Efficient Typing on a Visually Occluded Physical Keyboard

James Walker, Bochao Li, Keith Vertanen, Scott Kuhl
Michigan Technological University
{jwwalker | bochaol | vertanen | kuhl}@mtu.edu

ABSTRACT
The rise of affordable head-mounted displays (HMDs) has raised questions about how to best design user interfaces for this technology. This paper focuses on the use of HMDs for home and office applications that require substantial text input. A physical keyboard is a familiar and effective text input device in normal desktop computing. But without additional camera technology, an HMD occludes all visual feedback about a user's hand position over the keyboard. We describe a system that assists HMD users in typing on a physical keyboard. Our system has a virtual keyboard assistant that provides visual feedback inside the HMD about a user's actions on the physical keyboard. It also provides powerful automatic correction of typing errors by extending a state-of-the-art touchscreen decoder. In a study with 24 participants, we found our virtual keyboard assistant enabled users to type more accurately on a visually-occluded keyboard. We found users wearing an HMD could type at over 40 words-per-minute while obtaining an error rate of less than 5%.

ACM Classification Keywords
H.5.2 Information interfaces and presentation: Input devices and strategies

Author Keywords
Physical keyboard; text entry; decoder; head-mounted display

INTRODUCTION
In recent years, affordable head-mounted displays (HMDs) have entered the consumer market for the first time. Thus far, the main focus has been on entertainment applications such as games. But as HMDs become more common, we believe there will be a need to support productivity applications such as email and messaging. These applications often require substantial text input. However, if users wish to type extensively while wearing HMDs, we must overcome the challenges posed by the fact that users cannot see the real world while wearing many models of HMDs. We also note that, due to the prevalence, familiarity, and efficiency of QWERTY mechanical keyboards, many users may wish to continue using their desktop keyboards while wearing an HMD.

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
CHI 2017, May 06-11, 2017, Denver, CO, USA
©2017 ACM. ISBN 978-1-4503-4655-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025783

Figure 1. User typing on a keyboard while wearing an Oculus Rift HMD.

In prior work [7], we developed a system that facilitated the use of a physical keyboard while wearing an HMD. In this paper we extend our system to include an on-screen virtual keyboard that provides live feedback of the user's typing, as well as modifications to the decoder to improve error correction. We show our solution significantly reduced participants' typing error rates, suggesting that software-only solutions can help overcome barriers to using physical input devices while wearing an HMD.

Related Work
Relatively little work has implemented or compared text entry systems for HMDs. Bowman et al. [1] compared several methods, including a one-handed chord keyboard, speech recognition using a human instead of software, and a virtual keyboard controlled by a tablet and pen. They found that none of these approaches produced high levels of performance or usability.

The augmented reality system ARKB displays a holographic keyboard on a see-through HMD [3]. ARKB tracks a user's hand position to monitor when a user's fingers touch the keyboard. A more recent example is the Microsoft HoloLens keyboard in which a user's head position controls a pointer on a holographic keyboard with keys selected via a hand gesture. Yi et al. [8] developed ATK, a system that uses 3D hand tracking to enable mid-air typing. Users were able to type in mid-air with high accuracy, but at slower rates than what is typically achievable with a physical keyboard.

Another way to facilitate real-world interaction while wearing an HMD is mixed reality. In mixed reality, portions of the real world are superimposed over the virtual environment. McGill et al. [4] found that users' unassisted typing performance while wearing HMDs was significantly reduced, whereas blending real and virtual reality brought performance closer to baseline. In a study by Budhiraja et al. [2], users took drinks from a cup while interacting with a virtual environment. Users reported they preferred mixed reality solutions compared to removing

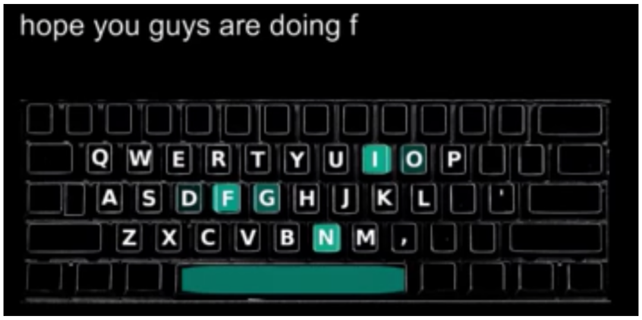

Figure 2. The virtual keyboard assistant as shown in the HMD.

the HMD to interact with the real world. However, such mixed reality methods add substantial software and hardware complexity, and poor imagery or self-occlusion can hamper effective user feedback.

