

# Tap2Count – Numerical Input for Interactive Tabletops

*Tobias Hesselmann*

OFFIS Institute for IT

Escherweg 2

26121 Oldenburg, Germany

tobias.hesselmann@offis.de

*Wilko Heuten*

OFFIS Institute for IT

Escherweg 2

26121 Oldenburg, Germany

wilko.heuten@offis.de

*Susanne Boll*

University of Oldenburg

Escherweg 2

26121 Oldenburg, Germany

susanne.boll@uni-oldenburg.de

## ABSTRACT

We present a technique to enter numbers on interactive multi-touch tabletops using the 10 fingers of both hands. We recognize the number of fingers simultaneously touching the screen and interpret them as digits from 0 to 9 to represent any number in the decimal system. Our technique works independent from the location and orientation of users at the tabletop and does not occlude screen space, rendering it an interesting alternative to commonly used techniques for numerical input on touchscreens, such as virtual keyboards and handwriting recognition systems.

**ACM Classification:** H.5.2 [Information interfaces and presentation]: User Interfaces – Input devices and strategies, Interaction Styles.

**General terms:** Design, Performance, Human Factors

**Keywords:** Numerical Input, Text Input, Interactive Surfaces, Interactive Tabletops, Multi-Touch

## INTRODUCTION

Numerical input is required by many applications for interactive tabletops and surfaces (ITS), ranging from point-of-sales systems to analytical, mathematical and educational systems. For that purpose, virtual keyboards and gestural handwriting systems [1] are two of the most used techniques. But in the context of ITS computing, the suitability of these approaches is questionable: Virtual keyboards typically need to be positioned next to a user's location and need to be orientated towards the user to ensure proper reachability and readability of the visualized keys (compare [5]). Unfortunately though, automatic layout of virtual keyboards is impossible for most setups, as the position and orientation of users at the table are typically unknown. Moreover, virtual keyboards need to be displayed on top of other contents on the screen to interact with them, occluding other visualized objects. This issue gets worse in multi-user scenarios, requiring a separate keyboard for each user for simultaneous data input.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*ITS 2011*, November 13-16, Kobe, Japan.

Copyright 2011 ACM 978-1-4503-0871-7/11/11....\$10.00.



Figure 1: A user enters the numbers 1, 8 and 0 using one (left), eight (middle) and ten fingers (right).

Gestural handwriting recognizers seem to suit ITS systems better at first sight. Users can write numbers on the tabletop at any position and facing any orientation, and these numbers can then be algorithmically transcribed into their digital representations. Moreover, handwriting recognizers do not require a visualization to interact with, so contents on the screen are usually not occluded using this approach. Nevertheless, larger ITS systems are usually “bare hands” setups. Thus, when users write numbers on the screen, friction induced by the movement of the fingers on the surface can render this type of input uncomfortable to users. Also, the high physical strain caused by gestural handwriting techniques can cause additional discomfort [3].

With TAP2COUNT, we present an alternative method for numerical input on interactive tabletops, which makes use of the multi-touch capabilities of modern touchscreens: We detect the number of fingers simultaneously touching the screen, and map this number to the digits 0 to 9, which enables us to express any number of the decimal system. We pursue a “best-of-both-worlds” approach: Similar to gestural handwriting methods, TAP2COUNT works independent of the location and orientation of users at the tabletop, and it does not occlude contents on the display. Moreover, the muscular movements needed to use our technique are comparable to those when using virtual keyboards, inducing less physical strain than gestural handwriting approaches.

## TAP TO COUNT

Traditional touchscreens can typically detect only one or two touches simultaneously. With the advent of true multi-touch surfaces, this limitation is becoming more and more obsolete. Modern ITS can easily recognize 10 or more simultaneous touches, paving the way for more sophisticated multi-touch interaction techniques.

With TAP2COUNT we exploit this characteristic and introduce a novel method for entering numbers on multi-touch screens: To enter a number, the user simply taps the screen with the appropriate number of fingers representing a particular number (see Figure 1).

Entering the numbers 1 to 9 using both hands is straightforward. Nevertheless, to display all numbers in the decimal system, we also need to find a suitable representation for entering zeroes, which imposes a problem in the ITS context: As tabletop systems typically only react to actual input and not the absence of input, zeroes cannot be expressed by simply not tapping the screen with any finger.

Instead, we represent the zero by simultaneously tapping the screen with all ten fingers of both hands (see Figure 1, right). While other representations are technically possible, e.g., pressing the heel of the hand on the surface or clenching a fist on the surface, this would imply a significant change of the user's hand posture, possibly causing negative effects on the input performance of users.

