

Selection-based Text Entry in Virtual Reality

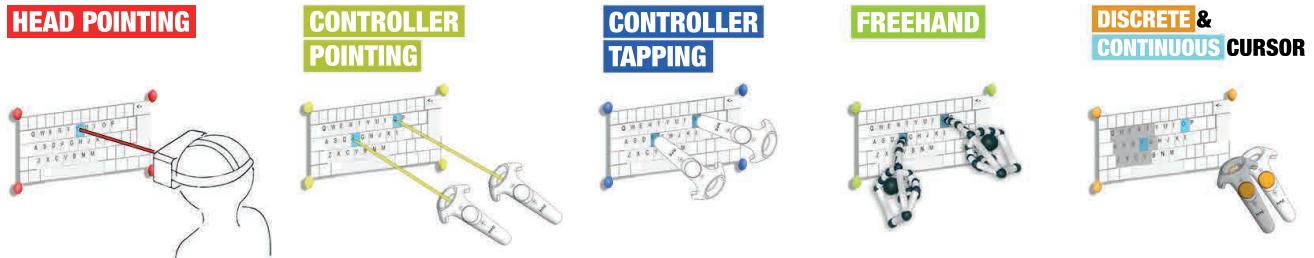
Marco Speicher^{1,3}Anna Maria Feit²Pascal Ziegler³Antonio Krüger^{1,3}¹DFKI GmbH²Aalto University³Saarland Informatics Campus

Figure 1. This figure illustrates our implemented selection-based text entry candidates for VR. From left to right: Head Pointing (HP, red), Controller Pointing (CP, yellow), Controller Tapping (CT, blue), Freehand (FH, green), Discrete (DC, orange) and Continuous Cursor (CC, light blue).

ABSTRACT

In recent years, Virtual Reality (VR) and 3D User Interfaces (3DUI) have seen a drastic increase in popularity, especially in terms of consumer-ready hardware and software. While the technology for input as well as output devices is market ready, only a few solutions for text input exist, and empirical knowledge about performance and user preferences is lacking. In this paper, we study text entry in VR by selecting characters on a virtual keyboard. We discuss the design space for assessing selection-based text entry in VR. Then, we implement six methods that span different parts of the design space and evaluate their performance and user preferences. Our results show that pointing using tracked hand-held controllers outperforms all other methods. Other methods such as head pointing can be viable alternatives depending on available resources. We summarize our findings by formulating guidelines for choosing optimal virtual keyboard text entry methods in VR.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords

Virtual reality; text entry; pointing; mid-air; user experience; task performance.

INTRODUCTION

As text-based communication is rarely studied in VR research regarding text entry performance [14, 35, 36, 43], there is a

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 2018, April 21–26, 2018, Montréal, QC, Canada.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-5620-6/18/04 ...\$15.00.

<https://doi.org/10.1145/3173574.3174221>

need for evaluating the user preferences (e.g. user experience, workload, and motion sickness) and characteristics of VR (e.g. immersion) regarding recent technology and interaction concepts for text entry in VR. Furthermore, design guidelines for text entry are directly adapted from non-VR systems without any further investigation, like head or controller pointing on a QWERTY keyboard. But what if tracked hand-held controllers aren't available, or low physical demand and motion sickness are of particular importance? Does natural (or isomorphic) interaction using fingers or pens for input have any impact on task performance or the user's preference? In this context, isomorphism characterizes the mappings between movements in the real world and their effect in the virtual environment [7, 49]. An isomorphic technique uses one-to-one mappings and is considered as the most natural approach. But traditional text entry might not work in non-desktop 3DUI, because VR users are not fixed in general. To overcome limitations in the tracking space or anatomical constraints, non-isomorphic techniques allow users to interact using "supernatural" metaphors.

To answer these questions, we present a design space for selection-based text entry in VR, based on MacLean et al. [33]. In this paper, we contribute an analysis of this design space using a methodology that forms a basis for the development of VR text entry methods in future VR applications and enables researchers to relate future analyses to ours. The design space of VR text input consists of three key components: questions (about what a design should do), options (answers to questions), and criteria (ways of assessing designs). Under those categories and based on related work, we chose six candidates for text entry. We used the head and controller pointing as the current standard for VR text entry, continuous and discrete cursor control from current 3D game consoles, and the controller tapping and freehand methods for a more isomorphic and direct approach, as it is the standard for non-virtual keyboard interaction. Further, we contribute an evaluation of the

Input Method	Qwerty	Eyes free	Hands	Haptic feedb.	Additional device	Device tracked	WPM in VR	WPM other
Soft button selection	✓	✗	1-2	✗	pointing device or hand	✓	4-7 [14]	33-36 [3]
Mid-air pointing	✓	✗	1-2	✗	pointing device or hand	✓	?	13-19 [35, 40]
Head pointing	✓	✗	0-1	✗	none (dwell) or button	✗	10-15 [48]	4.5 [13]
Eye gaze selection	✓	✗	0-1	✗	eye tracker, none (dwell) or button	✗	?	10-20 [34]
Discrete cursor (e.g. gamepad)	✓	✗	1-2	✓	directional controller or buttons	✗	?	6-7 [45]
Physical keyboard	✓	(✓)	1-2	✓	physical keyboard	✗	24- 40 [28, 43]	40-60 [30]
Finger gestures	✗	✓	1-2	✗	pointing device or hand	✓	6 [14]	22 [41]
Chording (Twiddler)	✗	✓	1	✓	12-button keypad	✗	3 [14]	47 [29]
Multi-tap	✗	✓	1	✓	9-button keypad	✗	12 [14]	20 [29]
Handwriting	✗	✗	1	✓	pen, stylus or finger	(✓)	?	15-20 [30]
Speech	✗	✓	0	✗	microphone	✗	13 [9]	11 [19]

✓: yes ✗: no ?: unknown/not applicable

Table 1. Overview of text entry methods evaluated in VR or potentially usable in VR. We compare the following factors: if the method uses a QWERTY layout, if it can be used without visual attention, how many hands are needed for control, if it provides haptic feedback, if it requires an additional device (e.g. hand-held controller, gloves or hand) or sensor and which one, and if that device is shown in VR. We then give performance estimates in WPM for evaluation in VR and in other conditions, with example references.

six selection-based text entry methods for immersive virtual environments.

We presented them to participants ($N = 24$) in an empirical study to analyze their text entry performance and user preferences against our design space for selection-based text entry in VR. The results showed that the performance, workload and user experience of our implemented pointing methods (head & controller) are above average compared to related work in VR [14, 48] or non-VR [35, 13]. However, Controller Tapping and Freehand performed worse, but better than expected due to their higher technical and physical limitations. Particularly in VR, user experience and task workload turned out to be essential factors for text entry performance. The design space and the evaluated methods provide a solid baseline for comparison of future selection-based text entry methods in VR. Hence, the main contributions of this paper are as follows:

- **Introducing a design space for selection-based text entry in VR**, which includes questions, options and criteria.
- **Evaluating six selection-based text entry methods for VR in an empirical study** with respect to text entry performance and user preference.
- **Providing design guidelines for text entry in VR** based on our design space and experimental results.

SELECTION-BASED TEXT ENTRY IN VR

Typing on a virtual keyboard providing live feedback has a crucial impact on users' typing performance while wearing an HMD [43, 28]. Here, physical keyboards would be more useful for text-heavy interactions but not for mobile use or while standing, which is normally the case in common VR setups (e.g. HTC Vive), although the implementation cost is low and no tracking devices are required. Therefore, methods using VR controllers (e.g. pointing or cursor), or no controllers at all (e.g. head pointing [48], FaceTouch [15] or speech [9]), would be more suitable. Apart from speech and head pointing, none could approach the performance of a physical keyboard (see Table 1), whereas several methods could be more useful, in particular with regard to the user's preferences instead of pure performance. We thus investigated how text entry using a virtual keyboard could be supported in VR with regard to both metrics, performance and preference.

