Scissors – A Precise Pointing Widget for Touch Screen Devices

Felipe Breyer, Luis A. Vasconcelos, Antônio Rivero, and Judith Kelner

Virtual Reality and Multimedia Research Group, Computer Science Center,
Federal University of Pernambuco, Recife, Pernambuco, Brazil
`{fbb3,lalv,ahr,jk}@cin.ufpe.br`

Abstract. A common complaint of touch interaction concerns the lack of precision and false negatives, especially in applications inherited from the PC and mouse context. This work introduces Scissors, a virtual widget for tasks that require precision while interacting with touch screens. It also prevents occlusion of targets, is effective for screen edges and is compatible with current interaction techniques. We developed two prototypes for testing: the first introduced a basic learning scenario and the second presented two sequences of targets with different sizes and locations. The system recorded, for each target, the time spent and number of attempts to select it. We performed tests with thirty-one users and evaluated their data according to statistical test, in this case, t-test for difference of means. The results showed that the Scissors widget was very effective in the scenarios that motivated its conception attaining an equivalent or up to 11.5 times higher success rate, thus achieving its main purpose.

Keywords: pointing widget, interaction technique, touch devices.

1 Introduction

The popularization of touch screen devices is changing the way we interact with digital information. The NPD DisplaySearch estimates that sales of tablets will overcome notebooks in 2014 and it will become the most used computing platform [5]. Accordingly, touch interaction is rapidly outgrowing mice and track pad devices. However, several design challenges follow this change (e.g. wide range of applications with display sizes, resolution, and pixel density), and the future of this technology requires deep investigation in order to design a better interaction and meet different user needs.

Although some patterns were already established [6] for the touchscreens and interactive surfaces, there is still ground for further development; the variety of possible gestures and ways to which systems respond and interpret can open infinite possibilities considering specific scenarios.

This paper is organized as follows; section 3 presents related works and gives deeper perspective around touch interaction techniques in order to stand as basis for the description of the solution in section 4. Evaluation methodology is detailed with test procedures, chosen usability metrics and hardware setups in section 5. Section 6

A. Marcus (Ed.): DUXU 2014, Part II, LNCS 8518, pp. 184–192, 2014.

shows the results obtained by user experiment, with some discussions on section 7 and conclusions and future works presented in section 8.

2 Touch Devices

Touch sensitive technologies have been under development since mid-‘60s, however, they only achieved massive popularity in the 2000s. Touch devices have some important characteristics that should be outlined, such as its size and tracking capabilities. These features will define for which tasks a device is suitable [2].

In order to analyze a touch device, firstly, we should differentiate them between touch-tablets and touch-screens, that is, if the device works only for input, as track-pads, or if there is a tactile-display working as input and output simultaneously, like palm-tops. Hereafter, another important characteristic that directly concerns HCI (Human-Computer Interaction) is the size of the device, which will determine with how many fingers or hands the user may interact with it.

In turn, another feature and one of the most relevant for touch devices is how many points it is capable of tracking, i.e. single-touch or multi-touch. [2] also reminds us that if the device is single-touch, the interaction paradigm will not change when compared with mouse, trackball or joystick.

Particularly, large and multi-point touch devices, such as touch-tables and interactive boards, may require to be operated by multiple users. So, the device can utilize different techniques to differentiate them, either by individual interaction regions or unique ids for each stylus. This brings us to another characteristic, if the interaction will be direct with the fingers or with some extra tool, such as, a stylus or another tangible interface.

Considering the input information captured by the device, some are capable of not only capturing a single point at the touch surface but the whole contact area. This way, we can differentiate the interaction through the touch of a finger and when we hold the device against our cheek. Besides the contact area, some touch devices can also identify the angle of approach and the pressure applied to the surface.

Lastly, on a higher level of abstraction, it is found the possibility of the system interprets gestures, requiring from the application to track the position throughout the time. Different interaction devices have varying combinations of these features, thus, the designer must be aware of them all to adapt the system design and interaction techniques to better suit the hardware in which the application will run.

3 Related Work

A common complaint of touch interaction concerns the lack of precision and false negatives, especially in applications inherited from the PC and mouse context. To date, it is possible to find contextual solutions that magnify the area been touched and require user to confirm his/her intention, commonly seen in browsers. In other solutions, such as a few text readers, users can zoom in, out and pan, repeatedly, thus avoiding false positives but also reducing visualization. However, these approaches either present extra effort from users or cause information occlusion. Occlusion may also occur due to users' fingers or hands, while attempting to click in a button or a hyperlink, since touch screens are both input and output devices.

