

Slide: Fast Text Input in Virtual Reality

ABSTRACT

Text input in VR is challenging because of the latency in visual feedback and the lack of proprioception. Our design Slide, a virtual keyboard that requires only a track pad, is based on 3 concepts: (1) relative vectors: to enter a character, the user slides from what is assumed to be the center of the keyboard to the key of interest. (2) directional, with error correction: all keys are grouped in 6 tiles, and the user only has to slide in the direction of the tile containing the key. A Bayesian word recommender determines the word entered. (3) Intention-based hybrid keyboard: the keyboard automatically switches between the 26-key and 6-tile keyboard based on the user's finger movements. Our user study suggests that Slide can be learned quickly, and delivers an average text entry speed of 34 words per minute with an error rate of 4%.

ACM Classification Keywords

H.5.1. Information Interfaces and Presentation (e.g. HCI): Multimedia Information Systems; I.3.7. Computer Graphics: Three-Dimensional Graphics and Realism

Author Keywords

virtual reality; text input; text entry; multimodal.

INTRODUCTION

While the primary use of virtual reality is in games and videos today, it can be used in numerous application domains, from education, architectural reviews, healthcare, large group chats, and data visualization, exploration of space and other dangerous locations, scientific visualization, manufacturing, journalism, traveling, architecture, and shopping.

Virtual reality (VR) systems have become available to the general public recently and the virtual reality controllers introduced by HTC, Facebook, and Google have similar characteristics. Each includes a controller, shown in Figure 1. The Daydream controller, shown in Figure 1c, is only for a single hand whereas the others have a controller for each hand. The three major controllers all contain a way to track the thumb, at least 3 degrees of freedom tracking (rotational), and at least four buttons. Vive (e) and Oculus (b) use an absolute tracking system to minimizing sensor drift and to allow for rotational and translational tracking [20]. The Vive and Daydream (c) employ a thumb track pad to capture finger movement. Oculus controller uses joysticks instead.

Paste the appropriate copyright statement here. ACM now supports three different copyright statements:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single spaced.

Every submission will be assigned their own unique DOI string to be included here.

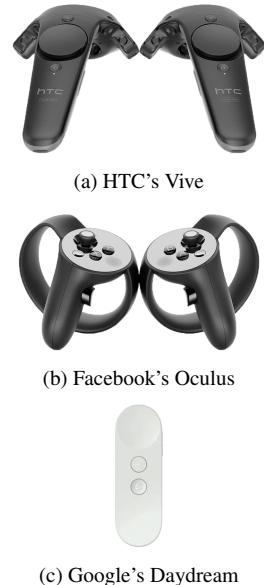


Figure 1: Virtual reality controllers

Even though typing is not prevalent in immersive virtual reality, text entry is still important, with a wide range of uses: from typing in account names and passwords, searching items, specifying parameters used in the applications, referring to friends, writing notes and chatting with others, etc.

The average typing performance on a regular keyboard was found to be about 60 words per minute[56], thanks to the dexterity of our fingers, the haptic feedback from the physical keyboards, and visual feedback on the display. In the context of virtual reality, it is often not an option to use a physical keyboard. We are usually not sitting down and it is hard to switch between the VR controllers and a physical keyboard, further complicated by the fact that we cannot see the keyboard in VR. To avoid having to remove the head gear to type, it is desirable to enter text using existing VR controllers.

Typing is challenging in virtual reality because of the latency in visual feedback and the lack of proprioception. The industry standard is to enter text with gaze. A virtual keyboard is presented in VR, and users type a letter by gazing at the desired key and tapping on the controller. It was found that the users can typically type at about 7 words per minute [39]. Entering text via speech is attractive, especially in the immersive virtual reality environment. Speech may not be suitable when entering names and URLs, especially in public spaces due to privacy issues. Furthermore, speech often needs to be corrected via text editing. Other alternatives proposed including using gesture, motion controllers, Leap Motion or tangibles, handheld controllers, or relying on a subset of keyboard and

mouse commands [7]. All these existing solutions, while potentially good for simple tasks, aren't adequate for tasks that require a greater bandwidth of input [43].

Methodology

The goal of the paper is to understand how we can enter text simply, quickly, and accurately via the touchpad found in commodity VR controllers today.

1. We started our investigation by exploring existing ideas:

- (a) **Gazing.** Users enter keys by gazing at the keys in a virtual keyboard.
- (b) **Drumming.** Keys are displayed as a virtual drum kit, users enter keys by hitting the corresponding drums using the controllers as drumsticks[30].
- (c) **Augmented reality.** A camera is used to stream the view of the touchpad and fingers into virtual reality.
- (d) **26-key tapping.** Users use the touchpad to tap on the virtual keyboard displayed in the VR.
- (e) **26-key swiping.** Users use the touchpad to swipe on the virtual keyboard displayed in the VR.

We built a prototype of each of these systems and tested it informally on 3 to 5 users. This study revealed major deficiencies in each of these approaches.

2. We propose a new virtual keyboard, Slide, that incorporates the learning from our initial investigation. It is based on three main design concepts:

- (a) **Relative vectors.** To enter a character, the user slides from what is assumed to be the center of the keyboard to the key of interest.
 - (b) **Directional input, with error correction.** All keys are grouped in 6 tiles, and the user only has to slide in the direction of the tile containing the key. A Bayesian word recommender determines the word entered.
 - (c) **Intention-based hybrid keyboard:** the keyboard automatically switches between the 26-key and 6-tile keyboard based on the user's finger movements.
3. We implemented the proposed technique in the Vive system; note that Slide is applicable to the other VR controllers as well. We conducted a study with 15 users and evaluated the speed, accuracy, as well as the learnability of the system.

