



## INSTITUT FÜR INFORMATIK

Sonderforschungsbereich 342:  
Methoden und Werkzeuge für die Nutzung  
paralleler Rechnerarchitekturen

# **Gestyboard v1.0: A 10-finger-system and gesture based text input system for multi-touchscreens w ith no need for tactile feedback**

Tayfur Coskun, Eva Artinger, Lorenzo Pirritano, Daniela Korhammer,  
Amal Benzina, Claudia Grill, Andreas Dippon, Gudrun Klinker

TUM-I1111  
SFB-Bericht Nr. 342/22/11 I  
Dezember 11

TUM-INFO-12-I1111-150/1.-FI

Alle Rechte vorbehalten  
Nachdruck auch auszugsweise verboten

©2011 SFB 342 Methoden und Werkzeuge für  
die Nutzung paralleler Architekturen

Anforderungen an: Prof. Dr. A. Bode  
Sprecher SFB 342  
Institut für Informatik  
Technische Universität München  
D-80290 München, Germany

Druck: Fakultät für Informatik der  
Technischen Universität München

# Gestyboard: A 10-finger-system and gesture based text input system for multi-touchscreens with no need for tactile feedback

Tayfur Coskun, Eva Artinger, Lorenzo Pirritano, Daniela Korhammer,  
Amal Benzina, Claudia Grill, Andreas Dippon, Gudrun Klinker  
Technische Universität München<sup>1</sup>

## ABSTRACT

This paper presents a new text-entry approach based on the 10-finger-system<sup>1</sup> for multi-touch devices. A unique finger gesture-to-key mapping system is developed to overcome the lack of tactile feedback on touchscreens. The missing tactile feedback is in fact the main reason behind the low performance of virtual touchscreen keyboards with soft buttons. For this reason, we use unique gestures for all 10 fingers based on the German QWERTZ layout of the classical hardware keyboard. There are two classes of gestures: Taps and Slidings (Sliding gestures). Taps activate the central row letters ‘a’, ‘s’, ‘d’, ‘f’, and ‘j’, ‘k’, ‘l’, ‘;’, while slidings are used for the remaining neighboring letters. The idea behind this is to perform the gestures without looking at the “virtual keyboard” and without tactile feedback. Since the gestures are based on the QWERTZ-layout, we expect the users to easier adapt to this new text-input system. By coupling the gestures to the highly efficient 10-finger-system the performance of the text entry can also be improved. We developed a first prototype, called Gestyboard, and conducted an evaluation with 41 pupils who were familiar with the 10-finger-system.

Through this evaluation, we are aiming to investigate the potential usage and the learnability of the first Gestyboard prototype. The results revealed that despite the complexity of the finger gestures, the users were able to perform the text entry with our prototype, and the results also highlighted additional improvements to increase the performance of our concept. As expected the Gestyboard performance achieved neither that of the classical hardware nor the virtual touchscreen keyboard because of their vast familiarity and the current limitations of the first Gestyboard prototype. However, the Gestyboard prototype showed a large acceptance and enthusiasm among the users. This concept has been developed

<sup>1</sup>This is also called the touch typing system in literature, for example: <http://www.articleclick.com/Article/The-10-finger-system—touch-typing-for-you-11004290>

within the SpeedUp project<sup>2</sup>.

## Author Keywords

Virtual Keyboard, Touch typing, 10-finger-system, Multi-touch Device, Gestures, No tactile feedback

## ACM Classification Keywords

H.5.2 User Interfaces: Multi-touchscreen keyboards

## General Terms

Algorithms, Experimentations, Human Factors, Performance

## INTRODUCTION

The increased distribution and popularity of smartphones and tablet devices have led throughout the years to new user interface challenges in terms of applications and input mechanisms. Those new challenges are mainly because of the lack of keyboard and mouse input in such devices. Due to the high efficiency and the vast acceptance of keyboard and mouse as input devices among the users, it is a tremendously known challenge and an ongoing research to develop touchscreen input mechanisms to compete with them. Consequently, many new touchscreen user interfaces have been developed in the last years. In this work, we are introducing a novel concept for text entry.

Many different concepts for smartphones and tablet devices have been researched and developed, the T9 system, Swype [6] or a touchscreen version of the QWERTZ layout might be some famous examples. In literature the latter is often referred to as virtual touchscreen keyboard or soft keyboard, and is provided in most of today’s operating systems running on touchscreen devices, mostly mobile devices and multi-touch tables. Although the virtual touchscreen keyboards perform quite well and are very intuitive to the user, their performance does not match that of the classical hardware keyboard. It is well known that the missing tactile feedback on touchscreens hinders the performance of soft buttons [7]. Additionally, in the case of the QWERTZ hardware keyboard the keys ‘f’ and ‘j’ contain ridges which can be sensed

<sup>2</sup>The project SpeedUp is funded by the German Federal Ministry of Education and Research (BMBF) within the program “Research for Civil Security” (May 1st, 2009 - April 30th, 2012, FKZ: 13N10175). Website: <http://www.speedup-projekt.de>

and hence recognized by the user. The ridges are there to allow the users to blind type on the keyboard and enable them to sense the keys without activating them. Since such tactile feedback is missing with the virtual touchscreen keyboard, the text entry is more error prone and slower. These tactile limitations have to be overcome by a virtual keyboard to perform comparably to a classical hardware keyboard. For that purpose, the virtual touchscreen keyboard has to fulfill the same requirements as the hardware one, which are discussed in Section “Requirements”. Many research projects have proposed solutions to reach the classical hardware keyboard text entry performance, some of them are briefed in Section “Related Work”.