Currently there is no standard method for VR text entry and current commercial systems implement their own techniques. However, most of them are based on a selection paradigm, where characters are selected sequentially on a virtual keyboard floating in front of the user. This paradigm is familiar to users from other systems, suitable for mobile use (i.e. while standing or walking) and thus easy to adapt for VR. Since there is no taxonomy of text input methods using virtual keyboards in VR which we can refer to, we chose the term “selection-based” as used by prior work [35]. Nevertheless, performance and preference with such systems could greatly differ depending on the input method used for selection.

In Table 1 we categorize common text entry techniques based on their input method and compare different aspects of their design and achievable performance. Methods differ in the number of hands they require, if they make use of the QWERTY layout, if they require visual attention, if they provide haptic feedback, and what input device they require and if it must be tracked in VR. The given performance is based on example prior studies conducted in VR, as well as others. We observe that performance studied outside VR is generally higher. However, note that these are estimates from prior work, which are not consistent in their methodology and are thus not directly comparable. Moreover, some methods have not been studied in VR at all. Besides the lack of comparative performance evaluations in VR as well as other contexts, little is known about users' preference of these methods, including VR specific factors such as immersion or motion sickness. Therefore, this paper compares six common selection-based text entry methods using a virtual keyboard, covering a wide range of design aspects, by the same rigorous methodology for assessing performance and users' preference in VR based on literature review.

DESIGN SPACE FOR SELECTION-BASED TEXT ENTRY

One purpose of this research is to develop principles for the design and evaluation of effective and usable selection-based text entry in immersive virtual environments. So, we introduce a design space based on MacLean et al. [33], which is a standard approach to represent design rationale and uses a semi-formal notation, called QOC (Questions, Options, and Criteria), for example to fill the gap between VR and selection-based text

entry. The questions as the key parts of this design space approach identify key design issues and options providing potential answers to the questions, and criteria for assessing and comparing the options. As we wanted to build on prior work, we decided to use Markussen et al.'s [35] design space for selection-based text entry as a basis and adapted their QOCs for investigating selection-based text entry in VR using consumer hardware.

Questions and Options

There is only a few research in comparing text entry methods in VR [9, 14, 48], so we believe that there are still open questions, and especially designers of VR applications are still provided with little guidance as there is no common link between VR and text entry.

Q: Which keyboard layout? The QWERTY layout has been the standard keyboard layout of the last century [38], which supports the assumption that layout would have a positive impact on preference and performance, although it is not at all superior for expert performance [31, 29]. Even an imaginary “optimal” layout with faster keystrokes would only differ to a small extent [21]. Due to its habitual use in daily life, novel layouts (e.g. Dvorak [11] or OPTI [32]) would require large amount of adaptiveness and instrumentation until the first signs of improvement. However, Bowman et al. [9] suggest using the QWERTY layout for 3DUIs, if symbolic input will be infrequent or if most users will be novices. In summary, we would recommend using the QWERTY layout for VR text entry, if user preference is more important than performance.

Q: 2D or 3D? Bowman et al. [9] state that reducing the number of degrees of freedom (DOF) in 3DUIs is crucial when high-precision interaction is needed, in particular for text entry, a task that has been 2D for decades. So, 2D text entry methods could have a positive impact on intuitiveness, especially when using the QWERTY keyboard layout. But using 2D interfaces in 3D environments can decrease the awareness and feeling of presence (e.g. by taking off the HMD to use a physical keyboard or overlay it in the virtual scene [43, 28, 36]). But as selection-based text entry involves interaction with a virtual keyboard, there is also the question of how to represent the keyboard in the environment. While 2D approaches can mimic typing on a surface and enable users to imagine a keyboard floating in front of them, 3D would suggest more interactivity and therefore increase the user experience and immersion. Considering the form factor, curved keyboards might be more ergonomic, whereas non-curved are expected to be more intuitive, because its familiarity (e.g. smartphones, touchscreens). However, the question of curved or not needs to be explored in more detail, but is beyond of the scope of this work.

Q: Typing in relation to what? Another question is whether typing should be in relation to an explicit reference point for gestures or selections. Touch-based surfaces (e.g. a Vive controller pad) can implicitly maintain a reference point for the user, whereas this is not the case with mid-air interactions. Mid-air techniques (e.g. keyboard input using fingers or stylus) make use of absolute reference points [37]. Here, the virtual representation of the keyboard is placed at a fixed location in the environment. By using a room-tracking VR system the

user can walk around or even through the keyboard, which can increase the feeling of presence. As no empirical study on VR specific text entry method has covered this question yet, it is hard to determine the impact of this question on the criteria. In contrast to a fixed absolute position of the keyboard, it could also be positioned relative to the user, e.g. to the head for constant distance while typing or to the non-dominant hand while typing with the dominant [16]. While relative reference points make the interaction more flexible, they can cause high instrumentation and complexity of the interface itself, which consequently influences the performance and preference.

Q: Position and size of the keyboard? The size of the keyboard representation matters especially for distance-dependent text input methods (e.g. direct input) but also for ergonomic reasons. Current text input methods for VR don't allow one to change position or size of the representation. Bachynskyi et al. [4] identified several clusters in input space for pointing gestures and advise to split the input space for right and left hand (if possible) and make the input representation fit the lower and peripheral input space. Nevertheless, larger input representations require a less precise tracking technique, but may afford more head movements, which consequently could result in higher motion sickness and workload. The immersion, however, is driven by the interactivity, i.e. the absence of customization could reduce the feeling of presence when the user cannot manipulate the virtual world as expected [26].

Q: Feedback or not? Although typing on a virtual keyboard lacks real tactile feedback, it can be compensated for by using vibrator feedback from VR controllers, or pseudo-haptic feedback when pressing the virtual keys [25]. Here, the 3D key moves in depth when intersecting with the user's finger or controller, which simulates a physical button press in combination with auditory and visual feedback. Nevertheless, when using indirect keyboard interaction, users need feedback on tracking of their controller and cursor movements. Pointing-based methods using tracked hand-held controllers usually comes with the drawing of the corresponding ray intersecting the keyboard. However, at least cursor visualization is generally of high importance regardless of the input type, performance and user's preference, because otherwise it can be very confusing for the user. While visual feedback can support the perspicuity of novel methods, it may increase intuitiveness but also influence the feeling of presence. But less feedback can result in higher error rates and ‘trial and error’ learning.

Criteria

Text entry in VR-based applications require validation of findings and answers to new questions and should therefore rely on certain criteria. In particular, novel VR apps and their interaction techniques for text entry are mostly based on guidelines and standards from non-VR areas. The following design criteria for selection-based mid-air text entry methods in VR will enable designers to build more usable and effective VR systems including text input and help to move towards a stronger theoretical basis and more principled design guidelines for virtual reality text entry using a virtual keyboard. We combine the metrics of text entry performance, 3DUI evaluation [7], and the characteristics of virtual reality.

Text Entry Performance. These metrics indicate to what extent users are able to cope with the task and interaction method. Text entry evaluations usually focus on the same objective statistics, *speed* and *accuracy*. Nevertheless, repeated trials are necessary to generate great volumes of data, consisting of presented text (the stimulus, i.e. what they were asked to enter) and transcribed text (what they actually entered). When conducting text entry experiments, where participants have to enter multiple sentences in a row, the words-per-minute (WPM) for measuring speed should be chosen, which is also the current standard for text entry evaluations [30]. We further recommend determining accuracy by the error rate.

User Preference. The user's preference usually consists of subjective feedback, such as *user experience*, *workload* and *motion sickness* [7]. User experience includes classical usability (efficiency, perspicuity, dependability) and special user experience aspects (attractiveness, novelty, stimulation) [24]. Task workload includes mental, physical and temporal demand, as well as subjective effort, performance and frustration ratings [17]. Finally, motion sickness ratings reflect how nauseous, dizzy or unpleasant the user feels [12].

Characteristics of Virtual Reality. Furthermore, we also want to include characteristics of VR, like *immersion*, interactivity, and presence [26]. Immersion and presence can be combined into one characteristic, i.e. to what extent the user's senses are isolated from the real and stimulated by the virtual world, and the subjective experience of being in one environment, but physically situated in another. Interactivity indicates to what extent users can participate in manipulating virtual content in real time.