Contribution

This paper proposes Slide, a VR text entry technique that requires only a touch pad. To address the visual latency and lack of proprioception in VR, we have adopted relative positioning with a heavy use of auto-correction. The virtual keyboard is divided into six tiles placed around the center of the keyboard. The user needs to slide his finger in one of the six directions towards the tile containing the character, and error correction is used to pick the most likely word. Our study shows that users can type at the speed of 34 words per minute on average, with an error rate of 4%.

Slide allows for graceful degradation, when better control over the text entry is necessary, such as for typing in passwords. The

user can fall back to sliding to the precise key; our keyboard automatically detects the use of the more precise mode by observing the speed of the sliding motion. The users can type at about 12 words per minute using this slow but more precise entry method.

RELATED WORK

In this section, we discuss studies applicable to virtual reality based text entry. For a full review of text entry methods, see [36, 64, 9].

Novel QWERTY Keyboards

Researchers leverage the familiarity of the QWERTY [47] keyboard with new text entry methods. The Half-QWERTY keyboard [42] is arm-mounted and has a reduced size by using only half of a QWERTY keyboard. The Pinch Gloves [8] controls a virtual QWERTY keyboard with hand rotations and finger-mounted buttons. Another glove, KITTY [29] uses the tapping of contacts on each finger with the thumb to mimic a keyboard. DigiTap [50] is a similar idea but uses a wrist mounted camera instead of a glove.

Gloves, Chorded Keyboards, and Touch Pads

Other approaches abandon the QWERTY keyboard entirely. Chorded keyboards [46], such as Twiddler [33] use a combination of a few keys to produce characters. Chording is done with gloves, such as Chording Glove [51]. The VirtualPhonepad [1] imitates text entry via a 3×3 mobile phone keypad. Virtual Notepad [49] allows for creating handwritten annotations in virtual reality. Connecting the Dots [15] approximates handwriting by having users connect dots on a 3×3 virtual dot matrix to scribe a single character.

In a literature review, the Pinch Gloves, pen and tablet, chorded keyboard, and a speech are compared [8]. The pen and tablet and speech are fastest. The pen and tablet had the fewest errors. The comparison was done before specialized virtual reality input device were invented.

Word Gesture Keyboards

Word gesture keyboards such as ShapeWriter [62, 63] and SHARK 2 [28], are gesture based text input methods. Words are written through a gesture that links the letters of the intended word on a soft keyboard. A language model selects the most likely word from a list of possible candidates. For out of vocabulary and hard-to-predict words, a deterministic input method is still needed. VelociTap [58] takes the predictive model further by only predicting the whole sentence.

Home Entertainment Input Devices

Text entry methods is sought after in other application areas, such as home entertainment systems. SpeeG [22] is a Kinect [16] version of Speech Dasher [57] text input technique. The user speaks and then corrects mistakes using an interface that scrolls through the recognized text as well as alternatives. SpeeG2 [23] uses a grid instead of a scroll.

Speech Recognition

Spoken language is effective for human-human interaction but often has limitations when applied to human-computer

interaction. Speech is slow for presenting information and interferes significantly with other cognitive tasks [55].

Speech recognition is limited when typing passwords, names, and URLs [6]. A lack of privacy, the perception of bothering others, and the awkwardness of speaking to a machine [53] restrict the use cases in which speech is useful. Some users find speech unreliable when an accent or ambient sound negatively affect accuracy. Studies have shown that users find it “harder to talk and think than type and think” and considered the keyboard to be more “natural” than speech for text entry [27].

A virtual reality speech-based text-entry system represents candidate words recognized by the speech recognition engine as blocks that can be moved around in the virtual environment [48].

For mobile phones, speech entry is three times faster than typing for English and Mandarin [52]. Participants were given a set of phrases to type and speak. However, normal speech is filled with hesitations, repetitions, changes of subject in the middle of an utterance [14] as the user thinks. This could lead to an unrealistic accuracy and speed for speech. The study excludes punctuation marks and capital letters from the phrases. Generating accurate punctuation automatically with speech input is not yet possible and must be done manually [11].

Gaze Directed Typing

Gaze directed typing [38] places a mouse pointer directly at the user’s point of gaze over a virtual keyboard. This is the main text input method in systems such as Oculus GearVR [19]. An advantage of this method is that it only uses the hardware available in the head-mounted display without the need for additional controllers. This method is slow [40, 10, 34] and inconvenient since the whole head has to move to select each character. Moving the head in a virtual environment while the soft keyboard remains static can cause nausea and dizziness [4].

Virtual Reality Controllers

Input methods and controllers are being developed specifically for use in virtual reality. FaceTouch [18] uses the backside of the head mounted display as a touch sensitive surface. The user selects virtual content by touching the corresponding location on backside of the display. It is useful for short periods of time until the users’ arms get tired.

Belt [13] is an unobtrusive input device for wearable displays with a touch surface encircling the users’ hips. The device suffers from low throughput, arm fatigue, and social acceptance. Nenya [3] uses a finger ring as an input mechanism that is always available. It is fast to access, allows analog input, and is socially acceptable by being embodied in a commonly worn item.

