TypeRight: a Keyboard with Tactile Error Prevention

Alexander Hoffmann
RWTH Aachen University
52056 Aachen, Germany
alex.hoffmann@rwth-aachen.de

Daniel Spelmezan
RWTH Aachen University
52056 Aachen, Germany
spelmezan@cs.rwth-aachen.de

Jan Borchers
RWTH Aachen University
52056 Aachen, Germany
borchers@cs.rwth-aachen.de

ABSTRACT

TYPERIGHT is a new tactile input device for text entry. It combines the advantages of tactile feedback with error prevention methods of word processors. TYPERIGHT extends the standard keyboard so that the resistance to press each key becomes dynamically adjustable through software. Before each keystroke, the resistance of keys that would lead to a typing error according to dictionary and grammar rules is increased momentarily to make them harder to press, thus avoiding typing errors rather than indicating them after the fact. Two user studies showed that TYPERIGHT decreases error correction rates by an average of 46%.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces—*Haptic I/O*

Author Keywords

Adapting input device, tactile feedback, haptic, text entry, error prevention.

INTRODUCTION

A human working with a computer perceives information from three input channels: visual, auditory, and haptic. Although sight and hearing are the primary sources of information [2], touch is also important. To give just a few examples: We need touch to evaluate if things are cold or hot, to estimate the stability of a glass we are holding without breaking it, or to feel the weight or texture of an object.

Haptic input devices provide the user with tactile feedback. In virtual reality [7], medical testing scenarios [5], training assistants [6], and games, tactile feedback is used to provide the user with a more authentic experience. While the user is operating the system, tactile feedback can indicate that input errors need to be corrected, or to convey additional information [1].

On software level, word processors also offer various error correction methods to prevent and correct typos. Error correction methods, such as spell checkers, intervene when the error was already made. Prevention methods, on the other hand, such as city selection menus in navigation systems, limit the user's interactions and are only useful for special applications with very constrained ontologies.

TYPERIGHT bridges the gap between tactile feedback and data entry error prevention. Each key provides tactile feedback to prevent errors during text entry, instead of reporting them after they have been made. The TYPERIGHT keyboard consists of keys with adjustable pressure sensitivity. Our blocking algorithms make those keys significantly harder to press that would lead to a typing error or misspelling.

We start by summarizing related work that addresses existing text entry error prevention and correction methods. We then present the first prototype system, which offered numerical input with TYPERIGHT technology, and report results from its evaluation. This initial pilot study influenced design and construction of our final system, a complete alphanumeric keyboard. We discuss the results of our final user study with the full TYPERIGHT keyboard, summarize insights gained from our experiments, and conclude with future directions for tactile error prevention.

RELATED WORK

TYPERIGHT combines a tactile input device with an integrated error prevention method. We provide an overview of the most relevant research in both domains.

Devices with Tactile Feedback

Scheibe et al. [7] present a tactile feedback system for finger-based interactions in virtual reality applications. The system consists of tracked thimbles for the fingers with thin shape memory alloy wires wound around each thimble. The wires can be shortened by slightly heating them up, which results in tactile feedback at the fingertips. Users preferred to work with the tactile system compared to a VR system without any feedback. To use this approach for preventing typing errors, the system would need to accurately locate finger positions. In case the user is about to press a wrong key, the tactile impulse would have to be triggered shortly before pressing the key. However, wearing thimbles does not work well in many typing scenarios.

The addition of tactile feedback to touchscreens [4] significantly improves finger-based text entry, resulting in typing performance close to that of real physical keyboards. Once

CHI 2009, April 3 - 9, 2009, Boston, MA, USA.

more, however, preventing typing errors is not an option considered or available in these systems.

Error Prevention Methods for Data Entry

Typing errors are addressed in one of three ways: *Prevention*, *live correction*, and *aftercare* [3]. *Prevention* informs users before making an error. *Live Correction* corrects him directly during writing. *Aftercare*, the most widely established correction method, marks possible errors for later correction, manually or with the help of a spell checker.

Long-established word processors like Microsoft Word, as well as novel systems like Apple's iPhone and Nuance's XT9 input method for mobile phones, adopt live correction algorithms. Typical examples include correcting ordering errors ("teh" becomes "the"), capitalization errors ("peter" becomes "Peter"), and even CapsLock errors ("pETER" becomes "Peter"). Their impact on usability is debatable; while many mistakes are caught this way, these mechanisms frequently annoy users enough to disable them.

Prevention is the strictest method. The *look-ahead* function when entering street or city names into in-car navigation systems is a popular example. It does not allow the user to make mistakes. However, this also means that input is limited to a given set of words and selections. This method clearly avoids errors, but the accuracy and usefulness of the system strongly depends on the completeness of its database.

