RingBoard - A Dynamic Virtual Keyboard for Fist Based Text Entry

Bradley Wojcik, Tony Morelli, Bryan Hoeft

Central Michigan University

wojci1bc@cmich.edu. morel1a@cmich.edu, hoeft1bk@cmich.edu

Abstract

The combination of a touch sensitive mat and a projector allow a dynamic keyboard to be projected around a user's fist for text entry. RingBoard is a soft keyboard approach using standard off the shelf components that allows people with motor impairments to interact with a computer. The research presented here focuses on increasing the accessibility of text input to a computer utilizing an innovative dynamic keyboard approach. Two different input methods were analyzed for speed and accuracy in a user study consisting of 20 participants. The results of this study show a fist based touch interaction method is as effective as a finger pointing interaction method when using a large touch mat with a top facing projected image.

Keywords

Input, Keyboard, Accessibility, Typing

Introduction

Interaction with an electronic computing device such as a phone or a computer is typically done through the use of a keyboard. Mobile devices traditionally use an onscreen virtual keyboard, while computers use a physical keyboard. The research in this paper focuses on creating a virtual keyboard for use with a personal computer such that a person with a mobility disability who cannot utilize a standard physical keyboard would be able to better interact with a standard computer.

Background

Computer users with upper mobility issues may have difficulties using a standard computer keyboard. It has been shown that people with physical disabilities such as Cerebral Palsy, Spina Bifida, and tremors due to stroke or Parkinson's disease take longer to complete typing tasks on a standard keyboard (Trewin 1999). Results of the Trewin study showed that users with motor impairments committed errors that can be summarized by the following characteristics:

- Long key press – holding a key longer than the designated key repeat delay.
- Additional key – pressing a key adjacent or near to the intended key in addition or instead of the intended key.
- Missing key – the intended key was not pressed either because it was missed or not pressed hard enough.
- Dropping – two keys were failed to be held at the same time (using the shift key).
- Bounce – the intended key was accidentally pressed more than once.
- Remote – a key different from the intended key was pressed by a body part other than the one being used for the intended key.
- Transposition – two keys were transposed.

There have been attempts to correct these issues through the use of technology. Some modifications have come through software solutions using standard keyboards such as adjusting the repeat time of a key, looking for sticky keys, and using CAPS Lock (Trewin 2002). A soft keyboard application was developed for a floor based touch interface such that the users would be able to input data into the computer through the use of the feet (Nguyen 2012). This type of interface would be useful to someone who had no use of the hands. Another approach (Ahsan 2014) was to print out a keyboard on a physical piece of paper, and then attach a laser pointer to a user's head. A web cam combined with visual analysis would determine where on the keyboard the laser pointer was highlighting and use that as text input. Onscreen soft keyboards have attempted to allow people to interact with keyboards in different ways. For example, a graphical keyboard (Missimer 2010) that allows the user to change the size and location of virtual onscreen keyboard buttons was created. In addition to customizing character input, alternative onscreen keyboards have been created for specific applications where commonly typed phrases specific to an application (Norte 2007) are already populated on the virtual keyboard, and where the underlying system determines commonly typed phrases for each user (Wandmacher 2008) and prepopulates the keyboard with user specific common words and phrases.

In addition to research papers, there are some commercially available products to assist with typing for people with motor impairments. Keyguards are hard plastic covers that frame the outside of each key. This allows a user with tremors to rest the hand on the keyboard without pressing any keys, then locating the correct key and pressing down. This still requires fine motor control of the fingers but the ability to rest the hand without pressing any keys does lessen the typing effects of tremors. Mac and Windows operating systems also have built in software functionality such as Sticky Keys (eliminates required simultaneous key presses like ctrl-alt-delete), filter keys (handles debouncing of multiple simultaneous presses of the same key) and custom keyboard shortcuts to perform frequent tasks. The settings for sticky and filter keys have been made adaptive (Trewin 2004) such that a user does not need to find the correct values – the underlying software can adjust and tune itself. It has been suggested (Dietz 2009) that standard keyboards receive and upgrade such that the force applied to each key could represent the font size. This type of interaction will be difficult for someone who already has issues interacting with a standard keyboard.

