more research questions concerning this, but since collaboration is hard to achieve through tablets, we are not intending to go further.

High resolution

Larger screens (e.g., wall-sized displays) can display larger amount of information at a glance, which could potentially mitigate the problem of increasing data complexity [5]. Shared work or presentation tasks are appropriate to implement on large displays, such as tabletops and whiteboards. But again we are not going further since we will be focus on medium sized touch screen.

Anywhere, anytime, anyone

Today, People want to stay up-to-date with information, and make appropriate decisions without the limit of time or places. Thus designing tools and techniques for efficient data access to be immediately and intuitively accessible are becoming increasingly important [5]. Portable devices such as smartphones or tablets bring new possibilities.

Novel interaction of visualization also broadens the range of users beyond analysis experts. Keefe and Isenberg [62] see NUI as an opportunity to make extremely complex scientific visualization systems more broadly accessible. That is, scientists and engineers can leverage the most powerful data-driven computer graphics methodologies without needing to learn to program or script new workflow. With touch screens coming into daily life, people can analyze personal or business data with their smart phones, playing with data on tablets [12], [18], [19], or obtaining deeper knowledge in public places such as libraries and museums [47].

Sensors

Smartphones, and sometimes tablets, could serve as sensing device or adapt themselves depending on the physical context, using their contained sensors such as a gyroscope, light sensor, pressure sensor, accelerometer and GPS [63]. This is once discussed by Drucker et al. [12] as a type of interaction, actions that involved physical movement or sensing on the device (e.g., tilting the tablet to sort the data).

4.2 Limitations and Tradeoffs Compared to WIMP

As Bill Buxton puts it, everything is best for something and worst for something else. Those who try to replace the mouse are playing a fool's game. The challenge with new input should be finding devices that work together with the mouse, or things that are strong where the mouse is weak, thereby complimenting it [6]. Therefore we discussed where desktop outperform touch screens as well, and briefly discuss several studies aims to alleviate the problems and restrictions.

Worse precision - 'fat finger problem'

Mouse cursor could select as small as a pixel, while fingers perform quite badly in precise selection. This problem has been named as 'fat finger' [55], [64]. There are basically two ways to mitigate this problem: 1) make the target larger, or 2) utilize precise tools such as pen. Sears and Shneiderman conclude that targets of size larger than 0.64 cm in width were faster to select by touch than a mouse [65]. And companies such as Samsung provide a digital pen to face the problem. But in research community, pens or tokens are employed because of their more nature function, rather than a compromised solution (see literature review section for more detail).

Education

Another challenge is education. Designers or researchers need to teach users to perform and recognize new gestures, which will take longer in training than self-explaining WIMP interface. But thanks to the kinetic memory, touch display interface can actually achieve higher retaining rate in some cases. On the other hand, expert users could perform actions quicker than using mouse after practising, just as shortcut keyboard in Windows dose(e.g., "ctrl+A" to select all).

Occlusions

As a cost of direct manipulation, hand would obscure the visualization sometimes while performing operations (see figure 4-7). Designing interactions beyond surface could help mitigate this problem, for example, in-air and on-bezel gestures (Valdes et al. 2014). Yet when users don't need to see the entire visualization during manipulating, this problem seems trivial.

Figure 4-7 occlusion caused by hand

Not self-explaining

Unlike WIMP, it is not possible to provide whole list of options all the time. But there are tools that support complicated functions, such as Photoshop, require long list of options. And works better with all the options always available. Therefore, WIMP interface should be employed instead of touch interface in those cases when the tasks are various and numerous.

Low typing speed and high typing errors

With physical keyboard, people can type quicker and with fewer mistakes. Thus if the visualization require quantity of typing, such as detailed annotation, traditional WIMP interface might be a better choice. There are researches addressing this challenge, such as solutions for keyboard of phones [66].

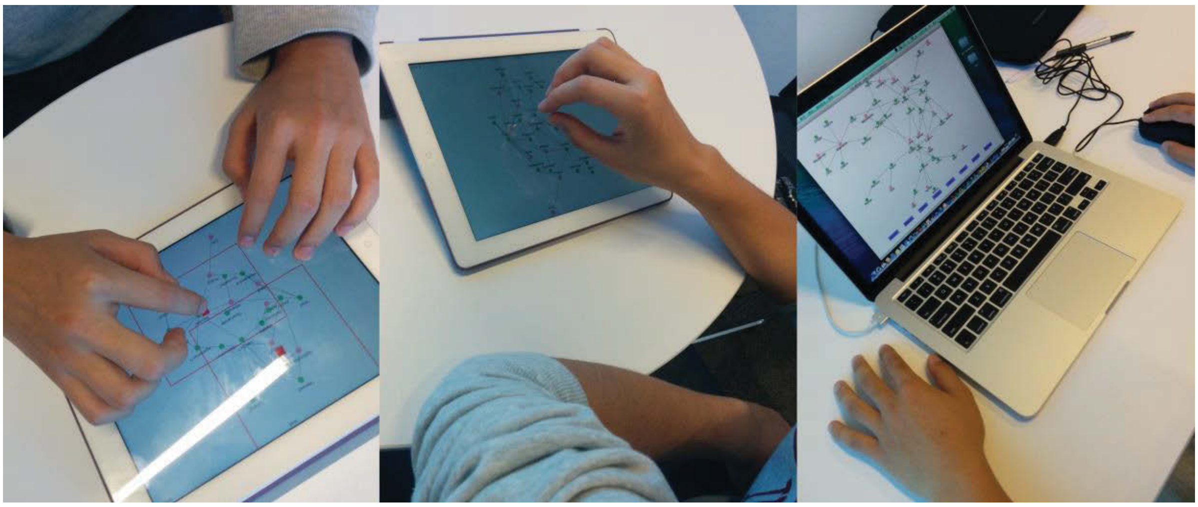
Hard to implement light weighted highlight

This is a potential problem that was seldom brought up by researchers. Using mouse cursor hovering to highlight an item, or to show more detailed information is very effective and common in WIMP based visualization systems. This design helps to avoid clutter by selectively showing information at a time, and reduces cognitive effort through highlighting the interested item. But this style cannot be easily or effectively ported to touch displays. Just image, selecting items for highlight on touch surfaces (e.g., a tap) takes much more trouble than mouse hover, and meanwhile the object of interest is obscured by the finger. This could be one of the reasons why the data used in touch based visualization system is usually smaller in scale and less complicated than those in WIMP based systems.

Chapter 5 DA-TU on IPad: Clustered Data Visualization with Gestural Interface

DA-TU is an interactive visualization system first proposed in 1998 [67] for visualizing, navigating and retrieving large relational data through clustered graph visual abstraction. Navigation process is conducted through a sequence of interactions across four layers of CGA interaction framework, which are picture layer, abridgement layer, clustering layer and graph layer. It enables users to not only interactively 'create', 'merge' and 'dismiss' clusters, but also change the 'abstract views' of graph in the abridgement layer through the 'open', 'close' and 'extra force scan (gather)' operations.

Despite DA-TU has been proved that it is effective in dealing with large scale data visualization, its navigational operations across different layers are still very inefficient. We had to employ a Control Panel, setting up eight corresponding operational modes for these operations. Before performing a particular operation, we need to swap to an appropriate mode by ticking the corresponding check-box on Control Panel. This is extremely inefficient and these extra HCI overheads (mouse clicks on Control Panel) will not only cause inconveniency in human direct interaction procedure, but are also error prone. On the other hand, exploring different input modalities could limit the number of menus and controls, allowing analysts to focus on data exploration tasks rather than operating an interface [54].