EVALUATED TEXT INPUT TECHNIQUES

This section describes how each of the six candidates for VR text entry is used to select a character from the user's perspective, the important parts of the implementation, what parts of the design space this method covers, and finally the commonalities and differences among methods according to our design space.

Head Pointing

Pointing is one of the most fundamental patterns for selection [20] and is usually done with the head or hand-held controller. The basic idea is to follow an imaginary ray from the point of view (or object of interest) through the virtual environment. In our approach of *Head Pointing (HP)*, we extend a ray from the main camera position towards the viewing direction provided by the SteamVR Unity plugin (see Figure 1). If the first intersected object is a character, it can be selected by a user-controlled trigger (button or dwelling time).

Controller Pointing

The implementation of our *Controller Pointing (CP)* method is analogous to HP but allows bi-manual pointing using both controllers. Moreover, it is designed for scenarios where controller tracking is available and extends a ray from each controller (see Figure 1). It is actually not novel, because it was used for the Nintendo Wii via optical tracking using infrared. Here, the user controls the cursors' positions on the virtual keyboard by

simply pointing at the character to select it, which is further visualized by changing the key's color. The character selection is then done by pressing a trigger button.

Controller Tapping

The *Controller Tapping (CT)* method is implemented within the pen & tablet metaphor [2] and provides a more isomorphic, direct and realistic interaction than pointing. Text entry using a digital pen is less common but advancing fast with the rise of the Apple Pencil and several other tablets which can be operated quickly and accurately with a stylus. In our approach, the HTC Vive controllers are used for tapping the virtual keys by reversing and holding them like digital pens (see Figure 1). In contrast to pointing, this method requires physical manipulation, i.e. the virtual keys need to be pushed physically with the tracked hand-held controllers.

Freehand Technique

Our most isomorphic and realistic text entry candidate is the *Freehand (FH)* technique, where the user's fingers are tracked to type directly on a virtual keyboard. This technique doesn't require any tracked hand-held controllers, but instead the tracking of the fingers (e.g. using gloves or Leap Motion) [39]. In our approach, we decided to use the Leap Motion, which is mounted on the HMD. It is worth to mention that the Leap Motion provides no accurate tracking due to its technical limitations [44] and could therefore have a negative impact on the performance and preference. The potential gap in tracking precision compared to the other methods was therefore included in our discussion of the experimental results. However, we have chosen the Leap Motion as it is the current most affordable and available method for finger input and visualization.

Pad-based Discrete Cursor Control

The majority of gaming consoles use text entry methods using the attached controller, more precisely the directional pad (d-pad) or thumb-sticks. In most instances, the text input is performed by controlling one discrete cursor over a virtual QWERTY keyboard for character selection, and conforming the input by a trigger button. We transferred this to the HTC Vive controller, which is also equipped with a d-pad and is controlled by the user's thumb. Furthermore, we improved the standard method for bi-manual input using both controllers simultaneously. Finally, in our implemented *Discrete Cursor (DC)* method, the user controls a cursor per controller for character selection, whereas each was intended for the corresponding half of the keyboard (see Figure 1). In order to separate the two keyboard sections and to make it easier for the user, the left keys were colored darker than the right ones. The text input is then done by pressing the corresponding button.

Pad-based Continuous Cursor Control

The second method using the controllers' d-pad is the so-called *Continuous Cursor (CC)* method, which extends the functionality of DC. The difference between these two methods is that CC allows continuous control of the cursors. Here, we use the 2D thumb position on the touch pad and set the cursor of the corresponding keyboard half accordingly. Pressing a touch pad triggers the text input. Apart from that, the mode is analogous to DC.

Commonalities and Differences among Methods

In the following, we discuss what the six candidates for text entry in VR using a virtual keyboard have in common, as well as what divides them, according to our design space. First, we use the common QWERTY layout and fixed keyboard position and orientation for all methods according to prior work [38, 9], as our methods should aim at non-experts. The optimal position, orientation and size of the virtual keyboard representation is calibrated in advance for each user and method. In Table 2 we list which decision was made for each factor. The main limitation of pointing (HP, CP) is that the user cannot perform eyes-free text entry, which can cause higher mental and also physical demand (e.g. neck pain for HP, hand tremor and gorilla arm for CP). Despite the major advantage of direct mid-air input (FH), i.e. seeing the hands or controllers, there are several challenges including the lack of a touch sense: gorilla arm and the line-of-sight requirement [20]. In addition, FH is the only method where no hand-held controller is used at all, but instead a camera for tracking the user's fingers. Consequently, FH involves stable and solid tracking of the user's fingers, which is still a challenge in current research [44]. Besides that, CP and CT use the position and orientation of the tracked hand-held controllers, while DC and CC get along without tracking. Although techniques using non-tracked hand-held controllers (e.g. gamepad or joystick) aren't physically demanding in principle, there is still a high risk of suffering the so-called 'texting thumb' pain. Apart from CT and FH, all methods use a button to confirm text entry. Furthermore, we use visual and auditory feedback for all methods except for FH and CT. Here, we use a combination of visual, auditory and pseudo-haptic feedback when selecting a character in order to amplify the amount of realism. Finally, while FH and CT involve 3D manipulation of the controllers or hands, all other methods control 2D cursors on a planar virtual keyboard.

EMPIRICAL STUDY

We conducted a controlled laboratory experiment to compare the six text entry methods with respect to the task performance (speed, accuracy) and user preference (user experience, workload, immersion, motion sickness). Most VR applications

	HP	CP	CT	FH	CC	DC
Which input device?						
Eyes free?	✗	✗	✗	(✓)	✓	✓
Hands?	0-1*	1-2	1-2	1-2	1-2	1-2
Hand-held controller?	✗*	✓	✓	✗	✓	✓
Device tracked?	✗*	✓	✓	✓**	✗	✗
Trigger button?	✓	✓	✗	✗	✓	✓
Feedback or not?						
Colored cursor?	✓	✓	✗	✗	✓	✓
Pseudo haptics?	✗	✗	✓	✓	✗	✗
2D or 3D?						
Character selection?	2D	2D	3D	3D	2D	2D
Key representation?	2D	2D	3D	3D	2D	2D

Table 2. This table shows which decision was made for each method with regard to our design space questions and options. (*) No additional controller is needed for HP except for a button to trigger the input, as head orientation is given for any VR hardware. (**) Fingers are tracked by an external camera, e.g. Leap Motion.

require the user to infrequently enter short phrases. Thus, we were interested to compare the methods in a short text entry task, rather than a longitudinal study.

Task

The task was to transcribe five phrases (trials) with each text entry method as fast and as accurately as possible. Error correction was allowed using backspace. All phrases were randomly chosen from a set of 100 memorable phrases from the Enron corpus [42], with 20–28 characters each. We have chosen the Enron towards MacKenzie's phrase set, because of its higher validity for mobile text entry. Prior work has shown that both are comparable in terms of memorability, performance, and errors [23].

Design

The experiment was a within-subjects design, with one independent variable (Input Method) with six levels and six dependent variables related to the performance (speed, accuracy) and preference of users (user experience, workload, motion sickness, immersion). The input method conditions were counterbalanced using a Latin square. Aside from training, this amounted to:

$$24 \text{ participants} \times 6 \text{ input methods} \times 5 \text{ phrases} = 720 \text{ trials.}$$

Conditions

We evaluated the following six techniques to transcribe phrases in VR using a virtual keyboard (see Figure 1):

- *Head Pointing (HP)* — the participant selects a character by pointing to it with her head. Text input is performed via trigger button.
- *Controller Pointing (CP)* — the mode is analogous to mode HP, but the participant uses two tracked hand-held controllers for pointing this time.
- *Controller Tapping (CT)* — the controllers are used as pens by reversing and holding them like pencils. Text input is performed by pressing the keys with the controller endings.
- *Freehand (FH)* — the mode is analogous to mode CT, but the participant used her fingers for text entry and no additional hand-held controllers.
- *Discrete Cursor (DC)* — the user navigates the cursor from one key to another by pressing buttons representing one of the four directions (here: edges of the touch pad). Text input is performed via trigger button.
- *Continuous Cursor (CC)* — the mode is analogous to mode DC, but the participant uses continuous cursor control instead of a discrete movement selection.