Although the research presented in this paper focuses on creating new methods of text input for regular computers, the results of this could be applied to mobile computers with touch interfaces through the use of an external Bluetooth touch keyboard as it has been shown (Armstrong 2016) that users are OK with using an external keyboard for interaction on a mobile device. Another approach (Gkoumas 2016) to mobile device input includes predictive typing where the keyboard is altered such that letters that are likely to appear next (Goodman 2002) are enlarged. Predictive typing on virtual keyboards can be improved by taking into account the position of the hand (Yin 2013) as well. It has also been suggested (Rashid 2008) that a virtual onscreen keyboard for mobile devices be relative to the location where the user starts typing instead of having the keys always in the same location.

Solution

This paper presents a soft keyboard approach using standard off the shelf components that allows people with motor impairments to interact with a computer. The research presented here focuses on increasing the accessibility of text input to a computer utilizing an innovative dynamic keyboard approach. Two different input methods are analyzed for speed and accuracy.

The dynamic keyboard described here was implemented on an HP Sprout all in one computer. The HP Sprout is a computer running Windows 10 with a touch sensitive mat, a touch screen, a built in down facing projector and camera. The idea was to have the projector show a new style of virtual keyboard such that the user could enter text via the touch mat instead of a traditional keyboard. The touch mat lays flat on a desk surface like a traditional keyboard. The projector could change the keyboard layout as needed. In order to test the effectiveness of this style of input and to determine any potential limitations of the projector and touch mat setup, two different input methods were used - Fist and Finger. In Fist, users would interact with the keyboard using the dominant hand shaped into a fist, and in Finger, users would interact with the keyboard using the index finger on the dominate hand. The purpose of this study was to see if the Fist method was as good or better than the Finger method. The Fist method could be used by someone who has tremors and would benefit from being able to rest the full hand on a solid surface while typing as the key guard allows, but to not require fine motor movements like using your finger to press through the key guard.

The software for this experiment was created in Unity3D and the code was written in C#. A full keyboard was designed including numbers and special characters, however this user study focused on lower case character entry. When a user puts his fist or finger to the touch mat, a keyboard would be shown in a ring shape around the center of the touch point. Evenly spaced throughout the ring were the vowels A,E,I,O and U (Figure 1). If the user wanted to type an A, he would slide his fist or finger over to the letter A. When it was determined that the user's touch area had centered on the A, a letter A would be typed. The user would then remove his hand from the touch pad and re-touch to begin the process over again and type the next letter. If the user wanted to type a consonant, he would once again place his fist or finger on the touch mat, when the ring of vowels was drawn, he would drag his fist or finger between the vowels where the constant would be in alphabetical order. For example if the user wanted to type a C, he would drag his fist or finger to the space between the A and the E (Figure 2). Or if the user wanted to type an M, he would drag his fist or finger to the space between the I and the O. When the system determined that the user had intended to type a consonant that alphabetically fell between two vowels, the ring of vowels would be erased, and the new ring of consonants would be shown around the current position of the hand. At this point the user would drag his finger or fist over the desired consonant. If at any point the user made a mistake, lifting the hand would start the process over again, and if an incorrect letter was typed a large BACKSPACE virtual key was always visible.