Recently we conducted a pilot study in which five users typed normally, typed on a keyboard occluded by a cardboard cover, and typed while wearing an HMD [7]. No visual feedback was provided to users during typing. Autocorrection was performed on a user's sentence via the VelociTap decoder [6]. VelociTap takes a sequence of noisy touch locations and searches for the most probable text. We modified VelociTap to operate on physical keyboard input. In our pilot study, users were slower and less accurate when the keyboard was occluded. Further, users performed worse when their vision was occluded by an HMD compared to when it was occluded by a cover. VelociTap corrected most typing errors in all conditions.

We build on our previous work, further optimizing VelociTap for physical keyboard input. We address the lack of visual feedback during typing by adding a virtual keyboard assistant. Our on-screen virtual keyboard provides high-contrast visual feedback about which keys are being pressed and can be made arbitrarily large.

SYSTEM DESCRIPTION

The system consists of three components: a keyboard client, a recognition server, and a display server. These components communicate via TCP and can be run on a single computer or across multiple computers. The keyboard client displays the stimuli phrases, logs keyboard events, and sends those events to both the recognition and display servers. The recognition server infers the most likely sentence based on the noisy keystroke data for a sentence. Once decoding completes, the recognition server sends its result to the display server.

The display server renders the text and optionally the virtual keyboard assistant. When the keyboard client detects a keypress on the physical keyboard, it forwards it to the display server which lights up the corresponding key on the virtual keyboard. The glow gradually fades to black over half a second. The gradual fading allows users to see not only the last key hit, but other recently hit keys (see Figure 2). The virtual keyboard has labels for the letters, apostrophe, and comma. Other unlabeled key outlines (e.g. number keys, shift keys) are also shown and these keys light up if pressed. The rest of the scene is black. The display server outputs the text and virtual keyboard to either a desktop monitor or to an HMD.

Decoder

In this work, we improve on the physical keyboard version of the VelociTap decoder. We repeat details from [7] and [6] for clarity. VelociTap searches for the most probable hypothesis given a noisy sequence of input observations. On a touchscreen, observations are the x- and y-locations of taps on the screen. However, since this work uses a physical keyboard with discrete keys, each keypress instead results in an observation at the center of the pressed key. The keyboard was measured to identify the location of the center of each key.

During decoding, an observation can generate a key's letter based on the probability under a two-dimensional Gaussian centered at the key. Two parameters adjust the x- and y-variances of the Gaussians and all letters share the same x- and y-variances. Although keypresses result in an observation, some observations can be ignored depending on a configurable deletion penalty—causing no output character to be generated. Extra characters of output can also be generated without consuming an observation by paying an insertion penalty.

VelociTap uses both a character and a word language model. A character probability is assessed after a letter is produced (including space). A word probability is assessed on a space or the end of a sentence. The character and word language models are combined with the keyboard model probabilities via two separate scale factors. If a word is not in the word language model, an out-of-vocabulary penalty is assessed. We used a 12-gram character model trained with Witten-Bell discounting (2.2 GiB on disk) and a 4-gram word model trained with modified Kneser-Ney smoothing (3.8 GiB on disk). Both were trained on billions of words of Twitter, usenet, blog, social media, and movie subtitle data. VelociTap uses beam pruning to control the speed-accuracy tradeoff of its search.

We conjectured that transpositions are a common error type on bimanual keyboard typing. We modified VelociTap to explicitly model transposition by allowing adjacent observations to be swapped instead of requiring a multi-step process of deleting a character, inserting a new one, deleting the next character, and inserting another new one. Swapping observations incurred a new transposition penalty. All parameters of VelociTap were optimized with respect to pilot data obtained from [7] and from data recorded by the authors.

STUDY

We designed a study to test two hypotheses: 1) When users cannot see the keyboard, the virtual keyboard with live feedback improves typing performance, and 2) Users' typing performance would be worse while wearing an HMD than while the keyboard was merely occluded. To test this second hypothesis, we included non-HMD conditions in which we placed a physical cover over the top of the keyboard.