(a) (b) (c)

Figure 5-1 Screens collected from user studies

(a): Participant is performing a 'close' operation in gesture-based interface. (b): Participant is performing a 'gather', in cozy atmosphere. (c): Participant is experiencing with a WIMP based desktop version of DA-TU [68]

In the following sections, we will discuss in detail about how we try to use a new multi-touch enabled NUI for interactions in DA-TU. We aim to increase the efficiency of HCI process in DA-TU. To achieve this, we first ported DA-TU system from WIMP based PC to NUI based iPad. We then designed a set of new multi-finger gesture vocabularies to support DA-TU operations on tablet. By adopting such set of rich HCI vocabularies, we achieved one-to-one mapping between gesture vocabulary and type of DA-TU operation.

5.1 Designing a Multi Touch Gestures Set

To develop a set of multi-touch commands that is efficiency enough to complete all DA-TU operations, and provide a smooth and attractive interaction experience, we mainly considered the following elements through the designing process: mappings between action and function, discriminability of each gesture, and user's physical and metal feelings. In the process of trying to understand and analyze the operations of DA-TU, we classified 10 operations (8 of them are the same as origin DA-TU, and 2 supplementary ones) into 3 categories (see Table 5-1), from simple to complicate based on their action scope: 1) no particular target (None-T), 2) target is one object(1-T), and 3) target includes multi objects(Multi-T). Notice that the complexity is judged according to the difficulties of the performance in the original DA-TU. A complete set of proposed gesture vocabularies is shown in table 5-2.

Table 5-1 10 operations are categorized based on number of target

category	Operations
None-T	gather, dismiss, scale, show, edit
1-T	open, hide
Multi-T	create, destroy, close

*Note: All of them are inherited from origin DA-TU except *show* and *edit*.

5.1.1 Classification Based on Action Scopes

In traditional DA-TU, None-T operations are 'gather', 'dismiss', and 'scale', which implemented by simply clicking on their corresponding buttons in control panel. The other two operations 'show' and 'edit' in this category are ancillary operations. They are used to switch between visible/invisible and edit-on/edit-off states, as the pre step of relevant operations. 1-T and Multi-T operations are similarly conducted in traditional WIMP DA-TU. Both through a series of mode switch operations. When using multi touch screen as a user interface, there is no need to go through this complicated procedure. 1-T operation can simply be completed by performing a gesture upon the target.

Although operations 'destroy' and 'close' are multiple objects (multi nodes) operations, they can actually be converted into single target (a single cluster) operation through two steps. In fact, two direct single/none object operations (or two steps) are much more efficient than one process of selecting multiple objects. Because it could significantly save the time spent in detecting or adjusting multiple objects selected by the user. The first step is an ancillary step, showing clusters visual attributes which represent the cluster, say the centroids. In second step, we operate on the selected centroid as operating one target.

Unfortunately, multi-object selection seems to be inevitable when conducting 'create' operations. Inspired by prevalence of lasso and pinch & stretch gesture, we first developed a starting step by four-finger stretch to activate and start the selection at the same time. But due to mixed user feedback, the starting step is simplified. In the revised version, 'create' and 'delete' are only available under the 'edit' mode. Instead of four-finger stretch, a user could simply tap and place four vertexes simultaneously to initiate a selection area and start 'create' under the 'edit' mode.

5.1.2 Design Guidelines

Mapping functions to user's expectations

We expect to minimize the time and energy a user spends on training. Thus the closer the gesture matches people's expectation and understanding of the operation, the better the design. However, designers can not represent users, and individual user's feeling could diverse. That is one of the reasons why we conduct usability study one. For example, the 'dismiss' operation will destroy all clusters at the clustering layer. This effect matches the real world action through 'wiping' or 'erasing'. Hence we designed a 'three-finger swiping' gesture for the 'dismiss' operation.

Balance of Association and Complexity

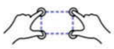
It certainly obeys the aim of abridge the distance between people's intent and action upon the interface if the gesture is too complex to perform and remember. As a result, delicate moving and drawings are not adopted in our design. At the same time each gesture should be distinguished to avoid confusing, even if the referents are similar.

For example, None-T operation 'gathering' can be achieved by a four/five-finger pinch, while 'close' operation could also be a possible effect of this action. However, in our design, as demonstrated in Table 5-2, there is little possibility to confuse between them.

Mix of gestures and widgets

As noted before, if a system replaces all its traditional commands with a new complete set of gestures, it would be very hard for users to learn. Furthermore, from the practice of a user-defined gesture set proposed by Wobbrock et al. [9], we can see the significant influence from the traditional WIMP based operations. (e.g., drag the object to the bottom of the surface to minimize it). Therefore, rather than force people to accept completely familiar NUI vocabularies, we take advantage of combining widgets into the design.

Table 5-2 List of proposed gesture vocabularies in DA-TU

Categorical Operations			Description	Effect
None-Target	Picture layer	*gather*	 Use four or five fingers to pinch.	pulling all sibling nodes within a cluster closer together, and makes the cluster disjoint.
		scale	 Using two or more fingers, spread to zoom in, or pinch to zoom out.	This is used to zoom in or zoom out the view.
		show	 Place three fingers on the screen and hold.	The boundaries and virtual/central node of the clusters are appeared. It acts as the pre step of the relevant operations.
	Clustering layer	*edit*	 Tap twice quickly with three fingers on the screen.	This operation activates the *edit* mode, while 'create' and 'destroy' operations are activated.
		dismiss	 Swipe with three fingers and make the track forming like an inclined 'Z'.	This operation will destroy all clusters at once.
One-Tatget	Abridgement layer	*hide*	 Tap the target node and hold until it disappears.	If node *u* is in the basis of the current abridgment, then it is deleted from the basis.
		open	 Tap the target node representing a cluster.	It replaces the node representing the cluster by its children.
Multi-Target		*close*	 While holding three fingers outside the boundaries of clusters, use another hand to tap the target nodes.	All children of cluster *u* are deleted from the basis, and the node representing *u* is added.
	Clustering layer	*create*	 Step 1: Tap twice quickly with three fingers to enter *edit* mode.  Step2: place four fingers to initiate the selecting area.  Step3: drag on the vertex or edge to adjust selection area. Step4: Tap the confirm button to finish. Step5: Tap twice quickly again to quit *edit* mode if needed.	A new cluster of a set of selected nodes is created.
		destroy	 Step 1: Tap twice quickly with three fingers to enter *edit* mode. Step 2: Tap the target nodes and swipe quickly.  Step 3: Tap twice quickly again to quit *edit* mode if needed.	A selected cluster is deleted, and children of the cluster become children of their grandparent.

*Note that the system does not detect which finger/ fingers a user is using, so user is free to perform differently from the demonstrated gesture, just make sure the number of touch points is correct.