### Technical Realization

Figure 2 shows the automatic interpretation of the presented approach using a typical hardware setup. The photo on the left shows a user tapping the screen of a vision-based multi-touch tabletop with 8 fingers. Infrared light is reflected by the fingertips touching the surface, which is picked up by a camera under the surface, resulting in the sampled camera image in the middle of the figure. After processing the image using an image filter chain, the touches of the user's fingertips can robustly be extracted from the image (Figure 2, right). The system then counts the maximum number of fingertips simultaneously touching the screen. When all fingers have been lifted from the surface again, the system maps the input to a corresponding digit.

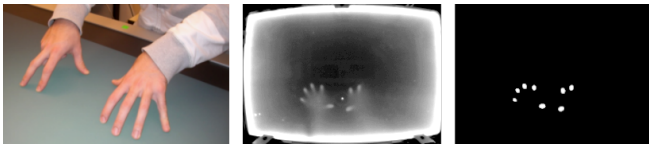


Figure 2: Technical realization of Tap2Count. Left: Participant enters an '8' on an interactive tabletop. Middle: Original camera image. Right: Processed camera image.

### RELATED WORK

With “Finger-Count”, Bailly et al. presented a technique similar to ours to launch functions from drop-down menus [1]. They used menus with a maximum depth and breadth of 5 items and consequently recognized numbers from 1 to 5. In contrast, we recognize the full range of the decimal system, i.e., the numbers 0-9. Also, their approach makes use of both hands sequentially, while our approach uses both hands in parallel. Thus, our approach extends the range of possibilities in contrast to [1], but also raises new research questions, such as the intuitiveness, cognitive demand, and performance of a parallel dual-handed approach.

The mapping of fingers to numbers, i.e., the numeric encoding used in our work, is similar to a finger-counting technique presented by Lepouras [4]. Nevertheless, this technique is targeted towards virtual environments and differs fundamentally from ours, as it uses data gloves to detect individual hand postures in mid-air, whereas we detect fingertips touching the surface of a horizontal surface.

### EVALUATION AND CONCLUSION

In an initial pilot evaluation, we asked 12 users to enter series of 10 pseudo-randomized digits using TAP2COUNT on a typical tabletop system (see Figure 2, left). As we aimed for a qualitative impression only, we performed no automatic recognition of digits and employed a wizard-of-oz approach instead, observing participants and changing the number series manually after they entered them.

One encouraging result of our studies was that TAP2COUNT was easy to understand by all participants. After explaining the concept, only one participant asked again how to express a zero, while all others instantly internalized the technique. On the downside, participants assessed the mental demand using TAP2COUNT as fairly high, probably caused by the unfamiliar simultaneous usage of both hands to enter zeroes and numbers greater than 5. Yet, most participants also felt they improved considerably after some test runs. One participant even stated she found TAP2COUNT easier to learn than typing on a traditional keyboard. Another participant mentioned that she found the physical demand less demanding than typing on an ordinary keyboard.

Interestingly, we observed great variations in performance. Some participants performed very fast from the very beginning of the evaluation without noticeable errors, while others made many errors (which they noticed themselves) and thought for a longer time before entering a certain number.

Concluding, we believe that TAP2COUNT is an interesting alternative to other commonly used methods for numerical input on ITS, such as virtual keyboards or handwriting recognizers. Conceptually, it works independent of the location and orientation of users at tabletop systems, and feedback from participants in our pilot studies indicates that friction and physical strain were not found problematic.

Future research should aim at gathering more insights about the performance and learning curve of TAP2COUNT in real life settings, probably comparing our approach to other techniques for numerical input on tabletop systems.

### REFERENCES

1. Bailly, G., Lecolinet, E., Guiard, Y.: Finger-count and radial-stroke shortcuts: 2 techniques for augmenting linear menus on multi-touch surfaces. In: Proc. CHI 2010, ACM (2010) 591–594
2. Hinrichs, U., Hancock, M., Collins, C., Carpendale, S.: Examination of text-entry methods for tabletop displays. In: Proc. Tabletop 2007. (2007) 105–112
3. Hirche, J., Bomark, P., Bauer, M., Solyga, P.: Adaptive interface for text input on large-scale interactive surfaces. In: Proc. Tabletop 2008, IEEE (2008)
4. Lepouras, G.: Numerical input techniques for immersive virtual environments. In: Proc. VECIMS 2009, IEEE (2009) 240–245
5. Ryall, K., Forlines, C., Shen, C., Morris, M.R., Everitt, K.: Experiences with and observations of direct-touch tabletops. In: Proc. TABLETOP 2006, IEEE Computer Society (2006) 89–96