We developed a virtual text input concept and implemented a first prototype which fulfills the requirements mentioned before. This concept is based on finger gestures, which consist of tapping the finger or sliding it horizontally, vertically and diagonally on the touchscreen. Depending on the finger performing the gesture, a key press event is triggered to perform a text entry. Since the idea is basically a keyboard based on gestures, we call it **Gestyboard**. The core idea of this concept is to bind and map the key to the finger gesture following the 10-finger-system (sometimes also called the touch typing system). This is described in detail in Section “Concept”. We implemented a first prototype and evaluated it with 41 pupils who had been taught the 10-finger-system at school. We investigated among other things, the performance of our Gestyboard prototype by comparing it to different existing text input mechanisms: the classical hardware keyboard and the standard virtual touchscreen keyboard. Even though the performance of both the hardware keyboard and the virtual touchscreen keyboard could not be reached, the evaluation results show that there is a high potential for the acceptance of the Gestyboard by the users. The performance could also be increased by improving the prototype and offering more training to the users. Our user test is described in Section “Evaluation”, the results and discussion are then presented in Section “Results”. The conclusion and the next steps for the enhancement of the Gestyboard are discussed in Section “Conclusion And Future Work”.

## REQUIREMENTS

The Gestyboard concept is based on the following requirements:

- (R1) No additional hardware, just the multi-touchscreen
- (R2) No dictionary / No word prediction / No database
- (R3) Enable fast typing - comparable to the classical hardware keyboard
- (R4) Enable typing without missing the keys
- (R5) Enable blind typing

## RELATED WORK

Many research projects have been trying to solve the issues introduced in Section “Requirements”. Since the major source of the problem is the missing tactile feedback on touchscreens,

many proposed solutions have enhanced the touchscreen with tactile feedback. Hoggan et al. investigated the effectiveness of tactile feedback for touchscreen smartphones [4] with a finger-based text input system. According to their results, the performance of the finger-based text entry can be significantly improved by providing tactile feedback. To simulate the tactile feedback, the built-in smartphone vibration actuator has been used. In 2010 Bau et al. introduced a way to provide tactile feedback for larger devices, like multi-touch tables: the so called Tesla Touch technology, based on the electrovibration principle[2].

Even though the tactile feedback improves the performance of the text entry, in our opinion the speed and the accuracy of the classical keyboard can not be attained as long as the tactile feedback does not allow the user to clearly distinguish the edges between the keys by sensing them without activating them. Additionally, large multi-touchscreen displays that provide tactile feedback are infrequent and expensive. Therefore, an alternative is to develop completely new concepts for touchscreen text input systems with no need for tactile feedback. Kölsch et al. [5] conducted a survey on virtual keyboards in 2002 and classified them into four main categories: 1.) Speech Recognition, 2.) Handwriting recognition, 3.) Sign Languages and 4.) Touch Typing. The last category encapsulates all touch based text input systems; therefore, the virtual keyboard is a sub-category of Touch Typing. However, Kölsch et al. did not distinguish between touch typing with or without requiring tactile feedback. We introduce in this work a promising multi-touchscreen keyboard, called Gestyboard, that is part of Kölsch category 4: Touch Typing. The Gestyboard was designed in such a way that it does not rely on tactile feedback.

The Section “Non-Touch Concepts” introduces one related work which is not based on a touchscreen, whereas Section “Touchscreen Concepts” presents touchscreen related text-input-concepts.

## Non-Touch Concepts

### *A virtual system based on multi level feature matching*

There are several projects focusing on hand tracking algorithms to realize a virtual keyboard. For example Du et al. introduced a new hand-tracking algorithm in 2008 which can generate high precision matching results for hand typing postures in near real-time [3]. Since their focus was on the matching algorithm to improve the detection speed and the accuracy, there are no results regarding the performance of the virtual keyboard itself. This hand tracking method could be used in our Gestyboard concept in future work as an enhancement for the touchscreen finger detection algorithm to increase the robustness of our solution. However, this would require additional optical tracking hardware which does not adhere to our first requirement (R1).

## Touchscreen Concepts

Firstly, Section “Virtual Touchscreen Keyboard” describes the standard touch keyboard. Secondly, two examples of touchscreen keyboards are presented in Sections “Microsoft’s Split Keyboard” and “LiquidKeyboard”.



### Virtual Touchscreen Keyboard

The most common way to develop a virtual keyboard on a touchscreen is to clone the layout of the classical hardware keyboard (QWERTY) and to implement the key pressing detection with the soft buttons. Since cloning the well known hardware keyboard is intuitive for the user, this concept is being used in many of today's operating systems. Windows 7, for example, delivers such a touch keyboard for touchscreen devices. In fact, we used the Windows 7 touchscreen keyboard in our evaluation to be able to compare the results of Gestyboard with it. Recently, even small screen device operating systems offer such touchscreen keyboards.

### A Virtual Touchscreen Keyboard Adapted to the Users Hand - 2008

In his bachelor thesis, Patrick Bader developed a touchscreen keyboard which can be adapted to the user's hands [1]. The layout and positioning of the keys were designed like a human hand, as shown in Figure 1. The goal was to reach the performance of the real hardware keyboard. Unfortunately, no user oriented evaluation results could be found.

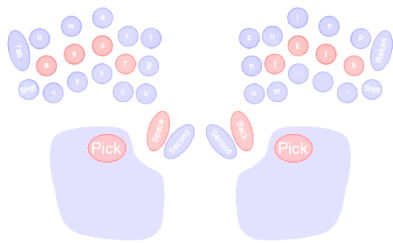


Figure 1. The layout of the touchscreen keyboard from Patrick Bader.

### Microsoft's Split Keyboard - 2009

Microsoft published the *US patent 2009/0237361 A1* in 2009 about a virtual touchscreen keyboard which automatically adapts to the user's hands [9]. To do so, the keyboard is split in two parts, one for the left hand and one for the right hand as shown in Figure 2. If one hand moves or rotates while it is still touching the surface of the touchscreen, the corresponding part of the keyboard follows it.

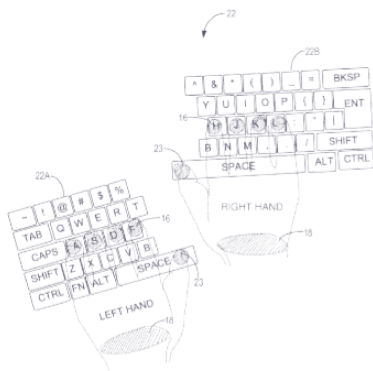


Figure 2. The layout of the touchscreen keyboard from Microsoft.