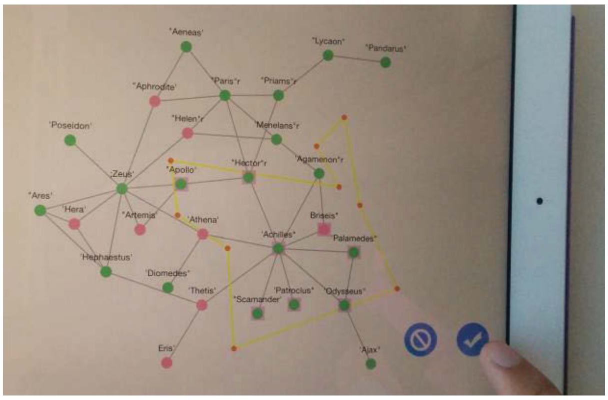


Figure 5-2 Confirm selection and finish create operation

This could be achieved through a two-phase operation. For example, whenever a 'close' is operated on a cluster u, a button like node will replace children and inside links of u, its appearance would be attempting to be tapped. And a tap would 'open' the cluster. Widgets also facilitate the 'create' operation. During the nodes selection, there are points and lines revealing the area, and two round buttons in the corner of the screen providing choice for the cancelation or confirmation of the operation (see Figure 5-2).

5.2 Evaluation

5.2.1 Study One: Evaluate the Reliability and User Satisfaction

One of major goals of this study is to see if the system could reliably detect the gestures performed by users. Each user has their own unique physical features of fingers and gesture habit. It is interface designers' responsibility to adjust the system to fit the majority of users, those with different finger features and moving habits. We observed and collected participants' responses during the study, and ask them to fill in a rating sheet after all tasks being done. People's affection for its free from mouse and keyboard, and rich expression is one of the biggest merits of NUI.

Note that this study is conducted with 1st version of the design. The 2nd version changes the starting step of 'create' operation and improves instructions of performance. The changes are made based on the result of this study.

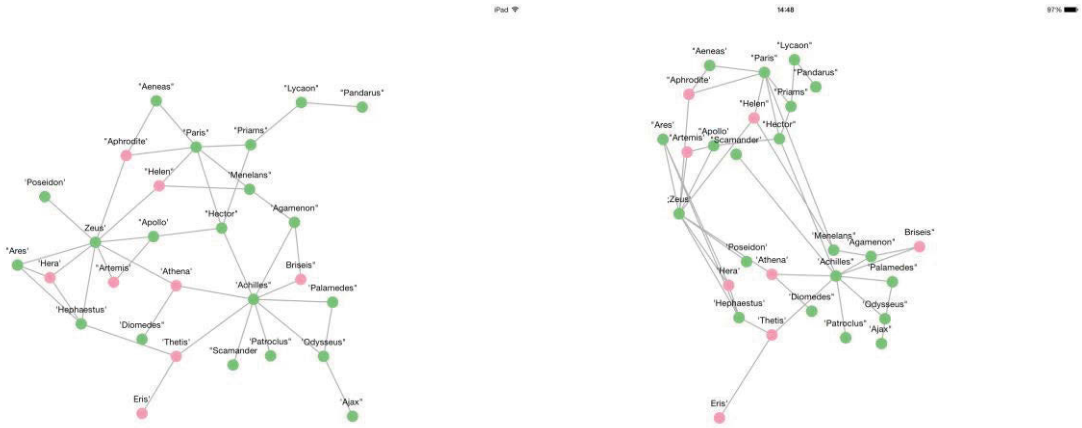
Performing the study

We invited 13 students (aged 21-24, 7 females, all right-handed) to participate in our usability study. Seven of them have backgrounds in information visualization. And seven of them can regularly access to iPad.

We used a force-directed graph drawing to show the relational structure among human roles appearing in Trojan War (see Figure 5-2). The relationships could be based on blood connections (e.g. parent and child, siblings with same parents), or marriages (couples and lovers), and deep friendships or hatred.

Participants were required to first create three clusters: Trojan's angle (e.g., prince Hector), Greek's angle (e.g., the famous Achilles) and neutral (that is Zeus, who constantly changed his preference).

Once the clusters are formed, we then perform a 'gather' operation to make the alliance and hostility clearer. Hints are available during the grouping (or clustering). Participants then could freely perform any of the following operations, such as 'close', 'hide', 'open', 'close', and 'delete' under the instructions. This process ends with a 'dismiss' operation. Note that 'scale' operation is not in the evaluation list due to its dispensability. In the small data set, using 'stretch to zoom in, pinch to zoom out' has almost evolved into a common sense.



(a) (b)

Figure 5-3 Trojan War Dataset

(a): Before gather; (b) The effect after performing gather, the three clusters are showing

The whole procedure consists of three parts: training, executing tasks, and rating. The average time spent for completing this study is about 17 minutes per participant. At the executing stage, participants are required to complete the tasks described in the Task Specification Sheet. The instructor's responsibility is to observe participants' behavior and recognition rate of the system. The last step for users is to fill in a rating sheet, so that we could collect their personal preference and subjective feelings about the design.

Results

The recognition accuracy of the system is quite promising. All of the operations could be successfully detected at the first trial, except 'dismiss' (mean=0.8, sd=1.2 trials failed before success) and 'create' (mean=2.7, sd=3.1 trials failed before success). Note that recognition of 'dismiss' operation is largely improved when adding the description of the track as 'inclined 'Z' shape'. And users could easily achieve 'create' operation at first trial in revised version.

Most participants were able to complete the tasks independently without the intervene of instructor. The easiest gestural operations are show, hide and create, less than two people referred to instruction for completing their tasks.

Ratings about how 'ease' and 'goodness' of the new NUI could be seen in Figures 5-4 and 5-5. Ease is defined as how easy to perform the gesture, while goodness means how good the action matches the task or intent.

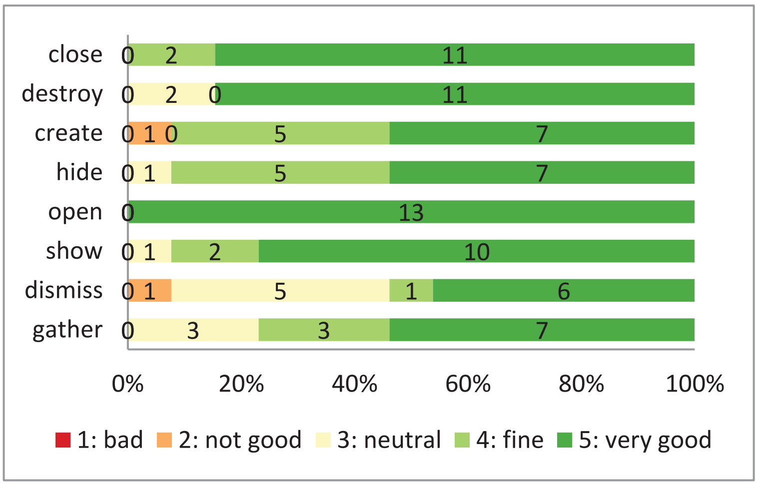


Figure 5-4 Ratings of Goodness

'very good' means the action matches the function very well while 'very bad' means the opposite.

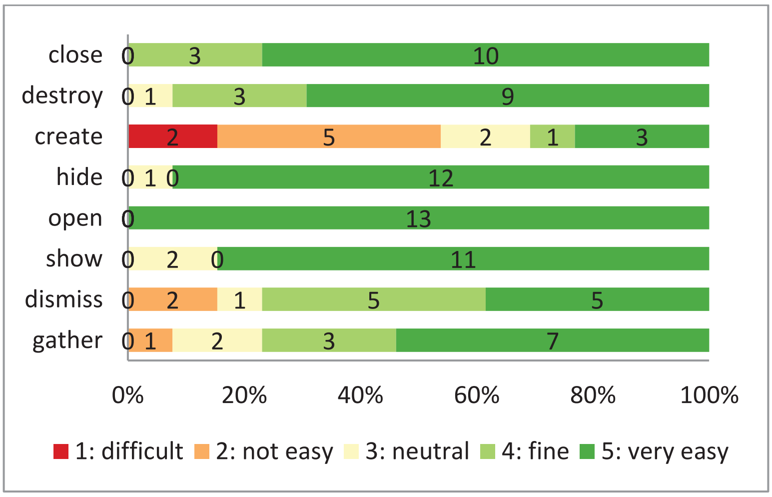


Figure 5-5 Ratings of Ease

'very easy' means the action is easy to perform while 'difficult' means the opposite.