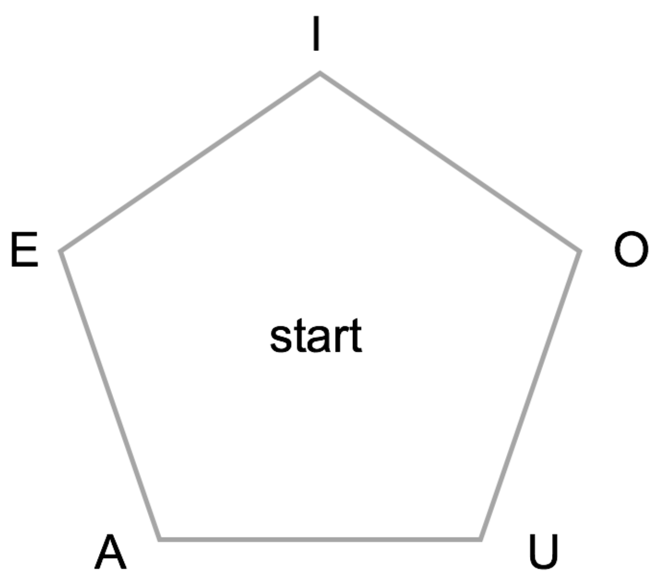

Fig. 1. Primary Keys

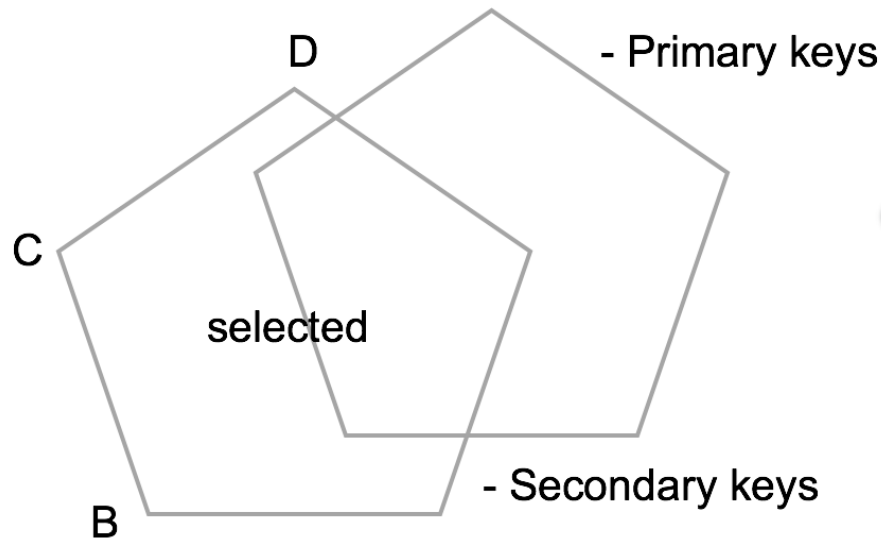

Fig. 2. Secondary Keys

The system as designed should assist with the multiple presses as the user is only allowed to type one character at a time and tremors should be less of an issue as the user is resting his hand on a solid surface in order to interact with it.

Once all the characters were implemented it was noted that several common pieces of keyboard input were missing. For example there was no method to input numbers, or special characters such as punctuation. There was no method to differentiate between capital and lower case letters. There was also no method for entering tabs, spaces, or back spaces. To handle these situations, several buttons were created that were placed on the far left and far right of the touch area. Future versions of the RingBoard may incorporate these into the ring layout. A final screenshot of the RingBoard can be seen in Fig 3.

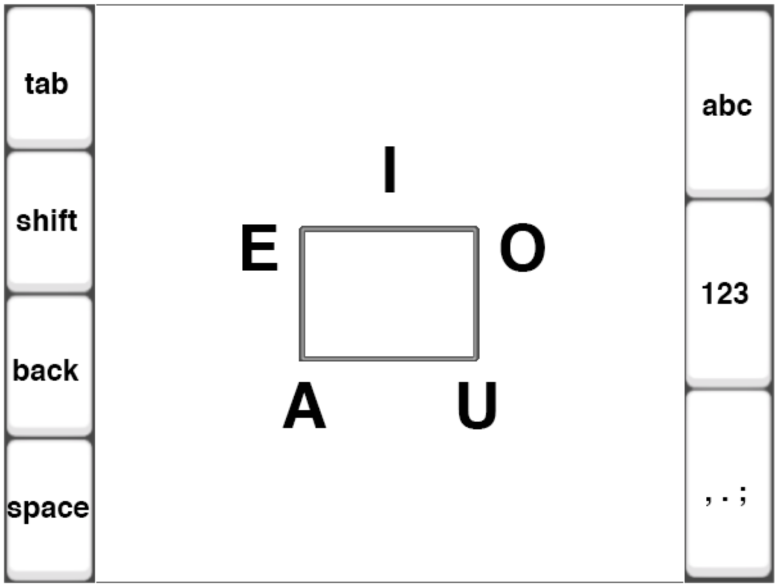

Fig. 3. Complete RingBoard Layout

User Study

In order to test the feasibility of such a system, a 20 person user study was conducted. None of the users reported any mobility related disability. Each user was asked to type out a list of 10 words using the finger, then asked to type out a different list of 10 words using the fist. The sequence of the finger and the fist was chosen at random, and sequence of which list was given to the user first was also chosen at random. At the conclusion of each trial, an individual had typed 20 words, 10 with the fist and 10 with the finger. The data was then analyzed for speed and accuracy. Using a small pointing device such as the finger is expected to be more accurate and faster due to it not obstructing the projected image as much as the fist and it is a more natural way to type as most people interact with their index finger on touch screens as opposed to their entire fist. If the fist method is not significantly slower or more error prone than the index finger, this type of interface could potentially improve the accessibility of a computer by a person who has motor impairments or tremors. The results are broken down into speed and accuracy based on individual characters.