Participants

A total of 24 unpaid participants (19 male, 5 female) volunteered in this experiment, aged between 22 and 29 years ($M = 25.29$, $SD = 1.89$). Two participants preferred the US QWERTY keyboard layout; the rest preferred the German QWERTZ layout. All participants rated themselves as able to read and copy English sentences ($M = 3.67$, $SD = 0.65$; on a scale from 1 (beginner) to 5 (native speaker)). All participants were right-handed. 28.6% had a visual impairment (glasses or contact lenses), but no participant was color blind. As the HTC

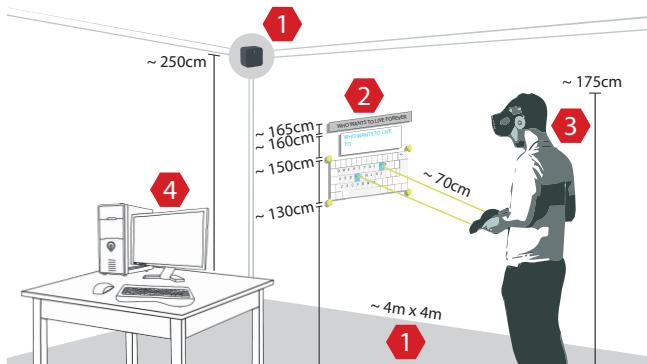


Figure 2. This figure illustrates the experimental setup. (1) HTC Vive optical tracker (at 2.5m) and tracking space with $4 \times 4m^2$. (2) Virtual keyboard, stimulus and text input field. (3) Participant wearing HTC Vive and tracked hand-held controllers. (4) PC for experiment control and filling out questionnaires.

Vive allows the user to wear glasses, no further adjustment was needed. On average, participants rated their experience in VR and that with each text entry method on a scale from 1 (novice) to 5 (expert) as follows:

- VR: $M = 2.57$, $SD = 1.50$
- Head Pointing: $M = 1.52$, $SD = 1.01$,
- Controller Pointing: $M = 2.71$, $SD = 1.52$
- Controller Tapping: $M = 2.57$, $SD = 1.34$
- Freehand: $M = 1.90$, $SD = 1.16$
- Gamepad/Joystick Cursor Control: $M = 3.81$, $SD = 1.14$

Apparatus and Setup

The VR system used an HTC Vive and ran on a Windows 10 machine with Unity 5.4. A standard desktop computer was used with an i7 CPU, 16 GB RAM and a Nvidia GeForce GTX 980Ti graphics card to fill out the questionnaires and control the experiment. Besides the Freehand technique, where the Leap Motion device was used to track the fingers, the two Vive controllers were used for the other methods, because they are tracked hand-held controllers equipped with d-pad and trigger buttons. The HTC Vive optical trackers (or lighthouses) were installed about 2.5m above the ground in two opposite corners to span a maximum Vive tracking area of approximately $4m \times 4m$. The participants were standing in its center while performing the tasks, as shown in Figure 2.

The virtual environment consisted of a virtual representation of a standard QWERTY keyboard in the participants' interaction zone at $1.3 - 1.7m$ in a comfortable distance for mid-air interaction, a text area for the output at eye sight, and the stimulus above (see Figure 3). The position and orientation of the keyboard could be adjusted for each participant. Apart from those three elements, the virtual environment showing a sunset was empty, which made the environment more immersive but not distracting [9]. Visual feedback was similar for all methods. Here, hovering over a virtual key highlights the key in blue and symbolized the virtual cursors. For auditory feedback, the participants wore headphones and got audio feedback when selecting the virtual keys.

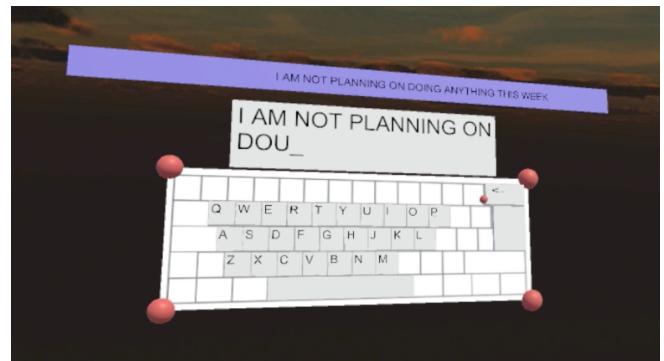


Figure 3. This figure shows the virtual environment including stimulus (purple), text input field, and the virtual keyboard.

Procedure

The experiment started with a 5-minute SteamVR tutorial to get familiar with the headset and the controllers. It explains the bounding box of the tracking area how to use the controllers and their corresponding buttons. After a short break, the main part of the experiment started, where the participant was to perform all six tasks in Latin-square order, which lasts about 15-30 seconds on average for one of at least five trials per task. Before each condition, the interaction technique was explained and practiced in a warm-up phase of about 5 minutes. Participants received only minimal instructions about the functionality of the different interaction techniques, so that no explicit conceptual model was assigned to them. The participant was instructed to transcribe phrases as fast and as accurately as possible. Consequently, he or she was allowed, but not forced, to correct errors by using backspace.

After each set of trials, the participant was asked to take off the HMD and fill out the post-task questionnaires to gather subjective feedback about the user's preference. After all trials were performed and post-task questionnaires were filled out, the participant was asked to fill out a final questionnaire collecting demographic data (age, gender, experience). Overall, the experiment took about 60 minutes per participant in total.

Evaluation Metrics

We measured task performance in the form of objective data (speed, accuracy) and collected data describing users' preference to the methods, including subjective feedback (user experience, workload, motion sickness, immersion).

Task Performance

For each participant, we measured text input speed and accuracy by calculating the average words per minute and error rate across the five entered phrases, in accordance with the common standards in text entry research [46], as follows:

- **Words per minute (WPM)** was computed by dividing the number of transcribed words (any 5-character string) by the time it takes to transcribe the text, formally:

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

where S is the time (in seconds) from the first to the last key press and $|T|$ is the number of characters in the transcribed text.

- **Error rate (%)** was computed by calculating the minimum string distance (MSD) between the presented and transcribed text and dividing it by the larger number of characters, formally:

$$\frac{100 * \text{MSD}(P, T)}{\max(|P|, |T|)} \quad (2)$$

where P and T denote the presented and transcribed text. MSD is calculated using Levenshtein's algorithm [27].

User Preference

We collected a variety of subjective feedback to assess *user experience* and *workload*, but also *immersion* and *motion sickness*, important in VR applications. Therefore, we used the following questionnaires:

- **User Experience Questionnaire (UEQ):** consists of 26 short questions to measure user experience [24]. The scales cover classical usability (efficiency, perspicuity, dependability) and user experience aspects (attractiveness, novelty, stimulation). The higher the score the better the experience.
- **NASA TLX:** a commonly used questionnaire to asses task workload based on six factors (mental, physical and temporal demand, effort, performance and frustration) [17]. The lower the rating the lower the workload.
- **Motion Sickness Assessment Questionnaire (MSAQ):** assesses the motion sickness based on 16 questions rated on a 9-point scale [12]. The scales cover four dimensions of motion sickness, which were defined as gastrointestinal, central, peripheral, and soplete-related. The lower the score the better.
- **Slater-Usoh-Steed Questionnair (SUS):** a commonly used questionnaire to measure the user's immersion and presence in a virtual environment. The SUS Count shows the amount of 6 or 7 scores in average amongst the 6 questions, while SUS Mean uses the mean score across the 6 questions instead. The higher the score, the higher the immersion and presence.

RESULTS

Throughout this results section and in the following discussion we use abbreviations and color indications for the six text input methods we tested: **Head Pointing (HP, red)**, **Controller Pointing (CP, yellow)**, **Controller Tapping (CT, blue)**, **Freehand (FH, green)**, **Discrete (DC, orange)** and **Continuous Cursor (CC, light blue)**. The results of the experiment were analyzed by us using IBM SPSS Statistics 24.

Task Performance

The task performance metrics include quantitative measurements such as speed and accuracy (or precision). These metrics indicate to what extent users are able to cope with the task and interaction method. They are computed per participant and input method as the average over the five trials.

