H. Wandke, S. Kain & D. Struve (Hrsg.): Mensch & Computer 2009: Grenzenlos frei!?
München: Oldenbourg Verlag, 2009, S. 223-231

Overcoming the Impreciseness of Touch on interactive large-screens

Thomas Herrmann, Tillmann Neben, Marc Turnwald

Information- and Technology Management, University of Bochum

Abstract

This paper analyzes the possible reasons for imprecise pointing in large-screen touch environments and derives a categorization scheme for the problems. On this basis a solution to the imprecise selecting problem is presented. For our setting in which we use a 5x2 meter large, seamlessly interactive touch screen, a combination of two separate concepts, SmartPointing and ClickAgain has proven to be most suitable and to offer the most predictable user experience. The proposed solutions also meet the special requirement of being feasible in both a mouse-controlled as well as a touch-enabled setting.

1 Introduction

The need for a new interaction design emerged in the late 2007 when we tested the usage of a graphical Editor ‘SeeMe2000’ with a large touch screen. The interactive screen consists of three seamlessly connected 80“ rear-projection units from Barco Inc. resulting in a virtual display size of 4,8m x 1,2m (w/h) and an overall resolution of 4320 x 1050 pixels. Smart Technologies has installed a touch detection system of three units each consisting of two infrared cameras per screen, with overlapping viewing angles at each connection between two screens for the logical handover of touch events between the physically separated units. The detection area is calibrated such way, that any object that comes in an approximately 2cm range to the walls surface causes a detection event. The software we mostly use is a graphical editor, which we have developed to support socio-technical process design with the modeling notation SeeMe (Herrmann 2009). The interactive large screen was installed to display, discuss and modify large process models in workshops with approx. 8-12 participants. During these workshops, people are expected to switch flexibly between working with their desktop and interacting with the large screen, both running the SeeMe editor application. Such a setting is presented in fig.1.

Testing our large interactive screen running for the described purpose we could distinct three problem categories:

1. Imprecise selecting and pointing capabilities, e.g. if another than the intended object is selected or if a user does not succeed in hitting the handle of a graphical object to resize it.
2. Inaccurate distinction between steady and moving user input, e.g. between simple selecting a rectangle or moving its position.
3. Lack of suitable tooling concepts instead of the old-fashioned toolbar, e.g. if dialogue boxes appear 2m away from the current spot of interaction.