In [4], the authors consider intuition as an important yet ill-defined factor when designing effective multi-touch interactions. In their research, they perform an extensive literature review on multi-touch interactions. Based on their findings, the researchers constructed a framework of five factors that determine the intuition of multi-touch interactions: direct-manipulation, physics, feedback, previous knowledge, and physical motion. Also, the authors point two major research problems derived from the increasing complexity of multi-touch applications. The first is that multi-touch interaction should not be considered in isolation but as part of a whole, and secondly, evaluations should be brought into a realistic environment.

The research in [1] deals with the integration of planar multi-touch surfaces into the desktop environment in order to give users an additional input channel to enrich the current interaction paradigm. The authors create a prototype called Magic Desk which enables them to try various potential regions and configurations that could become multi-touch enabled. One of the drawbacks resides in the negatives effects associated with the right region which was displaced from the keyboard to leave room for the mouse. This fact reinforces the need for devices or techniques that allows precise interaction, such as mice, but native to the multi-touch paradigm, promoting a real continuous workspace.

The study developed in [3] presents Fluid DTMouse as a solution for mouse support in touch applications. The authors underline some requirements in order to emulate a smooth and natural mouse interaction on a touch-based surface. Firstly, it must be easy to precisely position the mouse cursor, which is particularly challenging because the user's finger occludes the cursor. Also, it must be simple to toggle between mouse-over and mouse-drag modes, and it is undesirable for this toggling mechanism to require movement of the cursor itself. Their system draws the cursor between the first two fingers to touch the screen, and the user must activate the cursor using a third finger. However, since the clickable area is defined as the bounding box created by the two first fingers, this implementation causes the clickable area to disappear when the fingers are aligned vertically or horizontally. Another issue in this technique concerns the difficulty to reach into corners of the surface.

4 Solution

Scissors is a virtual selecting tool for touch screen devices, it uses three reference points: two interdependent fingers for pointing and one for selecting. In order to prevent occlusion, the activation pixel is located at the midpoint of the two pointing fingers. However, to select targets on edges or corners, users can approximate their fingers and, when they reach a threshold of 300 pixels, Scissors bends into an arrowhead shape capable of pointing objects in such difficult to reach areas (Fig. 1A). Users may also flip the direction in which Scissors point, which helps selecting targets located under their hands (Fig. 1B). This widget has no correct hand position for its use: the two orientation points can be defined by any combination of users' fingers. For instance, users with long nails might choose to use index plus middle fingers for pointing (Fig. 1C), while thumb or ring fingers for selecting. In fact, any other touch input will activate the selection, as it is an independent action.

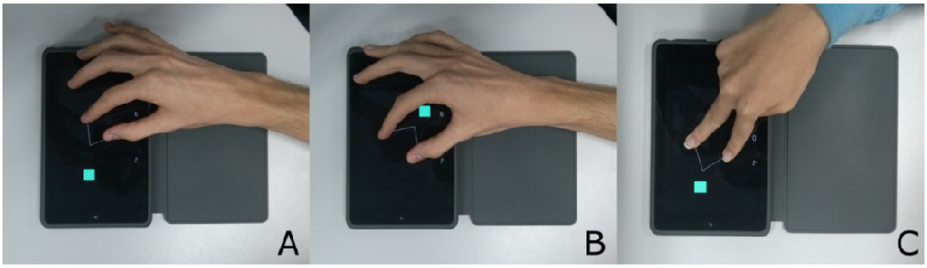

Fig. 1. Scissors can bend in different directions (A, B) and used with any combination of fingers (C)

As recommended by [4], a well-designed multi-touch interaction is directly related to five characteristics: direct-manipulation, physics, physical motion, previous knowledge, and feedback – Scissors was conceived respecting all these five requirements. Since it's a virtual widget, the user manipulates it directly over the system interface. The joint/bend capabilities simulate a physics behavior and motion, while the scissors lookalike metaphor evokes the feeling of an everyday object, favoring the user to recall previous experiences. The constant feedback given by Scissors during its different states keeps the user aware of its condition and interaction possibilities.