WPM ranged between 5.31 ($SD = 1.05$) for DC and 15.44 ($SD = 2.68$) for CP (see Figure 4). A univariate ANOVA

showed significant differences between the methods regarding WPM ($F(5, 120) = 36.28, p < 0.01, \eta^2 = 0.60$). Bonferroni corrected pairwise comparisons showed significant differences between all methods, except FH-HP, FH-CC, and HP-CC. So, FH, CC and HP can be seen as one group with relation to speed. Overall, the average corrected error rate was low with CP as the best ($M = 0.01, SD = 0.01$) and FH the worst method ($M = 0.76, SD = 0.08$). There was a significant effect between the input methods ($F(5, 120) = 8.20, p < 0.01, \eta^2 = 0.26$).

User Preference

User preference metrics consist of subjective feedback concerning the user experience, task workload, immersion and motion sickness.

Averaged over all input methods, the user experience was rated at 0.41 ($SD = 1.13$) on average on a scale between -3 (very bad) to 3 (excellent). A univariate ANOVA showed significant differences between them ($F(5, 120) = 5.95, p < 0.01, \eta^2 = 0.21$). The pointing techniques were rated best in total (CP: $M = 1.17, SD = 0.78$; HP: $M = 0.67, SD = 1.16$). CP outperformed all other methods across the UEQ subscales, even 'excellent' in terms of perspicuity ($M = 1.61, SD = 0.50$),

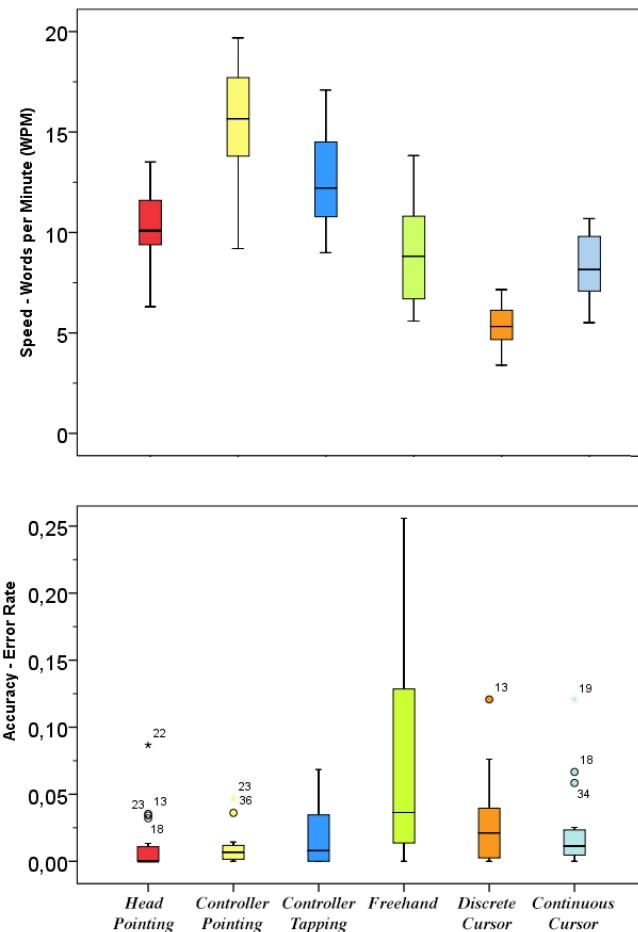


Figure 4. The upper figure show the speed measurements, given in words per minute (WPM). Below, the corrected error rate measurements, given in percent (%).

Method	WPM	Error Rate (%) (corrected)	User Experience	Physical Demand	Frustration	Subjective Performance
Head Pointing (HP)	III: 10.20 ± 1.91	II: 1.15 ± 2.14	II: 0.67 ± 1.16	IV: 41.43 ± 0.00	II: 41.07 ± 0.00	II: 32.86 ± 0.00
Controller Pointing (CP)	I: 15.44 ± 2.68	I: 0.97 ± 1.19	I: 1.17 ± 0.78	III: 37.86 ± 27.82	I: 28.10 ± 24.42	I: 28.33 ± 20.88
Controller Tapping (CT)	II: 12.69 ± 2.27	III: 1.94 ± 2.22		IV: 0.56 ± 1.17	VI: 51.90 ± 26.05	III: 42.86 ± 32.12
Freehand (FH)	IV: 9.77 ± 4.78	VI: 7.57 ± 7.69	III: 0.55 ± 1.18	V: 46.43 ± 26.28	IV: 50.71 ± 27.85	IV: 40.00 ± 29.41
Discrete Cursor (DC)	VI: 5.31 ± 1.05	V: 2.79 ± 3.02	VI: -0.40 ± 0.88	II: 30.71 ± 24.15	VI: 62.14 ± 23.64	VI: 54.05 ± 28.31
Continuous Cursor (CC)	V: 8.35 ± 1.58	IV: 2.15 ± 2.93	V: -0.07 ± 0.92	I: 28.81 ± 21.62	V: 57.38 ± 28.44	V: 47.86 ± 24.22
$F_{(5,120)}$	36.28	7.00	5.95	3.26	3.76	3.02
p	< 0.01	< 0.01	< 0.01	< 0.02	< 0.01	< 0.04
η^2	0.60	0.23	0.21	0.11	0.14	0.10

Table 3. Objective measurements and subjective feedback ratings with significant differences between the text input methods. The best method per scale is visualized in dark green, the second in green. Furthermore, the ranking for each scale is represented by Roman numerals.

except for the novelty aspect, where FH had a slightly better rating (see Figure 5).

The overall task workload was rated at an average of 47.10 ($SD = 20.94$). On average, CP was rated the best ($M = 38.43$, $SD = 22.32$) and DC ($M = 52.67$, $SD = 18.78$) the worst. We found no significant differences in overall task workload between the six input methods using univariate ANOVA ($p = 0.35$). But considering the NASA-TLX subscales, a multivariate ANOVA showed significant effects and differences between the methods regarding physical demand ($F(5, 120) = 3.26$, $p < 0.02$, $\eta^2 = 0.11$) and frustration ($F(5, 120) = 3.76$, $p < 0.01$, $\eta^2 = 0.14$), as well as performance ($F(5, 120) = 3.02$, $p < 0.04$, $\eta^2 = 0.10$). Details can be seen in Table 3.

The motion sickness total score was 19.8% on average ($SD = 1.32\%$). On average, HP was rated worst ($M = 22.29\%$, $SD = 14.40\%$) and FH best ($M = 17.99\%$, $SD = 8.91\%$). A multivariate ANOVA with all four MSAQ factors (G: gastrointestinal, C: central, P: peripheral, S: soplete-related) as dependent variables and single task as factor was conducted. It showed that there were no significant differences between the single tasks regarding the MSAQ factors (G: $p = 0.85$; C: $p = 0.47$; P: $p = 0.94$; S: $p = 0.81$).

The overall SUS count was 1.42 ($SD = 1.77$) and SUS mean was 4.33 ($SD = 1.19$) on average. However, the SUS counts for CT ($M = 1.62$, $SD = 2.04$) and FH ($M = 1.57$, $SD = 1.78$) are slightly higher on average than the rest. Nevertheless, the SUS mean is slightly higher for CP ($M = 4.50$ on average, $SD = 0.86$) than for CT ($M = 4.40$, $SD = 1.30$). However, there were also no significant differences between the methods regarding immersion ($p = 1.0$).