EXPLORING THE DESIGN SPACE OF TEXT INPUT IN VR

Challenges of Text Entry in Virtual Reality

Text entry using a physical keyboard is a complex interaction that involves the placement of the user’s hands, the kinesthetic feedback of fingers, and the physical constraints of the keyboard [43]. Together this interaction is able to create an efficient text entry mechanism.

However, in the context of virtual reality, it is often not possible to use a physical keyboard. Additionally, it is desirable to be able to type with existing virtual reality specific input devices, so the user doesn’t have to remove the head-mounted display to type. In this next section we explore the challenges of text in virtual reality.

Latency in Visual Feedback

Typing is complex interaction that requires a high-bandwidth feedback loop [43]. Latency is a barrier in achieving a seamless interaction between the real and virtual [31].

In virtual reality, latency is the time between movement of the hand and the updated image being displayed on the screen. This pipeline includes the times for fusion, image transmission, rendering, sensor response and display response. This is commonly referred to as the *motion-to-photon* latency.

To achieve a genuine sense of presence [54] in virtual reality, the total *motion-to-photon* latency must be less than 20 milliseconds [24, 25, 5]. Some existing systems have a latency as high as 80 milliseconds [32, 12].

Compared to head movement, touch-based interactions need even tighter tolerances on latency. Research on touchscreens finds that for a user to perceive display elements as naturally being affected by touch, there is a perceptual floor between 2 and 11 milliseconds below which users do not notice lag [26, 45]. Further, for dragging interactions, a latency of 2.38 milliseconds is required to give the perception of a natural interaction [26, 45].

Limited Proprioception

Manipulation in virtual reality is difficult because users must do without the haptic contact with keyboard buttons they rely on in the real world to orient themselves. Interactions involving the body and the external world become difficult. For example, even the simple task of placing the finger on the correct key can become difficult given the lack of proprioceptive sensations in virtual reality.

Reduced Precision

Haptic feedback and physical constraints in real world input devices help to define interactions. Fine-grained control is difficult in virtual reality and restricts users to the coarse manipulation of virtual objects. Input devices that exclude the use of the fingers are slower [65].

Initial Experimentation

Our exploration of virtual reality input methods starts by building a few prototypes. With each prototype, we ask 3 to 5 users to type a few sentences. For each test we give little to no instruction about how the input method works and try to let the user interact with it naturally. The goals of this prototyping process is to explore the performance potential of each input device and to investigate if users could understand the visual interface. We record typing speeds, their understanding of the input device, and their verbal feedback.

Design #1 - Gaze Input

Many virtual reality apps today use a gaze keyboard for text entry [44]. A virtual keyboard is presented in virtual reality.

	Virtual Reality			
	Speech [52]	Mobile [52]	Gaze [39]	Slide
<i>Hardware</i>	microphone	touch screen	HMD & button	controller
<i>Keys</i>		26+	26+	6
<i>Special Characters</i>		✓	✓	✓
<i>Granularity</i>	word	character	character	word or character
<i>Hands</i>	0	1 or 2	1	1
<i>Real – Time Feedback</i>	word	character	character	word
<i>Cursor Mode</i>	NA	NA	persistent	snap-to-home
<i>Silent</i>		✓	✓	✓
<i>Muscle</i>	vocal chords	fingers	neck	thumb
<i>WPM</i>	161	53	7	34
<i>Error Rate</i>	2.93%	3.68%	.54%	4%

Table 1: A table comparing speech, mobile keyboard, gaze, and Slide. Slide is the text entry method presented in the paper. For comparison, gaze is the main text entry method in use in systems such as Oculus GearVR.

The users types a letter by gazing at the desired key and tapping on a button located on the side of the head mounted display. Users find the process slow and the motion of the head with the fixed keyboard can causes nausea and dizziness [4]. It was found that users type around 7 words per minute using this method [39].



Figure 2: The 26-Key tap keyboard design.

Design #2 - Drumming

One proposal is to model typing as drumming [59]. Controllers in the real world are mapped to drum sticks in the virtual world. Typing a character means hitting one of the 26 drums laid out like a keyboard. Users find it fun. We found that it could be difficult to hit the correct key and some users found the constant arm movement to be strenuous. We found users type around 20 words per minute using this method.

Design #3 - External Camera

We next explore the use of a virtual keyboard on a touchscreen, the input method used on mobile phones today. To address the problem that we cannot see our fingers in virtual reality, we experimented with mixed reality (augmented plus virtual) with a camera. Our prototype points the front facing camera on the head mounted display points at the mobile keyboard, and streams the view of our fingers and the keyboard into the virtual environment.

The goal of this prototype is to investigate how latency and presence, particular of the fingers, affect performance. We found that the latency is an inhibitor to fast input. Our fingers are so fast that we can only see where our fingers were. What we see is constantly wrong and just makes it even harder to type. This prototype demonstrates how challenging the visual latency is for typing. We found users type around 10 words per minute using this method.

Design #4 - 26-Key Tap

In this experiment, we explore how feedback can be added to a soft keyboard on the mobile phone to make it usable in virtual reality. We want to explore the limit of touch typing and muscle memory for a user in a virtual environment. We build a prototype keyboard, shown in Figure 3, for a mobile phone that transmits to the head-mounted display when where the user touches on the keyboard. In this way, the user gets feedback where a finger is located on the keyboard with every tap. The prototype keyboard is running alongside the standard Android keyboard that includes autocorrect to adjusted for typos and phonetic misspelling.