It is quite obvious that 'create' scores poor upon ease (avg. 2.85). However, the notable fact is that 4 participants were satisfied with it (3/13 marked 5, 1/14 marked 4). They even feel so enjoyable and thought it a little dull when invited to try the simplified version. On the other hand, for those who failed several times, they found that it is really delighted when using the revised version of NUI.

5.2.2 Study Two: Evaluate the Efficiency

Performing the study

Ten research students participated in this study, all of them have the knowledge in data clustering and visualization. Four of them are female.

Apart from the Trojan War dataset used in study one, we also used a graph generated from the Game of Thrones. Participants are required to complete six tasks by using WIMP based DA-TU and NUI based DA-TU on corresponding devices. The order of the experiencing devices as well as the datasets is randomly assigned.

To save participant's time, only one cluster needs to be created in task 1. Six tasks based on Trojan War (see Table 5-3) are designed similar to the ones based on Game of Thrones. This ensures the operation complexity is controlled at the same level. For example, task 6 could be changed from "separate them as human and deities" to "separate them as the alive and the dead".

Table 5-3 Tasks designed for character map from Trojan War

Tasks	Operations
1. Make the alliance and hostility clearer.	create, gather
2. Which side has more deities?	create, gather
3. Which side relies more on deities?	close
4. In each side, what can be found between royal family and helpers?	destroy, open, create, close
5. Locate a character most interesting to you.	scale, (create, close)
6. Give up initial clustering, separate them as human and deities.	dismiss, create, gather

Results

Eight out of ten participants preferred the gestural interface, and some of them felt enjoyable with this novel and smooth experience. Particularly if they are familiar with the dataset. One of them even shows interest to be invited again if a new version is developed.

Figure 5-6 shows the box plot of qualitative result. In average, participants spent less time in using new gestural based interface. In task 1, because that the number of visual objects is not large, the advantage of using gestural interface is not significant, that only slightly faster than WIMP interface. However, this difference between two interfaces becomes significant when executing more complicated tasks in Task2, Task4, and Task6. Since the answers could be subjective and diverse, accuracy rate could not be collected. However, we did witness most participants "drowning" in mode swapping when using WIMP interface.

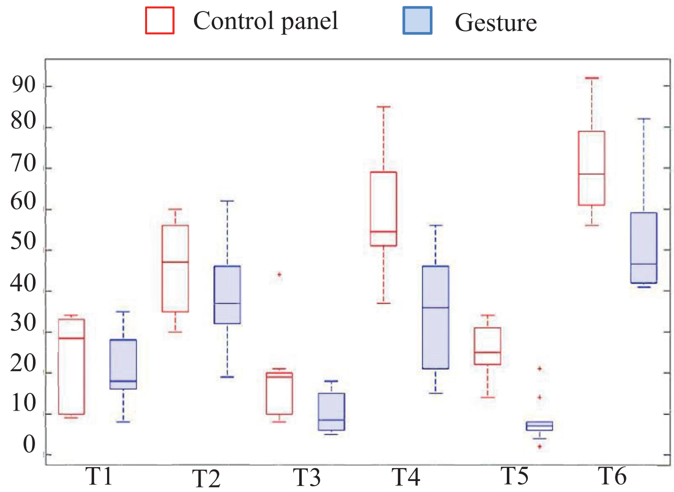


Figure 5-6 Time (seconds) spent by the participants on each task.

Comparing gesture and control panel based interactions in DA-TU. 'T1' to 'T6' represents 'Task 1' to 'Task 6'

One of the participants used the word 'play' to express his experience of using proposed gestural interface, which suggests that people would feel relaxed and more promoted to do data exploration with more nature interface. Another participant mentioned that he felt himself more 'powerful' in manipulating with multi touch gesture system than manipulating one-point based mouse interactions. In general, most of the participants commented 'interesting' during the interaction through touch screen.

Chapter 6 Radial Cursor: an Efficient Interaction Technique to Explore Radial Hierarchy Visualizations on Touch Screens

Mouse hover interaction is commonly employed in data visualization designs and applications on desktops. It is one of the effective solutions for the "presentation problem"- insufficient space restricts the user in showing detail and context contemporaneous [69]. While hover interaction is so universe with desktop and mouse for its effortless to perform and cheap to implement, it still remains un-developed or overlooked with touch screens, or just awkwardly translated by a tap.

2014 Pew Research survey estimated that 42% of the American adults own a tablet [70]. Portable and touch enabled devices such as smartphones and tablets are bringing new possibilities to people's life. Touch interactions outperform traditional desktop and mouse interactions in many ways, such as increased bandwidth, additional degrees of freedom to express user's intentions, panel free, kinetic feedback, and touch unlocks the limits of time, space and background of the user.

Yet touch interaction also has its limitations and tradeoffs. The lack of tracing cursor on the screen makes it hard to implement 'hover to highlight' or 'hover to see more detail'. One can never tap each one of the data points, as a visualization can easily contain hundreds of data points. Besides, the targets might be too small to be precisely selected, which is known as 'fat finger' problem [55], [64]. What is worse, the hand itself would obscure the view.

We are introducing radial cursor to address the above problems. Radial cursor enables people to hover over radial hierarchical trees easily and neatly. See figure 6-1 (b), with one dominant hand controlling the angle of cursor for sequential browsing, another hand controls the level of current cursor simultaneously. We used radial metaphor for the cursor, and stimulated physical switch with fixed levels for level slider.
Radial cursor embraces the concept of focus plus context [71] by arranging the information on three views: focus view, detail view and context view, see figure 6-1 (a). Together with the cursor interaction, the design enable viewers to see the object of primary interest presented in full detail while at the same time getting an overview - impression of all the surrounding information or context available [72].

Our design also improves the legibility of large radial tree on relatively small screens. By showing part of the tree at one time, it encourages people to start exploring immediately instead of staring at a whole big tree and wonder where to start. Also, all the text gets to align in the same and easy-to-read direction, see figure 6-1 (c).

(a) (b) (c)

Figure 6-1 Screenshots and photo of radial cursor

(a): screenshots with annotations; (b): the nature way to play with radial cursor; (c): radial node-link layout. The demo video of radial cursor can be accessed through https://youtu.be/rugrzMUxXLw

In brief, the main contributions of radial cursor include:

- cheap implementation of 'hover' interaction on touch devices with no occlusions
- mitigate 'fat finger' problem
- improve legibility of large radial tree on tablet-sized screens

6.1 Design and implementation

As a precondition process, we first investigated related works regarding mouse hover solutions and interactive tree visualizations on small size screens.

Fluid DTMouse [73] uses the average of two finger touch points as the displayed position of the cursor. However, it takes too much effort to hold the gesture and move the cursor. Besides, it brings a lot occlusion. Shadow tracking [74] presents another typical solution for implementing cursor positioning and engagement on touch devices: with the help of additional devices. In this case, those are ceiling-mounted light and rear-mounted camera, users can control the cursor without touching the table. But we are seeking technique with no need of extra devices or extra effort.

