TypeNinjia - Typing and Control Interface for Public Displays

XinXin HE
Department of Human-Centered
Computing
SOIC, IUPUI
USA
xinxhe@iupui.edu

Wei Pan
Department of Human-Centered
Computing
SOIC, IUPUI
USA
panwei@iupui.edu

Aeshvarya Verma
Department of Human-Centered
Computing
SOIC, IUPUI
USA
aeverma@iupui.edu

Abstract

We proposed TypeNinjia, a mid-air typing method with Leap motion for large displays. We firstly surveyed various keyboard layouts and mid-air gestures for wall-sized screens, finding out the possible input modalities for our project. Then, we evaluated the benefits and problems of these modalities based on which we designed our optimal keyboard layout. To better demonstrate our design, we come up with a potential use of the keyboard on large displays - a restaurant aid system in the airport. With that system, users can search and browse, find routes, and leave comments of the restaurants in the airport, and all of these manipulations are done with Leap motion. Our main focus is to implement the keyboard for leaving comments, thus we can evaluate the actual effect of our design on large display later.

Author Keywords

Leap Motion; Text Input; Large Display; Virtual Keyboard

ACM Classification Keywords

H.5.2. [User Interfaces]: Input devices and strategies, Prototyping.

General Terms

Design, Experimentation.

Introduction

With the development of ubiquitous computing, devices like big displays, Google glass or Kinect are starting to weave into our daily life, shifting us from the traditional keyboard-and-screen interactive mode to a more natural and humanized one. One hot topic of this area is to develop new interactive techniques for large displays, we are quite interested in that and decide to explore optimal typing techniques for large displays with Leap motion.

Related Works

A lot of researches were done in terms of gesture-controlled user interfaces. Researchers have identified several different areas in which gesture technology will have a big opportunity: entertainment, controlling home appliance, training, education, elderly care [1].

Obviously we could see the potential of gestural interaction, and a lot of research has been done to understand it [3][4][5]. For example, Mathieu Nancel identified three key factors for designing gestured based pan-and-zoom techniques on wall-sized display [3]; Radu-Daniel Vatavu conducted a research to find users' preferences for gesture commands for basic TV control tasks [4]. In a context where very large display is considered, it is more natural to use gesture to manipulate objects on the screen, such as pan-and zoom [3]. However, traditional text input modalities, namely keyboards, are often inappropriate for use when standing in front of a very large wall display [7]. Shoemaker has suggested some techniques for text input in a wall-size display scenario by using a Wiimote from a Nintendo Wii video game system as the input device [7]. However, depending on the representation of the keyboards, such point-and-select input method suffers from either distance-dependent or visibility-dependent problems [7]. They concluded that distance-independent techniques might be best for use with very large wall displays. With that conclusion, leap motion has the potential to be used as a mid-air text input device since it would become distance-independent if it becomes wireless and it could be visibility-independent by simply using carefully-designed gesture to enlarge or shrink the keyboards depending of the distance between user and the display.

While it seems compelling to use Leap Motion as an input with large-sized display, there are certain things that need considerations:

1. Input techniques:

Hand drawing: With the promising precise and pointing-object recognition that Leap Motion support, it may be natural to use handwriting for text input. However, as what Shoemarker [7] has pointed out, Handwriting is not always the best choice for text input. Writing input speed is known to be at best around 20 wpm [10]. Furthermore, recognition algorithms for the purpose of digitizing and archiving written data are error-prone [7].

Stroke-based: Stroke-based writing: "The user strokes through all the letters in the word on a virtual keyboard without lifting his or her finger". "For stroke-based entry, the lack of tactile-feedback is less of a problem and there is only the need to use one finger, which suits both technologies that would register any touch and contexts where the user is not in a comfortable position to use all ten fingers". Stroke-based text entry techniques contain ambiguity, which requires a useful algorithm for mapping the stroke sequence to a word and must provide a usable interface for resolving ambiguity. One examples of this kind of input technique is Cirrin[13], a word-level unistroke keyboard for pen input.

