Interactive visualization on small screens

Siavash A. Bigdeli
University of Neuchatel
Siavash.arjomand@unine.ch

ABSTRACT

In this paper*, problems related to interaction with small screens and visualization of data in such displays is discussed. Two-step zooming and summary thumbnails are introduced for visualizing large data to user. In the case of interacting with the device, two feedback-based methods with sound and vibration have been shown. A pseudo-transparent display that enables the user to interact from the behind of the screen is discussed as well. Last but not least a user-defined gesture study has been reviewed in details.

Keywords

Smartphone, Small screen, Interactive visualization

1. INTRODUCTION

The word is stepping from personal computers to mobile devices (e.g. Smartphones, Portable game consoles and etc.). It is their portability which makes them more applicable and demanding. Although the screen resolution is increasing, the physical size will never be comparable to desktop in any case. Nowadays memory and processor of these devices are improving and soon they will have the power to run many complex interactive applications.

While people were used to look at big screens while interacting with devices or scrolling and typing comfortably with mouse and keyboard, they are now interacting with small interfaces as size as their hand for all the mentioned tasks.

Most of the applications being used today have complex interfaces with toolbars, pallets, help window and etc. All take a lot of space even without considering the workspace. Also in simple desktop, user can have multiple gadgets in the screen. These gadgets can be applications, web browsers, weather widgets and etc. But now when using a portable device users are not being able to appreciate much in their screen and they are forced to go throw back and forth pages or scroll a large list for viewing an entity. All of these facts indicate desktop applications and data visualization technics cannot be used for small screens and they should be modified as well.

On the other hand, users will be obligated to interact with a device as the size of their hand. Giving an input to these small devices is not convenient. Touch screens have made it easier to interact with small screens because they save the space by combining the keyboard and the display. However screen-keyboard with comparably large fingertip makes typing hard and tiresome. When a user wishes to click on the screen, his fingertips occlude the graphical elements that he is working with.

Highly affective approaches to overcome related issues to small screens have been discussed in section 2) Device communication: for the problems with visualizing large data and section 3) User interaction: to describe commanding methods and input modalities. The introduced feedback-based methods are also included in section 3 since they are in the process of inputting data.

2. DEVICE COMMUNICATION

This type of communication is required in case of data visualization, notification, feedback and similar tasks. The main used feature of the device is the screen to show the data. However sound and vibration notification were generally used since the creation of these devices. Multimodality is very effective for notifications like ringing plus vibration or showing the text message plus a short sound to attract the user's attention. In this section two different models are discussed which help visualizing big interfaces (images, webpages and etc.).

2.1 A Comparison of Fisheye, Zoom, and Panning Techniques

The research done in [1] is a comparison of three different methods for visualizing information on small screens which are described below:

Panning: This is a 'sliding window' technique. The big interface is shown in its original size but only part of the interface is shown in the screen. To view other parts of the interface, user is able to shift the screen by moving the background.

Two-level zoom: In this technique there are two different magnifications for user: an overview, which shows the entire screen in a small size; and a zoomed view, which is the same as the panning system. Users can switch between the two magnifications. In the overview mode user is able to see the rectangle that is going to be showed in zoomed view.

Fisheye view: in this approach an overview of the entire large screen, like the two-level zoom is shown which also includes a detail region inset into the same view. A nonlinear magnification of the interface is used to provide smoothness of the overall screen. User is also able to move the magnification which is set to show the source screen at normal size.

The tasks that have been chosen in proposed method in [1] covered a range of pointer-based interactions like editing a document, navigating the web, and monitoring a control panel.

The results in [1] have shown that:

- Even in best case scenario, navigation on the small screen is considerably slower compare to the normal screen.

* This paper is a deliverable of the Msc research seminar on "Multimodal Interaction on Mobile Devices", organized in 2012 in the department of Informatics of the University of Fribourg:

https://diuf.unifr.ch/main/diva/teaching/seminars/seminar-multimodal-interaction-mobiles-devices

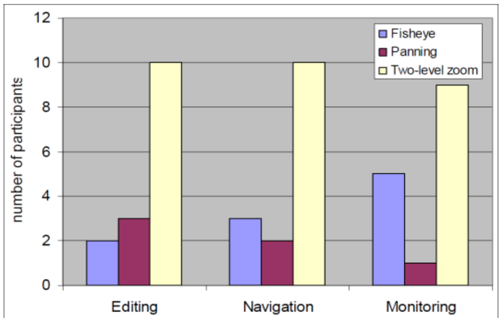

Figure 1. User preferences for the experiment in [1].

- Having an overview look of the entire interface is important, both because the global context allows faster navigation and because many interactions can be carried out at the overview level.
- Fisheye views and two-level zoom systems are both effective techniques for navigating interfaces.
- Panning strategies perform poorly for large movements and are disliked by users.
- As it is shown in figure 1, users like the two-level zoom, even when its performance is less than optimal.