Most of the designs and systems on tablets/smartphones concentrate on two key aspects: space-efficiency and legibility. Tablorer [75] proposed expandable table to visualize hierarchical tree structure, where a sibling blocks contains all the siblings under one parent node, and a sibling block is displayed either horizontally or vertically according to their levels. Tablorer also employs the principles of focus plus context. Radial Edgeless Tree (RELT) [76] recursively partition the display area, and parents locate at the upper left direction while their children locate at the low right direction. The interactions require surrounding buttons for direction controlling and center button for new root selection.

6.1.1 Iterative Prototyping and Testing

For the layout strategy, we initially built two prototypes. One shows a quarter of the radial tree which centers at the corner of the screen, another shows half of the tree. Then we invited some co-workers to try out both of them. They preferred the quarter one because it felt more intuitive and natural to move their finger on the corner of the screen. Besides, we are able to put context view and detail view on the screen due to the smaller space occupation of quarter layout.

There came another problem, this prototype only supported the change of cursor's angle, that is, only the leave nodes can be accessed, which hinders its adaptability. We need an effective interaction to change between levels.

We brainstormed a list of solutions: 1) on-screen buttons indicating each level, which apparently didn't really leverage touch interaction; 2) quick swipe within the touch area. This solution was implemented for testing, but it was not working well due to the limited size of touch area. We had to suspend the project for a while with no luck of finding the proper solution. Eventually we came up with this level slider, which can be controlled simply by left hand sliding.

What is worth mentioning, all the participants in the process learnt how to play with radial cursor immediately with little instruction, and enjoyed the experience.

6.1.2 Visual Components of Radial Cursor

Cursor controllers

- *Interested item/node* is the node being highlighted.
- *Touch area* responds to cursor scrolling action, which controls the cursor's sequential position within the same level. Touch area locates at the right bottom corner, overlaying where the root node should be. And it is where thumb movement is the easiest to perform.
- *Level switch* is used for the relative changing of the level of current cursor, which locates at the right bottom corner to facilitate bimanual operation. A slider will be triggered once second finger touches the activating area, showing the current level (opaque polygon), position of the touch point (white circle), and options for changing the level (semi-opaque polygon).

Views applying focus-plus-context

- *Focus view* shows the structural information of the interested item by showing one fourth of the entire radial tree.
- *Detail view* exhibits further textual or visual information of the interested node. It would be activated as the cursor being triggered, and keep updating as the cursor moves. For instances,

detail view of flare package shows the size of the corresponding file, detail view of Flicker shows the simulated pictures been posted by the individual, and detail view of flow diagram of web pages shows the preview of the web pages.

- *Context view* is an overview showing the current positions of the first-level nodes (e.g., the disks cdefg if it is the visualization of file system of Windows). It indicates where you are by highlighting the ancestor node of the currently interested node.

Layout controller

- *Layout switch* is used for switching between two layouts: sunburst and node-link. We will talk more about it later.

6.1.3 Access All the Nodes by Bimanual Operation

As shown in figure 6-1 (b), our proposed technique achieves the accessibility of all the nodes in two dimensions: angle and radius. One can scroll his/her finger within the touch area to move the cursor among different nodes at the same level. While that finger is still on the screen, use a finger of left hand to travel through different level: right/downward to the parent of currently highlighted node, or left/upward to its children at the precise radial position. One can also rotate the layout as a wheel to change the focus view.

6.1.4 Radial Layouts: Node-link and Sunburst

Circular tree layout makes it easy to visualize mid-sized trees (up to several thousand leaves).
Radial node-link tree achieves efficient, tidy arrangement of layered nodes. The root node of the tree is at the center, with leaves on the circumference. The angle of each arc corresponds to its count of children. The depth of nodes is computed by distance from the root, leading to a ragged appearance.
Sunburst layout is designed by John Stasko [77]. A sunburst is similar to the treemap, except it uses a radial layout. We are reusing the space distribution of the radial node-link tree.

6.2 Use cases

Radial cursor can be applied to different hierarchical data ranged from file system to structural diagram of web pages. It can fit as small as 30 nodes, and as large as 400 nodes. Radial cursor works perfectly where mouse hover is applicable, and serve well for those who trying to get a general sense of an unfamiliar data quickly, or those who trying to learn more of or constantly reviewing a known-complex-data.

6.2.1 Flickr Personal Taxonomies

Flickr, a photo sharing service, allows users group related photos into sets and related sets within collections. We extracted data of one user (uid=237, number of paths=361) from 7,121 Flickr users who belong to wildlife and nature photography public groups of the table anonym_folder [78]. Let's assume this user named Lee, and has been updating photos to Flickr for more than 2 years. We implemented this dataset to Radial Cursor, where the detail view shows an preview of the picture, see figure 6-2 (a).

Lee wants to select some photos to make personalized postcards. He only has a rough idea of what an ideal collection should be: each image is unique both in the hue and the subject.

He quickly scrolls through the photos, 'my photos looks better than I had imaged', says Lee. Then he slows down so that he can review the photos one by one and more carefully. Lee really likes the one taken at the beach during the journey to Singapore (node path: journey->asia->singapore)'. But he recalls that this is not the only one with the blue-green sea and golden sand, similar image have flashed once somewhere, maybe a quarter away. He would have to remember the exact path, and click back and forth through all the nested directories if the images are stored in the traditional file system style. Luckily with Radial Cursor, he just need to rotate the wheel-like focus view a bit anticlockwise, the context view rotates accordingly. Then Lee starts searching for the candidate image by scrolling the cursor. A couple of seconds later, Lee found it, and settled with the Singapore beach one as the first postcard of the collection. Then he continued with his searching for the rest ones.

May is a user of Flickr, who is reviewing Lee's uploaded photos using radial cursor. She scrolls through the photos and meanwhile perceives the structural information, ie. flora_fauna->animal->dolphin. Within ten seconds, May gets to understand Lee's preference of what kind of photo to upload and how he organizes them, so that she can decide whether to follow Lee or not.

6.2.2 Other Applicable Dataset

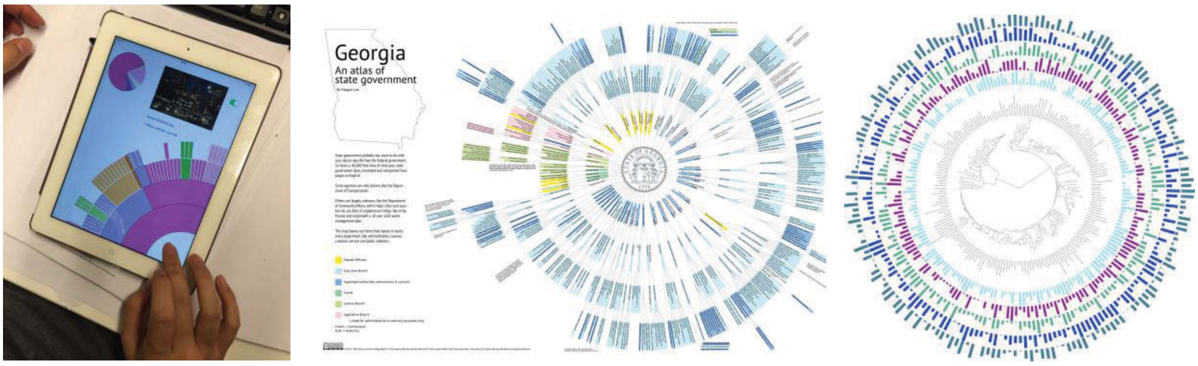
Flare Package

This data is used to build the first demo as shows in figure 6-1. Each leaf node represents a file in the Flare [79] source code. And the hierarchy tree structure indicates package membership. The 'Size' property refers to the size of the source code in bytes, which serves as the only detailed information showed at detail view.