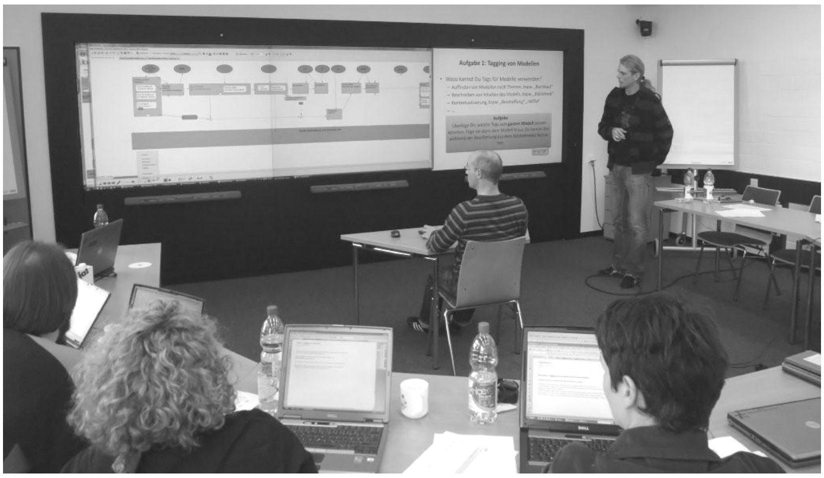


Figure 1: Modlab at Bochum

The problem stated in (2) is very common. This might be due to technical limitations or wrong (jittery) user behavior. The lack of a concept on how to offer tools and options to the user is another problem (3). The traditional centralized toolbar fails since with larger screens the distance between user and wherever the toolbar is located might be impractical long. Even with smaller devices it has to be acknowledged that a certain distance covered with a mouse by a wink, could mean exhausting movements of the whole forearm. In this paper we focus on (1), the reasons of and solutions to imprecise selecting and pointing. However the boundaries between (1) and (2) are sometimes blurred, so that some issues from (2) are also covered, if relevant. We intend to support specific settings of workshops where working with desktop computers and with the large screen has to be smoothly intertwined (see fig. 1). Users will work with the desktop version of the SeeMe editor alone or in small groups, interrupted by presentation and joined modeling phases using the interactive large screen. The possible ways of interacting with the SeeMe-Editor –traditional desktop and touch screen – have to be as similar as possible to meet the users' expectation of consistency. Using the graphical editor with a mouse-controlled desktop computer is the more common scenario, but should also prepare the user to be able to interact with the editor on the large wall.

2 The Problem of Imprecise Selecting and Pointing

All touch-enabled devices suffer of some sort of in-accuracy. Small devices like Apples iPhone struggle because the human's finger is often larger than the target objects on the screen. We find however a more structured approach is needed and a distinction between two kinds of settings in which in-accuracy happens must be made. The first is the stationary setting, in which the user uses the device in a traditional person-device arrangement, e.g. a user holding a touch-enabled PDA or standing in front of a touch-enabled ticket-machine at the railway station. Second there is the in-stationary setting, in which the user acts in front of a larger interactive device or a set of devices, deployed along the room and which leads to imprecise selection due to a suboptimal body positioning, selecting during or at the end of a movement, or the problem of reaching a certain element which is not positioned near to the body.

We see the following categories of failure origins. Some of which are common to all touch-enabled devices, some are inherently given in any in-stationary setting where the screen dimension exceeds the physical arm length of the user.

1. Technical issues
   a. E.g. we have inappropriate sensor capabilities for a given purpose or the calibration is set off.
   b. Due to many overlaying objects or dense neighborhood of objects on the display area, even pointing with a sharply tipped pointing device instead of a finger, like e.g. a pencil, could not avoid the impreciseness.
2. User misplaces a touch
   a. E.g. because one is in or at the end of a movement, has only a poor angle of view, or has to stretch the arm so that impreciseness is enhanced.
   b. Handicapped user, the elderly, disabilities (e.g. because of defective sight, impaired motor skills etc.)
3. Inappropriate interaction design for touch-enabled devices

   We could also observe problems, which are neither originated in technical issues nor by human inabilities. These problems are caused by an inappropriate interaction- design for the given situation.

Real large screens come with a small but also unavoidable displacement of the finger position on the surface due to the technical limitations of the cameras, scanning the area (1a). A second origin of imprecise selecting by the user lies in the size of the large screen (2a). A several meter long large screen forces the user to walk along the UI and stretch herself to reach a certain area. This impedes the accuracy of the user input. In both cases the user selects an unintended spot on the screen. We assume that the most common problems are related to 1a), 1b) and 2a), whereby the majority of failed selections fall into the category 1b). These constellations of layouts (1b) are very common in diagrammatic representations. Similarly important is the category 2a) of problems where the approached element is simply too small or in a too obscured position, to hit it correctly with the finger or a pen (1.a). E.g. the element is above the user's eye line, so he has to look and stretch upwards. The failed selection due to a bad angle of view or touch (2a) is something very specific in our setting and other large screen systems. Many other problems can be observed when users try to use common desktop applications with mouse-based interaction design in the touch-enabled environment of the large screen (3). These problems are strongly related to the technical problems (1a, 1b) and also depend on the users' behavior, expectations and abilities (2a, 2b) and therefore pose the opportunity to overcome these problems with a new interaction design.