2.2 Summary thumbnails

The authors in [2] have proposed a method specifically for web content visualization on mobile devices. Unlike techniques that have been discussed in previous section, their approach is to change the normal view of a web page somehow that user can remember viewed material and distinguish the difference between visually similar areas. "Summary Thumbnails are thumbnail views enhanced with fragments of readable text" as the authors in [2] defined.

By preserving the original page layout, Summary thumbnails allow users to identify the overall appearance and structure of the pages. The readable text added to these thumbnails eliminates user's need to zoom and also provide assistance to identify similar stories.

In order to fit the webpage into the small screen, they used less text than the original webpage. However when the user zooms in, the thumbnail is replaced with full content. In this way if some parts of the web page interest the user, by zooming he can read the overall document.

Below is the detailed description of the proposed algorithm in [2]. After the page is loaded, a user-defined threshold is used to enlarge small texts. The generated page has same width of the original page; therefore it has extra lines than the old one due to having large texts. Then, the program will start removing text until the number of lines is the same as the old page and finally the page is saved in different formats (e.g. HTML, bitmap) for different devices.

For their experiment, they surveyed participants for their preferences and 9 out of 12 participants preferred Summary thumbnails. With summary thumbnails, participants zoomed 59% comparing to normal thumbnails and 51% scrolled less than using a normal desktop interface. Also participants were 41% faster using proposed method than Single column interface.

3. USER INTERACTION

In the case of user interaction problem, users also need to interact with these small devices and using the old fashion ways such as mouse and keyboard, which are not available since carrying theme will question the whole portability feature. As the popularity of such devices are growing, users tend to expect more than dialer keyboards and want to experience most computer features in them while lack of screen size obligate users to work with small buttons, sometimes smaller than a fingertip. Additionally, these devices brought new applications to daily life which will be better performed with new modalities that normal PCs cannot cover. There are researches about how to improve this interaction and what new modalities can replace the old ones. In this section new approaches that cover this issue is widely discussed and their results indicates their effectiveness.

3.1 Sonically-enhanced buttons

The work in [3] presents a feedback feature for touch commands and describes a small pilot study and two formal experiments to investigate the usability of these sonically-enhanced buttons. The main hypothesis is that presenting sound information about the buttons will improve usability by decreasing workload and by increasing the amount of input data which will let the developers to use smaller buttons in data entry type applications without much loss in quantitative performance.

For the experiment, they used a 3Com Palm III PDA via a stylus for all participants. And two different button sizes to compare if the smaller ones (4x4 pixels) with sound would be as effective as the larger ones (16x16 pixels) without sound.

The task was that each participant had to enter 5 number digits that was randomly generated (but same for everyone). They maximized the number of button taps and stylus movement by forcing the participants to press an Ok button after each digit has been pressed.

As the result of the pilot study, in general, sounds were useful and enhanced sounds performed the same or better than the basic ones. The buttons with sound allowed more data to be entered than those without. But standard size buttons without sound still performed better that the smaller ones with sound.

The quantitative results of the first experiment also back up the hypothesis that sounds will improve usability as participants were able to enter more codes with sound feedback for both button sizes in the experiment. But again the experiment conclusion in [3], yield to "The effect of size out-weighed the effect of sound". But the qualitative results showed that sound can significantly reduce workload. Also as an annoyance factor of this experiment, sounds did not annoy participants and surprisingly buttons with sound were rated less annoying than normal buttons.

The second experiment was just varying in environment and it took place in a street which simulated the normal outdoor environment to get a more realistic result. The result in table 1 also indicated that sonically-enhanced buttons perform better than the ones without sound (with the same size) and also in this environment there were no significant difference between small buttons with sound and the standard salient buttons (Table 1), therefore, in the daily basis, sonically-enhanced buttons can be an improvement for small screen devices.

Table . Results of two experiments in [3], mean number of entered codes and mean overall workload score

	Standard button		Small button	
	sound	**no sound**	**sound**	**no sound**
Experiment 1 (indoor)	58	45	42	30
Experiment 2 (outdoor)	42	32	28	18

3.2 Tactile Interface

Like the method in section 3.1, the proposed method of [4] developed a feedback interface for touch screens. Similar method of vibrating is nowadays used in most of the smartphones available in the market. The main goal is to provide a haptic feedback that gives the user the real feeling of pushing a key down. Visual and audio feedback (e.g. Section 3.1) can be very effective but not effective as feeling something under the fingertip since it is a touch interaction. On the other hand, touch responses much faster than vision and in a real environment audio feedbacks can be blocked by noise and can be inappropriate in some situations as well. This tactile feedback is to give the experience of pushing physical switches to feel GUI controls more real.

