TTracker: Using Finger Detection to Improve Touch Typing Training

Elvin Kollie, Fernando Loizides, Thomas Hartley, Adam Worrallo

TTracker: Using Finger Detection to Improve Touch Typing Training

Elvin Kollie[1], Fernando Loizides[1], Thomas Hartley[1], Adam Worrallo[1]

{ElvinKollie ,fernando.loizides, T.Hartley2, a.worrallo} @wlv.ac.uk

University of Wolverhampton

Abstract. Touch typing software teaches a user to use the correct finger combinations with the correct keyboard buttons. The ultimate goal is to teach the typist to type faster, more accurately and ergonomically correct. Our research presents the working prototype of a software and hardware setup that tracks not only the speed and accuracy of the correct buttons being pressed but also which fingers are used to press them; a dimension of training that has previously not been integrated into touch typing tutorials. We use novel technology (leap motion) to detect the accurate interaction between the user and the keyboard, giving precise feedback to the user in order for him or her to improve.

Keywords. Finger Detection, Leap Motion, Touch Typing

1 Introduction and Motivation

Touch-typing is a psychomotor skill involving combining a users' cognitive functions with physical movement. Touch-typing involves more than simply increasing typing speed, in that it enables typists to split the cognitive processing by offloading information from the visual channel. This means typists can designate more focus to interacting with the screen and therefore inadvertently increasing their typing speed in the process [5]. This makes touch typing beneficial in note taking, programming, live communication and many other aspects of computer use.

Touch-typing is conventionally learned via websites that teach touch-typing concepts and finger mappings [1]. The websites or stand-alone software identify which keys are pressed and the time taken to do so; thus giving feedback to the users in order to improve their performance. The positive influence that feedback has on typing proficiency has been proven multiple times on multiple devices [2, 3, 4]. What the current feedback does not provide the user is the fingers used to press the keys, one of the areas in which the most mistakes may be made. In order to provide this type of feedback on touch-typing a method to track finger key associations needs to be in place. It would be impossible for a human to consistently detect finger-key associations over long periods of time.

Leveraging hand tracking technology, it is possible to identify if a user is touch-typing or not, providing a method to give feedback to people learning how to touch-type. The feedback provided could prove to be a more effective method of teaching people how to touch-type. In order to prove the concept of the new technology, a system is developed using the Leap Motion controller[1] to track hands actively typing on a keyboard.

In this paper we present the working prototype system and setup created to facilitate finger detection and feedback. We also present a pilot user test to identify initial interaction effectiveness of the prototype and to feed back in the development according to a user centered design approach. The next section presents the technology (software and hardware) while the subsequent section presents the pilot test with some of the main findings.

2 System Description

The prototype system consists of two parts, the software called TTracker and the hardware setup. The hardware part consists of a laptop or PC with a Leap Motion mounted on top of the keyboard area (See Figure 1). The Leap Motion controller is a USB peripheral device capable of tracking hands with a submillimeter accuracy. The leap motion uses infrared cameras to collect information at up to 200 frames per second. The effective range of the Leap Motion controller extends from approximately 25 to 600 millimeters above the device (1 inch to 2 feet). The field of view is 150° with roughly 8 cubic feet of interactive 3D space. If for some reason a finger were to be obstructed from the view of the Leap Motion it would use its predictive model to infer the location [6]. The Leap Motion then filters all of the unneeded information within the snapshot and expresses the results as a series of frames. These frames hold data such as fingertip location, palm location and finger direction.

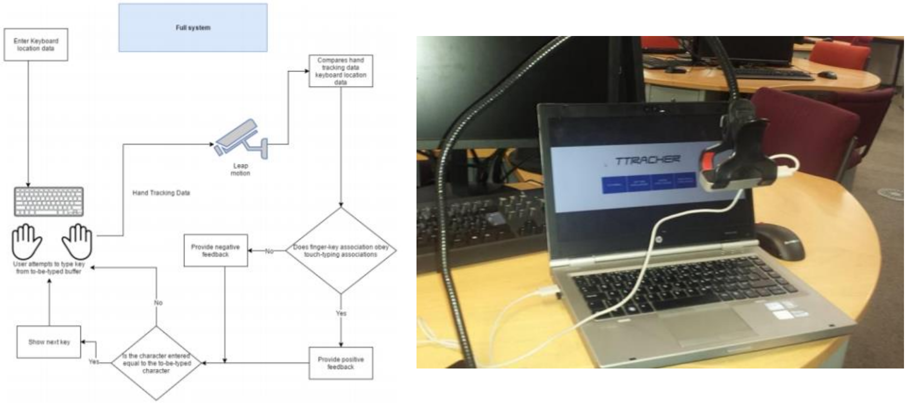
Full system
Enter Keyboard location data
Compares hand tracking data keyboard location data
Leap motion
Hand Tracking Data
User attempts to type key from to-be-typed buffer
Provide negative feedback
No
Does finger-key association obey touch-typing associations
Yes
No
Show next key
Yes
Is the character entered equal to the to-be-typed character
Provide positive feedback
TTRACKER

Fig. 1. (left) the prototype architecture (right) the physical setup

[1] https://www.leapmotion.com/ (accessed December 2016)

The TTracker software was written in Unity[2] and, although not open source, can be provided to researchers upon request (See Figure 2). The software has two modules working together. The first module is the finger-key tracking module, which gathers information from the Leap Motions. The second module is the learning environment, which was put together with the first module to create the complete system.