### LiquidKeyboard - 2011

Another virtual keyboard concept for touchscreens, the LiquidKeyboard, has been introduced by Sax et al. in 2011 [11], [12]. The keys of the virtual keyboard are placed at the position of the user's fingers when the display is touched. The standard QWERTY layout is used to define the position of the keys. When moving a finger on the screen, the assigned group of virtual keys follows it. In this way, the keyboard automatically adapts to the user's hand physiology. The key activation method is to tap the keys. Two versions of the LiquidKeyboard were developed and empirically tested. As of now, there are no user performance related results available, but this is planned for future research. Figure 3 shows the LiquidKeyboard when touched with 8 fingers.

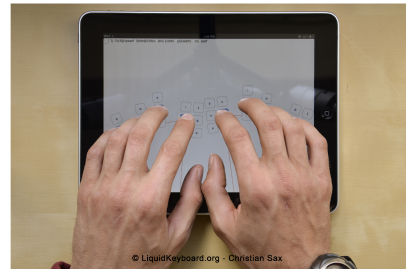


Figure 3. The LiquidKeyboard developed at the University of Technology Sydney (UTS)

## GESTYBOARD CONCEPT AND FIRST PROTOTYPE

### Activation of the Gestyboard

To start the text entry process, an activation state is needed initially for the Gestyboard. When the user touches the multitouch surface with all 10 fingers the keyboard is activated and a visual feedback is displayed on the touchscreen (see Fig. 8). The central row of keys ('a' 's' 'd' 'f' and 'j' 'k' 'l' ';') are placed at the position of the user's fingers with a QWERTZ layout keyboard and are activated by tapping them. By sliding a finger horizontally, vertically or diagonally, the user is able to select and activate the direct neighboring keys of the central ones. Hence, a group of keys is assigned to each finger (a horizontal central key plus two or five direct neighboring keys). In this first prototype, the Gestyboard does not update the displacement of the user's hand when it moves on the touchscreen. The adaptivity of the Gestyboard to the finger displacement is left out in this prototype, to focus on the Gestyboard text entry main feature and the design of finger gestures to successfully perform the 10-finger-system. Therefore, each finger gesture has to be uniquely mapped to a keystroke, which is the main feature inherited from the 10-finger-system as mentioned earlier. For simplicity, we are referencing the fingers as finger 1 to finger 10 (see Fig. 4) in the following.

### Keystroke Activation Gestures

Each finger can activate one and only one key group assigned to it, based on the 10-finger-system. Figure 5 shows the conceptual layout of the keys. When the keyboard is activated, the keys 'a' 's' 'd' 'f' are assigned to finger 1 to 4 and the keys 'j' 'k' 'l' ';' to fingers 7 to 10, respectively. A space key is assigned to fingers 5 and 6 (thumbs) and appears at



Figure 4. The numbering of the fingers to be more clear by referencing them.

their position in the visualization. It is important to mention that all these keys are not virtual buttons, in a sense that they are not activated by a finger touch. In fact, their only purpose is to visualize the QWERTZ layout to guide the users while typing if they do not master the 10-finger-system.

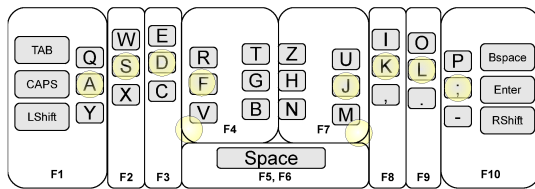


Figure 5. The conceptual layout of the Gestyboard. Fx : Finger x

#### The fingers' gestures for the horizontal central keys

Tapping with a finger always activates the central key assigned to that finger, whether or not the tap occurs within the corresponding square of the visual representation, e.g. a tap of finger 1 always activates the key 'a'. Even if the users tap on key 'q' with finger 1, key 'a' will still be activated because the tap represents a unique gesture for this finger. The same holds true for key 's' and finger 2, key 'd' and finger 3, and so forth.

#### The fingers' gestures for the neighboring keys : upper and lower keys

The upper buttons ('q' 'w' 'e' 'r' and 'u' 'i' 'o' 'p') are activated by performing a sliding-up gesture with the corresponding finger. For example, to activate the key 'q', the corresponding finger, finger 1 in this case, has to be slid upwards the screen. If the finger displacement exceeds a predefined threshold in the upper direction, the letter 'q' will be activated and hence typed. The same procedure is used for the activation of the lower keys ('y' 'x' 'c' 'v' and 'n' 'm' ';' '.'). Again, there is no need to physically hit the corresponding visual representation of the desired keys, the sliding gesture upwards or downwards with the associated finger defines a unique keystroke activation.

#### The fingers' gestures for the vertical central keys

The keys 't', 'g', 'b', and 'z', 'h', 'n' are a special case, since they add an extra gesture overhead to fingers 4 and 7. Finger 4 has to be slid to the right and finger 7 has to be slid to the left to activate the key 'g' and 'h' respectively. To activate the keys which are positioned diagonally, a diagonal finger sliding to the corresponding direction has to be performed.

*The fingers' gestures for Space, Enter, Backspace and Shift*  
 Performing a tap gesture with finger 5 or 6 (thumbs) activates the space key. The 'Enter' key is activated by a sliding gesture to the right with finger 10, while the 'Backspace' key is activated by sliding the same finger diagonally to the upper right. Lastly, the 'Shift' key can be activated with either finger 1 (diagonal sliding to the lower left) or finger 10 (diagonal sliding to the lower right). An overview and a sketch of all the gestures is given in Figure 6.

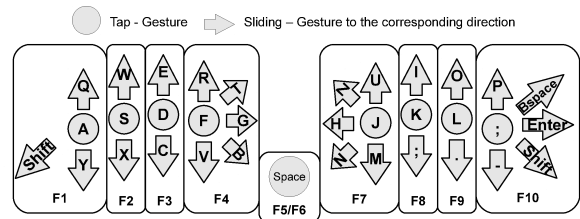


Figure 6. The overview of the gestures. Fx : Finger x.