For the device they used a Sony Clié PDA and placed the actuators in the middle of the TFT and touch-sensitive glass which will push the user's fingertip from the screen. A soft silicon damper has been installed to prevent the device to shake with the actuators. These actuators have a small displacement but with fast acceleration they perform a nice feedback. Other mechanical and electrical detail of the tactile interface is out of the scope of this paper and can be found in [4].

Tactile feedback structure is shown in Figure 5 which is proposed in 5 states for each GUI:

- On touch down (T1): This is when the user clicked on an object which will provide a different effect for buttons and menus.
- On holding (T2): in this state actuators respond continuously in case of repeating action like scrollbar buttons and for other cases they have provided a low frequency vibration to inform the user that the GUI is still active.
- On dragging (T3): The same response for clicking has been applied to this state but in several iterations.
- On lift off: There is a click feedback in case the user lift from the button (T4) and no feedback in case of lifting outside of button boundary like the undo task (T5).

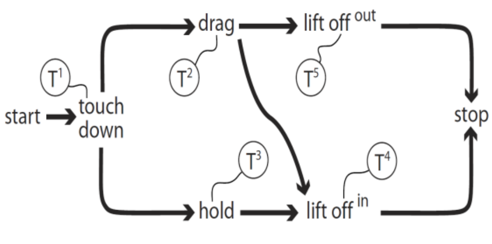

Figure 2. Feedback structure of tactile interface [4].

Three conditions have been chosen to test the effectiveness of Tactile feedback: audio feedback, Tactile feedback and without any feedback. As the results, Tactile was well-received by 9 participants out of 10 and the other one preferred audio feedback. All participants noted that they are more familiar with audio feedback and perhaps they need more time to get used to it. In holding down and dragging gestures, Tactile was much more effective since audio feedbacks cannot give the right feeling of them comparing to Tactile.

3.3 LucidTouch

While developers try to minimize the size of GUIs in the interface, the problem of occlusion will appear. Users lose their vision of the GUI object while trying to put their finger tip on it. Also in comparison with these small objects, fingertip is much larger which makes it harder to interact. For the solution to these problems, similarly to Tactile interface, the authors of [5] proposed the addition of a new feature to small screen devices to improve the interaction. To overcome this occlusion problem, they enabled the device to be interacted from the behind. In this method, the fingertips would not affect the visibility. However in case of see-trough screen, the background scenery will show up and can be a distraction. As shown in Figure 3, LucidTouch uses a pseudo-transparent screen that overlays a live image of users hand. Using this pseudo-transparent display not only solves the occlusion problem, but also the tracking issue. Simple computer vision algorithms have been used to visualize the fingertips touch point and also to use it as a multi touch screen.

In the design of this method, they used four layers in the display. (i) Front touch screen (ii) LCD display (iii) Multi-touch screen behind the display (iv) Micro-array imaging device embedded in the back of display to provide a 2D map of fingertip location.

A user survey has been carried out to test this device. Participants were asked to perform three tasks, Map browsing, Text entry and docking (selection + drag).

As the result of the test, participants found pseudo-transparency quite helpful. However 4 out of 6 reported difficulties in keeping track of the fingertip points. Also 4 out of 6 preferred LucidTouch to be included in precision required tasks (typing and docking). 3 of the participants preferred the video overlays in map browsing, 3 noted that pseudo-transparency made the task more difficult and 2 of them found the precise visualization of the fingertip point were unnecessary. "There is a need to vary the pseudo-transparency between, and possibly within, an application, or to otherwise modify the rendering so as to maximize awareness while minimizing intrusiveness." [5] Was the conclusion of these tests. As mentioned the overall feedback showed the usefulness of this device however its appearance in different tasks show be reviewed.

3.4 User defined motion gestures

In contrast with LusidTouch and Tactile interfaces, authors of [6] make use of sensors which are embedded in most of the small devices nowadays. The first normal gesture interaction is touch screen where it's a 2D gesture and the second one is a 3D gesture that could be performed by translating or rotating the device. The goal in [6] is to find natural 3D gestures for interaction. A test study has been done where the users where asked to define a gesture for a given task. The tasks were in two categories of Action tasks (Answering a phone call) and Navigation tasks (panning and zooming). Participants listed their preferred gestures and after that they did the gesture and the sensor signals were recorded. After this text, 380 gestures were