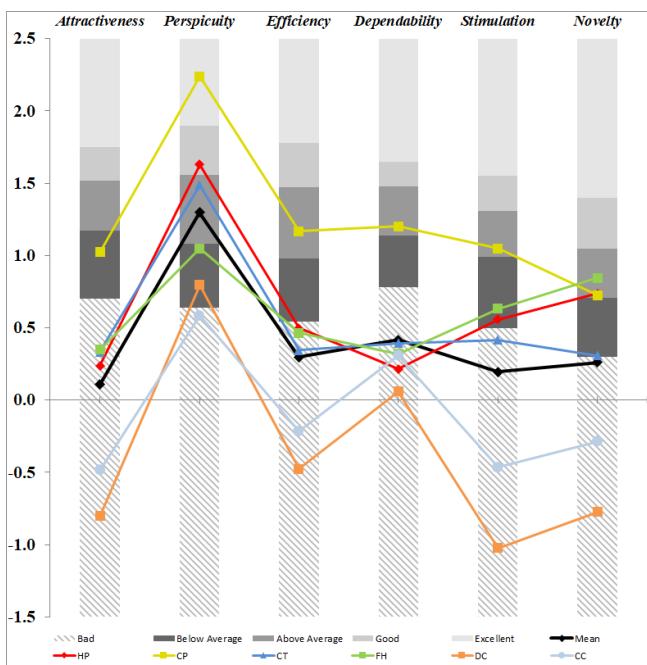


Figure 5. User Experience Questionnaire (UEQ) ratings with respect to comparison benchmarks.

DISCUSSION

VR hardware (HMDs and controllers) are now widely available and affordable. But the question of new VR approaches to text entry has still not been widely explored. This research has thus attempted to explore this area by investigating task performance and user preferences in VR. In this section, we discuss the results, together with our observations and present a design decision tool that will be beneficial to future researchers and designers wanting to build on our findings.

Task Performance

Our results show that 15.4 WPM on average for CP can compete with the comparable QWERTY based pointing approaches from prior work [40, 35]. Although CP outperformed all other methods with regard to speed, tracked hand-held controllers are still required for this method. The controller-less alternative HP was with 10.2 WPM within the scope of the related VR head pointing approach from Yu et al. [48] and is comparable to the 6–13 WPM for speech input [19, 14]. Furthermore, HP was even faster than other head pointing techniques in non-VR studies [10, 6], which could be explained by the benefits VR involves.

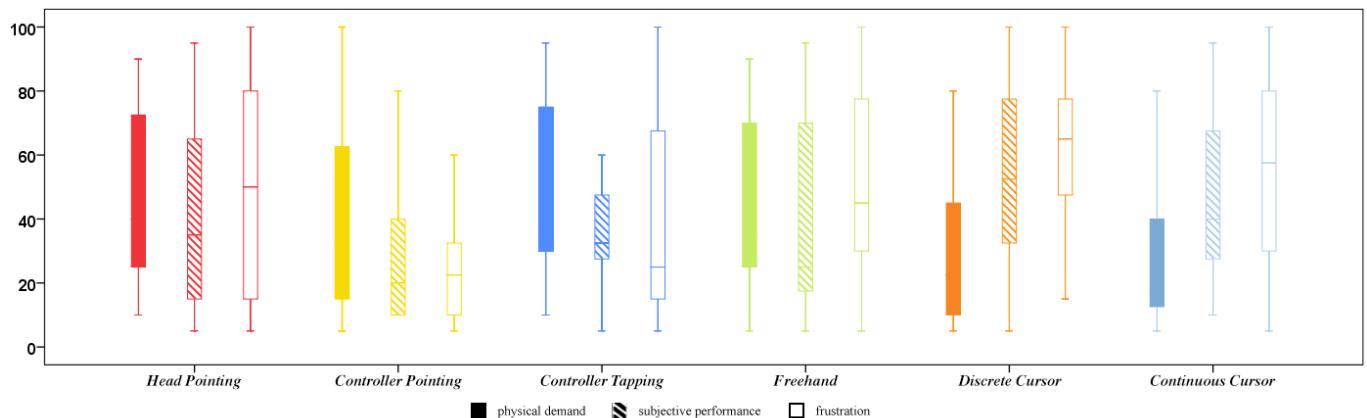


Figure 6. This figure shows the NASA subscales with significant differences between the six text input methods. Non-significant subscales (mental demand, temporal demand, and effort) are not shown for better clarity.

However, character selection by pointing is usually not appropriate when realistic interaction is required, and controller or hand pointing can be imprecise due to natural hand tremor [22]. Concerning the isomorphic candidates, FH performance measures were relatively low, because the hardware we used couldn't deliver a satisfying experience. Keyboard input using optically tracked fingers implies crucial challenges (occlusion, accurate tracking sensitivity), so FH is strongly confounded by the accuracy of the Leap motion. Overall, we couldn't confirm the assumption that FH would have an effect on performance. Nevertheless, the technical and physical limitations of finger tracking techniques, especially for the Leap Motion device we used, have still a crucial impact on the task performance. In consideration of the fact that the accuracy attainable by the human hand is around 0.4 mm on average, the Leap Motion achieved 1.2 mm on average, whereas comparable consumer controllers (e.g. Kinect) were not able to achieve this accuracy [44]. But with 9.8 WPM on average, FH was faster compared to the related studies with 6 WPM [14].

User Preference

User preference is an aspect, which is not considered sufficiently within text entry in VR research. In the following discussion, we want to make the subjective feedback and observations more transparent.

User Experience

Considerable methods due to their good usability and experience ratings are CP, HP and FH. But only the ratings for CP can be seen as above average to good, whereas HP and FH are rather below average. Because CT uses the same input device as CP and performed worse in all other measures (speed, accuracy, etc.) it can be disregarded if tracked hand-held controllers are available. Even though FH performed better in WPM than other studies found [9, 14], which could be explained by its naturalism and realism. However, the DC and CC methods were rated worse, so we would only consider them if the other methods are not possible at all. FH as the most natural way to type text on a keyboard has a positive impact on user experience, especially because participants liked the novelty and stimulation of the method compared

to the others. In summary, we can claim that using tracked hand-held controllers (e.g. CP or CT) result in better user experience, while pad-based cursor techniques (CC, DC) should be completely disregarded concerning user preference.

Workload

Due to Bowman et al. [8], natural (or isomorphic) interaction and especially mid-air writing provide little additional productivity, but could make the task more complicated and unnecessarily cumbersome. In addition, gestural interaction normally involves more muscles than other interaction techniques [5]. Considering the workload ratings of our mid-air and isomorphic techniques, FH and CT had a negative effect on physical ease. Here, participants needed to interact with the virtual keyboard in mid-air at an uncomfortable height, which consequently resulted in higher workload. So, if physical demand is a decision-making factor, mid-air techniques shouldn't be considered, due to potential gorilla arm fatigue.

But HP also resulted in high physical demand ratings and participants complained about neck pain and slight dizziness, which could have been slightly reduced by using eye tracking instead. CP was rated better, but still worse regarding physical demand even though the user needs to lift her arms at least to waist level. However, as the duration of the experiment, more precisely the single tasks, was not very long, the infamous 'gorilla arms' couldn't become a severe problem. Moreover, while pointing-based techniques do not enforce mid-air interaction, there is still a potential to suffer from hand tremor when using controllers at waist level, or neck pain if only head pointing is involved. Two participants complained about slight hand tremor after performing the CP task. However, due to the significantly better frustration and performance ratings, HP and CP should be always preferred to mid-air or pad-based text entry methods, if physical demand or realism can be neglected.

Nevertheless, tracked hand-held controllers and bare-hand input combined with appropriate feedback cues can help to make spatial relationships seem more concrete to the user and enhance presence by simulating physical interaction [18]. Although those problems can be bypassed when using Vive controllers within their larger tracking space, there is still one

potential drawback to it all: It is still unknown, whether current VR controllers can match the immersive quality of virtual hands and fingers visualization. Altogether, the experimental results would appear to give priority to all other methods over CC or DC. But disregarding performance and all other measures, physical demand ratings for pad-based methods were significantly lower than all others. When using non-tracked hand-held controllers (e.g. gamepad, joystick [47] or smartphone [1]), the user doesn't have to lift her arms or move her head. Only the danger of suffering a ‘texting thumb’ remains. However, this doesn't apply for pad-based methods using a d-pad on the HMD itself (e.g. Samsung Gear VR).

Motion Sickness & Immersion

HP resulted in slightly (not significantly) higher motion sickness than all other methods. Of course a comparison regarding user preference of related non-VR techniques needs to be done in order to say more. However, there were no significant differences between the tested methods regarding immersion and motion sickness, which indicates that text entry in VR has no impact on immersion or motion sickness. Nevertheless, two participants stated that they would have preferred to have the keyboard on a virtual desk instead of typing on a floating virtual keyboard, which broke the immersion for them.