3 Related Approaches

The early work in the field of touchable interface design of Potter et al. (1988) describes three basic strategies, to handle the problem of selecting items by using the systems internal representation or data to improve selection accuracy. One type of suitable touch screen interaction is called "Take Off". It leads to a lower mean error-rate than the two other methods "Land On" and "First Contact", as described by the authors. The "Take Off" method is based on fixed offset between cursor and the user's finger touching the screen. Selection is then made upon release if the cursor is above the item. In spite of the good results with "Take Off" , it cannot be applied to our current problem, because it is based on the "drag-and-release" paradigm and therefore collides with the basic requirement of the pursued interaction design, namely to keep things as similar as they are within the world of mouse-controlled interaction. Therefore many newer work results that are based on "Take Off" (Albinsson et al. 2003) are also not feasible within our constraints. Similarly, all types of solutions and ideas for precise selection which are based on Multi-Touch (Benko et al. 2006) are not feasible, since we intend to employ multi-touch for simultaneous interaction of multiple users who then use the large screen in the point-and-click-style.

Other techniques like "Semantic Pointing", (Blanch et al. 2004) are designed to improve target acquisition by adapting the movement ratio between an input device and a displayed virtual pointer. This causes a change of the pointers speed when it to comes in range or is above a possible selection target. These techniques are basically intended for mouse interaction support and could therefore not be applied in our case, because applying this method to a touch screen would cause a variably changing offset of the cursor in relation to the hand, leading to situations where the user's hand or arm unpredictably covers the cursor. Another shortcoming is that the touchscreen cannot distinct between intentions such as pointing vs. clicking on an item with a touch.

Although Potter et al. (1988) found early that "Land On" based selection produces the worst results compared to the other two basics interactions, it seems to be the most suitable method to meet our requirements. "Land on" is the most intuitively interaction compared to a natural pointing gesture, because it uses the metaphor of landing on a selection target which is selected when the user's fingertip lands on.

Special to our current setting is that items can be partly or completely overlapped by other items, and often have no clear visual boundaries, because the SeeMe-Editor supports the repeated nesting of items into other items. For example, the boundaries of the text box, which has to be selected in fig. 2, are not visualized if the text is not selected (fig 2c). End-points of an arrow are another example of items, which are difficult to distinguish from other items. As shown in fig. 3, they are always connected with larger items and have no visual displayed boundaries in the unselected state. In such a setting, pure “Land On” would surely often result in the selection of the embedding item if the boundaries of the intended item are not at the expected location. The “First Contact”-strategy could not lead to better results in the case of embedded items, because it may happen that the parent-item is selected instead of the nested element.

4 A Two-Way Approach to Improved Pointing

Our proposed solution has emerged out of two design cycles, each of which has added certain functionalities. Also the existing desktop-based SeeMe client and its interaction paradigm has strongly influenced the design cycles.

4.1 First Design Cycle (Smart Pointing)

When applying the existing interaction process from the SeeMe desktop-client to the touchable interface of the large screen, two major problems have been observed. As mentioned earlier, a touch on the screen has to be directly assigned to Mouse Button-Pressed or -Clicked Events, to lead to any reasonable effect. Therefore the user has no possibility to place the Mouse Cursor over the intended item (hover) before a touch is applied. The formerly split processes of first pointing at an item and then clicking or pressing the Mouse Button to properly select it, is merged into one step. The success of this ‘selection by touch’ is firstly related to the user’s hand-eye coordination capability (2a) and secondly to the lateral touch detection resolution of the wall (1a).

Regarding the goal of hitting a graphical item with a single touch, it seems clear, that the intended item must lie in a range of pixels around the detected touch point. This range depends on the minimal discrete step length between two points that the system can detect as different locations. As a first approximation this distance was set to 15 pixels.