For more flexible text input, the hard boundaries of *prevention* methods can be softened. A prominent recent example is, again, the iPhone: The software constantly scans its dictionary during text input. If a key would lead to a mismatch while its neighboring key would not, the neighboring key's active touch area is enlarged, and the touch area of the key that would lead to a mismatch is diminished. This approach has interesting parallels to the TYPERIGHT keyboard: TYPERIGHT uses physical resistance to lower the likelihood of mistakes, while the iPhone uses 2-D spatial characteristics — the size of each key in touch-sensor space.

Another approach to "soften" prevention are timeouts and delays: Consider the SET button on a digital watch, which is only activated after being pressed for a certain duration. This crude, yet long-established method, can also be applied at a much smaller timescale, as on the current Apple keyboard[1]. Its Caps Lock key only activates after a very short delay, presumably to decrease accidental activations. The disadvantages of any time-based interactions such as these are well-known, however: Timeouts impair usability because they are too long or too short in most situations; they take the control and feeling of flow away from the user; and they lead to invisible, hard-to-detect features.

Apart from delay mechanisms, such as the one used in the Apple Keyboard, none of the presented systems use hardware-based error correction methods. TYPERIGHT populates a new area of devices that deploy hardware-enabled tactile feedback to prevent typing errors.

[1]http://www.apple.com/keyboard/

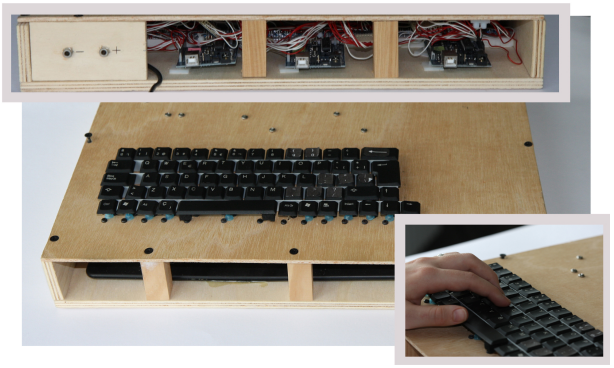

Figure 1. TYPERIGHT: Full-keyboard prototype.

HARDWARE PROTOTYPES

To explore the potential of tactile feedback keyboards for error prevention, we built two prototypes. The first prototype consisted of a numerical keypad with 12 keys (0...9, delete, and a free, programmable key). This system addressed numerical data entry and was used during an exploratory pilot study. The second prototype was a full alphanumerical keyboard (Fig. 1).

Both keyboards are similar in their electrical and mechanical design, and support controlling the resistance of each individual key from software. After considering various techniques such as bi-metals, hydraulic shock absorbers, or magnetorheological fluids to alter the pressure resistance of keys, we decided to work with small electromagnets, so-called solenoids.

We embedded a standard computer keyboard inside a wooden box. Solenoids were screwed to the lid of the box in such a way that, when the lid was placed onto the box, the lower pin extensions of the solenoids would touch the contact areas of the keyboard that register individual key presses. The upper pin extensions of the solenoids' plungers extend above the box and were covered with key caps. Fig. 2 illustrates the cross section of a modified key.

In their passive state, solenoids have no influence on the force necessary to press a key x (at $U_{Sx} = 0V$, 0.8 N yield normal pressure sensitivity). Thus, the resistance to perform a keystroke is similar to that of common keyboards. To block a key, we switch on the corresponding magnet, which then creates an additional, electromagnetic force that the user needs to overcome to press key x (at $U_{Sx} = 5V$, 4.9 N yield a blocked key).

To control the current to the solenoids, which adjusts the pressure sensitivity of the modified keys, three Arduino[2] microcontrollers connect the solenoids to a desktop computer. This computer runs blocking algorithms that modify the resistance of individual keys via software. An external power supply provided power for the magnets. MOSFET high-power semiconductor amplifiers receive the control signal

[2]http://www.arduino.cc/

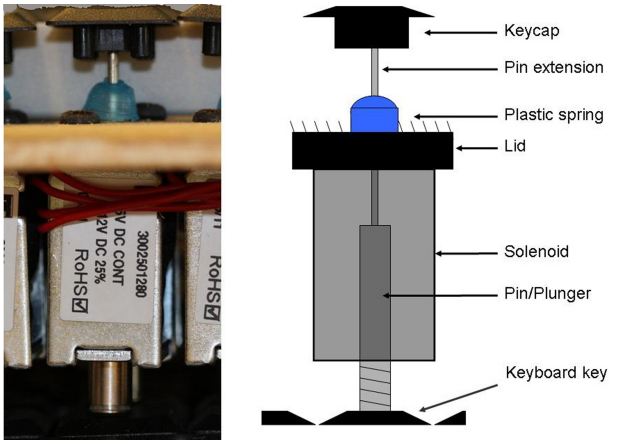

Figure 2. TYPERIGHT: Cross section of a key. The solenoid controls key resistance.

from the Arduinos and adjust the current coming to each individual solenoid from the supply.