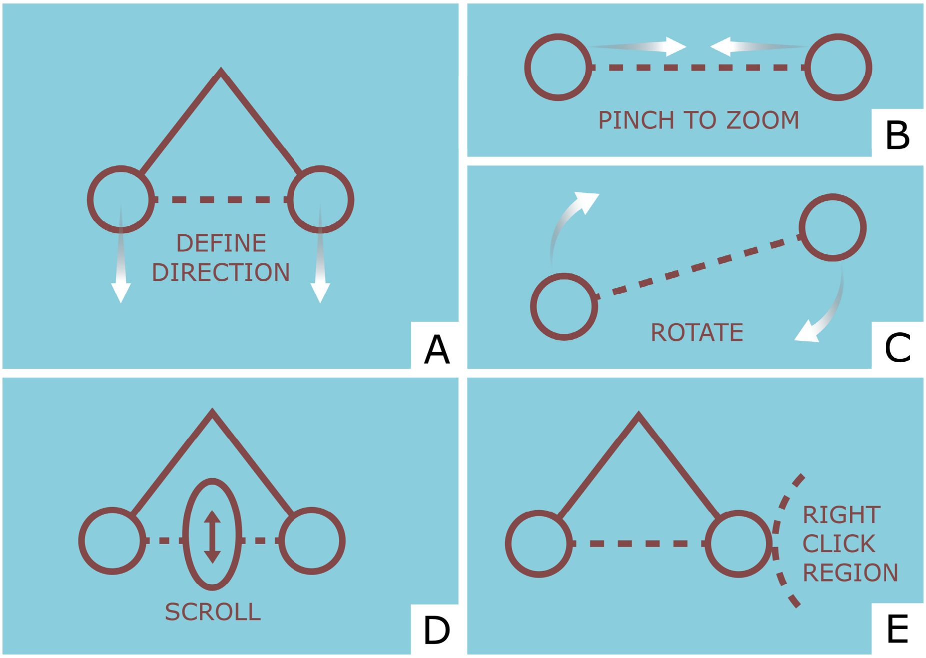

Fig. 2. Scissors possible interactions

Scissors was designed not to conflict with other common touch interaction techniques such as pinch to zoom and rotate. Knowing that pinch movement is done over the axis (Fig. 2B) defined by first two reference points, it will not activate selection. Since the direction that Scissors bends is defined by a movement to the same side of the reference axis (Fig. 2A), it does not affect rotation which is done by moving the fingers in opposite directions, as shown in Fig. 2C. Another important function integrated with Scissors is scrolling, positioned at Scissors' reference axis middle point (Fig. 2D); users should use index finger if using thumb/middle as references or middle finger in case of index/ring as references. Finally, right click functionality was adapted by defining a specific region at right from Scissors (Fig. 2E). Therefore, we were able to merge all function established for mouse and touch devices.

5 Evaluation Methodology

We developed two functional prototypes to evaluate the concept. Both prototypes were developed with libGDX, a game and multimedia development framework based on Java that can export the application to different platforms such as Android, iOS and HTML5. It facilitates the manipulation and tracking touches as well as rendering tasks for the application. The prototypes were installed on a first generation Google Nexus 7 tablet equipped with a 7-inch screen, which has 1280x800 resolution, resulting into a screen density of 216 ppi.

The first prototype worked as a learning scenario setting users free to try Scissors so they could get used to this new interaction tool, as we considered unfair to compare a new interaction technique with other to which users were already familiar. This prototype worked in an infinite loop, creating hundred pixels square buttons so the user could test Scissors to select them at random position in the screen. The users were encouraged to test this prototype as long as they wanted until they felt comfortable and confident with it.

The second prototype presented two sequences of eighteen square targets of six different sizes divided in three categories: large targets with 200 and 100 pixels, medium targets with 50 and 25 pixels, and the smaller ones with 10 and 5 pixels. These targets were located in three distinct screen areas: center (ct), edges (ed), and corners (cn). The targets would reduce to next sizes after each three different regions were correctly selected.

In the first sequence, we asked users to aim and select targets the way they traditionally do in touch screens, just touching the targets with a finger. In the second sequence, users should use Scissors for the same task.

As usability evaluation metrics, the time to complete a task was used as measure of efficiency, i.e. the time (Tm) to touch or select each target. For measuring effectiveness, the number of attempts (At) to touch or select the target by the user was registered. The system recorded the results for each participant in a log file for further data analysis.

Two hypotheses were defined for the experiment:

- **H1:** Scissors will be more precise than direct touch interaction on small targets.
- **H2:** Precision provided by Scissors will compensate the time spent with its manipulation.

