Tapulator: A Non-Visual Calculator using Natural Prefix-Free Codes

Vaspol Ruamviboonsuk
Computer Science and Engineering
University of Washington
Seattle, WA 98195
vaspol@cs.washington.edu

Shiri Azenkot
Computer Science and Engineering
University of Washington
Seattle, WA 98195
shiri@cs.washington.edu

Richard E. Ladner
Computer Science and Engineering
University of Washington
Seattle, WA 98195
ladner@cs.washington.edu

ABSTRACT
A new non-visual method of numeric entry into a smartphone is designed, implemented, and tested. Users tap the smartphone screen with one to three fingers or swipe the screen in order to enter numbers. No buttons are used—only simple, easy-to-remember gestures. A preliminary evaluation with sighted users compares the method to a standard accessible numeric keyboard with a VoiceOver-like screen reader interface for non-visual entry. We found that users entered numbers faster and with higher accuracy with our number entry method than with a VoiceOver-like interface, showing there is potential for use among blind people as well. The Tapulator, a complete calculator based on this non-visual numeric entry that uses simple gestures for arithmetic operations and other calculator actions is described.

Categories and Subject Descriptors
K4.2 [Social Issues]: Assistive technologies for persons with disabilities, **H.1.2 [User/Machine Systems]:** Human factors

General Terms
Design, Human Factors.

Keywords
Number entry, Calculator, Non-visual interface, Blind.

1. INTRODUCTION
Simple math calculation is one of the daily tasks that we usually perform using a calculator. Currently, calculators are easy to use and can be found everywhere. Calculators on smartphones commonly have soft buttons with 20 buttons or more where ten are reserved for the digits 0 through 9. There are additional buttons for the decimal point, five arithmetic operations, *clear current number*, *clear current calculation*, *backspace*, and *equals* (calculate). Smartphone calculators can be made non-visually accessible by using a touch screen reader pioneered by Kane *et al.* [3] and found on iPhones (VoiceOver) and Android phones (TalkBack). Blind and low-vision users of smartphones typically use these screen readers that allow a user to explore the screen with a finger without activating any controls, listening to what is under the finger. Activation of a button, for entering a digit or other action, requires a double tap anywhere or split tap while touching the button.

In this demonstration, we present the Tapulator, a new non-visual approach to a multi-touch screen calculator that uses simple gestures instead of buttons. In particular, a natural prefix-free code for entering digits with up to three fingers is used. Similar to a screen reader, auditory, rather than visual, feedback is used to indicate what digits and operations are entered.

The Tapulator is universally designed and can be used by anyone to make a calculation in a non-visual way, and there is no need to learn how to use a touch screen reader. For blind and low-vision users, the Tapulator can be faster for doing a calculation than a typical accessible button calculator because there is no need to explore the screen to find the right button to enter a digit or other operation.

For this demonstration, we examine in detail number entry without a decimal point. We provide some preliminary results that number entry is faster and more accurate using the Tapulator than using a button calculator with a touch screen reader. We then briefly describe the design of the full Tapulator.

ASSETS '12, October 22-24, 2012, Boulder, Colorado, USA.

2. RELATED WORK
Text entry, including numeric digits, is a long studied field. The H4-Writer of MacKenzie *et al.* uses Huffman codes [3], which are optimal prefix-free codes to enter text [6]. Such codes are not generally easy to remember, but once they are memorized, good input rates can be achieved. For Tapulator numeric input we also use a prefix-free code, but it is natural and easy to learn. Another key difference between MacKenzie *et al*'s work and ours is that the implementation of H4-Writer uses buttons, while the Tapulator implementation accepts touches anywhere on the screen.

The Perkinput text entry method, introduced by Azenkot *et al*. [1], is a Braille-based text entry system for multi-touch screens. Like the Tapulator, Perkinput does not have buttons, but unlike the Tapulator, Perkinput requires calibration to determine which fingers correspond to which touch points.

A study of gestures for multi-touch screens created by blind users by Kane *et al*. [5] indicates that blind people have different preferences for gestures than do sighted users. In addition, blind users were generally not as precise as sighted users when making gestures that required the production of a specific shape such as a circle or triangle. Following these findings, the Tapulator does not use any gestures other than taps and swipes.

3. TAPULATOR NUMERIC ENTRY
For numeric entry, the Tapulator uses a simple prefix-free code that uses multi-finger taps with one to three fingers and swipes. A prefix-free code requires that no codeword is a prefix of another. This allows a multi-digit number to be entered unambiguously as a sequence of codewords. The code is natural, where 0 is represented by a swipe; the digits 1 and 2 are represented by one and two finger taps, respectively; the digit 3 is

represented by a two-finger tap followed by a swipe (i.e., 3 and 0). The digit 4 is represented by a three-finger tap followed by a one-finger tap (i.e.. 3 and 1). Table 1 describes the complete code.