#### Added value of the Gestyboard concept

The rigorous requirements that the Gestyboard follows, listed in Section "Requirements", are the main advantages of the concept. R1 is fulfilled, since no additional hardware like cameras or digital gloves is needed. Additional hardware is expensive, reduces flexibility, and in some cases requires additional effort for calibration. R2 is also fulfilled, since the Gestyboard is not based on any dictionary or database. We consider that an advantage, because the user should be able to type everything, even mixed languages or slang, as it is the case for the classical keyboard. Of course, text entry performance can be dramatically improved with a dictionary and word prediction. However, this is more appropriately done by the applications itself. R3 can not be verified with the current user evaluation results, since the users did not have as much training with the Gestyboard as they had with the classical keyboard. In order to attain high speeds with the Gestyboard, users need time to get familiar with the new finger gestures. We believe, that people can get used to these gestures, seeing that piano players are able to learn and perform much more complicated finger movements. R4 is however fulfilled by the concept itself. This is due to the unique finger gesture key mapping system. If the user for example wants to activate the key 'q', finger 1 has to be slid upwards. In the previous example, the upward movement does not need to be perfectly vertical, since there is no risk to activate another nearby key with the same gesture. Typing errors can of course still occur if the wrong finger is used, or by moving multiple fingers at once accidentally. Fulfilling R5 might be one of the most important advantages and an expected outcome of the 10 finger system: We expect that it is possible to blind type text if the user get used to the finger gestures. For the beginners, the keys visual representation is quite important to actually guide them while typing. We also expect that the longer the users use the Gestyboard in their daily life, the higher the chance that they do not rely on the Gestyboard keys visualization.

#### Special Challenges

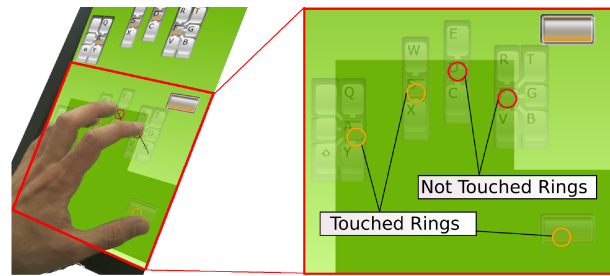
This novel approach also leads to some new special challenges. The users first have to understand the new input mechanism. To help them, we provide a visualization for the fingers gestures which always represent the current state of the Gestyboard. This current state representation is one of the 10 usability heuristics of Nielsen [10] that each UI should fulfill. Another challenge is the human fingers' ergonomics. Sometimes it is not easily possible to move one finger independently from the others, whether intentionally or not. When one finger is slid back to the center again, the other fingers of the same hand will also slightly deviate from their centered position. We are investigating potential solutions to this for future work.

### The first prototype of the Gestyboard

To get a first impression of the Gestyboard concept, prototype version 1.0 was developed using Microsoft Windows Presentation Foundation framework (WPF). In this version, the basic gesture detection algorithm was implemented through a state machine for each finger which determines whether a sliding or a tap gesture is performed. The direction of the gesture is not yet integrated in the state machine. Instead, a hit event is triggered once a key visual representation is touched as a way to determine the direction of the sliding gestures. A desired key activation with a sliding gesture occurs when the following conditions are satisfied: 1.) a sliding gesture is detected; and 2.) a hit event is triggered; and 3.) the correct finger id according to the 10-finger-system is used. If all three conditions are satisfied, then the desired unique finger gesture-to-key mapping is triggered. Although this implementation works similarly to the conceptual Gestyboard, there are some important differences which lead to some limitations in this version. As a consequence, requirement R5 - enabling blind typing - is not fully fulfilled, since it is still possible to not trigger a hit event by miss touching a virtual key while performing a sliding gesture. Another limitation can be found in the finger detection algorithm. Initially determining the finger's id can be simply done by numbering them from the left to the right. When a finger is released from the screen, its last position is stored. When the finger touches back the screen, its new position is compared with the previously stored one. If the distance between those 2 points is less than a specified offset, then the system considers this finger to be the one released earlier. Otherwise, the finger is simply ignored by the system. The users have to try to reposition their fingers within an offset from their initial position on the screen. Those offsets are visualized as rings and are at all times visible on the screen as long as the Gestyboard is active (see Fig. 7). These limitations do not conform with the concept and are derived from our willingness to test the first prototype and get early results to evaluate our concept. The next version will satisfy the whole concept and additionally include improvements drawn from the user evaluation described in Section Evaluation.

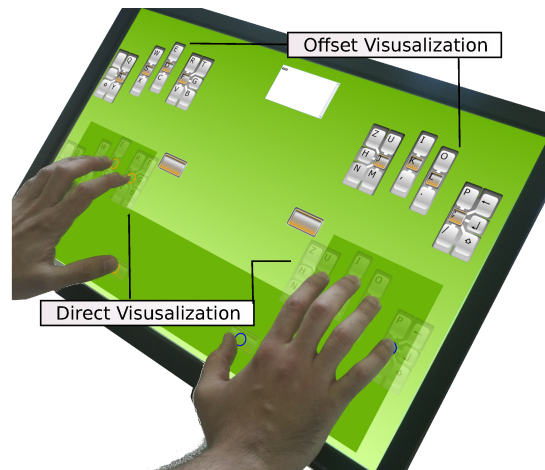
### The visualization of the prototype

The visualization of the Gestyboard included both: the direct key visualization and the offset visualization. The direct key visualization is placed directly below the users fingers



**Figure 7. Prototype 1.0 of the Gestyboard - The rings visualize the last fingers positions known by the system. Fingers 1,2 and 5 are touching the screen, while 3 and 4 are hovering above it.**

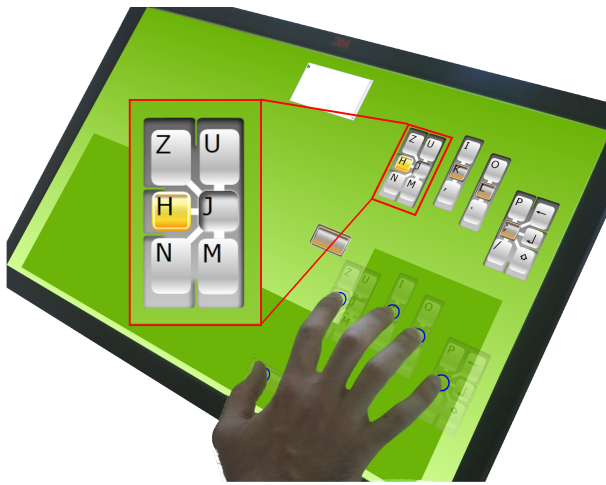
and might be more intuitive, since a one to one mapping is used here which requires less mental overload from the users compared to the offset visualization. The latter shows the same version of the Gestyboard shifted by a y-offset. In this case, the users have to map their finger movements to the deviated visualization of the keys, which might require more mental effort. Despite that fact, in the offset visualization the user's hands do not occlude the keys visualization, this is the main reason behind adding the offset to the visualization. We decided to provide both visualizations to the user to find out their favorite. To avoid overloading our system with too much visual information such as animations. The direct visualization is transparent ( $\alpha = 0.2$ ) which makes it less pronounced. Despite the transparency, both visualizations are rendering exactly the same thing; this is demonstrated in Figure 8.