We notice an unexpected phenomenon using this prototype. Even hitting the same key twice is difficult. Every time a user lifts a finger and lowers the finger, there is a nontrivial chance that the finger has drifted to another key. Even though the keyboard layout is familiar, and even error correction is applied, the error rate in this method make it unusable. From this experiment, we hypothesize that for fine motor skills, some degree of either visual or haptic feedback is needed.

Design #5 - 26-Key Gesture

Gesture keyboards are shown to be effective in speeding up typing on mobile devices. Since it does not involve lifting the finger, we postulated that it may also be useful in virtual



Figure 3: The 26-Key gesture keyboard design.

reality. Similar to the 26-key tap, the layout of the mobile QWERTY keyboard is preserved, except tapping is replaced by dragging to specific keys. We show the virtual keyboard and gesture trail to indicate where the finger was in the virtual environment, shown in Figure 3. We find that gestures works better than tapping. However, there are two main problems:

Firstly, the users cannot get the first letter right when starting the gesture. We modify the shape recognition algorithm to as follows. When the user puts their finger on the mobile screen, it always starts in the center. The first inflection point is then discarded and the rest of the input is then done at the mobile phone system level using the standard gesture keyboard.

However, the speed with which the user creates gestures with the head-mounted display on is around three times slower than with the head-mounted display off. Even for users that are familiar with gesture keyboards on their mobile phone, it is difficult for them to generate gestures in an efficient manner in virtual reality.

In summary, the use of gesture keyboard over a conventional keyboard is the best among the choices we explored in the initial phase. Nonetheless, the speed of around 23 words per minutes, with the help of auto-correction, is still lower than desired.

DESIGN CONCEPTS

We learned from our investigation that it is slow for users to hit the right key on a virtual keyboard due to the lag in visual feedback and limited proprioception. We thus propose Slide, described below, to speed up text entry by using relative positioning and reducing the need for accuracy through auto-correction.

Relative Vectors as Key Entries

To minimize the need for visual feedback, we propose to use relative positioning. Relative movement, instead of absolute, has also been found to be beneficial in reducing fatigue [21] so the user can start the interaction from their current position.

The virtual keyboard in Slide is laid out exactly as the original keyboard, so the users are already familiar with the placement of the keys. The positioning of the keyboard is relative to the user's first touch. No matter where the finger is placed on the touchpad, it is assumed to touch the center of the keyboard, right between the "f" and "g" letters. To enter a letter, the user slides his finger to the key of interest. As he lifts his finger, the key is entered. Because of the familiarity of the keyboard layout, the user knows roughly even without looking at the

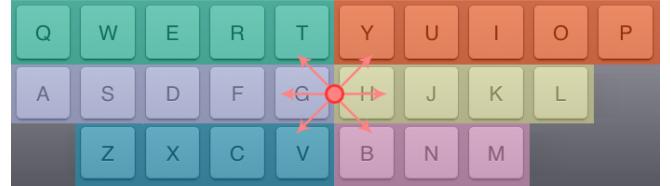


Figure 4: The six tiles of the direction-based keyboard.

keyboard. Visual feedback is provided by showing the virtual keyboard and the trail his finger makes.

Text entry in Slide is different from swiping in several important ways.

1. With swiping, the user enters a word through a continuous movement, connecting pairs of consecutive letters. Without good visual feedback, the drift in our sense of our finger position accumulates with each letter. With Slide, every letter input is independent. The user needs only to slide always from the center of the keyboard to the letter of interest.
2. As discussed above, we need to discard the first letter entry when we swipe in VR, since users cannot hit the first key correctly. No such special provision is needed for Slide.
3. With swiping, the worst-case distance traveled for each letter is the full length of the keyboard. The worst case in Slide is just half the length of the keyboard.

Directions as Key Entries

While an improvement over swiping, the technique proposed above still requires the user to slide his finger precisely to the right key, which can be half the length of the keyboard. We reduce the accuracy required of the user by relying on auto-correction. We divide the keyboard into 6 tiles, as shown in Figure 4. The user only needs to slide his finger in the direction of the tile that contains the key of interest. It is easy to indicate the direction accurately, with a minimum distance traveled for all keys, without visual feedback. Figure 6 shows how the word "hello" is entered using this method.



Figure 5: Users are presented with words represented by the same sequence of tile inputs.

We use a Bayesian word recommender to infer the most probable words among words that correspond to the same tile