A desktop application was implemented to conduct user studies. It displays a box that shows entered text. We decided to use Java Swing for the implementation, and Sun's javax.comm package implements the data communication between software and Arduinos. The application also runs three different tests to determine which keys have to be blocked: a dictionary test against a database of 46000 English words, a grammatical test (e.g., no capital letters within words are allowed), and a context test (e.g., no double spaces are allowed). After every key press, the system calculates which keys can potentially yield wrong words, and updates the pressure sensitivity of all keys accordingly. Keys that do not match the current word context are blocked. Keys that conform to the word context are released and can be pressed without extra force.

NUMERICAL KEYPAD PILOT STUDY

We first evaluated our 12-key TYPERIGHT keypad in a pilot study with 24 participants (20-31 years) to analyze if error rates and execution times during date entry can be decreased compared to a common keypad. A blocking algorithm controlled the syntactical correctness of entered dates. For example, typing "9" after typing "08.08." is invalid and cannot be entered, except when the user overcomes the force required to press blocked keys. No training session was included before the experiment. We performed video analysis of the experiment, and administered post-experiment questionnaires to test whether we reached our design goals.

This pilot study showed that 30% of the mistakes made without tactile feedback were grammatically incorrect and could have been avoided with the aid of TYPERIGHT. Only one participant did not agree that TYPERIGHT helped avoiding errors. In date entry applications, TYPERIGHT lowered the task completion time of some users by as much as 50%. More details about the design process and the evaluation can be found in [3]. Motivated by these results, we expanded the keypad to a fully functional keyboard.

FINAL EVALUATION WITH FULL KEYBOARD

This study aimed at comparing typing performance on a full alphanumeric keyboard under two conditions: (1) graphical highlighting of mistyped words, (2) tactile error prevention feedback. We hypothesized that text entry with TYPERIGHT, which provides tactile feedback, increases typing speed and decreases error rates compared to traditional methods that highlight mistyped words. We designed a user study to compare task performance times, error and correction rates between these two text entry conditions.

Experimental Setup

The TYPERIGHT keyboard was placed on a table in front of a 14" LCD screen. All key presses were logged with their timestamps. Twelve users aged 23-37 participated. They were given a short introduction explaining the functionality and the idea behind the system. None of the participants had previous experience with TYPERIGHT. No training session was included before the experiment.

Study Design

The participants' task was to faultlessly copy two handwritten texts of 140 words each. For one text, participants received tactile feedback using TYPERIGHT. For the other text, participants received graphical feedback by highlighting mistyped words in yellow (no keys were blocked). The dictionary comprised 46000 words. The two texts were always presented in the same order. The order of presentation of the two feedback conditions was counterbalanced across subjects.

Results

We found that fewer corrections are required with tactile feedback, as compared to graphical feedback ($p < 0.001$, dependent t-test). On average, the number of backspace key presses was reduced by 46% in the tactile feedback condition.

We also found that TYPERIGHT significantly prevents typing errors. Tactile feedback reduced the number of mistyped letters by 87% ($p < 0.001$, dependent t-test). Tactile feedback forced users to reconsider their doing, thereby preventing them from typing wrong letters. On average, the number of mistyped words that were not in the dictionary was about eight times higher with graphical feedback compared to the condition with tactile feedback (Fig. 3).

Average execution times were similar in both conditions (522 s with tactile feedback vs. 520 s with graphical feedback). We were not able to prove the expected time benefit of the tactile system with this experiment, although our observations suggest that trained users will be faster with tactile feedback than without. Users quickly adapted to the functionality of the keyboard. Questionnaires confirmed that 75% of participants did not consider TYPERIGHT to be a "big changeover compared to typing on a standard keyboard".