Our study had two independent variables, whether the virtual keyboard was shown, and whether visual feedback was via an HMD or via a desktop monitor. This resulted in four within-subject conditions. In two desktop display conditions, participants typed on a keyboard that was occluded by a cover either with the virtual keyboard assistant (DESKTOPASSISTANT) or without the assistant (DESKTOP). In two HMD conditions,

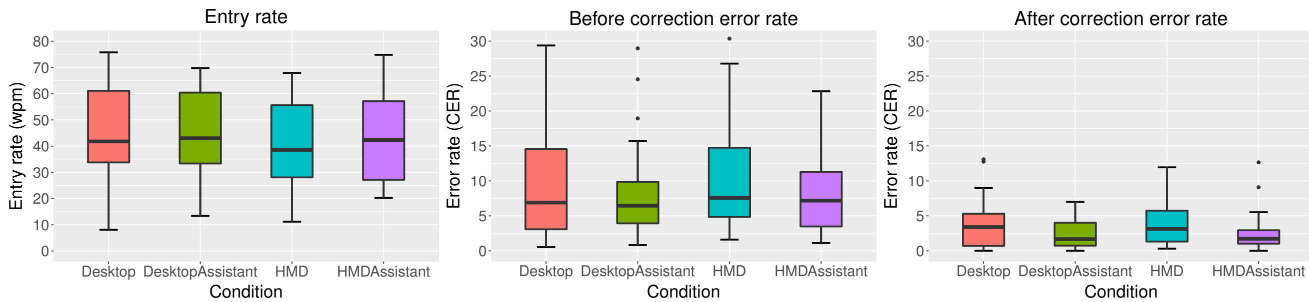

Figure 3. Participants' average entry rate (left), error rate before automatic correction (center), and error rate after automatic correction (right).

Condition	Entry rate (wpm)	Before correction error rate (CER)	After correction error rate (CER)
DESKTOP	44.7 ± 18.6 [8.1, 75.8]	11.2 ± 11.0 [0.5, 40.9]	3.9 ± 3.7 [0.0, 13.1]
DESKTOPASSISTANT	44.7 ± 16.3 [13.4, 69.8]	8.3 ± 7.3 [0.8, 29.0]	2.6 ± 2.2 [0.0, 7.0]
HMD	41.2 ± 17.5 [11.2, 67.9]	11.8 ± 11.4 [1.6, 49.6]	4.0 ± 3.2 [0.3, 11.9]
HMDASSISTANT	43.7 ± 17.0 [20.2, 74.8]	8.4 ± 6.2 [1.1, 22.8]	2.6 ± 2.9 [0.0, 12.7]

Table 1. Participants' average performance in each condition in the study. Results are formatted as: mean ± sd [min, max].

participants wore an HMD and typed with the virtual keyboard assistant (**HMDASSISTANT**) or without the assistant (**HMD**).

We recruited 24 participants via convenience sampling (7 female, 19 touch typists, ages 19–28). Participants received either course credit or $10. Participants were seated at a desk in a quiet office and typed on a Dell SK-8115 keyboard. In the covered conditions, visual feedback was on a 24 inch LCD monitor (1920 × 1080 resolution) positioned about two feet in front of the participant. In the HMD conditions, participants used an Oculus Rift DK2 HMD (resolution of 960 × 1080 per eye). We trained participants how to adjust the HMD.

Participants first completed a practice session where they typed 20 phrases. The goal of the practice phase was to allow participants to gain experience with the occluder that covered the keyboard and their hands, to gain experience wearing and looking through the HMD, and to become familiar with the virtual keyboard assistant. After this practice session, participants completed the conditions in counterbalanced order.

In all conditions, participants were shown 20 random memorable phrases from the Enron mobile data set [5]. Phrases contained only the letters A–Z and apostrophe. Participants were told to memorize each phrase before starting to type. As soon participants began typing, the phrase disappeared and was replaced by the literal characters typed (Figure 2).

In this study we disabled the backspace key. While VelociTap can handle backspaces either deterministically or probabilistically, we wanted users to type quickly and trust the decoder for error correction. We were concerned that allowing backspace might introduce variability between participants. For example some users might carefully correct errors with backspace while others rely on autocorrection. As we will show, our after correction error rates of $< 5\%$ show even without backspacing, users typed quickly with accuracy acceptable for many casual text entry tasks, e.g. messaging in a game.

If the virtual keyboard was enabled, it was shown at the bottom of the screen. Participants submitted a phrase for decoding by performing a *long keypress* by holding down any key for at least 300 ms. After each entry, participants were shown the entry rate and error rate for their last entry. Participants moved to the next phrase via another long keypress.