**Figure 8. Two visualizations of the Gestyboard are provided to the user. Direct: Appears directly below the user's fingers. Offset: An additional visualization with a y-offset to 1.) avoid occlusion issues; and 2.) find out which visualization is preferred.**

To let the user know where the fingers are relative to the Gestyboard, a slider is displayed with each group of keys. The slider represents the current fingers position within the Gestyboard. Thus if, for instance, an 'h' is typed the slider is placed above the key visualization of the letter 'h'. However, the key label is still visible, since the slider is transparent (see Fig. 9).





**Figure 9.** The current position of each finger relative to the Gestyboard is visualized by a slider (see Letter 'H').

There is a slider for each finger. Consequently, multiple sliders can be moved simultaneously, similar to pressing multiple keys on the classic keyboard. The letters are also typed in the same order as they are activated by the user. The only difference to the classical keyboard is that there is no auto-repeat feature in this version of the prototype. This will be added in future work.

## EVALUATION

We performed a within-subject evaluation to compare our Gestyboard (G) with both a virtual touchscreen keyboard (T) and a classical hardware keyboard (C).

### Participants

We evaluated our first Gestyboard prototype with 41 pupils from a junior high school, 26 males and 15 females, all right-handed. All of them have been taught the 10-finger-system in school on a standard computer with a classical hardware keyboard for at least 2 years. It was important that all our participants knew how to type with the 10-finger-system as this is an essential prerequisite skill to use the Gestyboard. We evaluated pupils from different class levels to see if there was any difference in the performance between beginners and more experienced typists. The youngest among them had been taught typing at school for 2 years, these were 21 pupils aged between 12 and 14 (group 1). The school requires that the pupils of this level have a typing speed of 80 characters per minute. Additionally, 6 pupils had a learning experience of 3 years (age 14-15) with a typing speed of 90 characters per minute (group 2) and the last 14 had an experience of 4 years (age 14-17) with a typing speed of 100 characters per minute (group 3). We selected participants with various typing skills in order to have a large variety in terms of experience.

Of the 41 participating pupils, 29 owned a touch device such as a smart phone. Any kind of touch devices are used at least once a week by 25 pupils, 9 pupils use these devices rarely (less than once in a week) and 7 have never used any touch

device at all. 29 pupils use computer daily and 34 type a text more than once a week. Text messages are written by 21 of the pupils on their smart phones at least once a day, 4 have never written any text messages.

### Apparatus

The tests were conducted during normal class hours at the school in a separate room next to the class room. We had six different stations for six different tasks, these included three typing stations in order to compare the three different keyboards. The hardware keyboard station consisted of a normal computer monitor and a standard hardware keyboard with German Layout (position of 'Y' and 'Z' are swapped). For the virtual touchscreen keyboard we used a ruggedized tablet PC with dual touch input. The Windows 7 onscreen keyboard was used for text input on the tablet PC. The Gestyboard was shown on a 3M Multitouch Monitor with a size of 22" and a multitouch capability of more than 20 fingers at a time. As a software we used Tipp10<sup>3</sup>, which is an open source typing tutor. We created our own typing lessons in Tipp10 - one for trying out the Gestyboard which consisted of two sentences and another one for the actual evaluation. The latter lesson consisted of seven English sentences and an English pangram, which contain every letter of the alphabet. The complete text sums up to 248 characters. The seven sentences were chosen from a predefined phrase set for evaluating text entry techniques by MacKenzie et al.[8]. The frequency of each letter in each sentence represents the frequency of the letter in the English language in general. Although the current prototype allows use of the shift-key by combining the shift-key gesture with any other key gesture, uppercase letters have been removed from the sentences in order to focus on the gestures which can be activated with just one single finger-gesture. To type uppercase letters, the user would have to perform both the gesture for the shift-key as well as the gesture for the desired letter. This feature was kept out of the first evaluation to avoid the additional complexity, and it is planned to investigate two-level-gestures in details in future work.

Tipp10 presents the text to type in a continuous text ticker. The speed of the ticker is controlled through typing. The ticker stops when a wrong character is pressed and proceeds again by pressing the correct letter. We measured the following data with Tipp10: duration to type the eight sentences, overall number of errors and the number of errors per character.

### Procedure

In order to make the evaluation as efficient as possible, we tested three pupils at once - each supervised by a different instructor. The following six different tasks were executed by every participant:

1. Introduction + demographic data: We introduced the participants to the evaluation environment and collected demographic data

<sup>3</sup><http://www.tipp10.com/>

2. Ergonomic test: A test to evaluate the user's ability to move single fingers separately
3. Text entry with the standard hardware keyboard
4. Text entry with the virtual touchscreen keyboard
5. Learning phase of the Gestyboard with two practice phrases, followed by text entry with Gestyboard
6. Interview and Feedback: Qualitative feedback about the preferences and the problems encountered with the different keyboards was collected through an interview after the test

Each task was explained beforehand to the participants. We demonstrated first how to type with Gestyboard. A learning phase followed, where they wrote two practice sentences.