Besides Flare package, one can also use radial cursor to explore open code sources from Github, and get the general sense of the code quickly. Detail view can show the functions, variables or any other information of the interested file.

Federal Budget Data

This is publicly available data from the Congressional Budget Office [80]. The goal is to allow a user to quickly see the hierarchical flows of spending in our government, at the federal, state, and local level. It is now implemented using D3 [81].



(a) (b) (c)

Figure 6-2 Use case studies

(a): A photo taken during the case study of Flicker personal taxonomies; (b): Branches of government [82]; (c): Multi value bar chart of phylogenetic trees [83]

Branches of Government

The radial node-link tree created by Lee shows the state government of Goegia [82], see figure 6-2 (b). A citizen who is interested in understanding the structures of government, or a freshman majored in politics will find radial cursor quite useful. They can actually start looking into the data quickly and clearly, and use the cursor to stay focus on one branch, meanwhile learning duty of that branch displaying at detail view.

Multi Value Barchart of Phylogenetic Trees

Interactive Tree Of Life [83] is an online tool for the display and manipulation of phylogenetic trees. Besides the tree structure, radial cursor can also be applied to the interaction with multi value bar chart, see figure 6-2 (c).

Flow Diagram of Web Pages

Diagram is an essential tool for communicating information architecture and interaction design in Web development teams. Information architects and interaction designers use them to develop detailed navigational and interface requirements for each page. And project sponsors and managers use them to obtain a general sense of the scope and form of the project. Radial cursor can help with this process by provide cursor over the page structure, and showing the preview of the highlighted page.

Chapter 7 Conclusion

Touch-enabled applications are ubiquitous in today's world. They liberate users from rigid analytical activities and bring new opportunities. During the research, aiming to investigate how to best leverage touch interaction in data visualization, we proposed a classification touch action styles based on our reviewed works. And we further organized the seven styles into three categories based on data dependency as follows:

- data dependent: direct-on-data manipulation, virtual instrumental interaction, unique techniques;
- data independent: touch panel;
- mixed: multi-modal interactions; gestural trigger, drawing gestures.

Then we organized the literature review according to this proposed framework. As mentioned in background section, we did not consider systematically the role of display size in visualization, nor the relating issues such as social environment. Meanwhile, number of the applications applying interactive surfaces to visualization is still limited at the time we conduct the survey.

Chapter three introduces the research methodology our study used. Both as problem-driven studies, visualization project and HCI industry project share similar process such as goal definition and specification, designing and prototyping, and testing and evaluation. Visualization project also highlights the pre-step of literature review and knowledge gaining, and the paper composing as extra individual deliverable and contribution to the community. The investigation of data science methods serves as user research and problem inspiration, as well as a practical handbook for data acquiring and wrangling.

To improve the interactions for DA-TU, we proposed a new set of gestural commands. And we implement our design as an iPad application. To demonstrate our research hypothesis, we have conducted two preliminary usability studies. The first study was conducted to validate and refine the design. Generally the gesture actions map well to user's intent according to the survey result and observation. But the action to trigger the create operation is too difficult to perform for more than half the participants, which result in the 2nd version sets with an edit mode. Then the second study was conducted to compare quantitively between old and new interfaces. We had participants performing two set of tasks on the two interfaces and recorded the time used to complete them. The results show that the new gestural interface has greatly improved the HCI process in DA-TU, both in efficiency and user experience.

However, the number of participants involved in usability study needs to be increased in the future. We also plan to use several scientific methods to evaluate the overall approach as well as each gesture vocabulary one-by-one in deep details. At the same time, we will further transplant proposed NUI based

DA-TU to Smart Table, a device that support up to 40 touch points for implementing multi-user collaborative interactions.

We also proposed an innovative touch interaction technique, Radial Cursor. Radial cursor supports hover interaction on touch screens, which works particularly well with hierarchical data. With Radial Cursor, user can peek at their interested item through detail view, and meanwhile has a clear idea of where they are with the help of focus view and context view. Without occlusions caused by hand, Radial Cursor implement a cheap and easy mouse hover interaction on touch screens. Besides, radial cursor makes it easier to access nodes from a large tree, which is usually represented as tiny items on the screen. Moreover, large radial trees become more readable thanks to the layout partition and better text alignment. To validate our design, we performed a use case study with the Flickr personal dataset. And Radial cursor was effective and playful to use during the process.

In the future, we plan to employ a dynamic collapsing display form, which makes it possible for radial cursor to handle larger data structures. And for those situations where changes of level are not needed, for instance, viewing images as leave nodes from the Flickr taxonomy, radial cursor can also be adopted to smart phone applications.

Regarding to the question of what kind of data is suitable for visualization on ITSs. We can tell from the visualization from figure 2-1 that spatial data is constantly represented as map (though not obvious here, but already shows in previous studies[7]), relational data as graph (node-link particularly) and discrete data as thumbnail. They are common and mature types being interacted by fingers. And there lies great potential and innovative ways to interact with abstract data visualized as charts (e.g., multi-dimension data).

We also discussed the new possibilities and unique values provided by touch interactions, with evidence from the survey. These values are 1) physical metaphor; 2) increased input bandwidth; 3) additional degree of freedom to express intentions; 4) panel free; 5) effective zooming/scaling; 6) trajectory enabled; 7) kinetic feedback; 8) collaborative work; 9) high resolution; 10) anywhere, anytime, anyone and 11) sensors. But we are not suggesting that mouse should be replaced. As Bill Buxton [6] puts it, mouse is great for many things, just not everything. The challenge with new input should be finding devices that work together with the mouse, or things that are strong where the mouse is weak, thereby complimenting it. Therefore, we also listed where are the tradeoffs of touch interactions as 1) worse precision; 2) education; 3) occlusions; 4) no self-explaining; 5) low typing speed and high typing error; and 6) hard to implement highlight.

Publication

He, Shizhe, Huang, Mao Lin, & Zhu, Lin. Natural User Interface Design in DA-TU: An Interactive Clustered Data Visualization System. In *19th International Conference on Information Visualization*, 2015

He, Shizhe, Huang, Mao Lin. Radial Cursor: an Efficient Technique to Explore Radial Hierarchy Visualizations on Touch Screens. In *27th Australian Conference on Human-Computer Interaction (OzCHI),* 2015

Reference

[1] S. CARD, J. MACKINLAY, and B. SHNEIDERMAN, "Information visualization," in *Readings in information visualization: using vision to think*, 1999, pp. 1–36.

[2] M. Sedlmair, M. Meyer, and T. Munzner, "Design study methodology: Reflections from the trenches and the stacks," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2431–2440, 2012.

[3] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*, 1996, pp. 336–343.

[4] E. A. Johnson, "Touch displays: a programmed man-machine interface," *Ergonomics*, vol. 10, no. 2, pp. 271–277, 1967.