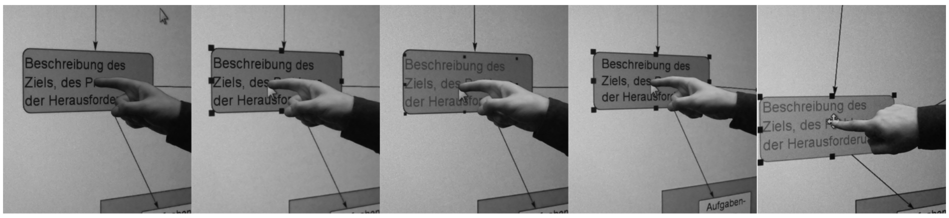
As a first step, all items that lie completely or partly in this range are considered as *intended* and are stored in a list of candidates. The question, which of these items was actually meant by the user, is then answered by bringing the list’s items into the order of decreased intention probability, so that the most intended item is the first item in the list. As a heuristic approach we decided that the order should depend on the size of the items, the distance to the center of the selection range (the point where the touch was detected), the common order in that items can be embedded into each other and the type of the element. This is based on the idea that smaller items are usually more difficult to hit than larger items and that items which are embedded, as child items in a larger parent item must be smaller than the parent element.

Also, small elements have a smaller hit sensitive area than large ones. A user touching close to but not exactly on a small element that is surrounded by a large one (e.g. by mistake, see problems 2.a, 2.b), conveys a lot about his intention to the system, even if the touch was misplaced. It is likely that the user has actually meant to touch the small element and not the large one in behind, since very close to the small elements border is just not the natural touching area of choice if one intends to hit the large element. The large element offers a lot of open and easily accessible hit area; no one would choose exactly this dense spot for selecting it.

We named this technique Smart Pointing, carrying the notion of preprocessing the users input before actually processing the event. As mentioned above, the actual smartness of Smart Pointing depends on the underlying heuristics, which in our setting are customized to fit the SeeMe editor application and its various shapes and objects. Even if the heuristics ought to be customized to each different application, the basic idea of the concept is generic.

4.2 Second design cycle (Click Again)

The Smart Pointing approach was then tested using regular mouse interaction on a desktop computer and manually optimized with respect to the preferred order of item types, by asking the users about their actual expectations when performing the click. Although this form of Smart Pointing leads to a better rate of hitting the intended items on the touch screen, the large number of possible item constellations could still lead to a selection of the wrong item, mainly caused by problems of category 1b). In this case a simple user interaction is needed to change the proposed selection to the one that was intended. In our setting we found that the easiest paradigm would be, if the user could just "click again" somewhere inside the detection radius, to select the next item from the candidates list.



a) Finger tip is getting close to the surface

b) 1st touch is recognized → Rectangle is selected

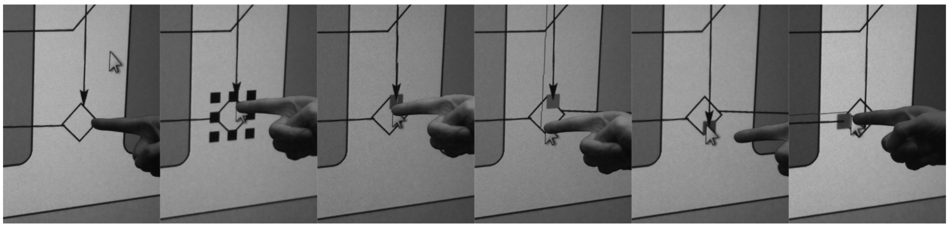
c) 2nd touch is recognized at the same spot → Textbox is selected

d) 3rd touch is recognized → the rectangle is selected …

e) … the touch has not been released and the objected has been dragged

Figure 2: Selection with repeated touches on overlapping objects

Therefore the location of the first point that was touched has to be stored. In the case that a point is already stored when a touch event fires, it has to be calculated whether the actual touch point is in the click again range of 15 pixels to the stored point. Is this the case, the actual item in the list is deselected, and its successor gets selected. If not or if no point has been previously stored, the currently touched point is stored and a new list of items is generated by their Smart Pointing order as described above.



a) finger tip is getting close to the surface

b) 1st touch: the diamond is selected

c) 2nd touch: the tip of the arrow at the diamond, is selected

d) The 2nd touch is not released to drag the selected point

e) The touch is released

f) 3rd touch selects starting point of an arrow at the diamond

Figure 3: Selection of closely located targets with repeated touches

The implemented interaction strategy determines that the first click selects the first item of the list, the second click within the same range selects the second element and so on as illustrated in fig. 2 and 3. In fig. 2, the list consists of the yellow rectangle and the text box while in fig. 3 the list consists of the diamond at the first position and the endpoints of the connected arrows at the following positions.