In summary, the design of the study was as follows:  
 3 techniques x 248 characters = 744 characters per person;  
 744 characters per person x 41 persons = 30504 characters entered in total;  
 additionally 2 practice phrases with 62 characters x 41 persons: 2542 characters

### Hypotheses

Our hypotheses before the evaluation were the following:

- (H1) The hardware Keyboard is faster in terms of typing speed than the virtual touchscreen keyboard and the virtual touchscreen keyboard is faster than Gestyboard. We suggest this due to the fact that the hardware keyboard is the one with which most people are familiar, the virtual touchscreen keyboard has the disadvantage of lacking haptic feedback and Gestyboard is a completely new input mechanism.
- (H2) 'a', 's', 'd', 'f', 'j', 'k', 'l' are the characters with the lowest error rate. We predict this because the fingers are initially placed on these characters in the initial typing position.
- (H3) The characters 'z', 'h', 'n', 't', 'g', 'b' are most error-prone due to their position on the keyboard. Only both index fingers (finger 4 and 7) are in charge of typing these 6 letters; each index finger has to move vertically, horizontally and diagonally to type these letters.
- (H4) The letters 'n' and 'b' have a higher error rate than 't' and 'z', because the index finger has to perform a diagonal movement down to type 'n' and 'b'.

### RESULTS

The typing speed, the total error rate and the error rate per character were calculated using the values which were measured with Tipp10. A correlation analysis was performed in order to find out if for example, the typing speed of the touch keyboard correlates with the typing speed of the Gestyboard. Our results of the the evaluations are presented below.

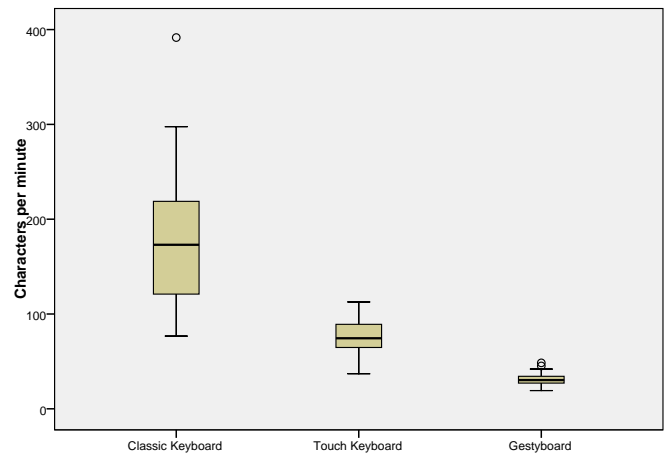


Figure 10. Characters per minute with 3 different keyboards. Whiskers represent 95% confidence interval.

### Typing Speed

We calculated the typing speed as characters per minute (cpm) through this formula:  $\frac{\text{total characters}}{\text{total time in minutes}}$

Our results show that all of the pupils had a faster typing speed with the classical hardware keyboard than required from the school (group 1 had an average typing speed of 137.66 cpm (required is 80 cpm), group 2: 168.84 cpm (required is 90 cpm), group 3: 225.47 cpm (required is 100 cpm)). The overall average text entry speed is: 174.41 cpm with a  $STD = 68.75$  for the classical hardware keyboard, 75.79 cpm with  $STD = 17, 51$  for the virtual touchscreen keyboard and 31.03 cpm with  $STD = 6.81$  for Gestyboard, see also Figure 10.

The classical hardware keyboard is significantly faster than the virtual touchscreen keyboard with  $t(36) = 10.07, p < 0.01$  and the virtual touchscreen keyboard is significantly faster than Gestyboard with  $t(36) = 14.20, p < 0.01$ . Hence, H1 is verified.

### Total error rate

Errors are counted per character and not by the number of wrong inputs. If the user typed more than one wrong character after each other, it is counted as one error for the specific character. For example the correct sentence is:

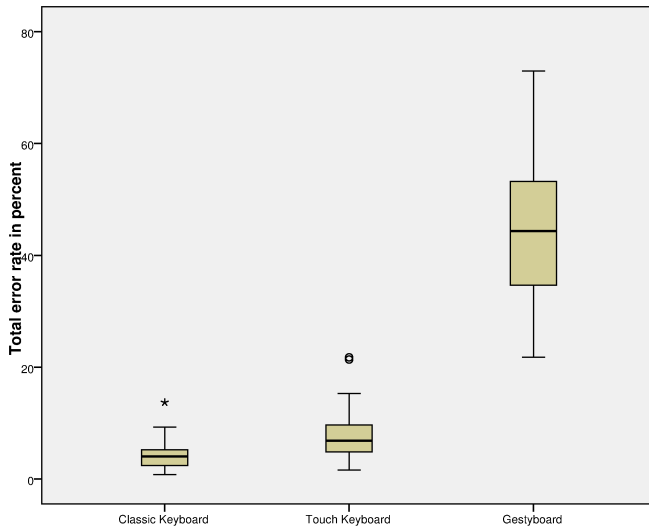
*see you later alligator*

the user typed:

*sedfe asyou latwer alligator*

This is in total 3 errors, 2 errors for letter 'e' and 1 error for letter 'y'. For the total error rate all errors of all characters are summed up. The total error rate in percentage is calculated as the following:  $\frac{\text{total errors}}{\text{total characters}} * 100$

Mean error rate was the following for the three keyboards: The classical hardware keyboard 4.32% with  $STD = 2.60$ , the virtual touchscreen keyboard 7.90% with  $STD = 4.88$  and Gestyboard 45.23% with  $STD = 13.75$ , see Figure 11. The classical hardware keyboard and the virtual touchscreen keyboard are very significantly different  $t(36) = -4.74$ ,

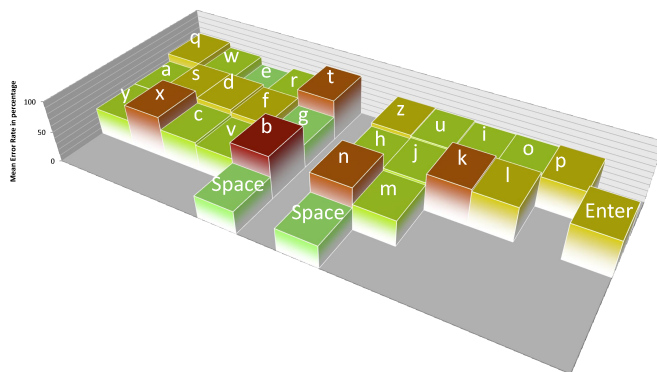


**Figure 11. Total error for 3 different keyboards. Whiskers represent 95% confidence interval.**

$p < 0.01$ , the same is also correct for the virtual touchscreen keyboard and Gestyboard  $t(36) = -15.56, p < 0.01$ . There was no significant difference however in the error rate nor the speed between the three different classes, which had unequal experience with typing.