We performed the tests with thirty-one regular users of touch devices. 16 of them were undergraduate students while 15 were postgraduate. All subjects were students at the Informatics Centre from Federal University of Pernambuco, Brazil. The gathered data were analyzed according to the t-test for difference of means, in order to evaluate the variation between the results, as presented in next section.

6 Results

The results showed that the Scissors widget was very effective, especially in the scenarios that motivated its conception. The experiment's results are deeply discussed below.

As can be seen through the task execution time chart (Fig. 3), subjects using Scissors presented higher task duration when compared to traditional touch interaction for big and medium sizes in all conditions. Scissors' worst result happened in the event of 50 pixels target located on the screen corner, obtaining an execution time approximately six times slower than touch interaction (Tm50cnTouch = 0,8424s) (Tm50cnScissors = 4,8936s). On the other hand, when analyzing the data from smaller targets, 10 and 5 pixels large, it is possible to observe a change in this behavior. For 10 pixels target at the screen centre, time was considered statistically equal for both samples (dTm10ct = 0,0922 > 0,05), as well as for the 5 pixels target at screen centre and corner (dTm5ct = 0,1520 > 0,05) (dTm5cn = 0,8358 > 0,05). However, it is important to note that the result for task execution time at the screen edge with a 5 pixel target with Scissors was 50% less than with touch interaction (Tm5edScissors = 10,1243s; Tm5edTouch = 18,8095s).

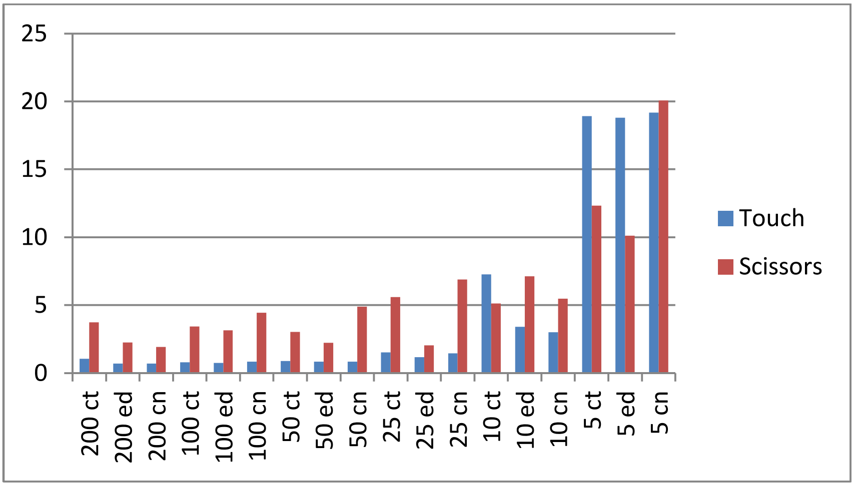

Fig. 3. Mean time for diverse target sizes at different locations of the screen

On the number of attempts, most of the results for the larger and medium targets were considered statistically equal between using Scissors or traditional touch interaction (Fig. 04). The only exceptions occurred in the cases of screen centre and edge for 100 pixels target which were considered statistically different (dAt100ct = 0,0487 < 0,05) (dAt100ed = 0,0100 < 0,05), even with numerical values close to a single attempt. Still, for the screen edge 25 pixels target, Scissors obtained a favorable result in comparison with touch interaction (dAt25ed = 0,0446 < 0,05). With respect to 10 and 5 pixels targets, Scissors shown better results in all three evaluated regions, with highlights to the case of 5 pixels target on screen edge, in which Scissors has a value eleven times better than touch interaction (At5edTouch = 38,9354) (At5edScissors = 3,3548).

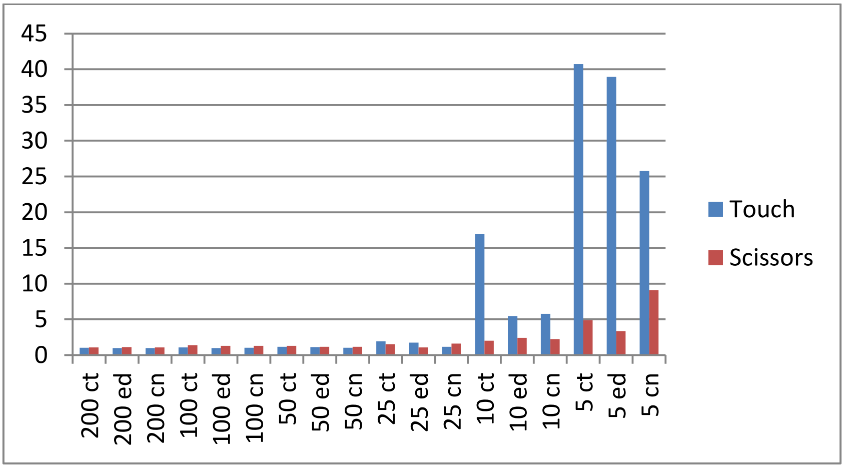

Fig. 4. User attempts for selecting diverse target sizes at different screen locations