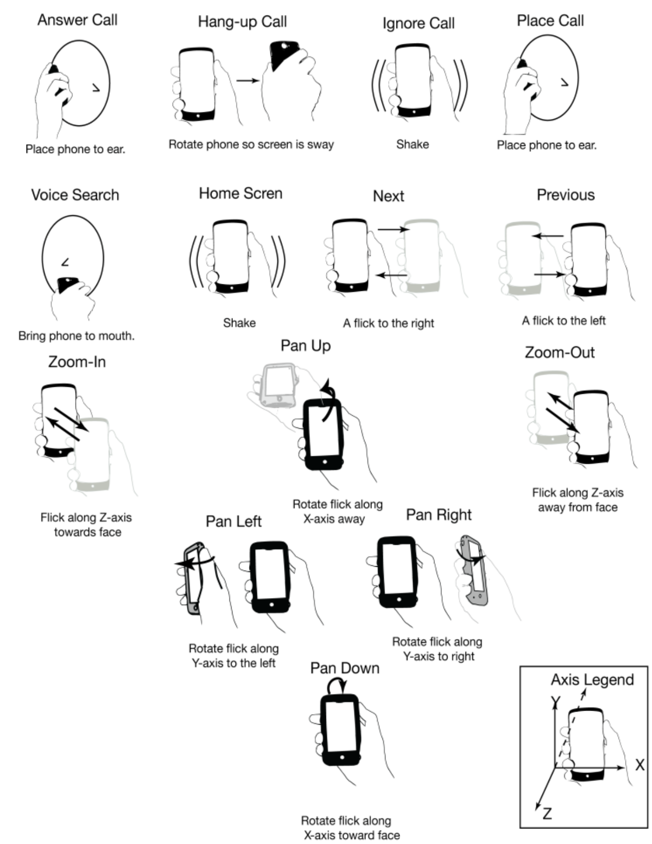

Figure 4. User defined Gestures by the proposed method in [6].

collected. The identical gestures for each task were grouped and the largest group was taken as the gesture for that task.

The results in [6] showed that, agreement on gestures varies from task to task which shows that most of the participants agree on using a specific gesture to answer a call, while a there were a lot of different gestures for next application task.

Figure 4 illustrates some of the user defined gestures that had the highest agreement in the survey. A flick is defined by a fast movement and returning back to the same position. Two navigation task of next and previous application were not shown because of their low agreement within the participants.

4. CONCLUSION

Finally, the problems related to the interaction with small devices have been categorized into user-device and device user interactions. Small screen size makes it difficult to interact with the device or to visualize data to the user. On the other hand, popularity of these devices makes the users to expect a nice interaction like they have with their PCs. Portability of such devices is their main reason of popularity, therefore attaching big user friendly interfaces like keyboard and mouse makes no sense.

In section 2, problems to visualize large data to user have been shown. Two general methods to overcome this issue have been discussed with their relevant results.

In section 3, interaction from user to the small devices have been covered with four different methods that each try to ease this interaction either by combining modalities (Sound and tactile) or attaching a tool to create new modality for interaction.

While some of the proposed methods to cover the visualization on small screens are general, most of methods have an application based benefits. The general ones perform weak on some applications that users prefer the traditional interaction and the application-based modalities perform week on normal tasks and other application.

The results show that adding a new model of interaction to the device can be useful, but on the other hand defining the input (e.g. gesture) is better to be left with the developer of the application. This makes it hard for developers since they have to implement the interaction system in their application. And for users, they need to learn the interaction system for each application. Additionally, with the large variety of sensors in such devices, combinations of modalities can also improver the interaction. As a semi-Multimodal approach, LucidTouch is good example of a utilization of more than one sensor at a time.

On the other hand several tasks can be done more easily with small screens. For example, autostereoscopic 3D displays perform better on small scale than large scale. This small size will assure the developers that only two directions is enough to display 3D object since the screen position according to the observer is highly predictable. However in large displays it is almost impossible to display 3D information for more than one person with same effect. Therefore these small screens can perform other tasks that cannot be done with large displays. Also there are some tasks and interactions that are proposed and developed after the popularity of these small screens such as Cardio trainer applications. This shows that beside the weakness of these small screens comparing to large ones, there are new and different tasks that are best performed by small ones.

5. REFERENCES

[1] Gutwin, C., Fedak, C. *Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques*. Proceedings of Graphics Interface 2004, pp. 145-152

[2] Lam, H., Baudisch, P. *Summary Thumbnails: Readable Overviews for Small Screen Web Browsers*, Proceedings of the CHI 2005, pp. 681-690.

[3] Brewster, S. A. *Overcoming the lack of screen space on mobile computers*. Personal and Ubiquitous Computing (6, 3) 2002, pp. 188-205.

[4] Poupyrev, I. and S. Maruyama. *Tactile interfaces for small touch screens*. Proceedings of UIST 2003, ACM Press 2003, 217-220.

[5] Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. *LucidTouch: a see-through mobile device*. Proceedings of UIST 2007, pp. 269-278.

[6] Ruiz, J., Li, Y., and Lank, E. *User-Defined Motion Gestures for Mobile Interaction*. Proceedings of CHI 2011, pp. 197-206.