[5] P. Isenberg, T. Isenberg, T. Hesselmann, B. Lee, U. Von Zadow, and A. Tang, "Data visualization on interactive surfaces: A research agenda," *IEEE Comput. Graph. Appl.*, vol. 33, no. 2, pp. 16–24, 2013.

[6] B. Buxton, "Multi-Touch Systems that I have known and Loved," 2007. [Online]. Available: http://www.billbuxton.com/multitouchOverview.html.

[7] P. Isenberg and T. Isenberg, "Visualization on Interactive Surfaces: A Research Overview/Visualisierung auf interaktiven Oberflächen: Ein Forschungsüberblick," *i-com*, vol. 12, no. 3, pp. 10–17, 2013.

[8] C. Valdes, D. Eastman, C. Grote, S. Thatte, O. Shaer, A. Mazalek, B. Ullmer, and M. K. Konkel, "Exploring the design space of gestural interaction with active tokens through user-defined gestures," *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, Toronto, Ontario, Canada, pp. 4107–4116, 2014.

[9] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, "User-defined gestures for surface computing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009, pp. 1083–1092.

[10] P. Isenberg and S. Carpendale, "Interactive tree comparison for co-located collaborative information visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1232–1239, 2007.

[11] W. Yee, "Potential limitations of multi-touch gesture vocabulary: Differentiation, adoption, fatigue," in *Human-Computer Interaction. Novel Interaction Methods and Techniques*, Springer, 2009, pp. 291–300.

[12] S. M. Drucker, D. Fisher, R. Sadana, J. Herron, and M. c. Schraefel, "TouchViz: a case study comparing two interfaces for data analytics on tablets," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Paris, France, pp. 2301–2310, 2013.

[13] R. Sadana and J. Stasko, "Designing and implementing an interactive scatterplot visualization for a tablet computer," *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*. ACM, Como, Italy, pp. 265–272, 2014.

[14] J. S. Yi, Y. ah Kang, J. T. Stasko, and J. A. Jacko, "Toward a deeper understanding of the role of interaction in information visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1224–1231, 2007.

[15] M. Walker, "Information Visualization beyond Mouse and Keyboard," *Beyond Deskt.*, p. 55, 2013.

[16] B. Shneiderman, "direct manipulation: a step beyond programming languages," *Sparks Innov. human-computer Interact.*, vol. 17, 1993.

[17] S. Schmidt, M. A. Nacenta, R. Dachselt, and S. Carpendale, "A set of multi-touch graph interaction techniques," in *ACM International Conference on Interactive Tabletops and Surfaces - ITS '10*, 2010, p. 113.

[18] D. Baur, B. Lee, and S. Carpendale, "TouchWave: kinetic multi-touch manipulation for hierarchical stacked graphs," *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*. ACM, Cambridge, Massachusetts, USA, pp. 255–264, 2012.

[19] J. M. Rzeszotarski and A. Kittur, "Kinetica: naturalistic multi-touch data visualization," in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, 2014, pp. 897–906.

[20] M. Beaudouin-Lafon, "Instrumental interaction: an interaction model for designing post-WIMP user interfaces," *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, The Hague, The Netherlands, pp. 446–453, 2000.

[21] D. P. K, #228, ser, M. Agrawala, and M. Pauly, "FingerGlass: efficient multiscale interaction on multitouch screens," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Vancouver, BC, Canada, pp. 1601–1610, 2011.

[22] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann, "A Survey on Interactive Lenses in Visualization," in *EuroVis-STARs*, 2014, pp. 43–62.

[23] B. Kondo and C. Collins, "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation," *IEEE Trans. Vis. Comput. Graph.*, vol. PP, no. 99, p. 1, 2014.

[24] C. Kelleher and G. Grinstein, "The Fractal Perspective Visualization Technique for Semantic Networks," in *15th International Conference on Information Visualisation (IV)*, 2011, pp. 211–215.

[25] K. Wetzel, "Pebbles-using circular treemaps to visualize disk usage," *URL http//lip. sourceforge. net/ctreemap. html*, vol. 2, 2003.

[26] J. de Jesus Nascimento da Silva Junior, B. S. Meiguins, N. S. Carneiro, A. S. G. Meiguins, R. Y. da Silva Franco, and A. G. M. Soares, "PRISMA Mobile: An Information Visualization Tool for Tablets," in *16th International Conference on Information Visualisation (IV), 2012*, 2012, pp. 182–187.

[27] C. Xiang, A. D. Wilson, R. Balakrishnan, K. Hinckley, and S. E. Hudson, "ShapeTouch: Leveraging contact shape on interactive surfaces," in *3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008.* , 2008, pp. 129–136.

[28] M. Frisch, J. Heydekorn, and R. Dachselt, "Investigating multi-touch and pen gestures for diagram editing on interactive surfaces," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09*, 2009, p. 149.

[29] B. Lee, R. H. Kazi, and G. Smith, "SketchStory: Telling more engaging stories with data through freeform sketching," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2416–2425, 2013.

[30] J. Walny, B. Lee, P. Johns, N. H. Riche, and S. Carpendale, "Understanding pen and touch interaction for data exploration on interactive whiteboards," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2779–2788, 2012.

[31] H.-C. Jetter, J. Gerken, Z. Michael, #246, llner, H. Reiterer, and N. Milic-Frayling, "Materializing the query with facet-streams: a hybrid surface for collaborative search on tabletops," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Vancouver, BC, Canada, pp. 3013–3022, 2011.

[32] K. Hinckley, K. Yatani, M. Pahud, N. Coddington, J. Rodenhouse, A. Wilson, H. Benko, and B. Buxton, "Pen+ touch= new tools," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 2010, pp. 27–36.

[33] A. C. Long Jr, J. A. Landay, and L. A. Rowe, "Implications for a gesture design tool," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 1999, pp. 40–47.

[34] "iOS Developer Library." [Online]. Available: https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIGestureRecognizer_Class/index.html.

[35] Y. Magallanes, A. Molina, and J. A. Sanchez, "Combining gestures and graphical elements for collaboration using multi-touch surfaces," in *22nd International Conference on Electrical Communications and Computers (CONIELECOMP), 2012* , 2012, pp. 173–178.

[36] J. Leek, "datasharing," 2013. .

[37] D. Jannach and M. Zanker, *Recommender systems: an introduction*. Cambridge University Press, 2011.

[38] J. McLane, "A Chart of Lecture 1.3 - Evaluating Designs and Methods for Evaluation." [Online]. Available: https://d396qusza40orc.cloudfront.net/hci%2FResourcesFromStudents%2FHCI-1point3.PNG. [Accessed: 26-Jun-2015].

[39] P. R. Keller and M. M. Keller, *Visual cues: practical data visualization*, vol. 2. IEEE Computer Society Press Los Alamitos, 1993.

[40] A. Hernando, R. Moya, F. Ortega, and J. Bobadilla, "Hierarchical graph maps for visualization of collaborative recommender systems," *J. Inf. Sci.*, 2014.

[41] R. J. K. Jacob, A. Girouard, L. M. Hirshfield, M. S. Horn, O. Shaer, E. T. Solovey, and J. Zigelbaum, "Reality-based interaction: a framework for post-WIMP interfaces," in *Proceedings of the*

SIGCHI conference on Human factors in computing systems, 2008, pp. 201–210.

[42] N. Elmqvist, A. Vande Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. J. Jankun-Kelly, "Fluid interaction for information visualization," *Inf. Vis.*, p. 1473871611413180, 2011.