### Error rate per character

In addition to the total error rate, the error rate per character was also analyzed. Figure 12 shows the mean values of the error rate in percentage over all participants. Each character is ordered as in the Gestyboard layout in three rows and two groups, for the left and the right hand. The height and the color encode the error rate of each character. The color coding goes from light green for a low error rate to a dark red for a high error rate.



**Figure 12. Mean error per character for Gestyboard input.**

The mean error rate for the upper row is 49.0%, for the middle row is 46.0% and for the lower row is 55.4%. What we can say about H2 is that not all of the horizontal central keys from the initial middle position ('a' 's' 'd' 'f' and 'j' 'k' 'l') are easier than the rest of the characters, especially 'k'

(60.25%) one of the direct tapping keys is nearly as error prone as 'enter' (59.61%) which requires a right sliding gesture of the little finger. This result is also represented in our statistics as we have no significant difference between the initial direct tapping characters and the neighboring sliding letters.

Concerning H3: The index finger letters, which are not directly under the finger ('z' 'h' 'n' 't' 'g' and 'b') have a mean error rate of 56.65%, whereas the rest of the characters have a mean error rate of 46.27%. A paired sample t-test showed values of  $t(5) = 2.5$  with  $p < 0.05$ . We have a significant difference between the two groups of letters.

The five toughest characters were 'b' (70.76%), 'n' (69.23%), 'x' (69.23%), 't' (66.74%), 'k' (60.25%). Least problems appeared for 'e' (33.72%) and 'space' (37.21%). The most problematic characters 'b' and 'n' are typed both through a diagonal down sliding movement. 't' (66.74%) and 'z' (50.0%) which are written through a diagonal up movement of the index finger have lower mean error rate, however the values are not significantly different from 'b' and 'n' as we suggested in H4.

### Correlation analysis

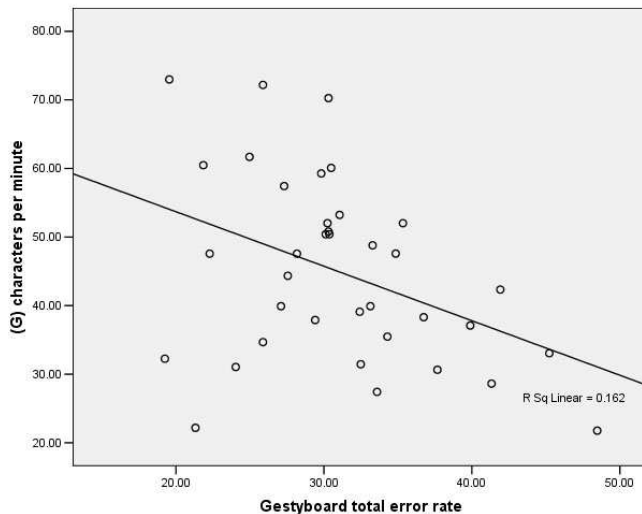
A correlation analysis and a following bivariate linear regression analysis showed that there is a significant linear dependence between the error rate and the characters per minute with the Gestyboard ( $p < 0.01$ ), with a regression line of  $y = 40.16 - 0.203x$ , see Figure 13. The correlation is a weak negative relationship. This means, that if the error rate increases, the typing speed falls. We suggest, that the users who had a lower error rate adopted quicker to the Gestyboard and could type faster than those who had problems with the usage of Gestyboard.

Pearson's correlation coefficient is  $r = 0.402$  and the coefficient of determination that will give some information about the goodness of fit of the model, shows a distribution of  $r^2 = 0.162$  or 16%. This means also that 84% of the variance in the diagram is influenced by other variables not part of the model. Hence, we run a multi regression analysis in order to find other influencing variables. We tested gender, class level, error rate of the touch keyboard, error rate of the classical hardware keyboard as well as the multitouch experience. However, none of them showed a significant difference. Neither had the usage of the mobile phone, keyboard or computer and also not the experience with the computer any significant influence.

### Qualitative Feedback

We asked the participants to express their suggestions for improvement, further usage, difficulties, preference, physical and mental demand as well as whether their knowledge of the 10-finger-system has some benefit or not. This interview was conducted after the evaluation with every participant individually.

### Improvements and difficulties



**Figure 13.** Scatter plot showing the correlation between error rate and characters per minute with the Gestyboard.

Several different categories were mentioned for improving the Gestyboard.

**Movement** The sliding gesture for the upper and the lower row as well as performing two different gestures, sliding and tapping, were mentioned frequently as being problematic. The movements were uncommon and therefore not that much liked. Especially sliding had the disadvantage that it is not always possible to move all fingers individually without moving other fingers too. This resulted in mistyping. Some of the pupils would therefore prefer tap movements instead of the combination of both movements.

Some subjects described that they had less problems with the movement after practicing it for a while. The pupils reported, that their motivation increased during the test to get even better after some more training. Interestingly, we got also some sporadic answers saying that the movement was not a problem at all.

During our test we had also the chance to speak to a teacher, who teaches the pupils typing. We got very valuable feedback concerning the two different movements. She explained that besides the standard process of typing with a classic keyboard, Gestyboard requires, that the typist has to differentiate for each letter if a sliding or tapping movement has to be performed. This adds additional mental load for typing with Gestyboard.

**Automatic reset to initial position** Furthermore, the pupils mentioned that it is cumbersome to push the sliders of Gestyboard upwards to type any key in the upper row or downwards to type any key in the lower row and then drag it back again to the initial position in the middle row. An automatic reset is preferred, which sets the slider automatically back to its initial position if it is released somewhere else than there.