7 Discussions

Based on the data analysis presented above, we consider the first hypothesis for this research (H1) as confirmed, due to the fact that the number of user attempts for target selection with Scissors was considerably lower, confirming its efficiency for small targets scenarios. The second hypothesis (H2) could not be confirmed because Scissors presents a better execution time only for screen edge 5 pixel target, while in other three cases were considered statistically equal (10ct, 5ct, 5cn), and for the two scenarios left (10ed, 10cn), the results for traditional touch interaction were better. These results require a deeper investigation and showed that more tests are needed.

Moreover, some spontaneous comments from participants were recorded after the tests. In general, the volunteers considered Scissors easy to use, claiming that its purpose was clear from the moment they realised the needed to interact with greater precision with the touch screen. Frequently, the researchers were questioned about its official release or where users could download it even knowing it still under development. Particularly, women who participated in the experiment shown significant interest in the project, due to the possibility of using any two fingers to manipulate Scissors even with large nails, which fits their specific needs.

One last issue we could perceive with the experiment is related to subjects' behavior towards their failure. Due to the frustration of successively missing smaller targets after several attempts, a portion of the users started using the strategy of tapping the screen repeatedly, without caring for the consecutive mistakes. In the scenario presented by the prototype, there were no other interactive objects surrounding the target so that the error did not activate another function. This way, without this risk of false positives, it was more likely that users adopted this tapping strategy. On the gathered data, this strategy caused a reduction in the task execution time but raised the number of attempts for small targets. Regardless the participant's behavior, traditional touch interaction technique has a disadvantage in one of the metrics, reaffirming its inadequacy in cases where precision is an important factor.

8 Conclusions and Future Work

In this study, we presented that multi-touch interaction can still benefit from applied researches. We believe there are still plenty of possible developments in this fast growing technology and it has not reached its full potential. Many other multi-touch applications are still under development for different scenarios with many specific requirements and only by focusing on their special needs we may achieve real innovation. In such manner, the results presented in this paper encourage the continuity and development of this research. As the 'small target scenario' can be found in many multi-touch applications and since Scissors is compatible with other interaction patterns, we believe this project should coexist with the current interaction paradigm, giving the user a wider range of choices. Accordingly, this research fulfilled its objective as it presents an interaction technique capable of precise selection as shown by the tests results.

For future work, we intend to perform experiments in smaller devices in order to stress the use of three-finger interaction and observe the user behavior to greater occlusion situations. It is also important to investigate different thresholds from the 300 pixels used in the initial design and determine its impact on the user experience; such as 200 pixels, for example. The combination of factors, screen size, and the widgets size should also be examined. Finally, since the initial results were promising and we considered a controlled environment an adequate choice for a first experiment, a real test scenario should be put to run.

References

1. Bi, X., Grossman, T., Matejka, J.: Fitzmaurice. G.: Magic Desk: Bringing Multi-Touch Surfaces into Desktop Work. In: CHI 2011, Vancouver, BC, Canada, May 7-12 (2011)
2. Buxton, B.: Multi-Touch Systems that I Have Known and Loved (2013), `http://www.billbuxton.com/multitouchOverview.html`

3. Esenther, A., Ryall, K.: Fluid DTMouse: Better Mouse Support for Touch-Based Interaction. In: AVI 2006, Venezia, Italy (2006)
4. Ingram, A., Wang, X., Ribarsky, W.: Towards the Establishment of a Framework for Intuitive Multi-touch Interaction Design. In: AVI 2012, Capri Island, Italy, May 21-25 (2012)
5. NPD DisplaySearch: Tablet PC Market Forecast to Surpass Notebooks in 2013. In: Quarterly Mobile PC Shipment and Forecast Report (2013), `http://www.displaysearch.com/cps/rde/xchg/displaysearch/hs.xsl/130107_tablet_pc_market_forecast_to_surpass_notebooks_in_2013.asp`
6. Saffer, D.: Designing Gestural Interfaces. O'Reilly Media, Sebastopol (2008)