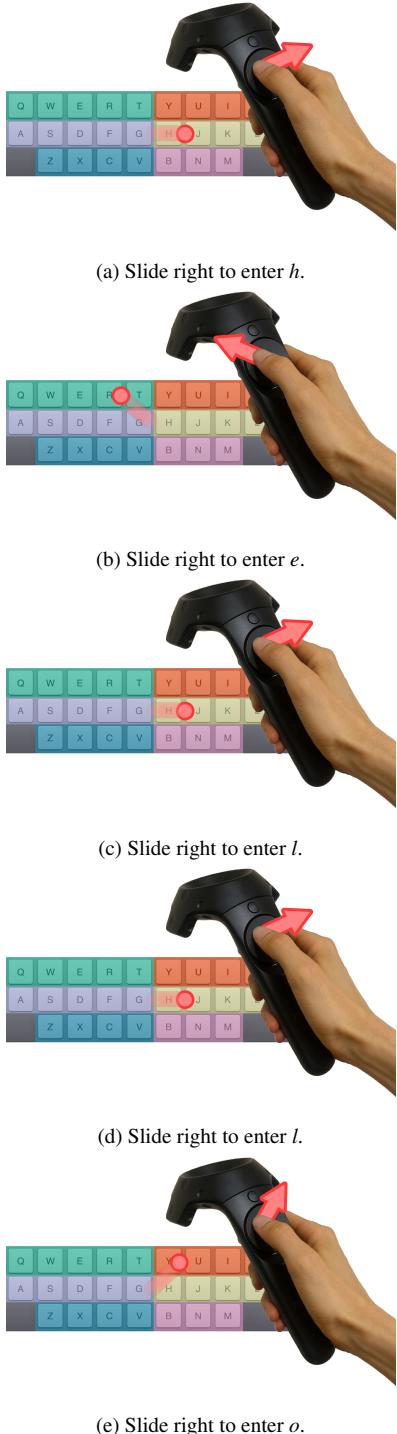


Figure 6: An example of how the word *hello* is entered using Slide.

sequence. For example, the words “hello” and “jelly” has the same representation. This recommender algorithm use 2-gram and 3-gram data from Google Books Ngram Viewer¹, part of speech frequency for English dictionary, individual word frequency for the entire English language, and produces a prediction score. The word with highest prediction score will be selected by default as the most probable word. The whole word is presented all at once, instead of a letter at a time, and the user is given the option to choose a less probable word among the possibilities, as shown in Figure 5.

A Hybrid Keyboard

While the directional approach is faster, it lacks the precision necessary for entering numbers, punctuations, or words not in the dictionary, such as user names, passwords, and URLs. Our solution is to use the relative-vector technique as a backup for the direction-based keyboard. Since both these techniques have the same keyboard layout, the user simply can mix these two entry techniques easily. Our keyboard automatically detects the user’s intention based on the speed with which the keys are entered.

In addition, we introduce a few shortcuts for the frequently used “delete” and “space” keys. Users can double tap the touch screen to indicate delete. For the space key, the user can click a specific physical button on the controller, such as the “trigger” button on the bottom side of the Vive controller. We have also experimented automatically inserting a space after a certain timeout period between 150 and 500 ms. However, we found that users preferred to have the control of when to insert the space themselves.

Variations of Slide

To arrive at the proposed design above, we have tried a few other variations, summarized below.

1. 6-tile design and tapping with absolute positioning. As in Slide, we divide the keys into 6 tiles, and users are asked to just tap on the corresponding tile. We found that users still cannot hit the right tile, if we adopt absolute positioning.
2. 6-tile design and swiping with absolute positioning. This design is faster than swiping on a 26-key keyboard, but it is still slower than our proposal.
3. 8-tile design with two cursors, and relatively positioning. Since most people type with two hands on the keyboard, we experimented introducing two cursors into the proposed design. We split the keyboard into the left and right half, and the initial positions of the left and right thumbs are assumed to be the center of each half. Instead of sliding to one of 6 directions for each hand, we simplify it to four directions each (up, down, left, right). By having 8 rather 6 tiles, this can also reduce the number of ambiguous entries and improve the success of error correction.

Testing it on a few users quickly pointed out the flaw of this design. It is too complicated, as the user has to decide which thumb to use, and then which of the four directions. Furthermore, we found that with the original proposal, the

¹<http://storage.googleapis.com/books/ngrams/books/datasetsv2.html>

user typically uses the other hand to enter the special the special keys.

Because none of the variations come close to the proposed entry method, our evaluation below focuses primarily on understanding the characteristics of our proposal.

EVALUATION OF SLIDE

Participants

Fifteen subjects, 5 female and 15 male, participated in the study. All were university students and active users of technology. Each subject took about 30 minutes to complete the study.

Measures

In the field of text entry, several metrics are used to characterize a method's performance [60, 2]. Here, we discuss the performance metrics we use to evaluate the performance of the input device.

Words per Minute

Words per minute (*WPM*) is perhaps the most widely reported empirical measure of text entry performance [60]:

$$WPM = \frac{|T| - 1}{S} \times 60 \times \frac{1}{5}. \quad (1)$$

Where, *S* is the time in seconds from the first key press to the last, which means that the entry of the first character is never timed, which is the motivation for the “- 1” in the numerator of Equation 1 [61]. English words by convention are treated as having five characters [61].

Error Rate

Error Rate (*ER*) is the ratio of the total number of incorrect characters in the transcribed text to the length of the transcribed text:

$$ER = \frac{INF}{|T|} \times 100\%. \quad (2)$$

Where, Incorrect Not Fixed (*INF*) is the number of unnoticed incorrect characters in the transcribed text.

Experiment #1 - 26-Key Slide Keyboard

Our first experiment is to measure the speed at which users can type in uncommon English words. The user has to slide his finger to the specific keys to enter the characters precisely. There is no auto-correct used in this mode as the primary use is for words that are not in the language model.

In this experiment, the users were asked to type in two phrases: one that contained a proper noun and one that contained an abbreviation. We found that the users can enter these phrases with an average entry rate of 12 WPM, with a standard deviation of 1.8. We observed that practice has no significant effect on improving the speed. 12 words per minute is a reasonable input rate, considering that input words are uncommon, and errors in entry cannot be corrected automatically.