Design Guidelines for Text Entry in Virtual Reality

We wrap up this section with a set of general design guidelines for text entry in VR using a virtual keyboard, highlighting the major points discussed earlier in the design space, the experimental results and lessons we learned while moving from the task of text entry to the design and development of a VR methods. When designing for experiences in VR a new set of design considerations comes into play than when designing for 2D interfaces. To help upcoming VR designers and developers of VR text input to create experiences that don't frustrate or make users feel nauseous, we created the following decision support tool based on Gonzales et al. [14] to guide the work (see Figure 7). Based on our findings, their tool for text input needs to be updated. The decision which input device to choose will restrict the choice of techniques, if additional devices (e.g. gloves, cameras, keypads, etc.) or device tracking are needed. Nevertheless, hand-held controllers with at least one button to trigger input are recommended because of their robustness and familiarity. However, if the first choice ‘User needs their hands free’ is answered with yes, or no hand-held controller is available, Head Pointing using dwelling time should definitely be considered in addition to speech recognition. Speech has major drawbacks like recognition problems, privacy issues, and error-correction problems [47]. Regarding task performance metrics including speed and accuracy, as well as user preference, Controller Pointing performed significantly better than all other methods. However, if no tracked hand-held controller is available, the next method to choose would be HP, except when low physical demand is of particular importance. Only then is the Continuous Cursor Control the right choice.

CONCLUSION AND OUTLOOK

In this paper, we have studied text entry in VR using a virtual keyboard and discussed the design space including criteria

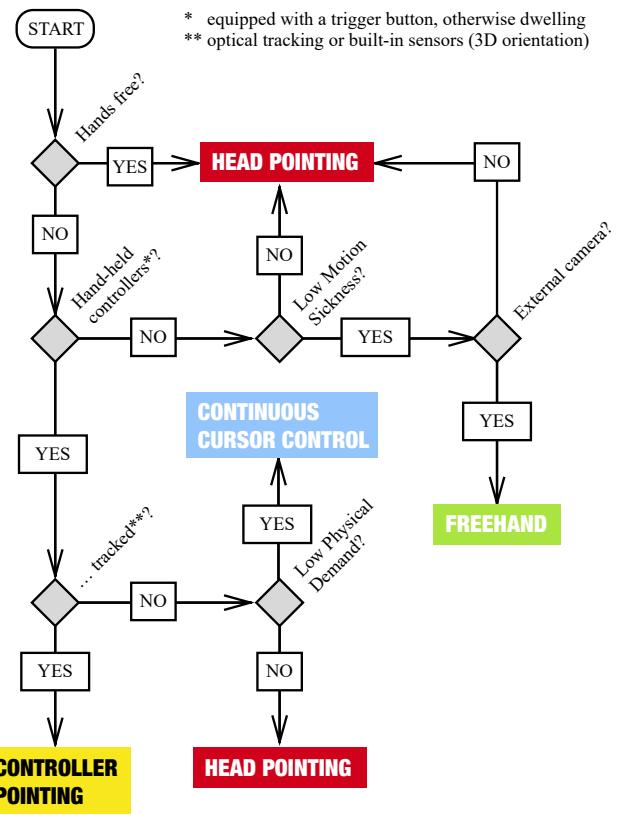


Figure 7. Decision support tool for VR text input on a virtual keyboard. Discrete Cursor is not considered, because of the bad results across all measurements. Controller Tapping performed slightly worse than Pointing, so it is not considered due to its higher technical requirements.

for assessing VR text entry methods. We have introduced six candidates that span different parts of the design space and evaluate their text entry performance and user preference. Although the general conclusion is to choose Controller Pointing for text entry in VR, its usage is dependent on certain criteria and limitations (e.g. tracked hand-held controllers). In addition, isomorphic keyboard interaction, as in the Freehand method, performed badly, even though it had promising user experience results. To sum up, and putting our findings together with related work and our design space, in this paper we present an example decision support tool in the form of a flowchart, so that the results can be easily used by future VR designers and researchers.

Text entry is an essential part of human computer interaction and there is still much research needed. Design annotation (e.g. for 3D artists or architects), filename entry or parameter setting, and communication between users are just a few applications for text entry in VR. Future VR systems (e.g. diaries, shops or social networks) may be designed to enable the user to stay in VR for longer times and therefore longer text entry needs to be feasible, too. Finally, the qualifying techniques need to be evaluated in the context of interactive immersive virtual environments.

REFERENCES

1. Jaewoo Ahn and Kyungha Min. 2006. VirtualPhonepad: a text input tool for virtual environments. *Advances in Artificial Reality and Tele-Existence* (2006), 58–64. DOI: http://dx.doi.org/10.1007/11941354_7
2. Ian G Angus and Henry A Sowizral. 1995. Embedding the 2D interaction metaphor in a real 3D virtual environment. In *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology*. International Society for Optics and Photonics, 282–293. DOI: <http://dx.doi.org/10.1117/12.205875>
3. Shiri Azenkot and Shumin Zhai. 2012. Touch Behavior with Different Postures on Soft Smartphone Keyboards. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 251–260. DOI: <http://dx.doi.org/10.1145/2371574.2371612>
4. Myroslav Bachynskyi, Gregorio Palmas, Antti Oulasvirta, and Tino Weinkauf. 2015. Informing the Design of Novel Input Methods with Muscle Coactivation Clustering. *ACM Trans. Comput.-Hum. Interact.* 21, 6, Article 30 (Jan. 2015), 25 pages. DOI: <http://dx.doi.org/10.1145/2687921>
5. Thomas Baudel and Michel Beaudouin-Lafon. 1993. Charade: Remote Control of Objects Using Free-hand Gestures. *Commun. ACM* 36, 7 (July 1993), 28–35. DOI: <http://dx.doi.org/10.1145/159544.159562>
6. Nikolaus Bee and Elisabeth André. 2008. Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. In *International Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems*. Springer, 111–122. DOI: http://dx.doi.org/10.1007/978-3-540-69369-7_13
7. Doug Bowman, Ernst Kruijff, Joseph J LaViola Jr, and Ivan P Poupyrev. 2004. *3D User Interfaces: Theory and Practice, CourseSmart eTextbook*. Addison-Wesley.
8. Doug A. Bowman, Ryan P. McMahan, and Eric D. Ragan. 2012. Questioning Naturalism in 3D User Interfaces. *Commun. ACM* 55, 9 (Sept. 2012), 78–88. DOI: <http://dx.doi.org/10.1145/2330667.2330687>
9. Doug A. Bowman, Christopher J. Rhoton, and Marcio S. Pinho. 2002. 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. DOI: <http://dx.doi.org/10.1177/154193120204602611>
10. Wei Ding, Ping Chen, Hisham Al-Mubaid, and Marc Pomplun. 2009. A Gaze-Controlled Interface to Virtual Reality Applications for Motor-and Speech-Impaired Users. *HCI International, San Diego, CA* (2009). DOI: <http://dx.doi.org/10.1.1.159.8055>
11. August Dvorak, Nellie L Merrick, William L Dealey, and Gertrude C Ford. 1936. Typewriting behavior. *New York*: American Book Company 1, 6 (1936). DOI: <http://dx.doi.org/10.1177/003151255100300300308>
12. Peter J Gianaros, Eric R Muth, J Toby Mordkoff, Max E Levine, and Robert M Stern. 2001. A questionnaire for the assessment of the multiple dimensions of motion sickness. *Aviation, Space, and Environmental Medicine* 72, 2 (2001), 115.
13. Yulia Gizatdinova, Oleg Špakov, and Veikko Surakka. 2012. Comparison of Video-based Pointing and Selection Techniques for Hands-free Text Entry. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. ACM, New York, NY, USA, 132–139. DOI: <http://dx.doi.org/10.1145/2254556.2254582>
14. Gabriel González, José P Molina, Arturo S García, Diego Martínez, and Pascual González. 2009. Evaluation of text input techniques in immersive virtual environments. In *New Trends on Human–Computer Interaction*. Springer, 109–118. DOI: <http://dx.doi.org/10.1007/978-1-84882-352-5>
15. Jan Gugenheimer, David Dobbelstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. 2016. FaceTouch: Enabling Touch Interaction in Display Fixed UIs for Mobile Virtual Reality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 49–60. DOI: <http://dx.doi.org/10.1145/2984511.2984576>
16. Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. 2010. Imaginary Interfaces: Spatial Interaction with Empty Hands and Without Visual Feedback. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 3–12. DOI: <http://dx.doi.org/10.1145/1866029.1866033>
17. S. G. Hart and L. E. Stavenland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Human Mental Workload*, P. A. Hancock and N. Meshkati (Eds.). Elsevier, Chapter 7, 139–183. http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20000004342_1999205624.pdf
18. Ken Hinckley, Mary Czerwinski, and Mike Sinclair. 1998. Interaction and Modeling Techniques for Desktop Two-handed Input. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST '98)*. ACM, New York, NY, USA, 49–58. DOI: <http://dx.doi.org/10.1145/288392.288572>
19. 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 (AVI '12)*. ACM, New York, NY, USA, 156–163. DOI: <http://dx.doi.org/10.1145/2254556.2254585>
20. Jason Jerald. 2015. The VR book: human-centered design for virtual reality. (2015).