Tap-based: Tap based typing generally use the same modality as when using a physical keyboard, which requires ten-finger input, and a press or tap on keys to type in the character. It shares criteria as stroke-based text entry: The longer the distances between letters, the longer it takes to reach them with either tapping or stroking [14].

Exclude the hand drawing, the expected performance of rest two input techniques (stroking, tapping) are impacted by the keyboard layout.

2. Layout of the keyboard:

A lot of researches have been done in exploring which layout will be more effective. Basically there are three main types of keyboard layouts: QWERTY keyboard [8][9], Alternative keyboard layouts, and "quasi-QWERTY "[11] [4] layouts which are somewhere between QWERTY and the alternatives.

QWERTY keyboard: QWERTY technique is suitable for adoption. this approach carries the significant benefit of being instantly familiar to users. Indeed, the latter property has been a driving factor in the consumer space, and most text produced today on smart devices is by miniature QWERTY keyboards.

Alternative keyboard layouts: "A second approach is to leave the QWERTY keyboard design behind. Despite user resistance, this approach is popular because performance gains can be significant." Mankoff has suggested a word-level unistroke keyboard called Cirrin for pen input.

"Quasi-QWERTY " and "QWERTY -like" layouts, allowing keys to be relocated a limited distance from their traditional positions. This facilitates easier visual search for users familiar with Qwerty.

Rick [9] has conducted a study to investigate influence of keyboard layout on expert text-entry performance for stroke-based text entry. And he identified and generated two optimized virtual keyboards for stroke-based text entry on touch-based tabletops. With the module he introduced, he predicts that a switch from tapping keys on Qwerty to stroking them will bring a speed improvement of 17.%. He also suggested that for new technologies, such as handheld devices and interactive tabletops, there is an opportunity to introduce a new layout and a new text-entry technique as long as it improves performance.

Therefore, relating to our context - wall-sized display in airport, we are going to take leap motion as the input device, since it has the potential to be distance-independent and visibility-independent, therefore allowing users to type and manipulate from distance while seeing the whole screen. And QWERTY-like layouts will a better choice since we could modify the keyboard so that frequent function keys could be easier to be tapped while still familiar to passengers.

Design and Rational

We are pursuing a scenario of an Airport wherein people have difficulty finding a good restaurant keeping in mind the limited time they have. In order to utilize the power of Leap Motion controller we have formed two tasks that people would perform at the Airport.

Finding a restaurant according to their choice and leaving comments about the food they had eaten in those restaurants. Both of these tasks are to be done on a large display present at various locations throughout the airport. We selected this scenario because of the following reasons:

I. A lot of people would come interact with the proposed interface on a large display

II. People have comparatively very less amount of spare time at the airport to gather information and act on it

III. They usually carry luggage with them, which restrict their hand movement to locate information on any kiosk

IV. There are people from all walks of life providing a wider audience for our interface design

After having read all the research papers in the literature review we decided on a QWERTY keyboard for the gesture based typing on the large display. Our argument in favor of the QWERTY keyboard lies in the fact that most of the savvy travelers are familiar with that keyboard layout whether it's on their laptop keyboard or their touchscreen mobile device. We, however, made certain subtle changes in order to utilize the full large screen of the display. From the literature review, we found that researchers have already experimented with various keyboard layouts with varying success seen in the testing results. In some of them the keys where placed in a circular shape, in one of the designs the layout contains hexagonal keys clubbed together forming a cluster.