RESULTS

An unanticipated problem with our procedure was that some participants prematurely typed a long keypress. We believe this happened because many participants used the spacebar for their long keypresses at the end of each phrase. Another long press was required before the next phrase appeared. Therefore some participants accidentally pressed the spacebar for too long while typing a space between words while in the middle of an entry. To prevent these mistakes from interfering with our analysis, we removed 15 entries in which a participant was missing three or more words at the end of a phrase. Also, due to a logging bug, we had to drop another 38 entries. At most, this bug occurred three times to any participant in any condition's set of 20 phrases. Our analysis was on the remaining 1,867 entries from the original set of 1,920.

We tested for significance using a two-way repeated measures ANOVA. The two independent variables were whether the keyboard assistant was shown or not (denoted Feedback), and whether visual feedback was provided on an HMD or on a desktop monitor (denoted Display). Details of our statistical analysis appear in Table 2.

Entry rate

Entry rate is reported in words-per-minute (wpm). A word is defined as five characters including spaces. Entries were timed from the first key press until the recognition result was displayed to the user. This includes the time required for a user to perform the long keypress, networking delays, and recognition delays. In the study, network and recognition delays averaged 0.35 s per phrase (sd 0.71).

Entry rates were similar across all conditions: DESKTOP 44.7 wpm, DESKTOPASSISTANT 44.7 wpm, HMD 41.2 wpm, and HMDASSISTANT 43.7 wpm (Figure 3 and Table 1). Participants' slower entry rates using an HMD display were significant (Table 2). The similar or faster entry rates in the presence of the virtual keyboard assistant were not significant.

Entry rate (wpm)	Feedback	$F_{1,23} = 2.10$	$p = .16$	$\eta^2 = 1.5\times10^{-3}$
	Display	$F_{1,23} = 7.79$	$p < .05$	$\eta^2 = 4.4\times10^{-3}$
	Feedback x Display	$F_{1,23} = 3.61$	$p = .07$	$\eta^2 = 1.4\times10^{-3}$
Before correction error rate (CER %)	**Feedback**	$F_{1,23} = 7.21$	$p < .05$	$\eta^2 = 2.9\times10^{-2}$
	Display	$F_{1,23} = 0.22$	$p = .64$	$\eta^2 = 4.2\times10^{-4}$
	Feedback x Display	$F_{1,23} = 0.15$	$p = .70$	$\eta^2 = 2.4\times10^{-4}$
After correction error rate (CER %)	**Feedback**	$F_{1,23} = 6.78$	$p < .05$	$\eta^2 = 4.9\times10^{-2}$
	Display	$F_{1,23} = 0.02$	$p = .90$	$\eta^2 = 6.8\times10^{-5}$
	Feedback x Display	$F_{1,23} = 0.02$	$p = .89$	$\eta^2 = 1.0\times10^{-4}$

Table 2. Details of two-way repeated measures ANOVA. Significant differences are highlighted in bold.

Error rate

We report typing and recognition accuracy using character error rate (CER). CER is the number of character substitutions, insertions, and deletions needed to transform a participant's text into the target text divided by the characters in the target (multiplied by 100). We report the *before* and *after* correction CER. The before correction CER used the literal text the participant typed before any autocorrection. The after correction CER used the recognition result of our decoder.

Before correction error rates were lower in conditions with the virtual keyboard: DESKTOPASSISTANT 8.3% and HMDASSISTANT 8.4% versus DESKTOP 11.2%, and HMD 11.8% (Figure 3 and Table 1). Participants' lower before correction error rates using the virtual keyboard assistant were significant (Table 2). There was not a significant difference in before correction error rates between the HMD and desktop display.

Conditions with the virtual keyboard also had a lower error rate after decoding: DESKTOPASSISTANT 2.6%, and HMDASSISTANT 2.6% versus DESKTOP 3.9% and HMD 4.0%. Participants' lower after correction error rates using the virtual keyboard assistant were significant (Table 2). There was not a significant difference in after correction error rate between between the HMD and desktop display.

Before correction error rates were quite variable by participants as shown in Figure 4. The decoder reduced error rates for all participants, with some participants experiencing large gains in accuracy. All but four participants achieved an after correction error rate of less than 5% CER.