Results

As far as speed goes, on average the finger method was faster than the fist method by taking 96.89% (SD=16%) of the time. This was not a significant difference, however. A paired-T test shows $t(29)=1.86$, $p > 0.05$. The speed of the first level characters (vowels) versus the second level characters (consonants) was also compared. Vowels had an average seek time of 0.156 seconds (SD=0.006) for fist and 0.160 (SD=0.01) for finger. Consonants had an average seek time of 1.00 seconds (SD=0.17) for fist and 0.91 seconds (SD=0.14) for finger.

Error rates were measured as the percentage of characters that were incorrectly chosen when compared to the next character in the given word. On average finger pointing error rates were 85.5% (SD=128%) of the fist method. Average vowel error rates for the fist method were 0.019 (SD=0.02) and for the finger method were 0.034 (SD=0.037). For consonants, error rates for the fist method were 0.046 (SD=0.044) and for the finger method 0.056 (SD=0.053).

Conclusion

The results of this study show a fist based touch interaction method is as effective as a finger pointing interaction method when using a large touch mat with a top facing projected image. This shows promise that a touch based mat could be used as an alternative to hardware based key guards for users who need the support of resting the typing hand but who also lack fine motor movements of the fingers. Future improvements of this type of interaction could be using the touch mat as an external input mechanism for mobile devices. A follow up user study should be performed that analyzes the speed and accuracy of utilizing the special characters on the keyboard and to determine if the special characters are better placed within the ring, or having them on the outside of the touch area is sufficient.

Works Cited

Ahsan, Hiba, et al. "Vision Based Laser Controlled Keyboard System for the Disabled." Proceedings of the 7th International Symposium on Visual Information Communication and Interaction. ACM, 2014.

Armstrong, Patrick, and Brett Wilkinson. "Text entry of physical and virtual keyboards on tablets and the user perception." Proceedings of the 28th Australian Conference on Computer-Human Interaction. ACM, 2016.

Dietz, Paul H., et al. "A practical pressure sensitive computer keyboard." Proceedings of the 22nd annual ACM symposium on User interface software and technology. ACM, 2009.

Gkoumas, Apostolos, Andreas Komninos, and John Garofalakis. "Usability of Visibly Adaptive Smartphone keyboard layouts." Proceedings of the 20th Pan-Hellenic Conference on Informatics. ACM, 2016.

Goodman, Joshua, et al. "Language modeling for soft keyboards." Proceedings of the 7th international conference on Intelligent User Interfaces. ACM, 2002.

Missimer, Eric S., et al. "Customizable keyboard." Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility. ACM, 2010.

Nguyen, Tuan, et al. "The UITK: Towards the designing a ubiquitous soft keyboard for disabled people." Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services. ACM, 2012.

Norte, Stéphane, and Fernando Graça Lobo. "A virtual logo keyboard for people with motor disabilities." ACM SIGCSE Bulletin. Vol. 39. No. 3. ACM, 2007.

Rashid, Daniel R., and Noah A. Smith. "Relative keyboard input system." Proceedings of the 13th international conference on intelligent user interfaces. ACM, 2008.

Trewin, Shari. "An invisible keyguard." Proceedings of the fifth international ACM conference on Assistive technologies. ACM, 2002.

Trewin, Shari. "Automating accessibility: the dynamic keyboard." ACM SIGACCESS Accessibility and Computing. No. 77-78. ACM, 2004.

Trewin, Shari, and Helen Pain. "Keyboard and mouse errors due to motor disabilities." International Journal of Human-Computer Studies 50.2 (1999): 109-144.

Wandmacher, Tonio, et al. "Sibylle, an assistive communication system adapting to the context and its user." ACM Transactions on Accessible Computing (TACCESS) 1.1 (2008): 6.

Yin, Ying, et al. "Making touchscreen keyboards adaptive to keys, hand postures, and individuals: a hierarchical spatial backoff model approach." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2013.