In our unique keyboard layout we focused on the following requirements:

I. We focused on the fitt's law i.e. maintaining an optimal interactive distance and the visibility of the actions

II. A very low learning curve in using the keyboard for typing

III. Provision of gestures along with specific buttons corresponding to single unique operation IV. Unistroke gesture for typing and a visual feedback for the users

V. Utilizing the overall space available for the keyboard on a large screen

In our design, we got rid of the special characters keys and specific the number keys. Instead we provide separate buttons for caps, delete and numbers apart from individual gestures for each one of them. We took out the space bar from the layout and expanded it to match the whole horizontal row at the bottom. Similarly, the enter button is expanded at the right of the layout. This way, we provide retention of the actions of users as well as utilizing the large screen space. We provide easy to use gestures, namely,

I. Left swipe for delete

II. Right swipe for space

III. Down swipe for punctuations

IV. Up swipe for word selection

Our design is a gradual shift from the touchscreen-based keyboards to a gesture-based keyboard making use of the Leap Motion Controller.

Scenario

Elsa will take a plane from Indy to New York at 10:00 pm. At 6:00 pm, she arrives at the Indianapolis airport. Since there are still 4 hours left, she decides to find a restaurant and have dinner firstly in the airport. When entering the hall, she sees a large display showing an airport map and a list of restaurants with rating stars. Elsa goes close to the display and figures out that she can search for restaurants with gestures. Following the instructions, she chooses the terminal where she is firstly. Then she randomly clicks a restaurant on the right list and a window showing detailed information and comments about this restaurant pops up. Elsa glances at the price, location and comments a little while, and then she switches to another restaurant. After browsing several restaurants, she decides to go to a Starbucks with four stars for a cup of coffee and some snacks. She clicks the map on the top right corner for direction, gets to know the direction from her place to the Starbucks, and then heads for it. It turns out that this Starbucks is really a nice place with good coffee and delicious sandwiches, so Elsa decides to leave some positive feedback for it. When she returns to the hall, she goes to the large display, finds the Starbucks she was just at, and clicks "leave a comment" button. A simple keyboard soon pops up below the comments box, Elsa uses her right finger typing a short comment, slides five stars, clicks the submit button and leaves the comments successfully.

Prototype

Structure

The prototype includes an interactive map, restaurant search and result list as shown in Figure 1, restaurant page for detailed information and reviews as shown in Figure 2, virtual keyboard for typing as shown in Figure 3. All the manipulations are done through various mid-air gestures using a leap motion.

Fig 1: Homepage

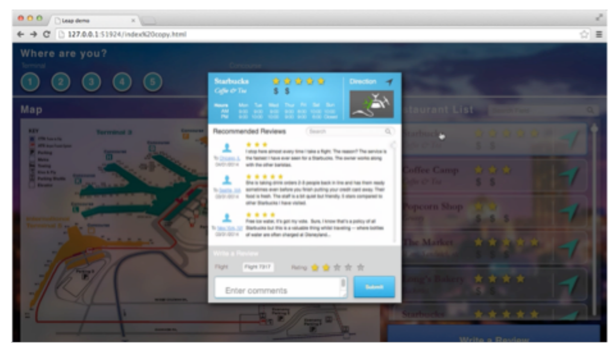

Fig 2: Restaurant page for detailed information and comments

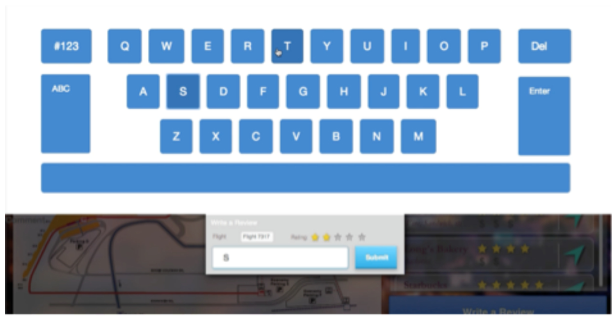

Fig 3: Keyboard layout

```
<div class="panel">
    <div class="thumbnail leap-thumbnail leap-interactive" leap-disable-tap="false">
        <h4>Restaurant #2</h4>
    </div>
</div>
```

```
<div class="number">
  <button type="button" class="btn btn-primary btn-lg leap-interactive">#123</button>
</div>
```

Fig 4: Code for list panel and keyboard buttons

Implementation

We developed the system using LeapStrap - a HTML5 based front-end framework. Leapstrap is a complete toolkit for building a Leap web site or web application. It requires 1 CSS file, 2 plugin files (jQuery and LeapJS) and the Leapstrap JS file, Leapstrap detects all anchor (e.g. hyperlinks) and makes them leap-interactive.