Counter-balancing the test conditions, as we did in this study, did not avoid learning effects. Participants' typing

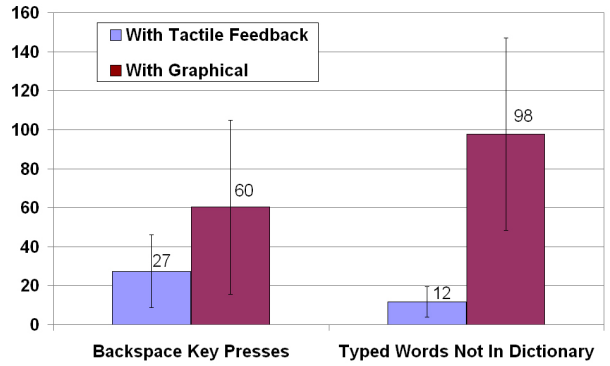

Figure 3. Evaluation of Errors: Comparison between the number of backspace key presses and typed words that are not in the dictionary during text entry with graphical and tactile feedback support.

speed increased the more they typed. Task completion times for typing the 2nd text was on average 111 seconds (25%) faster than for the first text, independent of the feedback condition ($p < 0.01$, dependent t-test). It might also be that the second text was easier to type than the first. To avoid these effects, future studies should include practice sessions and control for task difficulty.

LONG-TERM STUDY

One member of our team (aged 28) used the system over the course of three months. Unlike the novice subjects in the previous study, this expert learned how to react to possible tactile feedback, i.e., he was not surprised about changing key resistance. Moreover, he did not try to overcome a key's resistance when this key was blocked.

After the training period, we had the expert perform the same tasks as the study described previously, starting with the tactile condition. The results were consistent with our assumption that TYPERIGHT reduces typing time in the long run. The execution time with the first text was 10% faster than with the second text with graphical feedback. With tactile feedback activated, 16 corrections were necessary, compared to 23 corrections with graphical feedback (a 44% increase). With graphical feedback, the user typed 78 words that were not part of the dictionary, compared to zero(!) words with tactile feedback on the first text. Obviously these findings need to be confirmed by a study involving several expert users.

CONCLUSION AND FUTURE WORK

This case study described the development and evaluation of TYPERIGHT, a tactile feedback keyboard that supports typists by preventing typing errors. Inexperienced people who mostly look at the keyboard while typing do not discover mistakes in time. TYPERIGHT provides immediate feedback. The core concept is to increase the force required to press keys that would lead to typing mistakes. We believe that this promising technique will reduce beginners' mistakes and benefit novices in touch-typing.

Two user studies, one with a numerical keypad and one with a full keyboard prototype, suggested that TYPERIGHT prevents about 87% of typos in the form of words unknown to a dictionary. Furthermore, the use of the backspace key decreased by 46% on average. For a user trained on the TYPERIGHT system, typing speed increased due to tactile feedback: This user learned to react to tactile "messages" of the system, and saved time that would otherwise be required for corrections.

More than half of the study participants agreed, and the rest strongly agreed, that blocked keys made them aware of possible typos and helped to prevent errors. Nobody agreed or strongly agreed that the changing key resistance was disturbing. Further improvements are necessary to reduce the noise level, however, as the sound of magnets switching on and off was considered to be distracting. The question whether changing key resistance interrupted our users' workflow was not answered conclusively.

These results are only preliminary and more extensive studies need to be done. TYPERIGHT also needs to be compared to auto-correction and predictive text entry strategies. Moreover, common text-entry evaluation methods, standard metrics that better assess typing performance, and a comparison to existing user studies addressing text-entry on mobile phones should be considered. We plan to further extend the ideas behind TYPERIGHT to other data entry tasks and devices.

ACKNOWLEDGMENTS

This work was funded in part by the German B-IT Foundation.

REFERENCES

1. S. Brewster and L. M. Brown. Tactons: structured tactile messages for non-visual information display. *Proc. Australasian user interface*, 15–23, Darlinghurst, Australia, 2004. Australian Computer Society, Inc.

2. A. Dix, J. Finlay, G. Abowd, and R. Beale. *Human-computer interaction*. Prentice Hall, New York, 1993.

3. A. Hoffmann. Haptic keyboard prototype for data entry. Master's thesis, RWTH Aachen, University, 2008.

4. E. Hoggan, S. A. Brewster, and J. Johnston. Investigating the effectiveness of tactile feedback for mobile touchscreens. *Proc. CHI'08*, 1573–1582. ACM.

5. S.-Y. Kim, J. Park, and D.-S. Kwon. Palpation simulator for laparoscopic surgery with haptic feedback. *Proc. Biomedical Engineering*,478–482, 2004.

6. A. Nakamura, S. Tabata, T. Ueda, S. Kiyofuji, and Y. Kuno. Multimodal presentation method for a dance training system. *Ext. Abstracts CHI 2005*, 1685–1688.

7. R. Scheibe, M. Moehring, and B. Froehlich. Tactile feedback at the finger tips for improved direct interaction in immersive environments. *3dui*, 2007, IEEE Computer Society.