**Difficult keys** It was also mentioned that some keys were

more difficult than others. Especially the movement of the little finger caused problems. This finger is also used for the Enter key. Furthermore, also the space key was mentioned as being problematic. Both keys appear very frequently in a text and we are investigating other specific gestures or placement in order to type them more easier.

Typical statements from the users were:

“Typing is strange, because with the classic keyboard one normally pulls the finger up.”

“...But the switch between tapping and sliding is not that good.”

“There is a ring under my finger, if this is away I need to click once more.”

“I think it is better, if it is just tapping like on a normal smartphone.”

#### *Physical demand*

We asked the users if typing was exhausting for their hands and if they felt any pain in their hands after the session. The answers did not show a clear direction. Some subjects reported their hands' fatigue, while others did not. After further analysis it was observed that, that more female users expressed their hands fatigue.

#### *Mental demand*

The pupils also described, that the mental demand was higher with Gestyboard than on a classical hardware keyboard and subjects had to think more. Primarily the searching for the letters was a reason for that. Although the pupils knew the 10-finger-system, it was difficult to get used to Gestyboard with its division into two parts for the left and right hand. It was often reported, that it was especially hard to use initially, but became easier with practice. In this category there were three times as many negative statements as positive ones.

#### *Benefit of 10 finger typing knowledge*

We got 3 times more positive than negative answers when we asked if they had any profit from knowing the 10-finger-system. Positive responses were 300% greater in number than negative when pupils were asked if they benefitted from knowing the 10-finger-system.

#### *Further usage and preference*

The majority of the participants would like to use the Gestyboard in future. A lot of them would like to train to get better in typing with it. There were only very few people who did not want to use Gestyboard any further and prefer a touch keyboard, because it was too slow or unfamiliar. The preference for the Gestyboard is visible in the answers as we got almost 150% more positive answers about the familiarization than negative.

Several typical statements from the pupils that show their opinion:

“If I would use it more, than I could write very quick.”

“It is funny to use and interesting.”

“There is a difference to the classic keyboard, but if I type longer I would get used to it.”

“Gestyboard makes more fun. I prefer it, if I had to chose.”

Despite several weaknesses Gestyboard shows high potential for the future with further improvements. If it adapts better to the user needs, it will reduce current disadvantages. We believe also that with further training and regular use Gestyboard can overcome its weak points.

### CONCLUSION AND FUTURE WORK

We present through this paper a novel text entry mechanism based on the 10-finger-system for multi-touch devices. Most virtual touchscreen keyboard performance suffers mainly from the lack of tactile feedback. For this reason, we designed unique gestures for all 10 fingers based on the QWERTZ layout. A unique finger gesture-to-key mapping system is developed to overcome the lack of tactile feedback on touchscreens. The gestures are constituted of taps and slidings. Taps activate the central row letters while slidings are used for the direct neighboring letters in the other rows. Since the gestures are based on the QWERTZ-layout, we expect the users acquainted with the 10-finger-system to easily adapt to Gestyboard and to be able to, after some training, perform the key gestures naturally and blind type. We implemented a first prototype and compared it to both the classical hardware and the virtual touchscreen keyboard in an extensive evaluation.

The results of our qualitative interviews with the test users show that most of our users liked the Gestyboard concept and would like to continue using it. Besides the training to improve the users' performance of the Gestyboard there are also some possibilities to improve our first prototype. Among other things, we plan to realize the full concept of the Gestyboard by changing the keystroke activation method from the hit-test to a pure gesture based one in the future to fully support blind typing. Another improvement will be enhancing the finger detection mechanism. By eliminating the need for retouching the virtual rings when fingers are released the speed of the text-entry will also be improved. To avoid the mental effort required by the need for differentiating between sliding or tapping for each letter we plan further to change the concept in a way, where just sliding gestures are used to perform a keystroke. Additionally, we plan to change the space key activation gesture to the following: tap the screen with both hands. The reason behind this is, that the thumbs can than be used for other keys which are not included in the concept yet, like CTRL and ALT. As a side effect, the users will automatically reset the fingers positions after typed each word. Finally, we plan to conduct two additional evaluations with the next version of the Gestyboard: 1.) a longterm evaluation to extract some results about learnibility; and 2.) an evaluation with no visual feedback to assess the Gestyboard blind typing abilities.

### ADDITIONAL AUTHORS

### REFERENCES

1. Bader, P. Entwicklung eines virtuellen Eingabegeräts für Multitouch Screens. Bachelor's thesis, Hochschule der Medien, Stuttgart, Germany, GE, Aug. 2008.
2. Bau, O., Poupyrev, I., Israr, A., and Harrison, C. Teslatouch: electrovibration for touch surfaces. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, ACM (2010), 283–292.
3. Du, H., and Charbon, E. A virtual keyboard system based on multi-level feature matching. In *Human System Interactions, 2008 Conference on*, 176–181.
4. Hoggan, E., Brewster, S., and Johnston, J. Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ACM (2008), 1573–1582.
5. Kölsch, M., and Turk, M. Keyboards without keyboards: A survey of virtual keyboards. In *Workshop on Sensing and Input for Media-centric Systems, Santa Barbara, CA*, Citeseer (2002).
6. Kushler, C. Gesture-based repetition of key activations on a virtual keyboard, Nov. 19 2010. US Patent App. 12/950,835.
7. Landauer, T. K., and Prabhu, P. V. *Handbook of Human-Computer Interaction*. Elsevier Science Inc., New York, NY, USA, 1998.
8. MacKenzie, I. S., and Soukoreff, R. W. Phrase sets for evaluating text entry techniques. In *CHI '03 extended abstracts on Human factors in computing systems*, CHI EA '03, ACM (New York, NY, USA, 2003), 754–755.
9. Mosby, T. J., and Wiswell, C. N. Virtual keyboard based activation and dismissal. US-Patent 2009/0237361, sep 2009.
10. Nielsen, J. *Usability engineering*. Morgan Kaufmann, 1993.
11. Sax, C., Lau, H., and Lawrence, E. Liquid keyboard., Aug. 2011.
12. Sax, C., Lau, H., and Lawrence, E. Liquidkeyboard: An ergonomic, adaptive qwerty keyboard for touchscreens and surfaces. In *ICDS 2011, The Fifth International Conference on Digital Society* (2011), 117–122.