Learnability of Slide

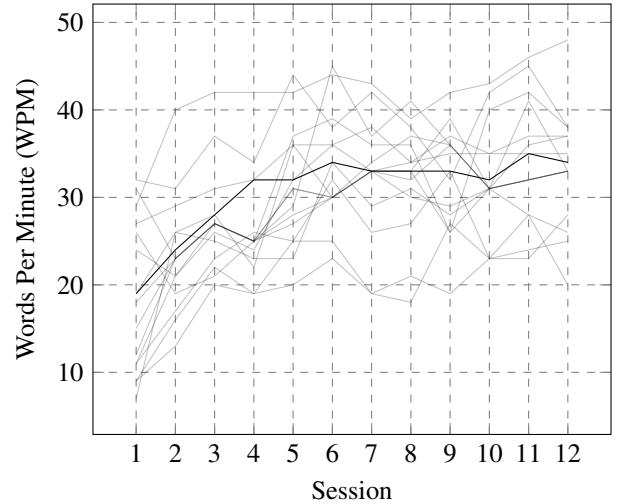


Figure 7: Graph of learnability for the 6-slide Keyboard. Entry rate on each block of 20 sentences. Each line is a single participant.

Experiment #2 - 6-slide Keyboard

For this experiment, we selected fifty phrases from a collection of five-hundred phrases commonly used for text entry evaluations [37]. Example of phrases from the set are “video camera with a zoom lens”, “have a good weekend”, and “what a monkey sees a monkey will do”. To allow for cross-comparison to other text entry methods, the corpus doesn't include phrases with punctuation marks. Non-alphanumeric symbols are rarely considered in text input research [37] even though some punctuations (.-'()") are more frequent than the least common English letter, *q* [41].

Copying pre-selected phrase is usually the preferred method for text entry when doing evaluations in a lab setting [35, 37]. Although, copying pre-selected entry rates should be regarded differently from true, “in the wild” results when the user is composing text instead of transcribing prompted phrases.

One of our metrics in the experiment is the learnability of the system so we don't give users practice trials before beginning the actual test. For each user we conduct twelve sequential sessions. In each session, a user enters six phrases.

The collective learning curve is shown in Figure 7. The mean starting rate for all users was 18 WPM. Every user improved their WPM rate by at least 20% from their starting rate. To calculate the steady state average rate for text entry, we discard the first six sessions and find the steady state rate is 34 WPM with a standard deviation of 6.3. The average error rate for text entry was 4% with a standard deviation of .018.

CONCLUSION

Overcomes Latency and Proprioception with Slide

The lack of proprioception in virtual reality is compounded by the latency in visual feedback. For a user to be able to steer their finger to a key, the time needed to transmit information from the real to virtual and back to the “real” through the screen and into the eyes of the user is too long. Other commonly used input methods, such as gesture keyboards, are subject to the same problem. Therefore, we devise a method of input that does not depend on a feedback loop that requires the user to reach an absolute position on a keyboard.

Our technique only depends on relative movement. The user slides the finger on the touchpad to indicate the relative position of the key of interest with respect to the center of the keyboard. We further reduce the precision required by grouping keys into 6 tiles, and the user only needs to indicate the direction of the tile, again, with respect to the center of the keyboard. This reframes the domain of text input from a closed feedback control problem to an open feedback control problem. As such we have created a system that lets users enter text in virtual reality efficiently. With the help of error correction, and including the time of error correction, users can type at about 34 words per minute with an error rate of 4%. Furthermore, when precision is desired, the user can simply go slower and hit the precise keys. The user can still precisely, which is useful for passwords, etc, at a rate of 12 words per minute.

Smooth Transition From Newbie to Novice to Expert

A smooth pathway learnability is an important factor for an input device to gain widespread adoptability. Some users will immediately reject an input device that requires too much up front practice before it becomes useful. Slide has a natural path to transition novice (low threshold) to expert (high ceiling) users [17].

Slide offers a clear path from newbie to novice to expert. Slide is instantly usable: newbie users can use the system quickly by looking at the controller and slowly moving their thumb to a key. Novice users would periodically glance down to look at the virtual keyboard if they forgot the position of a key. With a little of practice, users can remember the location of the keys. Experts can just type without having to look at the virtual keyboard.

General-Purpose Text Entry

As virtual reality matures, we expect VR systems to have a standard text-entry method at the OS level, that can be learned once and used across applications. This is analogous to the adoption of a pointing device for PCs. This text-entry method must be general-purpose, accommodating the entry of proper nouns, abbreviations, capital letters, colloquialism, passwords, etc. Allowing both fast entry of common words and precise, albeit slower, of the characters, Slide is positioned well as a candidate for the keyboard of a VR operating system.