[43] K. Kin, M. Agrawala, and T. DeRose, "Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation," *Proceedings of Graphics Interface 2009*. Canadian Information Processing Society, Kelowna, British Columbia, Canada, pp. 119–124, 2009.

[44] #228, R.-A. Albinsson, and S. Zhai, "High precision touch screen interaction," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Ft. Lauderdale, Florida, USA, pp. 105–112, 2003.

[45] D. Chivers and P. Rodgers, "Gesture-Based Input for Drawing Schematics on a Mobile Device," in *15th International Conference on Information Visualisation (IV), 2011* , 2011, pp. 127–134.

[46] A. Endert, L. Bradel, and C. North, "Beyond Control Panels: Direct Manipulation for Visual Analytics," *Comput. Graph. Appl. IEEE*, vol. 33, no. 4, pp. 6–13, 2013.

[47] A. Thudt, U. Hinrichs, and S. Carpendale, "The bohemian bookshelf: supporting serendipitous book discoveries through information visualization," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, 2012, p. 1461.

[48] M. Heilig, S. Huber, M. Demarmels, and H. Reiterer, "ScatterTouch: a multi touch rubber sheet scatter plot visualization for co-located data exploration," *ACM International Conference on Interactive Tabletops and Surfaces*. ACM, Saarbrücken, Germany, pp. 263–264, 2010.

[49] R. Kosara, "Poster: Indirect Multi-Touch Interaction for Brushing in Parallel Coordinates," *IEEE Trans. InfoVis. Comput. Graph*, 2010.

[50] T. Nagel, M. Maitan, E. Duval, A. Vande Moere, J. Klerkx, K. Kloeckl, and C. Ratti, "Touching transport - a case study on visualizing metropolitan public transit on interactive tabletops," *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*. ACM, Como, Italy, pp. 281–288, 2014.

[51] T. Isenberg, M. H. Everts, J. Grubert, and S. Carpendale, "Interactive exploratory visualization of 2d vector fields," in *Computer Graphics Forum*, 2008, vol. 27, no. 3, pp. 983–990.

[52] C. North, T. Dwyer, B. Lee, D. Fisher, P. Isenberg, G. Robertson, and K. Inkpen, "Understanding multi-touch manipulation for surface computing," in *Human-Computer Interaction–INTERACT 2009*, Springer, 2009, pp. 236–249.

[53] N. Elmqvist, M. Hlawitschka, and J. Kennedy, "Eliciting Multi-touch Selection Gestures for Interactive Data Graphics," 2014.

[54] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale, "Beyond mouse and keyboard: Expanding design considerations for information visualization interactions," *IEEE Trans. Vis. Comput. Graph.*,

vol. 18, no. 12, pp. 2689–2698, 2012.

[55] D. Wigdor and D. Wixon, *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier, 2011.

[56] T. Nagel, E. Duval, and F. Heidmann, "Visualizing geospatial co-authorship data on a multitouch tabletop," in *Smart Graphics*, 2011, pp. 134–137.

[57] Y. A. Ivanov, C. R. Wren, A. Sorokin, and I. Kaur, "Visualizing the History of Living Spaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1153–1160, 2007.

[58] K. Ryall, N. Lesh, T. Lanning, D. Leigh, H. Miyashita, and S. Makino, "QueryLines: approximate query for visual browsing," in *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, 2005, pp. 1765–1768.

[59] M. Wattenberg, "Sketching a graph to query a time-series database," in *CHI'01 Extended Abstracts on Human factors in Computing Systems*, 2001, pp. 381–382.

[60] P. Dourish, *Where the action is: the foundations of embodied interaction*. MIT press, 2004.

[61] J. Wagner, E. Lecolinet, and T. Selker, "Multi-finger chords for hand-held tablets: recognizable and memorable," in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, 2014, pp. 2883–2892.

[62] D. F. Keefe and T. Isenberg, "Reimagining the Scientific Visualization Interaction Paradigm," *IEEE Comput.*, vol. 46, no. 5, pp. 51–57, 2013.

[63] S. Sadowski and F. Heidmann, "A Visual Survey of Information Visualizations on Smartphones," 2014.

[64] S. Voida, M. Tobiasz, J. Stromer, P. Isenberg, and S. Carpendale, "Getting practical with interactive tabletop displays: designing for dense data, fat fingers, diverse interactions, and face-to-face collaboration," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, 2009, pp. 109–116.

[65] A. Sears and B. Shneiderman, "High precision touchscreens: design strategies and comparisons with a mouse," *Int. J. Man. Mach. Stud.*, vol. 34, no. 4, pp. 593–613, 1991.

[66] K. Vertanen, H. Memmi, J. Emge, S. Reyal, and P. O. Kristensson, "VelociTap: Investigating Fast Mobile Text Entry Using Sentence-Based Decoding of Touchscreen Keyboard Input," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 659–668.

[67] M. L. Huang and P. Eades, "A fully animated interactive system for clustering and navigating huge graphs," in *Graph Drawing*, 1998, pp. 374–383.

[68] P. Eades and M. L. Huang, "Navigating clustered graphs using force-directed methods," *J. Graph Algorithms Appl.*, vol. 4, no. 3, pp. 157–181, 2000.

[69] J. C. Dürsteler, “Focus+Context,” 2002. [Online]. Available: http://www.infovis.net/printMag.php?num=85&lang=2.

[70] K. Zickuhr and L. Rainie, “Tablets and E-reader Ownership,” *PewResearch*, 2014. .

[71] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[72] InfoVis: Wiki Team, “InfoVis: Wiki.” [Online]. Available: http://www.infovis-wiki.net/index.php/Focus-plus-Context.

[73] A. Esenther and K. Ryall, “Fluid DTMouse: better mouse support for touch-based interactions,” in *Proceedings of the working conference on Advanced visual interfaces - AVI ’06*, 2006, p. 112.

[74] F. Echtler, M. Huber, and G. Klinker, “Shadow tracking on multi-touch tables,” in *Proceedings of the working conference on Advanced visual interfaces - AVI ’08*, 2008, p. 388.

[75] H. Shin, G. Park, and J. Han, “Tablorer--An Interactive Tree Visualization System for Tablet PCs,” in *Computer Graphics Forum*, 2011, vol. 30, no. 3, pp. 1131–1140.

[76] J. Hao and K. Zhang, “A Mobile Interface for Hierarchical Information Visualization and Navigation,” in *2007 IEEE International Symposium on Consumer Electronics*, 2007, pp. 1–7.

[77] J. Stasko and E. Zhang, “Focus + Context Display and Navigation Techniques for Enhancing Radial , Space-Filling Hierarchy Visualizations,” 2000.

[78] A. Plangprasopchok, K. Lerman, and L. Getoor, “Growing a tree in the forest: constructing folksonomies by integrating structured metadata,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 949–958.

[79] UC Berkeley Visualization Lab., “flare data visualization for the web,” 2009. [Online]. Available: http://flare.prefuse.org/.

[80] “Congressional Budget Office.” [Online]. Available: www.cbo.gov.

[81] brightpointinc, “Federal Budget.” [Online]. Available: http://www.brightpointinc.com/interactive/budget/index.html?source=d3js.

[82] M. Lee, “Branches of Government,” 2014. [Online]. Available: http://greencracker.net/wp-content/uploads/2013/11/final-cafe-press-dimensions-1106.jpg.

[83] “iTOL.” [Online]. Available: http://itol.embl.de/.