Finally, we found during our tests of this Smart Pointing with Click Again Extension, on both the touchable Screen and with the Desktop/Mouse system, that it works on each system with respect to the users' expectations.

5 Conclusion

We have demonstrated a technique for making precise selections, feasible both for touch-enabled and also for plain desktop environments. For this we have divided the task of selecting into two separate steps: Smart Pointing and ClickAgain. The combination of both has lead us to a more accurate selecting experience. The combination of the Smart Pointing mechanism with a Click Again Extension to switch to originally intended items is a holistic interaction design for the selection of graphical items for both mouse-controlled and Touch Screen oriented interaction.

While Smart Pointing tries to identify the most probably intended item, the Click Again mechanism proposes an easy method for switching to the intended item in cases of unintended selection

For future research, it is reasonable to survey our approach under laboratory settings, that is, observing users' behavior and their overall perceived usability experience in controlled experiments. As demonstrated in Hurst et al. (2008) major attention has to be paid to the factor of Slipping and, in case of ClickAgain, the mean factor of attempts (re-clicks) till the user's intention is satisfied under various possible layout settings of surrounding distracting elements.

The experimental design of further investigations can be related to the mouse-based interaction with desktop computers as a bench-mark constellation. The hypotheses, which have to be tested, will refer to more complex task as they are underlying fig. 2 and 3. The basic assumption of the hypotheses will be that our interaction design for touch screen will get closer to the benchmark than others with respect to effectiveness, efficiency and user satisfaction. Within these kinds of experiments we can also decide which constellation of our parameters (range of points of the click-again area, heuristics for ordering) are more successful.

References

Albinsson, P. & Zhai, S. (2003). High precision touch screen interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, New York, NY, USA, pp. 105-112.

Benko, H., Wilson, A. & Baudisch, P. (2006). Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 1263-1272.

Blanch, R., Guiard, Y. & Beaudouin-Lafon, M. (2004). Semantic pointing: improving target acquisition with control-display ratio adaptation. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, pp. 519-526.

Herrmann, T. (2009). Systems Design with the Socio-Technical Walkthrough. In Aldo Whitworth, Brian; de Moore, ed. *Handbook of Research on Socio-Technical Design and Social Networking Systems*. Idea Group Publishing, Hershey, pp. 336-351.

Herrmann, T., Kunau, G., Loser, K. & Menold, N. (2004). Socio-technical walkthrough: designing technology along work processes. In *PDC 04: Proceedings of the eighth conference on Participatory design*, ACM, 132-141.

Hurst T., Mankoff J. & Hudson S.E. (2008). Understanding pointing problems in real world computing environments. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pp. 43-50.

Potter, R. L., Weldon, L. J. & Shneiderman, B. (1988). Improving the accuracy of touch screens: an experimental evaluation of three strategies. *In CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, pp. 27-32.

Contact

Prof.Dr.-Ing. Thomas Herrmann

Institute of Applied Work Science
Department Information and
Technology Management
44780 Bochum, Germany
Tel.: +49 (0) 234-32-277720
Fax: +49 (0) 234-32-14207
E-Mail: thomas.herrmann@iaw.rub.de

B.Sc. Tillmann Neben

Institute of Applied Work Science
Department Information and
Technology Management
44780 Bochum, Germany
Tel.: +49 (0) 234-32-22045
Fax: +49 (0) 234-32-14207
E-Mail: tillmann.neben@iaw.rub.de

Dipl.-Inf. Marc Turnwald

Institute of Applied Work Science
Department Information and
Technology Management
44780 Bochum, Germany
Tel.: +49 (0) 234-32-22045
Fax: +49 (0) 234-32-14207
E-Mail: marc.turnwald@iaw.rub.de