REFERENCES

1. Jaewoo Ahn and Kyungha Min. 2006. VirtualPhonepad: a text input tool for virtual environments. In *Advances in Artificial Reality and Tele-Existence*. Springer, 58–64.
2. Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of text entry performance metrics. In *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*. IEEE, 100–105.
3. Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nenya: subtle and eyes-free mobile input with a magnetically-tracked finger ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2043–2046.
4. Rowel Atienza, Ryan Blonna, Maria Isabel Saludares, Joel Casimiro, and Vivencio Fuentes. 2016. Interaction techniques using head gaze for virtual reality. In *2016 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 110–114.
5. Randall E Bailey, Jarvis J Arthur III, and Steven P Williams. 2004. Latency requirements for head-worn display S/EVS applications. In *Defense and Security*. International Society for Optics and Photonics, 98–109.
6. Issam Bazzi. 2002. *Modelling out-of-vocabulary words for robust speech recognition*. Ph.D. Dissertation. Massachusetts Institute of Technology.
7. Mark Billinghurst and Hirokazu Kato. 1999. Collaborative mixed reality. In *Proc. Int’l Symp. Mixed Reality*. 261–284.
8. Doug A Bowman, Christopher J Rhon, and Marcio S Pinho. 2002. Text input techniques for immersive virtual environments: An empirical comparison. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 46. SAGE Publications, 2154–2158.
9. Guiard. Sellene Buxton, Billinghurst and Zhai. 2011. Human Input to Computer Systems: Theories, Techniques and Technology. *Technical Report* (2011).
10. Stuart K Card, Allen Newell, and Thomas P Moran. 1983. The psychology of human-computer interaction. (1983).
11. C Julian Chen. 1999. Speech recognition with automatic punctuation.. In *EUROSPEECH*.
12. Mikael Dallaire-Côté, Philippe Charbonneau, Sara St-Pierre Côté, Rachid Aissaoui, and David R Labbe. 2016. Animated self-avatars for motor rehabilitation applications that are biomechanically accurate, low-latency and easy to use. In *2016 IEEE Virtual Reality (VR)*. IEEE, 167–168.
13. David Dobbelstein, Philipp Hock, and Enrico Rukzio. 2015. Belt: An unobtrusive touch input device for head-worn displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2135–2138.
14. Markus Forsberg. 2003. Why is speech recognition difficult. *Chalmers University of Technology* (2003).
15. Scott Frees, Rami Khouri, and G Drew Kessler. 2006. Connecting the Dots: Simple Text Input in Immersive Environments.. In *VR*. 265–268.

16. Daphne J Geerse, Bert H Coolen, and Melvyn Roerdink. 2015. Kinematic Validation of a Multi-Kinect v2 Instrumented 10-Meter Walkway for Quantitative Gait Assessments. *PloS one* 10, 10 (2015), e0139913.
17. Shuchi Grover and Roy Pea. 2013. Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher* 42, 1 (2013), 38–43.
18. Jan Gugenheimer, David Dobbelstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. 2016. FaceTouch: Touch Interaction for Mobile Virtual Reality. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 3679–3682. DOI: <http://dx.doi.org/10.1145/2851581.2890242>
19. Jeff Hecht. 2016. Optical Dreams, Virtual Reality. *Optics and Photonics News* 27, 6 (2016), 24–31.
20. Thomas Hilfert and Markus König. 2016. Low-cost virtual reality environment for engineering and construction. *Visualization in Engineering* 4, 1 (2016), 1.
21. Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasiyan, and Pourang Irani. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-air Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1063–1072. DOI: <http://dx.doi.org/10.1145/2556288.2557130>
22. Lode Hoste, Bruno Dumas, and Beat Signer. 2012. SpeeG: a multimodal speech-and gesture-based text input solution. In *Proceedings of the International working conference on advanced visual interfaces*. ACM, 156–163.
23. Lode Hoste and Beat Signer. 2013. SpeeG2: a speech-and gesture-based interface for efficient controller-free text input. In *Proceedings of the 15th ACM on International conference on multimodal interaction*. ACM, 213–220.
24. Jason Jerald and Mary Whitton. 2009. Relating scene-motion thresholds to latency thresholds for head-mounted displays. In *2009 IEEE Virtual Reality Conference*. IEEE, 211–218.
25. Jason J Jerald. 2010. *Scene-motion-and latency-perception thresholds for head-mounted displays*. Ph.D. Dissertation. University of North Carolina at Chapel Hill.
26. Ricardo Jota, Albert Ng, Paul Dietz, and Daniel Wigdor. 2013. How Fast is Fast Enough?: A Study of the Effects of Latency in Direct-touch Pointing Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2291–2300. DOI: <http://dx.doi.org/10.1145/2470654.2481317>
27. Clare-Marie Karat, Christine Halverson, Daniel Horn, and John Karat. 1999. Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 568–575. DOI: <http://dx.doi.org/10.1145/302979.303160>
28. Per-Ola Kristensson and Shumin Zhai. 2004. SHARK 2: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*. ACM, 43–52.
29. Falko Kuester, Michelle Chen, Mark E. Phair, and Carsten Mehrling. 2005. Towards Keyboard Independent Touch Typing in VR. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '05)*. ACM, New York, NY, USA, 86–95. DOI: <http://dx.doi.org/10.1145/1101616.1101635>
30. Google Daydream Labs. 2016. Daydream Labs: exploring and sharing VR's possibilities. URL. (19 May 2016). Retrieved May 19, 2016 from <https://developers.googleblog.com/2016/05/daydream-labs-exploring-and-sharing-vrs.html>.
31. Brian Lee, Rafael Ballagas, and Maureen Stone. Designing for Latency: In Search of the Balance Between System Flexibility and Responsiveness. (????).
32. Peter Lincoln, Alex Blate, Montek Singh, Turner Whitted, Andrei State, Anselmo Lastra, and Henry Fuchs. 2016. From Motion to Photons in 80 Microseconds: Towards Minimal Latency for Virtual and Augmented Reality. *IEEE transactions on visualization and computer graphics* 22, 4 (2016), 1367–1376.
33. Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and EW Looney. 2004. Twiddler typing: one-handed chording text entry for mobile phones. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 671–678.
34. I Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction* 7, 1 (1992), 91–139.
35. I Scott MacKenzie and R William Soukoreff. 2002a. A character-level error analysis technique for evaluating text entry methods. In *Proceedings of the second Nordic conference on Human-computer interaction*. ACM, 243–246.
36. I Scott MacKenzie and R William Soukoreff. 2002b. Text entry for mobile computing: Models and methods, theory and practice. *Human–Computer Interaction* 17, 2-3 (2002), 147–198.
37. I Scott MacKenzie and R William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*. ACM, 754–755.
38. Päivi Majaranta. 2009. *Text entry by eye gaze*. Tampereen yliopisto.
39. Päivi Majaranta, I Scott MacKenzie, Anne Aula, and Kari-Jouko Räihä. 2006. Effects of feedback and dwell time on eye typing speed and accuracy. *Universal Access in the Information Society* 5, 2 (2006), 199–208.