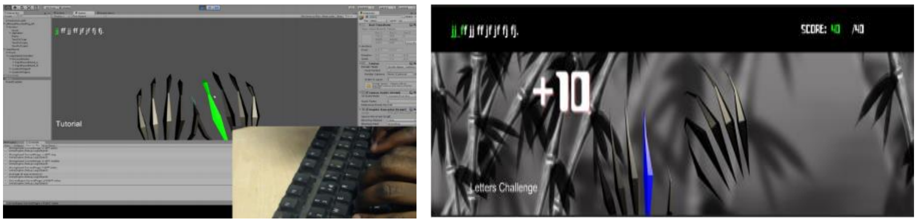

Fig. 2. (left) the development process of TTracker (right) the prototype software showing the left index finger as the suitable finger for the next button press.

Specialized sentences and letters appear on the screen for the user to type. There are different levels depending on the skill level of the user. In order to teach touch mapping without breaking the user's focus from the screen, the application was set up so that the next finger to use on the to-be-typed screen flashes blue. After a key is pushed, the application checks if touch-typing was used, the finger used glows red for incorrect finger to key touch or green if the correct finger to key is used. The user gains 5 points for typing the correct letter but if they used touch-typing, they gain 10 points thereby incentivizing the use of touch-typing. If the user types the wrong letter, he / she loses 1 point. Sound was added to the application; a different audio file is played depending on touch-typing implementation.

3 Pilot User Testing

We wanted to test the prototype to improve the usability before releasing a beta version to the general public. To this end a pilot study took place with 10 participants aged 20-60 for qualitative and quantitative feedback. The test involved a pre-study interview in order to record the participants' profiles, a hands-on test with the prototype which lasted no more than 20 minutes and a semi-structured post-test interview.

The pre-study interview revealed some demographics about the participants and the level of touch typing experience that the participants had. 6 of the participants had knowledge of what touch typing was. 3 of the participants had previously tried (limited) to learn touch typing using an online resource. None of the participants were able to touch type in any formal way. All participants would use a computer at least 5 days a week and 7 would use it daily.

We want to find out how many errors users make while touch-typing and attempt to discover new ways to prevent these errors. Identifying how long it takes users to learn

[2] https://unity3d.com/

the touch-typing mapping is also imperative. During the prototype testing we were able to successfully record errors in using the correct finger to press specific keys (see Figure 3). With this method we can give individual feedback to each user on the areas they need to be improve specifically and also be able to feedback from all the users on the most common problematic keys and adjust the software training to cater for those.

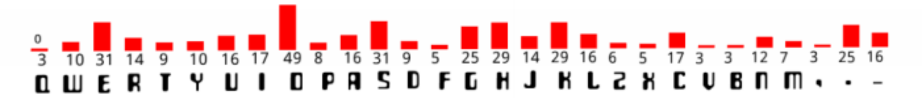

Fig. 3. Errors in matching the correct finger to the correct key press for an individual user

In the post-study interview, the main questions revolved around the usability of the system. All participants agreed that the software' learnability was easy to grasp. The participants were unanimous in agreeing that highlighting the finger that is needed to press the next key on the screen was sometimes confusing. Visual attention is focused on the letters appearing on the screen and is therefore difficult to be shared with the image of the fingers. Longitudinal testing we hypothesize will gradually help the users learn the finger positions at which point they will not rely on the on screen finger animation. The general outcome was that the software would be better for registering and giving feedback after, rather than at, real time training. Only one participant reported the accuracy of the setup (finger to key detection) as less than 'extremely accurate' and we attribute this as an outlier due to calibration issues. The next version of the software was also suggested to include different auditory feedback for different types of presses.

4 References

1. Poole, D. and Preciado, M. (2016) Touch typing instruction: Elementary teachers' beliefs and practices.Computers & Education, 102, pp. 1-14.
2. Tittelbach, D., Fields, L. and Alvero, A. (2008) Effects of Performance Feedback on Typing Speed and Accuracy. Journal of Organizational Behavior Management, 27(4), pp. 29-52.
3. Kim, J. and Tan, H. (2014) A Study of Touch Typing Performance with Keyclick Feedback. In Haptics Symposium, New York: IEEE, pp. 227-233.
4. Kim, J. and Tan, H. (2015) Effect of Information Content in Sensory Feedback on Typing Performance using a Flat Keyboard. In World Haptics Conference, New York: IEEE, pp. 228- 234.
5. Feit, A., Weir, D. and Oulasvirta, A. (2016) How We Type: Movement Strategies and Performance in Everyday Typing. In #Chi4good, pp. 4262-4273.
6. Colgan, A. (2014) How Does the Leap Motion Controller Work?. Leap Motion Blog, [online]. [Accessed 14 September 2016]. Available at: <http://blog.leapmotion.com/hardware-tosoftware-how-does-the-leap-motion-controller-work/>.
7. Nielson, J. (2012) Usability 101: Introduction to Usability. Nngroup.com, [online]. [Accessed 24 September 2016]. Available at: https://www.nngroup.com/articles/usability-101-introduction-to-usability/