Table 1. Tapulator numeric codes.

Numbers	Code	Gestures
0	0	swipe
1	1	1-finger tap
2	2	2-finger tap
3	3 + 0	3-finger tap + swipe
4	3 + 1	3-finger tap + 1-finger tap
5	3 + 2	3-finger tap + 2-finger tap
6	3 + 3 + 0	2 × (3-finger tap) + swipe
7	3 + 3 + 1	2 × (3-finger tap) + 1-finger tap
8	3 + 3 + 2	2 × (3-finger tap) + 2-finger tap
9	3 + 3 + 3	3 × (3-fingers tap)

As can be readily seen, this code is easy to remember and this is validated to some extent in our preliminary evaluation. Note that this is not an optimal prefix-free code, assuming that all digits are equally likely. An optimal code that uses four symbols has an average of 1.8 actions per digit, while the Tapulator code has 2.1 actions per digit, where an action is a tap or a swipe. An optimal code would likely require more time to learn, which may inhibit adoption.

During our design process, we considered a five-symbol code where a four-finger tap represents the number 4. We found through formative evaluations that since the fourth finger (the "pinky") is typically much shorter than the other three fingers, four-finger tapping is somewhat awkward to perform on small touch screens.

4. PRELIMINARY EVALUATION

To compare Tapulator numeric entry with standard accessible numeric entry (hereafter, the "standard"), we present (1) a theoretical analysis of the techniques and (2) an empirical comparison with five users.

In the standard calculator there are two components to entering a digit. The first is the seek time to find the digit's button. The effort needed during the seek time is difficult to quantify but it does require listening to the buttons touched until the correct button is found. When the button is found, a double or split tap is needed to enter the digit. At first blush, this appears to be more difficult than an average of 2.1 taps per digit.

Our empirical evaluation included a study with five sighted participants, who entered 10 six-digit numbers on a smartphone using the Tapulator and the standard method. Participants held the smartphone beneath a desk so that they were unable to see the screen. After a brief training period, participants entered numbers at an average rate of 1.99 seconds per digit (*SD* = 1.25) with Tapulator, and 2.77 seconds per digit (*SD* = 1.24) with the standard method. The error rates of the final transcribed numbers were far lower for Tapulator entry: participants averaged a Mean String Distance (MSD) of just 1.0% on the Tapulator and 14.2% with the standard method. Thus, the Tapulator out-performed the standard calculator in both speed and accuracy. Unsurprisingly, all five participants preferred the Tapulator to the other method.

While these results are only preliminary, they show the Tapulator has potential to out-perform a calculator with buttons for blind and low-vision people. We plan to conduct a more rigorous evaluation with blind users in the future.

5. TAPULATOR GESTURES

We do not have space to describe all the gestures needed for the Tapulator. All the operations, other than numeric entry, will begin with a three-finger swipe so as not to be confused with numeric entry. The gestures are natural corresponding to the operations as printed, making them also easy to learn. For example, "equals" is a two-finger swipe from left to right and plus is two consecutive swipes, up to down, then left to right. This is similar to the technique presented by Findlater *et al.* [2] for entering non-alphanumeric characters.

6. CONCLUSION AND FUTURE WORK

The Tapulator is in an early stage of development. More evaluation has to be conducted with blind participants with a fully implemented Tapulator, comparing it to a button calculator. We also plan to compare various potential codes for their ease of learning and use.

ACKNOWLEDGEMENTS

This work was partially funded by the National Science Foundation Grant No. IIS-1116051 and the Department of Education Grant No. H327A100014.

REFERENCES

[1] S. Azenkot, J.O. Wobbrock, S. Prasain, and R.E. Ladner (2012). Input Finger Detection for nonvisual touch screen text entry in Perkinput. In *Proceedings of Graphics Interface 2012* (GI '12). Toronto, Ontario: Canadian Information Processing Society, 121-129.

[2] Findlater, L., Lee, B.Q. and Wobbrock, J.O. (2012). Beyond Qwerty: Augmenting touch screen keyboards with multi-touch gestures for non-alphanumeric input. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '12). New York: ACM Press, pp. 2679-2682.

[3] D.A. Huffman (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the I.R.E*. 1098–1102.

[4] S.K. Kane, J.P. Bigham, J.O. Wobbrock (2008). Slide Rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS '08). 73-80.

[5] S.K. Kane, J.O. Wobbrock, R.E. Ladner (2011). Usable gestures for blind people: understanding preference and performance. *Proceedings of the 2011 Annual Conference on Human factors in Computing Systems* (CHI '11). 413-422.

[6] I.S. MacKenzie, R.W Soukoreff, J. Helga (2011). 1 thumb, 4 buttons, 20 words per minute: design and evaluation of H4-writer. *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11). 471-480