40. Päivi Majaranta and Kari-Jouko Räihä. 2002. Twenty years of eye typing: systems and design issues. In *Proceedings of the 2002 symposium on Eye tracking research & applications*. ACM, 15–22.
41. Sana Malik and Leah Findlater. 2013. Punctuation Input on Touchscreen Keyboards: Analyzing Frequency of Use and Mode-switching Costs. *College Park: The Human-Computer Interaction Lab, University of Maryland* (2013).
42. Edgar Matias, I Scott MacKenzie, and William Buxton. 1993. Half-QWERTY: A one-handed keyboard facilitating skill transfer from QWERTY. In *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*. ACM, 88–94.
43. Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. 2015. 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, 2143–2152. DOI: <http://dx.doi.org/10.1145/2702123.2702382>
44. Netflix. Netflix App for Oculus. (????). <https://www.oculus.com/experiences/gear-vr/890562797701371/>
45. Albert Ng, Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul Dietz. 2012. Designing for Low-latency Direct-touch Input. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 453–464. DOI: <http://dx.doi.org/10.1145/2380116.2380174>
46. J Noyes. 1983a. Chord keyboards. *Applied Ergonomics* 14, 1 (1983), 55–59.
47. Jan Noyes. 1983b. The QWERTY keyboard: a review. *International Journal of Man-Machine Studies* 18, 3 (1983), 265–281.
48. Noritaka Osawa and Yuji Y Sugimoto. 2002. Multimodal text input in an immersive environment. In *Proceedings of ICAT*. 85–92.
49. Ivan Poupyrev, Numada Tomokazu, and Suzanne Weghorst. 1998. Virtual Notepad: handwriting in immersive VR. In *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*. IEEE, 126–132.
50. Manuel Prätorius, Dimitar Valkov, Ulrich Burgbacher, and Klaus Hinrichs. 2014. DigiTap: An Eyes-free VR/AR Symbolic Input Device. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology (VRST '14)*. ACM, New York, NY, USA, 9–18. DOI: <http://dx.doi.org/10.1145/2671015.2671029>
51. Robert Rosenberg and Mel Slater. 1999. The chording glove: a glove-based text input device. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 29, 2 (1999), 186–191.
52. Sherry Ruan, Jacob O Wobbrock, Kenny Liou, Andrew Ng, and James Landay. 2016. Speech Is 3x Faster than Typing for English and Mandarin Text Entry on Mobile Devices. *arXiv preprint arXiv:1608.07323* (2016).
53. Nitin Sawhney and Chris Schmandt. 2000. Nomadic radio: speech and audio interaction for contextual messaging in nomadic environments. *ACM transactions on Computer-Human interaction (TOCHI)* 7, 3 (2000), 353–383.
54. Martijn J Schuemie, Peter Van Der Straaten, Merel Krijn, and Charles APG Van Der Mast. 2001. Research on presence in virtual reality: A survey. *CyberPsychology & Behavior* 4, 2 (2001), 183–201.
55. Ben Shneiderman. 2000. The limits of speech recognition. *Commun. ACM* 43, 9 (2000), 63–65.
56. Laviola Joseph J. Varcholik, Paul D. and Charles E. Hughes. 2012. *International Journal of Human-Computer Studies* 70, 10 (2012), 657–672.
57. Keith Vertanen and David JC MacKay. 2010. Speech dasher: fast writing using speech and gaze. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 595–598.
58. Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. Veloctap: 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*. ACM, 659–668.
59. Google VR. Daydream Labs: Drum Keys. (????). <https://www.youtube.com/watch?v=QYwzSEAyN2M>
60. Jacob O Wobbrock. 2007. *Measures of text entry performance*. San Francisco: Morgan Kaufmann.
61. Hisao Yamada. 1980. A historical study of typewriters and typing methods: from the position of planning Japanese parallels. *Journal of information processing* 2, 4 (1980), 175–202.
62. Shumin Zhai and Per Ola Kristensson. 2012. The word-gesture keyboard: reimagining keyboard interaction. *Commun. ACM* 55, 9 (2012), 91–101.
63. Shumin Zhai, Per Ola Kristensson, Pengjun Gong, Michael Greiner, Shilei Allen Peng, Liang Mico Liu, and Anthony Dunnigan. 2009. Shapewriter on the Iphone: From the Laboratory to the Real World. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. ACM, New York, NY, USA, 2667–2670. DOI: <http://dx.doi.org/10.1145/1520340.1520380>
64. Shumin Zhai, Per-Ola Kristensson, and Barton A Smith. 2005. In search of effective text input interfaces for off the desktop computing. *Interacting with computers* 17, 3 (2005), 229–250.
65. Shumin Zhai, Paul Milgram, and William Buxton. 1996. The Influence of Muscle Groups on Performance of Multiple Degree-of-freedom Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '96)*. ACM, New York, NY, USA, 308–315. DOI: <http://dx.doi.org/10.1145/238386.238534>