Improvements in various iterations

Several improvements are made through iterations: 1. The goal for the first iteration was mainly to enable and test Leap functions. We generated a low-fidelity layout of the interface. 2. In the second iteration, layout for the keyboard and the UI was improved upon and we also integrated them together to be a live demo. 3. In the third iteration, we developed a high-fidelity leap motion controlled prototype.

Conclusion

We designed a novel way to use mid-air gestures for typing on large-screen displays. The project explores accurate tracking functionalities of the Leap Motion Controller. We started with in-depth literature review studying previous work done in the large-displays and on screen keyboard. Next, we formulized the user requirements for our prototypes and built several iterations using LeapStrap framework. The current prototype uses hover gesture to type-in the words. In the future work, we would like to include a swipe gesture in the keyboard and a more robust prototype for large screen displays.

References

[1] Bhuiyan, M., & Picking, R. (2009, November). Gesture-controlled user interfaces, what have we done and what's next. In *5th Collaborative Research Symposium on Security, E-Learning, Internet and Networking* (pp. 59-60).

[2] O'hara, K., Harper, R., Mentis, H., Sellen, A., & Taylor, A. (2013). On the naturalness of touchless: putting the "interaction" back into NUI. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *20*(1), 5.

[3] Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., & Mackay, W. (2011, May). Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 177-186). ACM.

[4] Vatavu, R. D. (2012, July). User-defined gestures for free-hand TV control. In*Proceedings of the 10th European conference on Interactive tv and video* (pp. 45-48). ACM.

[5] Aigner, R., Wigdor, D., Benko, H., Haller, M., Lindlbauer, D., Ion, A., ... & Redmond, W. A. (2012). Understanding mid-air hand gestures: A study of human preferences in usage of gesture types for hci. *Microsoft Research TechReport MSR-TR-2012-111*.

[6] Nacenta, M. A., Kamber, Y., Qiang, Y., & Kristensson, P. O. (2013, April). Memorability of pre-designed and user-defined gesture sets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1099-1108). ACM.

[7] Shoemaker, G., Findlater, L., Dawson, J. Q., & Booth, K. S. (2009, May). Mid-air text input techniques for very large wall displays. In *Proceedings of Graphics Interface 2009* (pp. 231-238). Canadian Information Processing Society.

[8] Oney, S., Harrison, C., Ogan, A., & Wiese, J. (2013, April). ZoomBoard: A diminutive QWERTY soft keyboard using iterative zooming for ultra-small devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2799-2802). ACM.

[9] Rick, J. (2010, October). Performance optimizations of virtual keyboards for stroke-based text entry on a touch-based tabletop. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (pp. 77-86). ACM.

[10] Bailey, R. W. (1982). *Human performance engineering: A guide for system designers*. Prentice Hall Professional Technical Reference.

[11] Bi, X., Smith, B. A., & Zhai, S. (2010, April). Quasi-qwerty soft keyboard optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 283-286). ACM.

[12] Hwang, S., & Lee, G. (2005, April). Qwerty-like 3x4 keypad layouts for mobile phone. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems* (pp. 1479-1482). ACM.

[13] Mankoff, J., & Abowd, G. D. (1998, November). Cirrin: a word-level unistroke keyboard for pen input. In *Proceedings of the 11th annual ACM symposium on User interface software and technology* (pp. 213-214). ACM.

[14] MacKenzie, I. S., & Buxton, W. (1992, June). Extending Fitts' law to two-dimensional tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 219-226). ACM.