21. Robin Kinkead. 1975. Typing Speed, Keying Rates, and Optimal Keyboard Layouts. *Proceedings of the Human Factors Society Annual Meeting* 19, 2 (1975), 159–161. DOI:<http://dx.doi.org/10.1177/154193127501900203>
22. Regis Kopper, Doug A. Bowman, Mara G. Silva, and Ryan P. McMahan. 2010. A human motor behavior model for distal pointing tasks. *International Journal of Human-Computer Studies* 68, 10 (2010), 603 – 615. DOI:<http://dx.doi.org/https://doi.org/10.1016/j.ijhcs.2010.05.001>
23. Per Ola Kristensson and Keith Vertanen. 2012. Performance Comparisons of Phrase Sets and Presentation Styles for Text Entry Evaluations. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces (IUI '12)*. ACM, New York, NY, USA, 29–32. DOI:<http://dx.doi.org/10.1145/2166966.2166972>
24. Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and evaluation of a user experience questionnaire. In *Symposium of the Austrian HCI and Usability Engineering Group*. Springer, 63–76. DOI:http://dx.doi.org/10.1007/978-3-540-89350-9_6
25. Anatole Lecuyer. 2009. Simulating Haptic Feedback Using Vision: A Survey of Research and Applications of Pseudo-Haptic Feedback. *Presence: Teleoperators and Virtual Environments* 18, 1 (2009), 39–53. DOI:<http://dx.doi.org/10.1162/pres.18.1.39>
26. Kun Chang Lee and Namho Chung. 2008. Empirical analysis of consumer reaction to the virtual reality shopping mall. *Computers in Human Behavior* 24, 1 (2008), 88 – 104. DOI:<http://dx.doi.org/10.1016/j.chb.2007.01.018>
27. Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. 707–710.
28. Jia-Wei Lin, Ping-Hsuan Han, Jiun-Yu Lee, Yang-Sheng Chen, Ting-Wei Chang, Kuan-Wen Chen, and Yi-Ping Hung. 2017. Visualizing the Keyboard in Virtual Reality for Enhancing Immersive Experience. In *ACM SIGGRAPH 2017 Posters (SIGGRAPH '17)*. ACM, New York, NY, USA, Article 35, 2 pages. DOI:<http://dx.doi.org/10.1145/3102163.3102175>
29. Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E. W. Looney. 2004. Twiddler Typing: One-handed Chording Text Entry for Mobile Phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 671–678. DOI:<http://dx.doi.org/10.1145/985692.985777>
30. I. Scott MacKenzie. 2002. *KSPC (Keystrokes per Character) as a Characteristic of Text Entry Techniques*. Springer Berlin Heidelberg, Berlin, Heidelberg, 195–210. DOI:http://dx.doi.org/10.1007/3-540-45756-9_16
31. I. Scott MacKenzie and Kumiko Tanaka-Ishii. 2007. *Text Entry Systems: Mobility, Accessibility, Universality*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
32. I. Scott MacKenzie and Shawn X. Zhang. 1999. The Design and Evaluation of a High-performance Soft Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 25–31. DOI:<http://dx.doi.org/10.1145/302979.302983>
33. Allan MacLean, Richard M. Young, Victoria M. E. Bellotti, and Thomas P. Moran. 1991. Questions, Options, and Criteria: Elements of Design Space Analysis. *Hum.-Comput. Interact.* 6, 3 (Sept. 1991), 201–250. DOI:http://dx.doi.org/10.1207/s15327051hci0603&4_2
34. Päivi Majaranta and Kari-Jouko Räihä. 2007. Text entry by gaze: Utilizing eye-tracking. *Text entry systems: Mobility, accessibility, universality* (2007), 175–187. <http://citeseervx.ist.psu.edu/viewdoc/summary?doi=10.1.1.102.5479>
35. Anders Markussen, Mikkel R Jakobsen, and Kasper Hornbæk. 2013. Selection-based mid-air text entry on large displays. In *IFIP Conference on Human-Computer Interaction*. Springer, 401–418. DOI:http://dx.doi.org/10.1007/978-3-642-40483-2_28
36. 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>
37. Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. 2011. Mid-air Pan-and-zoom on Wall-sized Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 177–186. DOI:<http://dx.doi.org/10.1145/1978942.1978969>
38. Jan Noyes. 1983. The QWERTY keyboard: A review. *International Journal of Man-Machine Studies* 18, 3 (1983), 265–281. DOI:[http://dx.doi.org/10.1016/S0020-7373\(83\)80010-8](http://dx.doi.org/10.1016/S0020-7373(83)80010-8)
39. Gang Ren and Eamonn O'Neill. 2013. Freehand Gestural Text Entry for Interactive TV. In *Proceedings of the 11th European Conference on Interactive TV and Video (EuroITV '13)*. ACM, New York, NY, USA, 121–130. DOI:<http://dx.doi.org/10.1145/2465958.2465966>
40. Garth Shoemaker, Leah Findlater, Jessica Q. Dawson, and Kellogg S. Booth. 2009. Mid-air Text Input Techniques for Very Large Wall Displays. In *Proceedings of Graphics Interface 2009 (GI '09)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 231–238. <http://dl.acm.org/citation.cfm?id=1555880.1555931>

41. Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. 2015. Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3643–3652. DOI: <http://dx.doi.org/10.1145/2702123.2702136>
42. Keith Vertanen and Per Ola Kristensson. 2011. 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, 295–298. DOI: <http://dx.doi.org/10.1145/2037373.2037418>
43. James Walker, Bochao Li, Keith Vertanen, and Scott Kuhl. 2017. Efficient Typing on a Visually Occluded Physical Keyboard. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5457–5461. DOI: <http://dx.doi.org/10.1145/3025453.3025783>
44. Frank Weichert, Daniel Bachmann, Bartholomus Rudak, and Denis Fisseler. 2013. Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors* 13, 5 (2013), 6380–6393. DOI: <http://dx.doi.org/10.3390/s130506380>
45. Andrew D. Wilson and Maneesh Agrawala. 2006. Text Entry Using a Dual Joystick Game Controller. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 475–478. DOI: <http://dx.doi.org/10.1145/1124772.1124844>
46. Jacob O Wobbrock. 2007. *Measures of text entry performance*. San Francisco: Morgan Kaufmann. DOI: <http://dx.doi.org/10.1016/B978-012373591-1/50003-6>
47. Jacob O. Wobbrock, Duen Horng Chau, and Brad A. Myers. 2007. An Alternative to Push, Press, and Tap-tap-tap: Gesturing on an Isometric Joystick for Mobile Phone Text Entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 667–676. DOI: <http://dx.doi.org/10.1145/1240624.1240728>
48. Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. 2017. Tap, Dwell or Gesture?: Exploring Head-Based Text Entry Techniques for HMDs. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4479–4488. DOI: <http://dx.doi.org/10.1145/3025453.3025964>
49. Shumin Zhai. 1995. Human Performance in Six Degree of Freedom Input Control. (1995).