Recall that we removed 15 entries from our analysis due to participants erroneously ending their sentence three or more words early. We also conducted statistical analysis on the data without these entries removed. All the statistical conclusions were the same. The main difference in results was a somewhat elevated before correction error rate: DESKTOP 11.5%, DESKTOPASSISTANT 8.8%, HMD 12.8%, and HMDASSISTANT 8.5%. The after correction error rate was also somewhat elevated: DESKTOP 4.3%, DESKTOPASSISTANT 3.1%, HMD 5.2%, and HMDASSISTANT 2.7%.

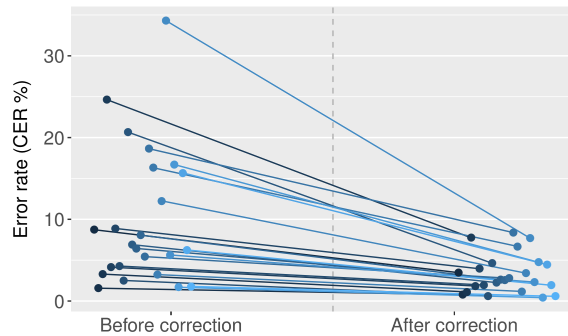

Figure 4. Participants' average error rate across all conditions before and after correction.

DISCUSSION AND CONCLUSIONS

Our results show the live feedback offered by the virtual keyboard assistant significantly reduced participants' error rates. Even though our feedback was of already committed keystrokes, users were still able to leverage this feedback to improve performance. We conjecture the feedback may have allowed users to correct mistakes such as offset hand position making them more accurate from that point on. It may also have provided a visual reminder of the QWERTY layout. Because most participants self-reported as touch typists, these results are only applicable to proficient typists.

Our autocorrection algorithm clearly improved users' accuracy across all conditions, even in the face of inaccurate input. Typing while wearing an HMD was slower, but not substantially more error prone than typing in a more normal desktop situation with a visually-occluded keyboard. We conjecture that HMD entry rates were slower due to unfamiliarity with the device, but since the visual information was largely the same across displays, error rates were similar. Further work is needed to better understand why HMDs seemingly affected speed but not accuracy.

We only tested entry of A–Z and apostrophe. We think our approach of a virtual keyboard assistant might be especially useful in the case of entry of less common characters such as symbols, numbers, or chorded keystrokes, though such text would also be more difficult to recognize accurately.

In summary, we found a virtual keyboard providing live feedback coupled with an autocorrection algorithm substantially improved users' typing performance while wearing an HMD. A notable advantage of our approach is that it does not require any tracking devices or external cameras, making it readily usable without additional hardware. This could be a significant factor for some users who are on a budget or do not have the space to install additional devices. A system that does not "intrude" on the virtual environment by superimposing real-world imagery over the virtual imagery might also be less distracting for HMD users. Our findings suggest software-only solutions for improving users' ability to interact with physical devices while wearing an HMD are worth continued investigation.

REFERENCES

1. Bowman, D. A., Rhoton, C. J., and Pinho, M. S. Text input techniques for immersive virtual environments: An empirical comparison. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 46*, 26 (2002), 2154–2158.
2. Budhiraja, P., Sodhi, R., Jones, B., Karsch, K., Bailey, B., and Forsyth, D. Where's my drink? Enabling peripheral real world interactions while using HMDs. *arXiv preprint arXiv:1502.04744* (2015).
3. Lee, M., and Woo, W. ARKB: 3D vision-based augmented reality keyboard. In *Proceeding of the 13th International Conference on Artificial Reality and Telexistence* (2003).
4. McGill, M., Boland, D., Murray-Smith, R., and Brewster, S. A dose of reality: Overcoming usability challenges in VR head-mounted displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, ACM (New York, NY, USA, 2015), 2143–2152.
5. Vertanen, K., and Kristensson, P. O. A versatile dataset for text entry evaluations based on genuine mobile emails. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, ACM (New York, NY, USA, 2011), 295–298.
6. Vertanen, K., Memmi, H., Emge, J., Reyal, S., and Kristensson, P. O. VelociTap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, ACM (New York, NY, USA, 2015), 659–668.
7. Walker, J., Kuhl, S., and Vertanen, K. Decoder-assisted typing using an HMD and a physical keyboard. In *Extended Abstracts of the the ACM Conference on Human Factors in Computing Systems*, CHI '16 workshop (New York, NY, USA, 2016).
8. Yi, X., Yu, C., Zhang, M., Gao, S., Sun, K., and Shi, Y. ATK: Enabling ten-finger freehand typing in air based on 3d hand tracking data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, ACM (New York, NY, USA, 2015), 539–548.